



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

SISTEMA DE ADMINISTRACIÓN, CONTROL
Y ATENCIÓN DE LAS SALAS DE UNICA
(SCOSU)

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

P R E S E N T A:

VÍCTOR DUEÑAS TELLO



DIRECTORA DE TESIS:

ING. MARÍA DEL ROSARIO BARRAGÁN PAZ

CIUDAD UNIVERSITARIA, MÉXICO D. F.

2009

Agradecimientos

A lo largo de mi vida y de mi carrera existieron varias personas que me apoyaron para poder realizar este sueño, un sueño que en un principio se veía largo e inalcanzable y que hoy después de varios años y un gran esfuerzo por fin se ha vuelto realidad.

Gracias a mis padres Fernando Dueñas Salazar y Justina Tello Navarrete que han sido mi más grande apoyo a lo largo de mi vida y de mi carrera. Yo sé que para mi papá fue un esfuerzo sobrehumano poder mantener mi carrera, pero te doy gracias de todo corazón por haberlo hecho, gracias por enseñarme a nunca darme por vencido con ese esfuerzo que realizas cada día al levantarte temprano e ir a trabajar, créeme, me gustaría poder ser al menos la mitad de trabajador de lo que tú eres. Gracias a ti mamá, tú que siempre has estado ahí en mis momentos más difíciles para brindarme todo tu apoyo y palabras de aliento, desde niño cuando me sentía mal siempre me reconfortaron tus abrazos y aun hoy, cuando no me siento bien, un abrazo tuyo sigue siendo muy cálido, en verdad no sabes la paz que me da tu compañía. Gracias a ambos por esas charlas que tuvimos donde yo les contaba mis metas, sueños y fantasías. Esta tesis es el fruto del esfuerzo de ambos es por eso que esta tesis es para ustedes. Y por favor nunca lo olviden son las personas más importantes de mi vida.

Gracias a mi hermana Maribel que fue una de las principales responsables de que yo terminara estudiando Ingeniería, ella me indico el camino del bien. Gracias a mi hermana Araceli que siempre ha sabido escucharme y aconsejarme. Gracias a ambas por todo el apoyo y palabras de aliento que siempre recibí de ustedes. Las quiero mucho.

Gracias en general a toda mi familia, incluyendo a mis sobrinos Juan Carlos y Adrian por soportar tantos años de continuas desveladas.

Gracias a Chary por guiarme en esta muy larga carrera llamada tesis y por brindarme todo su apoyo y amistad. Gracias a mi jefa Bety por confiar en mí y darme la oportunidad de crecer como programador y por ayudarme en cierto momento... salvo mi navidad.

Gracias a la gloriosa Facultad de Ingeniería y a la UNAM por abrirme las puertas y hacer de mí un excelente profesionalista.

Gracias a UNICA que me doto de las herramientas necesarias para poder enfrentar cara a cara el mundo laboral y por haberme dado la oportunidad de conocer a Claus, la persona que cambiaria mi vida.

Gracias a la comunidad más grande y respetable de la Facultad de Ingeniería “Los Campanudos” y especialmente a Lidia, Natalia, Zamora, Carry, Rolando, Joe, Haro, Niño y Nakamura gracias a su apoyo y compañía la carrera fue más fácil y divertida, juntos pasamos momentos inolvidables.

Gracias a mis amigos Adolfo, José, Pancho y Chava personas que conocí al entrar a la prepa 4 y que se volvieron una parte indispensable en mi vida, gracias por todo su apoyo, jamás voy a olvidar esa etapa de mi vida, son los mejores amigos que pude haber encontrado.

*Por último quiero agradecer a la mujer que entro en mi vida y en mi corazón, mi niña, **Claudia Nelly Arriaga Hernández**, te doy gracias por enseñarme a ver al mundo de diferente manera, por haber cambiado mi vida por completo, eres esa parte que le hacía falta a mi alma, simplemente eres el corazón de mi alma. Llegaste cuando más te necesitaba y supiste brindarme todo tu amor, ese amor que me llena de dicha y felicidad, gracias por hacerme parte de tu vida y por todos esos momentos llenos de magia que hemos pasado juntos. Muchas gracias por el apoyo incondicional que me brindaste a lo largo de toda mi tesis y por tus palabras de aliento cuando me frustraba, siempre has sido muy buena para tranquilizarme y hacerme reír. En verdad no se qué sería de mí si no te hubiera conocido y ni quiero imaginarlo.*

***TE AMO** mi amor y así va ser toda mi vida, esa vida que deseo de todo corazón pasarla a tu lado.*

ÍNDICE

INTRODUCCIÓN.....	1
Antecedentes	1
UNICA	1
Estructura organizacional y funciones.....	4
CAPÍTULO 1 . DEFINICIÓN DEL PROYECTO.....	12
1.1 Objetivo	12
1.1.1 Objetivo general.....	12
1.1.2 Objetivos Particulares	12
1.2 Alcance	13
1.3 Análisis y problemática actual.....	13
1.3.1 Problemas y causas.....	16
1.4 Requerimientos por parte de los departamentos	17
1.4.1 Departamento de Servicios Académicos (DSA)	17
1.4.2 Coordinación de Salas Computo (CSC)	18
1.5 Evaluación y determinación de costos	21
1.6 Infraestructura.....	22
1.7 Beneficios.....	22
CAPÍTULO 2. ANÁLISIS Y DISEÑO	24
2.1 Propuesta tecnológica	26
2.2 Modelado del sistema.....	26
2.2.1 Diagrama de casos de uso.....	33
2.2.2 Diagrama de clases	49
2.2.3 Diagrama de actividades	55
2.3 Diseño de la Base de datos	63
2.3.1 Identificación de entidades.....	63
2.3.2 Descripción de entidades	64
2.3.3 Relaciones entre entidades	65
2.3.4 Diseñando la Base de datos	69
2.3.5 Esquema de la Base de datos	75
2.4 Diseño del sistema	76
2.4.1 Visual Basic 2005.....	76
2.4.2 ADO .Net.....	85
2.4.2.1 Clases de ADO .Net.....	88

CAPÍTULO 3. DESARROLLO.....	91
3.1 Implementación de la Base de datos	91
3.1.1 Herramienta CASE.....	91
3.1.2 Creación de la Base de datos.....	93
3.2 Programación de la aplicación en Visual Basic 2005.....	95
3.2.1 SCOSU v2	95
3.2.2 SCOSU v2 Modulo del Jefe	97
3.3 Integración de la aplicación con los diferentes sistemas de UNICA	99
3.3.1 Sistema de Inscripción de Acceso a las Salas de UNICA (SIASU).....	99
3.3.2 Sistema de Control de Impresiones.....	100
3.3.3 Sistema de Control de Asesorías.....	100
CAPÍTULO 4. PRUEBAS.....	101
4.1 Objetivos de las pruebas.....	102
4.2 Diseño de casos de pruebas.....	103
4.3 Pruebas de caja blanca	104
4.3.1 Aplicación de pruebas de caja blanca a SCOSU v2.....	107
4.4 Pruebas de caja negra.....	111
4.4.1 Aplicación de pruebas de caja negra a SCOSU v2	116
CAPÍTULO 5. IMPLEMENTACIÓN.....	123
5.1 Elaboración de manuales	123
5.1.1 Manual de instalación	124
5.1.1.1 Creación del archivo de instalación de SCOSU v2	124
5.1.1.2 Creación del archivo de instalación de SCOSU v2 Modulo del Jefe.....	131
5.1.1.3 Instalación de SCOSU v2	138
5.1.1.4 Instalación de SCOSU v2 Modulo del Jefe	142
5.1.1.5 Instalación de Microsoft .Net Framework	144
5.1.2 Manual del usuario.....	147
5.1.2.1 Manual del usuario de SCOSU v2.....	147
5.1.2.1.1 Antes de entrar al sistema	147
5.1.2.1.2 Entrando al sistema	147
5.1.2.1.3 Las salas.....	148
5.1.2.1.3.1 Partes de la sala.....	149
5.1.2.1.4 Estados de los equipos	151
5.1.2.1.5 Asignación de equipo	153
5.1.2.1.5.1 Alumno duplicado.....	155
5.1.2.1.6 Tiempo de uso y renovación de tiempo	156
5.1.2.1.7 Estado Disponible y Estados Extras.....	156
5.1.2.1.7.1 Colocando disponible el equipo.....	156
5.1.2.1.7.2 Estados Extras	157

5.1.2.1.7.3 El mouse y los estados extras	160
5.1.2.1.8 Lista de Equipos Asignados y Equipos Liberados	160
5.1.2.1.9 Usuarios de las computadoras	161
5.1.2.1.10 Destinar sala.....	162
5.1.2.1.10.1 Aula en Curso	162
5.1.2.1.10.1.1 Nombre del curso	163
5.1.2.1.10.1.2 Fecha y hora de inicio	163
5.1.2.1.10.1.3 Fecha y hora de término	164
5.1.2.1.10.1.4 Seleccionando el aula.....	164
5.1.2.1.10.2 Inicio y fin del curso	165
5.1.2.1.10.2.1 Hora de inicio del curso	165
5.1.2.1.10.2.2 En el curso	166
5.1.2.1.10.2.3 Hora de fin del curso.....	167
5.1.2.1.10.3 Aula en Mantenimiento.....	167
5.1.2.1.11 Paquetería	169
5.1.2.1.12 Ver Sancionados	171
5.1.2.1.13 Ver Cursos	171
5.1.2.1.14 Sancionar Alumno.....	172
5.1.2.1.14.1 Sancionar un alumno cuando NO se encuentra utilizando un equipo.....	172
5.1.2.1.14.1.1 Datos del Sancionado	172
5.1.2.1.14.1.2 Datos de la Sanción	173
5.1.2.1.14.2 Sancionar un alumno cuando se encuentra utilizando un equipo.....	176
5.1.2.1.14.3 Termino de la sanción.....	177
5.1.2.1.15 Restaurar Sistema	178
5.1.2.1.16 Modo desconectado.....	178
5.1.2.1.16.1 Cambiando a Modo Desconectado	179
5.1.2.1.16.2 Cambiando a Modo Conectado.....	180
5.1.2.1.17 Inscribir Alumno a SCOSU v2.....	181
5.1.2.1.18 Ayuda de SCOSU v2.....	182
5.1.2.1.19 Contactar al Administrador	182
5.1.2.1.20 Acerca de SCOSU v2	185
5.1.2.1.21 Barra de herramientas.....	186
5.1.2.1.22 Buscar alumno	186
5.1.2.1.22.1 Búsqueda de alumno mediante su nombre	187
5.1.2.1.22.2 Búsqueda de alumno mediante su número de cuenta	188
5.1.2.1.23 Vista previa.....	189
5.1.2.2 Manual del usuario de SCOSU v2 Modulo del Jefe	191
5.1.2.2.1 Entrando al sistema	191
5.1.2.2.2 Reportes	192
5.1.2.2.2.1 Reportes por nombre y número de cuenta	193
5.1.2.2.2.2 Reportes por día e historial.....	195
5.1.2.2.2.3 Herramientas del reporte	196
5.1.2.2.3 Salir del sistema	198
5.1.2.2.4 Actualizar software	198
5.1.2.2.5 Agregar software nuevo.....	208
5.1.2.2.6 Cursos.....	212
5.1.2.2.7 Sancionados	214
5.1.2.2.8 Jefe de salas	215

5.1.2.2.9 Sala 3	217
5.1.2.2.10 Software	219
5.2 Capacitación	221
5.2.1 Capacitación del administrador	221
5.2.2 Capacitación de los usuarios	221
5.3 Instalación	222
5.4 Mantenimiento	223
CONCLUSIONES Y RESULTADOS	224
Conclusiones	224
Acciones a corto plazo	225
Bibliografía y Mesografía	227

INTRODUCCIÓN

Antecedentes

➤ UNICA

En 1994 el Centro de Cómputo Académico cambió su nombre de Centro de Cómputo Académico a Unidad de Servicios de Cómputo Académico (UNICA) quedando organizado en tres departamentos:



Figura 1.2 Estructura de UNICA hasta 1994

En el año 2000, se reestructuró la organización de UNICA y se modificó el nombre del Departamento de Cómputo Académico debido a las actividades que actualmente realiza, de tal forma que se nombro como Departamento de Redes y Operación de Servidores.

En el año 2003 se creó el Departamento de Seguridad en Cómputo el cual se puso en marcha en el año 2004.

Las funciones que desempeña la Unidad de Servicios de Cómputo Académico son:

- Mantener el liderazgo en cuanto a tópicos en cómputo.
- Continuar proporcionando recursos de cómputo de calidad a la comunidad de la Facultad.
- Impulsar a nivel de la Facultad la creación de una política de cómputo definida.
- Lograr la capacitación cada vez más completa y actualizada para la formación de recursos humanos.
- Aplicar todos los conocimientos y las herramientas de cómputo con los que cuenta la Unidad para realizar las actividades de forma más eficiente y segura.

Política de calidad

El objetivo de la Unidad de Servicios de Cómputo Académico (UNICA) es cumplir con los requisitos de los usuarios en el área de cómputo, teniendo como meta elevar la calidad en los productos y servicios que proporciona, para ello se comprometen en un proceso de mejora continua.

Misión

Proporcionar eficaz y eficientemente en el ámbito institucional, los servicios de cómputo y el apoyo en actividades relacionadas que conlleven al proceso integral de formación académica de la Facultad de Ingeniería.

Visión

La proyección de la Unidad de Servicios de Cómputo Académico al año 2011 es continuar siendo una Unidad líder en la prestación de servicios de cómputo de vanguardia a la Facultad de Ingeniería, al entorno universitario y a la sociedad en general.

- Siendo una Unidad líder en la prestación de servicios de cómputo de vanguardia a la Facultad de Ingeniería.
- Contando con la organización, administración y los recursos adecuados.
- Siendo líderes en la formación, capacitación y difusión de la cultura informática.
- Contando con las herramientas y convenios adecuados para el desarrollo y la investigación informática.
- Contando con los servicios y procesos de atención sistematizados y actuales en apoyo a los eventos de seguridad informática.
- Contando con la infraestructura adecuada y mecanismos para la actualización continúa del equipo de cómputo.

Valores

El ambiente de trabajo de los integrantes de la Unidad de Servicios de Cómputo Académico se basa en un clima de cordialidad, respeto, honestidad, responsabilidad, ética y compromiso.

Objetivos

Dentro de la Unidad de Servicios de Cómputo Académico se tiene bien identificados 4 objetivos esenciales:

1. Proporcionar a nivel institucional los servicios de cómputo académico a la comunidad de la Facultad, en apoyo al fortalecimiento del proceso enseñanza-aprendizaje.
2. Formar recursos humanos de calidad en el ámbito académico y profesional.
3. Mantener a la vanguardia en materia de cómputo en la Institución a través de la investigación y desarrollo, e incorporar a la actividad de la Facultad nuevas tendencias de cómputo.
4. Apoyar a Secretaría General en las actividades relacionadas al cómputo que involucren institucionalmente a la Facultad de Ingeniería.

Metas

1. Optimización de los recursos disponibles para la atención total de los servicios de cómputo académico solicitados por la comunidad de la Facultad.
2. Fortalecer el Programa de Formación de Becarios para capacitación.
3. Actualizar e incorporar en la Facultad nuevas tendencias en materia de cómputo mediante acciones de investigación y desarrollo.
4. Atención y seguimiento al total de las solicitudes de información y de servicios de cómputo requeridos a la Facultad interna y externamente.

➤ Estructura organizacional y funciones

Organigrama

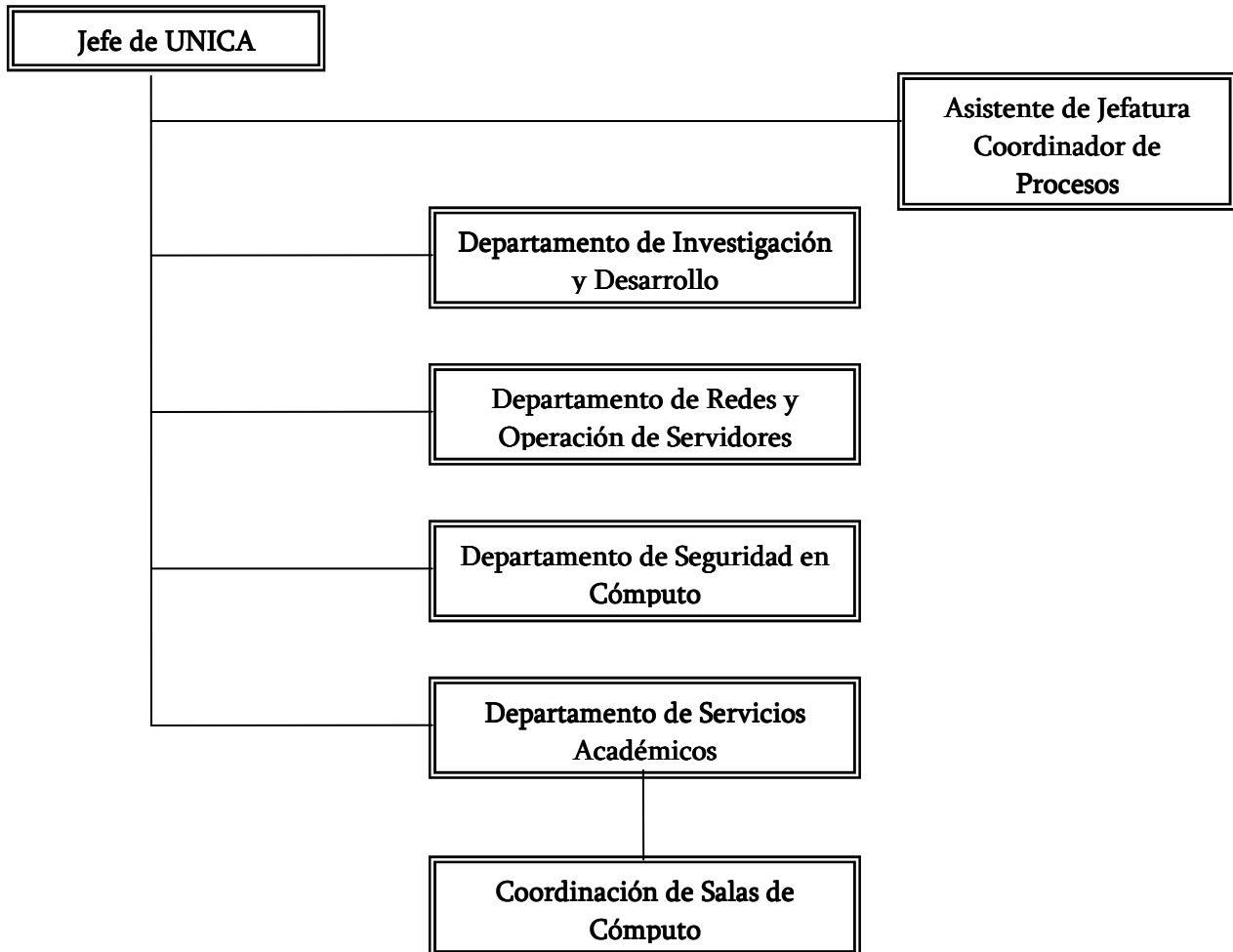


Figura 1.3 Organigrama de UNICA

Jefe de UNICA

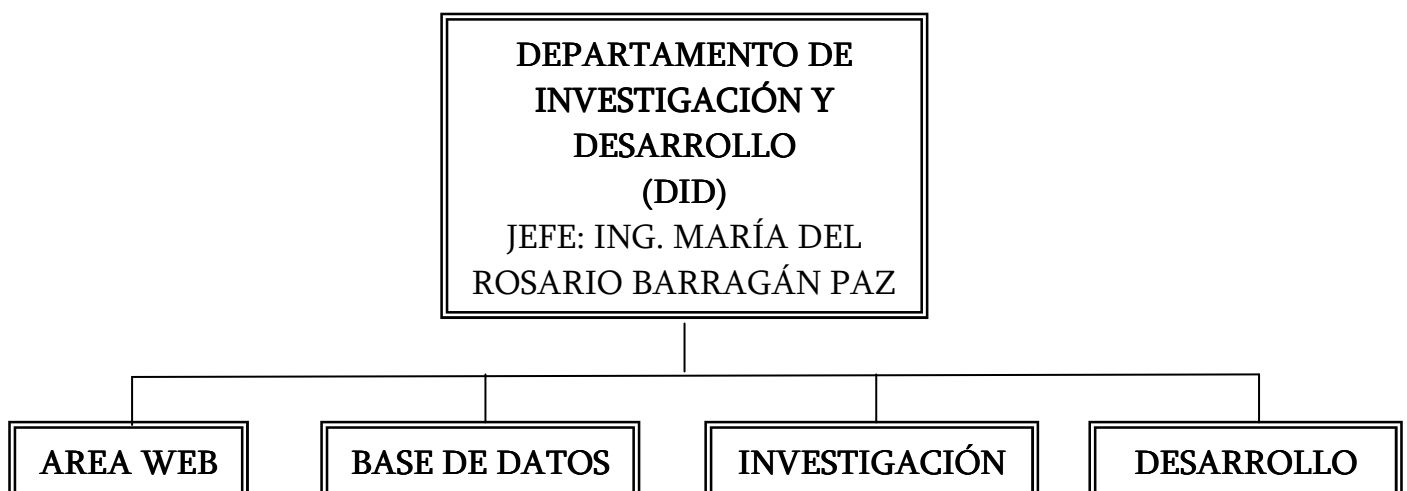
Tiene como funciones principales:

- A. Seleccionar al personal conforme la legislación universitaria y posteriormente proponer al Secretario General las personas idóneas para ocupar puestos académico-administrativo y de confianza; así mismo, es miembro del comité del programa “Plan de becarios” ocupando el cargo de Secretario.

- B. Supervisar el funcionamiento académico y administrativo de UNICA.

- C. Definir e implantar políticas de calidad en el servicio, planeación de proyectos, actividades y asignación de responsabilidades a cada jefe de departamento.
- D. Supervisar el desarrollo de proyectos y actividades analizando con los jefes de departamento los planes de trabajo e informes correspondientes.
- E. Administrar los recursos de cómputo de UNICA y establecer relaciones con proveedores de equipo de cómputo.
- F. Brindar apoyo a la dirección para cambio o ampliación del equipo que cubra los requerimientos de la Facultad de Ingeniería; esta atribución está en virtud de que realiza la investigación de características y especificaciones requeridas por las divisiones que componen la Facultad de Ingeniería.
- G. Responsable de llevar el inventario del equipo de cómputo con la colaboración de las áreas de Secretaria General.
- H. Proponer presupuesto anual para UNICA.
- I. Planear los apoyos Institucionales como el Censo, el Inventario y la Administración de la Partida Centralizada para la compra de equipo de Cómputo de UNICA.

Departamento de Investigación y Desarrollo (DID)

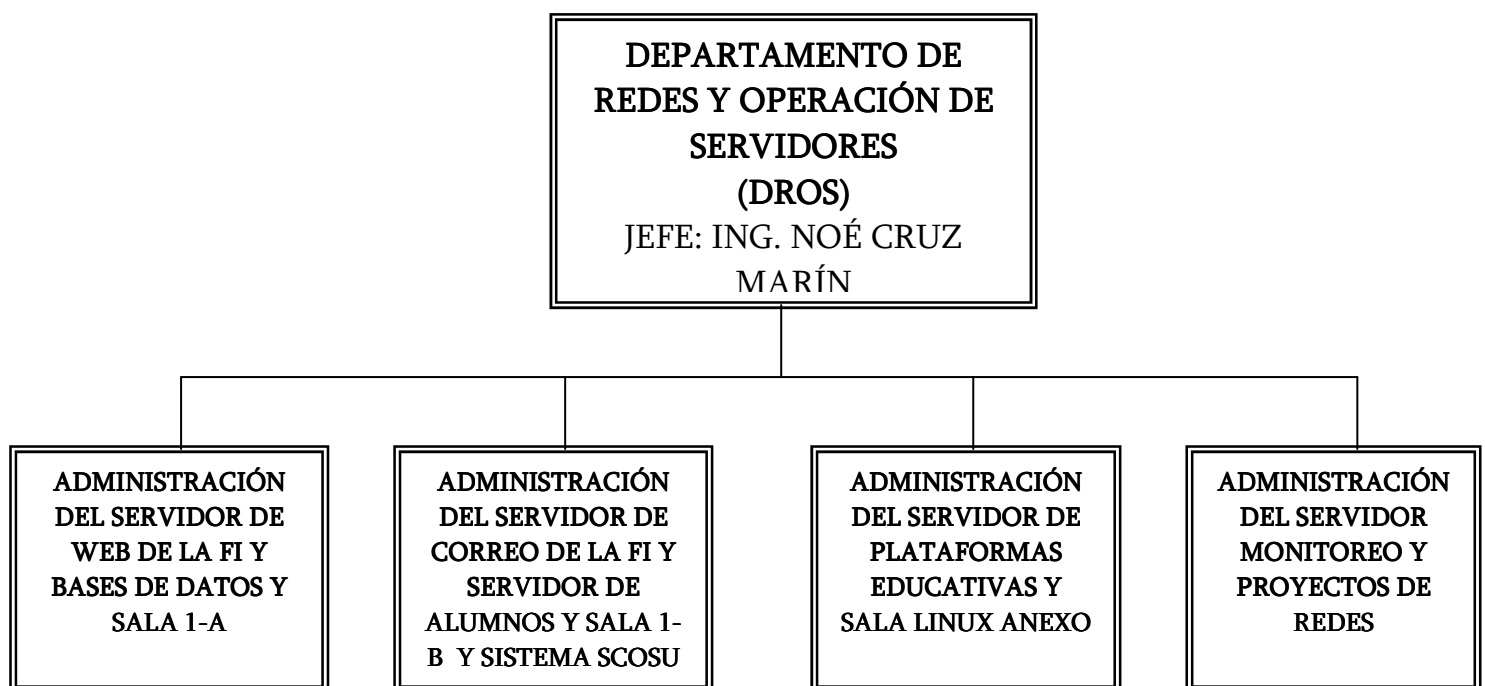


Sus funciones son:

- A. Desarrollo de sistemas de cómputo que requiere tanto UNICA como la Secretaría General.
- B. Investigación del software comercial y especializado que la actividad de la FI requiere y da servicio de asesoría sobre estos tópicos.
- C. Coordinación del área de multimedia
- D. Realiza actividades comunes a UNICA como: impartición de cursos, elaboración de material didáctico y asesorías.
- E. Diseño de Sistemas de Cómputo para solución de problemas.
- F. Coordinar las actividades de investigación en cómputo.
- G. Actualización constante para estar al día en cuanto a la tecnología y avances en materia de cómputo.
- H. Elaboración de artículos de difusión referentes al cómputo.
- I. Manejo, aplicación de métodos y técnicas para el desarrollo de sistemas.
- J. Investigación y desarrollo en el área Web Aplicaciones CAD.
- K. Responsables del Sistema de gestión de documentos SIGED.
- L. Investigación de nuevas tecnología en software desarrollo en Java, Javascript y Perl.
- M. Responsables de los Sistemas Control de Asesorías SIAA Sistema de Información del personal de la Facultad de Ingeniería SCIFI.
- N. Investigación metodología para el desarrollo de sistemas y modelado de Bases de Datos.
- O. Investigación de nuevas tecnología en software desarrollo en Java.
- P. Sistema de control de documentos SCD.

- Q. Responsables de los sistemas: Control de inventarios SICI. Control y Evaluación Académico para los becarios de la Unidad SYCEB.
- R. Responsables del Sistema de control de impresiones de UNICA SICISU y el Sistema de Información y Difusión para el Centro de Docencia SICED.
- S. Investigación y desarrollo en Lenguajes de Programación metodología para el desarrollo de sistemas.
- T. Investigación y desarrollo MoProsoft.
- U. Investigación y desarrollo en Lenguajes de Programación metodología para el desarrollo de sistemas. Sistema de control de impresiones de UNICA SICISU.
- V. Sistema de Información del Personal de la Facultad de Ingeniería SCIFI.
- W. Investigación y desarrollo en Lenguajes de Programación metodología para el desarrollo de sistemas.
- X. Sistema de Control de inventarios de equipo de cómputo de la Facultad de Ingeniería SICI.

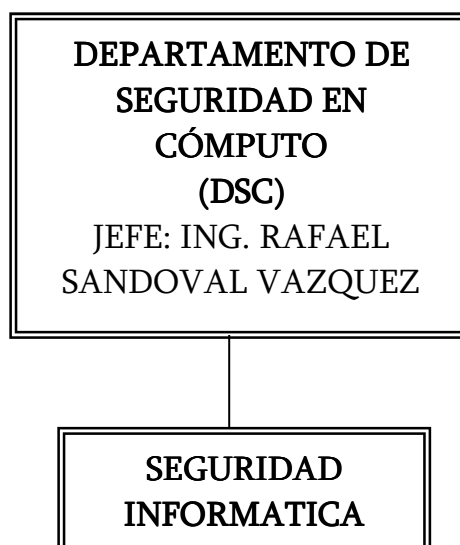
Departamento de Redes y Operación de Servidores (DROS)



Sus principales funciones son:

- A. Administración del equipo de la Sala de Cómputo 1.
- B. Administración de la red interna de UNICA
- C. Colaboración en el diseño e implementación de la red de cómputo de la Facultad de Ingeniería.
- D. Apoyo en el soporte técnico a Secretarías y Divisiones de la Facultad de Ingeniería.
- E. Actividades comunes a UNICA como impartición de cursos, asesorías y desarrollo de material didáctico.
- F. Administración del servidor de correo de la Facultad de Ingeniería y el servidor de alumnos y la sala 1-B y sistema SCOSU.
- G. Administración del servidor de plataformas educativas y sala Linux Anexo.
- H. Administración del servidor monitoreo y proyecto de redes.

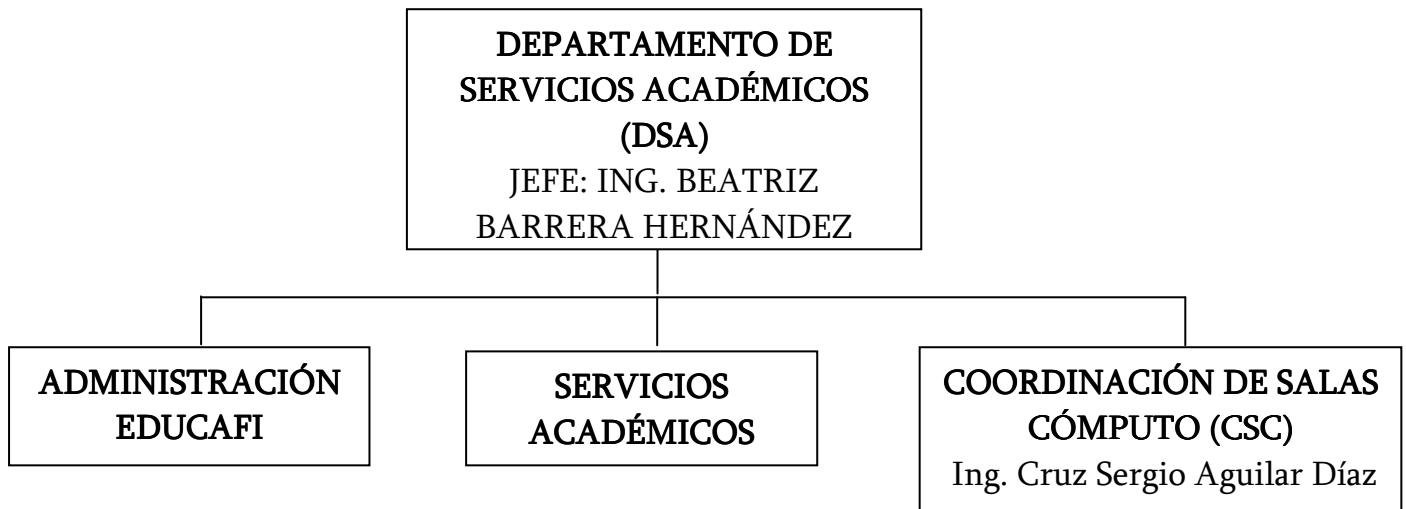
Departamento de Seguridad en Cómputo (DSC)



Sus funciones son:

- A. Administración de la Seguridad Informática.
- B. Manejo de incidentes.
- C. Identificar nuevos tipos de métodos de ataque.
- D. Proporcionar soporte técnico para respuesta de incidentes de Seguridad.
- E. Alertas y Anuncios.
- F. Manejo de Vulnerabilidades.
- G. Sistemas Detectores de Intrusos.
- H. Revisar y monitorear el ambiente de red existente para establecer una línea base de actividad de la red, con objeto de tener un punto de comparación en contra de potenciales anomalías.
- I. Educación y Capacitación. Crear una cultura de seguridad con cursos de capacitación, capacitación de los elementos del equipo de forma organizada y constante.
- J. Auditorias.
- K. Colaboración y coordinación con otros equipos de respuesta a incidentes en cómputo y áreas de seguridad.

Departamento de Servicios Académicos (DSA)



Funciones DSA:

- A. La planeación, organización y coordinación de los cursos de computación que imparte UNICA.
- B. Administra el Programa de “Formación de Becarios” de UNICA.
- C. Evaluación y rendimiento de los becarios de UNICA.
- D. Responsable del servicio de las Salas de Cómputo.
- E. Responsable de la capacitación interna.
- F. Responsable de la actualización del material didáctico de los cursos de cómputo.
- G. Responsable del servicio de la Plataforma Educativa EDUCAFI para los profesores de la Facultad de Ingeniería.
- H. Actividades comunes a UNICA como: Impartición de cursos, elaboración de material didáctico y asesorías.
- I. Coordinación del Servicio Social de UNICA.
- J. Responsable del Sistema de Gestión de Calidad de UNICA.

Funciones CSC:

- A. Coordinación y Administración de Personal.
- B. Administración, Configuración y funcionamiento de equipo de Cómputo y de Servidores de aplicaciones.
- C. Responsable del Mejoramiento del servicio y actualización de equipo e instalaciones.

CAPÍTULO 1

DEFINICIÓN DEL PROYECTO

El desarrollo de este sistema está fundamentado por la necesidad de mejorar el control de acceso a las salas de cómputo de UNICA, esto mediante la agilización del servicio al alumno de la Facultad de Ingeniería y la realización de estadísticas de uso del equipo. Por otra parte surge la necesidad de actualizarlo respecto al lenguaje de programación ya que los recursos y actualizaciones con los que disponía se están volviendo obsoletos debido a los avances que se tienen hoy en día.

1.1 Objetivo

1.1.1 Objetivo general

Diseñar un sistema para la administración, atención y control de las Salas de Computo de la Unidad de Servicios de Cómputo Académico.

1.1.2 Objetivos Particulares

1. Desarrollar un sistema para el control de acceso de los usuarios a las salas de cómputo de UNICA.

2. Obtener estadísticas que son solicitadas por el Departamento de Servicios Académicos.
3. Generar reportes: Anuales, Semestrales, por sala y por usuario.
4. Integración con los sistemas Control de impresiones, Checador Y Control de asesorías.
5. Desarrollar la documentación: Manual de operación, manual de usuario, manual del programador.
6. Cambiar el lenguaje de programación a programación orientada a objetos.
7. Realizar las conexiones a la base de datos de una manera eficiente, obteniendo así un sistema más eficaz para el control de los alumnos a las salas de UNICA.
8. Cambiar la interfaz grafica del sistema SCOSU para que sea más amigable para el usuario además de adaptarlo a los nuevos sistemas operativos como Windows Vista.

1.2 Alcance

El alcance de esta tesis consiste en mejorar en un cien por ciento la antigua versión del sistema SCOSU que cumpla con los objetivos anteriormente señalados. En resumen el sistema, aportará una mejora a la forma en la que se lleva el control de la información y realizar reportes detallados de forma automática.

1.3 Análisis y problemática actual

Durante muchos años UNICA ha ofrecido sus servicios a la comunidad de la Facultad de Ingeniería, entre los que destacan la preparación y conocimientos que se les brinda a los becarios mediante cursos de programación, los cuales ponen en práctica en los diferentes proyectos que se les asigna dentro de la unidad. Por otro lado se encuentran los cursos que se les da a las personas externas y propias de la facultad que son impartidos por los propios becarios de la unidad, estos han recibido una capacitación como instructores para poder brindar un excelente curso y de esa manera satisfacer a las personas que llegan a tomarlos.

Cabe destacar el servicio que mas a ayudado a los alumnos de la Facultad el cual es el de préstamos de equipos de cómputo, tomando en cuenta que se viene ofreciendo desde hace muchos años, es decir, desde que no era tan fácil tener una computadora en nuestros hogares.

Analizando el párrafo anterior podemos ver que el préstamo de computadoras ha beneficiado a miles de alumnos durante años, alumnos que no tienen los recursos necesarios para tener una computadora personal y aunque ya las computadoras personales son mas económicas siguen existiendo personas que no pueden pagarlas. Pero también podemos verlo desde otro punto de vista, por ejemplo, a algún alumno se le olvido una tarea que tiene que ser hecha a computadora además debe de ser entregada ese mismo día, puede ir a cualquiera de las salas de UNICA y realizarla, de esa manera cumplirá con todos sus deberes en las materias.

Como podemos ver son muchos los beneficios que se tienen gracias al préstamo de equipos de cómputo como realizar proyectos, tareas etc. Pero para dar un buen servicio a los miles de alumnos de la facultad es necesario tener una buena cantidad de computadoras y UNICA las tiene.

Cuenta con cuatro salas, una en el Edificio Principal ubicada debajo de los cubículos de los becarios de UNICA, dos en el área sur ubicadas en el Anexo de Ingeniería, una de ellas abajo del auditorio Sotero Prieto, mientras que la otra se encuentra donde anteriormente estaba la antigua biblioteca y una más en el primer piso del edificio de postgrado, sumando todas las salas tenemos que cuenta con más de 250 computadoras personales, 9 impresoras como equipo al servicio exclusivo de los alumnos el cual se complementa con servidores y las computadoras necesarias para el control de asignación y administración de los equipos de computo, las cuales son operadas por el personal de salas encargados de dar mantenimiento a los equipos y manejan el sistema SCOSU para poder realizar dichas tareas.

El horario de las salas es de 9 de la mañana a 8:30 de la noche, y como servicio extra tiene asesoría especializada brindada por becarios, prácticamente durante todo el día y en varias ramas, desde programación en diferentes lenguajes hasta problemas técnicos relacionados con equipos de cómputo

Los sistemas operativos con los que cuentan las computadoras son: Windows 2000, Windows XP, Windows Vista y Linux estos varían dependiendo en que sala se encuentren, y para complementar tienen instalados los programas de uso más frecuente entre los alumnos y las últimas versiones de los mismos.

Otro punto a tomar en cuenta es el gran número de alumnos que utilizan las computadoras, puesto que se tiene un registro en la base de datos de quien entra a las salas, en que sala entra y que computadora le asignan, se tiene un registro de aproximadamente 2000 alumnos por semana en total de las 4 salas. Es aquí donde entra el sistema SCOSU.

Anteriormente el registro se llevaba en hojas de papel, que por razones obvias no eran convenientes además de ser poco precisas, no se tenía un verdadero número de registros correctos, los datos de los alumnos que ingresaban en ocasiones eran imprecisos. Se tenían inconvenientes al momento de querer obtener estadísticas del número de alumnos que ingresaban, de cuánto tiempo utilizaban la computadoras cuantas veces y más. (las cuales son requeridas por DGSCA para analizar la actualización de los equipos de computo). Esto sucedía por que se perdían hojas o porque eran demasiados los datos y se llegaban a confundir con otros; pero uno de los inconvenientes más grandes era el tiempo que se tardaba para poder asignar una computadora, puesto que primero se tenía que tomar todos los datos del alumno que dejaba la maquina, en seguida tomar los datos del alumno que entraban y todo esto a mano, en pocas palabras era un sistema muy ineficiente.

Con el paso del tiempo y la llegada de los nuevos lenguajes de programación surgió la idea de crear un sistema efectivo y eficiente para poder administrar la entrada de los alumnos a las salas de UNICA almacenando los datos en una base y de esta manera hacer más eficiente el servicio que se ofrece, además de poder obtener estadísticas más confiables.

Las primeras versiones de SCOSU fueron creadas en Visual Basic 6.0 un lenguaje que constituye un IDE (entorno de desarrollo integrado o en inglés Integrated Development Environment) que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código (programa donde se escribe el código fuente), un depurador (programa que corrige errores en el código fuente para que pueda ser bien compilado), un compilador (programa que traduce el código fuente a lenguaje de máquina), y un constructor de interfaz gráfica o *GUI* (Graphical User Interface), es una forma de programar en la que no es necesario escribir el código para la parte gráfica del programa, sino que se puede hacer de forma visual.

Visual Basic es una forma de programación **basada en objetos** u **orientada al evento**. ¿Por qué esto? Al ser Windows un entorno multitarea, varias aplicaciones pueden ejecutarse a la vez (esta es una de las diferencias entre este sistema operativo y otros como MS-DOS); entonces las aplicaciones permanecen a la espera de acciones de parte del usuario, de parte de sí mismas o del sistema operativo para ejecutar una parte de su código y volver al estado de espera. Estas acciones se denominan eventos y el código que se ejecuta son funciones y métodos, todos estos se encuentran dentro de unas entidades llamadas Objetos.

Cabe mencionar que este lenguaje no está completamente orientado a objetos ya que le hacen faltas características como la herencia entre otras, es por eso que se encuentra limitada en ciertos aspectos que Visual Basic .Net cubre completamente (el cual se hablará más adelante).

Con todas estas características fue un gran avance poder realizar el primer sistema SCOSU, hizo que la espera de computadora fuera menor, ya que solo con teclear el número de cuenta del alumno se obtenían todos sus datos como nombre, carrera, la maquina que se le asignaba y más. También fue posible guardar todo estos datos en una base de datos realizada en PostgreSQL.

1.3.1 Problemas y causas

El sistema funciono bien, pero con algunos inconvenientes, como cuando existía un gran número de alumnos en las cuatros salas de UNICA, esto implicaba un gran número de consultas a la base de datos lo cual hacia que el sistema se cayera (dejaba de funcionar y se cerraba) ocasionando con esto que se alentara el servicio a los alumnos y se perdieran datos importantes.

Se opto por partir la base de datos en dos servidores y así dividir los datos, pero no funciono del todo. Como ya se menciono los datos eran almacenados en una base de datos realizada en PostgreSQL 8.1.4, la cual no era completamente una base de datos, ya que no se encontraba normalizada y existía inconsistencia de datos, además de no cumplir varios de los requisitos para poder ser llamada Base de datos, todo esto ayudaba a que el sistema en ocasiones fallara y la realización de las consultas para las estadísticas se complicara, debido a que no tenían una relación, en otras palabras se podría decir que eran simples tablas donde se almacenaban los registros.

Actualmente todavía se cuenta con SCOSU funcionando en las salas de UNICA, pero debido al número de usuarios que están incrementando con el paso del tiempo, el numero de registros aumenta por lo que es necesario limpiar la Base de datos de un servidor por lo menos cada semana, para que no exista una acumulación de datos que a la larga afectan al sistema y esté deja de funcionar. En cuanto a los recursos y actualizaciones con las que cuenta Visual Basic 6.0 en la actualidad podemos decir que todavía existen, pero está dejando de ser obsoleto debido a la aparición de Visual Basic .Net, otro punto en contra del sistema SCOSU. Por otra parte la interfaz grafica ya no es tan agradable como podría haber sido al principio y si a esto le sumamos que la base de datos no es tan funcional como se quisiera, tenemos que es necesario actualizar SCOSU desde adentro, es decir, desde la Base de datos pasando por la programación hasta poder administrar mejor los recursos en cuanto a consultas a la base de datos se refiere.

1.4 Requerimientos por parte de los departamentos

Para poder realizar SCOSU v2 fue necesario analizar las nuevas necesidades que tenía los diferentes departamentos para los cuales fue diseñado, en este caso está el Departamentos de Servicios Académicos (DSA) y Coordinación de Salas de Cómputo (CSC). Cada uno tiene necesidades propias de su departamento, a grandes rasgos sus actividades son:

- El DSA se encarga de las estadísticas del uso de los equipos de cómputo (Numero de alumnos que utilizaron las computadoras, durante cuánto tiempo, que días y más.)
- El CSC administra el acceso a las salas solicitando el número de cuenta del alumno y así asignar un equipo por cada usuario de la Facultad de Ingeniería, por otra parte controla las diferentes situaciones en las que puede estar una computadora, ya sea que se descomponga o esté en mantenimiento, todo esto se quedara registrado dentro del sistema, que es operado por integrantes del CSC.

1.4.1 Departamento de Servicios Académicos (DSA)

Como ya se menciona la labor de este departamento está enfocado a las estadísticas en cuanto al uso del equipo, es por eso que sus requerimientos se encuentran dirigidos a los datos que se puedan obtener de los alumnos que utilizan las salas de cómputo. Estos son sus requerimientos:

Fecha

Es necesario llevar las fechas de todos los días que se da servicio a los alumnos.

Usuarios de equipo de cómputo

Parte importante de poder brindar esta labor a los alumnos, es poder saber cuántos de ellos ocupan las computadoras, es por eso que este dato será de vital importancia para las estadísticas. Este número se obtendrá cada mes, de esta manera se obtendrá en qué cantidad y con qué frecuencia los alumnos utilizan las salas en todo el semestre.

Sesión de impresión (Usuarios y hojas)

Anteriormente SCOSU era el encargado de llevar estos registros, actualmente lo lleva un sistema especializado en la impresiones del alumno el Sistema de Control de Impresiones

(SCI), pero al igual que los datos de los usuarios, estos se guardarán en la base de datos de SCOSU v2.

Asesorías

Al igual que las impresiones, SCOSU era el encargado de llevar este control ahora se tiene un sistema que se encarga de llevar las asesorías que los becarios dan a los alumnos de la Facultad de Ingeniería, así mismo lleva el control de la hora de entrada y salida de los becarios en las respectivas salas y horas. Pero en esta ocasión sus datos son almacenados en su propia base de datos.

Todos los datos anteriormente mencionados se tendrán que obtener para las cuatro salas de UNICA.

Como podemos observar ya existen sistemas para obtener datos más específicos en ciertos puntos, pero una de las tareas de SCOSU v2 es la integración con estos sistemas para que todos puedan funcionar de manera conjunta.

1.4.2 Coordinación de Salas Computo (CSC)

Sus requerimientos son mas enfocados a la funcionalidad y aspecto de SCOSU v2, ya que aparte de lo que ya realizaba el antiguo SCOSU, ahora es necesario integrarle nuevas funciones y mejorar las anteriores en un cien por ciento, a continuación se muestra la lista de los nuevos requerimientos solicitada por el CSC:

Mensajes

- Cuando el usuario este sancionado que mande un mensaje de que esta sancionado y que aparezca nombre del sancionado, motivo de la sanción y la fecha de devolución.
- Cuando en dos computadoras este el mismo número de cuenta que mande mensaje de error (el mismo usuario está utilizando otro equipo) y que mande en que máquina está ubicado.

Asignación de equipos

- Cuando asignes máquinas, que te aparezca una lista en el extremo derecho, donde puedas visualizar las computadoras por sala y número de equipo que vas asignando, en orden consecutivo.

- Que cambie de icono cuando se asigna una computadora o se desocupe.
- Cuando liberes o se libere una computadora la ponga en una lista en el extremo derecho, donde puedas visualizar las computadoras por sala y número de equipo que se van liberando, en orden consecutivo.

Botón secundario del Mouse

- Que tenga otras opciones para colocar como por ejemplo mantenimiento y el icono cambie dependiendo en el estado que se ponga.

Búsqueda

- La búsqueda de usuarios a parte de buscar por nombre se agregara la opción de buscar por número de cuenta y nos mostrara el registro de los usuarios coincidentes con apellido o número de cuenta.
- Cuando ingresemos el nombre o número de cuenta nos envía a la sala y maquina donde se encuentra el usuario a buscar.

Diseño

- Más vistoso gráficamente.
- Se usaran iconos que simulen los equipos de la salas en lugar de botones
- La computadoras contarán con diferentes estados los cuales son:
 1. Ocupada
 2. Disponible
 3. Descompuesta
 4. Mantenimiento
 5. Curso
 6. Reparación
 7. Impresora
- Se cambiara la interfaz para introducir la contraseña y así entrar al sistema.

Registros o reportes

- Tendrá la opción para ordenar por apellido o número de cuenta.

- Mostrara los registros de los usuarios en todo el día dependiendo de la computadora que se elija, ya sea organizado por sala, número de cuenta o apellidos.
- Se podrá ver datos del usuario como cuantas veces ingreso al día, en que maquinas, total de horas y el horario en que estuvo usando el equipo.

Programas

- Que nos muestre los programas totales y como se encuentran instalados por salas y aulas.

Base de datos

- Mejorar la conexión al servidor que contiene la Base de datos de SCOSU v2.
- Administrar los recursos y las conexiones que utiliza SCOSU v2, para no ocasionar conflictos entre todas las salas.
- Tener todos los registros contenida en una sola Base de datos, esto es, que solo se utilice un solo servidor.
- Albergar una Base de datos alterna en caso de que el sistema sufra un fallo en cuanto a la conexión de la red.

Otros

- Si se tiene alguna duda o problema acerca del sistema que se puede enviar un correo electrónico a ejemplo@ejemplo.com
- Hora y fecha actual en pantalla principal del SCOSU v2.
- Que sea más dinámico.
- Que en las computadoras asignadas aparezca la hora de entrada y salida y que coloque como disponibles las maquinas cuando se termine el tiempo asignado.
- Tener una liga a la página <http://www.ingenieria.unam.mx/inscripciones/> para dar de alta a los alumnos que no se encuentren dentro del sistema.

- Podrá actualizar la paquetería (Solo jefe de cada sala y administrador).

1.5 Evaluación y determinación de costos

Al igual que cualquier sistema que se realice SCOSU v2 genera ciertos costos por todas las actividades que conllevan el desarrollar un sistema, desde la programación, las pruebas q se realizan, implementaciones, revisiones y más. Con estos costos se pretende dar una visión más amplia de la importancia que tiene el sistema.

A continuación se muestra la justificación de los costos generados por el desarrollo de SCOSU v2.

Punto 1. El factor humano

El desarrollador tiene un costo el cual se toma en cuenta dependiendo de su capacidad y experiencia, y para nuestro caso en particular esos dos puntos se encuentran bien cubiertos, si nos basamos en un sueldo base que otorga una empresa dedicada al desarrollo de software obtenemos que el sueldo mensual será de \$13,000. Ahora lo traducimos a una cantidad por hora. Es decir dividimos \$13,000 entre el número de horas de trabajo en un mes (Esto es, 8 horas diarias x 5 días a la semana x 4 semanas. Esto da por resultado 160 horas) obteniendo \$81.25.

Punto 2. Horas de trabajo

En la realización de SCOSU v2 se invirtieron aproximadamente 8 meses, ahora bien, si tomamos en cuenta que cada mes tiene 4 semanas y que se trabajaron 5 días a la semana durante 6 horas, obtenemos que las horas totales que se invirtieron en el desarrollo del sistema fueron de 960 horas.

Punto 3. Subtotal del costo

Para obtener el total simplemente multiplicamos el costo por hora x el número total de horas, esto es, 81.25×960 dando un total de **\$78,000**.

Punto 4. Costo total.

En el punto anterior se obtuvo el costo neto del desarrollo del sistema pero siempre es bueno dar cierta holgura al costo final, para poder enfrentar ciertos imprevistos y gastos de último momento, por lo que el monto total lo cerramos en **\$95,000**.

1.6 Infraestructura

Para que se logre desarrollar de una manera óptima el sistema, es necesario contar con la infraestructura adecuada, es decir, la plataforma donde se desarrollara, el equipo necesario que albergara esa plataforma, la aplicación y la base de datos.

La plataforma sobre la cual fue desarrollado el sistema es Visual Studio 2005 utilizando el lenguaje de programación Visual Basic .Net, el cual, debido a su gran capacidad de desarrollo y por todas las utilidades que contiene, se necesita un buen equipo donde alojar el software. En las siguientes líneas se muestra los requerimientos mínimos y recomendados que público Microsoft, para tener un mejor rendimiento del sistema:

- Procesador: mínimo 600 MHz, recomendado 1 GHz
- RAM: mínimo 192 MB, recomendado 256 MB.
- HD: mínimo 3 GB, recomendado 4.8 GB.
- Windows 2000 Professional SP4.

De igual manera se necesita un equipo con los mismos requerimientos o superiores, donde se instalara SCOSU v2 y el .Net [Framework](#) 2.0 necesario para que funcione la aplicación. Los equipos se encontraran en todas las salas de UNICA.

La base de datos se estará en PostgreSQL, este manejador y la base se encuentran en el servidor Tork que es propiedad de UNICA y del cual es responsable el Departamento de Redes y Operación de Servidores (DROS) de su administración.

1.7 Beneficios

En seguida se enumeran todos los beneficios obtenidos por SCOSU v2, pasando por lo administrativo hasta lo económico:

1. Agilizara el acceso de los alumnos de la facultad a las salas de cómputo de UNICA.
2. Reducirá del tiempo de atención al alumno.
3. El acceso al sistema será de forma fácil, amigable y segura.
4. Obtendrá estadísticas del servicio brindado por UNICA.

5. Generara reportes de los alumnos que utilizan las salas de UNICA.
6. Optimizara la conexión a la base de datos.
7. Contara con una base de datos correctamente diseñada, que ayudara a SCOSU v2 a tener una conexión más rápida, además, se podrán obtener estadísticas exactas y de una manera eficiente.
8. Evitara un costo considerablemente fuerte, el cual se podrá ocupar en otras áreas de UNICA.
9. Contara con documentación del sistema:
 - a. Manual de Instalación.
 - b. Manual del usuario.
 - c. Manual del programador.

CAPÍTULO 2

ANÁLISIS Y DISEÑO

El análisis de sistemas en computación es una actividad compleja y altamente dependiente de consideraciones humanas; lo que consiste en efectuar un análisis completo de todo el problema o sistema existente, con la finalidad de proponer un modelo para su solución.

Este modelo no puede existir sin que se haya especificado con claridad todos y cada uno de los componentes estructurales del sistema, en el sentido de que constituyen un conjunto de pasos bien especificados que se acercan paulatinamente a una solución deseada.

Para poder realizar el análisis y diseño de algún software se cuenta con herramientas llamadas Herramientas de modelado que ayudan a descifrar el problema con ciertos diagramas y modelos.

¿Para Qué Se Utilizan las Herramientas de Modelado?

La labor del análisis involucra el modelado del sistema que desea el usuario, hay muchos tipos diferentes de modelos que se pueden elaborar, como modelos diferentes puede hacer de una casa nueva un arquitecto.

Los modelos de análisis de sistema son representaciones abstractas de lo que al final será una combinación de hardware y software. El término 'modelo' puede parecer formal, pero representa un concepto manejado durante la mayor parte de la vida. Consideremos los siguientes tipos de modelos:

- **Mapas:** modelos bidimensionales del mundo en que vivimos.
- **Diagrama de Flujo:** representaciones esquemáticas de las decisiones y la secuencia de actividades para llevar a cabo un determinado procedimiento.
- **Dibujos Arquitectónicos:** representaciones esquemáticas de un edificio o un puente y más.

Podríamos preguntarnos *¿porqué se construye modelos y no simplemente el sistema mismo?* La respuesta es que podemos construir modelos de manera tal que enfatizamos ciertas propiedades críticas del sistema, mientras que simultáneamente desacentuamos otros de sus aspectos. Esto nos permite comunicarnos con el usuario de una manera enfocada, sin distraernos con asuntos y características ajenas al sistema.

Además, si nos damos cuenta que nuestra visión de los requerimientos no fue la correcta (o que este cambia respecto a sus requerimientos), se podrán hacer cambios en el modelo o desecharlo y hacer uno nuevo de ser necesario.

La alternativa es realizar reuniones con el cliente para poder reunir toda la información necesaria y luego construir el sistema, corremos el riesgo de que nuestro cliente no esté conforme con el producto final y los cambios a esta altura podrían resultar muy costosos y complicados.

Entonces utilicemos las herramientas de modelado para:

- Concentrarse en las propiedades importantes del sistema, y restarle atención a las menos importantes.
- Discutir cambios y correcciones en los requerimientos de usuarios, a bajo costo y riesgo mínimo.
- Verificar que el analista comprenda correctamente el ambiente del usuario y que lo haya respaldado con información documental para que los diseñadores de sistemas y programadores puedan construir el sistema.

2.1 Propuesta tecnológica

El método seleccionado para dar solución al problema será la automatización del servicio a través de una aplicación .NET, debido principalmente por los recursos disponibles en la institución y los equipos con los que cuentan los usuarios potenciales del servicio. Se integrará una aplicación .Net de acceso al servicio, que en una primera parte de atención, estará enfocado a los usuarios finales que requieran de la aplicación.

Para el desarrollo del proyecto, se hará uso de herramientas de documentación de análisis y diseño denominada UML (Lenguaje Unificado de Modelado) junto con la herramienta CASE llamada StarUML.

Para la construcción de la aplicación, se utilizarán herramientas .Net (Microsoft Visual Studio 2005 y ADO .Net) así como de diseño de base de datos, tales como ERwin y PostgreSQL, Macromedia y en la realización de los reportes se usara Crystal Reports.

En la etapa de revisión y pruebas, se utilizarán procedimientos propios de UNICA para la aprobación de unidades, así como los procedimientos de control de ambientes de desarrollo y de la seguridad de los sistemas.

2.2 Modelado del sistema

UML (Unified Modeling Language)

Para realizar el análisis de SCOSU v2 se utilizo Unified Modeling Language (UML) el cual es un lenguaje visual para crear modelos de sistemas, es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos.

Pero ¿por qué UML? Hoy en día, está consolidado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo. Mediante UML es posible establecer la serie de requerimientos y estructuras necesarias para plasmar un sistema de software previo al proceso intensivo de escribir código.

En otros términos, así como en la construcción de un edificio se realizan planos previos a su construcción, en Software se deben realizar diseños en UML previa codificación de un sistema, ahora bien, aunque UML es un lenguaje, éste posee más características visuales que programáticas, mismas que facilitan a integrantes de un equipo multidisciplinario participar e intercomunicarse fácilmente, estos integrantes siendo los analistas, diseñadores, especialistas de área y desde luego los programadores.

Puesto que UML es empleado en el análisis para sistemas de mediana-alta complejidad, era de esperarse que su base radique en otro paradigma empleado en diseños de sistemas de alto nivel que es la *orientación a objetos*, por lo que para trabajar en UML puede ser considerado un pre-requisito tener experiencia en un lenguaje orientado a objetos.

UML consta de todos los elementos y diagramas que permiten modelar los sistemas en base al paradigma orientado a objetos. Los modelos orientados a objetos cuando se construyen en forma correcta, son fáciles de comunicar, cambiar, expandir, validar y verificar. El modelar en UML vuelve flexible el cambio y permite crear componentes plenamente reutilizable

Diagramas UML¹

Diagrama de clases

Muestran un conjunto de clases y sus relaciones. Los diagramas de clases proporcionan una perspectiva estática del sistema (representan su diseño estructural).

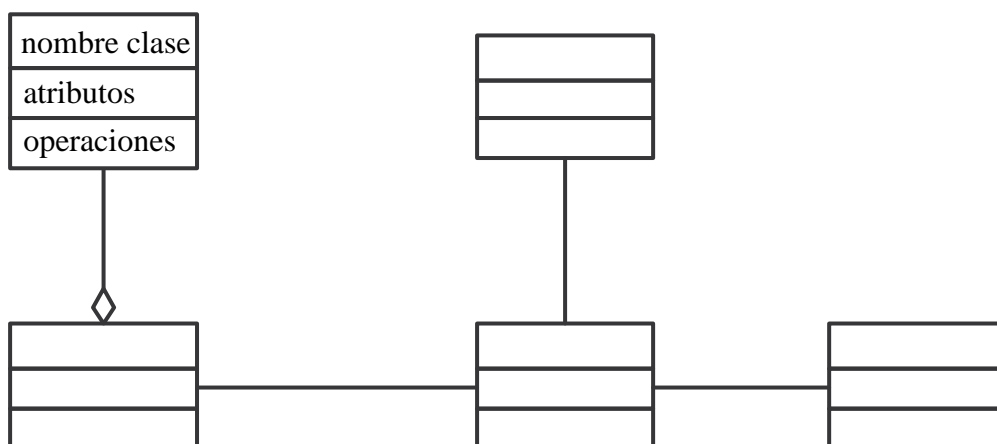


Figura 2.1 Diagrama de clases

Notación

Atributos

[visibilidad] nombre [multiplicidad] [: tipo [= valor_por_defecto]]

Operaciones

[visibilidad] nombre (([[in|out] parámetro : tipo [, &]])[:tipo_devuelto])

¹ Aprendiendo UML en 24 horas, Joseph Schmuller, Prentice Hall.

Diagrama de objetos

Muestran un conjunto de objetos y sus relaciones (una situación concreta en un momento determinado). Los diagramas de objetos representan instantáneas de instancias de los elementos que aparecen en los diagramas de clases. Un diagrama de objetos expresa la parte estática de una interacción. Para ver los aspectos dinámicos de la interacción se utilizan los diagramas de interacción (diagramas de secuencia y diagramas de comunicación/colaboración).

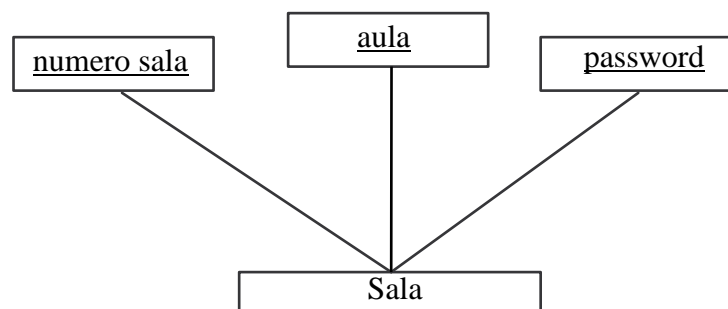


Figura 2.2 Diagrama de objetos

NOTA: Los identificadores subrayados indican que se trata de objetos.

Diagrama de Interacción

Es una representación gráfica de interacciones entre objetos. Hay dos tipos de diagramas de interacción:

- Diagrama de secuencia
- Diagrama de colaboración

Diagrama de secuencia

- Resaltan la ordenación temporal de los mensajes que se intercambian.
- En la parte superior aparecen los objetos que intervienen.
- La dimensión temporal se indica verticalmente (el tiempo transcurre hacia abajo).
- Las líneas verticales indican el período de vida de cada objeto.
- El paso de mensajes se indica con flechas horizontales u oblicuas (cando existe demora entre el envío y la atención del mensaje).
- La realización de una acción se indica con rectángulos sobre las líneas de actividad del objeto que realiza la acción.

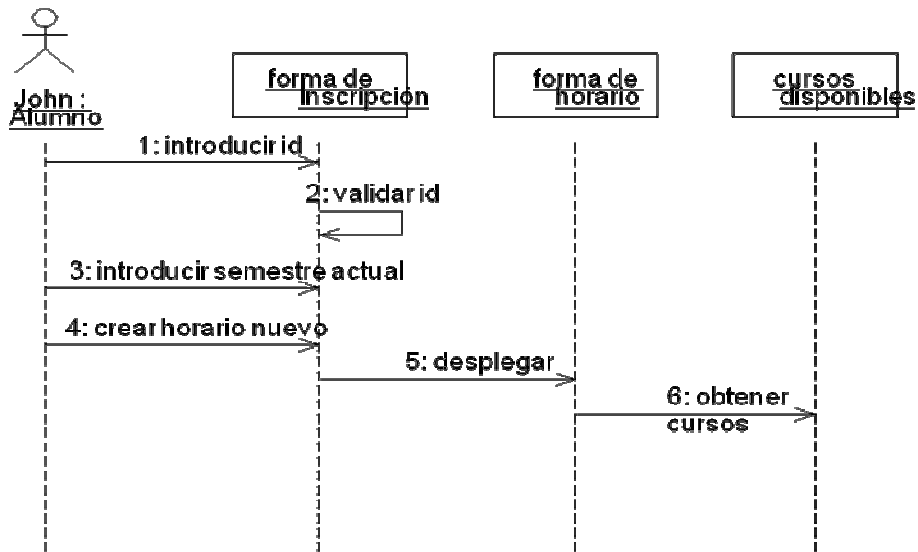


Figura 2.3 Diagrama de secuencia

Diagrama de colaboración

Resaltan la organización estructural de los objetos que intercambian mensajes.

- La perspectiva estática del sistema viene dada por las relaciones existentes entre los objetos (igual que en un diagrama de objetos).
- La vista dinámica de la interacción viene indicada por el envío de mensajes a través de los enlaces existentes entre los objetos.
- Los mensajes se numeran para ilustrar el orden en que se emiten.

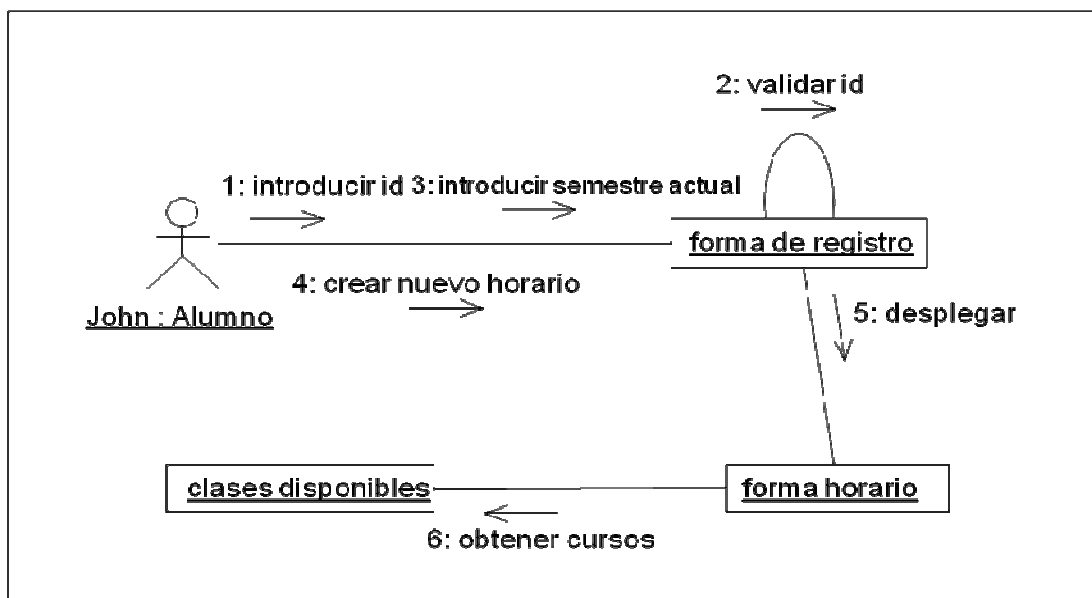


Figura 2.4 Diagrama de colaboración

Los diagramas de secuencia y de comunicación son isomorfos ya que un diagrama de secuencia se puede transformar mecánicamente en un diagrama de comunicación y un diagrama de comunicación se puede transformar automáticamente en un diagrama de secuencia.

Otros diagramas UML para representar aspectos dinámicos del sistema

Diagrama de casos de uso

Los diagramas de uso se suelen utilizar en el modelado del sistema desde el punto de vista de sus usuarios para representar las acciones que realiza cada tipo de usuario.

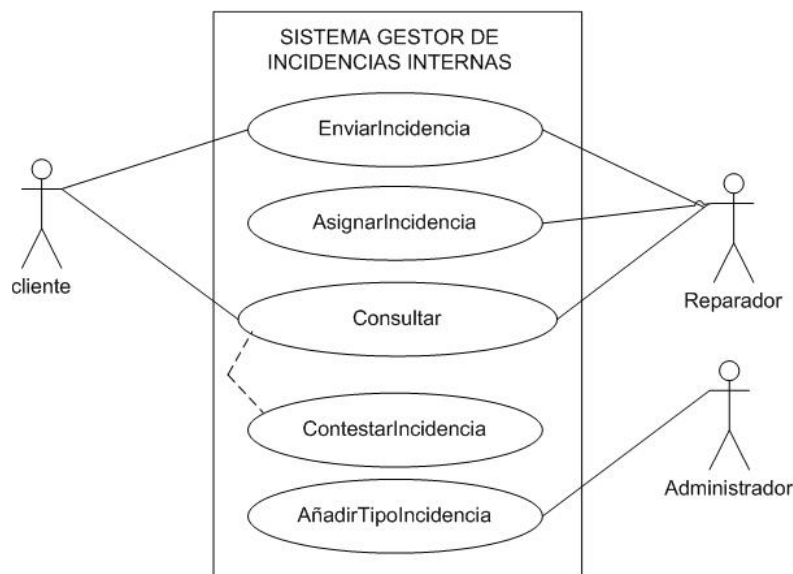


Figura 2.5 Actores y casos de uso

Diagrama de estados

Son especialmente importantes para describir el comportamiento de un sistema reactivo (cuyo comportamiento está dirigido por eventos).

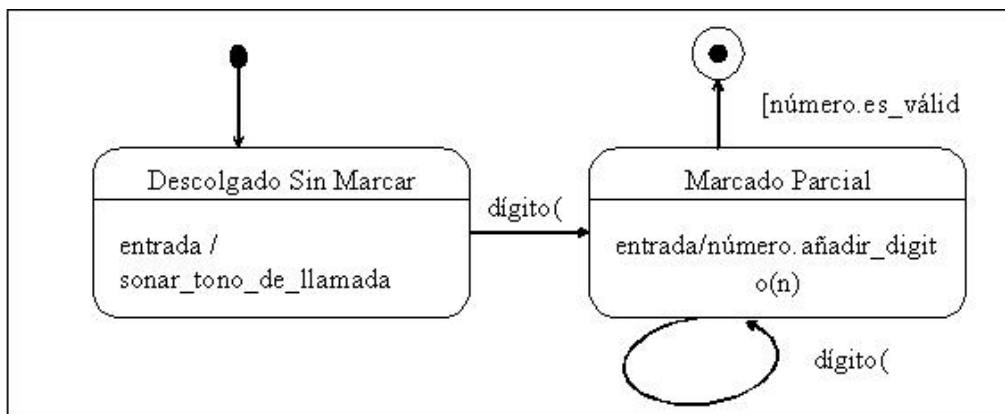


Figura 2.6 Estados y transiciones entre estados

Diagrama de actividades

Los diagramas de actividades muestran el orden en el que se van realizando tareas dentro de un sistema (el flujo de control de las actividades).

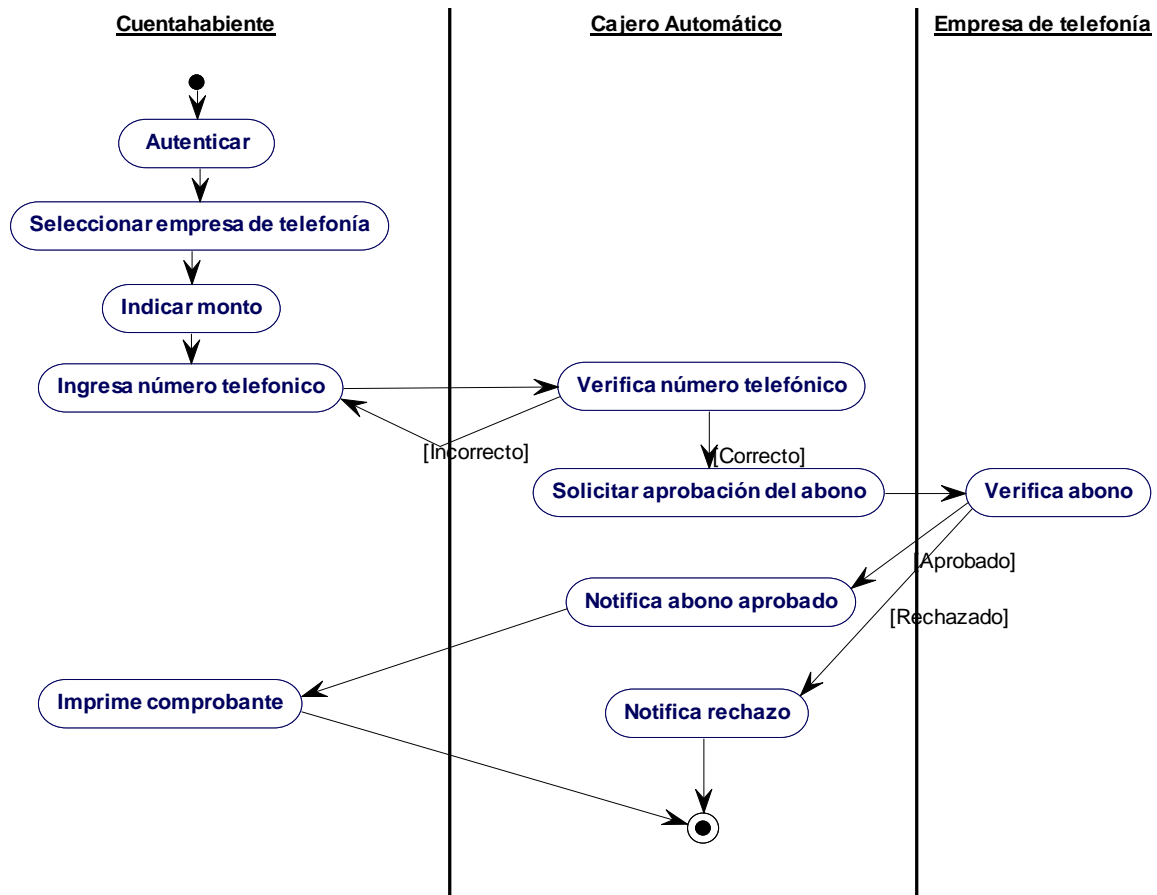


Figura 2.7 Flujo de control en el sistema

Diagramas UML para representar aspectos físicos del sistema

Diagrama de componentes

Muestra las organizaciones y dependencias lógicas entre componentes software, sean éstos componentes de código fuente, binarios o ejecutables. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo.

Los elementos de modelado dentro de un diagrama de componentes serán componentes y paquetes. En cuanto a los componentes, sólo aparecen tipos de componentes, ya que las instancias específicas de cada tipo se encuentran en el diagrama de despliegue.

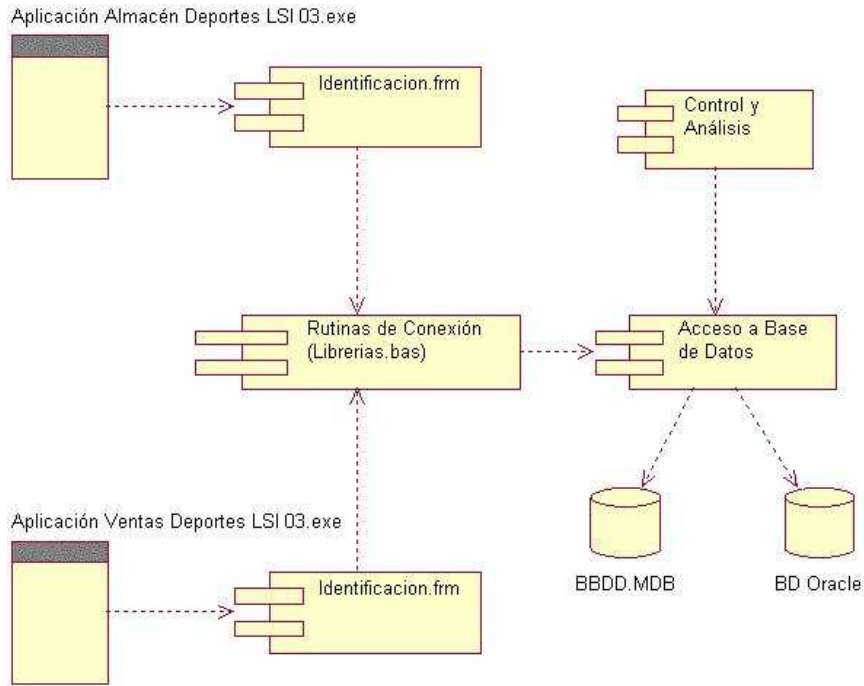


Figura 2.8 Componentes y dependencia entre ellos

Diagrama de despliegue

Configuración del sistema en tiempo de ejecución. Muestra las relaciones físicas entre los componentes *hardware* y *software* en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes *software* (procesos y objetos que se ejecutan en ellos).

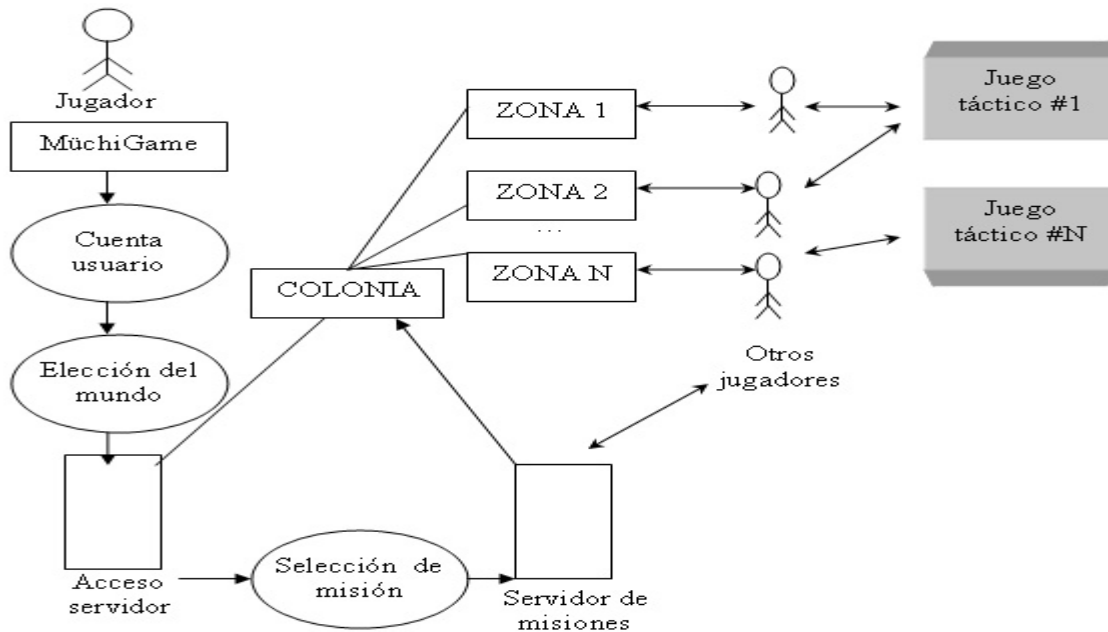


Figura 2.9 Nodos de procesamiento y componentes

Como podemos observar UML tiene varios diagramas para modelar los sistemas pero, ¿Cuántos diagramas debo de crear? No existe una respuesta específica, varios especialistas han coincidido en que se deben de crear los diagramas que mejor reflejen el modelado de la problemática y para el caso de SCOSU v2 se llegó a la conclusión de que creando los diagramas de casos de uso, de clases y de actividades quedara completamente modelado el problema, debido a su funcionalidad y a que es un sistema creado en un lenguaje orientado a objetos.

2.2.1 Diagrama de casos de uso

Actores

Tras analizar el problema de SCOSU v2 se obtuvieron los siguientes actores:

➤ **Administrador**

- El administrador es la persona que tendrá el control del sistema, es decir, podrá realizar modificaciones al mismo, esto dependerá de las necesidades del usuario. Por otra parte corregirá errores de aplicación o emergencias que puedan llegar a surgir, además de tener la posibilidad de quitar sancionados.

➤ **Personal de Salas**

- Pueden ser alumnos de la Facultad de Ingeniería que se encuentran realizando su servicio social, trabajadores encargados de las salas de UNICA o el mismo Jefe de salas.

➤ **Jefe de Salas.**

- Persona encargada de coordinar todas las actividades que se realizan dentro de las salas de cómputo. Contara con privilegios como poder quitar sancionados si lo considera necesario y actualizar paquetería cuando ésta la requiera.

➤ **Sistema de Control de Impresiones**

- Sistema externo que tiene una acción directa con el sistema SCOSU, es el encargado de verificar el número de impresiones que le sobran al alumno, además de proporcionar estadísticas y bitácoras de las impresiones.

➤ **Sistema de Control de Asesorías**

- Sistema externo que tiene como función verificar la hora de entrada y salida de los asesores, los cuales son becarios de UNICA y almacena el número de asesorías que realizar un asesor para futuras estadísticas.

Casos de uso

Los casos de uso que se obtuvieron de los diferentes actores se describen a continuación:

➤ **Personal de Salas**

<CU01> : <Manipular_Estados_Computadora>		
CARACTERISTICAS		
Resumen	Se realiza desde el momento en que el personal de salas asigna computadoras, disponibilidad y más.	
Pre-Condiciones	Introduce la contraseña correspondiente dependiendo en que sala se encuentre.	
Post-Condiciones	Tomara el control del sistema, pudiendo realizar varias tareas manteniendo el control de la sala de cómputo.	
ACTORES		
	Rol	Descripción del Rol
Actor Principal	Personal_Sala	Alumnos de la Facultad de Ingeniería que se encuentran realizando su servicio social, trabajadores encargados de las salas de UNICA o el mismo Jefe de salas.
Actores Secundarios		
CURSO NOMAL. (ESCENARIO NORMAL)		
Paso	Actor	Descripción
1	Personal_Sala	Da clic derecho a cualquier computadora la cual despliega todos los estados de las mismas
2	Personal_Sala	Selecciona cualquier el estado en que quiere dejar a la computadora
CURSO ALTERNATIVO Y EXTENSIÓN DEL CURSO NORMAL		
Paso	Variación	Descripción
PUNTOS ABIERTOS		
Punto	Descripción	

➤ Personal de Salas
Continuación...

<CU02> : <Asignar_Computadora>		
CARACTERISTICAS		
Resumen	Caso de uso que empieza cuando el personal de sala introduce el número de cuenta del alumno al sistema.	
Pre-Condiciones	El alumno proporciona su número de cuenta.	
Post-Condiciones	El personal de sala recibirá el número de cuenta del siguiente alumno.	
ACTORES		
	Rol	Descripción del Rol
Actor Principal	Personal_Sala	Alumnos de la Facultad de Ingeniería que se encuentran realizando su servicio social, trabajadores encargados de las salas de UNICA o el mismo Jefe de salas.
Actores Secundarios		
CURSO NORMAL (ESCENARIO NORMAL)		
Paso	Actor	Descripción
1	Personal_Sala	Introduce el número de cuenta del alumno al sistema.
2	Personal_Sala	Le asigna una computadora disponible.
3	Personal_Sala	Recibe el número de cuenta del siguiente alumno
CURSO ALTERNATIVO Y EXTENSIÓN DEL CURSO NORMAL		
Paso	Variación	Descripción
2.1	No está dado de alta	Se informa al alumno que debe de darse de alta y se cancela la acción.
2.2	Esta sancionado	Se informa al alumno de su sanción y se cancela la acción.
PUNTOS ABIERTOS		
Punto	Descripción	

➤ Personal de Salas

Continuación...

<CU03> : <Apartar_Sala>		
CARACTERISTICAS		
Resumen	Caso de uso que empieza cuando se quiere dar un curso o se tiene que dar mantenimiento a alguna de las aulas de las 4 salas	
Pre-Condiciones	Se informa al personal de salas que próximamente se llevara a cabo un curso o que se tiene que dar mantenimiento a un aula de esa sala.	
Post-Condiciones	Permanecerá la sala apartada hasta que termine el curso o terminen de dar mantenimiento.	
ACTORES		
	Rol	Descripción del Rol
Actor Principal	Personal_Sala	Alumnos de la Facultad de Ingeniería que se encuentran realizando su servicio social, trabajadores encargados de las salas de UNICA o el mismo Jefe de salas.
Actores Secundarios		
CURSO NOMAL (ESCENARIO NORMAL)		
Paso	Actor	Descripción
1	Personal_Sala	Apartar la el aula de la sala donde se realizara el curso o se dará mantenimiento.
2	Personal_Sala	No asignara ninguna computadora de esa sala.
3	Personal_Sala	Colocara como disponible la sala una vez terminado el curso o el mantenimiento.
CURSO ALTERNATIVO Y EXTENSIÓN DEL CURSO NORMAL		
Paso	Variación	Descripción
3.1	Colocar sala disponible manualmente o automático	Manualmente cuando se termina antes de lo previsto, automáticamente cuando llegue la fecha de termino del curso o del mantenimiento.
PUNTOS ABIERTOS		
Punto	Descripción	

➤ Personal de Salas

Continuación...

<CU04> : <Sancionar _Alumno>		
CARACTERISTICAS		
Resumen	Este caso de uso empieza cuando el alumno ha roto alguna de las reglas que tienen las salas.	
Pre-Condiciones	Se introduce el número de cuenta a la sección de sancionados.	
Post-Condiciones	Una vez sancionado no podrá ingresar a las sala hasta que cumpla su castigo.	
ACTORES		
	Rol	Descripción del Rol
Actor Principal	Personal_Sala	Alumnos de la Facultad de Ingeniería que se encuentran realizando su servicio social, trabajadores encargados de las salas de UNICA o el mismo Jefe de salas.
Actores Secundarios		
CURSO NOMAL (ESCENARIO NORMAL)		
Paso	Actor	Descripción
1	Personal_Sala	Introducir el número de cuenta en la sección de sancionados del sistema.
2	Personal_Sala	Colocar el tipo de sanción que se le dará
3	Personal_Sala	Informar al alumno de cuantos días será su sanción.
CURSO ALTERNATIVO Y EXTENSIÓN DEL CURSO NORMAL		
Paso	Variación	Descripción
1.1	Directo de la computadora	Esta acción se realiza sancionando directamente al alumno que se encuentra en la computadora sin necesidad de pedirle su número de cuenta.
2.1	El tipo de sanción no se encuentra en la clasificación	Se colocara la sanción manualmente y de cuantos días le corresponden quedara a criterio del personal de sala.
PUNTOS ABIERTOS		
Punto	Descripción	

➤ Personal de Salas
Continuación...

<CU05> : <Realizar_Busqueda_Alumno>		
CARACTERISTICAS		
Resumen	Empieza cuando una persona busca a algún alumno dentro de las salas.	
Pre-Condiciones	Se tiene que contar con el nombre o número de cuenta del alumno a buscar.	
Post-Condiciones	Se cierra el buscador y el sistema sigue con sus tareas normales.	
ACTORES		
	Rol	Descripción del Rol
Actor Principal	Personal_Sala	Alumnos de la Facultad de Ingeniería que se encuentran realizando su servicio social, trabajadores encargados de las salas de UNICA o el mismo Jefe de salas.
Actores Secundarios		
CURSO NOMAL (ESCENARIO NORMAL)		
Paso	Actor	Descripción
1	Personal_Sala	Introduce el número de cuenta del alumno o el nombre del alumno.
2	Personal_Sala	En caso de que se encuentre al alumno, observa donde está ubicado.
3	Personal_Sala	Informa al interesado al alumno que lo buscan en la entrada de la sala de UNICA
CURSO ALTERNATIVO Y EXTENSIÓN DEL CURSO NORMAL		
Paso	Variación	Descripción
2.1	No se obtiene ningún resultado	Después de introducir el nombre o el número de cuenta del alumno no se obtiene ningún registro, lo cual significa que no se encuentra en esa sala.
PUNTOS ABIERTOS		
Punto	Descripción	

➤ Personal de Salas

Continuación...

<CU06> : <Notificar_Administrador>		
CARACTERISTICAS		
Resumen	Sucede cuando se desea dar un aviso al administrador del sistema, ya sea de algún error o de algún cambio.	
Pre-Condiciones	El sistema sufre desperfectos en cuanto a programación o por culpa de los servidores.	
Post-Condiciones	Si el error fue grande se reinicia el sistema, en caso contrario sigue con su funcionamiento normal.	
ACTORES		
	Rol	Descripción del Rol
Actor Principal	Personal_Sala	Alumnos de la Facultad de Ingeniería que se encuentran realizando su servicio social, trabajadores encargados de las salas de UNICA o el mismo Jefe de salas.
Actores Secundarios		
CURSO NOMAL (ESCENARIO NORMAL)		
Paso	Actor	Descripción
1	Personal_Sala	Abre la sección para contactar al administrador.
2	Personal_Sala	Introduce su nombre y el correo electrónico del administrador.
3	Personal_Sala	Informa del motivo que lo orillo a escribirle, teniendo la oportunidad de anexar alguna imagen.
CURSO ALTERNATIVO Y EXTENSIÓN DEL CURSO NORMAL		
Paso	Variación	Descripción
PUNTOS ABIERTOS		
Punto	Descripción	

➤ Jefe de salas

<CU07> : <Eliminar_Registros>		
CARACTERISTICAS		
Resumen	Sucede cuando se desea eliminar algún registro de la base de datos, ya sea una sanción o un curso.	
Pre-Condiciones	El alumno es sancionado de manera equivocada o por alguna razón pide que se le quite su sanción. Un curso se termino antes de lo previsto.	
Post-Condiciones	El alumno tendrá acceso a las computadoras para poder realizar tareas escolares e investigaciones. La sala ya no se apartara para el curso que se tenía programado.	
ACTORES		
	Rol	Descripción del Rol
Actor Principal	Jefe de Sala	Persona encargada de coordinar todas las actividades que se realizan dentro de las salas de cómputo.
Actores Secundarios		
CURSO NOMAL (ESCENARIO NORMAL)		
Paso	Actor	Descripción
1	Jefe de Sala	Se busca al alumno dentro de la bitácora de sancionados.
2	Jefe de Sala	Analiza el motivo de la sanción
3	Jefe de Sala	Retira la sanción al alumno.
CURSO ALTERNATIVO Y EXTENSIÓN DEL CURSO NORMAL		
Paso	Variación	Descripción
1.1	Curso	Se busca el curso dentro de la bitácora de cursos.
2.1	Se termina el curso antes de tiempo.	Se elimina el curso para que no cause conflictos con la salas de UNICA.
3.1	No retira la sanción	Después de analizar el motivo de la sanción considera que no es pertinente retirársela.
PUNTOS ABIERTOS		
Punto	Descripción	

➤ Jefe de salas

Continuación...

<CU08> : <Modificar_Paqueteria>		
CARACTERISTICAS		
Resumen	Caso de uso que sucede cuando se cambia o se actualiza la paquetería (software) de alguna de las salas o de todas	
Pre-Condiciones	Se cambia la paquetería de las computadoras de laguna de las salas.	
Post-Condiciones	Se contara con toda la paquetería actualizada y nueva de la sala correspondiente.	
ACTORES		
	Rol	Descripción del Rol
Actor Principal	Jefe de Sala	Persona encargada de coordinar todas las actividades que se realizan dentro de las salas de cómputo.
Actores Secundarios	Personal_Sala	Alumnos de la Facultad de Ingeniería que se encuentran realizando su servicio social, trabajadores encargados de las salas de UNICA o el mismo Jefe de salas.
CURSO NOMAL (ESCENARIO NORMAL)		
Paso	Actor	Descripción
1	Personal_Sala	Actualiza el software en alguna de las salas o en todas.
2	Personal_Sala	Le comunica al jefe de sala de los cambios.
3	Jefe de Sala	Entra con su respectiva contraseña al modulo del jefe en la sección de paquetería.
4	Jefe de Sala	Realiza las modificaciones pertinentes.
CURSO ALTERNATIVO Y EXTENSIÓN DEL CURSO NORMAL		
Paso	Variación	Descripción
PUNTOS ABIERTOS		
Punto	Descripción	

➤ Jefe de salas

Continuación...

<CU09> : <Realizar_Reportes>		
CARACTERISTICAS		
Resumen	Comienza debido a la necesidad de saber ciertos datos de un alumno en particular, como cuando tiempo utilizo las salas de UNICA, en que salas y computadoras estuvo y más.	
Pre-Condiciones	Se le informa al jefe de Sala que se quiere el reporte de un alumno.	
Post-Condiciones	Se genera un reporte detallado con los datos del alumno, los cuales se podrán exportar a Excel para su uso o imprimirlo para mostrarlos.	
ACTORES		
	Rol	Descripción del Rol
Actor Principal	Jefe de Sala	Persona encargada de coordinar todas las actividades que se realizan dentro de las salas de cómputo.
Actores Secundarios		
CURSO NOMAL (ESCENARIO NORMAL)		
Paso	Actor	Descripción
1	Jefe de Sala	Entra al modulo de jefe con su respectiva contraseña en la sección de reporte.
2	Jefe de Sala	Introduce el nombre o el número de cuenta del alumno.
3	Jefe de Sala	Genera el reporte ya sea por día o por semestre.
4	Jefe de Sala	Imprime el reporte generado
CURSO ALTERNATIVO Y EXTENSIÓN DEL CURSO NORMAL		
Paso	Variación	Descripción
4.1	Exportar datos	En caso de ser necesario podrá exportar los datos a Excel para realizar estadísticas.
PUNTOS ABIERTOS		
Punto	Descripción	

➤ **Administrador**

Este actor como su nombre lo dice es el encargado de administrar todas y cada una de las funciones del sistema, por lo tanto podrá realizar todos los casos de uso tanto del Personal de Sala como del Jefe de Sala. En la Figura 2.10 Se muestra la relación del actor Administrador con los diferentes casos de uso del sistema.

➤ **Sistema de Control de Impresiones**

<CU10> : <Verificar_Num_Impresiones>		
CARACTERISTICAS		
Resumen	Comienza cuando un alumno entrega su credencial para imprimir.	
Pre-Condiciones	Contar con la credencia del alumno	
Post-Condiciones	Se dará o no autorización para que el alumno imprima.	
ACTORES		
	Rol	Descripción del Rol
Actor Principal	SCI	Encargado de verificar el número de impresiones que le sobran al alumno.
Actores Secundarios	Personal_Sala	Alumnos de la Facultad de Ingeniería que se encuentran realizando su servicio social, trabajadores encargados de las salas de UNICA o el mismo Jefe de salas.
CURSO NOMAL (ESCENARIO NORMAL)		
Paso	Actor	Descripción
1	Personal_Sala	Introduce su número de cuenta
2	SCI	Muestra mensaje de cuantas hojas podrá imprimir el alumno en ese día
3	Personal_Sala	Deja pasar al alumno para que imprima.
CURSO ALTERNATIVO Y EXTENSIÓN DEL CURSO NORMAL		
Paso	Variación	Descripción
2.1	Se termino el número de hojas que puede imprimir por día.	El sistema indica mediante un mensaje que no tiene derecho a imprimir y se cancela la operación.
3.1	No permite imprimir al alumno.	Se indica al alumno que no puede imprimir y se cancela la operación.
PUNTOS ABIERTOS		
Punto	Descripción	

➤ Sistema de Control de Impresiones
Continuación...

<CU11> : <Mostrar_Estadísticas_Impresiones>		
CARACTERÍSTICAS		
Resumen	Comienza cuando el personal de sala desea ver cuántas impresiones se han realizado durante el día	
Pre-Condiciones		
Post-Condiciones		
ACTORES		
	Rol	Descripción del Rol
Actor Principal	SCI	Encargado de verificar el número de impresiones que le sobran al alumno.
Actores Secundarios	Pesonal_Sala	Alumnos de la Facultad de Ingeniería que se encuentran realizando su servicio social, trabajadores encargados de las salas de UNICA o el mismo Jefe de salas.
CURSO NOMAL (ESCENARIO NORMAL)		
Paso	Actor	Descripción
1	Personal_Sala	Abrir la parte de estadísticas
2	Personal_Sala	Elegir la opción para visualizar las estadísticas.
3	SCI	Muestra el número de impresiones que se realizaron durante el día así como el número de usuarios.
CURSO ALTERNATIVO Y EXTENSIÓN DEL CURSO NORMAL		
Paso	Variación	Descripción
PUNTOS ABIERTOS		
Punto	Descripción	

➤ Sistema de Control de Impresiones
Continuación...

<CU12> : <Mostrar_Bitacora_Impresiones>		
CARACTERISTICAS		
Resumen	Caso de uso donde el personal de sala desea ver la bitácora de impresiones de algún alumno en particular.	
Pre-Condiciones		
Post-Condiciones		
ACTORES		
	Rol	Descripción del Rol
Actor Principal	SCI	Encargado de verificar el número de impresiones que le sobran al alumno.
Actores Secundarios	Personal_Sala	Alumnos de la Facultad de Ingeniería que se encuentran realizando su servicio social, trabajadores encargados de las salas de UNICA o el mismo Jefe de salas.
CURSO NOMAL (ESCENARIO NORMAL)		
Paso	Actor	Descripción
1	Personal_Sala	Abrir la parte de bitácora de impresiones
2	Personal_Sala	Introduce el número de cuenta del alumno.
3	SCI	Muestra bitácora del alumno del cual se introdujo el número de cuenta.
CURSO ALTERNATIVO Y EXTENSIÓN DEL CURSO NORMAL		
Paso	Variación	Descripción
3.1	No muestra bitácora	El alumno nunca a impreso en la salas. Se muestra mensaje y se cancela la acción-
PUNTOS ABIERTOS		
Punto	Descripción	

➤ Sistema de Control de Asesorías

<CU13> : <Controla_Horario_Asesor >		
CARACTERISTICAS		
Resumen	Sucede cuando un asesor llega a registrarse a una sala para dar su respectiva asesoría.	
Pre-Condiciones	Introducir el número de cuenta del asesor.	
Post-Condiciones	Registrar salida.	
ACTORES		
	Rol	Descripción del Rol
Actor Principal	Sistema de Control de Asesorías	Sistema externo que tiene como función verificar la hora de entrada y salida de los asesores.
Actores Secundarios	Pesonal_Sala	Alumnos de la Facultad de Ingeniería que se encuentran realizando su servicio social, trabajadores encargados de las salas de UNICA o el mismo Jefe de salas.
CURSO NOMAL (ESCENARIO NORMAL)		
Paso	Actor	Descripción
1	Personal_Sala	Introduce el número de cuenta del asesor
2	Sistema de Control de Asesorías	Registra la hora de entrada del asesor.
3	Sistema de Control de Asesorías	Registra la hora de salida del asesor
CURSO ALTERNATIVO Y EXTENSIÓN DEL CURSO NORMAL		
Paso	Variación	Descripción
PUNTOS ABIERTOS		
Punto	Descripción	

➤ Sistema de Control de Asesorías
Continuación...

<CU14> : <Almacena_Num_Asesorias >		
CARACTERISTICAS		
Resumen	Empieza cuando el asesor da una asesoría a cualquier persona que se lo pide	
Pre-Condiciones	Dar alguna asesoría.	
Post-Condiciones	Registrar la siguiente asesoría.	
ACTORES		
	Rol	Descripción del Rol
Actor Principal	Sistema de Control de Asesorías	Sistema externo que tiene como función verificar la hora de entrada y salida de los asesores.
Actores Secundarios	Personal_Sala	Alumnos de la Facultad de Ingeniería que se encuentran realizando su servicio social, trabajadores encargados de las salas de UNICA o el mismo Jefe de salas.
CURSO NOMAL (ESCENARIO NORMAL)		
Paso	Actor	Descripción
1	Personal_Sala	Introduce la asesoría dada por el asesor
2	Sistema de Control de Asesorías	Registra la asesoría
CURSO ALTERNATIVO Y EXTENSIÓN DEL CURSO NORMAL		
Paso	Variación	Descripción
PUNTOS ABIERTOS		
Punto	Descripción	

En la Figura 2.10 Se muestra cómo es que quedaron integrados todos los actores con sus respectivos casos de uso dentro del sistema SCOSU v2.



Figura 2.10 Diagrama de casos de uso de SCOSU v2

2.2.2 Diagrama de clases

Los diagramas de clase son los diagramas más comunes en el modelado de sistemas orientados a objetos ya que nos muestra un conjunto de clases, interfaces, y colaboraciones y sus relaciones. Se usan para modelar la vista de diseño estático de un sistema y son importantes no sólo para visualización, especificación y documentación de modelos estructurales, sino también para construir sistemas ejecutables.

Al igual que en los casos de uso se analizaron todos los detalles que integraban a SCOSU v2 teniendo como resultados las siguientes clases:

- **Usuario:** Esta clase es una clase abstracta, puesto que solo nos va a servir para que otras clases hereden los atributos de ésta. Sus propiedades son Nombre y apellidos de los usuarios.

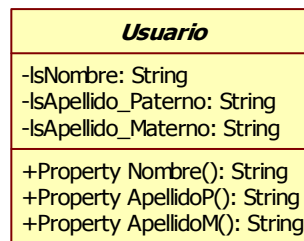


Figura 2.11 Clase Usuario

- **Registro_Historico:** Contiene los datos de la fecha, tales como hora de entrada, hora de salida y la fecha en la que el alumno ocupó una computadora dentro de la sala de UNICA.



Figura 2.12 Clase Registro_Historico

- **Alumno:** Clase con propiedades propias del alumno como su número de cuenta y carrera, además hereda los atributos del usuario complementado la clase con el nombre y apellidos paterno y materno.

Alumno
-ldbNo_Cuenta: Double -lsCarrera: String
+Property NoCuenta(): Double +Property Carrera(): String +Property Nombre(): String +Property ApellidoP(): String +Property ApellidoM(): String

Figura 2.13 Clase Alumno

- **Alumno_Sancionado:** Contendrá al alumno que incurra en alguna falta dentro de las instalaciones de las salas de UNICA, tiene como atributos el nombre y apellidos del alumno que cometa la falta, los cuales hereda de la clase abstracta Usuario y como atributos propios tiene el número de cuenta y la carrera.

Alumno_Sancionado
-ldbNo_Cuenta: Double -lsCarrera: String
+Property NoCuenta(): Double +Property Carrera(): String +Property Nombre(): String +Property ApellidoP(): String +Property ApellidoM(): String

Figura 2.14 Clase Alumno_Sancionado

- **Jefe_Sala:** Toda sala tiene un jefe responsable, la clase Jefe_Sala abarca las propiedades de éste, por otra parte también hereda de la clase Usuario teniendo así su nombre y apellidos además de su id como jefe de sala y un password que lo identifica como tal. Por otra parte tiene el método actualizarPaqueteria () ya que solo el jefe de sala podrá realizar esta acción.

Jefe_Sala
-liId_Jefe: Integer -lsPassword: String
+Sub actualizarPaqueteria(software: String) +Property IdJefe(): Integer +Property Password(): String +Property Nombre(): String +Property ApellidoP(): String +Property ApellidoM(): String

Figura 2.15 Clase Jefe_Sala

- **Responsable_Sancion:** Cuando se sanciona a un alumno siempre se tendrá registrado quien fue el responsable, con datos tales como su nombre y apellidos y un id propio por cada responsable.

Responsable_Sancion
-liId_Responsable: Integer
+Sub sancionar() +Property IdResponsable(): Integer +Property Nombre(): String +Property ApellidoP(): String +Property ApellidoM(): String

Figura 2.16 Clase Responsable_Sancion

- **Sala:** Esta clase representa las propiedades de importancia para SCOSU v2 como son el numero de sala, el aula y el password para poder acceder a las diferentes salas de UNICA ya que cada una posee su propia clave de acceso.

Sala
-liNumero_Sala: Integer -lcAula: Char -lsPassword: String
+Property NumeroSala(): Integer +Property Aula(): Char +Property Password(): String

Figura 2.17 Clase Sala

- **Reporte:** Clase encargada de contener los datos necesarios para registrar cuando y que reporte se realiza.

Reporte
-liNumero_Reporte: Integer -lsNombre_Reporte: String -ldtFecha_Creacion: Date
+Property NumeroReporte(): Integer +Property NombreReporte(): String +Property FechaCreacion(): Date

Figura 2.18 Clase Reporte

- **Cancelar_Sancion:** Fecha, hora y id del responsable de cancelar la sanción son los datos almacenados en esta clase, esto es con el fin de saber cuándo y quien retira a alguno alumno sancionado de la base de datos, de esta manera se pretende evitar conflictos.



Figura 2.19 Clase Cancelar_Sancion

- **Sancion:** Cuando se sanciona a un alumno siempre se tiene que tomar nota de la sanción, con esta clase abarcamos todos los detalles, desde el tipo de sanción hasta cuando termina la misma.

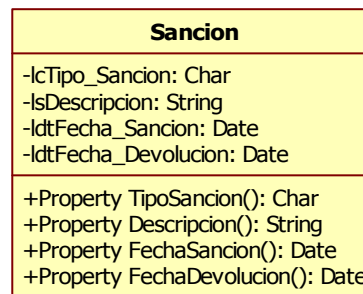


Figura 2.20 Clase Sancion

- **Computadora:** Una clase muy importante ya que aquí tratamos directamente a la computadora teniendo como propiedades el numero, el estado en que se encuentra y el tipo de sistema operativo que tiene, datos valioso para el correcto funcionamiento de SCOSU v2.

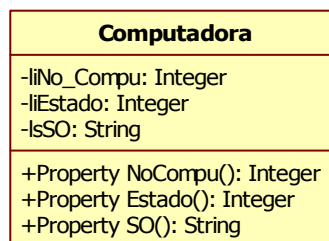


Figura 2.21 Clase Computadora

- **Modulo:** Para lograr un mejor funcionamiento en cuanto a los registros de SCOSU v2 se divido el tiempo de servicio en módulos de dos horas, la clase modulo contiene en número de modulo así como su tiempo de duración.

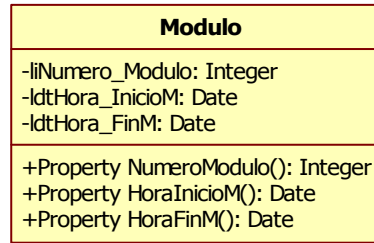


Figura 2.22 Clase Modulo

- **Curso:** UNICA como parte de la constante capacitación de los becarios, proporciona cursos internos, los cuales se imparten en la salas es por eso que es necesario apartar las salas con anticipación y los datos de ese cursos los maneja esta clase.

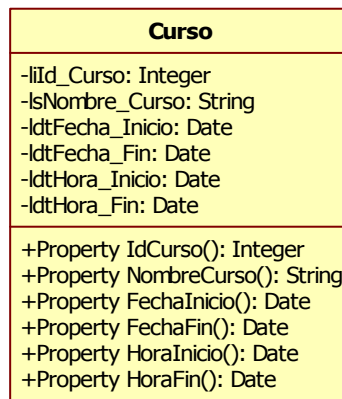


Figura 2.23 Clase Curso

- **Software:** Cada una de las salas contiene su propio software dependiendo del sistema operativo y del equipo con el que cuenta, además esta clase cuenta con una propiedad que maneja la última actualización para futuras revisiones.

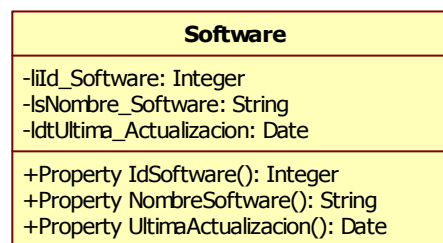


Figura 2.24 Clase Software

A continuación se muestra el diagrama de clases, aquí podemos observar como quedan relacionadas todas y cada uno de las ellas, por ejemplo la herencia que tiene entre algunas de las clases:

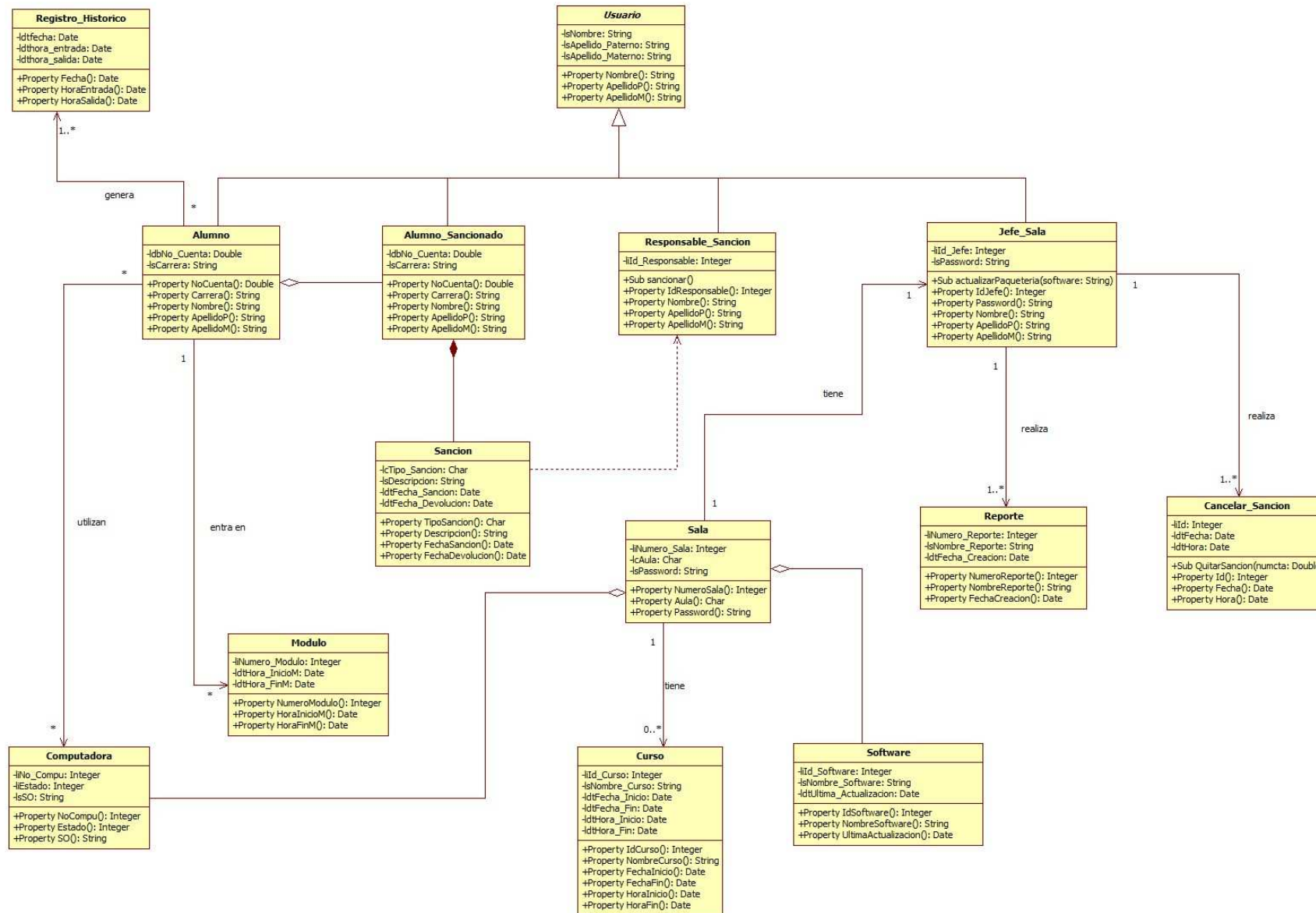


Figura 2.25 Diagrama de clases de SCOSU v2

2.2.3 Diagrama de actividades

El diagrama de actividades nos presenta una perspectiva de cómo será el flujo de las actividades que realiza el sistema. Estos se pueden obtener de diferente manera, ya sea basándose en una acción que realice o mediante los diferentes casos de uso que se obtengan, para el caso de SCOSU v2 se baso en los casos de uso obtenidos anteriormente.

A continuación se muestran los diferentes diagramas de actividades:

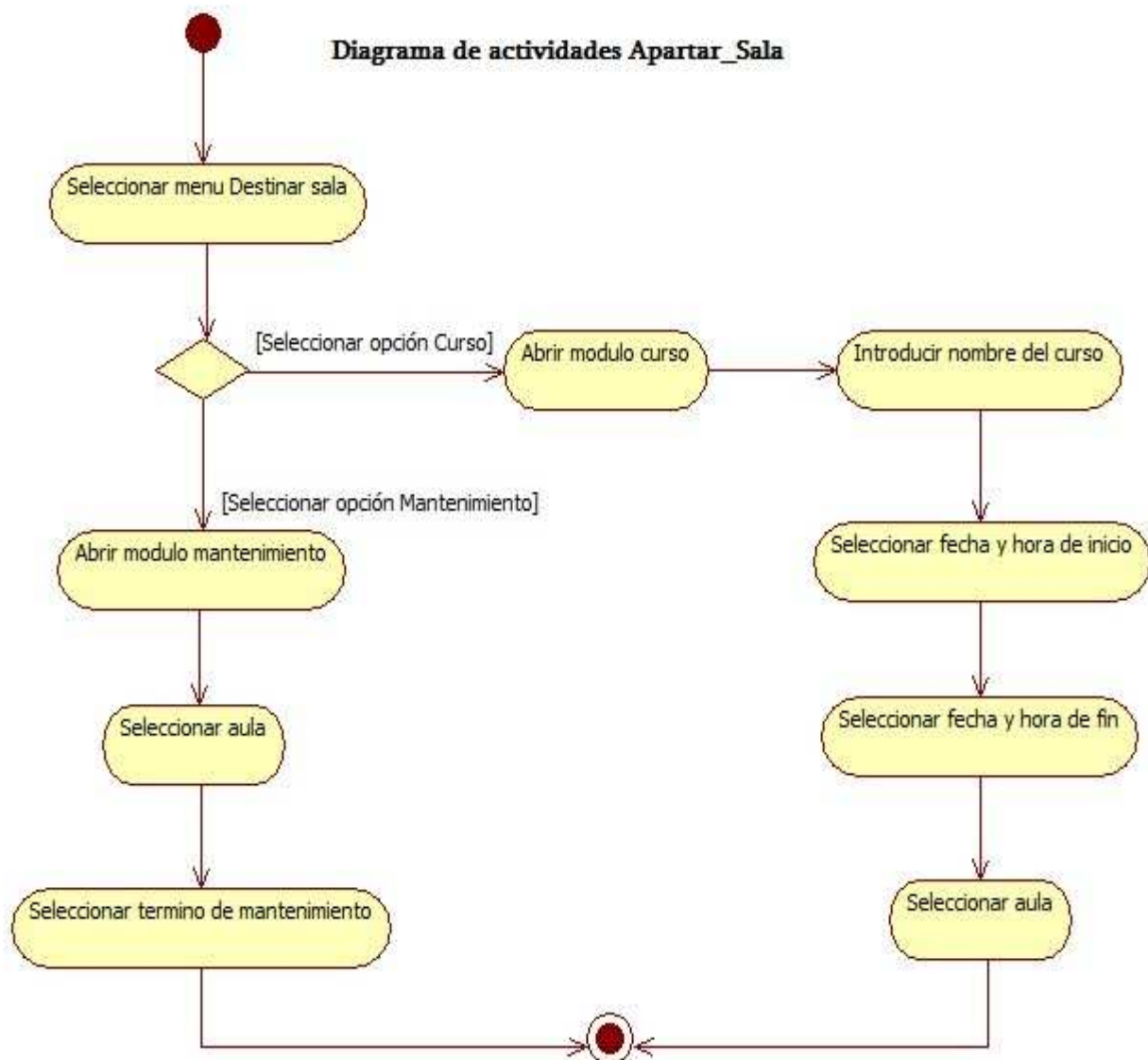


Figura 2.26 Diagrama de actividades. Casos de uso Apartar_Sala

Diagrama de actividad Asignar_Computadora

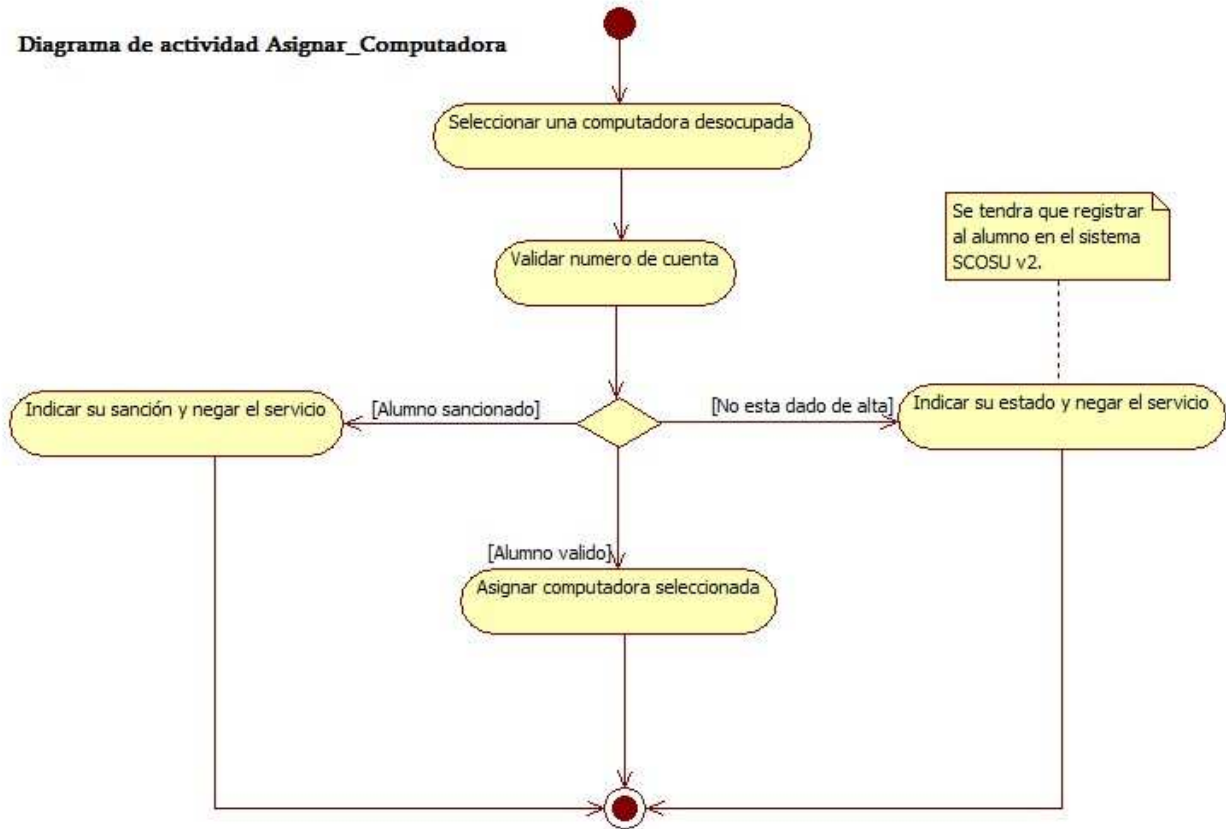


Figura 2.27 Diagrama de actividades. Caso de uso Asignar_Computadora

Diagrama de actividades Realizar_Busqueda_Alumno

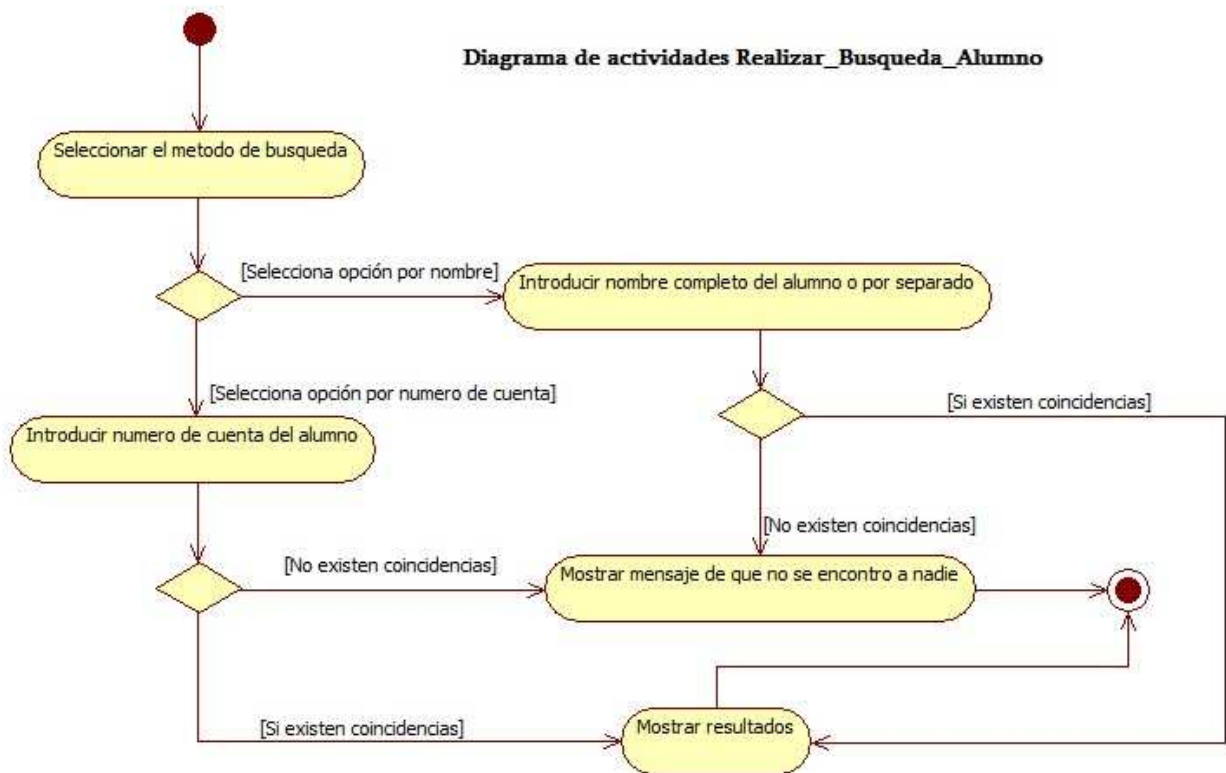


Figura 2.28 Diagrama de actividades. Caso de uso Realizar_Busqueda_Alumno

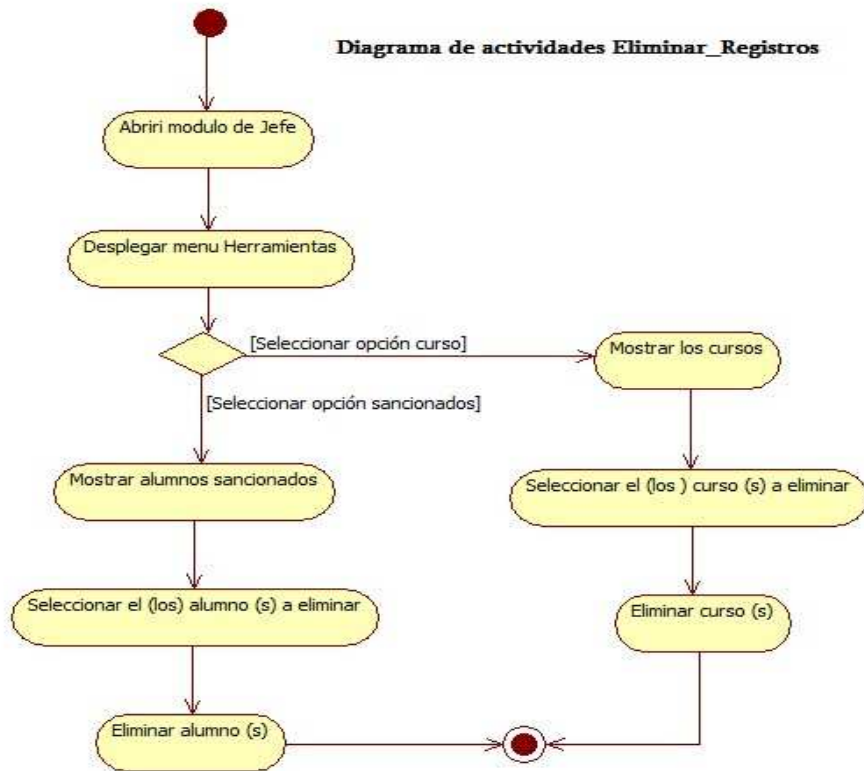


Figura 2.29 Diagrama de actividades. Caso de uso Eliminar_Registro

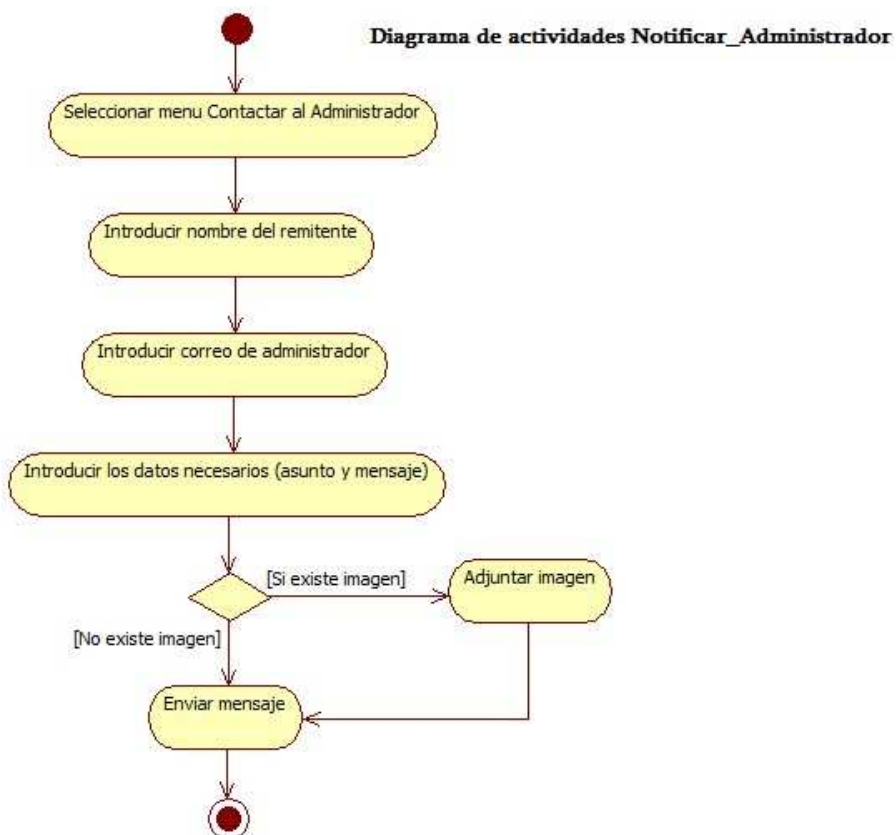


Figura 2.30 Diagrama de actividades. Caso de uso Notificar_Administrador

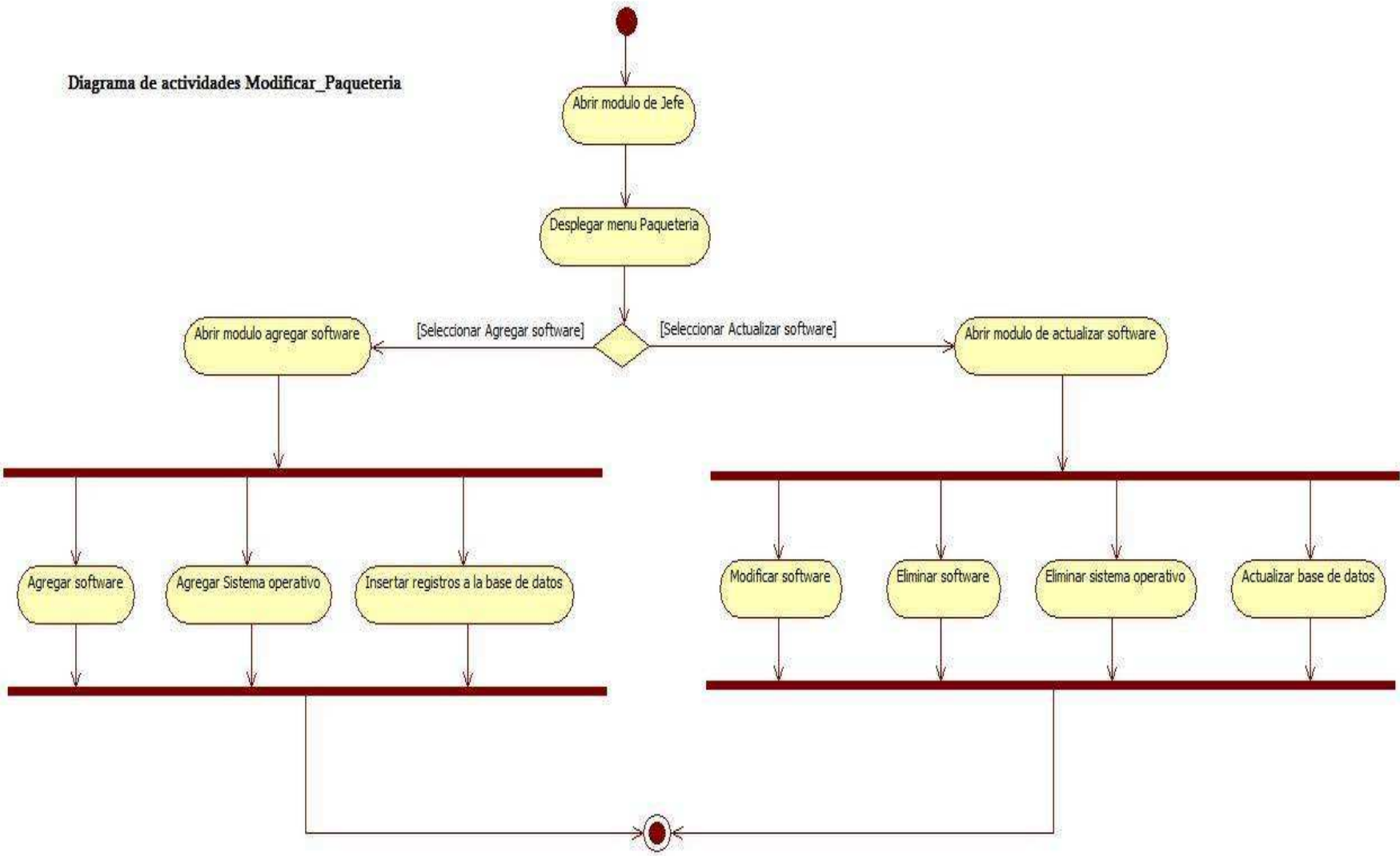


Figura 2.31 Diagrama de actividades. Caso de uso Modificar _ paquetería

Diagrama de actividades Manipular_Estados_Computadora

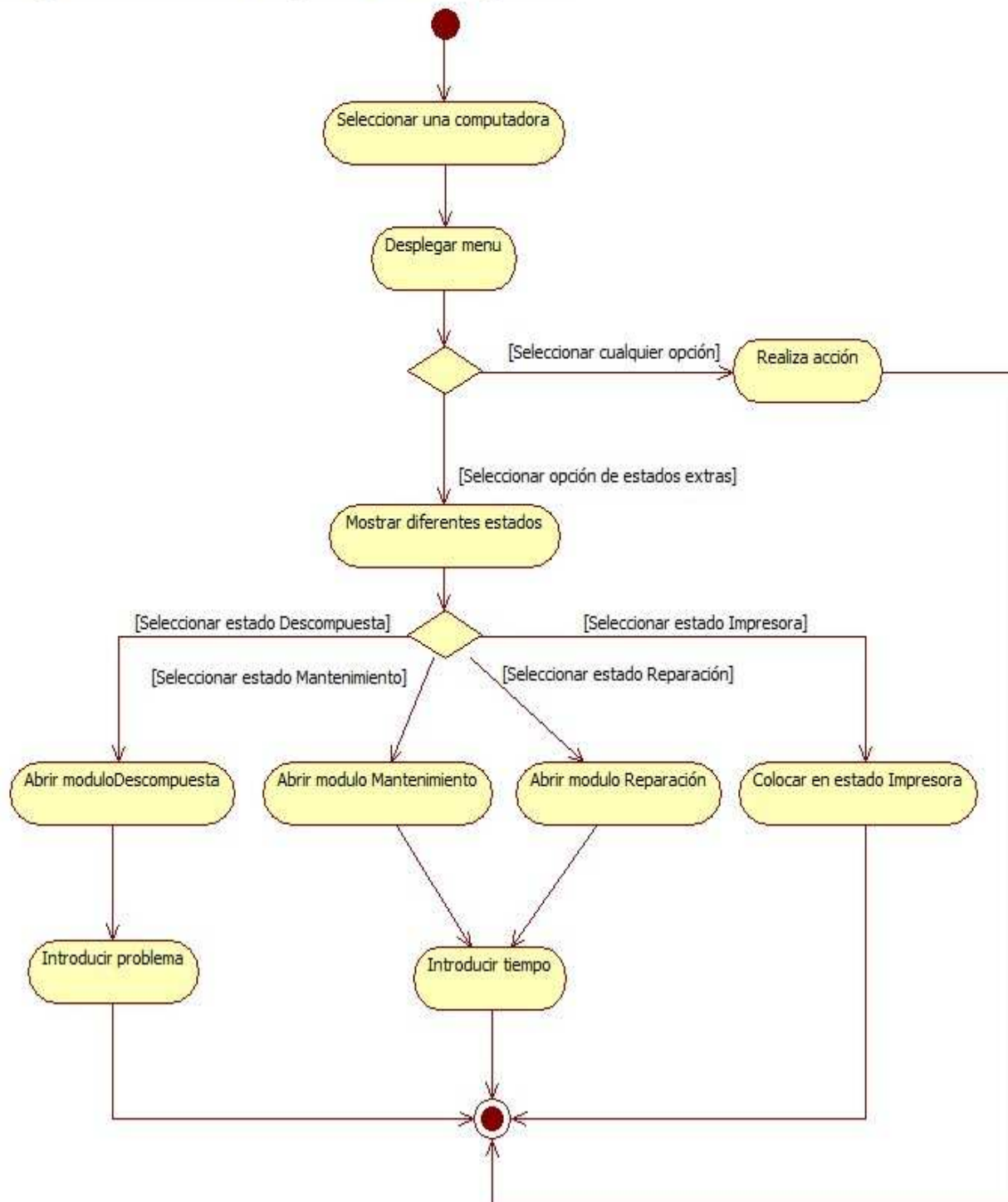


Figura 2.32 Diagrama de actividades. Caso de uso Manipular_Estados_Computadora

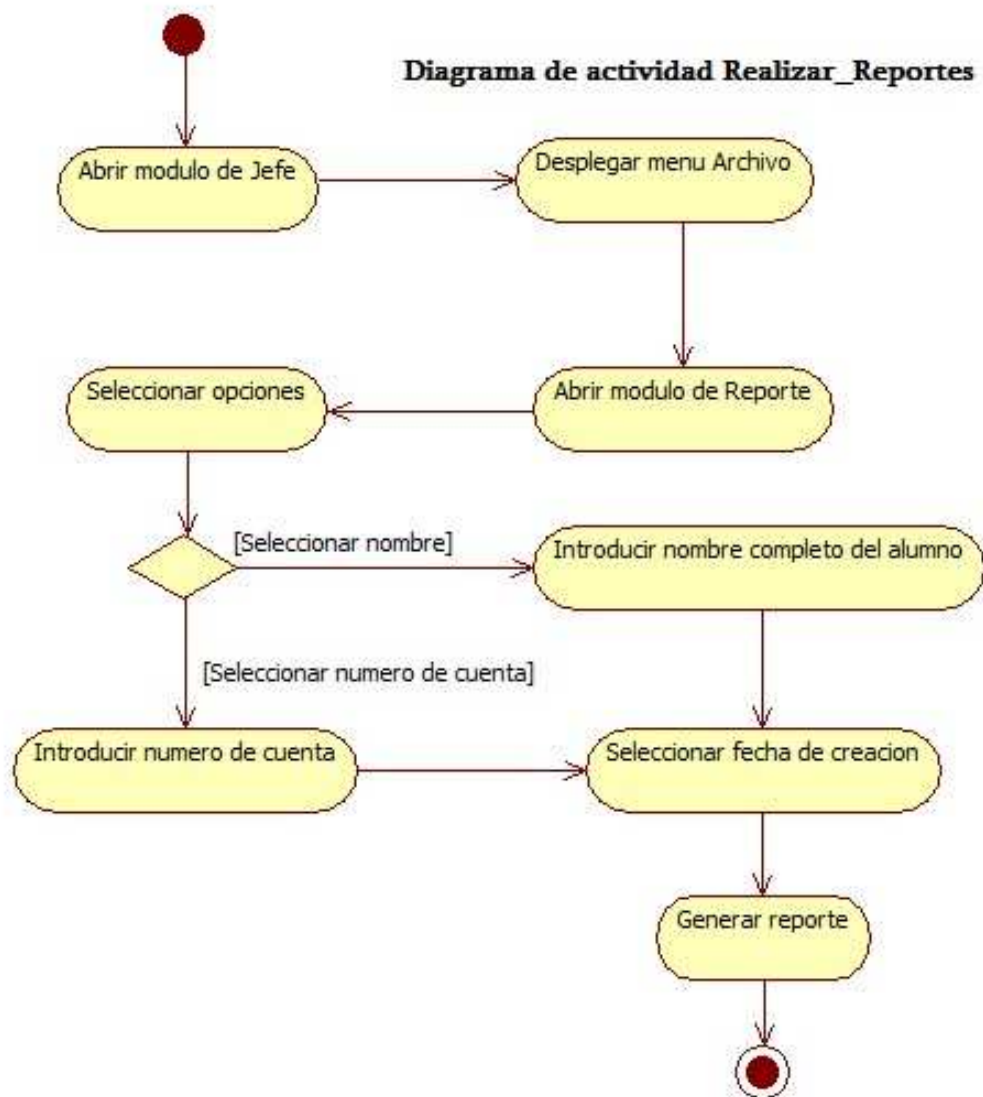


Figura 2.33 Diagrama de actividades. Caso de uso Realizar_Reportes

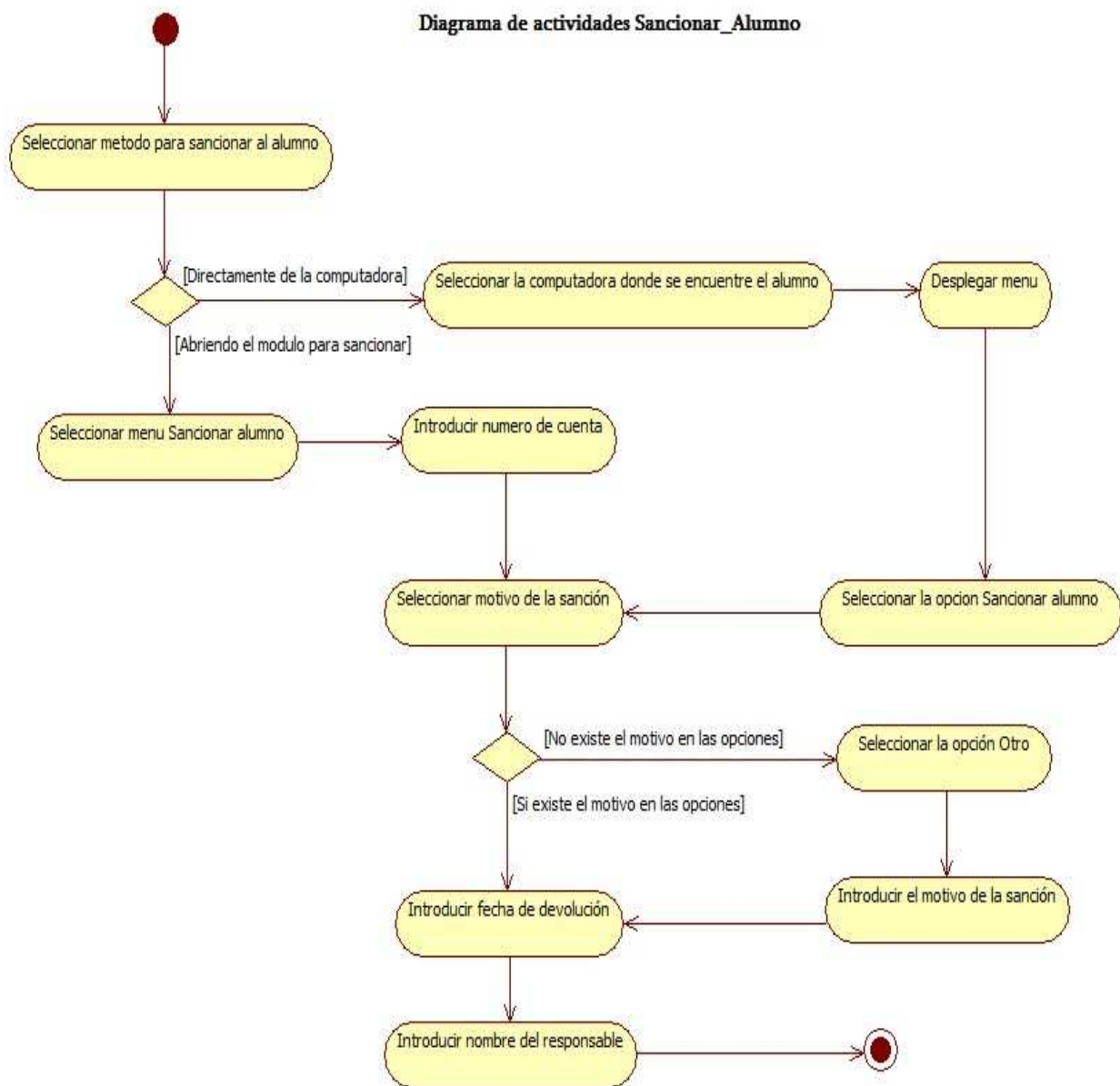


Figura 2.34 Diagrama de actividades. Caso de uso Sancionar_Alumno

Diagrama de actividades Sistema de Control de Asesorías

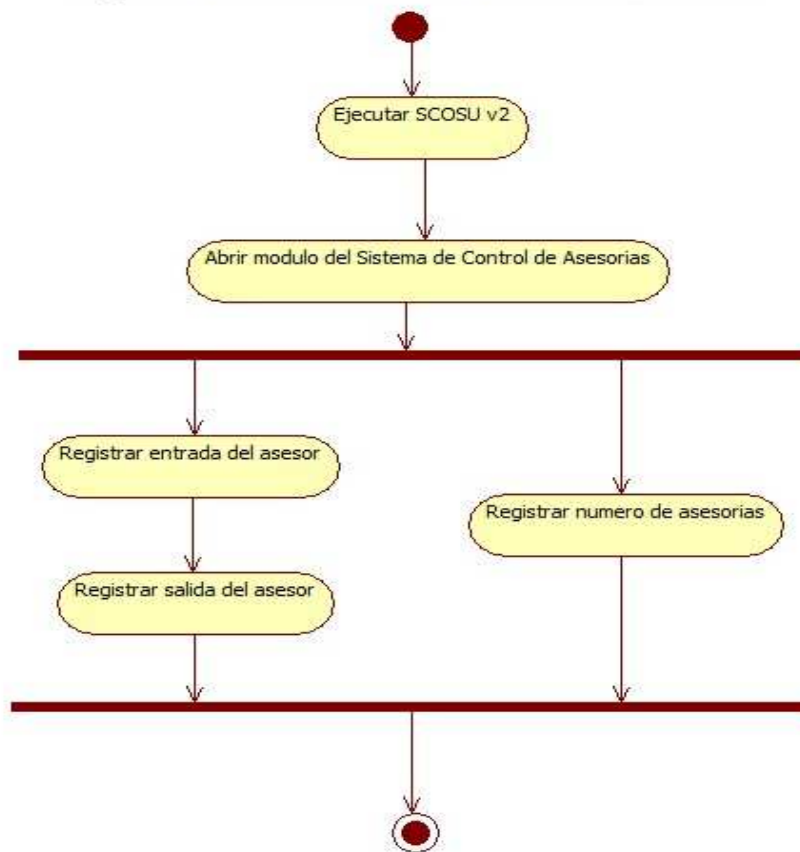


Figura 2.35 Diagrama de actividades. Sistema de Control de Asesorías

Diagrama de actividades Sistema de Control de Impresiones

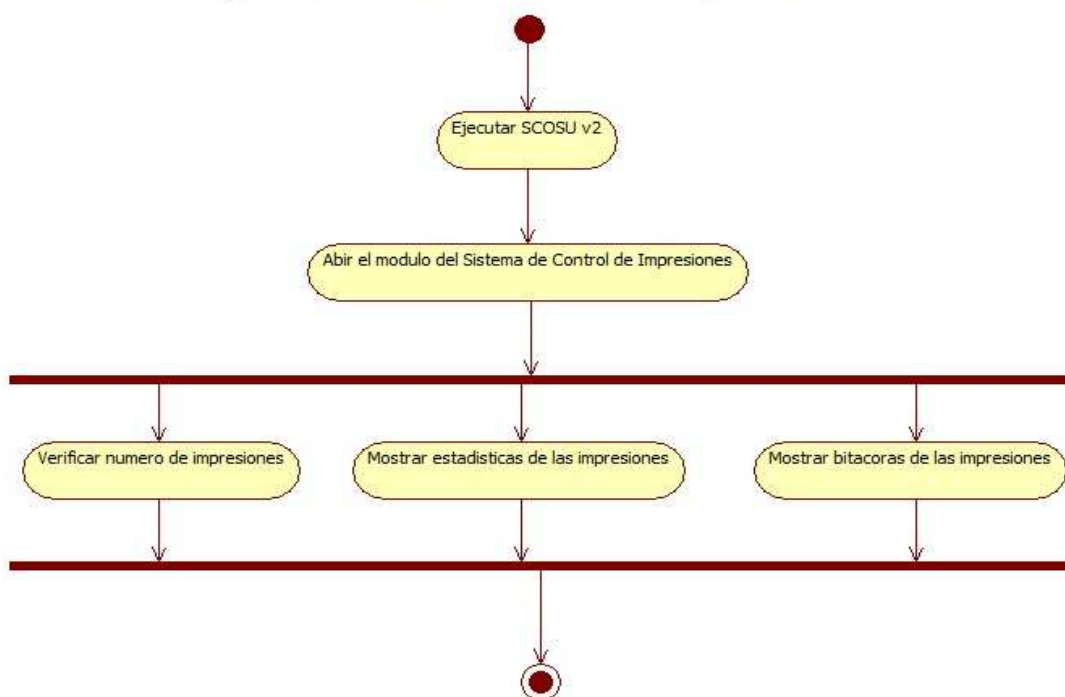


Figura 2.36 Diagrama de actividades. Sistema de Control de Impresiones

2.3 Diseño de la Base de datos

2.3.1. Identificación de entidades

Para el diseño de la base de datos se tomo en cuenta la problemática y las diferentes soluciones que se podía brindar. Se obtuvieron datos e información vital para poder crear las futuras tablas, obteniendo las siguientes entidades, las cuales ya se encuentran normalizadas:

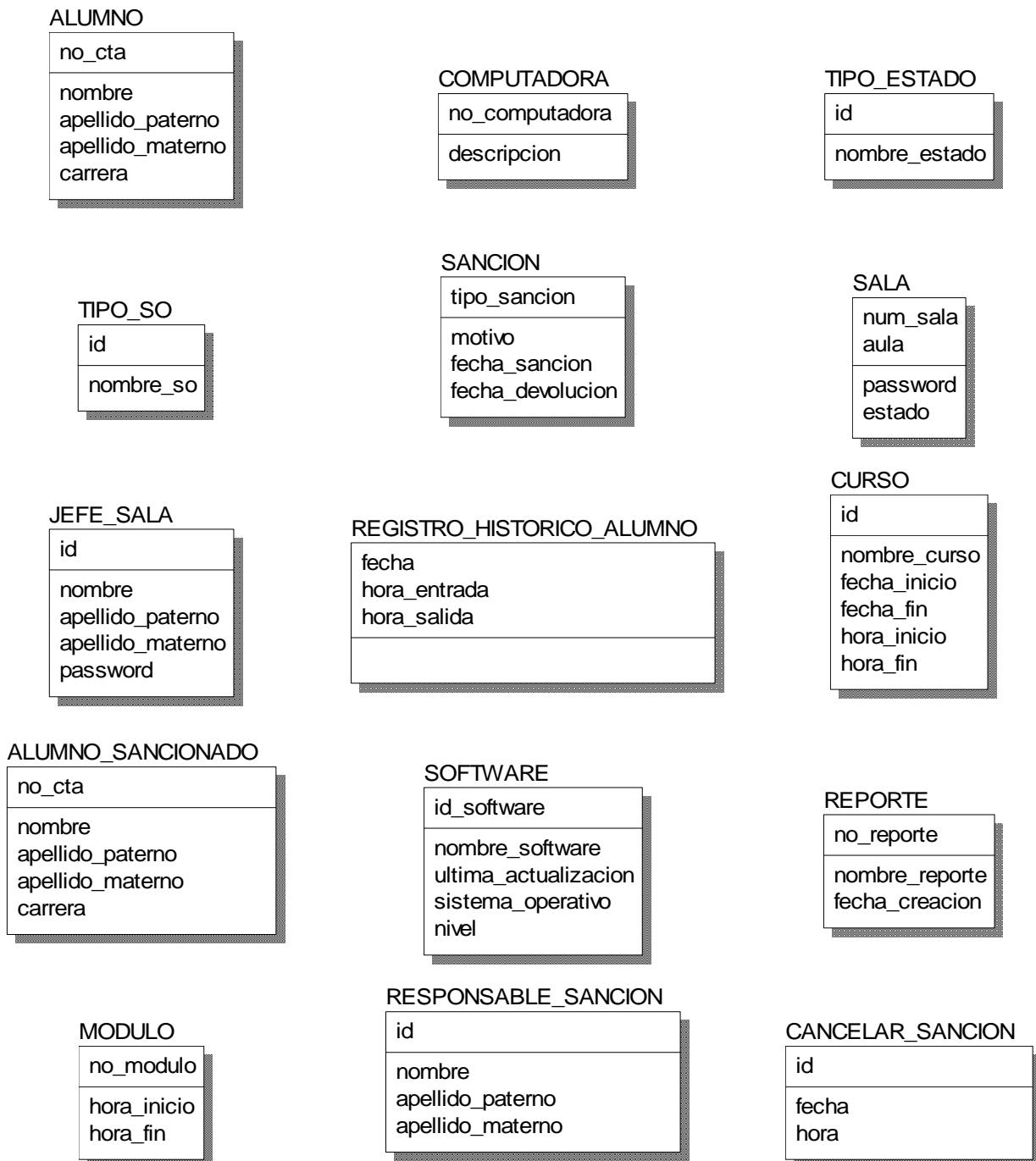


Figura 2.37 Entidades de la Base de datos de SCOSU v2

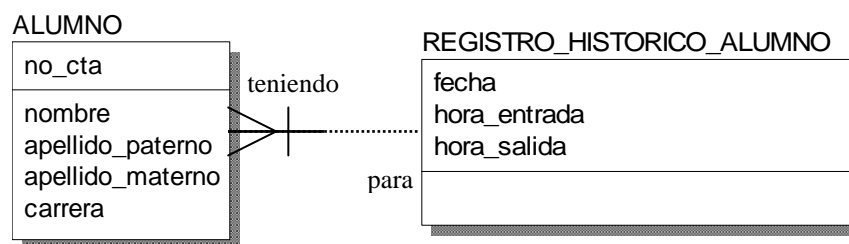
2.3.2. Descripción de entidades

- Un **ALUMNO** es un usuario que utilizara las computadoras de las salas de UNICA. Por ejemplo Claudia Nelly Arriaga Hernández.
- Una **COMPUTADORA** es un aparato que se encuentra dentro de la salas para realizar tareas o investigaciones. Por ejemplo computadora Dell con 80 GB de disco duro y 1 GB de memoria RAM.
- Un **TIPO_ESTADO** es como se encontraran las computadoras en las salas de UNICA. Por ejemplo descompuesta, ocupada, mantenimiento y más.
- Un **TIPO_SO** es el sistema operativo que tendrá cada computadora de las salas de UNICA. Por ejemplo Windows Vista.
- Una **SANCION** será el castigo que recibirá el **ALUMNO** en caso de que rompa con una regla de las salas de UNICA. Por ejemplo sanción tipo A.
- Una **SALA** es donde se encuentran las computadoras y donde se realizan los cursos. Por ejemplo sala 1.
- Un **JEFE_SALA** es la persona encargada de coordinar todas las actividades de cada una de las salas. Por ejemplo Cruz Sergio Aguilar Díaz.
- Un **REGISTRO_HISTORICO_ALUMNO** es la fecha, la hora de entrada y de salida que tendrá el alumno. Por ejemplo 19 de Febrero de 2007, 13:00, 15:00.
- Un **CURSO** es un evento de entrenamiento para los becarios de UNICA o un evento de aprendizaje para personas exteriores. Por ejemplo Visual Basic .NET.
- Un **ALUMNO_SANCIONADO** es un usuario que ha sido sancionado por haber cometido alguna falta al reglamento de las salas UNICA. Por ejemplo José Zamora García.
- Un **SOFTWARE** es un programa que se encuentra instalado en las computadoras de las salas de UNICA. Por ejemplo Microsoft Office 2007.
- Un **REPORTE** es un documento que realiza el jefe de la sala donde obtiene datos específicos de ciertos alumnos. Por ejemplo número de horas que ha utilizado las computadoras.

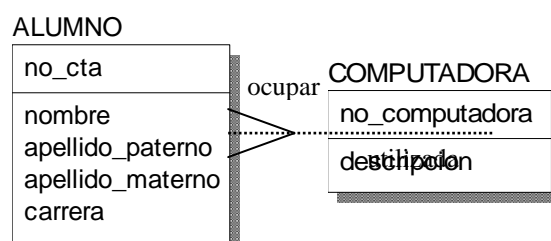
- Un **MODULO** es el tiempo en que está dividido el servicio que presta las salas a los alumnos, los cuales son de dos horas. Por ejemplo modulo 1 de 9:00 AM a 11:00 AM.
- Un **RESPONSABLE_SANCION** es la persona encargada de sancionar al alumno que cometa alguna falta al reglamento de las salas de UNICA. Por ejemplo Iván Carrillo Vera.
- **CANCELAR_SANCION** son los datos del día, mes, año y la hora en que un jefe de sala le quita la sanción a un alumno porque él así lo ha decidido.

2.3.3. Relaciones entre entidades

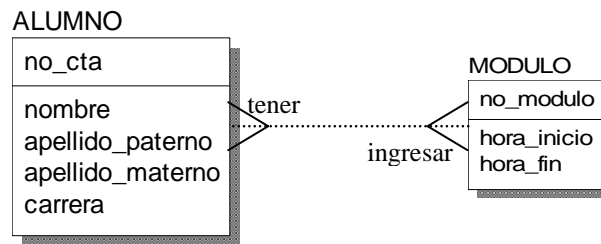
Una vez establecidas las diferentes entidades que surgieron a partir del problema, además de identificarlas mediante una breve descripción de lo que son cada una de ellas, se establecerá la relación existente entre cada una de las entidades



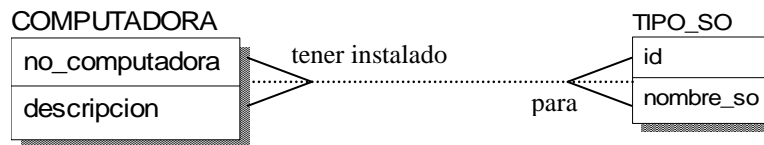
- Cada **ALUMNO** debe estar teniendo a uno y solamente un **REGISTRO_HISTORICO_ALUMNO**.
- Cada **REGISTRO_HISTORICO_ALUMNO** puede ser para uno o más **ALUMNOS**.



- Cada **ALUMNO** puede ocupar una y solo una **COMPUTADORA**.
- Cada **COMPUTADORA** puede ser utilizada por uno o más **ALUMNOS**.

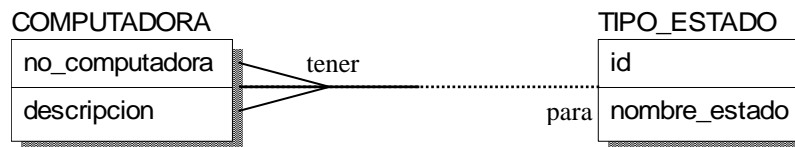


- Cada ALUMNO puede tener uno o más MODULOS.
- Cada MODULO puede ingresar a uno o más ALUMNOS.

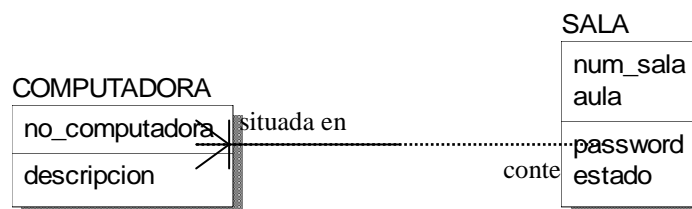


- Cada COMPUTADORA puede tener instalado uno o más TIPO_SO.
- Cada TIPO_SO puede ser para una o más COMPUTADORAS.

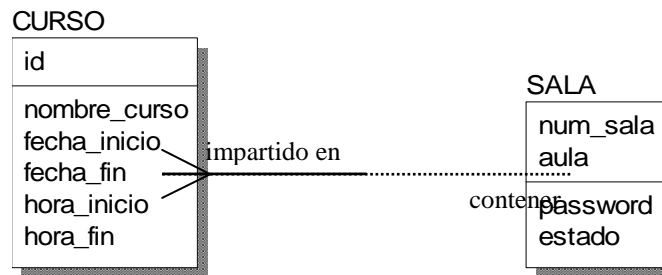
Nota: Las relaciones de M:M se rompen más adelante cuando se realiza el diseño en tablas.



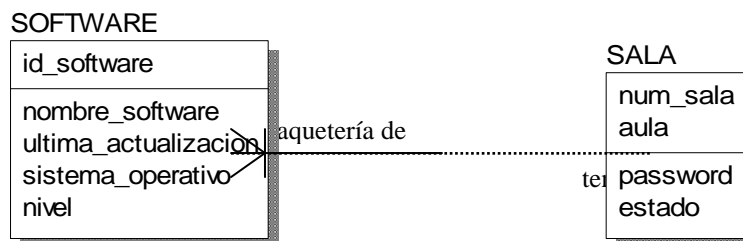
- Cada COMPUTADORA debe tener uno y solamente un TIPO_ESTADO.
- Cada TIPO_ESTADO puede ser para una o más COMPUTADORAS.



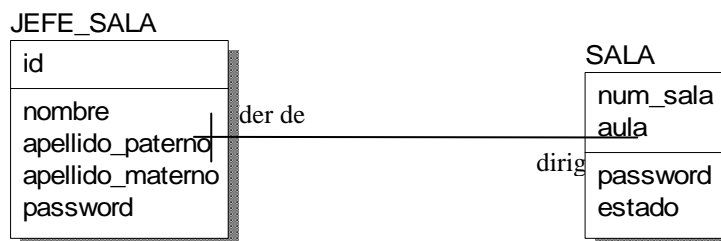
- Cada COMPUTADORA debe estar situada en una y solamente una SALA.
- Cada SALA puede contener una o más COMPUTADORAS.



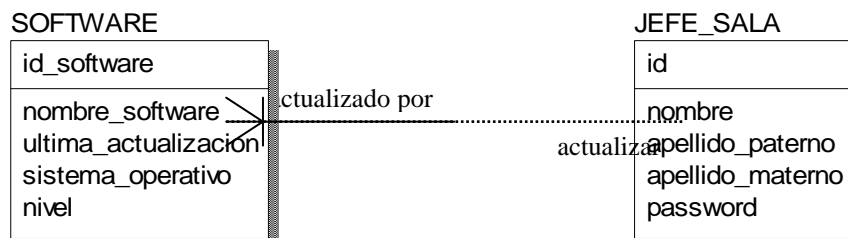
- Cada CURSO debe ser impartido en una y solamente una SALA.
- Cada SALA puede contener uno o más CURSOS.



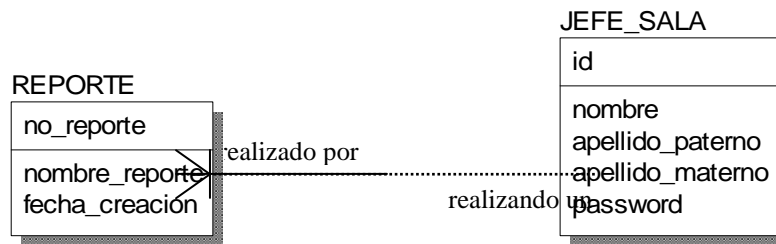
- Cada SOFTWARE debe ser paquetería de una y solamente una SALA.
- Cada SALA puede tener uno o más SOFTWARE.



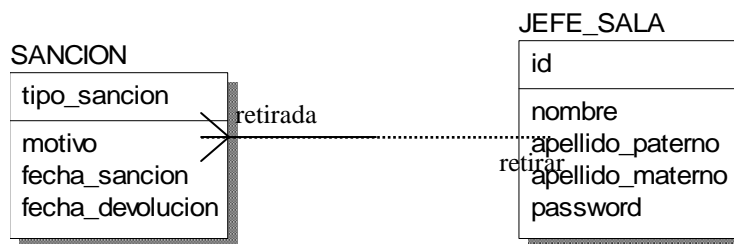
- Cada JEFE_SALA debe ser líder de una y solamente una SALA.
- Cada SALA debe ser dirigida por uno y solamente un JEFE_SALA.



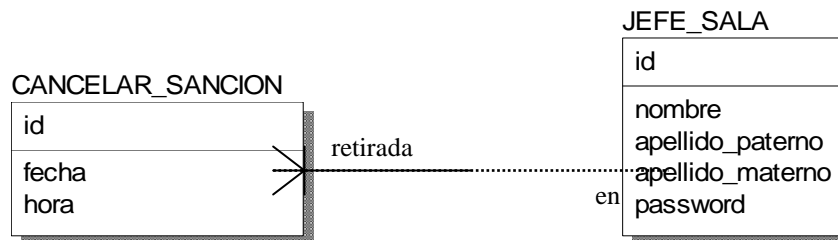
- Cada SOFTWARE debe ser actualizado por uno y solamente un JEFE_SALA.
- Cada JEFE_SALA puede actualizar uno o más SOFTWARE.



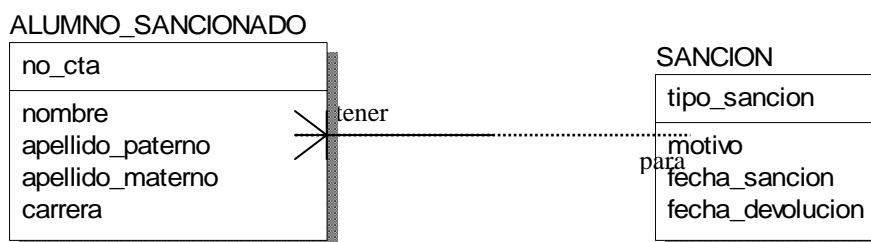
- Cada REPORTE debe ser realizado por uno y solamente un JEFE_SALA.
- Cada JEFE_SALA puede estar realizando un o más REPORTEES.



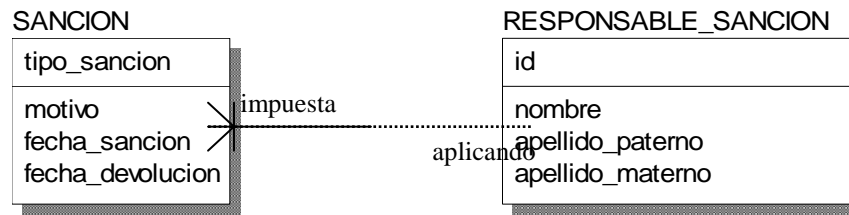
- Cada SANCION debe ser retirada por uno y solamente un JEFE_SALA.
- Cada JEFE_SALA puede retirar una o más SANCIONES.



- Cada CANCELAR_SANCION debe ser retirada por uno y solamente un JEFE_SALA.
- Cada JEFE_SALA puede ser uno o más CANCELAR_SANCIONES.



- Cada ALUMNO_SANCIONADO debe tener una y solamente una SANCION.
- Cada SANCION puede ser para uno o más ALUMNOS_SANCIONADOS.



Cada SANCION debe ser impuesta por uno y solamente un RESPONSABLE_SANCION. Cada RESPONSABLE_SANCION puede estar aplicando una o más SANCIONES.

2.3.4. Diseñando la Base de datos

A continuación se muestra cómo es que quedaran las tablas dentro de nuestras Base de datos basándonos en las relaciones mostradas anteriormente, en ellas se verán todos los atributos de cada tabla, incluyendo las llaves foráneas y las llaves primarias. Cabe destacar que las tablas ya se encuentran normalizadas.

Nombre de la tabla: RESPONSABLES_SANCIONES

COLUMNA	ID_RESPONSABLE	NOMBRE	APELLIDOP	APELLIDOM
TIPO LLAVE	PK			
NULOS/UNICOS	NN - U	NN	NN	NN
EJEMPLO	1	Sergio	Cruz	Aguilar
	2	Claudia	Arriaga	Hernández
	3	Javier	Ocampo	Montoya

Nombre de la tabla: CANCELAR_SANCIONES

COLUMNA	ID_CANCELAR	FECHA	HORA	ID_JEFE	NUM_SALA	AULA
TIPO LLAVE	PK			PK2,FK1	PK3,FK2	PK4,FK3
NULOS/UNICOS	NN - U	NN	NN	NN - U	NN - U	NN - U
EJEMPLO	1	19/02/2007	18:36:12	2	2	A
	2	23/10/2007	10:48:01	3	3	F
	3	10/04/2008	13:45:52	2	2	D

Nombre de la tabla: SALAS

COLUMNA	NUM_SALA	AULA	PASSWORD	ESTADO
TIPO LLAVE	PK			
NULOS/UNICOS	NN - U	NN	NN	NN
EJEMPLO	1	1	sala_uno	4
	2	A	sala_dos	4
	3	H	sala_tres	5

Nombre de la tabla: TIPOS_ESTADOS

COLUMNA	ID_ESTADO	NOM_ESTADO
TIPO LLAVE	PK	
NULOS/UNICOS	NN – U	NN
EJEMPLO	1	Desocupada
	2	Descompuesta
	3	Mantenimiento

Nombre de la tabla: MODULOS

COLUMNA	NO_MODULO	HORA_INICIO	HORA_FIN
TIPO LLAVE	PK		
NULOS/UNICOS	NN – U	NN	NN
EJEMPLO	1	09:00:00	10:59:59
	2	11:00:00	13:59:59
	3	14:00:00	16:59:59

Nombre de la tabla: TIPOS_SO

COLUMNA	ID_TIPO	NOMBRE_SO
TIPO LLAVE	PK	
NULOS/UNICOS	NN – U	NN
EJEMPLO	1	Windows 2000
	2	Windows XP
	3	Linux

Nombre de la tabla: REGISTROS_HISTORICOS_ALUMNOS

COLUMNA	FECHA	HORA_ENTRADA	HORA_SALIDA
TIPO LLAVE	PK		
NULOS/UNICOS	NN – U	NN	NN
EJEMPLO	20/10/2007	13:24:56	15:02:41
	18/11/2007	18:05:02	20:00:05
	26/08/2007	19:32:12	20:25:39

Nombre de la tabla: COMPUTADORAS

COLUMNA	NO_COMPUTADORA	DESCRIPCION	ID_ESTADO	NUM_SALA	AULA
TIPO LLAVE	PK		FK1	PK, FK2	PK2, FK3
NULOS/UNICOS	NN – U	N	NN	NN – U	NN - U
EJEMPLO	23	Duración: 10 días	1	2	
	23	Problema: virus	4	3	
	45		1	4	

La siguiente tabla surgió de haber roto la relación M:M de las entidades COMPUTADORA con TIPO_SO.

Nombre de la tabla: COMPUTADORAS_SO

COLUMNA	NO_COMPUTADORA	NUM_SALA	AULA	ID_TIPO
TIPO LLAVE	PK1, FK1	PK2, FK2	PK4,FK4	PK3, FK3
NULOS/UNICOS	NN – U	NN – U	NN – U	NN – U
EJEMPLO	1	2	A	3
	45	3	H	1
	19	1	2	3

La siguiente tabla surgió de haber roto la relación M:M de las entidades ALUMNO con MODULO.

Nombre de la tabla: ALUMNOS_MODULOS

COLUMNA	NO_CTA	NO_MODULO	FECHA	HORA_ENTRADA	HORA_SALIDA
TIPO LLAVE	PK1, FK1	PK2, FK2	PK3, FK3	PK4, FK4	PK5, FK5
NULOS/UNICOS	NN – U	NN – U	NN – U	NN – U	NN - U
EJEMPLO	099103008	4	05/05/2008	15:52:05	16:20:30
	099335737	1	22/03/2008	09:05:23	10:30:21
	300548976	3	10/02/2008	13:50:00	14:45:23

Nombre de la tabla: ALUMNOS

COLUMNA	NO_CTA	NOMBRE	APELLIDO P	APELLIDO M	CARRERA	NUM_COM P	NUM_SAL A	AULA	FECHA	HORA_ENT	HORA_SAL
TIPO LLAVE	PK					FK1	FK2	FK3	PK2,FK4	PK3, FK5	PK4, FK6
NULOS/UNICOS	NN – U	NN	NN	NN	NN	NN	NN	NN	NN – U	NN – U	NN - U
EJEMPLO	099103008	Víctor	Dueñas	Tello	Ing. en Computación	56	2	A	22/03/08	13:02:00	14:52:03
	300250668	Juan Carlos	Bruno	Cuevas	Ing. en Computación	102	3	E	02/05/08	17:00:02	18:49:02
	402567892	Iván	Carrillo	Vera	Ing. en Computación	15	1	1	07/04/08	10:20:00	11:56:00

Nombre de la tabla: ALUMNOS_SANCIONADOS

COLUMNA	NO_CTA	NOMBRE	APELLIDOP	APELLIDOM	CARRERA	TIPO_SANCION	ID_RESPONSABLE
TIPO LLAVE	PK					PK2,FK1	PK3,FK2
NULOS/UNICOS	NN – U	NN	NN	NN	NN	NN – U	NN – U
EJEMPLO	099103008	Víctor	Dueñas	Tello	Ing. en Computación	A	1
	099335737	José	Zamora	García	Ing. Eléctrica Electrónica	F	3
	099654685	Antonio	Carrizosa	Martínez	Ing. en Computación	B	5

Nombre de la tabla: CURSOS

COLUMNA	ID_CURSO	NOM_CURSO	FECHA_INICIO	FECHA_FIN	HORA_INICIO	HORA_FIN	NUM_SALA	AULA
TIPO LLAVE	PK						FK1	FK2
NULOS/UNICOS	NN – U	NN	NN	NN	NN	NN	NN	NN
EJEMPLO	1	Linux Básico	05/02/2008	24/02/2008	09:00:00	12:00:00	2	B
	3	Visual Basic 6.0	05/02/2008	24/02/2008	09:00:00	12:00:00	3	H
	8	Fundamentos de Java	22/09/2007	07/10/2007	09:00:00	13:00:00	3	E

Nombre de la tabla: SANCIONES

COLUMNA	TIPO_SANCION	MOTIVO	FECHA_SANCION	FECHA_DEVOLUCION	ID_RESPONSABLE	ID_JEFE	NUM_SALA	AULA
TIPO LLAVE	PK1				PK2, FK1	FK2	FK3	FK4
NULOS/UNICOS	NN – U				NN - U	NN	NN	NN
EJEMPLO	A	Más de 15 Hojas	17/09/2007	01/10/2007	3	1	2	C
	B	Ambos lados de la hoja	18/09/2007	02/10/2007	2	3	4	4
	F	Paginas XXX	27/09/2007	27/10/2007	5	2	1	2

Nombre de la tabla: JEFES_SALAS

COLUMNA	ID_JEFE	NOMBRE	APELLIDOP	APELLIDOM	PASSWORD	NUM_SALA	AULA
TIPO LLAVE	PK					PK2, FK1	PK3, FK2
NULOS/UNICOS	NN – U	NN	NN	NN	NN	NN - U	NN - U
EJEMPLO	1	Sergio	Cruz	Aguilar	\$C0sU	2	D
	2	Javier	Montoya	Pérez	15SFs#	3	G
	3	Gabriela	Camacho	Álvarez	%&Gabi	4	4

Nombre de la tabla: SOFTWARES

COLUMNA	ID_SOFTWARE	NOM_SOFTWARE	ULTIMA_ACTUALIZACION	NIVEL	SISTEMA_OPERATIVO	ID_JEFE	NUM_SALA	AULA
TIPO LLAVE	PK					PK2,FK1	PK3,FK2	PK4,FK3
NULOS/UNICOS	NN – U	NN	NN	NN	NN	NN – U	NN - U	NN - U
EJEMPLO	12	Office 2007	20/02/2007	1	Windows Vista	1	2	F
	5	Macromedia MX	15/11/2006	1	Windows XP	2	3	G
	1	Internet Explorer 6.0	22/06/2007	1	Windows 2000	3	4	E

Nombre de la tabla: REPORTEES

COLUMNA	NO_REPORTE	NOM_REPORTE	FECHA_CREACION	ID_JEFE	NUM_SALA	AULA
TIPO LLAVE	PK			PK2, FK1	PK3, FK2	PK4, FK3
NULOS/UNICOS	NN –U	NN	NN	NN – U	NN - U	NN - U
EJEMPLO	3	Desperfectos	10/10/2007	2	3	H
	1	Fallas técnicas	30/11/2007	3	4	4
	6	Avisos	20/01/2008	1	2	A

2.3.5. Esquema de la Base de datos

Por último se muestra en la siguiente figura el Diagrama Entidad Relación, nótese que se muestra cómo es que se rompieron las relaciones M:M, por otra parte los atributos que se encuentran antes de la línea son las llamadas llaves primarias que como se puede observar algunas son compuestas.

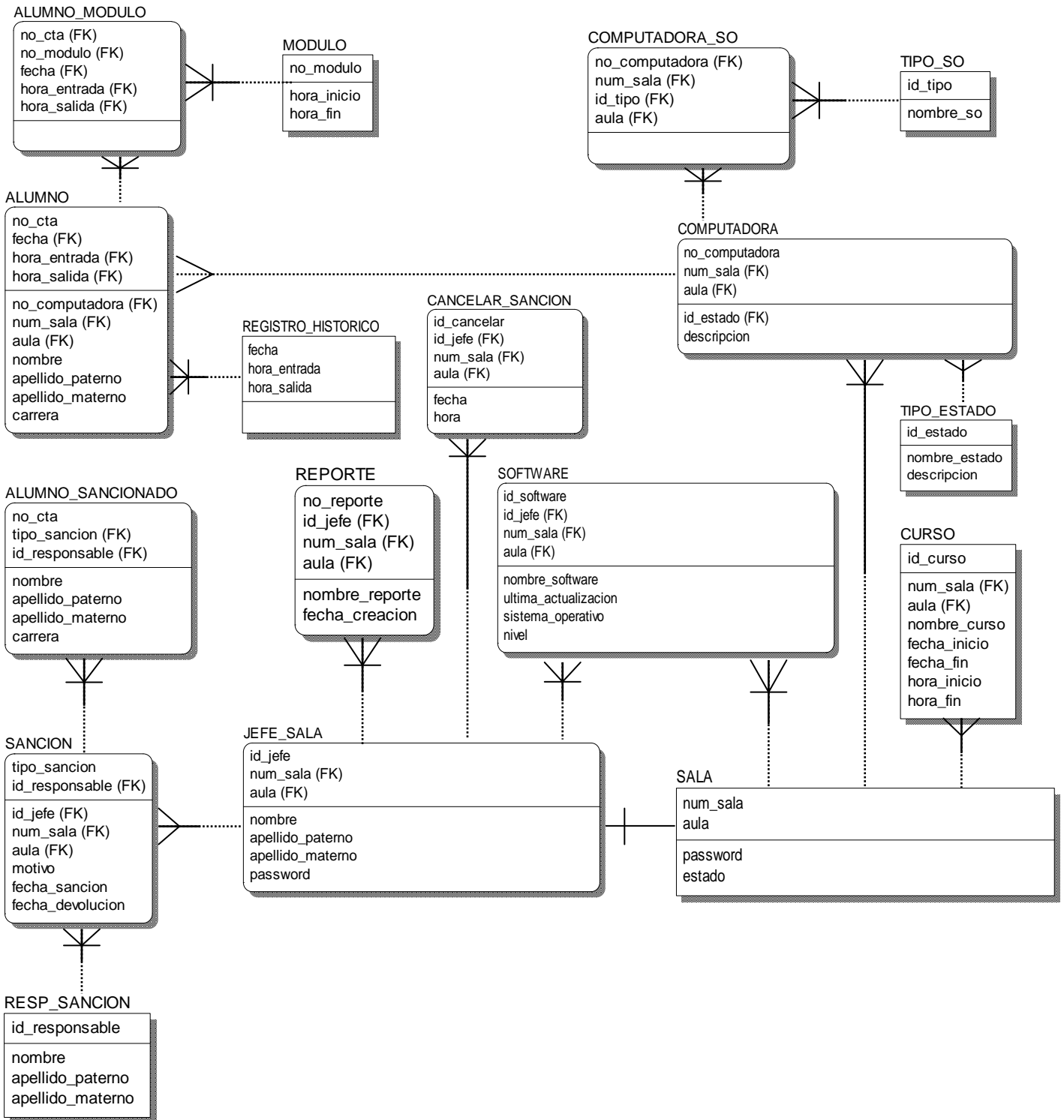


Figura 2.38 Diagrama Entidad Relación de SCOSU v2 (en Erwin)

2.4 Diseño del sistema

¿Por qué Visual Basic?

La interfaz grafica es muy importante en cualquier sistema, y gracias a los nuevos lenguajes es posible realizarlas de una manera más sencilla y con un formato amigable para el usuario y al mismo tiempo profesional, también Visual Basic se podrá correr (ejecutar) en casi todos los sistemas operativos de Microsoft, y lo mejor, bueno a mi opinión, es mucho mas fácil de realizar los códigos, ya que tiene una ayuda de auto complementar

Es por eso que Visual Basic 2005 fue una excelente elección, tiene un Entorno de Desarrollo Integrado (IDE por sus siglas en ingles) muy eficiente y potente el cual te permite realizar aplicaciones con una excelente presentación. Cabe mencionar que VB .Net es uno de los muchos lenguajes que contiene Visual Studio 2005, los restantes que maneja son Visual C#, Visual J# y Visual C++.

En cuanto al lenguaje también es muy potente, VB .Net esta completamente orientado a objetos teniendo los beneficios de este lenguaje tales como la reutilización de código, herencia y más. Por todo esto y por que la conectividad a la Base de datos es eficiente y fácil de administrar, además de ser segura, fue que se eligió VB .Net por encima de los otros lenguajes de programación que maneja Visual Studio 2005.

Nota: Se eligió esta versión debido a que cuando se inicio el sistema todavía no se encontraba en el mercado la más reciente (Visual Studio 2008).

A continuación se tratara mas a fondo las características de este VB .Net viendo sus ventajas y desventajas, así mismo se verá la alternativa para maneja la base de datos la cual fue ADO .Net el cual es parte de VB .Net.

2.4.1 Visual Basic 2005²

¿Qué es .NET?

.NET es toda una nueva arquitectura tecnológica, desarrollada por Microsoft para la creación y distribución del software como un servicio. Esto quiere decir, que mediante las herramientas de desarrollo proporcionadas por esta nueva tecnología, los programadores podrán crear aplicaciones basadas en servicios para la Web. Las características principales que conforman .NET son las siguientes:

² Programación en Visual Basic .Net, Luis Miguel Blanco, Grupos EIDOS.

- La plataforma .NET Framework 2.0, que proporciona la infraestructura para crear aplicaciones y el entorno de ejecución para las mismas.
- Servicios para .NET desarrollados por terceros fabricantes, que podrán ser utilizados por otras aplicaciones que se ejecuten en Internet.

Por otra parte .Net es una nueva tecnología que responde a la necesidad de comunicación total entre negocios de sistemas heterogéneos y con gran capacidad de procesamiento.

Las aplicaciones creadas en la plataforma .Net podrán ser usadas por un gran número de clientes, tales como PC, PC de bolsillo (Palm PC), celulares digitales, PC de juegos (PC Games), PC de auto (Auto PC), PC Televisor (Ultimate TV) y más.

.NET Framework 2.0

.NET Framework 2.0 constituye la nueva plataforma y elemento principal sobre el que se asienta Microsoft .NET. De cara al programador, es la pieza fundamental de todo este nuevo modelo de trabajo, ya que proporciona las herramientas y servicios que necesitará en su labor habitual de desarrollo.

.NET Framework 2.0 permite el desarrollo de aplicaciones a través del uso de un conjunto de herramientas y servicios que proporciona, y que pueden agruparse en tres bloques principales: el Entorno de Ejecución Común o Common Language Runtime (CLR a partir de ahora); la jerarquía de clases básicas de la plataforma o .NET Framework Base Classes; y el motor de generación de interfaz de usuario, que permite crear interfaces para la web o para el tradicional entorno Windows, así como servicios para ambos entornos operativos. La Figura 2.39 muestra un diagrama con la distribución de elementos dentro del entorno de Microsoft .Net Framework.

En la base del entorno de ejecución, se encuentra el CLR, que constituye el núcleo de .NET Framework 2.0, encargándose de la gestión del código en cuanto a su carga, ejecución, manipulación de memoria, seguridad y más.

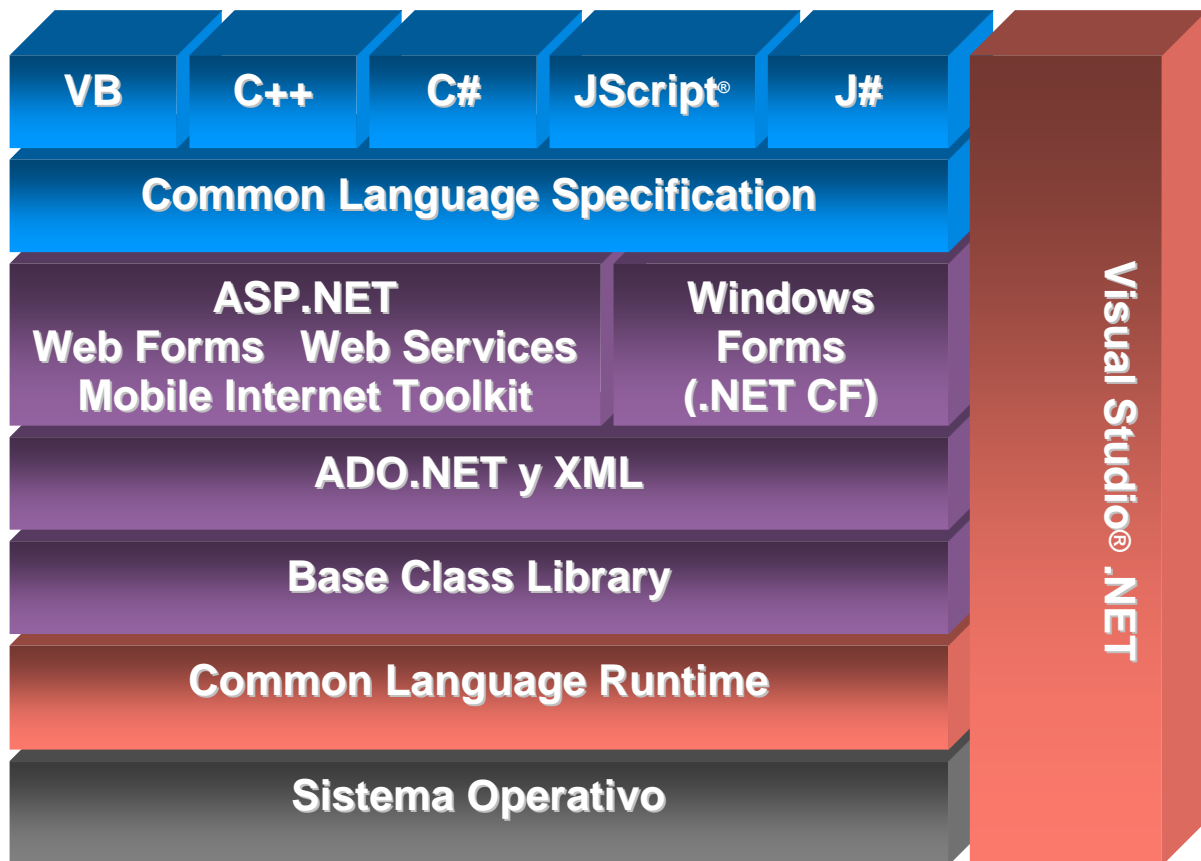


Figura 2.39 Esquema de componentes dentro de la plataforma .NET Framework.

En el nivel intermedio, se sitúa la jerarquía de clases básicas del entorno de ejecución, que constituyen un sólido API de servicios a disposición del programador, para multitud de tareas como, gestión del sistema de ficheros, manipulación multihebra, acceso a datos y más.

Finalmente, en el nivel superior, encontramos las clases que permiten el diseño del interfaz de usuario de nuestras aplicaciones. Si necesitamos desarrollar aplicaciones para Internet, utilizaremos ASP.NET, que nos provee de todo lo necesario para crear aplicaciones para la red: web forms, web services y más. Aunque se ve que todo en .NET Framework es programación para Internet no lo es, ya que pone a disposición de los programadores de sistemas para Windows los denominados Windows Forms, la nueva generación de formularios, con características avanzadas y muy superiores a las del motor de generación de formularios de Visual Basic 6.

Adicionalmente, existe la posibilidad de que necesitemos servicios del sistema que no requieran interfaz de usuario en absoluto. Este aspecto también está contemplado por la plataforma, permitiéndonos, por ejemplo, la creación de servicios para Windows 2000 y NT.

El entorno de desarrollo integrado (IDE)

El disponer en la actualidad de un entorno de desarrollo (IDE) para una herramienta de programación, puede parecer algo natural o incluso elemental para el propio lenguaje. Ello hace que en ocasiones no se conceda la importancia que realmente tiene a este aspecto del desarrollo. A continuación se muestran los elementos más importantes del IDE de Visual Basic 2005.

De forma predominante y ocupando la mayor parte del IDE, encontramos la ventana o zona principal de trabajo. Esta ventana principal contiene todos los editores de código y diseñadores que vayamos abriendo, organizados en base a unas fichas o pestañas, que nos permiten trasladarnos de uno a otro cómodamente.

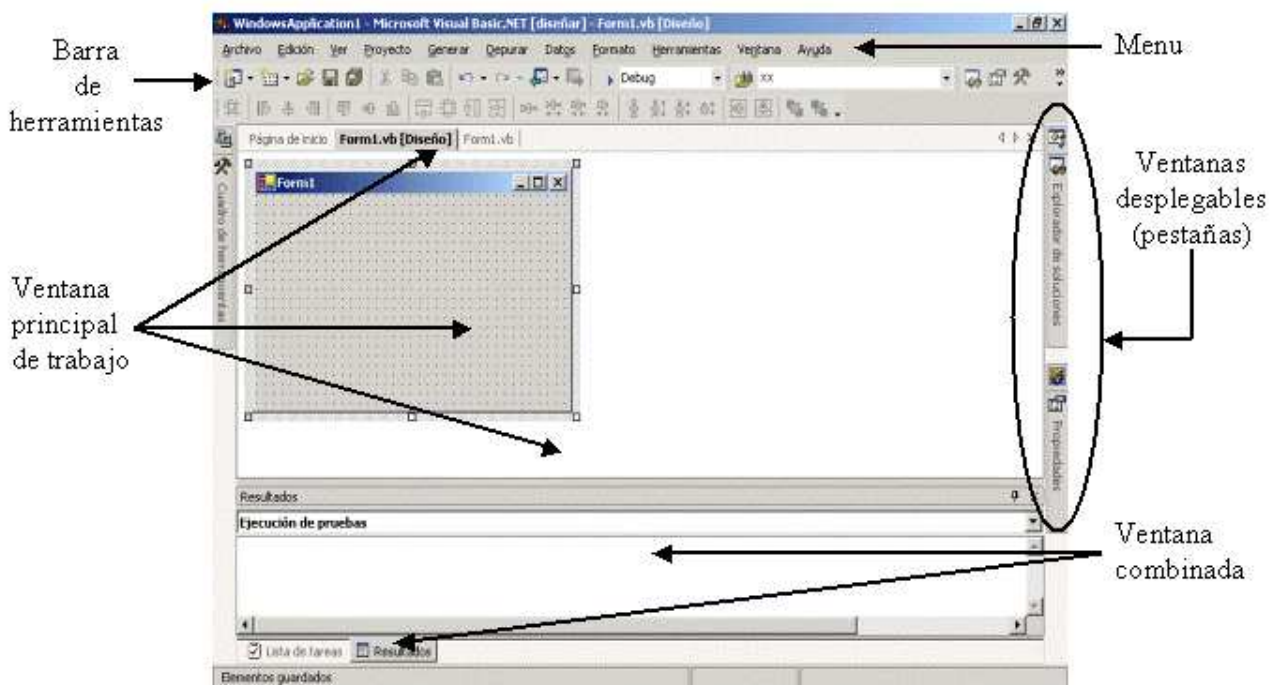


Figura 2.40 Ventana de diseño

El editor de código es como si nombre lo dice, donde colocamos todo lo relacionado con la programación, aquí es donde se programa todos y cada uno de los eventos que realizara nuestro sistema.

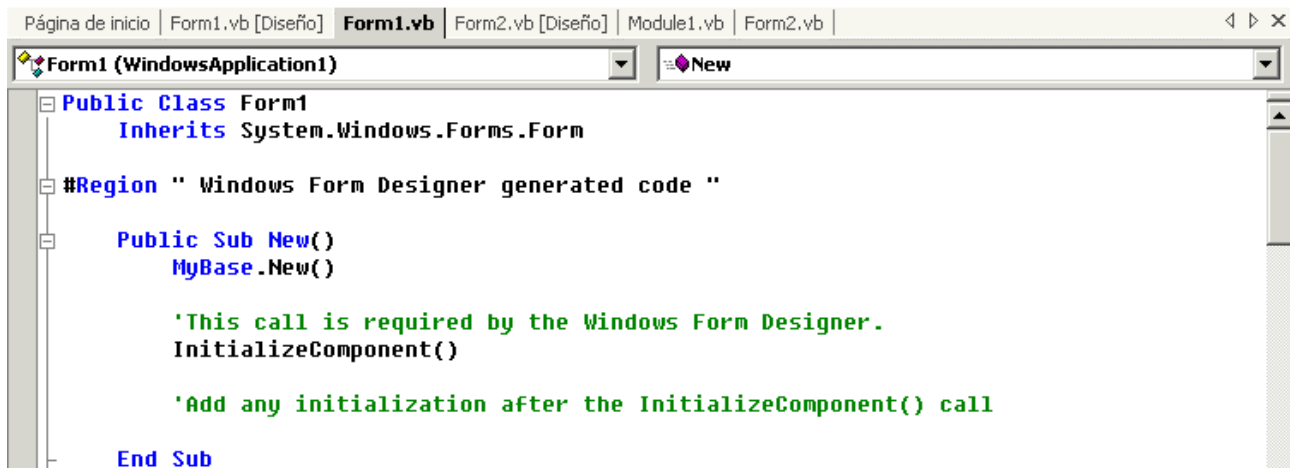


Figura 2.41 Editor de código

La barra de herramientas contiene todo lo necesario para poder compilar y modificar código, así como dar formato al diseño de nuestro programa.



Figura 2.42 Barra de Herramientas

El explorador de soluciones representa todos los formularios que se utilicen dentro del sistema, al igual que otros elementos como podemos verlo en la Figura 2.43.

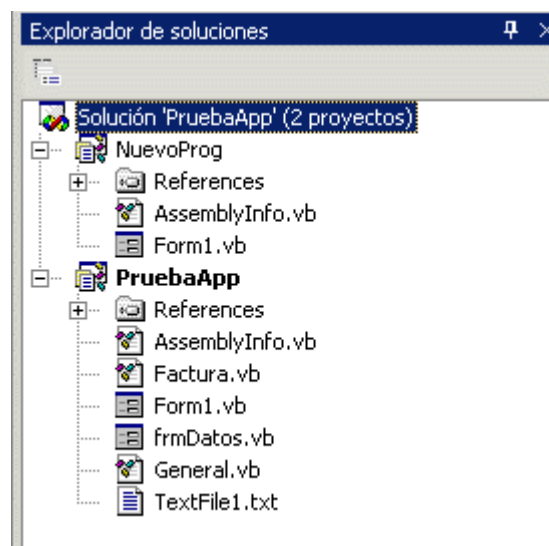


Figura 2.43 Explorador de soluciones

El cuadro de propiedades contiene todas las propiedades que podrán ser modificadas en tiempo de diseño del elemento que se elija, en la siguiente figura se muestran las propiedades del formulario llamado Form1.

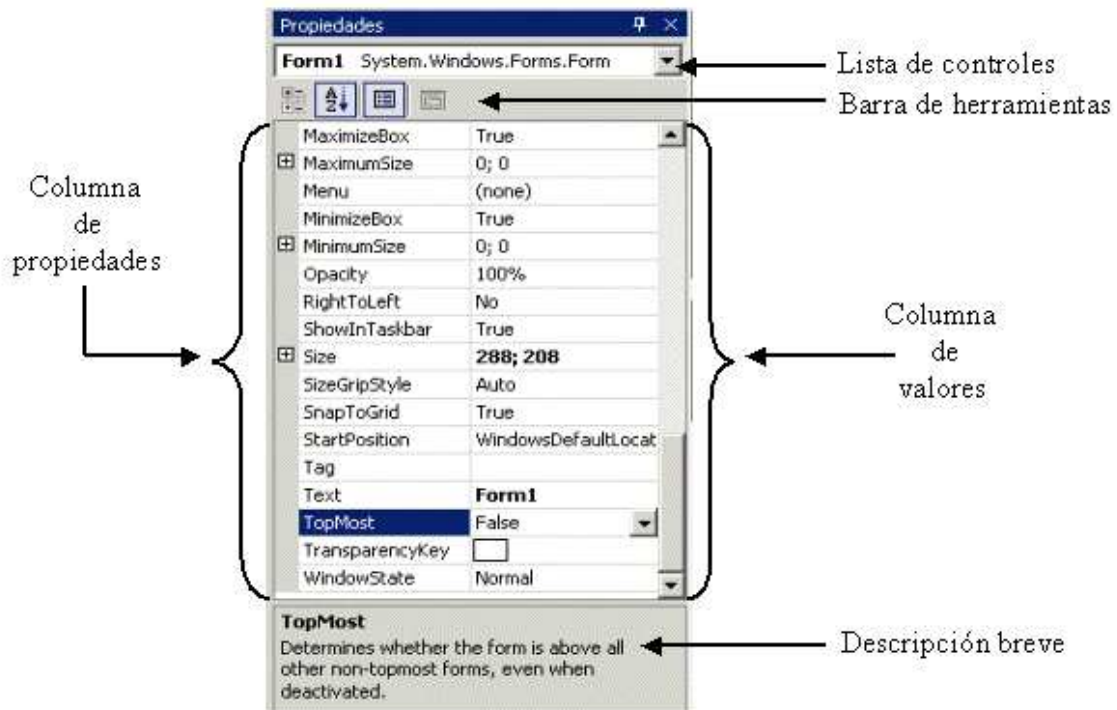


Figura 2.44 Cuadro de propiedades

Estos son algunos de los elementos del IDE de Visual Basic 2005 pero claro no son todos, cuenta con una amplia gama de opciones para poder desarrollar de una manera relativamente sencilla, rápida y profesional.

Programación Orientada a Objetos (OOP)

La programación orientada a objeto (OOP por sus siglas en inglés) es una de las nuevas ventajas de Visual Basic 2005, su antecesor (Visual Basic 6.0) no era completamente orientada a objetos si no que era orientada a eventos. La OOP se trata de una evolución de la programación procedural basada en funciones, que permite agrupar elementos de código (rutinas y datos) con funcionalidades similares, bajo un sistema unificado de manipulación y acceso a dichos elementos. En la programación estructurada procedural, basada en procedimientos y funciones, el crecimiento de una aplicación hace que el mantenimiento de la misma se convierta en una tarea difícil, debido al gran número de procedimientos interrelacionados que podemos llegar a tener. El mero hecho de efectuar una pequeña modificación en un proceso, nos puede suponer el tener que recorrer un gran número de funciones del programa, no relacionadas por un nexo común, pero gracias a la OOP se facilita este tipo de problemas.

La OOP proporciona las siguientes ventajas sobre otros como la de Visual Basic 6.0

- Uniformidad. Ya que la representación de los objetos lleva implícito tanto el análisis como el diseño y la codificación de los mismos.

- **Comprensión.** Tanto los datos que componen los objetos, como los procedimientos que los manipulan, están agrupados en clases, que se corresponden con las estructuras de información que el programa trata.
- **Flexibilidad.** Al tener relacionados los procedimientos que manipulan los datos con los datos a tratar, cualquier cambio que se realice sobre ellos quedará reflejado automáticamente en cualquier lugar donde estos datos aparezcan.
- **Estabilidad.** Dado que permite un tratamiento diferenciado de aquellos objetos que permanecen constantes en el tiempo sobre aquellos que cambian con frecuencia permite aislar las partes del programa que permanecen inalterables en el tiempo.
- **Reusabilidad.** La noción de objeto permite que programas que traten las mismas estructuras de información reutilicen las definiciones de objetos empleadas en otros programas e incluso los procedimientos que los manipulan. De esta forma, el desarrollo de un programa puede llegar a ser una simple combinación de objetos ya definidos donde estos están relacionados de una manera particular.
- Uno de los puntos clave a remarcar es que la programación orientada a objetos no sustituye a ninguna metodología ni lenguaje de programación anterior. Todos los programas que se realizan se pueden realizar igualmente mediante programación estructurada. Su uso en la actualidad se justifica porque el desarrollo de todas las nuevas herramientas basadas en un interface de usuario gráfico como Windows. Es mucho más sencillo

Objetos

Un objeto es una agrupación de código, compuesta de propiedades y métodos, que pueden ser manipulados como una entidad independiente. Las propiedades definen los datos o información del objeto, permitiendo consultar o modificar su estado; mientras que los métodos son las rutinas que definen su comportamiento.

Un objeto es una pieza que se ocupa de desempeñar un trabajo concreto dentro de una estructura organizativa de nivel superior, formada por múltiples objetos, cada uno de los cuales ejerce la tarea particular para la que ha sido diseñado.

Clases

Una clase no es otra cosa que el conjunto de especificaciones o normas que definen cómo va a ser creado un objeto de un tipo determinado; algo parecido a un manual de instrucciones conteniendo las indicaciones para crear el objeto o un molde de galletas, siendo los objetos las galletas que se obtienen de este molde.

Los términos objeto y clase son utilizados en OOP con gran profusión y en contextos muy similares, por lo que para intentar aclarar en lo posible ambos conceptos, diremos que una clase constituye la representación abstracta de algo, mientras que un objeto constituye la representación concreta de lo que una clase define.

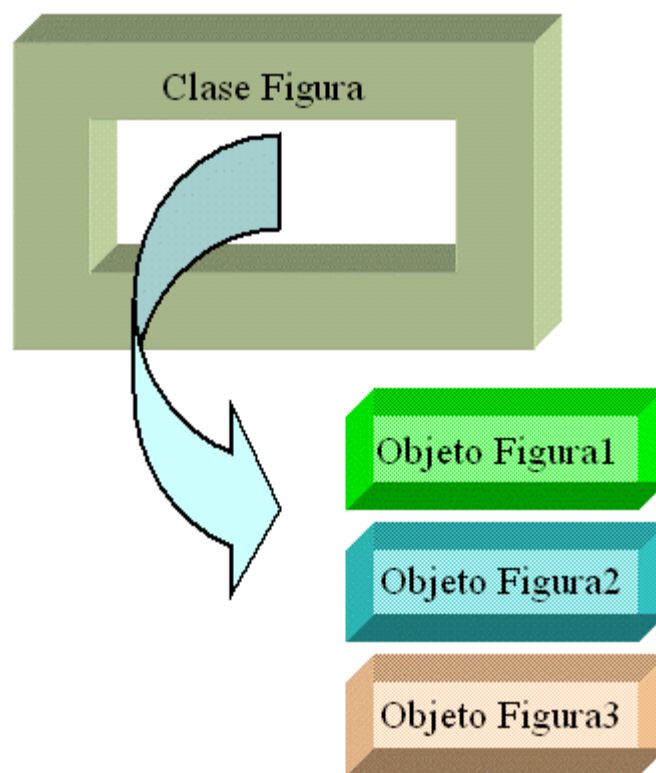


Figura 2.45 Instancia de un objeto a partir de una clase

Métodos

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

Estos representan las acciones que pueden ser realizadas por o sobre un objeto. Cada método tiene un nombre y un cuerpo que realiza la acción o comportamiento asociado con el nombre del método. El cuerpo del método consta de un bloque de código que ejecuta la acción requerida.

Abstracción

La abstracción es aquella característica que nos permite identificar un objeto a través de sus aspectos conceptuales. Del mismo modo que hacemos al identificar objetos reales, la abstracción nos ayuda a la hora de desarrollar una aplicación, permitiéndonos identificar los objetos que van a formar parte de nuestro programa, sin necesidad de disponer aún de su implementación; nos basta con reconocer los aspectos conceptuales que cada objeto debe resolver.

Por ejemplo, cuando abordamos el desarrollo de un programa de gestión orientado a objetos, realizamos una abstracción de los objetos que necesitaríamos para resolver los procesos del programa: en este caso tenemos el objeto Alumno, para gestionar a los usuarios de las computadoras de las salas de UNICA; un objeto Computadora, para gestionar las computadoras que contienen las salas; un objeto sala, para representar las salas de computo.

Encapsulación

La encapsulación establece la separación entre el interfaz del objeto y su implementación, aportándonos dos ventajas fundamentales.

Por una parte proporciona seguridad al código de la clase, evitando accesos y modificaciones no deseadas; una clase bien encapsulada no debe permitir la modificación directa de una variable, ni ejecutar métodos que sean de uso interno para la clase. Por otro lado la encapsulación simplifica la utilización de los objetos, ya que un programador que use un objeto, si este está bien diseñado y su código correctamente escrito, no necesitará conocer los detalles de su implementación, se limitará a utilizarlo.

Polimorfismo

El polimorfismo determina que el mismo nombre de método, realizará diferentes acciones según el objeto sobre el que sea aplicado. Al igual que sucedía en la encapsulación, el programador que haga uso del objeto, no necesita conocer los detalles de implementación de los métodos, se limita a utilizarlos.

Herencia

Se trata de la característica más importante de la OOP, y establece que partiendo de una clase a la que denominamos clase base, padre o superclase, creamos una nueva clase denominada clase derivada, hija, o subclase.

En esta clase derivada dispondremos de todo el código de la clase base, más el nuevo código propio de la clase hija, que escribamos para extender sus funcionalidades.

A su vez podemos tomar una clase derivada, creando una nueva subclase a partir de ella, y así sucesivamente, componiendo lo que se denomina una jerarquía de clases, que explicaremos seguidamente.

Existen dos tipos de herencia: simple y múltiple. La herencia simple es aquella en la que creamos una clase derivada a partir de una sola clase base, mientras que la herencia múltiple nos permite crear una clase derivada a partir de varias clases base. El entorno de .NET Framework 2.0 sólo permite utilizar herencia simple

2.4.2 ADO .Net³

ADO .NET es la nueva versión del modelo de objetos ADO (ActiveX Data Objects), es decir, la estrategia que ofrece Microsoft para el acceso a datos. ADO .NET ha sido ampliado para cubrir todas las necesidades que ADO no ofrecía, y está diseñado para trabajar con conjuntos de datos desconectados, lo que permite reducir el tráfico de red y con esto podremos evitar los problemas que se señalaron anteriormente en cuanto a la conexión con el servidor. ADO .NET utiliza XML como formato universal de transmisión de los datos. ADO .NET posee una serie de objetos que son los mismos que aparecen en la versión anterior de ADO, como pueden ser el objeto Connection o Command, e introduce nuevos objetos tales como el objeto DataReader, DataSet o DataView.

Comparando ADO (Visual Basic 6.0) con ADO .Net (Visual Basic 2005)

La siguiente tabla nos muestra una comparación entre las dos versiones para acceder a los datos, viendo los beneficios de la nueva tecnología en todos los aspectos.

Clases de objetos más estructuradas	El modelo de objetos de ADO .NET es mucho más rico que el de ADO. Incorpora nuevas clases de datos y encapsula mucha más potencia de uso a la par que corrige pequeños defectos de las versiones anteriores.
--	--

³Programación en Visual Basic .Net, Luis Miguel Blanco, Grupos EIDOS.

Independiente del lenguaje	Aprovechando la nueva arquitectura de servicios de la plataforma .NET, ADO .NET puede ser usado como un servicio del sistema, pudiendo ser utilizado por cualquier aplicación escrita en cualquier lenguaje.
Representaciones en memoria de los datos	En ADO .NET se emplea el DataSet, mientras que en ADO se emplea el Recordset. No es sólo un cambio de nombre. En memoria y en rendimiento las cosas han cambiado mucho.
Número de tablas	Un Recordset de ADO sólo puede contener un único resultado (fruto de consultas complejas o no). En cambio, un DataSet puede representar un espacio de múltiples tablas simultáneamente. Esto permite obtener una representación de espejo de la Base de datos a la que representa.
Navegación mejorada	En ADO sólo nos podemos mover secuencialmente fila a fila. En ADO .NET podremos navegar por filas en un DataSet, y consecuentemente avanzar en una fila/conjunto de filas de otro DataSet asociado a través de una colección de relaciones.
Acceso a datos Off-line (Desconectados)	En ADO .NET todos los accesos a datos se realizan en contextos desconectados de la base de datos. En ADO, es posible tener estructuras desconectadas, pero es una elección a tomar. Por defecto, ADO está pensado para contextos con conexión. Además, ADO .NET realiza todos los accesos a los servicios de datos a través del objeto DataSetCommand, lo que permite personalizar cada comando en función del rendimiento y la necesidad del programa.

<p>Compartimiento de datos entre capas, o bien, entre componentes</p>	<p>El traspaso de un Recordset Off-line en ADO es más complejo y pesado pues es necesario establecer un vínculo RPC y un proceso de COM marshalling (verificación de tipos de datos remotos). En ADO .NET para la comunicación entre componentes y capas se emplea un simple stream XML.</p>
<p>Tipos de datos más ricos</p>	<p>Debido al COM Marshalling los tipos de datos que se pueden usar vía RPC están muy limitados. En ADO .NET, gracias a los flujos XML, es posible enviar cualquier tipo de dato de manera sencilla.</p>
<p>Rendimiento</p>	<p>Tanto ADO como ADO .NET requieren de muchos recursos cuando se habla de envíos masivos de datos. El stress del sistema es proporcional al número de filas que se quieren procesar. Pero ADO .NET permite utilizar mucho más eficientemente los recursos, pues no tiene que realizar el COM Marshalling de datos, en lo que ADO sí (y eso supone un canal más de envío de datos de control que ADO .NET ahorra).</p>
<p>Penetración de firewalls</p>	<p>Por motivos de seguridad, las empresas suelen bloquear los sistemas de comunicaciones vía RPC. O bien, implementan pesados mecanismos de control de acceso y envío de la información. Lo que complica los diseños, sus rendimientos y su instalación y distribución. En ADO .NET, puesto que no se realizan llamadas al sistema, sino que se envían flujos de datos en XML, se simplifica enormemente la configuración de estos dispositivos.</p>

Tabla 2.1

Beneficios de ADO .Net

- **Interoperabilidad:** Las aplicaciones basadas en ADO .NET obtienen ventaja de la flexibilidad y la masiva aceptación del estándar XML para el intercambio de datos.
- **Mantenimiento:** Gracias a los DataSet, la tarea de portar y aumentar los procesos de datos y de negocio será más sencillo.
- **Programación:** Se puede acceder a un API de programación estructurado, de fuerte tipificado y que además se concentra en la correcta forma de presentar los datos.
- **Rendimiento:** Puesto que trabajamos con objetos de datos desconectados, todo el proceso se acelera, ya que no tenemos que estar comunicándonos por Marshalling con el servidor.
- **Escalabilidad:** Las aplicaciones Web tienen un número ilimitado de conexiones potenciales debido a la naturaleza de Internet.
- **Arquitectura de datos desconectado:** Con ADO .NET se consigue estar conectado al servidor sólo lo estrictamente necesario para realizar la operación de carga de los datos en el DataSet.

2.4.2.1 Clases de ADO .Net

ADO .Net por ser orientada a objetos tienen dentro de su espacio de nombres muchas clases a las cuales se puede recurrir, en este punto se dará una breve descripción de cuál es la función de cada una de ellas, sin llegar a profundizar en los métodos y propiedades que estas contienen.

DataSet

El API de ADO .NET proporciona una superclase, DataSet, que encapsula lo que sería la base de datos a un nivel lógico: tablas, vistas, relaciones, integridad entre todos ellos y más., pero siempre con independencia del tipo de fabricante que la diseñó. Aquí se tiene el mejor concepto de datos desconectados: una copia en el cliente de la arquitectura de la base de datos, basada en un esquema XML que la independiza del fabricante, proporcionando al desarrollador la libertad de trabajo independiente de la plataforma. La Figura 2.46 muestra una representación de este tipo de objeto.

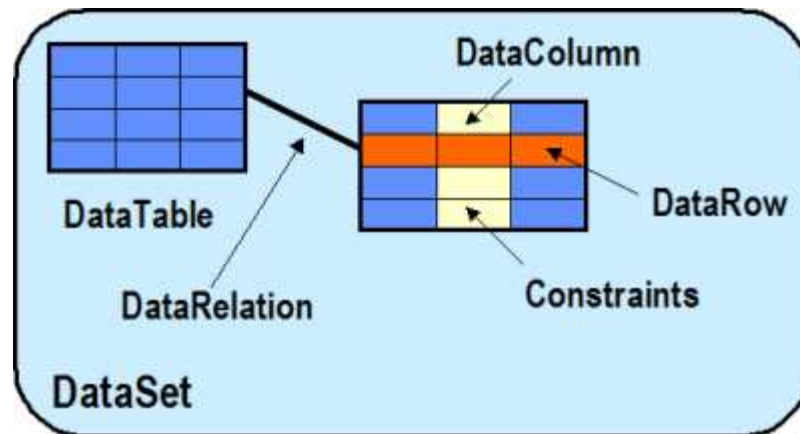


Figura 2.46

Clase Connection

El primer paso obligado en un acceso a datos consiste en establecer una conexión con un almacén de datos. Esto lo vamos a conseguir gracias a las clases Connection de ADO .NET, que nos permitirán conectarnos a un origen de datos (ya sea una base de datos o no), al igual que en ADO clásico empleábamos el objeto Connection.

Existen dos implementaciones para algunos de los objetos de ADO .NET, cada uno específico del origen de datos con el que nos vamos a conectar. Esto ocurre con el objeto Connection, que tiene dos versiones, una como proveedor de datos de SQL Server, a través de la clase System.Data.SqlClient.SqlConnection, y otra como proveedor de datos OLEDB, a través de la clase System.Data.OleDb.OleDbConnection.

Clase Command

Establecida una conexión con un almacén de datos, la siguiente operación lógica consiste en enviarle sentencias para realizar los distintos tipos de operaciones que habitualmente realizamos con los datos. Las clases Command de ADO .NET serán las usaremos para realizar tales operaciones. SqlCommand y OleDbCommand, son muy similares al objeto Command existente en ADO. El objeto Command nos va a permitir ejecutar una sentencia SQL o un procedimiento almacenado sobre la fuente de datos a la que estamos accediendo.

A través de un objeto Command también podremos obtener un conjunto de resultados del almacén de datos. En este caso, los resultados se pasarán a otros objetos de ADO .NET, como DataReader o DataAdapter; estos dos objetos los comentaremos más adelante. Un objeto Command lo vamos a crear a partir de una conexión ya existente, y va a contener una sentencia SQL para ejecutar sobre la conexión establecida con el origen de datos.

Clase *DataReader*

Un objeto *DataReader* permite la navegación hacia delante y de sólo lectura, de los registros devueltos por una consulta. Es lo más parecido al objeto *Recordset* de ADO de tipo *read only/forward only*. A diferencia del resto de objetos, que siguen un esquema desconectado de manipulación de datos, los *DataReader* permanecen conectados durante todo el tiempo que realizan el recorrido por los registros que contienen. Las clases que implementan este tipo de objeto son *SqlDataReader* y *OleDbDataReader*.

Para obtener un *DataReader*, ejecutaremos el método *ExecuteReader()* de un objeto *Command* basado en una consulta SQL o procedimiento almacenado.

En la siguiente figura se hace comparación con las clases de ADO.

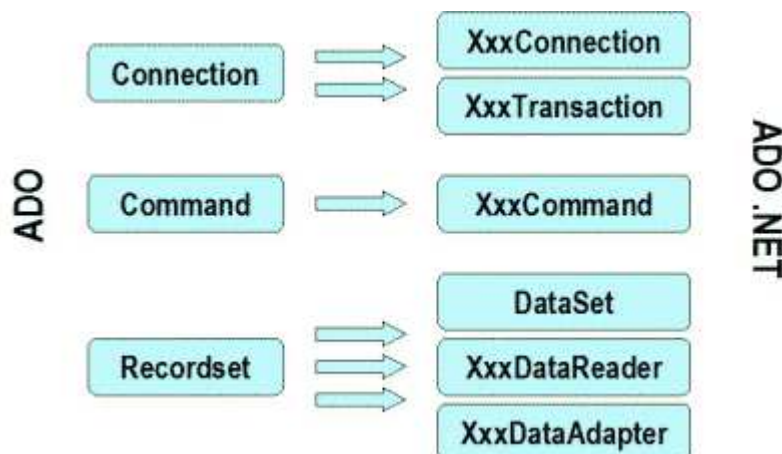


Figura 2.47 Comparación de clases de ADO y ADO .Net

Las clases anteriormente mencionadas son la base para realizar una conexión a una base de datos tanto local como remota, pero en el caso de SCOSU v2, fue necesario utilizar otro proveedor (proveedor) debido a que la conexión fue a PostgreSQL y VB .Net no lo tenía. En este caso se utilizó *Npgsql* el cual contiene sus clases para la conexión, pero lo único que cambia con respecto a las que ya se tiene en VB .Net es en el nombre, así la clase *Connection* es *NpgsqlConnection* o la clase *Command* sería *NpgsqlCommand*, con respecto a las funciones que realizan y como la realizan no cambian en nada.

CAPÍTULO 3

DESARROLLO

3.1 Implementación de la Base de datos

El proceso de implementación de una base de datos es complejo y se requiere de tiempo, afortunadamente existe las herramientas *CASE* (Computer Aided Software Engineering por sus siglas en ingles o Ingeniería de Software Asistida por Computadora), la cual definiremos a continuación.

3.1.1 Herramienta CASE

Una herramienta CASE son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero.

Estas herramientas nos ayudan en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, calculo de costos, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.¹

¹ <http://es.wikipedia.org/wiki/CASE>

Para el caso de SCOSU v2 se utilizó la herramienta CASE llamada Erwin.

Erwin es una herramienta de base de datos que ayuda a generar y mantener aplicaciones de base de datos de calidad y alto rendimiento. Se pueden observar las tablas de forma física al igual que las relaciones que se tienen entre las diversas entidades, así mismo, podemos ver los atributos y que roles cumplen cada uno de ellos.

Con Erwin se puede definir las bases de datos desde dos puntos de vista, el lógico y el físico, el primero es donde podemos definir los requerimientos de información y reglas del negocio que se definen en la base de datos de SCOSU v2, el segundo está orientado a optimizar las características específicas de la base de datos de destino, también permite visualizar la estructura adecuada y los elementos clave.

Los beneficios que nos ofrece son:²

- Mejora la productividad entre los desarrolladores cuando los diseños de la base de datos son divididos, compartidos, y reutilizados.
- El ambiente gráfico facilita la visualización de la estructura completa, los elementos claves y el diseño optimizado de la base de datos.
- Le ahorra tiempo al acelerar la creación de bases de datos de alta calidad y transaccionales de alto rendimiento.
- Mantiene los recursos y mejora la precisión al sincronizar el modelo y la base de datos.

Otra de las tareas que realiza es la generación de sentencias de SQL, con las cuales se crean todas las entidades que se diseñaron en el diagrama entidad relación de la base de datos, así como sus respectivos atributos y el tipo de dato, nos crea las llaves primarias, llaves primarias compuestas o foráneas que se definieron en el diagrama, además de crear las respectivas relaciones entre las entidades.

El manejador con el que se cuenta en UNICA es PostgreSQL, es por eso que se necesita adaptarlo a la sintaxis del lenguaje de ese manejador, el cual es una pequeña inconveniencia para todos los beneficios que nos brinda.

Como se puede ver, Erwin nos facilitaría en gran parte la tarea de implementar la base de datos, reduciendo el tiempo de su realización al mínimo.

² AllFusio Erwin Data Modeler; Marco A. Guevara Injoque, César R. Flores Nazario.

3.1.2 Creación de la Base de datos

Ya había mencionado que UNICA cuenta con un servidor que contiene el manejador de base de datos PostgreSQL, llamado Tork, es aquí donde estará albergada la base de SCOSU v2. Gracias al script generado por la herramienta CASE Erwin la creación fue rápida y de una manera optima.

Tablas

En la Figura 3.1 Se muestran las tablas creadas, además de sus respectivas llaves primarias y foráneas, el tipo de dato y más. Figura 3.2.

```
BDSCOSU=> \dt
List of relations
Schema | Name | Type | Owner
-----+-----+-----+-----
public | alumno | table | aion
public | alumno_modulo | table | aion
public | alumno_sancionado | table | aion
public | cancelar_sancion | table | aion
public | computadora | table | aion
public | computadora_so | table | aion
public | curso | table | aion
public | jefe_sala | table | aion
public | modulo | table | aion
public | registro_historico | table | aion
public | reporte | table | aion
public | resp_sancion | table | aion
public | sala | table | aion
public | sancion | table | aion
public | software | table | aion
public | tipo_estado | table | aion
public | tipo_so | table | aion
(17 rows)
```

Figura 3.1 Tablas de la base de datos de SCOSU v2

```
BDSCOSU=> \d alumno;
Table "public.alumno"
Column | Type | Modifiers
-----+-----+-----
no_cta | character varying(10) | not null
nombre | character varying(20) | not null
apellido_paterno | character varying(20) | not null
apellido_materno | character varying(20) | not null
carrera | character varying(30) | not null
fecha | date | not null
hora_entrada | time without time zone | not null
hora_salida | time without time zone | not null
no_computadora | integer |
num_sala | integer |
aula | character(1) |
Indexes:
"alumno_pk" PRIMARY KEY, btree (no_cta, fecha, hora_entrada, hora_salida)
"xpkalumno" UNIQUE, btree (no_cta, fecha, hora_entrada, hora_salida)
Foreign-key constraints:
"computadora_fk" FOREIGN KEY (no_computadora, num_sala, aula) REFERENCES computadora(no_computadora, num_sala, aula) ON DELETE RESTRICT
"reg_historico_fk" FOREIGN KEY (fecha, hora_entrada, hora_salida) REFERENCES registro_historico(fecha, hora_entrada, hora_salida) ON UPDATE CASCADE ON DELETE RESTRICT
```

Figura 3.2 Ejemplo de una tabla de la base de datos de SCOSU v2

Vistas

Una vista es el resultado de una consulta SQL de una o varias tablas, tienen la misma estructura que una tabla: filas y columnas. La única diferencia es que sólo se almacena de ellas la definición, no los datos. Los datos que se recuperan mediante una consulta a una vista se presentarán igual que los de una tabla.

Fue necesario crear las vistas para poder simplificar los *Query's* (Consulta, inserción, actualización, búsqueda o eliminación en una base de datos) dentro del sistema SCOSU v2, a continuación se muestran las vistas creadas.

```
BDSCOSU=> \dv
          List of relations
 Schema |      Name      | Type | Owner
-----+-----+-----+-----
 public | restaurar      | view | aion
 public | sancionados    | view | aion
(2 rows)
```

Figura 3.3 Vistas de la base de datos de SCOSU v2

Secuencias (Sequence)

Sequence introducirá una nueva secuencia generadora de números dentro de la base de datos. Se crearon las secuencias para reducir el trabajo del sistema al momento de introducir datos a las tablas cancelar_sacion, curso y resp_sancion, la Figura 3.4 Muestra las secuencias generadas para cada una de las tablas.

```
BDSCOSU=> \ds
          List of relations
 Schema |      Name      | Type   | Owner
-----+-----+-----+-----
 public | id_cancelar_seq | sequence | aion
 public | id_curso_seq    | sequence | aion
 public | id_resp_seq     | sequence | aion
(3 rows)
```

Figura 3.4 Secuencias de la base datos de SCOSU v2

3.2 Programación de la aplicación en Visual Basic 2005

Como parte de la nueva versión, SCOSU v2 se dividió en dos módulos, SCOSU v2 y SCOSU v2 Modulo del Jefe, esto, con el fin de solucionar ciertos inconvenientes que se tenían en la antigua versión, además de añadir otras funciones que ayudaran a mejorar el desempeño de los dos módulos.

3.2.1 SCOSU v2



Figura 3.5 Logotipo de SCOSU v2

SCOSU v2 es la nueva versión de SCOSU, realiza funciones como, asignar equipos a alumnos que se encuentren inscritos en la Facultad de Ingeniería, colocar computadoras como descompuestas, sancionar alumnos que incurran en alguna falta dentro de las normas que establece UNICA y más. Este modulo será instalado en los equipos de control de salas.

Uno de los puntos más importantes a resolver era la conexión a los servidores y que con ADO .Net se soluciono ese problema. Tomando en cuenta que se realizan cientos de consultas en todas las salas, era necesario realizarlas de manera eficiente para que no existieran conflictos, ADO .Net nos brinda suficientes herramientas para poder realizarlas de una manera óptima, teniendo así un sistema funcionando al cien por ciento.

Por otra parte SCOSU v2 tiene funciones adicionales que lo hacen un sistema mucho más eficiente, por ejemplo, el “modo desconectado”; podríamos decir que esta función es una de la mas importantes que tiene. En la versión anterior (SCOSU) tenía el inconveniente, de que en dado caso de que se perdiera la conexión con el servidor, el sistema sufría un error y se cerraba, esto ocasionaba que se perdiera tiempo en la atención al alumno, teniendo como consecuencias largas filas de espera y pérdida de registros.

Con el “modo desconectado” podemos seguir trabajando en caso de que se perdiera la conexión con el servidor, evitando la demora en el servicio y la pérdida de registros, los cuales se guardarían internamente en el sistema. Lo anterior es posible gracias a una de las características mencionada en el capítulo 2 (Véase, Comparando ADO (Visual Basic 6.0) con ADO .Net (Visual Basic 2005)). Los DataSet son los encargados de que podamos trabajar en un contexto desconectado, prácticamente podemos realizar una copia del esquema de nuestra base de datos y guardarla dentro de ellos, teniendo así nuestra base de datos de modo local. Adicionalmente, una vez que se recupere la conexión al servidor, nosotros podemos actualizar la base de datos original con los datos que se registraron dentro de los DataSet, y de esta manera no perder ningún registro.

Como parte de los requerimientos que me fueron solicitados añadí tres estados más a las computadoras, incluyo el mantenimiento, reparación e impresora, esto con el fin de dar más detalles de cómo es que se encuentran los equipos en la salas y que no exista confusión al momento de monitorearlas. En estado de descompuesta se coloca el tipo de mal que hizo que dejara de funcionar, en mantenimiento y reparación se registra en qué fecha más o menos estarán listas y funcionando, por último, el estado de impresora se coloca cuando es necesario utilizar un equipo que se encuentra dentro de la salas.

Si en algún momento se llegara a presentar algún inconveniente en el sistema o en la conexión con los servidores, los usuarios de la diferentes salas podrán comunicarse con el administrador mediante correo electrónico, que viene incorporado como una de las funciones que puede realizar SCOSU v2. Estas son parte de todas las funciones que se le incorporaron a la nueva versión.³

Refiriéndonos a la programación, como se ha mencionado Visual Studio 2005 se encuentra enfocado al paradigma de la programación orientada a objetos, y como pieza fundamental de este paradigma tenemos a los objetos, los cuales, “nacen” de las clases. Estas clases se obtuvieron mediante el modelado del sistema siguiendo las reglas que establece UML (Véase, capítulo 2, Diagrama de clases). Cada clase se utilizo para representar la estructura interna del sistema, y la relación que existe entre ellas.

³Para ver todas las funciones consultar el Manual del usuario.

Cada uno de los objetos que fueron creados cumplen con un rol dentro del sistema, facilitando de esta manera la interpretación de los datos y la programación, por ejemplo, de la clase Alumno creamos un objeto que tendrá como atributos, nombre, apellido paterno, apellido materno, número de cuenta y carrera, teniendo eso como datos es más sencillo ver con lo que estamos trabajando, además de que nos facilita el manejo de los registros de la base de datos.⁴

3.2.2 SCOSU v2 Modulo del Jefe



Figura 3.6 Logotipo SCOSU v2 Modulo del Jefe

SCOSU v2 Modulo del Jefe, una nueva incorporación creada para mantener el control de los registros, donde la decisión de los jefes de cada una de las salas de UNICA es primordial. Ayuda a tener un sistema más eficiente y actualizado, permite mantener el contenido de las salas con las últimas actualizaciones del diferente software que se tienen en cada una de las aulas. Este modulo por ser exclusivo para los jefes de las diferentes salas, estará instalado en sus respectivos equipos.

En ella se realizan acciones como agregar o borrar registro de la base de datos, cosa que en SCOSU v2 solo se pueden visualizar dichos registros, además de realizar reportes de los alumnos que ingresan a las salas. En seguida se enlistan las diferentes funciones que realiza.

⁴ Véase Manual del programador.

➤ *Agregar software nuevo*

Esta opción nos permite agregar el software que se tienen en cada una de las aulas de las diferentes salas de UNICA. En SCOSU para agregar el software nuevo a las diferentes aulas era necesario que el personal de las salas realizara su lista con todas las modificaciones que se hicieron y entregarla al administrador del sistema, éste a su vez, colocaba esta lista en el sistema pero todo a nivel código.

Lo anterior ocasionaba pequeños inconvenientes como el retraso de la actualización, por que el administrador debía programar toda la lista de software desde cero y si a esto le agregamos que se tenía que llevar la actualización a las 4 salas, nos da como resultado que no era muy práctico. Gracias a esta función cada sala podrá mantenerse actualizada día a día, sin necesidad de esperar una nueva actualización del sistema cada que se modifique la lista del software.

➤ *Actualizar software*

Nos permite actualizar el software que se tiene en las aulas sin necesidad que crear toda la lista desde cero. Cuando se utiliza Agregar software nuevo es porque será la primera vez que agregara software a las diferentes aulas de la sala, en el caso de Actualizar software se utilizara cuando ya se tenga software y solo se quiera realizar un cambio ya sea en el nombre de un programa, su versión o simplemente añadir uno o más software a la lista actual.

➤ *Cursos*

Al igual que SCOSU v2, esta opción muestra los cursos que están siendo impartidos en su respectiva sala, nos presenta datos cómo su nombre, fecha de inicio y fin, hora de inicio y fin, y la aula, pero en este caso se pueden eliminar los cursos, ya sea cuando éstos hayan terminado o cuando por alguna razón se cancelen o se terminen antes del tiempo indicado.

➤ *Sancionados*

Muestra los alumnos sancionados en las salas de UNICA con sus respectivas sanciones así como el día que fueron sancionados y la fecha en que podrán volver a utilizar las computadoras de las salas, entre otros datos. SCOSU v2 quita automáticamente las sanciones de los alumnos a los cuales se les venció su castigo, pero en SCOSU v2 modulo del Jefe, podrá quitar la sanción antes de que se termine, considerando que fue injusta la sanción o por cualquier otro motivo.

➤ *Jefe de sala*

Con esta opción es posible ver el nombre del jefe de la sala, además de que permite cambiarlo, así como su contraseña, la cual es necesaria para poder ingresar a SCOSU v2 Modulo del Jefe.

➤ *Reporte*

Como su nombre lo dice, nos crea reporte de los alumnos que ingresan a las diferentes salas de UNICA, obtiene el nombre del alumno, número de cuenta, carrera, datos de la sala y aula de donde estuvo, la computadora que utilizo, a qué hora, cuánto tiempo y el número de veces que ingreso, todo estos datos pueden ser por día o por semestre. Una vez obtenidos los informes podemos trabajar con ellos de diferente manera:

1. Podemos imprimirlos, cuenta con un formato prediseñado para poder dar una mayor sencillez al momento de leerlo, mostrando los datos anteriormente mencionados.
2. Se pueden manejar los diferentes valores que se obtienen, como el tiempo que utilizo las maquinas o el número de veces que ingreso, esto gracias a que se puede exportar el informe a Excel y así trabajarlos de una manera más sencilla.

3.3 Integración de la aplicación con los diferentes sistemas de UNICA

En UNICA existen varios sistemas que ayudan a automatizar tareas y mejorar el servicio para los usuarios en diferentes áreas. SCOSU v2 se encuentra ligado a varios de esos sistemas, los cuales en conjunto realizan una tarea mayor, además de existir otros que aunque no se han integrado aun, es uno de los objetivos del proyecto. Estos son los sistemas con los que interactúa SCOSU v2 y con el que se integrara.

3.3.1 Sistema de Inscripción de Acceso a las Salas de UNICA (SIASU)

Registra a los alumnos que se encuentre inscritos en la Facultad de Ingeniería sin importar la carrera en la que se encuentren. Se les pide comprobante de inscripción o credencia de la Facultad para poder registrarlos, con esto se garantiza el acceso a las salas únicamente a los alumnos de Ingeniería. Una vez registrado el alumno acepta todos los lineamientos que impone UNICA, podrá acceder a cualquiera de las cuatro salas y utilizar las computadoras e impresoras que ahí se encuentren.

SCOSU v2 tiene entre sus funciones la de inscribir al alumno a SIASU como parte de la integración con éste sistema.

3.3.2 Sistema de Control de Impresiones

Ayuda a administrar el número de impresiones que puede realizar el alumno en un día. El Sistema de Control de Impresiones se encuentra instalado en las computadoras donde están conectadas las impresoras, el alumno introduce su número de cuenta y así poder imprimir su documento. Tienen como máximo 15 hojas de impresión con una tolerancia de 2 más, en caso de que superé el límite establecido se hace acreedor a una sanción de una semana sin poder utilizar los equipos de cómputo y las impresoras de cualquier sala.

La forma en que interactúa con SCOSU v2 es para verificar el número de impresiones que el alumno tiene antes de que ingrese a la impresora, de esta manera evita que imprima más de las que le sobran. Muestra reportes de cuantas impresiones se realizaron en el día o por cierto periodo, además de mostrar el número de impresiones que se realizaron en una sala o en todas.

3.3.3 Sistema de Control de Asesorías

Registra cada una de las asesorías que los becarios dan a los alumnos de la Facultad de Ingeniería, este dato es necesario para las estadísticas que realiza UNICA además de observar el desempeño del becario en sus respectivas asesorías, se integrara registrándola directamente en SCOSU v2.

Esto es en un principio, en el futuro y viendo el desempeño de la integración del Sistema de Control de Asesorías y SCOSU v2, se podrán añadir otras funciones y tareas para hacerlo aun más eficiente.

Por último, uno de los beneficios más importantes es que se podrán llevar mejores registros y con esto, mejores estadísticas de todos los datos que se obtienen y así mismo poder mejorar día a día el servicio que se ofrece a los alumnos.

CAPÍTULO 4

PRUEBAS

Las pruebas del software son un elemento crítico para la garantía de calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación. La creciente percepción del software como un elemento del sistema y la importancia de los «costes» asociados a un fallo del propio sistema, están motivando la creación de pruebas minuciosas y bien planificadas. No es raro que una organización de desarrollo de software emplee entre el 30 y el 40 por ciento del esfuerzo total de un proyecto en las pruebas. En casos extremos, las pruebas del software para actividades críticas (por ejemplo, control de tráfico aéreo, control de reactores nucleares) pueden costar de tres a cinco veces más que el resto de los pasos de la ingeniería del software juntos.

También podemos decir que el objetivo de la fase de pruebas de un programa es el de detectar todo posible malfuncionamiento antes de que entre en producción. Un error detectado por el programador puede ser costoso de reparar; pero siempre es peor que el error le aparezca al usuario final.

En esta idea, los casos prueba será de mayor calidad cuantos menos errores queden por descubrir tras haberla pasado. Y, viceversa, si un programa aún tiene muchos fallos tras haber superado los casos de pruebas, diremos que esos casos son de poca calidad.

Si pudiéramos probar un programa con todos los posibles datos de entrada, tendríamos unos casos de prueba perfecta, pues no hay lugar para las sorpresas. Lamentablemente, casi nunca es posible probar con todos los casos. En consecuencia necesitamos un criterio para elegir qué casos probar.

4.1 Objetivos de las pruebas ¹

En un excelente libro sobre las pruebas del software, Glen Myers establece varias normas que pueden servir acertadamente como objetivos de las pruebas:

- La prueba es el proceso de ejecución de un programa con la intención de descubrir un error.
- Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.
- Una prueba tiene éxito si descubre un error no detectado hasta entonces.

Los objetivos anteriores suponen un cambio dramático de punto de vista. Nos quitan la idea que, normalmente, tenemos de que una prueba tiene éxito si no descubre errores.

Nuestro objetivo es diseñar pruebas que sistemáticamente saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y de esfuerzo. Si la prueba se lleva a cabo con éxito (de acuerdo con el objetivo anteriormente establecido), descubrirá errores en el software.

Como ventaja secundaria, la prueba demuestra hasta qué punto las funciones del software parecen funcionar de acuerdo con las especificaciones y parecen alcanzarse los requisitos de rendimiento.

Además, los datos que se van recogiendo a medida que se lleva a cabo la prueba proporcionan una buena indicación de la fiabilidad del software y, de alguna manera, indican la calidad del software como un todo. Pero, la prueba no puede asegurar la ausencia de defectos; sólo puede demostrar que existen defectos en el software.

Por otra parte, el objetivo de la etapa de pruebas es garantizar la calidad del producto desarrollado. Esta etapa implica:

- Verificar la interacción de componentes.
- Verificar la integración adecuada de los componentes.

¹ Ingeniería del Software un enfoque practico; Roger S. Pressman; Quinta edición; Mc Graw Hill.

- Verificar que todos los requisitos se han implementado correctamente.
- Identificar y asegurar que los defectos encontrados se han corregido antes de entregar el software al cliente.
- Diseñar pruebas que sistemáticamente saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y esfuerzo.

La prueba no es una actividad sencilla, no es una etapa del proyecto en la cual se asegura la calidad, sino que la prueba debe ocurrir durante todo el ciclo de vida: podemos probar la funcionalidad de los primeros prototipos; probar la estabilidad, cobertura y rendimiento de la arquitectura; probar el producto final y más. Lo que conduce al principal beneficio de la prueba: proporcionar una retroalimentación mientras hay todavía tiempo y recursos para hacer algo.

La prueba no puede asegurar la ausencia de defectos; sólo puede demostrar que existen defectos en el software.

4.2 Diseño de casos de pruebas

Para poder garantizar el correcto funcionamiento del sistema es necesario saber la función específica para la que fue diseñado el producto, se pueden llevar a cabo pruebas que demuestren que cada función es completamente operativa y, al mismo, tiempo buscando errores en cada función; además de conocer el funcionamiento del producto, se pueden desarrollar pruebas que aseguren que todas las piezas encajan, o sea, que la operación interna se ajusta a las especificaciones y que todos los componentes internos se han comprobado de forma adecuada. El primer enfoque de prueba se denomina prueba de caja negra y el segundo, prueba de caja blanca.

Cuando se considera al software, la *prueba de caja negra* se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. O sea, los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa (por ejemplo, archivos de datos) se mantiene.

Una prueba de caja negra examina algunos aspectos del modelo fundamental del sistema sin tener mucho en cuenta la estructura lógica interna del software.

La *prueba de caja blanca* del software se basa en el minucioso examen de los detalles procedimentales. Se comprueban los caminos lógicos del software proponiendo casos de prueba que ejerciten conjuntos específicos de condiciones y/o bucles.

Se puede examinar el «estado del programa» en varios puntos para determinar si el estado real coincide con el esperado o mencionado. En seguida se tratarán más a fondo las pruebas anteriormente mencionadas con sus respectivas características y técnicas, además de analizar las pruebas que se realizaron a SCOSU v2.

4.3 Pruebas de caja blanca

La prueba de caja blanca, denominada a veces *prueba de caja de cristal* es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba. Mediante los métodos de prueba de caja blanca, el ingeniero del software puede obtener casos de prueba que:

- Garanticen que se ejercita por lo menos una vez todos los caminos independientes de cada módulo.
- Ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa.
- Ejecuten todos los bucles en sus límites y con sus límites operacionales.
- Ejerciten las estructuras internas de datos para asegurar su validez.

Con las pruebas de caja blanca lo que buscamos es encontrar fragmentos del programa que no son ejecutados por los casos de pruebas. Si encontramos que el resultado de estas pruebas es menor al 100%, debemos ejecutar otros casos para intentar llegar al 100%. Si aun así no lo conseguimos, deberíamos preguntarnos si sirve de algo esa parte del código.

Por otra parte la prueba de caja blanca incluye varias técnicas que se pueden aplicar al software desarrollado, las cuales como son:

Prueba del camino básico

La *prueba del camino básico* es una técnica de prueba de caja blanca propuesta inicialmente por Tom McCabe. El método del camino básico permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un *conjunto básico* de caminos de ejecución. Los casos de prueba obtenidos del conjunto básico garantizan que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa. Esta prueba basa su funcionalidad en la notación de grafo de flujo, el cual representa el flujo de control lógico en notación ilustrada.

Prueba de la estructura de control

La prueba del camino básico es una de las técnicas que se utilizan en las pruebas de estructura de control, aunque la de camino básico es sencilla y efectiva, siempre será necesario complementarla con otras variantes de la prueba de la estructura de control, dándole una mejor calidad a la prueba así como ampliar el alcance de la misma.

En seguida se especifican las variantes anteriormente mencionadas.

- **Prueba de condición:** Es un método de diseño de casos de prueba que ejercita las condiciones lógicas contenidas en el módulo de un programa. Una condición simple es una variable lógica o una expresión relacional, posiblemente precedida con un operador NOT.

Se centra en la prueba de cada una de las condiciones del programa. Las estrategias de prueba de condiciones tienen, generalmente, dos ventajas. La primera, que la cobertura de la prueba de una condición es sencilla. La segunda, que la cobertura de la prueba de las condiciones de un programa da una orientación para generar pruebas adicionales del programa.

El propósito de la prueba de condiciones es detectar, no sólo los errores en las condiciones de un programa, sino también otros errores que se puedan generar. Por otra parte la prueba de condición se encuentra constituida por dos estrategias, la *prueba de ramificaciones* y la *prueba del dominio*.

- *Prueba de ramificaciones:* Es posiblemente la estrategia de prueba de condiciones más sencilla. Para una condición compuesta "C", es necesario ejecutar al menos una vez las ramas verdadera y falsa de "C" y cada condición simple de "C".
- *Prueba del dominio:* Requiere la realización de tres o cuatro pruebas para una expresión relacional. Para una expresión relacional de la forma:

$$E_1 <\text{operador-relacional}> E_2$$

Se requieren tres pruebas, para comprobar que el valor de E, es mayor, igual o menor que el valor de E₂, respectivamente. Si el <operador-relacional> es incorrecto y E₁ y E₂ son correctos, entonces estas tres pruebas garantizan la detección de un error del operador relacional.

- **Prueba del flujo de datos:** Selecciona caminos de prueba de un programa de acuerdo con la ubicación de las definiciones y los usos de las variables del programa. Antes de definir la estrategia veamos los siguientes conceptos.

La definición de la variable X en la instrucción S se dice que está viva en la instrucción S' si existe un camino desde la instrucción S a la instrucción S' que no contiene otra definición de X .

Una *cadena de definición-uso* (o cadena DU) de la variable X es de la forma $[X, S, S']$, donde S y S' son números de instrucciones, X está en $DEF(S)$ y en $USO(S')$ y la definición de X en la instrucción S está viva en la instrucción S' .

Una sencilla estrategia de prueba de flujo de datos se basa en requerir que se cubra al menos una vez cada cadena DU. Esta estrategia se conoce como estrategia de prueba DU. Se ha demostrado que la prueba DU no garantiza que se cubran todas las ramificaciones de un programa.

Sin embargo, solamente no se garantiza el cubrimiento de una ramificación en situaciones raras como las instrucciones if-then-else en la que la parte then no tiene ninguna definición de variable y no existe la parte else. En esta situación, la prueba DU no cubre necesariamente la rama else de la sentencia superior.

Las estrategias de prueba de flujo de datos son útiles para seleccionar caminos de prueba de un programa que contenga sentencias if o de bucles anidados.

- **Prueba de bucles:** es una técnica de prueba de caja blanca que se centra exclusivamente en la validez de las construcciones de bucles. Se pueden definir cuatro clases diferentes de bucles: bucles simples, bucles concatenados, bucles anidados y bucles no estructurados.
- *Bucles simples:* A los bucles simples se les debe aplicar el siguiente conjunto de pruebas, donde n es el número máximo de pasos permitidos por el bucle:
 1. pasar por alto totalmente el bucle
 2. pasar una sola vez por el bucle
 3. pasar dos veces por el bucle
 4. hacer m pasos por el bucle con $m < n$
 5. hacer $n - 1$, n y $n+1$ pasos por el bucle

- *Bucles anidados*: Beizer sugiere el siguiente número de pasos para realizar las pruebas:
 1. Comenzar por el bucle más interior. Establecer o configurar los demás bucles con sus valores mínimos.
 2. Llevar a cabo las pruebas de bucles simples para el bucle más interior, mientras se mantienen los parámetros de iteración (por ejemplo, contador del bucle) de los bucles externos en sus valores mínimos. Añadir otras pruebas para valores fuera de rango o excluidos.
 3. Progresar hacia fuera, llevando a cabo pruebas para el siguiente bucle, pero manteniendo todos los bucles externos en sus valores mínimos y los demás bucles anidados en sus valores «típicos».
 4. Continuar hasta que se hayan probado todos los bucles.
- *Bucles concatenados*: Se pueden probar mediante el enfoque anteriormente definido para los bucles simples, mientras cada uno de los bucles sea independiente del resto. Sin embargo, si hay dos bucles concatenados y se usa el controlador del bucle 1 como valor inicial del bucle 2, entonces los bucles no son independientes.
- *Bucles no estructurados*: En este caso se deberá de rediseñar el bucle para que se ajuste a la construcción de programación estructurada.

4.3.1 Aplicación de pruebas de caja blanca a SCOSU v2

Para el caso de SCOSU v2 se realizaron las pruebas de caja blanca descritas anteriormente, cabe mencionara que no se realizo a todo el código, desgraciadamente, realizar una prueba exhaustiva presenta ciertos problemas logísticos. Incluso para pequeños programas, el número de caminos lógicos posibles puede llegar a ser enorme, es por eso que se tomo la parte modular del sistema, es decir, donde se concentra la mayor parte de las funciones de SCOSU v2.

Se tomo el Form de la sala 3 (tomando en cuenta que son 4 salas) que es donde existe un mayor número de computadoras y por lo tanto un mayor número de funciones y conexiones a la base de datos, el Form de Asignar computadora, el Form de Curso, el Form de Inicio y el modulo, todos estos elementos en conjunto son el pilar de SCOSU v2.

Prueba del camino básico: Para realizar esta prueba se tomo en cuenta todos los segmentos de código generados dentro de los Form, cada uno compuesto por diferentes bucles y condiciones. La finalidad de la prueba es garantizar que por lo menos se ejecute una vez cada sentencia del programa, lo cual concordó con el código en un 95 %, teniendo como resultado que en ciertos momentos no se ejecutaban todos los caminos posibles del código. Gracias a los casos de pruebas se corrigieron detalles en la programación, desde condiciones mal declaradas que no permitían ejecutar un camino en específico, hasta borrar código que no cumplían con su cometido. En seguida se muestra un ejemplo de los casos de pruebas que se realizaron:

```

Case 3
For Each tab As Control In frmSala3.TabControl.Controls ← 1
  For Each control In tab.Controls ← 2
    If TypeOf control Is Label Then ← 3
      If control.TabIndex = boton.TabIndex Then ← 4
        control.Text = boton.TabIndex & " " & Format(fecha, "HH:mm") ← 6
        sHoraSalida(oCompu.noCompu) = oRegistro.horaSalida ← 8 (codigo dentro del if numero 4)
        frmSala3.lstCompusAsignadas.Items.Insert(0, " Aula " & oAula.aula & " -----> " & " Equipo " & ← 9 (codigo dentro del if numero 5)
        If frmSala3.lstCompusAsignadas.Items.Count = 11 Then ← 5
          frmSala3.lstCompusAsignadas.Items.RemoveAt(10) ← 7
        End If
      End If
      Me.Close() ← 10
    End If
  End If
Next ← 11
Next ← 12
    
```

Figura 4.1 Código de SCOSU v2

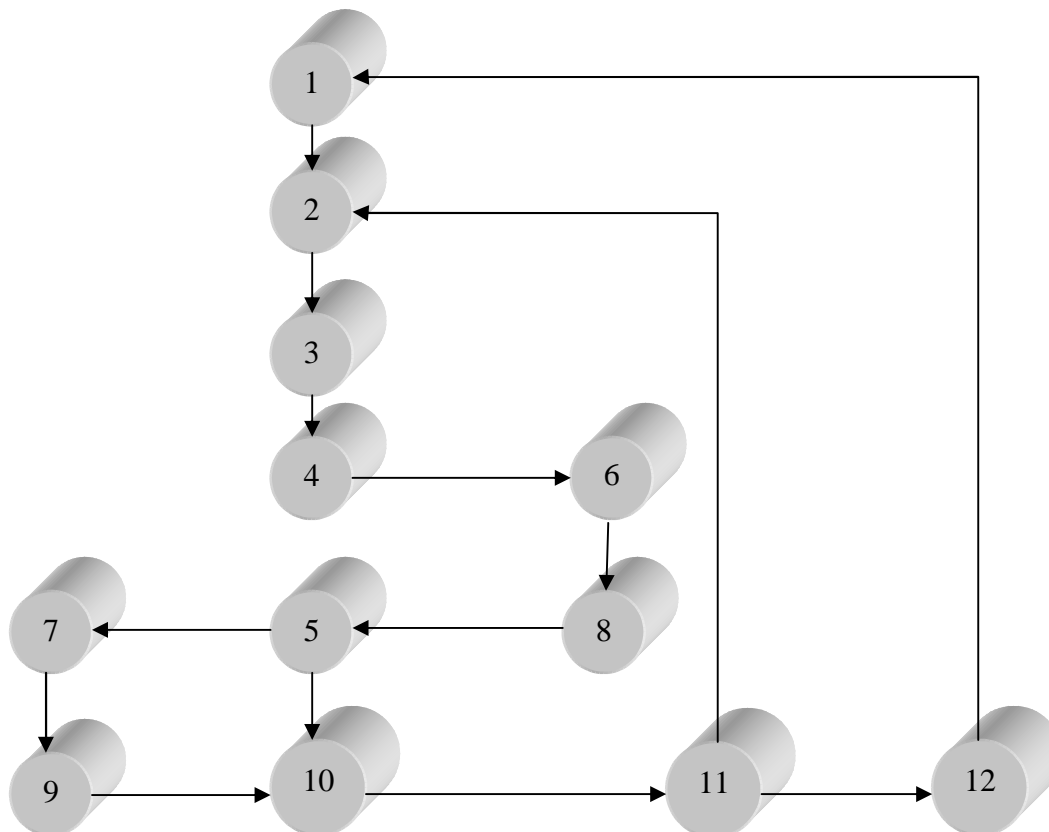


Figura 4.2 Grafo del flujo del procedimiento

Prueba de condición: Como se menciona la prueba del camino básico es muy buena por sí sola, pero siempre es recomendable realizar las pruebas de estructura de control como un completo. Aunque en la prueba anterior se aplicaron casos de pruebas a las condiciones que se encontraban en el camino, es necesario llegar a más fondo en estos puntos ya que son los causantes que se detonen código muy importante dentro del sistema.

Se realizaron los casos de pruebas en sus dos modalidades, pruebas de ramificaciones y prueba del dominio para el código que se encuentra en los Form que se consideran el pilar de SCOSU v2. La previa realización de la prueba del camino básico facilitó la tarea, ya que la base de la prueba fueron los grafos realizados anteriormente, con ellos se centraron los casos de pruebas en las condiciones establecidas en cada uno de los segmentos de código analizados, obteniendo resultado muy favorables, es decir, se encontraron pequeñas inconsistencias en algunas condiciones establecidas, como por ejemplo, en algunas de las sentencias “if” se observó que solo se realizaban una de las ramificaciones, o sea, nunca se realizaba el “else” de esa condición, por lo tanto, se quitó esa parte del código ya que no era necesario, depurando con esto el código innecesario. Otro ejemplo fue cuando se establecieron fechas límites, en un principio se fijó como fecha máxima hasta el año 2999, pero realizando los casos de pruebas se llegó a la conclusión de que no era necesario, por el contrario el sistema trabajaba más de lo que debía, por lo que se optó en dejar como fecha máxima el año 2020.

Como se puede observar no fueron errores “gigantesco” pero que si lo vemos desde el punto de vista de la eficiencia, esa pequeña depuración de código y el establecer correctamente los límites harán que el sistema sea más fácil de administrar además de que trabajara de una manera eficaz.

Prueba del flujo de datos: En la realización de los casos de pruebas del flujo de datos se siguió la recomendación y se realizaron al código donde se encontraban sentencias if y bucles anidados, además de tomar como base los grafos de la prueba del camino básico.

Primero se observaron las variables que se utilizaron dentro de los Form establecidos para los casos de pruebas, algunas se definieron en el módulo del sistema y otras eran objetos establecidos dentro de la biblioteca de clases. Una vez definidas las variables que se utilizarían, se encontraron todas las referencias dentro del código tomando aquellas con donde el código contenía sentencias if o bucles anidados. Como resultado de los casos de pruebas se volvieron a encontrar inconsistencias en el flujo de los datos, una de ellas fueron algunas variables que se definieron en el módulo del sistema para poder utilizarse de manera global, pero por cambios en la programación ya no fueron útiles, por lo que esas variables dejaron de funcionar y no se habían borrado del código.

Una vez detectadas las variables que no se utilizaban se borraron del código, facilitando con esto, una vez, la mejor administración del sistema. Por otra parte los casos de pruebas, también mostraron variables donde el alcance de las mismas se encontraba más allá de lo necesario, lo cual, en casos extraordinarios (pero que llegaban a suceder) ocasionaba pequeños errores, que no ocasionaban el cierre del sistema, pero se perdían registros de los alumnos que utilizaban las salas.

Los casos de pruebas de flujo de datos cumplieron con su cometido, mostrando esos detalles en el código que no se ven mientras se está programando pero que gracias a las pruebas se encontraron y se corrigieron obteniendo con esto un mejor desempeño.

Prueba de bucles: Esta prueba contiene cuatro variantes, bucle simple, bucle anidado, bucle concatenado y bucle no estructurado. Se hizo un análisis de cuáles eran las mejores opciones y se llegó a la conclusión de que eran la de bucle simple y anidado, el bucle concatenado se realiza con la misma técnica que el anidado, así que no tiene sentido realizar dos veces lo mismo y el bucle no estructurado solo se realiza en la programación estructura, y ya que SCOSU v2 se encuentra programado en un lenguaje completamente orientada a objetos, no tiene sentido realizar dicha prueba.

Los casos de pruebas de bucles al igual que las anteriores, se basaron en los grafos realizados en la prueba del camino básico. Las pruebas de bucles simples fueron de las más rápidas de realizar ya que como su nombre lo dice, se realizaron únicamente a código que contenían ese tipo de bucle.

En cuanto a los bucles anidados fue un poco más laboriosa su elaboración, esto fue porque SCOSU v2 cuenta con muchos bucles anidados que cuenta con varios niveles de anidamiento, pero gracias a las técnicas que se utilizaron se concluyeron con éxito todos los casos de pruebas propuestos.

Los resultados obtenidos fueron de los más favorables, los detalles encontrados fueron mínimos, esto fue posible gracias a las pruebas que se realizaron anteriormente, con la conclusión de las mismas, se corrigieron todos los inconvenientes que se encontraban en su camino, por lo tanto, en esta última prueba hubo pocos detalles que corregir.

Como podemos ver las pruebas de caja blanca ayudaron a SCOSU v2 a cambiar ciertas fallas que no se encontraron al momento de programar. Por otra parte cabe destacar la realización de los casos de prueba del camino básico, los grafos obtenidos sirvieron como base para realizar las pruebas de estructura de control, obteniendo resultados superiores a los esperados.

4.4 Pruebas de caja negra

Las pruebas de caja negra, también denominada *prueba de comportamiento*, se centran en los requisitos funcionales del software. O sea, la prueba de caja negra permite al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa.

Tomemos en cuenta que la prueba de caja negra no es opcional a la prueba de caja blanca, más bien es un complemento intenta descubrir diferentes tipos de errores que los métodos de caja blanca.

La prueba de caja negra intenta encontrar errores de las siguientes categorías:

1. Funciones incorrectas o ausentes.
2. Errores de interfaz.
3. Errores en estructuras de datos o en acceso a base de datos externas.
4. Errores de rendimiento.
5. Errores de Inicialización y de terminación.

Otra característica de las pruebas de caja negra, es que a diferencia de las pruebas de caja blanca que se realizan durante las pruebas, la de caja negra se realiza posteriores a ellas, ya que la prueba de caja negra ignora intencionadamente la estructura de control, centra su atención en el campo de la información, es decir, se basa en los datos que se obtienen una vez que el sistema se pone en marcha, cuando el usuario final utiliza las funciones programadas.

Mediante las técnicas de prueba de caja negra se obtiene un conjunto de casos de prueba que satisfacen los siguientes criterios:

- Casos de prueba que reducen, en un coeficiente que es mayor que uno, el número de casos de prueba adicionales que se deben diseñar para alcanzar una prueba razonable.
- Casos de prueba que nos dicen algo sobre la presencia o ausencia de clases de errores en lugar de errores asociados solamente con la prueba que estamos realizando.

Al igual que la prueba de caja de blanca, la prueba de caja negra se compone de varios métodos que se puede utilizar para obtener mejores resultados, los cuales son:

Métodos de pruebas basados en grafos

El primer paso en la prueba de caja negra es entender los objetos que se modelan en el software y las relaciones que conectan a estos objetos. Una vez que se ha llevado a cabo esto, el siguiente paso es definir una serie de pruebas que verifiquen que todos los objetos tienen entre ellos las relaciones esperadas.

Una vez realizado lo anterior, la prueba de software comienza creando un grafo de objetos importantes y sus relaciones, después diseñando una serie de pruebas que cubran el grafo de manera que se ejerciten todos los objetos y sus relaciones para descubrir los errores. Para llevar a cabo estos pasos, el encargado de las pruebas empieza creando un grafo (una colección de nodos que representan objetos); enlaces que representan las relaciones entre los objetos; pesos de nodos que describen las propiedades de un nodo (por ejemplo, un valor específico de datos o comportamiento de estado) y pesos de enlaces que describen alguna característica de un enlace.

¿Pero, como se representan? Los *nodos* se representan como círculos conectados por enlaces que toman diferentes formas. Un *enlace dirigido* (representado por una flecha) indica que una relación se mueve sólo en una dirección. Un *enlace bidireccional*, también denominado *enlace simétrico*, implica que la relación se aplica en ambos sentidos. Los enlaces paralelos se usan cuando se establecen diferentes relaciones entre los nodos del grafo.

Una vez entendido lo anterior podemos decir que las pruebas basadas en grafos empiezan con la definición de todos los nodos y pesos de nodos. O sea, se identifican los objetos y los atributos. Para proporcionar una indicación de los puntos de inicio y final del grafo, es útil definir unos nodos de entrada y salida. Una vez que se han identificado los nodos, se deberán establecer los enlaces y los pesos de enlaces. Conviene nombrar los enlaces que creemos.

En muchos casos, el modelo de grafo puede tener bucles (por ejemplo, un camino a través del grafo en el que se encuentra uno o más nodos más de una vez). La prueba de bucle (visto en las pruebas de caja blanca) se puede aplicar también a nivel de comportamiento (de caja negra). El grafo ayudará a identificar aquellos bucles que hay que probar. Cada relación es estudiada separadamente, de manera que se puedan obtener casos de prueba.

Cuando empieza el diseño de casos de prueba, el primer objetivo es conseguir la cobertura de nodo. Esto significa que las pruebas deberían diseñarse para demostrar que ningún nodo se ha omitido inadvertidamente y que los pesos de nodos (atributos de objetos) son correctos. Enseguida se trata la cobertura de enlaces. Todas las relaciones se prueban basándose en sus propiedades. Por ejemplo, una relación simétrica se prueba para demostrar que es, de hecho, bidireccional. Cuando se han especificado los pesos de enlace, se diseñan las pruebas para demostrar que estos pesos son válidos. Por último se invocan las pruebas de bucles.

Partición equivalente

La *partición equivalente* es un método de prueba de caja negra que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. Un caso de prueba ideal descubre de forma inmediata una clase de errores (por ejemplo, proceso incorrecto de todos los datos de carácter).

La partición equivalente se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar.

Basándonos en los conceptos definidos en las pruebas basadas en grafos, si un conjunto de objetos puede unirse por medio de relaciones simétricas, transitivas y reflexivas, entonces existe una clase de equivalencia donde una *clase de equivalencia* representa un conjunto de estados válidos o no válidos para condiciones de entrada, es decir, un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica.

Las clases de equivalencia se pueden definir de acuerdo con las siguientes directrices:

- Si una condición de entrada especifica un rango, se define una clase de equivalencia válida y dos no válidas.
- Si una condición de entrada requiere un valor específico, se define una clase de equivalencia válida y dos no válidas.
- Si una condición de entrada especifica un miembro de un conjunto, se define una clase de equivalencia válida y una no válida.
- Si una condición de entrada es lógica, se define una clase de equivalencia válida y una no válida.

En resumen podemos decir que el diseño de casos de prueba para partición equivalente se procede en dos pasos:

1. Se identifican las clases de equivalencia. Las clases de equivalencia son identificadas tomando cada condición de entrada y repartiéndola en dos o más grupos.
2. Se define los casos de prueba, donde se hace uso de las clases de equivalencia para identificar los casos de prueba.

Análisis de valores límites

El análisis de valores límite nos lleva a una elección de casos de prueba que ejerciten los valores límite. Por otra parte es una técnica de diseño de casos de prueba que complementa a la partición equivalente. En lugar de seleccionar cualquier elemento de una clase de equivalencia, lleva a la elección de casos de prueba en los extremos de la clase. En lugar de centrarse solamente en las condiciones de entrada, esta técnica obtiene casos de prueba también para el campo de salida.

Las directrices del análisis de valores límites son similares en muchos aspectos a las que proporciona la partición equivalente:

- Si una condición de entrada especifica un rango delimitado por los valores a y b , se deben diseñar casos de prueba para los valores a y b , y para los valores justo por debajo y justo por encima de a y b , respectivamente.
- Si una condición de entrada especifica un número de valores, se deben desarrollar casos de prueba que ejerciten los valores máximo y mínimo. También se deben probar los valores justo por encima y justo por debajo del máximo y del mínimo.
- Aplicar las dos directrices anteriores a las condiciones de salida. Por ejemplo, supongamos que se requiere una tabla de temperatura / presión como salida de un programa de análisis de ingeniería. Se deben diseñar casos de prueba que creen un informe de salida que produzca el máximo (y el mínimo) número permitido de entradas en la tabla.
- Si las estructuras de datos internas tienen límites preestablecidos (por ejemplo, una matriz que tenga un límite definido de 100 entradas), hay que asegurarse de diseñar un caso de prueba que ejercite la estructura de datos en sus límites.

Prueba de comparación

Hay situaciones en las que la fiabilidad del software es algo absolutamente crítico. En ese tipo de aplicaciones, a menudo se utiliza hardware y software redundante para minimizar la posibilidad de error. Cuando se desarrolla software redundante, varios equipos de ingeniería del software separados desarrollan versiones independientes de una aplicación, usando las mismas especificaciones. En esas situaciones, se deben probar todas las versiones con los mismos datos de prueba, para asegurar que todas proporcionan una salida idéntica. Luego, se ejecutan todas las versiones en paralelo y se hace una comparación en tiempo real de los resultados, para garantizar la consistencia.

Cuando se han producido múltiples implementaciones de la misma especificación, a cada versión del software se le proporciona como entrada los casos de prueba diseñados mediante alguna otra técnica de caja negra (por ejemplo, la partición equivalente). Si las salidas producidas por las distintas versiones son idénticas, se asume que todas las implementaciones son correctas. Si la salida es diferente, se investigan todas las aplicaciones para determinar el defecto responsable de la diferencia en una o más versiones.

Prueba de la tabla ortogonal

Puede aplicarse a problemas en que el dominio de entrada es relativamente pequeño pero demasiado grande para posibilitar pruebas exhaustivas. El método de prueba de la tabla ortogonal es particularmente útil al encontrar errores asociados con fallos localizados. Cuando se realiza la prueba de la tabla ortogonal, se crea una tabla ortogonal de casos de prueba. La tabla ortogonal tiene una propiedad de equilibrio. Es decir, los casos de prueba (representados por puntos negros en la Figura 4.3) están uniformemente dispersos en el dominio de prueba. El alcance de la prueba por todo el dominio de entrada es más completo.

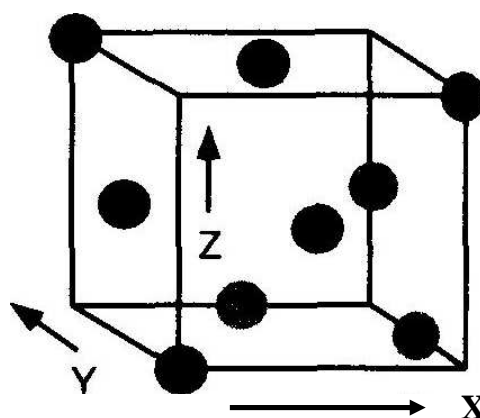


Figura 4.3 Ejemplo de una tabla ortogonal

La prueba de la tabla ortogonal nos permite proporcionar una buena cobertura de prueba con bastantes menos casos de prueba que en la estrategia exhaustiva. La prueba de la tabla ortogonal también se puede utilizar para:

- Detectar y aislar todos los fallos de modalidad simple. Un fallo de modalidad simple es un problema que afecta a un solo parámetro.
- Detectar todos los fallos de modalidad doble. Si existe un problema donde están afectados dos parámetros que intervienen conjuntamente, se llama fallo de modalidad doble.
- Fallos multimodales. Las tablas ortogonales pueden asegurar la detección únicamente de fallos de modalidad simple o doble. Sin embargo, muchos fallos en modalidad múltiple pueden ser detectados a través de estas pruebas.

4.4.1 Aplicación de pruebas de caja negra a SCOSU v2

Al igual que en la pruebas de caja blanca, se tomo la parte modular del sistema, es decir, los Form de la sala 1, 2 ,3 y 4, el Form de Asignar computadora, el Form de Curso y el Form de Inicio, además de que se añadió el Form de sancionar alumno. Por otra parte se analizaron todos los métodos que se tenían para realizar las pruebas de caja de negra, llegando a la conclusión de que se realizarían las pruebas basados en grafos, partición equivalente, análisis de valores limites y el de prueba de comparación.

Métodos de pruebas basados en grafos: En su realización se siguieron ciertos procesos y ciertos pasos para poder abarcar cada una de las funciones que realizaban los formularios anteriormente mencionados, o sea, fue necesario crear casos de pruebas donde se pudieran abarcar los objetos que se necesitaban para poder realizar los grafos. Se podría decir que después de realizar las pruebas de caja de blanca (con las técnicas descritas en ese tema) no se encontrarían inconveniente alguno, pero no fue así, por el contrario se encontró que el sistema una vez que se encontraba en funcionamiento tenía pequeños detalles que afectaba su rendimiento, con lo cual se puede decir que las pruebas cumplieron con su objetivo.

Dado que el sistema cuenta con muchos bucles, se hizo uso de la prueba de bucle (de la prueba de caja blanca) pero ahora aplicado a el comportamiento que tienen los nodos, y fue en estas instancias donde se encontraron los detalles anteriormente mencionados, se declararon varios arreglos de controles que se encontraban dentro de sentencias For, los cuales se declararon con una capacidad mucho mayor a la que necesitaban, ocasionando con esto, que el sistema en ciertas situaciones bajara su rendimiento.

Una vez encontrados los inconvenientes se procedió a corregirlos, viendo inmediatamente el cambio en la productividad del sistema. A continuación se muestra un ejemplo de los grafos utilizados en la realización de las pruebas de caja negra, en este caso se muestra el que se hizo para la acción de asignar una computadora a un alumno, así como cuando se desocupa.

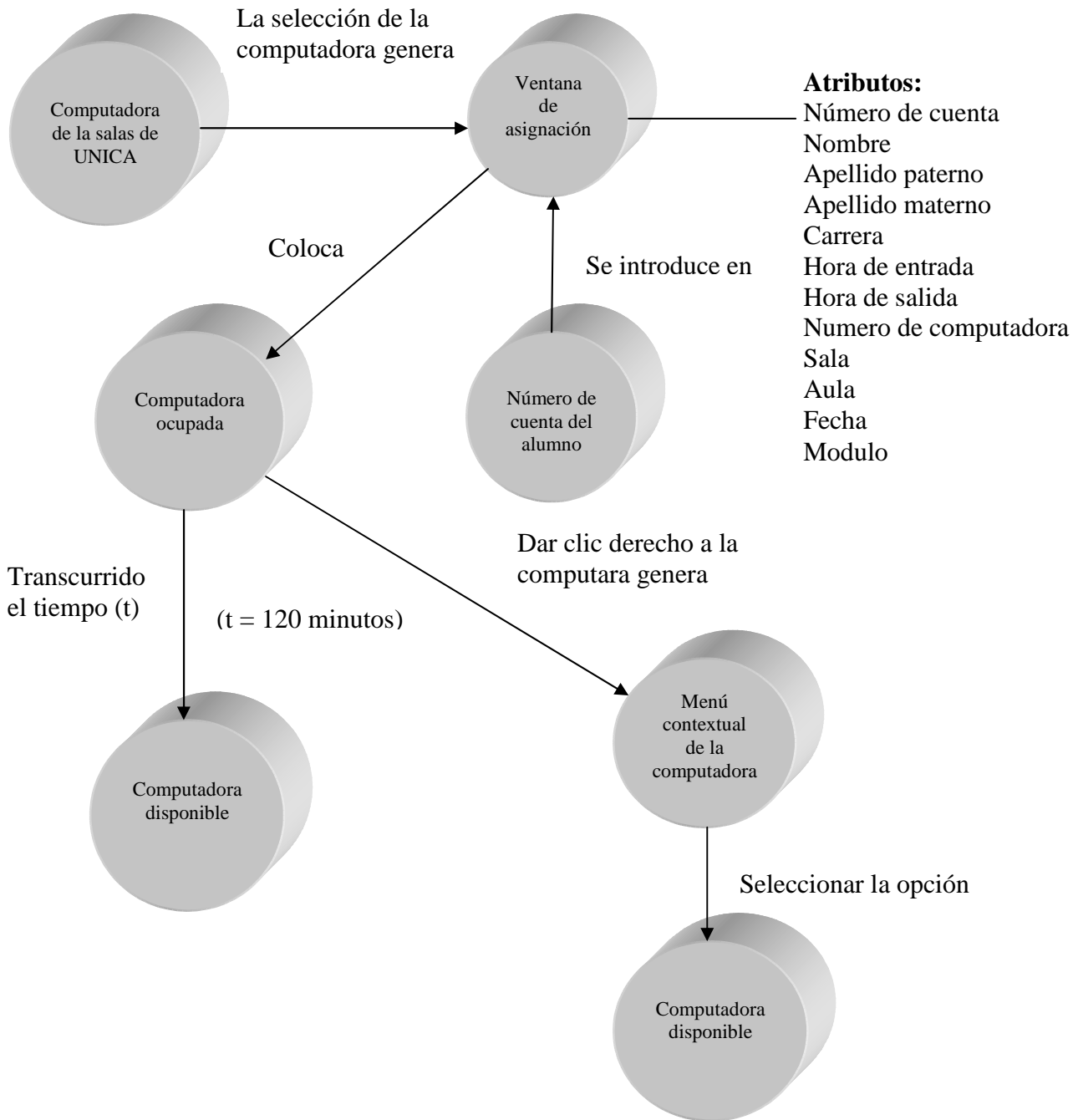


Figura 4.4 Grafo utilizado en la creación de casos de pruebas de caja negra

Partición equivalente: Cabe mencionar que esta técnica fue especialmente útil, gracias a los casos de pruebas obtenidos, fue posible verificar varias de las validaciones que se realizaron en el sistema. Como parte de la realización de la prueba también se tomo en cuenta el Form de inicio, éste Form, recibe una contraseña para poder entrar por lo que fue necesario anexarlo a los casos de pruebas.

Como el método nos indica se crearon un conjunto de estados validos y no validos para los diferentes casos de pruebas que se realizaron en los diferentes Form, un ejemplo de ello es el Form de Inicio, como se menciono, recibe una contraseña (véase Figura 4.5) que valida que sea el usuario correcto en cada una de las salas, por lo que se hicieron pruebas introduciendo contraseñas erróneas para ver que su validación se la correcta, llegando a la conclusión que las validaciones realizadas para ese campo fueron las adecuadas, no dejo ingresar al sistema si no era la contraseña indicada.



Figura 4.5 Form de inicio donde se debe introducir la contraseña

Otro ejemplo fue el realizado en el Form de asignar computadora, en él, se debe de introducir el número de cuenta del alumno para que se pueda validar que sea de la Facultad de Ingeniería, además de que se encuentre inscrito en el sistema y así poder utilizar las computadoras de las salas de UNICA.

Los casos de pruebas abarcaron muchos estados no validos. Se realizo colocando números de cuentas incorrectos, obteniendo en cada caso el mensaje adecuado, indicando que el dato que se había proporcionado era incorrecto. Y así, sucesivamente se hicieron pruebas que abarcaron todos los Form elegidos del sistema.

Al contrario de las pruebas anteriormente realizadas, en este caso, los detalles encontrados prácticamente fueron nulos, esto nos habla de que las validaciones realizadas en cada uno de los Form donde se realizaron los casos de pruebas fueron las correctas, por lo que podemos concluir que los casos de pruebas cumplieron con su cometido.

Análisis de valores limites: La realización de la prueba de partición equivalente resulto ser muy buena para verificar los campos donde era necesario saber si era correcto el dato que se insertaba o no, pero por razones que no llegan a ser muy claras muchos de los errores suelen suceder cuando se dan los valores límites del campo de entrada. Es por eso que se crearon casos de pruebas para saber qué es lo que ocurriría, en caso de llegar a los límites en ciertos campos, además de llevar esta misma prueba para los arreglos que se declararon en el sistema y así verificar su funcionalidad.

En esta prueba se anexo el Form de Apartar sala para curso, debido a que es necesario colocar una fecha de inicio y de fin, así como la hora en que comienza y termina el curso. Se comenzó por aplicar los caso de pruebas a los Form donde se manejaban fechas, como ejemplo tenemos el de Apartar sala para cursos y el de Colocar sala para mantenimiento (el cual viene incluido en los Form de las Salas 1, 2, 3 y 4).

Cundo se realizaron las pruebas a los limites inferiores no se encontró ningún detalle que pudiera alterar el servicio del sistema, esos campos se encontraban validados para que no se pudieran colocar fechas y horas inferiores a las actuales. Por otra parte, en la realización de pruebas en los imites superiores fue donde se encontraron inconvenientes que, aunque no afectaban la funcionalidad del sistema, se perdía el cometido del mismo, esto sucedió porque se podían colocar fechas de terminación hasta más allá del año 4000 cosa que no debe suceder, pues los cursos suelen tener una duración de entre 2 semanas y 3 meses y gracias a los casos de pruebas se pudo corregir ese error colocando como fecha límite un año después del actual.

Hablando de los casos de pruebas, pero enfocado a los arreglos declarados en el código, se encontraron detalles que en ciertos momentos ocasionaban el cierre el sistema. Como ejemplo tenemos un arreglo de controles que se creó para poder almacenar todas las computadoras que se encontraban en cada sala, comparando la cantidad total que obtenía, con la cantidad total de computadoras que existen en la base de datos.

En la base de datos se colocó una cantidad mayor de computadoras de las que se tenían en las salas, con el fin de facilitar la administración del sistema en caso de que sea necesario poner más computadoras. Al principio no ocurrió ningún tipo de conflicto, pero el problema surgió cuando se restauraba el sistema, esta función revisa el estado en que se encuentra cada una de las computadoras en la base de datos, las compara con las del arreglo de controles y coloca cada una en su estado antes de que se cerrara el sistema por cualquier conflicto, pero al haber más computadoras en la base de datos que en el arreglo, surgía un error que ocasionaba la terminación del programa. El error se corrigió colocando el mismo número de computadoras tanto en el arreglo de controles como en la base de datos.

Cada inconveniente que mostraban los casos de pruebas fue corregido, favoreciendo una vez más la efectividad de SCOSU v2.

Prueba de comparación: Para realizar los casos de pruebas se utilizaron los Form de las 4 salas de UNICA (las cuales se observan en las siguientes figuras) puesto que son similares, pero con características que las vuelven únicas, como el número de aulas y con ello la capacidad que tienen cada una de albergar computadoras.

Se realizaron los casos de pruebas comparando el funcionamiento que tenía SCOSU v2 en las 4 salas obteniendo pequeñas variantes entre ellas, por ejemplo en la Sala 2 en algunas ocasiones las computadoras se desocupaban automáticamente antes de las dos horas establecidas, esto fue ocasionado debido a que en ciertas computadoras no se colocó las horas correctas de salida, además de que en esa misma sala no se podía asignar correctamente un equipo a los alumnos lo cual sucedía por que la fecha del equipo no se encontraba correctamente configurada.

Otro punto muy importante que ayudó a esclarecer esta prueba fue la conexión de la sala 4, debido a que ésta sala se encuentra en un segmento de red diferente de las salas 1, 2 y 3 en ocasiones la conexión al servidor Tork de SCOSU (el antiguo sistema) era deficiente o nula. Gracias a que se utilizó ADO .Net en la programación de SCOSU v2 para establecer las conexiones este problema ya no surge, se compararon las conexiones en las cuatro salas de UNICA obteniendo prácticamente el mismo resultado en todas, es decir, su conexión al servidor Tork se realiza con éxito no importando de que salas se tratara.

Después de solucionar los inconvenientes encontrados en las pruebas anteriormente realizadas, se volvieron a proporcionar como entrada los casos de pruebas que se diseñaron, obteniendo las mismas salidas y resultados en cada una de las salas, asumiendo con esto que cada implementación de las mismas es correcta.

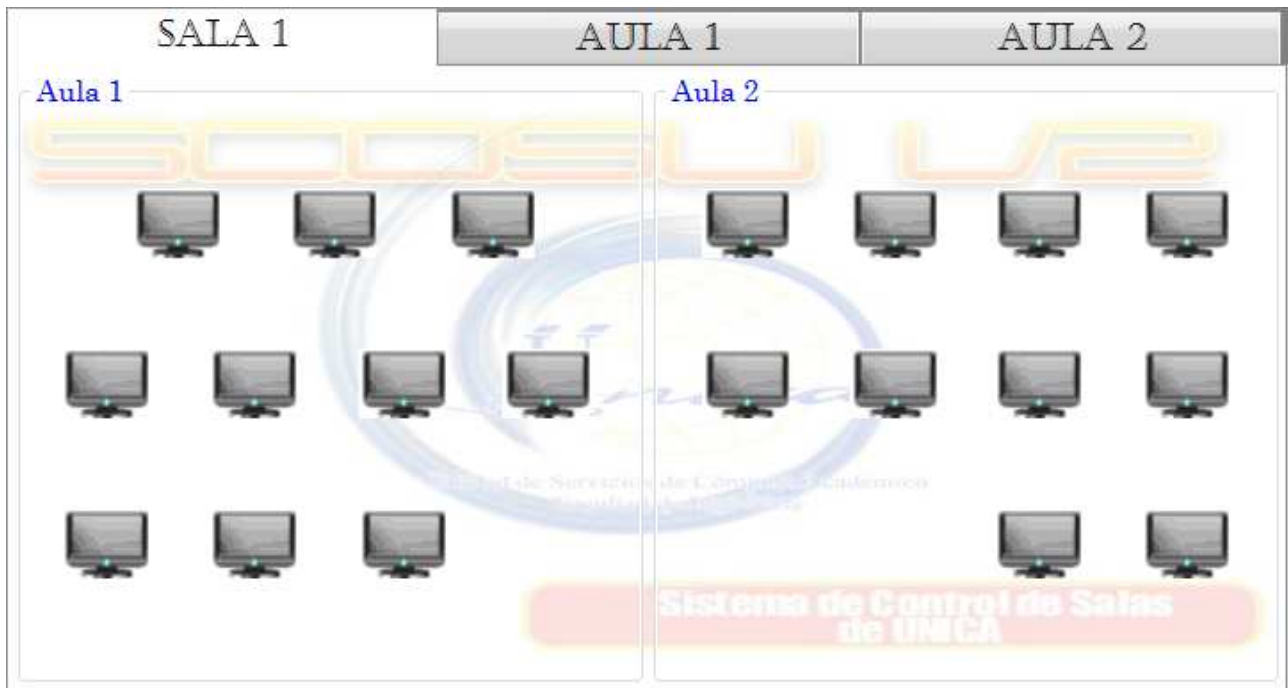


Figura 4.6 Sala 1 utilizada en la Prueba de comparación

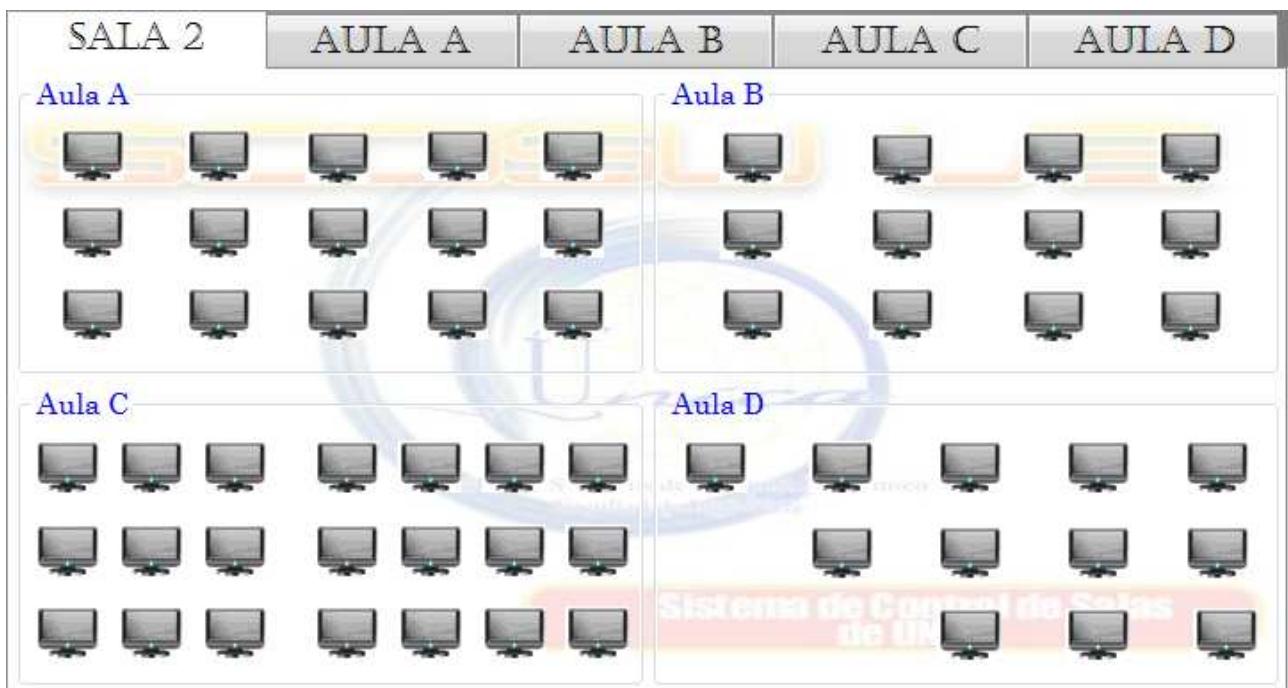


Figura 4.7 Sala 2 utilizada en la Prueba de comparación



Figura 4.8 Sala 3 utilizada en la Prueba de comparación

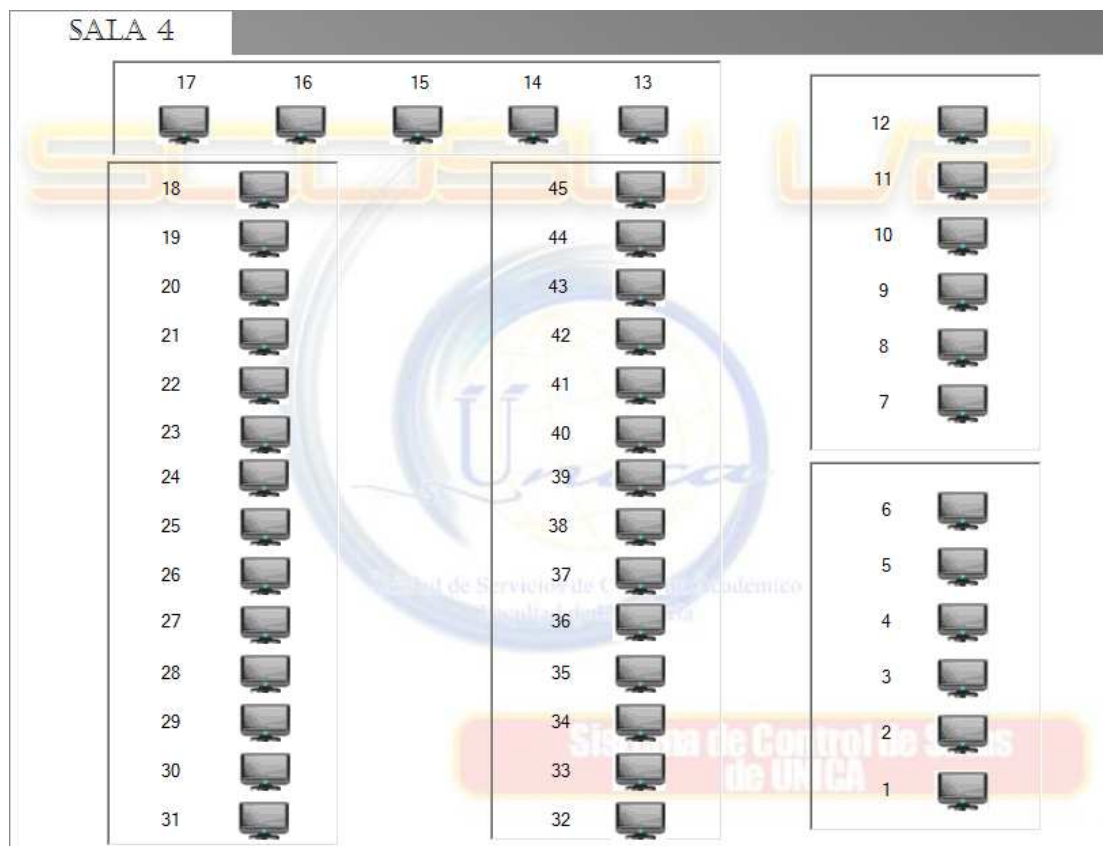


Figura 4.9 Sala 4 utilizada en la Prueba de comparación

CAPÍTULO 5

IMPLEMENTACIÓN

Un punto muy importante en la implementación del sistema SCOSU v2 son los manuales, los cuales ayudaran al personal que se encargue de operarlo a entenderlo de una manera fácil y rápida, de modo que se pueda administrar y utilizar correctamente.

5.1 Elaboración de manuales

Los manuales que se elaboraron fueron:

- **Manual de instalación:** Nos muestra paso a paso como instalar el sistema SCOSU v2, desde la aplicación hasta el Microsoft .Net Framework.
- **Manual del usuario:** Enseña el uso correcto de todas las funciones que realiza SCOSU v2, para poder obtener los resultados esperados.
- **Manual del programador:** Contiene parte del código de SCOSU v2 especificado y analizado desde el punto de vista del programador, esto con la finalidad de que cualquier persona que tome el control del sistema (administrador) tenga las suficientes herramientas para poder entender el código y de esa manera realizar modificaciones en caso de que sea necesario.

5.1.1 Manual de instalación

Para poder instalar SCOSU v2 en la computadora de los usuarios finales, es decir, las maquinas de control de las salas de UNICA y la de los jefes de sala, es necesario realizar algunas acciones antes de poder hacerlo.

5.1.1.1 Creación del archivo de instalación de SCOSU v2

Visual Studio 2005 nos brinda la oportunidad de crear proyectos de instalación e implementación, el cual se utiliza para crear el archivo que ocuparemos para instalar el sistema en todas las computadoras de las salas. Abrimos Visual Studio 2005 → Archivo → Nuevo proyecto.

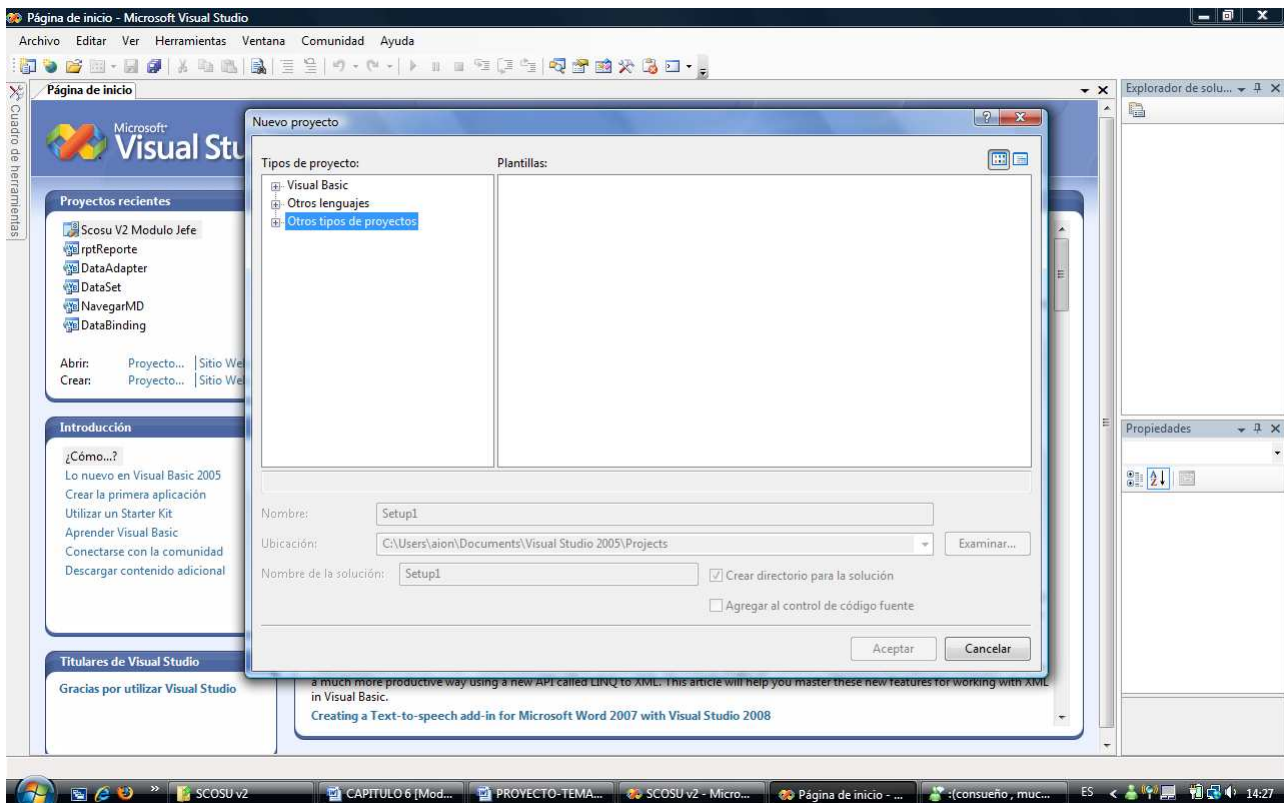


Figura 5.1 Selección del tipo de proyecto

En seguida en la parte de Tipos de proyectos seleccionamos Otros tipos de proyectos → Instalación e implementación y en Plantillas seleccionamos Proyecto de instalación, damos el nombre que queremos que tenga el proyecto, seleccionamos la ruta donde se creara la solución junto con los archivos de instalación y damos Aceptar.

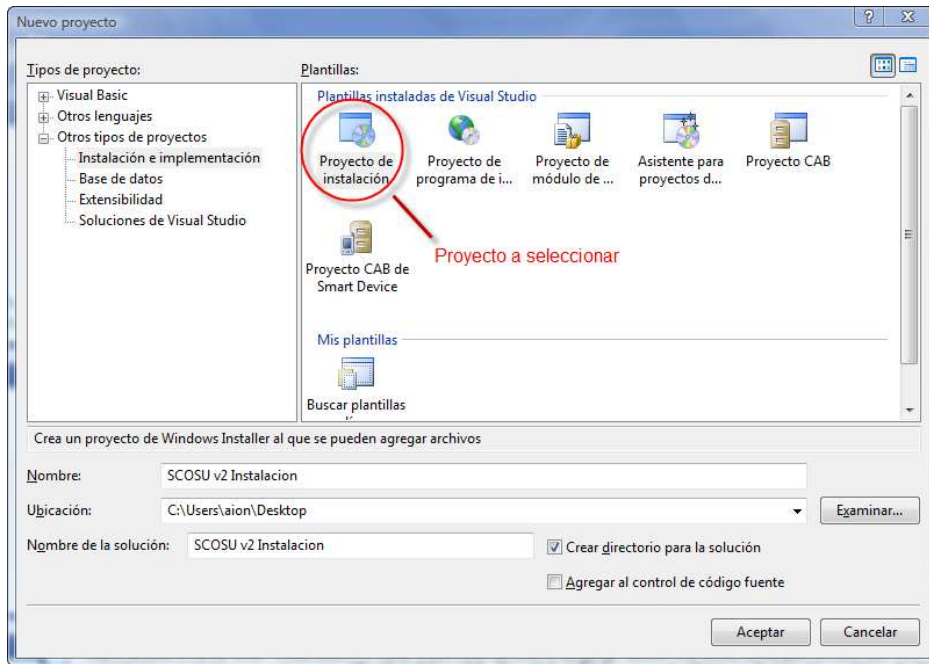


Figura 5.2 Ventana que muestra la plantilla a utilizar

Dentro del proyecto vemos que el área de trabajo se divide en dos, en el panel izquierdo que dice Sistema de archivos en el equipo de destino, seleccionamos Carpeta de la aplicación y en el panel derecho (que representa el contenido de la carpeta) damos clic derecho y seleccionamos Agregar → Archivo.

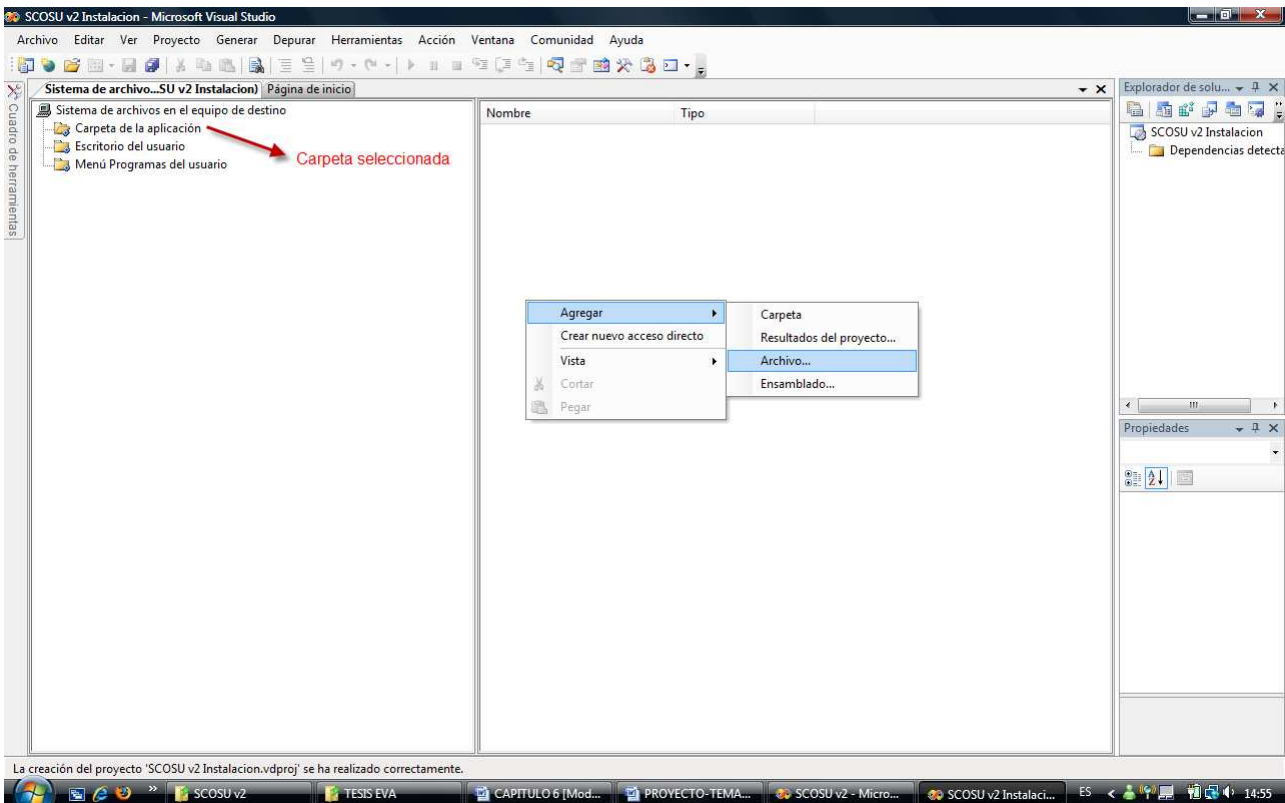


Figura 5.3 Carpeta de la aplicación

Nos abrirá una ventana donde debemos buscar el archivo ejecutable del sistema (.exe) que se encuentra dentro de la carpeta donde se guardó la solución, en este caso sería dentro de la carpeta ScosuV2 → abrir la otra carpeta ScosuV2 → abrir carpeta bin → abrir carpeta Debug y dentro de ésta se encuentra el archivo, lo seleccionamos y damos Abrir.

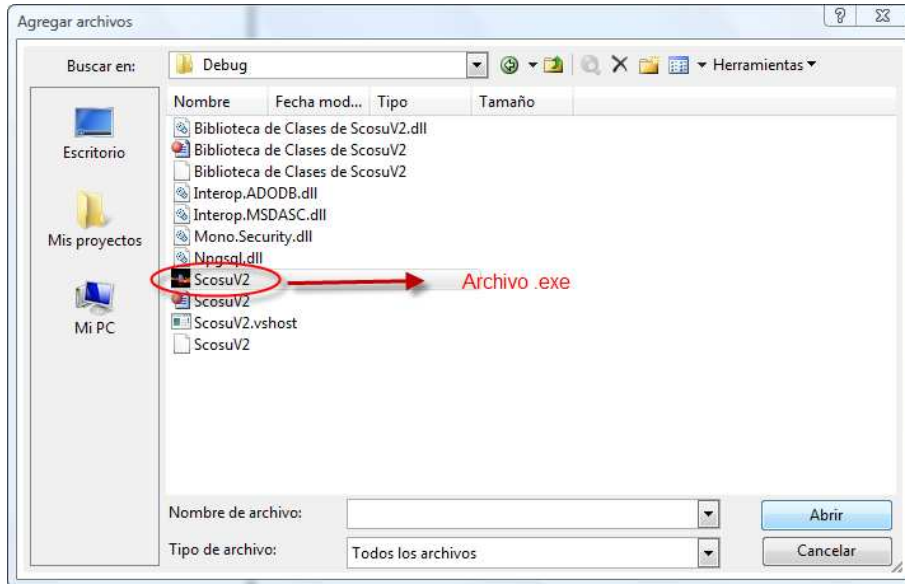


Figura 5.4 Agregando el archivo .exe

Como podemos observar en la siguiente figura, después de agregar el archivo ScosuV2.exe del sistema en la carpeta de la aplicación, automáticamente nos coloca todas las dependencias que detecto.

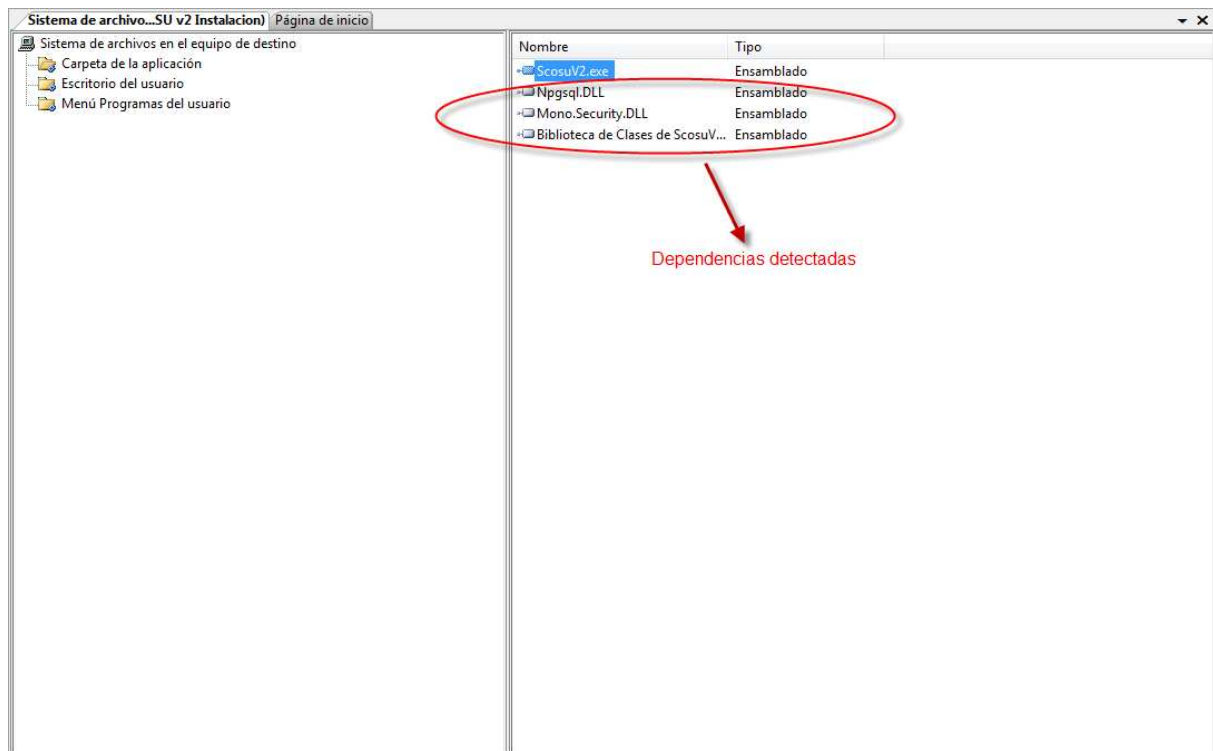


Figura 5.5 Dependencias detectadas

Ahora agregaremos el icono que llevara SCOSU v2, de la misma manera que agregamos el archivo ScosuV2.exe agregaremos el icono, entonces damos clic derecho en el panel derecho seleccionamos Agregar → Archivo y nos aparecerá la misma ventana de la Figura 5.4 Pero ahora buscamos el archivo .ico que se encuentra dentro de la carpeta Imágenes → carpeta logoSCOSU y dentro se encuentra el icono que utilizaremos.

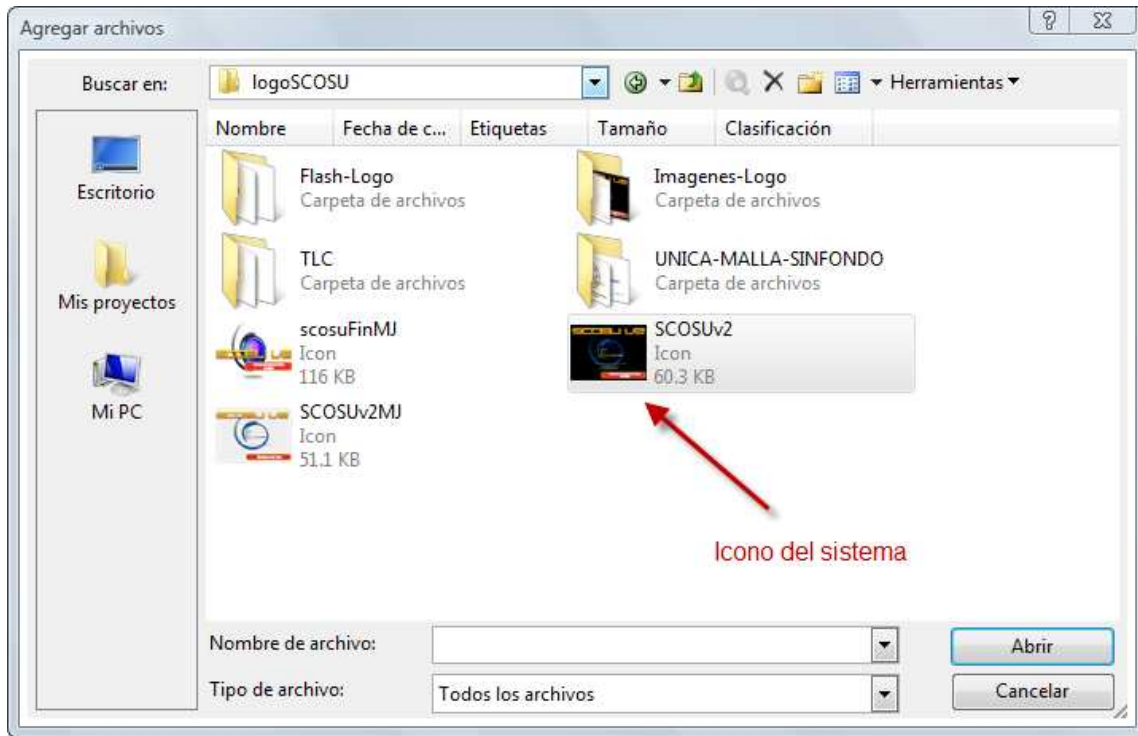


Figura 5.6 Icono de SCOSU v2

Al dar clic en Abrir veremos inmediatamente que el icono ya se encuentra agregado junto con las demás dependencias. Enseguida creamos dos accesos directos del archivo ScosuV2.exe, damos clic derecho sobre el archivo ScosuV2.exe y seleccionamos Crear acceso directo a ScosuV2.exe, a los cuales les cambiaremos el nombre a SCOSU v2.

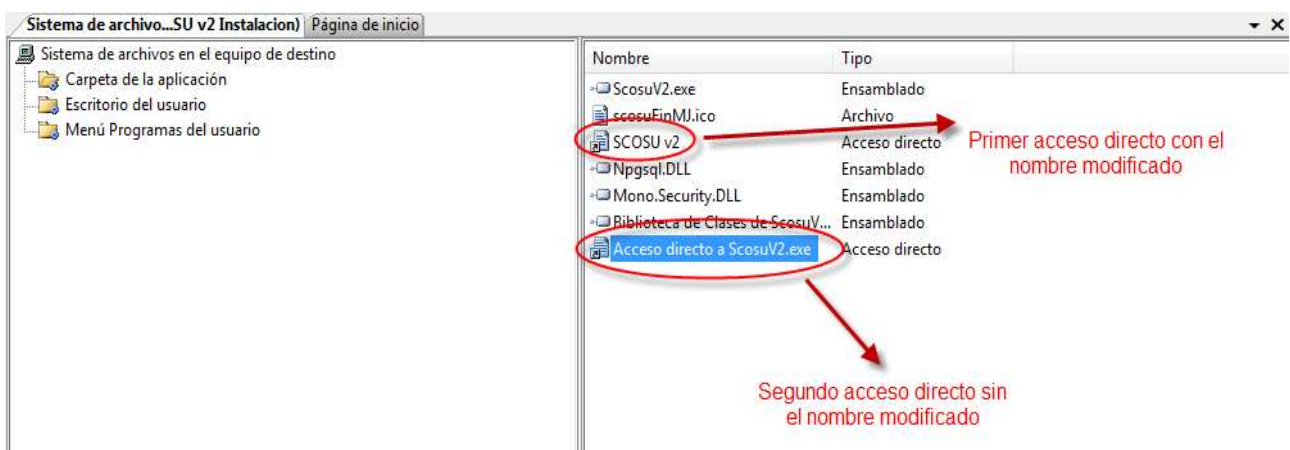


Figura 5.7 Creación de accesos directos

Después de crear los accesos directos seleccionamos alguno de ellos y le damos clic derecho → Cortar, después en el panel izquierdo elegimos la opción de Escritorio del usuario que como su nombre lo dice, representa lo que se verá en el escritorio del usuario final, enseguida damos clic derecho en el panel derecho y seleccionamos la opción Pegar.

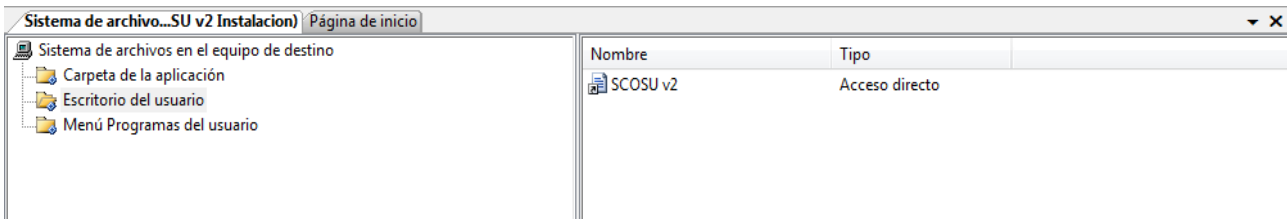


Figura 5.8 Acceso directo agregado a la carpeta Escritorio del usuario

Es aquí donde vamos a seleccionar el icono que se observara en el escritorio una vez que se instale SCOSU v2. Lo que tenemos que hacer es seleccionar el acceso directo SCOSU v2 recién agregado a la carpeta Escritorio del usuario y en la Ventana de Propiedades seleccionamos la propiedad Icon para poder establecer el archivo que servirá de icono.

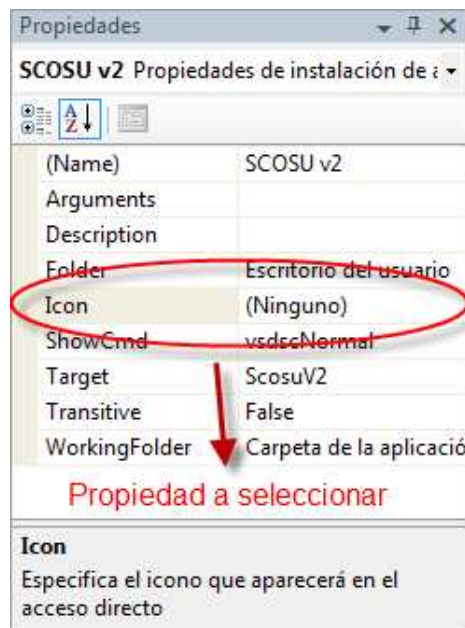


Figura 5.9 Propiedad Icon

Damos clic en Icon → (Examinar) nos mostrara una nueva ventana donde buscaremos el archivo, después damos clic en Examinar y aparecerá otra ventana donde se encuentran las carpetas que vemos en el panel izquierdo, con un doble clic sobre la Carpeta de la aplicación la abrimos y en ella se encuentra el icono, lo seleccionamos y damos Aceptar lo cual nos lleva a la ventana anterior, pero con la diferencia de que nos muestra el icono que se utilizara, le damos Aceptar.

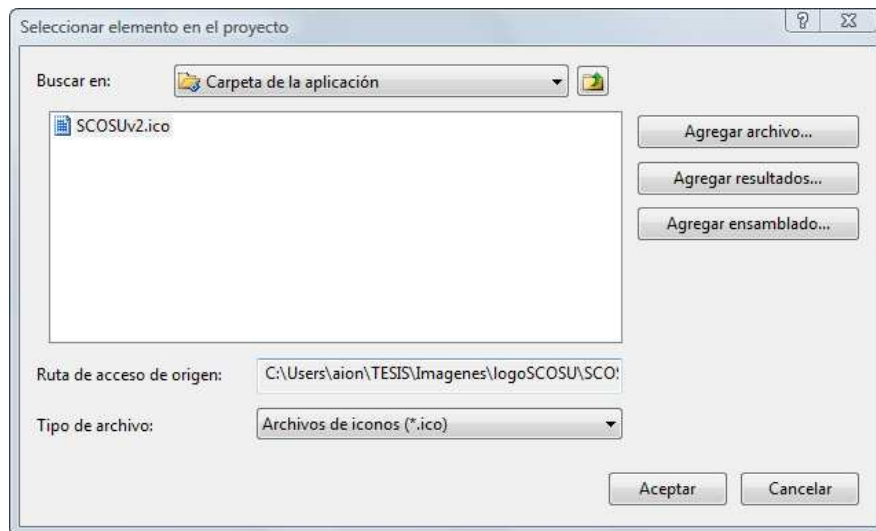


Figura 5.10 Icono dentro de la Carpeta de la aplicación

Realizado lo anterior, seleccionamos en el panel izquierdo Menú programas del usuario donde daremos clic derecho y escogeremos la opción Agregar → Carpeta, le colocamos el nombre UNICA. Ahora nos dirigimos a la Carpeta de la aplicación y cortamos el otro acceso derecho SCOSU v2 de la misma manera que él anterior, después seleccionamos la carpeta que dice UNICA y pegamos el acceso directo en el panel derecho.

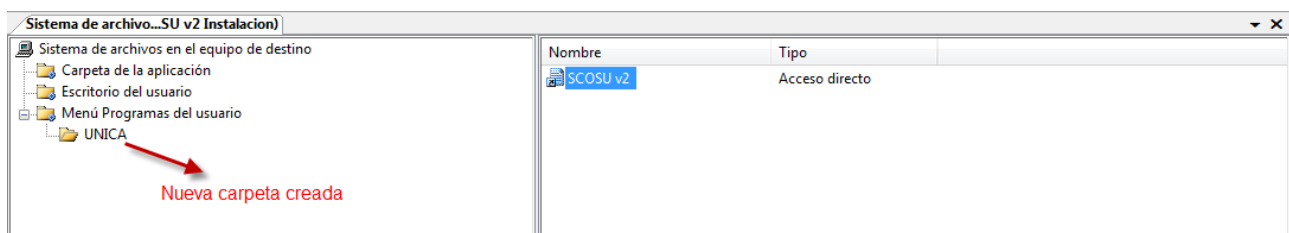


Figura 5.11 Acceso directo dentro de la carpeta UNICA

Lo siguiente es colocarle el icono al acceso directo SCOSU v2, lo cual se realiza de la misma manera que en el anterior, seleccionamos la propiedad Icon → (Examinar), en la ventana damos Examinar → doble clic en Carpeta de la aplicación → seleccionar el icono → Aceptar → Aceptar.

Ahora tenemos que definir los valores de las propiedades del proyecto por lo que seleccionamos el nombre del proyecto que se encuentra en el explorador de soluciones, para que nos muestre sus propiedades.

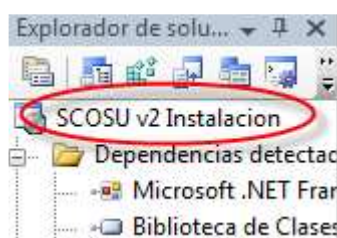


Figura 5.12 Nombre del proyecto

Las propiedades a cambiar son:

Propiedad	Valor
Author	Víctor Dueñas Tello
Manufacturer	UNICA
ProductName	SCOSU v2
Title	SCOSU v2
Version	Depende de la versión que sea, ejemplo: 2.0.0

Por último tenemos que compilar el proyecto, por lo que damos clic derecho dentro del Explorador de soluciones y seleccionamos la opción Generar.

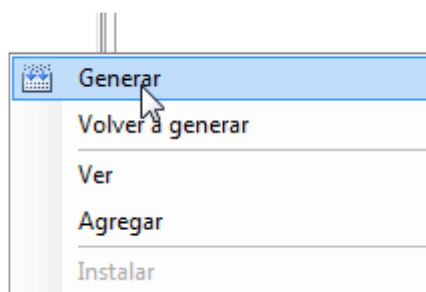


Figura 5.13 Generando la aplicación

Con esto creamos los archivos de instalación que se guardan dentro de la carpeta donde se creó el proyecto, que contiene también la solución (ver Figura 5.2), para el caso en específico de éste manual nos dirigimos a la carpeta SCOSU v2 Instalación → abrir la otra carpeta SCOSU v2 Instalación → y abrimos la carpeta Debug, es aquí donde se encuentran los archivos que ocuparemos para instalar SCOSU v2 en todas las máquinas de control de las salas de UNICA.

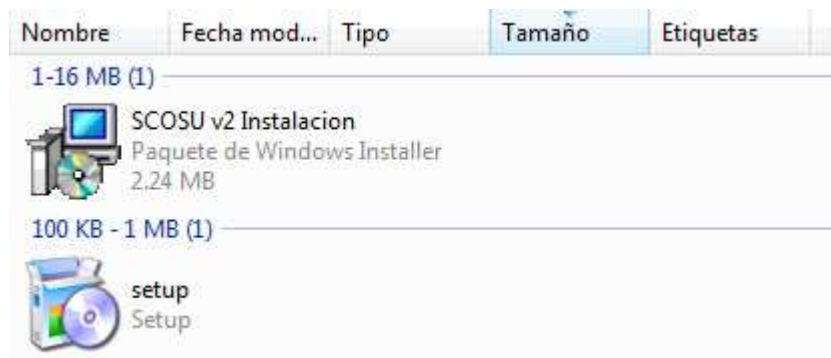


Figura 5.14 Archivos de instalación de SCOSU v2

Cuando se instale es necesario llevar los dos archivos pero se puede ejecutar el que sea.

5.1.1.2 Creación del archivo de instalación de SCOSU v2 Modulo del Jefe

Como se verá, el proceso de creación de los archivos para este modulo es casi lo mismo, solo con las diferencias propias de SCOSU v2 Modulo del jefe.

Lo primero que haremos será abrir Visual Studio 2005 → Archivo → Nuevo proyecto.

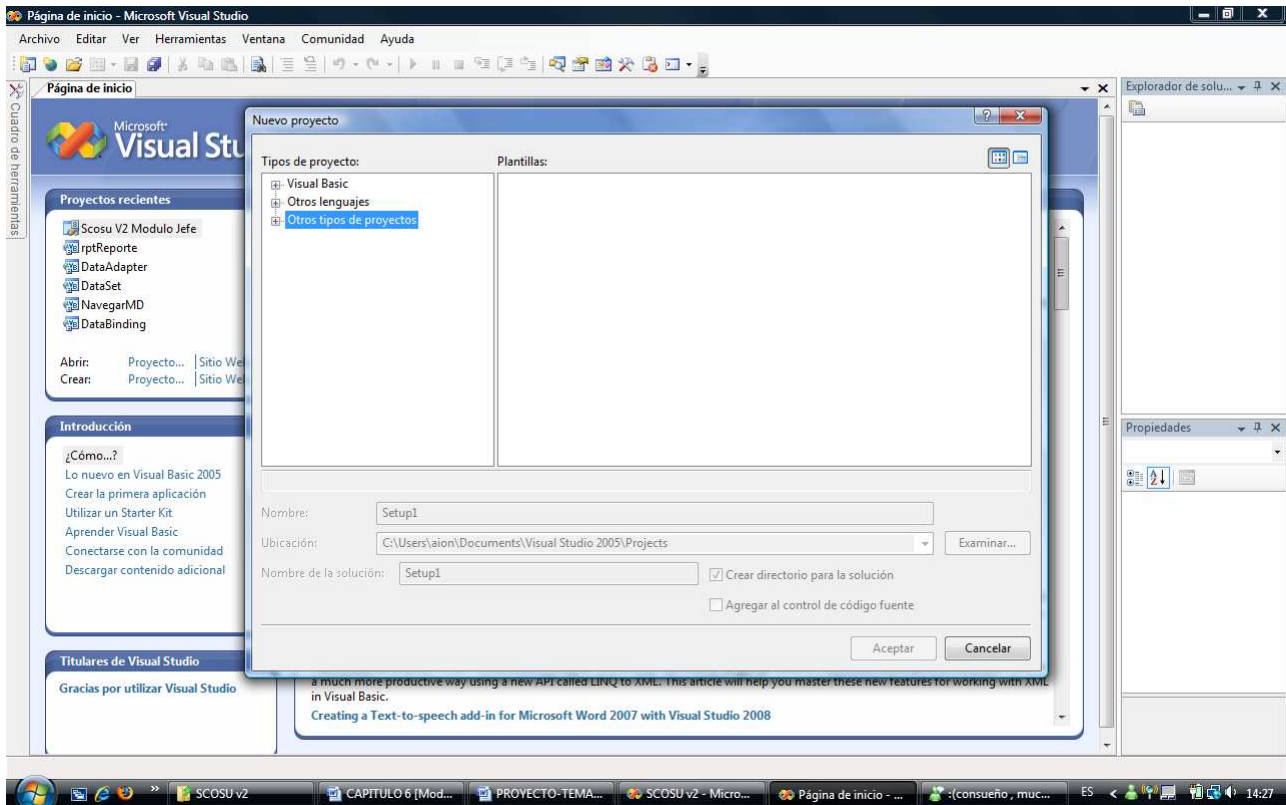


Figura 5.15 Selección del tipo de proyecto

En seguida en la parte de Tipos de proyectos seleccionamos Otros tipos de proyectos → Instalación e implementación y en Plantillas seleccionamos Proyecto de instalación, damos el nombre que queremos que tenga el proyecto, seleccionamos la ruta donde se creara la solución junto con los archivos de instalación y damos Aceptar.

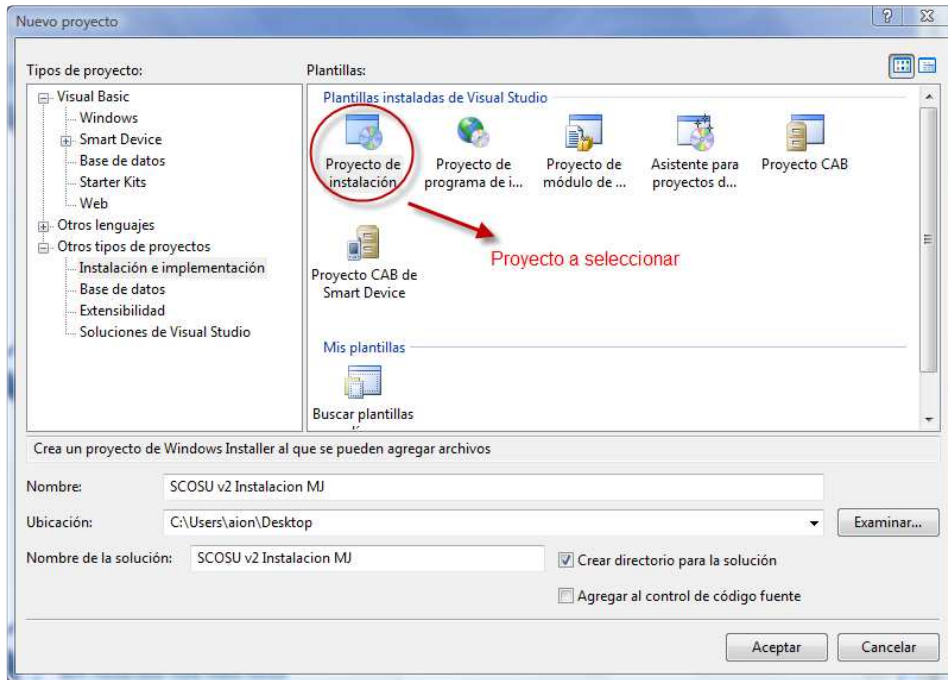


Figura 5.16 Ventana que muestra la plantilla a utilizar

Dentro del proyecto vemos que el área de trabajo se divide en dos, en el panel izquierdo que dice Sistema de archivos en el equipo de destino, seleccionamos Carpeta de la aplicación y en el panel derecho (que representa el contenido de la carpeta) damos clic derecho y seleccionamos Agregar → Archivo.

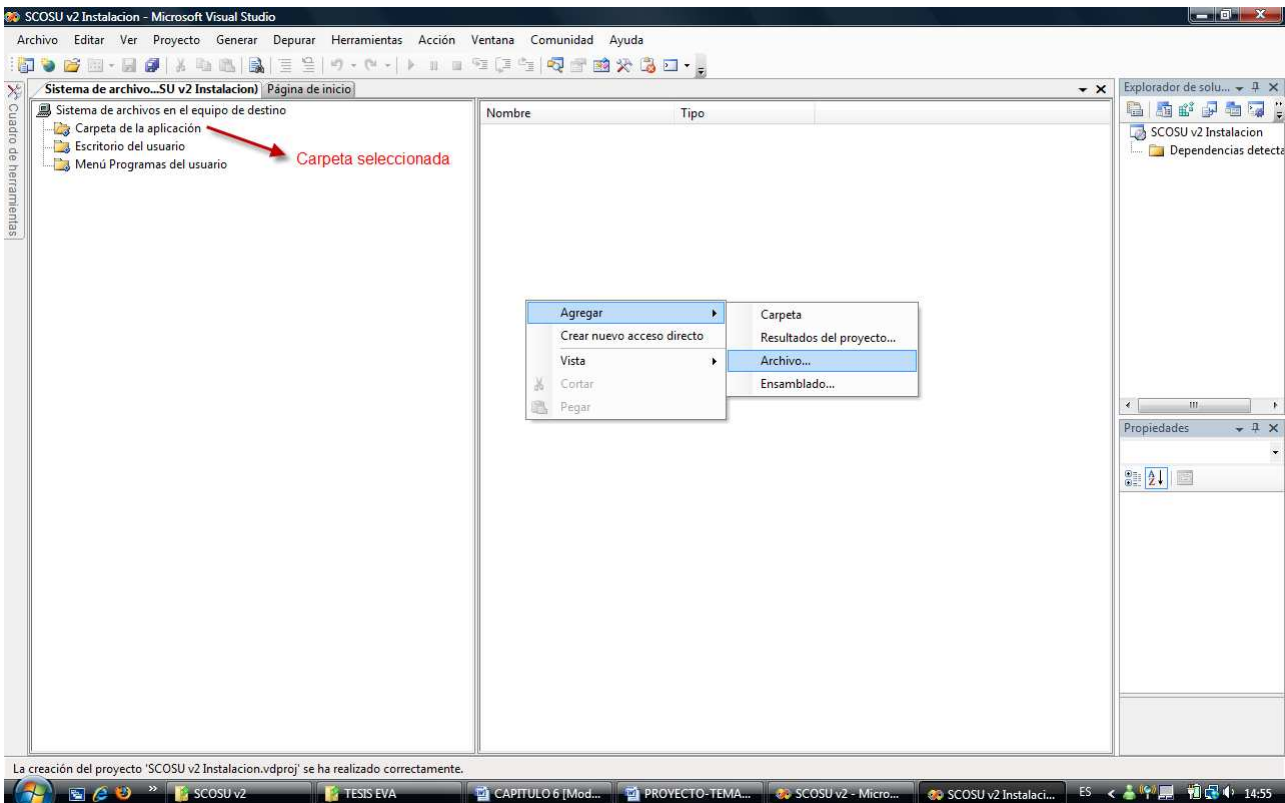


Figura 5.17 Carpeta de la aplicación

Nos abrirá una ventana donde debemos buscar el archivo ejecutable del sistema (.exe) que se encuentra dentro de la carpeta donde se guardó la solución, en este caso sería dentro de la carpeta Scosu V2 Modulo Jefe → abrir la otra carpeta Scosu V2 Modulo Jefe → abrir carpeta bin → abrir carpeta Debug y dentro de ésta se encuentra el archivo, lo seleccionamos y damos Abrir.

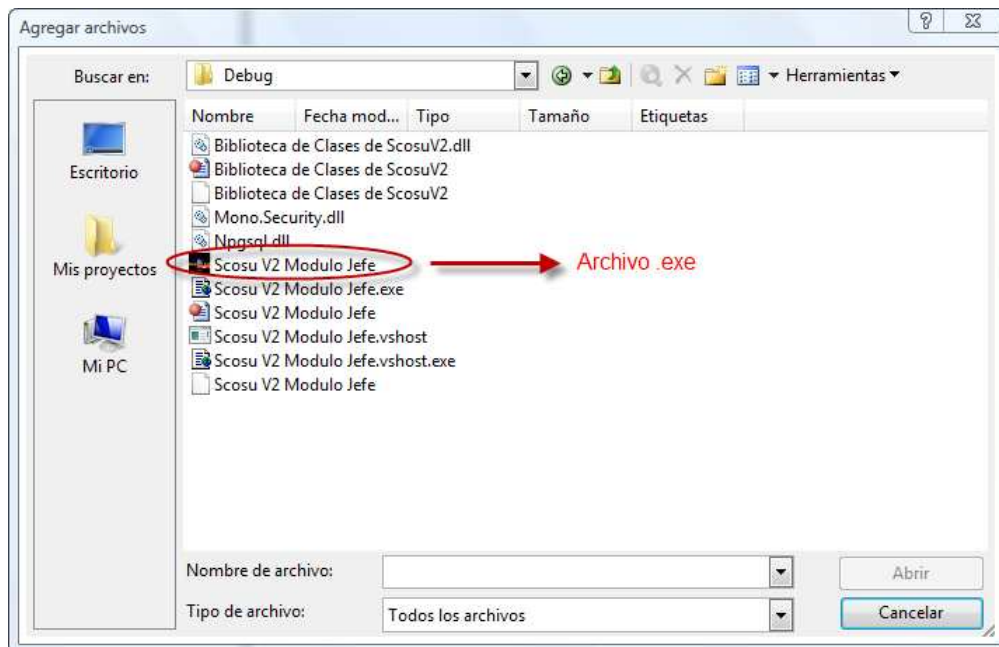


Figura 5.18 Agregando el archivo .exe

Como podemos observar en la siguiente figura, después de agregar el archivo Scosu V2 Modulo Jefe.exe del sistema en la carpeta de la aplicación automáticamente nos coloca todas las dependencias que detecto.

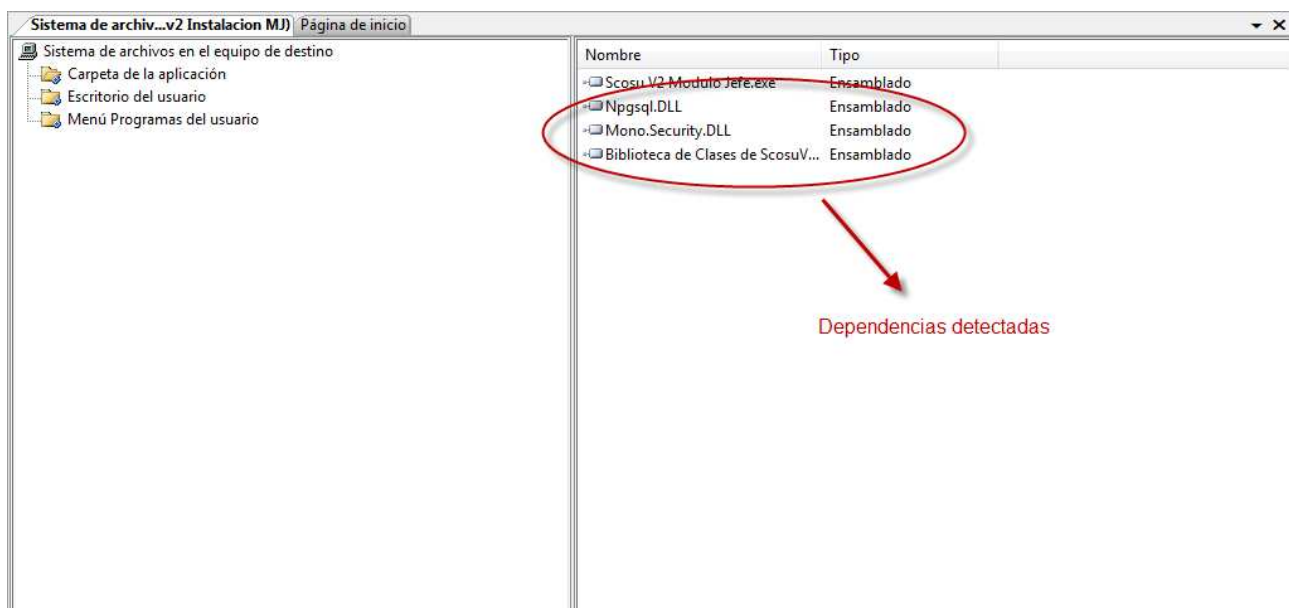


Figura 5.19 Dependencias detectadas

Ahora agregaremos el icono que llevara SCOSU v2 Modulo del Jefe, de la misma manera que agregamos el archivo Scosu V2 Modulo Jefe.exe agregaremos el icono, entonces damos clic derecho en el panel derecho seleccionamos Agregar → Archivo y nos aparecerá la misma ventana de la Figura 5.18 pero ahora buscamos el archivo .ico que se encuentra dentro de la carpeta Imágenes → carpeta logoSCOSU y dentro se encuentra el icono que utilizaremos.

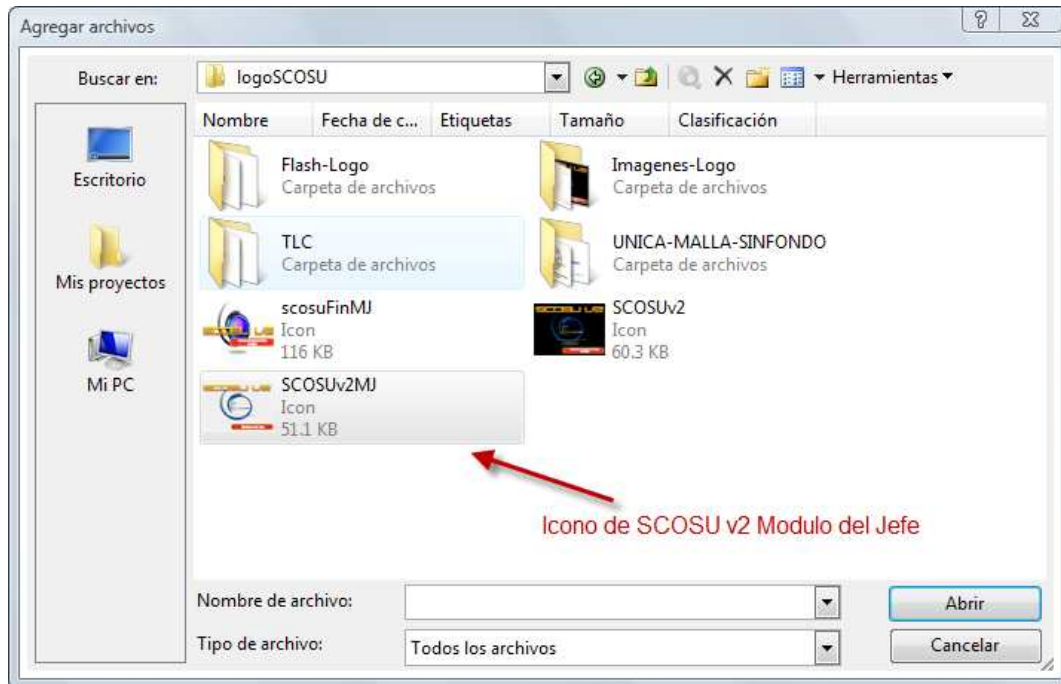


Figura 5.20 Icono de SCOSU v2 Modulo del Jefe

Al dar clic en Abrir veremos inmediatamente que el icono ya se encuentra agregado junto con las demás dependencias. Enseguida creamos dos accesos directos del archivo Scosu V2 Modulo Jefe.exe, damos clic derecho sobre el archivo Scosu V2 Modulo Jefe.exe y seleccionamos Crear acceso directo a Scosu V2 Modulo Jefe.exe, a los cuales les cambiaremos el nombre a SCOSU v2 Modulo del Jefe.

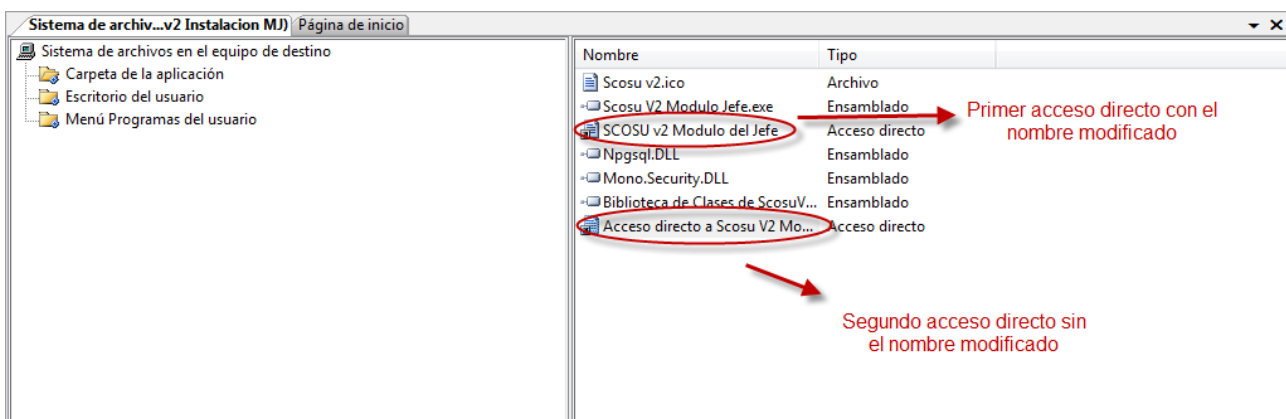


Figura 5.21 Creación de accesos directos

Después de crear los accesos directos seleccionamos alguno de ellos y le damos clic derecho → Cortar, después en el panel izquierdo elegimos la opción de Escritorio del usuario que como su nombre lo dice, representa lo que se verá en el escritorio del usuario final, enseguida damos clic derecho en el panel derecho y seleccionamos la opción Pegar.

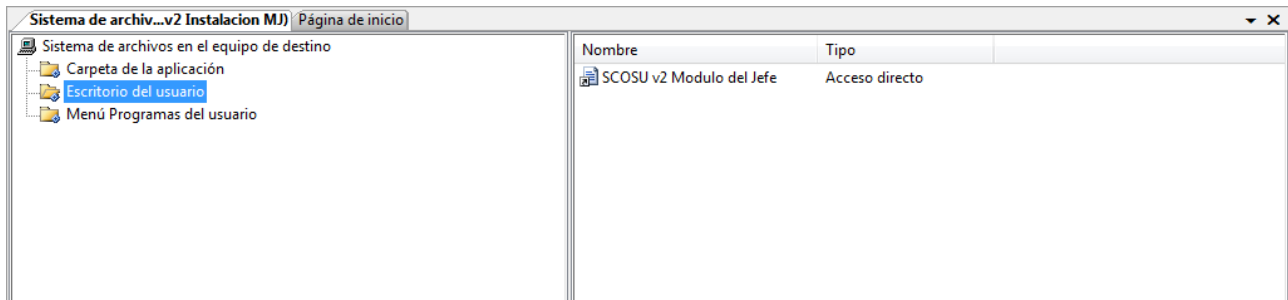


Figura 5.22 Acceso directo agregado a la carpeta Escritorio del usuario

Es aquí donde vamos a seleccionar el icono que se observara en el escritorio una vez que se instale SCOSU v2 Modulo del Jefe. Lo que tenemos que hacer es seleccionar el acceso directo SCOSU v2 Modulo del Jefe recién agregado a la carpeta Escritorio del usuario y en la Ventana de Propiedades seleccionamos la propiedad Icon para poder establecer el archivo que servirá de icono.

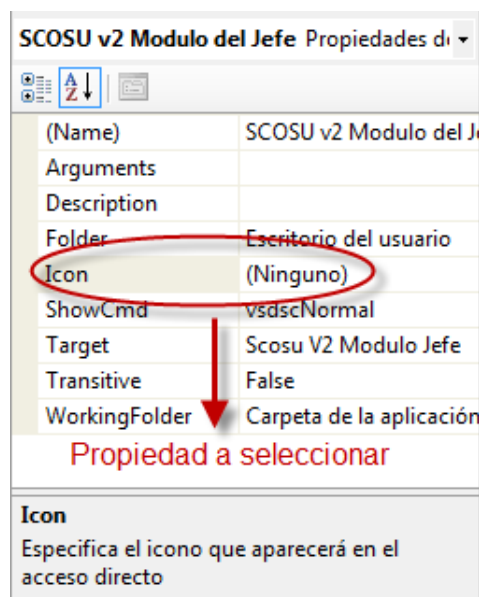


Figura 5.23 Propiedad Icon

Damos clic en el botón → (Examinar) nos mostrara una nueva ventana donde buscaremos el archivo, después damos en Examinar y aparecerá otra ventana donde se encuentran las carpetas que vemos en el panel izquierdo, con un doble clic sobre la Carpeta de la aplicación la abrimos y en ella se encuentra el icono, lo seleccionamos y damos Aceptar lo cual nos lleva a la ventana anterior, pero con la diferencia de que nos muestra el icono que se utilizara, le damos Aceptar.

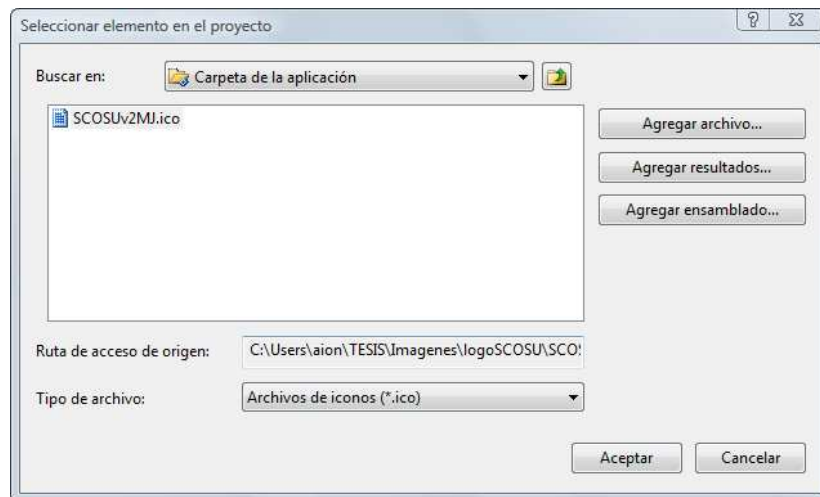


Figura 5.24 Icono dentro de la Carpeta de la aplicación

Realizado lo anterior, seleccionamos en el panel izquierdo Menú programas del usuario donde daremos clic derecho y escogeremos la opción Agregar → Carpeta, le colocamos el nombre UNICA. Ahora nos dirigimos a la Carpeta de la aplicación y cortamos el otro acceso derecho SCOSU v2 Modulo del Jefe de la misma manera que él anterior, después seleccionamos la carpeta que dice UNICA y pegamos el acceso directo en el panel derecho.

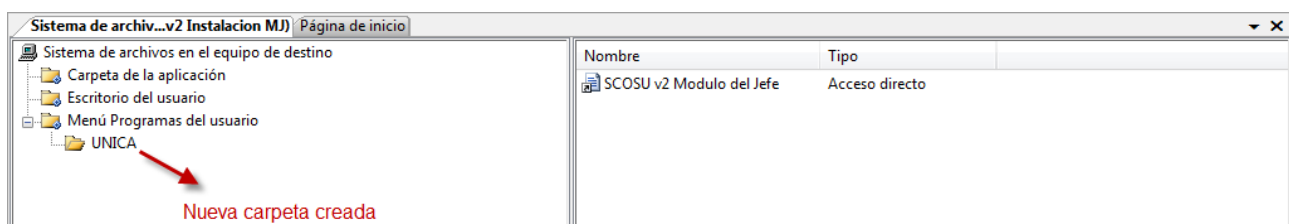


Figura 5.25 Acceso directo dentro de la carpeta UNICA

Lo siguiente es colocarle el icono al acceso directo SCOSU v2 Modulo del Jefe, lo cual se realiza de la misma manera que en el anterior, seleccionamos la propiedad Icon → (Examinar), en la ventana damos Examinar → doble clic en Carpeta de la aplicación → seleccionar el icono → Aceptar → Aceptar.

Ahora tenemos que definir los valores de las propiedades del proyecto por lo que seleccionamos el nombre del proyecto que se encuentra en el explorador de soluciones, para que nos muestre sus propiedades.

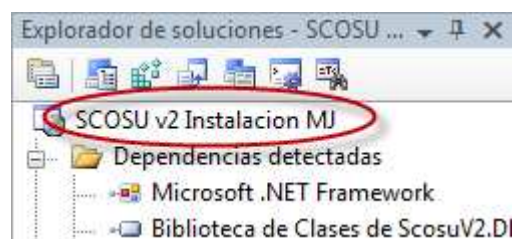


Figura 5.26 Nombre del proyecto

Las propiedades a cambiar son:

Propiedad	Valor
Author	Víctor Dueñas Tello
Manufacturer	UNICA
ProductName	SCOSU v2 Modulo del Jefe
Title	SCOSU v2 Modulo del Jefe
Version	Depende de la versión que sea, ejemplo: 2.0.0

Por último tenemos que compilar el proyecto, por lo que damos clic derecho dentro del Explorador de soluciones y seleccionamos la opción Generar.

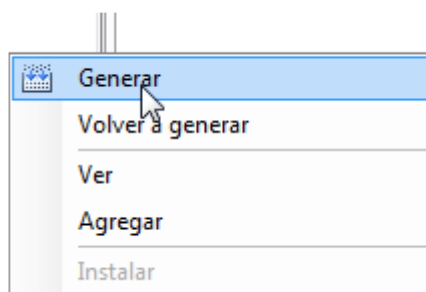


Figura 5.27 Generando la aplicación

Con esto creamos los archivos de instalación que se guardan dentro de la carpeta donde se creó el proyecto, que contiene también la solución (ver Figura 5.16), para el caso en específico de éste manual nos dirigimos a la carpeta SCOSU v2 Instalación MJ → abrir la otra carpeta SCOSU v2 Instalación MJ → y abrimos la carpeta Debug, es aquí donde se encuentran los archivos que ocuparemos para instalar SCOSU v2 Modulo del Jefe en todas las maquinas de los jefes de salas.

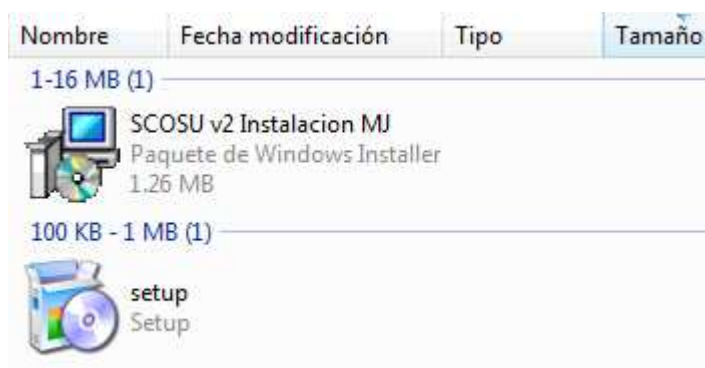


Figura 5.28 Archivos de instalación de SCOSU v2 Modulo del Jefe

Cuando se instale es necesario llevar los dos archivos pero se puede ejecutar el que sea.

5.1.1.3 Instalación de SCOSU v2

SCOSU v2 se instalara en las maquinas de control ubicadas en las cuatro diferentes salas.

Lo primero es darle doble clic a cualquiera de los dos archivos de instalación, con lo cual nos mostrara la siguiente ventana.

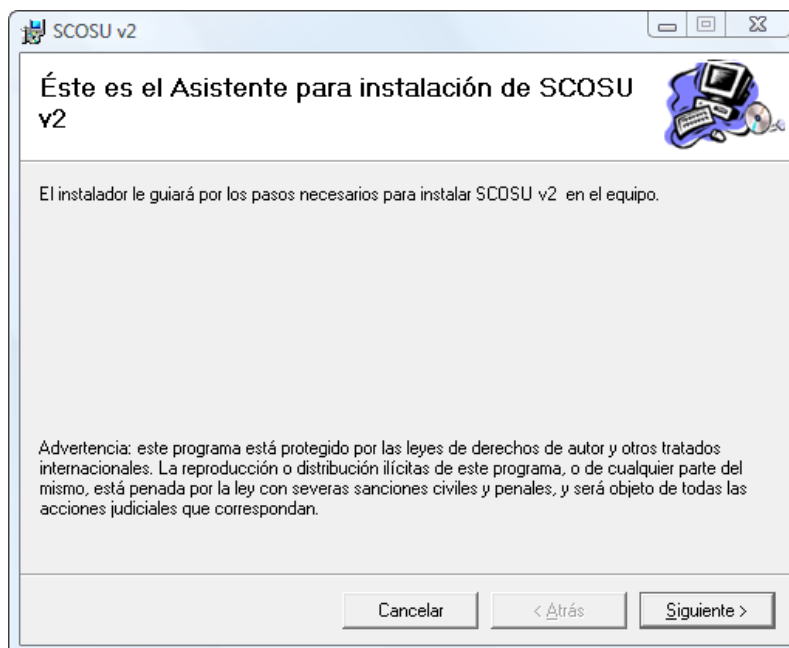


Figura 5.29 Ventana de instalación de SCOSU v2

Damos clic en Siguiete para que nos muestre otra ventana donde seleccionaremos la carpeta donde queremos que se instale.

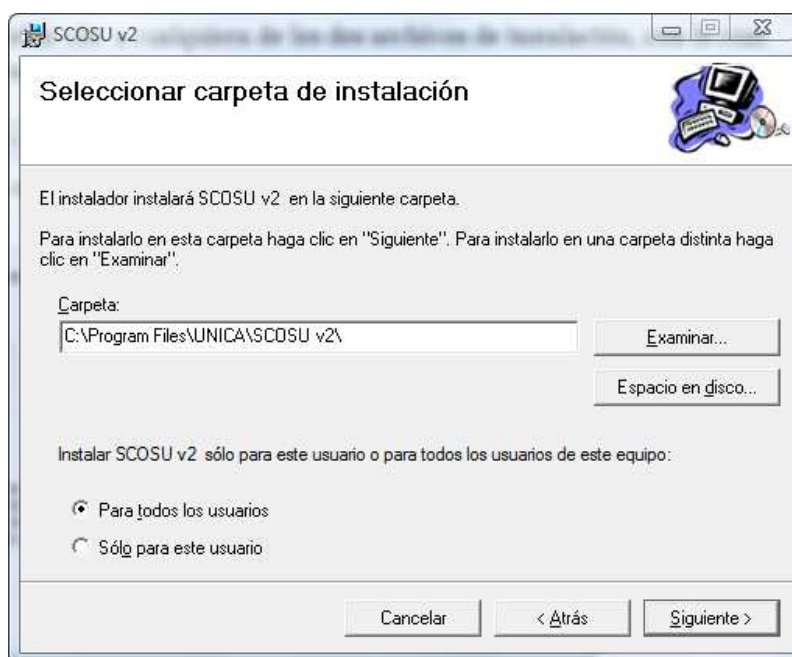


Figura 5.30 Ventana de selección de carpeta

En ella dejamos la ruta de la carpeta que tiene por default, además de seleccionar la opción Para todos los usuarios. Damos clic en Siguiente, lo cual nos lleva a una ventana donde nos menciona que el instalador se encuentra listo para instalar SCOSU v2 en el equipo, damos Siguiente para que se instale el sistema.



Figura 5.31 Instalando SCOSU v2

Una vez que termine, nos mostrara una ventana anunciando la correcta instalación de SCOSU v2.

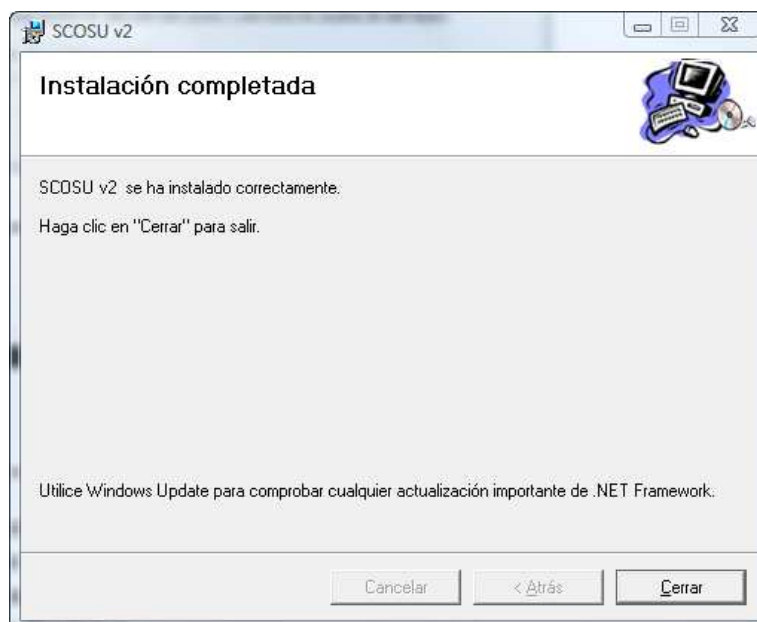


Figura 5.32 Instalación correcta

Damos clic en Cerrar y podemos ver que el icono de SCOSU v2 se encuentra en el escritorio del usuario y en el menú Inicio → Todos los programas, se encuentra una carpeta llamada UNICA la cual también contiene el sistema.

Lo siguiente es configurar el formato de la hora de la computadora donde se instaló SCOSU v2, así que nos vamos a Panel de control y seleccionamos Configuración regional y de idioma.

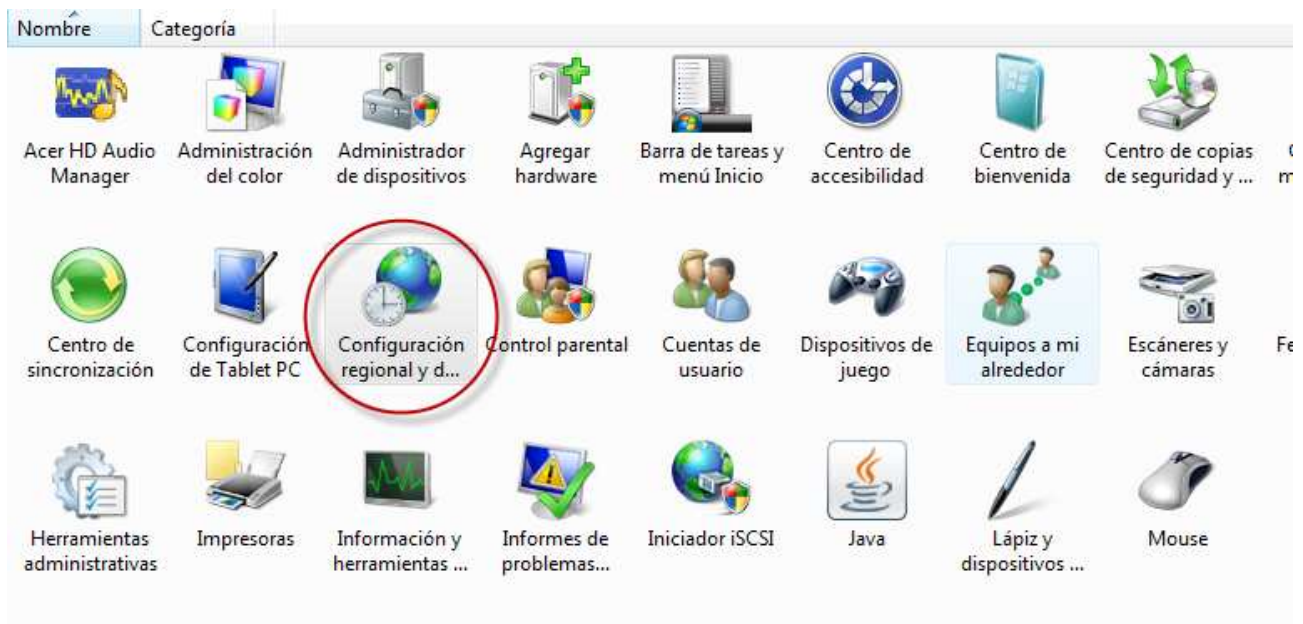


Figura 5.33 Panel de control

Nos abrirá una ventana, seleccionamos la pestaña de Formatos, donde dice Formato actual debe tener Español (México), en seguida damos clic en el botón Personalizar este formato.

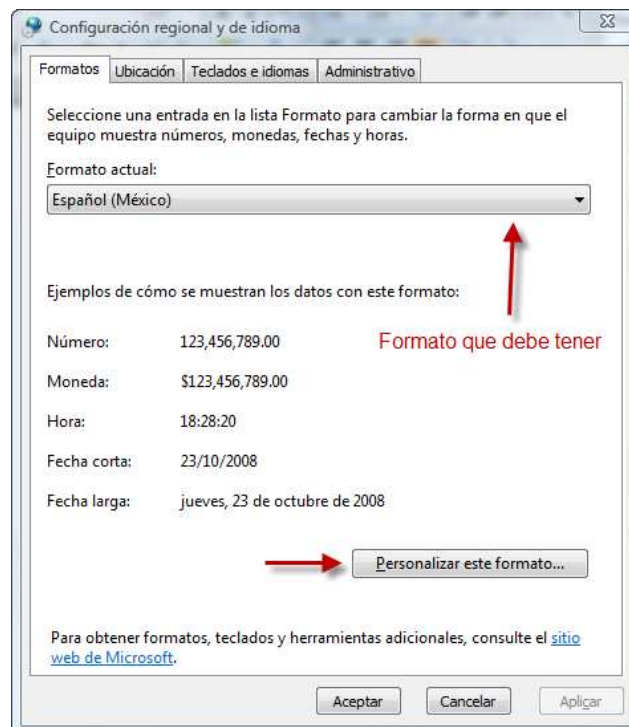


Figura 5.34 Ventana de formato

En la ventana que nos abre seleccionamos la pestaña Hora, en el Formato de hora seleccionamos H:mm:ss, en Símbolo a.m. borramos su contenido dejándolo en blanco, en Símbolo p.m. hacemos lo mismo, quedando de la siguiente manera:

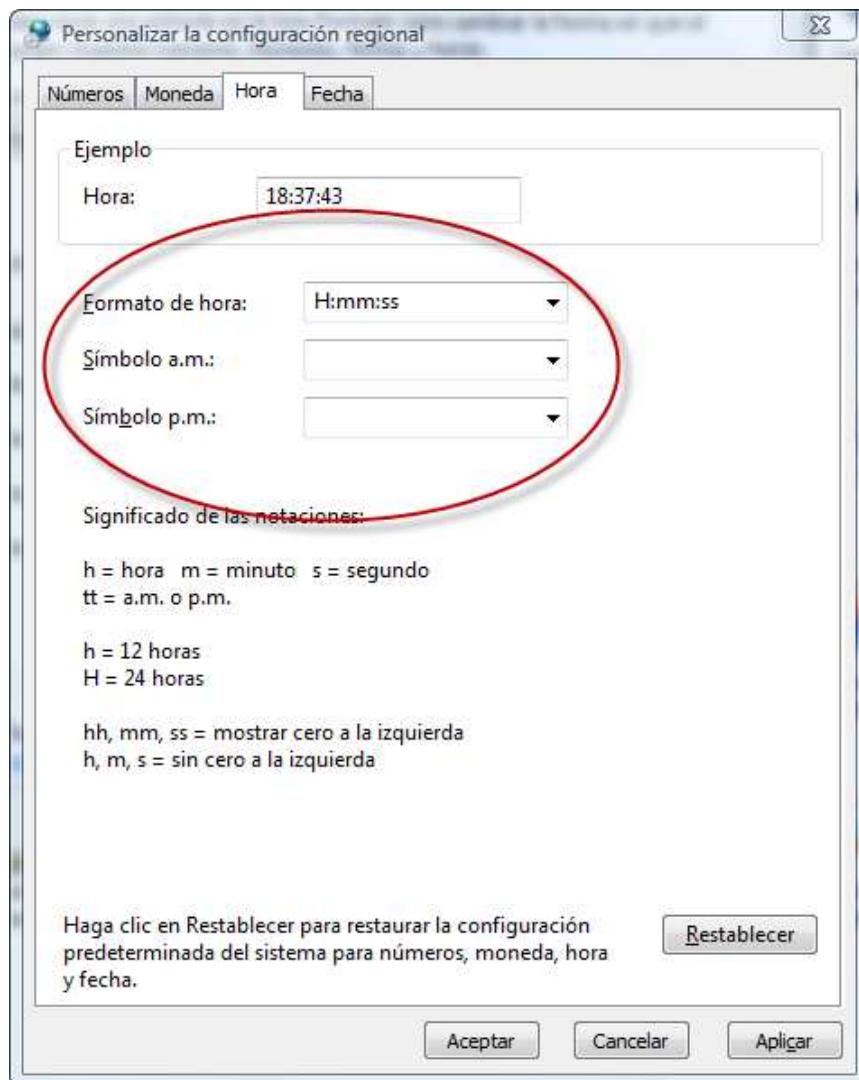


Figura 5.35 Configuración del formato de la hora

Después de realizar los cambios damos Aceptar, cerrando esta ventana y mostrándonos la anterior en la cual también damos Aceptar. Con esto configuramos el formato de las horas con las que trabaja SCOSU v2.

Nota: Si cuando se introduce la contraseña para entrar al sistema SCOSU v2 nos muestra el error de "La contraseña es incorrecta o no es la sala. Verifica los datos" y se encuentra completamente seguro de que coloca la contraseña correcta, debe de verificar que la computadora tenga la ip que le corresponde, si no se cuenta con ese dato, debe preguntarle al jefe de la sala donde se encuentre instalando SCOSU v2.

5.1.1.4 Instalación de SCOSU v2 Modulo del Jefe

SCOSU v2 Modulo del Jefe se instalara en las maquinas de los jefes de sala.

Lo primero es darle doble clic a cualquiera de los dos archivos de instalación, con lo cual nos mostrara la siguiente ventana.

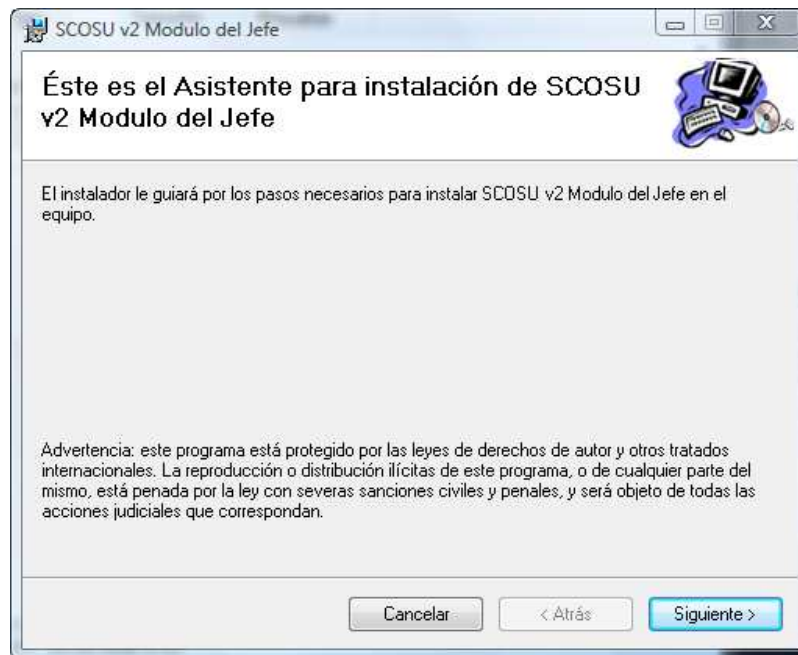


Figura 5.36 Ventana de instalación de SCOSU v2 Modulo del Jefe

Damos clic en Siguiete para que nos muestre otra ventana donde seleccionaremos la carpeta donde queremos que se instale.



Figura 5.37 Ventana de selección de carpeta

En ella dejamos la ruta de la carpeta que tiene por default, además de seleccionar la opción Para todos los usuarios. Damos clic en Siguiente, lo cual nos lleva a una ventana donde nos menciona que el instalador se encuentra listo para instalar SCOSU v2 Modulo del Jefe en el equipo, damos Siguiente para que se instale el sistema.

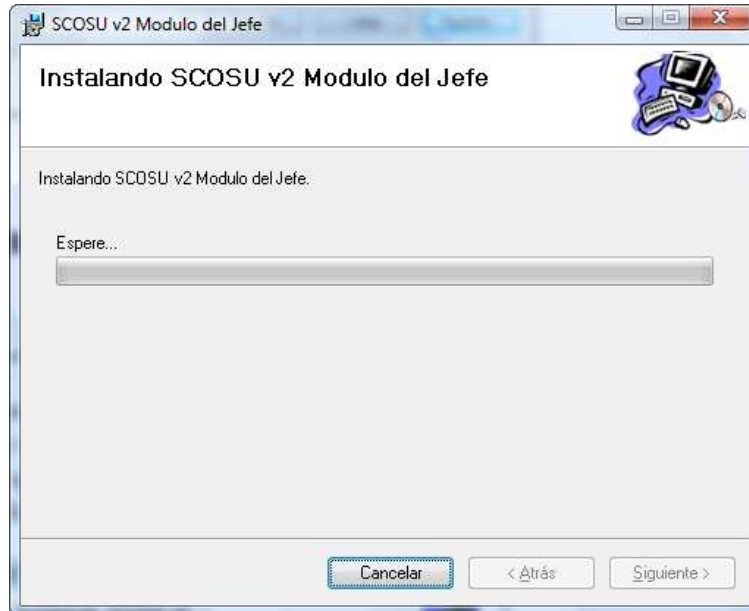


Figura 5.38 Instalando SCOSU v2 Modulo del Jefe

Una vez que termine, nos mostrara una ventana anunciando la correcta instalación de SCOSU v2 Modulo del Jefe.

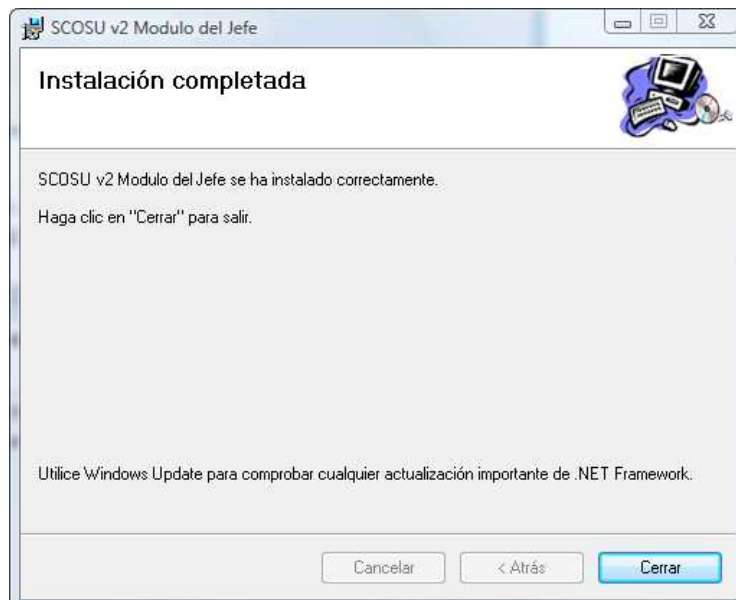


Figura 5.39 Instalación correcta

Damos clic en Cerrar y podemos ver que el icono de SCOSU v2 Modulo del Jefe se encuentra en el escritorio del usuario y en el menú Inicio → Todos los programas, se encuentra una carpeta llamada UNICA la cual también contiene el sistema.

5.1.1.5 Instalación de Microsoft .Net Framework

Microsoft .Net Framework se instalara en las computadoras que no se halla instalado previamente el Visual Studio 2005. SCOSU v2 se creó en la versión Microsoft .Net Framework 2.0 es por eso que se necesita instalar esta misma versión o superior, que sería la versión 3.5 que es la contiene Visual Studio 2008 actualmente. Cabe mencionar que para fines de éste manual se instalara la versión 3.5.

Antes de comenzar se tiene que verificar que la computadora en la cual se va a instalar el Microsoft .Net Framework 3.5 se encuentre conectada a internet, ya que el archivo dotNetFx35setup los descargara desde la página de Microsoft.

Para instalarlo comenzaremos dando doble clic al archivo dotNetFx35setup el cual se muestra en la siguiente figura.

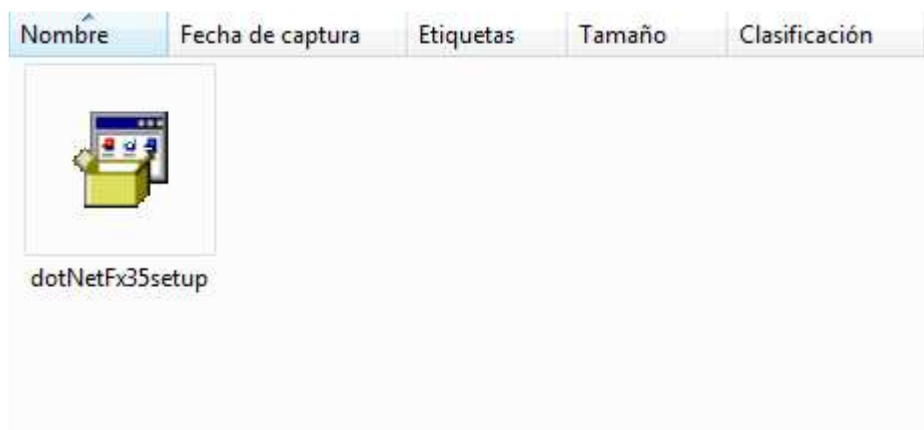


Figura 5.40 Archivo para instalar Microsoft .Net Framework 3.5

Enseguida nos muestra la siguiente ventana.

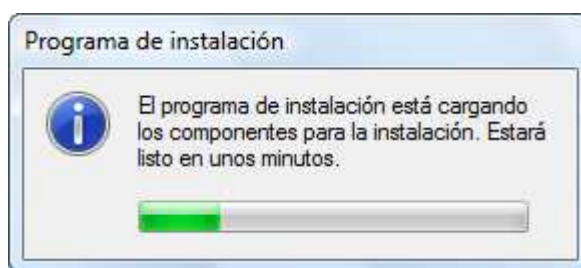


Figura 5.41 Cargando los componentes

Ya que termine de cargar los componentes necesarios para la instalación nos mostrar una venta donde hace referencia a los términos de la licencia de uso de software de Microsoft, seleccionamos la opción He leído los términos del Contrato de licencia y los ACEPTO, damos clic en instalar.



Figura 5.42 Términos del contrato

Inmediatamente comenzara a descargar el Microsoft .Net Framework 3.5, esto puede tardar desde unos segundos hasta varios minutos, dependiendo de la velocidad de la taza de transferencia con la que se cuente (ancho de banda).

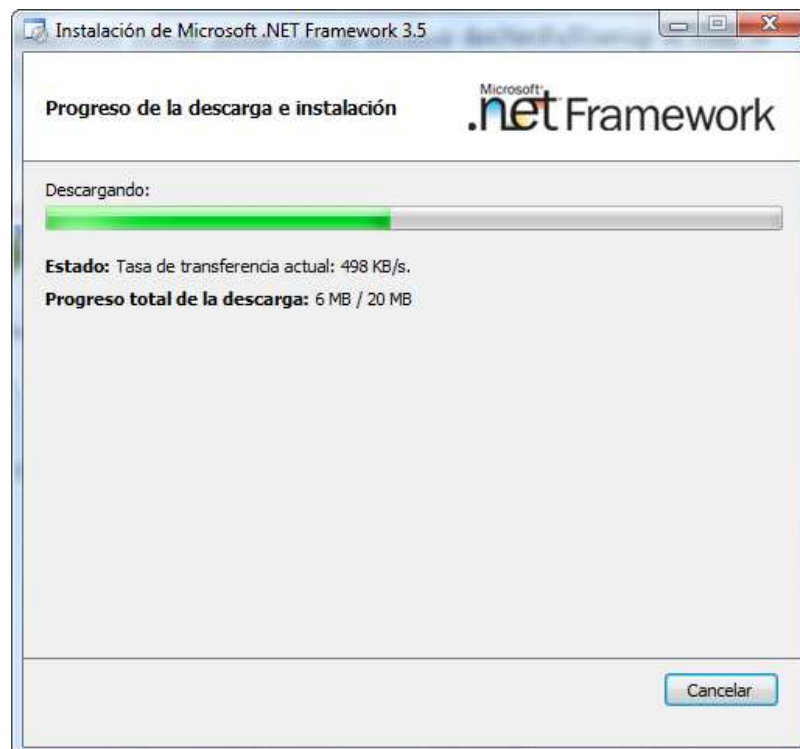


Figura 5.43 Descargando Microsoft .Net Framework 3.5

Una vez terminada la descarga comenzara a instalarse, con lo cual ya no es necesario seguir conectado a internet. Este proceso puede llegar a tardarse también varios minutos, dependerá de la capacidad de la computadora, puede variar de entre diez minutos hasta una hora.

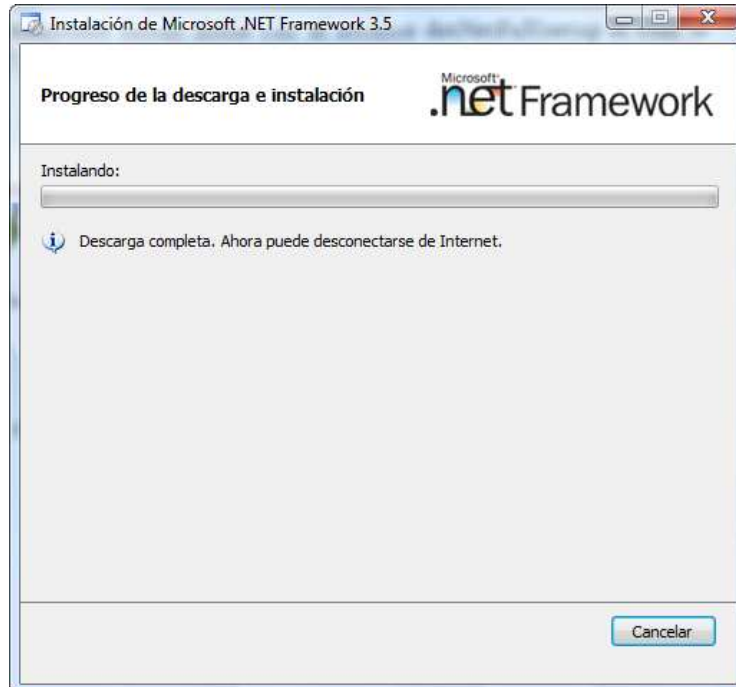


Figura 5.44 Instalando Microsoft .Net Framework 3.5

Cuando termine de instalarse nos mostrara una ventana mencionando la correcta instalación de Microsoft .Net Framework 3.5.

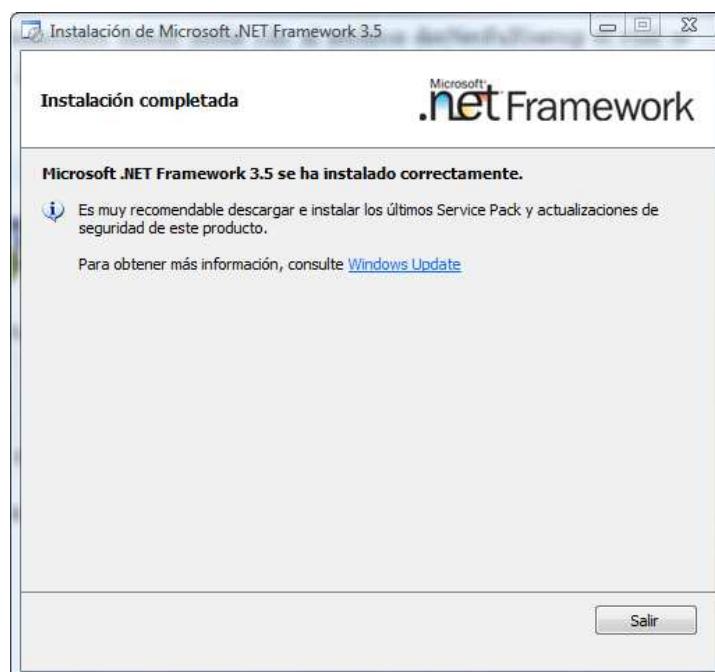


Figura 5.45 Instalación correcta

5.1.2 Manual del usuario

5.1.2.1 Manual del usuario de SCOSU v2

Comenzaremos viendo las funciones de SCOSU v2 y cómo se realizan.

5.1.2.1.1 Antes de entrar al sistema

Antes de acceder a SCOSU v2 es necesario verificar que se encuentre correctamente configurada la computadora, es decir, que su dirección ip sea la indicada a la sala donde se encuentra y que esté conectada a internet (si el equipo no está conectado a internet el sistema mostrara un mensaje indicándolo) esto determinara si puede entrar o no al sistema, ya que toma en cuenta la dirección ip de la máquina para saber si están entrando a la sesión que les corresponde.

5.1.2.1.2 Entrando al sistema

Al ejecutar el sistema nos muestra una ventana como la de la Figura 5.46 donde se deberá de seleccionar una sala del ComboBox *Sala* (Sala1, Sala2, Sala3, Sala 4) e introducir una contraseña; cada una de las salas cuenta con contraseñas distintas y solo se podrá acceder a la correspondiente en donde se encuentre instalado SCOSU v2.



Figura 5.46 Inicio de sesión

Si introducimos los datos y damos clic en Aceptar pero la contraseña es incorrecta nos mostrara el siguiente mensaje:

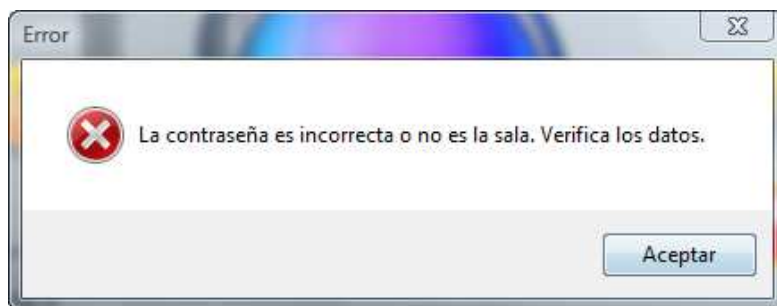


Figura 5.47 Mensaje de error

Si la contraseña si le pertenece a la opción seleccionada, pero sigue mostrando el mensaje anterior, quiere decir que está tratando de entrar a una sala que no le corresponde (véase 5.1.2.1.1 *Antes de entrar al sistema*).

Una vez colocados los datos correctos, damos clic en el botón *Aceptar* y entraremos a la sala, la cual dependerá de donde se encuentre instalado SCOSU v2. Si queremos salir del sistema damos clic en *Salir*.

5.1.2.1.3 Las salas

SCOSU v2 cuenta con 4 salas las cuales trabajan de manera similar, cambiando únicamente en su aspecto, es por eso que para fines de este manual utilizaremos la Sala 3 para explicar su funcionamiento.

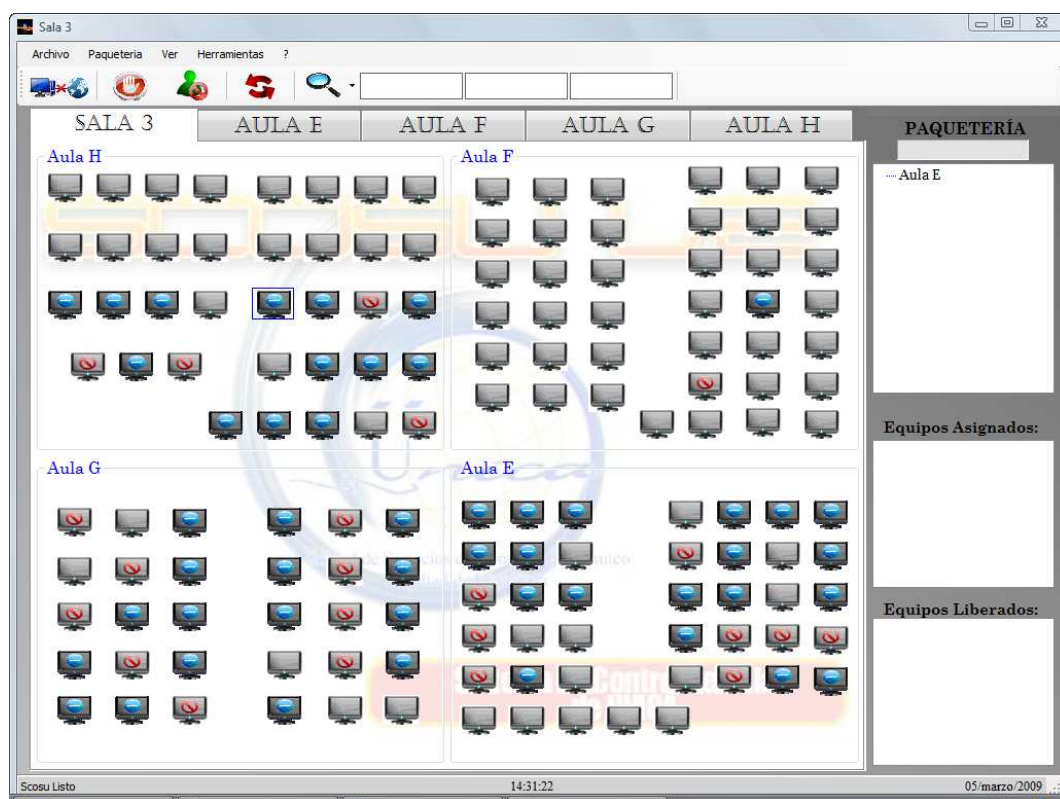


Figura 5.48 Sala 3

5.1.2.1.3.1 Partes de la sala

A. *Las aulas*: La siguiente figura muestra como se representan las aulas dentro de las salas, la primera pestaña muestra una vista general del estado en que se encuentran las computadoras de todas las aulas, excepto la Sala 4 por que solo tiene un aula y no es necesario colocarle una vista previa; las siguientes pestañas representan cada aula que contiene la sala con su respectivo número de computadoras.



Figura 5.49 Aulas de la Sala 3

B. *Paquetería*: Muestra el software disponible dependiendo del aula donde se encuentre y del tipo de sistema operativo seleccionado.

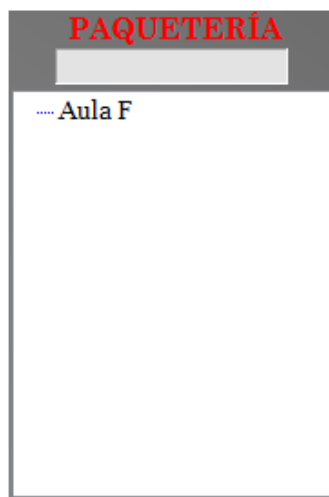


Figura 5.50 Paquetería

- C. *Lista de Equipos*: La lista de equipos asignados almacena las últimas diez computadoras, para llevar un mejor control de las mismas y así evitar confusiones, por otra parte la lista de equipos liberados almacena las últimas diez computadoras que fueron liberadas.

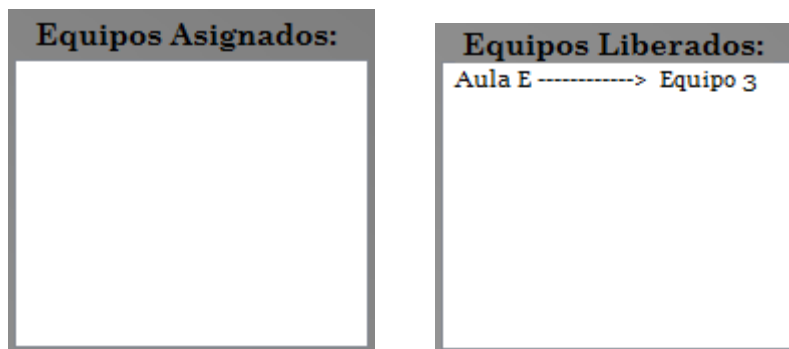


Figura 5.51 Listas de equipos Asignados y liberados

- D. *Barra de menús*: Muestra varias funciones que puede realizar SCOSU v2, desde sancionar un alumno, hasta poder comunicarse con el administrador del sistema.

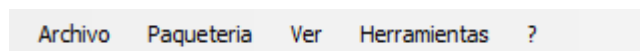


Figura 5.52 Barra de menús

- E. *Barra de herramientas*: Contiene los accesos directos a las funciones más comunes de la barra de menús.



Figura 5.53 Barra de herramientas

- F. *Barra de estado*: Contiene información del estado del sistema, la hora y la fecha actual.



Figura 5.54 Barra de estado

5.1.2.1.4 Estados de los equipos

Una computadora puede tener diferentes presentaciones dependiendo del estado en que se encuentre:

1. *Disponible*: Estado inicial del equipo de computo, la cual se puede asignar al alumno o cambiarla a un estado diferente.
2. *Ocupada*: Estado en que se coloca una computadora después de que se asigno a un alumno.
3. *Descompuesta*: Se utiliza cuando un equipo no funciona y por lo tanto no se puede asignar a un alumno.
4. *Mantenimiento*: En ocasiones no se podrá asignar un equipo a un alumno y no es porque se encuentre descompuesta, si no porque le están dando mantenimiento, en esas situaciones se utiliza este estado.
5. *Curso*: Las salas no solo se utilizan para que los alumnos realicen tareas escolares, sino también para impartir curso a los prebecarios o a los becarios (curso interno) en esos momentos se utiliza este estado para indicar que los equipos están siendo usados para dar curso.
6. *Reparación*: Un equipo cambia a estado de Reparación cuando en un principio estuvo en estado Descompuesta pero ya la están reparando.
7. *Impresora*: Aunque se cuenta con equipos para que los alumnos puedan imprimir, en ocasiones no son suficientes o no se cuenta con el espacio suficiente en la sala para colocar dichos equipos, es por eso que se tiene la necesidad de tomar una computadora de las aulas para impresiones, y este estado se encarga de indicar que equipo fue tomado para esa tarea.

En la siguiente tabla se muestran los diferentes estados de las computadoras:

Estado	Icono
1. Disponible	
2. Ocupada	
3. Descompuesta	
4. Mantenimiento	
5. Curso	
6. Reparación	
7. Impresora	

Tabla 5.1 Diferentes estados de los equipos de cómputo

5.1.2.1.5 Asignación de equipo

El equipo de salas de UNICA solo podrá ser asignado a los alumnos de la Facultad de Ingeniería, para ello será necesario que presenten su credencia.

Para asignar un equipo al alumno damos clic en alguna computadora que se encuentre en estado Disponible (5.1.2.1.4 *Estados de los equipos*) y nos mostrara la venta de la siguiente figura:



Figura 5.55 Ventana de asignación de equipo

Nos muestra los datos del alumno, así como los que el sistema necesita para su asignación, además de la fecha y el modulo en el que se encuentra. Colocamos el número de cuenta del alumno que debe ser de nueve dígitos, en la caja de texto correspondiente a *Número de Cuenta* y damos *Enter*. Si el número de dígitos colocados es menor a nueve, nos muestra un mensaje indicando nuestro error.



Figura 5.56 Número de cuenta incompleto

Ahora bien si el número de cuenta que se introdujo no es correcto el sistema muestra un mensaje al usuario indicando que el alumno no está dado de alta.

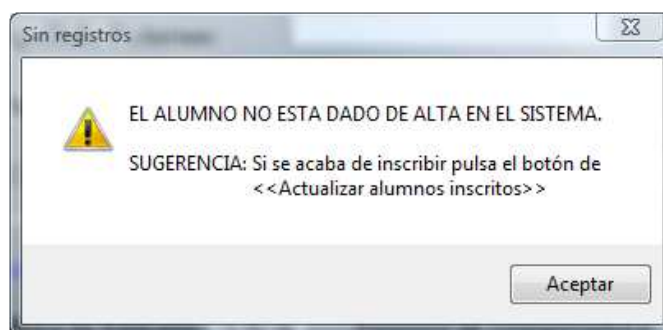


Figura 5.57 Número de cuenta incorrecto

Como podemos observar en la figura anterior, el sistema puede mostrar este mismo mensaje por que el alumno se acaba de inscribir y es necesario que se actualice la lista de alumnos inscritos en el sistema, para ello damos clic en el botón *Actualizar alumnos inscritos* de la Ventana de asignación de equipo, Figura 5.55.

Si el número de cuenta esta completo y esta dado de alta el alumno, inmediatamente después de dar Enter mostrar los datos del mismo.

A window titled "Asignación de Computadora". At the top, there are two fields: "Fecha:" with the value "12/febrero/2009" and "Modulo:" with the value "4". Below this, there are two sections. The first is "Datos del Alumno" and contains five rows of labels and text boxes: "Numero de Cuenta:" (099103008), "Nombre:" (VICTOR), "Apellido Paterno:" (DUEÁ'S), "Apellido Materno:" (TELLO), and "Carrera:" (ING. EN COMPUTACIÓN). The second section is "Datos del Sistema" and contains three rows of labels and text boxes: "Hora de Entrada:" (15:40:46), "Numero de computadora:" (44), "Hora de Salida:" (17:40:46), "Sala:" (3), and "Aula:" (F). At the bottom, there are two buttons: "Actualizar alumnos inscritos" and "Cancelar". Four red arrows point from the text "Datos del alumno" to the text boxes for Name, Paternal Surname, Maternal Surname, and Career.

Figura 5.58 Ventana de asignación con los datos del alumno

Una vez que los muestra volvemos a dar Enter. Si el alumno está sancionado muestra un mensaje como el de la Figura 5.59 indicando su nombre, el motivo de su sanción y la fecha en que podrá volver a entrar a la sala, por lo que no se le podrá asignar computadora, así que el sistema inmediatamente cierra la ventana de asignación y regresa a la sala correspondiente.

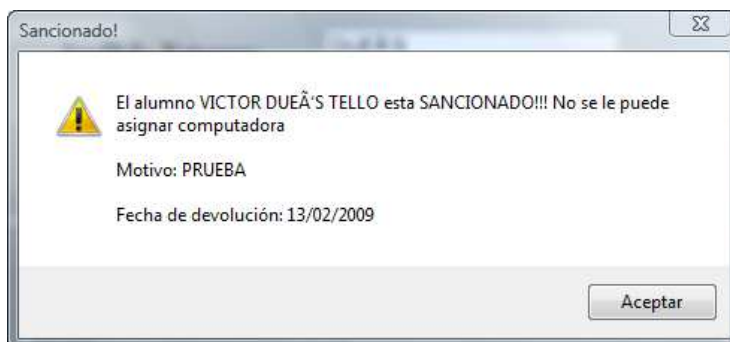


Figura 5.59 Alumno sancionado

Por el contrario si después de que nos muestra los datos del alumno damos Enter y el alumno NO está sancionado, se cierra la ventana de asignación de equipo y la computadora seleccionada cambia su estado a Ocupada cambiando también su presentación, así mismo nos muestra el numero de la maquina y la hora a la que entro el alumno, Figura 5.60.



Figura 5.60 Computadora recién asignada

5.1.2.1.5.1 Alumno duplicado

Un alumno solo podrá utilizar una computadora dentro de la sala, es por eso que en caso de que quiera utilizar otro equipo sin a ver dejado el anterior el sistema lo indica, mediante un mensaje mencionando que es necesario desocupar la computadora anterior antes de asignar la nueva, cerrando la ventana de asignación y regresando a la sala correspondiente, esto sucede después de que nos muestra los datos del alumno, damos Enter y el alumno NO está sancionado,

Por ejemplo en la siguiente figura el alumno Víctor Dueñas Tello ya estaba ocupando la computadora 115 y quiso utilizar otro equipo sin a ver dejado el que ya estaba utilizando.

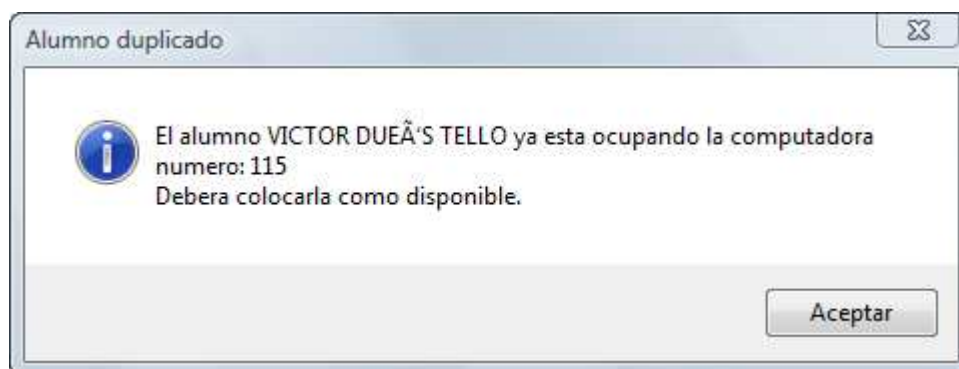


Figura 5.61 Alumno duplicado

5.1.2.1.6 Tiempo de uso y renovación de tiempo

El tiempo de uso con el que cuenta el alumno es de dos horas. Si queremos ver cuánto tiempo le falta para que termine su tiempo de uso, colocamos el puntero del mouse sobre una computadora que se encuentre en estado Ocupada e inmediatamente nos mostrara en horas y minutos, cual es su tiempo restante.



Figura 5.62 Tiempo de uso restante

En caso de que el alumno necesite más de las dos horas asignadas y es posible otorgárselo (la decisión dependerá del usuario) damos clic derecho sobre la computadora donde se encuentre el alumno, se desplegara un menú del cual seleccionaremos *Renovar tiempo* (Figura 5.63), con esto se cambiara automáticamente la hora de entrada y se reasignaran otras dos horas de uso.

El horario de servicio de las salas de UNICA es de nueve de la mañana a ocho y media de la noche, por lo que SCOSU v2 solo funcionara en ese lapso.

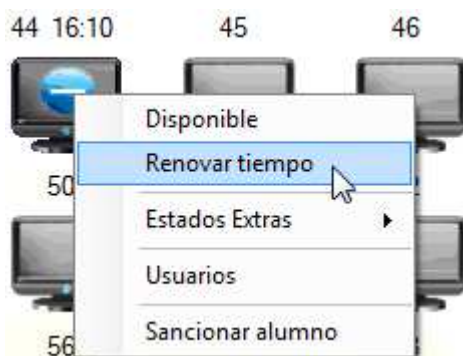


Figura 5.63 Renovar tiempo

5.1.2.1.7 Estado Disponible y Estados Extras

5.1.2.1.7.1 Colocando disponible el equipo

Después de que un alumno deja un equipo es necesario pasarlo del estado Ocupada a el estado Disponible para que se pueda volver a asignar, para ello damos clic derecho sobre la computadora que el alumno dejo y nos mostrara un menú, donde seleccionaremos la opción *Disponible*, Figura 5.64.

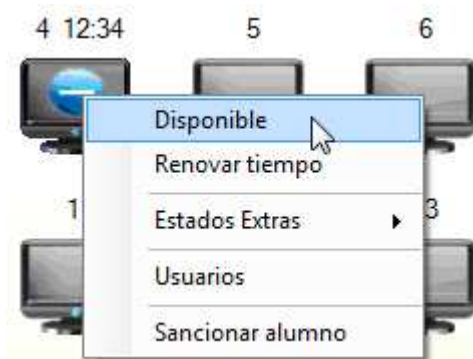


Figura 5.64 Cambiando a estado Disponible

Enseguida se cambia el estado de la computadora y su presentación a Disponible, quitando la hora a la que entro el alumno y dejando únicamente el número del equipo. Si se selecciona la opción *Disponible* en una maquina que ya se encuentra en ese estado el sistema lo ignora y no realiza ninguna acción.

Existe otro caso en el que el equipo cambia a estado Disponible y es cuando se termina su tiempo de uso, cuando eso ocurre el sistema cambia automáticamente el estado de dicho equipo a Disponible quitando la hora a la que entro el alumno y dejando únicamente el número del equipo.

5.1.2.1.7.2 Estados Extras

Se utilizan cuando la computadora va a tomar otros estados diferentes a Ocupada o Disponible, es decir, cuando se encuentra descompuesta, en mantenimiento, en reparación o se tomara como una impresora, Figura 5.65.

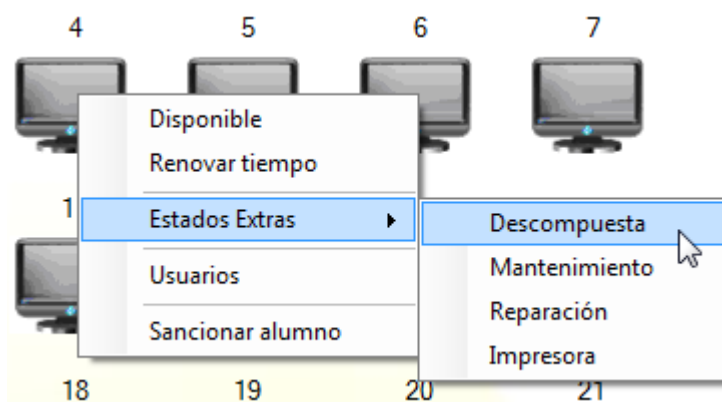


Figura 5.65 Estados extras

Para colocar una maquina en estado extra, damos clic derecho sobre dicha maquina, seleccionamos *Estados Extras* y de ahí el estado en que se encontrara la computadora.

- a) *Descompuesta*: Cuando seleccionamos este estado nos aparece una ventana donde deberemos de especificar el problema que tiene (Figura 5.66) y dar clic en *Aceptar*, sino lo hacemos se quedara con la leyenda *Escribe el problema*. Si damos clic en *Cancelar* cierra la ventana y cancela la acción. Después de escribir el problema y dar clic en *Aceptar* la computadora cambia de estado y de presentación a *Descompuesta*, Figura 5.69.

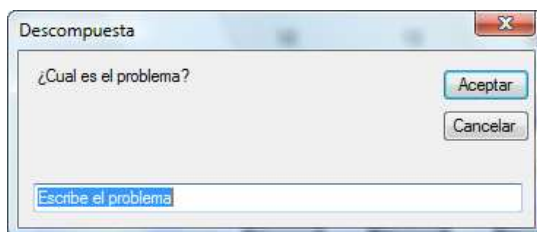


Figura 5.66 Descripción del problema

- b) *Mantenimiento*: Al dar clic en esta opción nos muestra una ventana que tiene como titulo *Mantenimiento* donde nos dice el número de la computadora seleccionada, además de mostrarnos un calendario donde deberemos de elegir la fecha de término del mantenimiento, Figura 5.67, una vez elegida damos clic en *Aceptar*, si la fecha es inferior a la actual el sistema muestra una advertencia indicando el error cometido, Figura 5.68, ahora bien si no seleccionamos ninguna fecha se utilizara la actual.

Entonces si la fecha es correcta y damos clic en *Aceptar*, cierra la ventana de mantenimiento y cambia el estado y presentación de la computadora a *Mantenimiento*, Figura 5.69. Si damos clic en *Cancelar* cierra la ventana y cancela la acción.



Figura 5.67 Ventana de mantenimiento



Figura 5.68 Advertencia por seleccionar una fecha inferior

- c) *Reparación*: Con este estado lo primero que nos muestra es una ventana que tiene como título *Reparación* donde nos dice el número de la computadora seleccionada, además de mostrarnos un calendario donde deberemos de elegir la fecha de término del mantenimiento similar al de la Figura 5.67, una vez elegida damos clic en *Aceptar*, si la fecha es inferior a la actual el sistema muestra una advertencia indicando el error cometido, como el de la Figura 5.68, ahora bien si no seleccionamos ninguna fecha se utilizara la actual.

Entonces si la fecha es correcta y damos clic en *Aceptar*, cierra la ventana de reparación y cambia el estado y presentación de la computadora a *Reparación*, Figura 5.69. Si damos clic en *Cancelar* cierra la ventana y cancela la acción.

- d) *Impresora*: Si elegimos esta opción quita el numero del equipo seleccionado y en caso de que lo tuviera, la hora de entrada del alumno y cambia su estado y presentación a *Impresora*, Figura 5.69.



Figura 5.69 Diferentes estados

Para volver a su estado original (*Disponible*) a las maquinas que se encuentren en cualquier estado extra, se seguirá el mismo procedimiento del apartado 5.1.2.1.7.1 *Colocando disponible el equipo*. Cabe mencionar que se puede cambiar el estado de un equipo no importando en cual se encuentre, con excepción de *Ocupada*, para cambiarlo a ese estado tiene que estar *Disponible*.

5.1.2.1.7.3 El mouse y los estados extras

Si intentamos asignar una computadora que se encuentre en algún estado extra, es decir, damos clic sobre ella, nos mostrara un mensaje de advertencia indicándonos su estado, a excepción de Impresora, con lo cual será imposible asignarla.

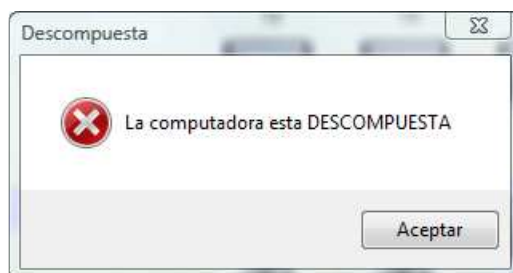


Figura 5.70 Advertencia por estado extra

Ahora bien, si pasamos el puntero del mouse por encima de una computadora en estado extra, nos mostrara los detalles del mismo, para Descompuesta indica cual es el motivo por el que se encuentra así, Figura 5.71, para Mantenimiento y Reparación señala la fecha tentativa en que se terminara su estado. Para el caso de la impresora no sucede ninguna acción.



Figura 5.71 Detalles del estado extra

5.1.2.1.8 Lista de Equipos Asignados y Equipos Liberados

Cada que un equipo es asignado al alumno, éste se agrega a la lista de *Equipos Asignados* donde se muestran las diez últimas computadoras asignadas con sus respectivas aulas, es decir, se agrega en la primera posición de la lista y se quita la que se encuentre en la decima posición, Figura 5.72, con el fin de que se lleve un mejor control de asignación.

En la lista de *Equipos Liberados* se encuentran los diez últimos que los alumnos han dejado o que el sistema ha puesto como Disponible (véase 5.1.2.1.7.1 *Colocando disponible el equipo*), Figura 5.72, de la misma manera agrega el equipo en la primera posición de la lista y se quita el que se encuentre en la decima posición, además si la computadora recién agregada en la lista de *Equipos Liberados* se encuentra en la de *Equipos Asignados* se quita de dicha lista.

Equipos Asignados:		Equipos Liberados:	
Aula H ----->	Equipo 118	Aula H ----->	Equipo 118
Aula H ----->	Equipo 117	Aula H ----->	Equipo 117
Aula H ----->	Equipo 116	Aula H ----->	Equipo 116
Aula H ----->	Equipo 115	Aula H ----->	Equipo 115

Figura 5.72 Listas de Equipos Asignados y Liberados

5.1.2.1.9 Usuarios de las computadoras

En el transcurso del día varios alumnos utilizan una misma computadora, SCOSU v2 tiene la opción de mostrarnos los alumnos que a lo largo del día han utilizado los equipos, para ello damos clic derecho sobre el equipo del cual queremos obtener esa información y seleccionamos la opción *Usuarios*.



Figura 5.73 Opción Usuarios

Si no se ha asignado ni una sola vez el sistema nos lo hace saber mediante un mensaje como el de la siguiente figura:

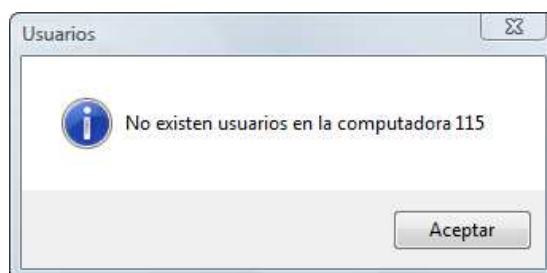


Figura 5.74 Sin usuarios en la computadora 115

En caso contrario nos mostrara una ventana con la lista de alumnos, indicando su número de cuenta, nombre completo, hora de entrada, hora de salida, número de la computadora y el aula donde se encuentra, Figura 5.75.

Usuarios de la computadora 24

	Num. Cuenta	Nombre	Entrada	Salida	Computadora	Aula
▶	304304299	JUAN MANUEL MENDEZ ROSALES	10:06:24	11:14:29	24	E
	304225738	OLEGARIO OSWALDO MUÑOZ RAMIREZ	11:14:38	11:28:43	24	E
	302229084	LEONARDO MONTES DE OCA HUERTA	11:28:49	12:00:09	24	E
	306199853	RAFAEL GONZALEZ PUIG	12:01:50	13:39:00	24	E
	304113631	ANTONIO FALCON BENITEZ	13:44:56	14:06:34	24	E
	305254616	SERGIO FABIAN GARCIA RUIZ	14:06:40	16:06:40	24	E
*						

Regresar

Figura 5.75 Lista de usuarios de la computadora seleccionada

Dando clic en el botón *Regresar* cierra la ventana de la lista de usuarios y regresa a la sala correspondiente.

5.1.2.1.10 Destinar sala

Como se menciono anteriormente la aulas de las salas de UNICA también se utilizan para poder impartir curso en ellas, es por eso que SCOSU v2 cuenta con la opción de destinar las aulas para dicha tarea, además de tener otra opción donde se puede destinar el aula para mantenimiento.

5.1.2.1.10.1 Aula en Curso

Para apartar un aula de la sala donde nos encontramos para un curso, seleccionamos menú Archivo → Destinar Sala → Curso.

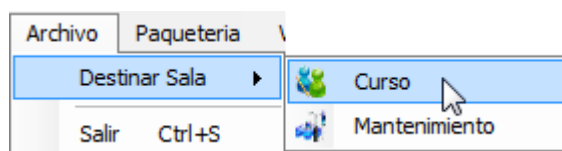


Figura 5.76 Curso

Mostrándonos la siguiente ventana:



Apartar sala para curso

Nombre del Curso:

Fecha y Hora de Inicio: lunes . 16 de febrero de 2009 13:56:23

Fecha y Hora de Termina: lunes . 16 de febrero de 2009 13:56:23

Sala 3 Aula:

Aceptar Cancelar

Figura 5.77 Ventana para apartar aula para curso

En donde estableceremos todos los datos necesarios para poder apartar el aula, datos como el nombre del curso, fecha de inicio y fin, hora de inicio y fin y el aula seleccionada.

5.1.2.1.10.1.1 Nombre del curso

En un principio vemos que el botón *Aceptar* no se encuentra habilitado, este se habilitara en el momento que todos los datos especificados sean correctos. Lo primero que haremos es introducir el nombre del curso.

5.1.2.1.10.1.2 Fecha y hora de inicio

Una vez colocado el nombre del curso especificamos las fecha y hora de inicio, la fecha deberá ser igual o mayor que la actual, ya que si se selecciona una fecha inferior inmediatamente cambia a la presente. Si se selecciona como fecha de inicio de curso a la actual la hora de inicio deberá de ser por lo menos una hora mayor que la hora actual (esto con el fin de dar tiempo para preparar la sala) en caso de no ser así, en el momento que se le quita el *Focus* muestra un mensaje de advertencia, Figura 5.78.

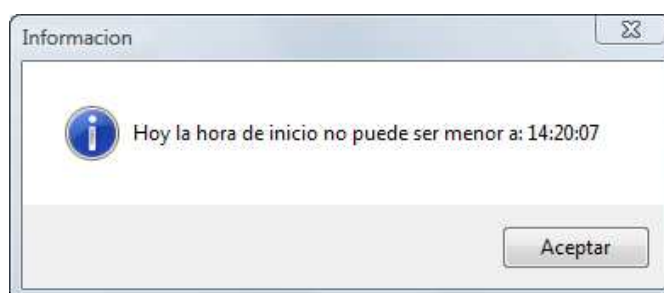


Figura 5.78 Advertencia por seleccionar una hora menor a la actual

Si la fecha de inicio seleccionada es mayor a la actual, se podrá seleccionar como hora de inicio cualquiera que se encuentre en el intervalo de servicio de las salas de UNICA, que es de nueve de la mañana a ocho y media de la noche.

5.1.2.1.10.1.3 Fecha y hora de término

La fecha de término del curso deberá ser mayor o igual que la fecha de inicio, en caso contrario el sistema muestra un mensaje de advertencia indicando el error cometido, Figura 5.79, de la misma manera la hora de término del curso tendrá que ser por lo menos una hora mayor que la hora de inicio, en caso de que no se cumplan con estas condiciones se mostraran distintos mensajes de advertencia dependiendo del error cometido, Figura 5.80. La hora de término del curso deberá de ser en el intervalo de servicio de las salas de UNICA, que es de nueve de la mañana a ocho y media de la noche.

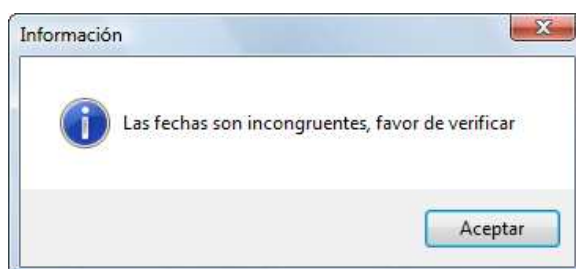


Figura 5.79 Advertencia por fecha de término menor a la fecha de inicio

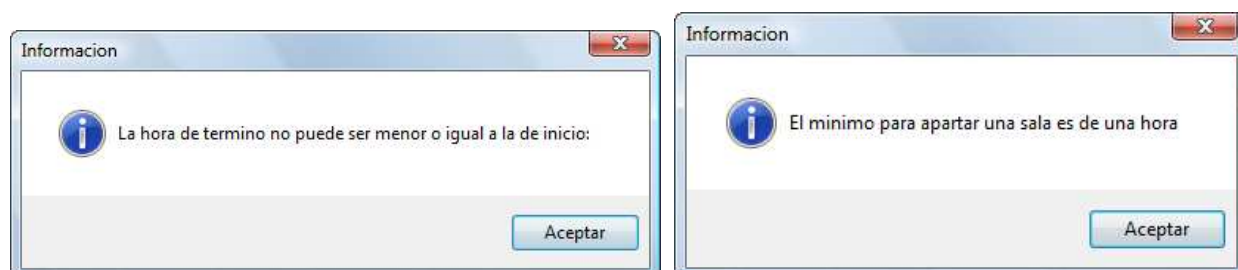


Figura 5.80 Advertencias por colocar mal la hora de término

5.1.2.1.10.1.4 Seleccionando el aula

Después de haber colocado las fechas y horas correctas seleccionaremos el aula donde se impartirá el curso, del ComboBox *Aula*, el cual se carga solo con las aulas de la sala donde se encuentre.

Una vez colocados todos los datos correctos el botón *Aceptar* se habilitara, enseguida damos clic sobre él para apartar el aula seleccionada, mostrándonos un mensaje parecido al de la Figura 5.81, inmediatamente cierra la ventana para apartar curso y regresa a la sala correspondiente. Si damos clic en el botón *Cancelar* cierra la ventana y cancela la acción.

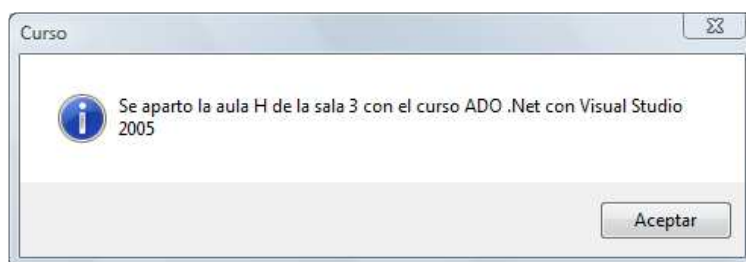


Figura 5.81 Aula apartada para el curso

5.1.2.1.10.2 Inicio y fin del curso

5.1.2.1.10.2.1 Hora de inicio del curso

Una vez llegada la fecha de inicio del curso, SCOSU v2 muestra un mensaje de aviso diez minutos antes de su hora de inicio, de que el curso está a punto de comenzar, Figura 5.82, indicando en que aula se llevará a cabo para que el persona de salas reubique a los alumnos que se encuentran ahí, dejando el aula libre para el curso.

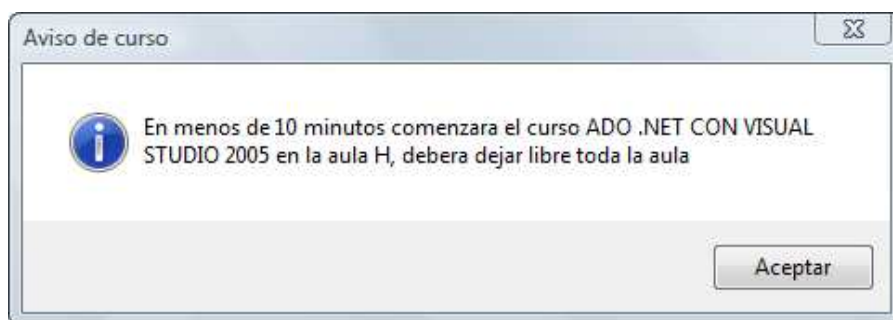


Figura 5.82 Aviso de inicio de curso

Llegada la hora el sistema manda un mensaje donde dice que aula se apartara y con qué curso, Figura 5.83, enseguida coloca automáticamente toda el aula en estado Curso, cambiando el estado y la presentación de los equipos, a excepción de los que se encuentran en algún estado extra (véase 5.1.2.1.7.2 *Estados Extras*) así mismo, coloca un icono al lado del nombre del aula (pestaña) señalando que se encuentra en curso, Figura 5.84.

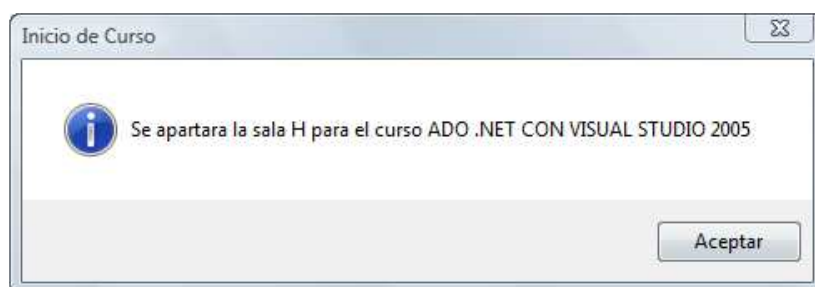


Figura 5.83 Inicio de curso

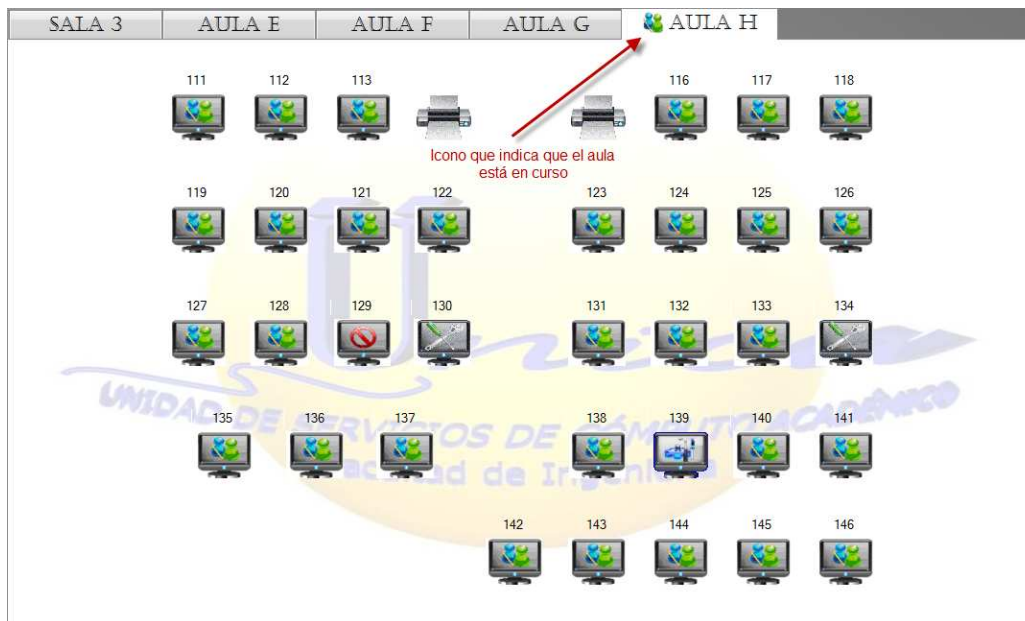


Figura 5.84 Aula en curso

5.1.2.1.10.2.2 En el curso

Ya que el aula se encuentra en curso, podemos ver los detalles del mismo colocando el puntero del mouse sobre el icono que indica que el aula está en curso, mostrándonos el nombre, la hora a la que inicio y la hora a la que va a terminar, Figura 5.85.



Figura 5.85 Detalles del curso

Si damos clic derecho sobre algún equipo en estado Curso, es decir, lo tratamos de asignar, el sistema nos mostrara un mensaje de que la computadora se encuentra en curso, haciendo imposible asignarla, Figura 5.86. Ahora bien, si damos clic derecho sobre cualquier equipo en estado Curso para que nos muestre el menú, este no aparecerá, ya que se deshabilita cuando se encuentra en este estado.

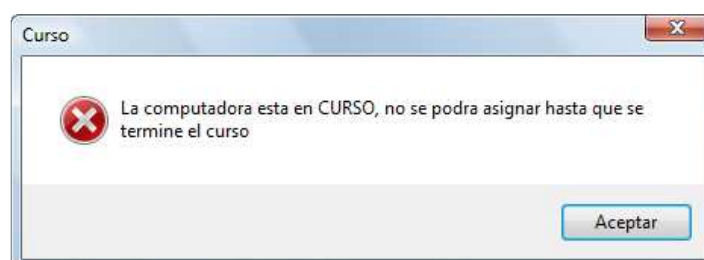


Figura 5.86 Mensaje en computadora en curso

Por lo tanto la única manera de que se quite el estado Curso de las computadoras, es que se termine el curso asignado.

5.1.2.1.10.2.3 Hora de fin del curso

Llegada la hora de fin del curso SCOSU v2 muestra un mensaje al usuario indicando que el curso ya se termino, Figura 5.87, por lo que cambiara únicamente todas las computadoras en estado Curso a Disponible, junto con la presentación de las mismas, además de quitar el icono que se encuentra al lado del nombre del aula.

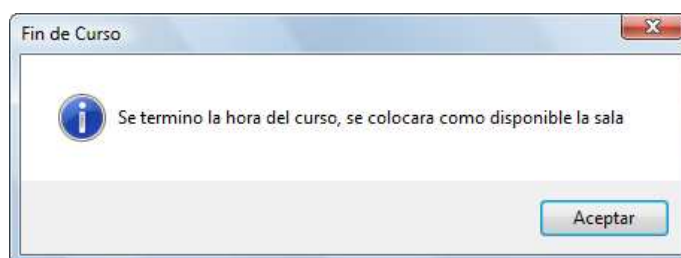


Figura 5.87 Mensaje de fin de curso

5.1.2.1.10.3 Aula en Mantenimiento

Para apartar un aula de la sala donde nos encontramos para mantenimiento, seleccionamos menú Archivo → Destinar Sala → Mantenimiento.

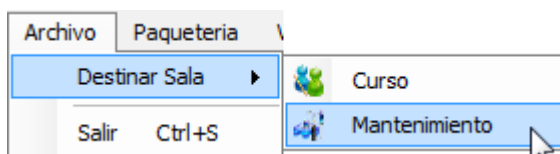


Figura 5.88 Mantenimiento

Nos mostrar la siguiente ventana

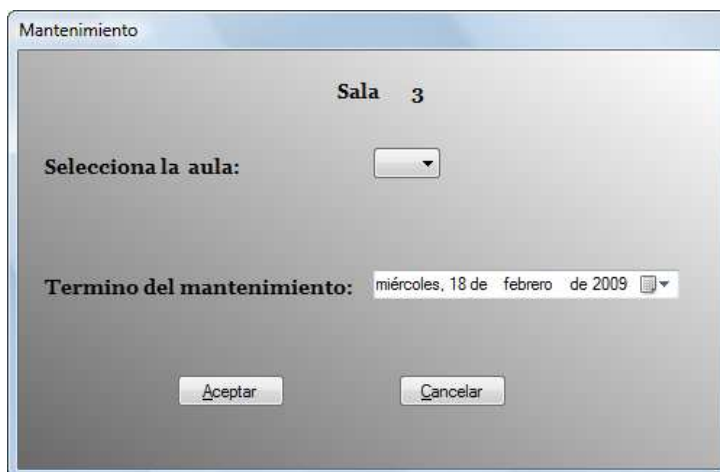


Figura 5.89 Venta de mantenimiento

En ella deberemos seleccionar el aula que pondremos en mantenimiento y la fecha tentativa de término del mismo y damos *Aceptar*, si no seleccionamos el aula el sistema nos muestra una advertencia indicando que debemos seleccionarla.

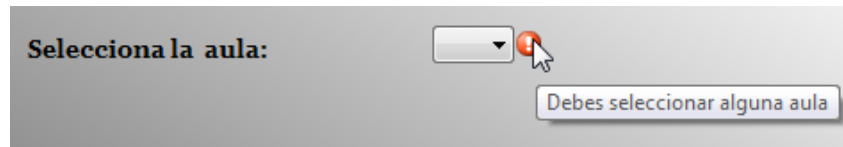


Figura 5.90 Mensaje de advertencia por no seleccionar un aula

Si la fecha seleccionada es menor a la actual SCOSU v2 muestra una advertencia indicando el error cometido.



Figura 5.91 Mensaje de advertencia por fecha incorrecta

Entonces, una vez seleccionada el aula y haber colocado la fecha correcta damos *Aceptar*, esta acción cierra la ventana y coloca todas las computadoras del aula en estado Mantenimiento, a excepción de las que se encuentran en algún estado extra (véase 5.1.2.1.7.2 *Estados Extras*), además de colocar un icono al lado del nombre del aula (pestaña) indicando que ésta se encuentra en mantenimiento, Figura 5.92.

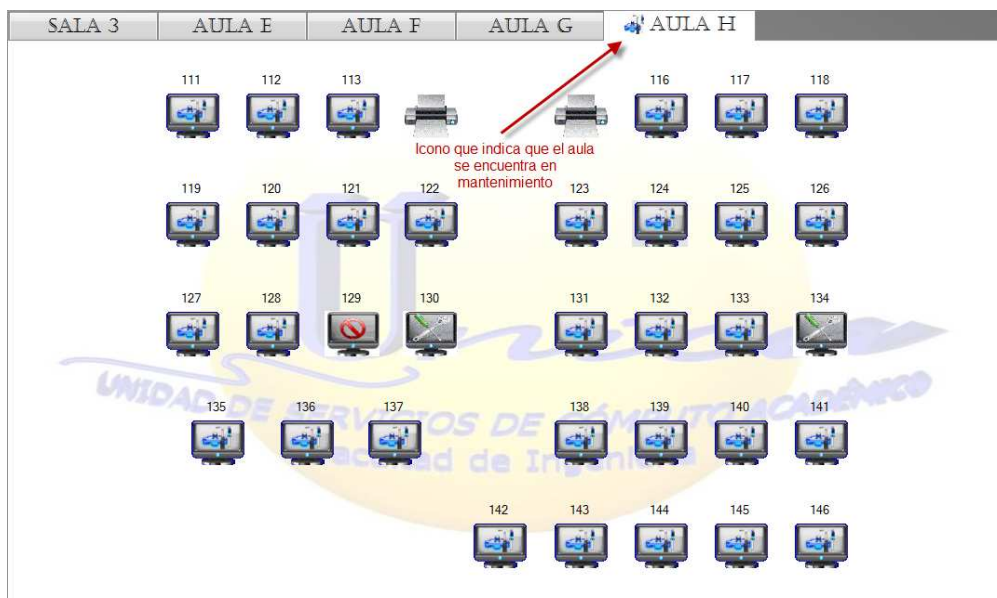


Figura 5.92 Aula en mantenimiento

Si tratamos de asignar una computadora, es decir damos clic sobre alguna de ella, el sistema nos mostrara un mensaje indicando el estado en que se encuentra, con lo cual será imposible asignarla.

Para colocar los equipos como disponibles una vez que se llegue a la fecha de término del mantenimiento o antes, si así se desea, se puede hacer de dos maneras, la primera es realizando el procedimiento del punto 5.1.2.1.7.1 *Colocando disponible el equipo* en cada una de las computadoras, la segunda es dando clic derecho sobre el icono que indica que el aula está en mantenimiento con lo cual nos aparecerá la opción *Colocar sala disponible*, Figura 5.93, la seleccionamos e inmediatamente nos cambiara únicamente las computadoras en estado Mantenimiento a Disponible así como su presentación, además de quitar el icono que se encuentra al lado del nombre del aula.

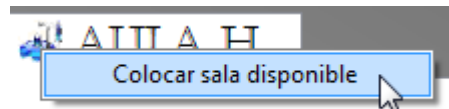


Figura 5.93 Opción del aula en mantenimiento

5.1.2.1.11 Paquetería

SCOSU v2 muestra la paquetería que contienen las computadoras de las diferentes aulas de la sala, las cuales pueden llegar a variar tanto en número como en el tipo de sistema operativo, al igual que en los software.

Para ver qué tipo de sistema operativo tienen las aulas, debemos de estar posicionados en ella, es decir, damos clic en la pestaña del aula y seleccionamos el menú *Paquetería*, con esto nos desplegara cuatro opciones que no son más que los cuatro tipos de sistemas operativos, de los cuales estarán habilitados solo los que contenga el aula donde estamos, Figura 5.94.

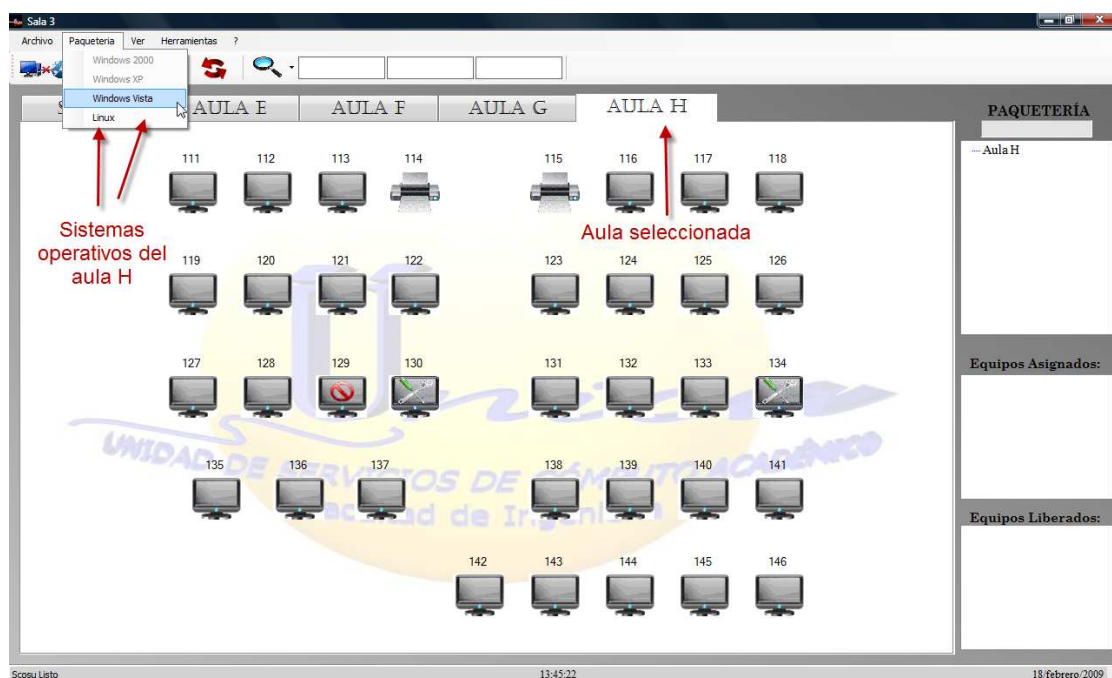


Figura 5.94 Tipos de sistemas operativos

Si el aula no tiene paquetería o estamos en la vista previa, (*ver punto 5.1.2.1.23 Vista previa*) el menú *Paquetería* no tendrá ninguna opción habilitada.

Enseguida, si queremos ver el software del sistema operativo, damos clic sobre el deseado (que tendrá que estar habilitado), el cual aparecerá en el TextBox *PAQUETERÍA*, Figura 5.95.



Figura 5.95 Paquetería seleccionada

Después damos doble clic sobre el nombre del aula del recuadro (que cambiara dependiendo donde estemos) para ver todo el software del sistema operativo seleccionado, Figura 5.96, cabe mencionar que cada que se cambie de aula (se dé clic en otra pestaña) se borrara el sistema operativo junto con su paquetería, teniendo que seleccionarlo otra vez si es que se quiere saber el software que contiene.

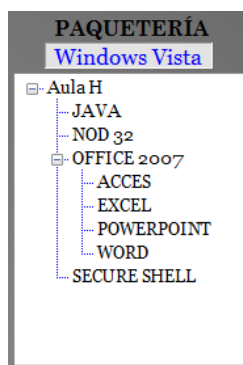


Figura 5.96 Software del sistema operativo

Si damos clic sobre el recuadro sin haber seleccionado ningún sistema operativo, SCOSU v2 muestra un mensaje indicando el error cometido, Figura 5.97.

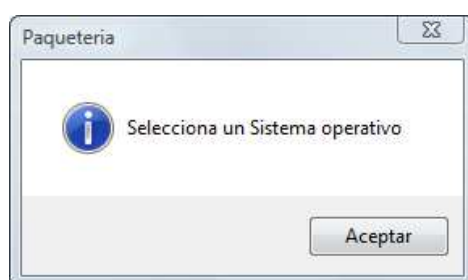


Figura 5.97

5.1.2.1.12 Ver Sancionados

Después de que un alumno es sancionado (véase el punto 5.1.2.1.14 Sancionar alumno) se guardan los datos de la sanción, tales como el número de cuenta del alumno sancionado, su nombre, carrera, el motivo de la sanción, el responsable de la sanción, fecha de la sanción y la fecha de devolución (cuando podrá volver a utilizar la sala) en una lista, Figura 5.98, la cual podemos visualizar seleccionando la opción *Sancionados* del menú *Ver*. Esta lista se puede ver desde cualquiera de las cuatro salas.

Num. Cuenta	Nombre	Carrera	Motivo	Responsable
407088199	GUSTAVO FIERRO MORENO	ING. EN COMPUTACIÓN	YOUTUBE, JUEGOS O VIDEOS	WENDY
302010431	HUMBERTO JAVIER SOLANA RIVERO	ING. EN COMPUTACIÓN	PAGS XXX	ANDUIN Y ESPE
407040957	JORGE ALEJANDRO ROMERO LOPEZ	ING. MECANICA	COLOR, REHUSO, ACETATO	ESPE
409004038	CARLOS XAVIER JIMENEZ LOPEZ	ING. PETROLERA	PRESTAMO (USUARIO)	GONZALO DEL S
306036682	ANA JESSICA TORRES PEREZ	ING. PETROLERA	PRESTAMO (DUEÑO)	GONZALO DEL S
304195048	ANTONIO DE JESUS GONZALEZ AMBROSIO	ING. EN COMPUTACIÓN	AMBOS LADOS DE LA HOJA	SEAN
306284450	ROGER EMMANUEL PORTILLO FRAGOSO	ING. EN TELECOMUNICACIONES	PARA 23 DE FEBRERO YA DEBE TRAER CREDENCIAL	DANY
305245294	ADRIAN AZTATZI SANTILLAN	ING. INDUSTRIAL	AMBOS LADOS DE LA HOJA	ESPE BUEN EST
305292274	ARMANDO JESUS HERNANDEZ ORROSTIETA	ING. EN COMPUTACIÓN	AMBOS LADOS DE LA HOJA	ESPERANZA
304327258	MARIO ALBERTO GONZALEZ BORBOLLA	ING. PETROLERA	HI5	GERARDO
303111782	DAVID DE JESUS CRUZ	ING. MECANICA	HI5	GERARDO
300266852	GIOVANNI ROBLES FLORES	ING. PETROLERA	INSTALAR O USAR MESSENGER	GERARDO

Figura 5.98 Lista de alumnos sancionados

Si queremos actualizar la lista de alumnos sancionados solo basta dar clic en el botón *Actualizar*. Si damos clic en *Regresar* cierra la lista de alumnos y regresa a la sala correspondiente.

5.1.2.1.13 Ver Cursos

Luego de que se aparta un aula para un curso se almacenan sus datos, tales como el nombre del curso, fecha de inicio y fin, hora de inicio y fin, la sala y el aula donde se impartirá en una lista, Figura 5.99. Esta lista se podrá ver en cualquiera de las cuatro salas, seleccionando la opción *Cursos* del menú *Ver*.

Curso	Fecha de inicio	Fecha de fin	Hora de inicio	Hora de fin	Sala	Aula
ADO .NET CON VISUAL STUDIO 2005	17/02/2009	20/02/2009	16:00:00	19:00:00	3	H
*						

Figura 5.99 Lista de cursos

Si damos clic en el botón *Regresar* cierra la lista de cursos y regresa a la sala correspondiente. Una vez que el curso se termine o si por alguna razón se termino antes, es obligación del jefe de la sala donde se está impartiendo eliminarlo (*véase Manual de usuario de SCOSU v2 Modulo del Jefe*), quitándolo así de la lista.

5.1.2.1.14 Sancionar Alumno

Cuando un alumno incurre en alguna falta al reglamento que tiene establecido UNICA se le sanciona, y la sanción establecida es no dejarlo utilizar las computadoras de ninguna sala durante algún tiempo, el cual dependerá de la falta que allá cometido. Existen dos formas de sancionar a un alumno, cuando está utilizando un equipo y cuando no.

5.1.2.1.14.1 Sancionar un alumno cuando NO se encuentra utilizando un equipo

Para sancionar a un alumno que no se encuentra utilizando un equipo seleccionamos la opción *Sancionar Alumno* del menú *Herramientas*, donde nos aparecerá una ventana como la de la siguiente figura, donde podemos observar que el botón *Aceptar* se encuentra deshabilitado y solo se habilitara hasta que se introduzcan todos los datos necesarios de la sanción.



The screenshot shows a window titled "Sancionados" with a subtitle "El cual dependerá de la falta que allí cometió. Salvo sea...". The window is divided into two main sections: "Datos del Sancionado" and "Datos de Sanción".

Datos del Sancionado:

- Número de Cuenta: [Empty text box]
- Nombre: [Empty text box]

Datos de Sanción:

- Sala: 3
- Motivo de la Sanción: [Dropdown menu]
- Fecha de Sanción: jueves, 19 de febrero de 2009 [Calendar icon]
- Fecha de Devolución: jueves, 19 de febrero de 2009 [Calendar icon]
- Nombre del Responsable: [Empty text box]

At the bottom of the window, there are two buttons: "Aceptar" (disabled) and "Cancelar".

Figura 5.100 Ventana para sancionar alumno

5.1.2.1.14.1.1 Datos del Sancionado

En ella lo primero que tenemos que hacer es introducir el número de cuenta del alumno en el TextBox Número de Cuenta (el cual solo acepta números) y dar *Enter*, si no introducimos ningún número el sistema nos muestra una advertencia indicando el error cometido, Figura 5.101.



Número de Cuenta:

Nombre:

Debes introducir el numero de cuenta

Figura 5.101

De la misma manera si el número de cuenta esta incompleto el sistema vuelve a mostrar un mensaje de advertencia indicando el error cometido, Figura 5.102.



Número de Cuenta: 47458854

Nombre:

Numero de cuenta incompleto

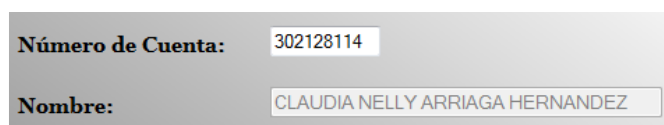
Figura 5.102

Ahora bien si el número de cuenta introducido está mal escrito o no esta dado de alta, SCOSU v2 muestra un mensaje al usuario indicando el error, Figura 5.103.



Figura 5.103

Si el número de cuenta es correcto y damos *Enter* nos aparece el nombre completo del alumno en el TextBox *Nombre*, Figura 5.104.




Número de Cuenta: 302128114

Nombre: CLAUDIA NELLY ARRIAGA HERNANDEZ

Figura 5.104

5.1.2.1.14.1.2 Datos de la Sanción

Del ComboBox *Motivo de la Sanción* seleccionamos la sanción que más se adecue a la falta cometida por el alumno. Dependiendo de la sanción que se seleccione automáticamente se da el tiempo de sanción, es decir, el tiempo que tendrá que esperar el alumno antes de que pueda volver a entrar a cualquier sala, Figura 5.105.



Sala: 3

Motivo de la Sanción: Mas de 15 hojas

Fecha de Sanción: jueves , 19 de febrero de 2009

Fecha de Devolución: jueves , 05 de marzo de 2009

Figura 5.105 Tiempo de sanción

Los tiempos de sanción son establecidos de acuerdo a la siguiente tabla.

Tabulador de Sanciones			
Sanción	Tipo de Sanción	Tiempo de Sanción	Aviso
Impresión	Más de 15 hojas	10 días hábiles	Al primer aviso, a los alumnos del segundo semestre en adelante, primer semestre al segundo
	Ambos lados de la hoja	10 días hábiles	Al primer aviso, a los alumnos del segundo semestre en adelante, primer semestre al segundo
	Color, Rehúso, Acetato	10 días hábiles	Al primer aviso, a los alumnos del segundo semestre en adelante, primer semestre al segundo
	Lo anterior c/Dolo	Resto del Semestre	Al primer aviso, a los alumnos del segundo semestre en adelante, primer semestre al segundo
	Tesis		Avisar al encargado
	Manual	1 mes	Al primer aviso, a los alumnos del segundo semestre en adelante, primer semestre al segundo
	Series Completas de más de 15 hojas	10 días hábiles	Al primer aviso, a los alumnos del segundo semestre en adelante, primer semestre al segundo
Credencial	Préstamo (dueño y usuario)	3 meses	Al primer aviso, a los alumnos del segundo semestre en adelante, primer semestre al segundo
Internet	Youtube, Juegos, videos	15 días hábiles	Al primer aviso, a los alumnos del segundo semestre en adelante, primer semestre al segundo
	Págs. XXX	1 mes	Al primer aviso, a los alumnos del segundo semestre en adelante, primer semestre al segundo
Quemadores	Música y Software	15 días hábiles	Al primer aviso, a los alumnos del segundo semestre en adelante, primer semestre al segundo

Programas	Instalar programas	1 semana	Al primer aviso, a los alumnos del segundo semestre en adelante, primer semestre al segundo
Personales	Desorden y comer	1 semana	Al primer aviso, a los alumnos del segundo semestre en adelante, primer semestre al segundo
Messenger	Instalar y usarlo en cualquier pág. De internet	5 días hábiles	Al primer aviso, a los alumnos del segundo semestre en adelante, primer semestre al segundo

Tabla 5.2 Tabulador de sanciones

Existen dos casos especiales donde el tiempo de sanción no se coloca automáticamente, ya que dependerá del criterio del usuario, y son las opciones de *Tesis* y *Otros*. Si seleccionamos *Tesis* nos mostrara un mensaje como el de la Figura 5.106, donde indica al usuario que le avise al encargado (jefe de sala) para que tome la decisión correcta, enseguida nos habilita la fecha de devolución donde seleccionaremos la más adecuada.



Figura 5.106

Si seleccionamos la opción *Otros* mostrara una ventana (Figura 5.107) donde colocaremos el motivo de la sanción, esta opción se utiliza cuando la sanción cometida por el alumno no se encuentra dentro de la lista que tiene el sistema.

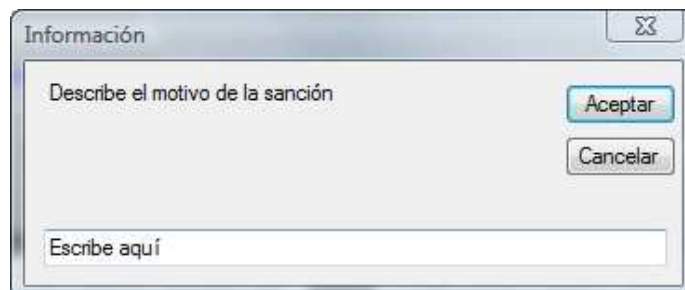


Figura 5.107 Motivo de la sanción

Si damos *Aceptar* se guarda el motivo y habilita la fecha de devolución donde seleccionaremos la más adecuada, si damos *Cancelar* se cierra la ventana y cancela la acción.

Una vez seleccionado el motivo de la sanción se debe de introducir el nombre del responsable, es decir, el nombre de la persona que sancionó al alumno, en el TextBox *Nombre del Responsable*, Figura 5.108.

Figura 5.108 Datos completos de la sanción

Después de colocar los datos completos del alumno y de la sanción se habilita el botón *Aceptar*, damos clic en él para establecer la sanción, inmediatamente cierra la ventana y regresa a la sala correspondiente.

Los datos del alumno sancionado se pueden ver en la lista de alumnos sancionados (*véase el punto 5.1.2.1.12 Ver Sancionados*).

5.1.2.1.14.2 Sancionar un alumno cuando se encuentra utilizando un equipo

Para sancionar un alumno cuando se encuentra utilizando un equipo, damos clic derecho sobre la computadora donde se encuentre y seleccionamos del menú la opción *Sancionar alumno*, Figura 5.109, cabe mencionar que la computadora seleccionada debe de estar en estado Ocupada o de lo contrario el sistema nos mostrara un mensaje indicando que no se encuentra nadie ocupando el equipo.

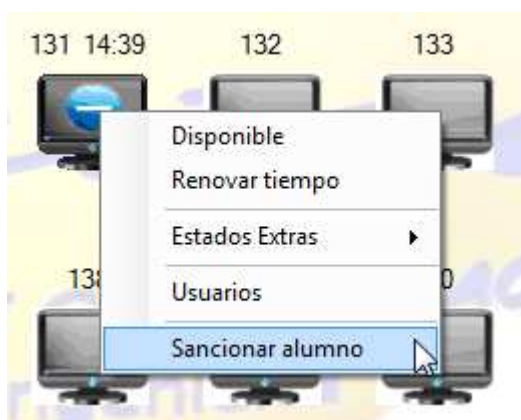


Figura 5.109 Sancionar alumno

Enseguida nos muestra la siguiente ventana.



The screenshot shows a window titled "Sancionados" with two main sections: "Datos del Sancionado" and "Datos de Sanción".

- Datos del Sancionado:**
 - Número de Cuenta: 99103008
 - Nombre: VICTOR DUEÁ'S TELLO
- Datos de Sanción:**
 - Sala: 3
 - Motivo de la Sanción: (dropdown menu)
 - Fecha de Sanción: jueves, 19 de febrero de 2009
 - Fecha de Devolución: jueves, 19 de febrero de 2009
 - Nombre del Responsable: (text field)

At the bottom, there are two buttons: "Aceptar" and "Cancelar".

Figura 5.110 Ventana para sancionar alumno

Como podemos observar nos coloca automáticamente el número de cuenta y el nombre del alumno que se encuentra en ese momento en la computadora seleccionada, los cuales no podremos cambiar por ningún motivo.

Dado que ya nos brinda los datos del alumno que queremos sancionar solo nos queda proporcionar los datos de la sanción, llevando a cabo el mismo procedimiento especificado en el punto 5.1.2.1.14.1.2 *Datos de la Sanción*.

5.1.2.1.14.3 *Termino de la sanción*

Una vez cumplido su tiempo señalado el sistema informa del mismo cuando se accede a cualquier sala de SCOSU v2 y retira automáticamente la sanción del alumno, eliminando sus datos de la lista de sancionados (*véase el punto 5.1.2.1.12 Ver Sancionados*) y permitiendo así su entrada a cualquiera de las salas de UNICA, Figura 5.111.

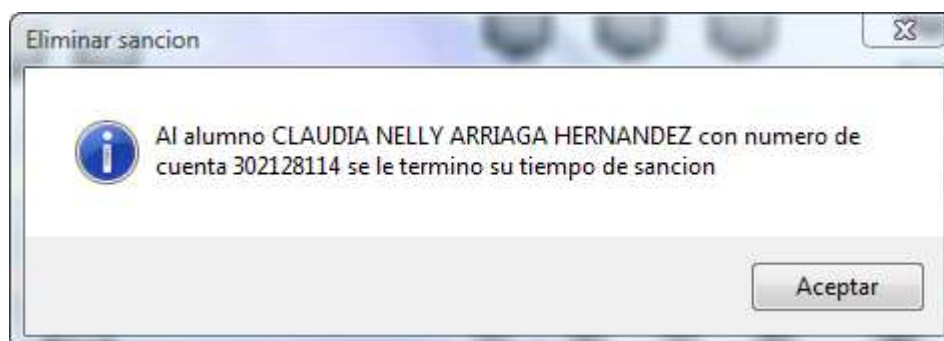


Figura 5.111 Tiempo de sanción cumplido

Existe otra forma de retirar la sanción al alumno antes de que se termine la misma, esta acción la realiza el jefe de la sala mediante SCOSU v2 Modulo del jefe (*véase Manual del usuario de SCOSU v2 Modulo del Jefe*), el cual determinara si las justificaciones del alumno son suficientes para retirarle la sanción antes de tiempo.

5.1.2.1.15 Restaurar Sistema

En ciertas ocasiones el sistema puede llegar a cerrarse ya sea por error del mismo, error humano o por una falla de la computadora donde se encuentra instalado, para esas ocasiones SCOSU v2 cuenta con una opción que restaura los estados todas las computadoras, es decir, les coloca el estado en que se encontraban antes de que el sistema se cerrara.

Para utilizar esta función después de que SCOSU v2 se cierra, volvemos a entrar a la sala correspondiente y seleccionamos la opción *Restaurar Sistema* del menú *Herramientas*, Figura 5.112.

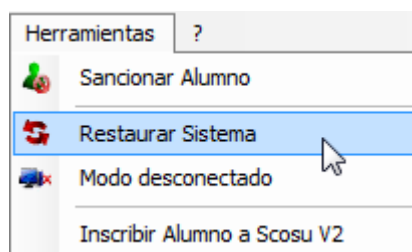


Figura 5.112

Restaurar Sistema solo restaura el estado Ocupada ya que los demás estados se colocan automáticamente cuando se entra a la sala.

5.1.2.1.16 Modo desconectado

El Modo desconectado se utiliza cuando por alguna razón la conexión al servidor Tork o Cozumel falla o cuando existen problemas con la conexión de internet. Esta opción nos permite trabajar de manera local guardando todos los registros en el sistema, por lo que es necesario que se actualice la base de datos en el servidor con los registros almacenados, una vez que la conexión se restablezca.

Un punto importante es que esta función solo se podrá utilizar si la conexión falla cuando el sistema está trabajando, si tratamos de entrar al sistema y no existe conexión a los servidores o a internet, no podremos hacerlo, debido a que SCOSU v2 valida la contraseña, comparando la que se escribe en el TextBox *Contraseña* (*véase el punto 5.1.2.1.2 Entrando al sistema*) con la que se encuentra en la base de datos en Tork.

5.1.2.1.16.1 Cambiando a Modo Desconectado

Seleccionamos la opción *Modo desconectado* del menú *Herramientas*, Figura 5.113.

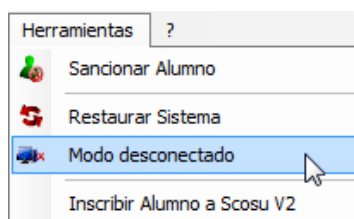


Figura 5.113

Enseguida nos muestra un mensaje advirtiéndonos que el sistema se cambiara a modo desconectado, Figura 5.114.

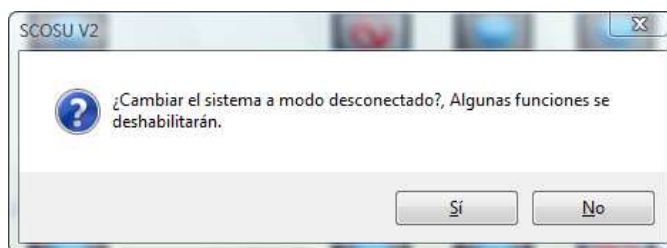


Figura 5.114 Modo desconectado

Si damos clic en Sí, se cambia a dicho modo, cambiando el botón de la barra de herramientas de *Modo Desconectado* a *Modo Conectado*, Figura 5.115, y deshabilitando algunas funciones de SCOSU v2 las cuales necesitan la conexión a internet, por lo que no funcionarían en Modo desconectado.



Figura 5.115 Cambio de modo

Las funciones que se deshabilitan son:

De la barra de menús:

- Destinar sala.
 - Curso
 - Mantenimiento
- Ver
 - Sancionados
- Herramientas
 - Restaurar Sistema
 - Modo desconectado
 - Inscribir Alumno a SCOSU v2

- ?
 - Contactar al Administrador

Del menú contextual de las computadoras

- Sancionar alumno (cuando se encuentra ocupando una computadora)

De la barra de herramientas:

- Sancionados
- Restaurar Sistema

5.1.2.1.16.2 Cambiando a Modo Conectado

Una vez restablecida la conexión es necesario actualizar la base de datos en Tork con los registros almacenados en el sistema, por lo que tendremos que cambiarlo a Modo conectado, para ello damos clic en el botón que se encuentra en la barra de herramientas *Modo Conectado*.

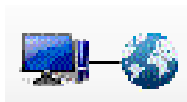


Figura 5.116 Modo Conectado

Mostrándonos el siguiente mensaje.

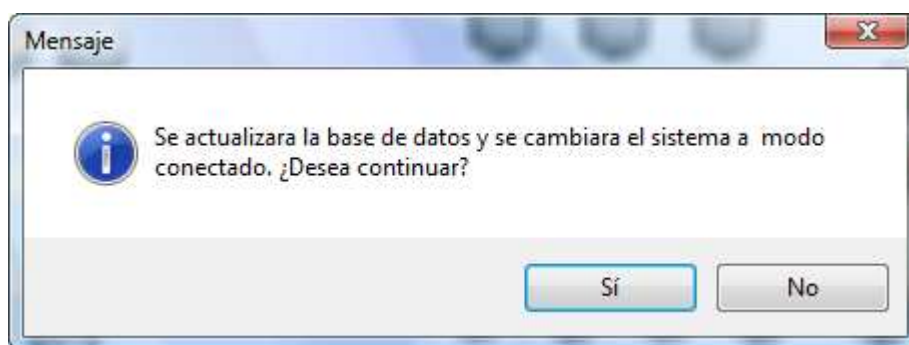


Figura 5.117

Si damos clic en Sí, nos muestra otro mensaje indicando que la base de datos ha sido actualizada y SCOSU v2 se cambia a Modo Conectado, cambiando el botón de la barra de herramientas de *Modo Conectado* a *Modo Desconectado*, Figura 5.118, y habilitando las funciones que se habían deshabilitado en el modo desconectado.



Figura 5.118 Cambio de modo

5.1.2.1.17 Inscribir Alumno a SCOSU v2

Si a un alumno se le va asignar una computadora y el sistema indica que no está dado de alta (véase el punto 5.1.2.1.5 Asignación de equipo), es necesario inscribirlo a SCOSU v2, esto se realiza seleccionando la opción *Inscribir Alumno a SCOSU v2* del menú *Herramientas*, Figura 5.119.

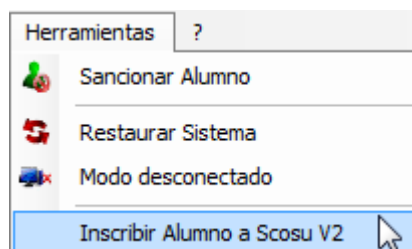


Figura 5.119

Con lo que nos abrirá la siguiente pagina.



Figura 5.120 Pagina de inscripción a SCOSU v2

Para poder entrar es necesario proporcionar un nombre de usuario y contraseña, sino se cuenta con ella deberá consultarlo con el jefe de sala. Una vez dentro se dará de alta al alumno.

Nota: El procedimiento de inscripción queda fuera del alcance de este manual.

5.1.2.1.18 Ayuda de SCOSU v2

La ayuda nos muestra a detalle el funcionamiento de SCOSU v2 mediante este manual, para utilizarla seleccionamos la opción *Ayuda de SCOSU v2* del menú *?*, Figura 5.121.

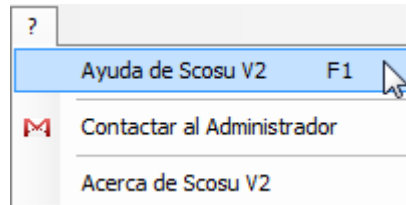


Figura 5.121

5.1.2.1.19 Contactar al Administrador

El usuario podrá contactar directamente al administrador utilizando a SCOSU v2 mediante correo electrónico, informándole de alguna falla en el sistema, en la conexión u otro asunto. Para utilizar esta opción seleccionamos *Contactar al Administrador* del menú *?*, Figura 5.122.

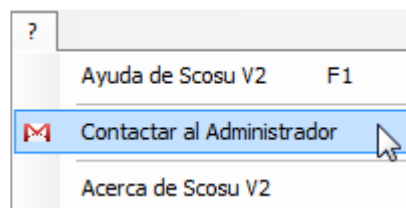


Figura 5.122

Con lo cual nos aparecerá la siguiente ventana.

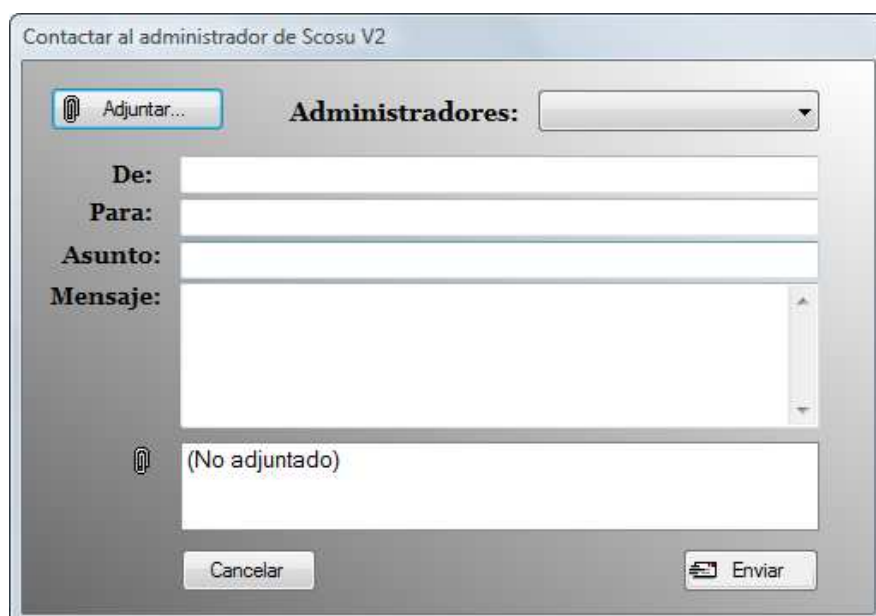
A screenshot of a dialog box titled 'Contactar al administrador de Scosu V2'. The window has a light blue header. Inside, there is an 'Adjuntar...' button with a paperclip icon. To its right is a dropdown menu labeled 'Administradores:'. Below these are four input fields: 'De:', 'Para:', 'Asunto:', and 'Mensaje:'. The 'Mensaje:' field is a large text area with a vertical scrollbar. At the bottom, there is a 'Cancelar' button and an 'Enviar' button with an envelope icon. A small paperclip icon is also visible next to the text '(No adjuntado)' in the message area.

Figura 5.123 Ventana de envío de correo

En ella se debe de especificar el nombre de quien envía el correo en el TextBox *De*, en la sección *Para* se colocara a quien va dirigida, para este caso cuenta con un ComboBox donde se encuentran el nombre de los administradores del sistema, Figura 5.124.

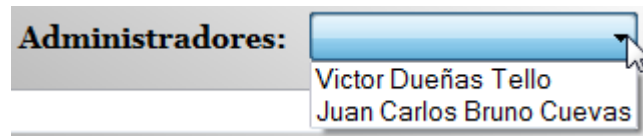


Figura 5.124 ComboBox de Administradores

Del cual se podrá seleccionar el que desee o colocar los dos, dependiendo de cual seleccione se colocara su correo electrónico en el TextBox *Para*, Figura 5.125. También es posible colocar una dirección de correo electrónico ajena a las de los administradores, en caso de que se desee comunicar con otra persona.

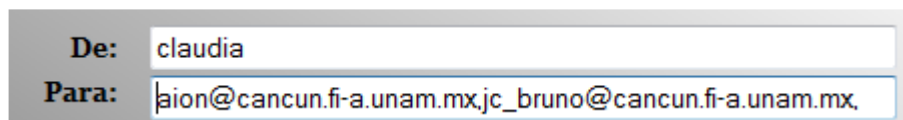


Figura 5.125

Después colocamos el asunto del correo en el TextBox *Asunto*, redactamos el mensaje en la sección *Mensaje* y si lo requiere podemos adjuntar un archivo, para ello damos clic en el botón *Adjuntar* apareciéndonos un cuadro de dialogo, donde seleccionaremos el archivo que necesitamos, Figura 5.126.

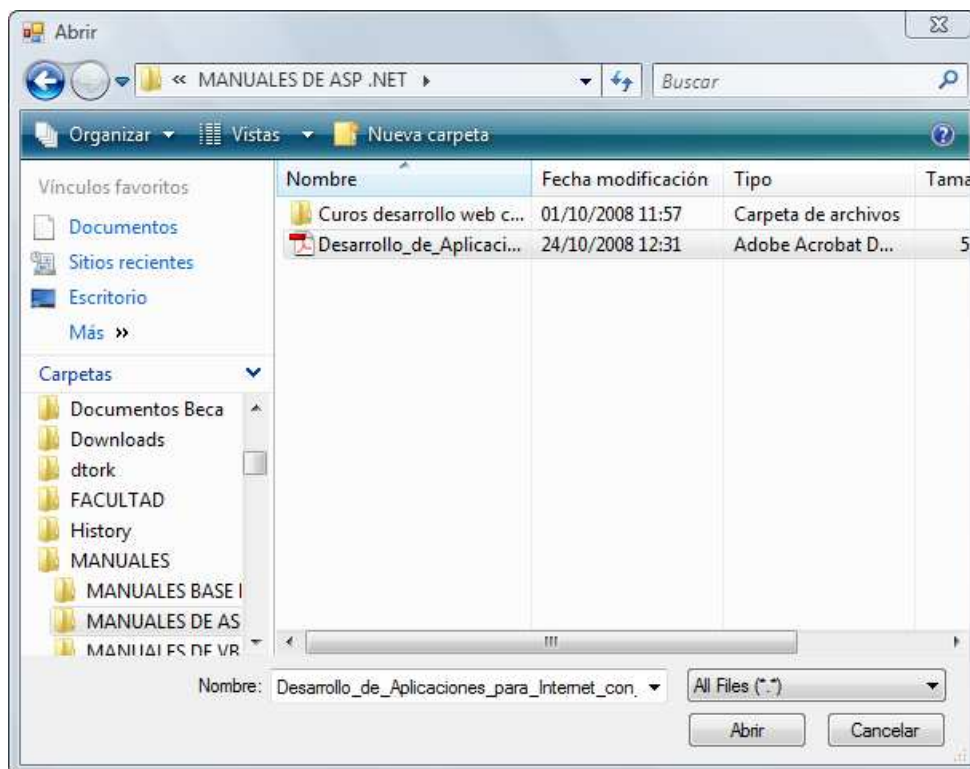
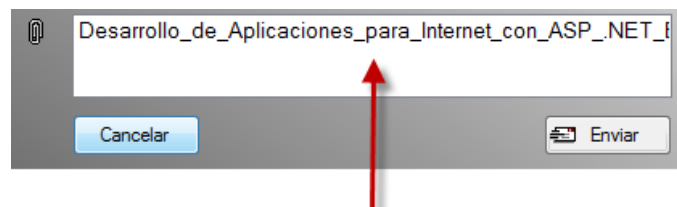


Figura 5.126 Cuadro de dialogo para adjuntar archivo

Una vez seleccionado el archivo damos clic en el botón *Abrir*, cerrando el cuadro de dialogo y regresándonos a la ventana de envío de correo donde podemos observar que el archivo ya se encuentra adjuntado, Figura 5.127, si damos clic en el botón *Cancelar* del cuadro de dialogo, este se cierra y se cancela la acción.



Archivo adjuntado

Figura 5.127

Enseguida damos clic en el botón *Enviar*. Si no especificamos de parte de quien es el correo el sistema nos muestra una advertencia indicando el error cometido, Figura 5.128.

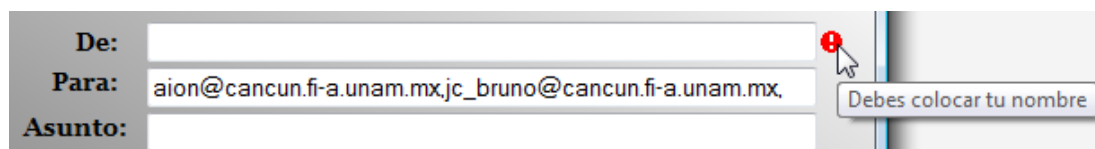


Figura 5.128

De la misma manera nos muestra un error similar si no colocamos para quien va dirigido, además si no ponemos la estructura correcta de un correo electrónico el sistema nos muestra una advertencia indicando el error, Figura 5.129.

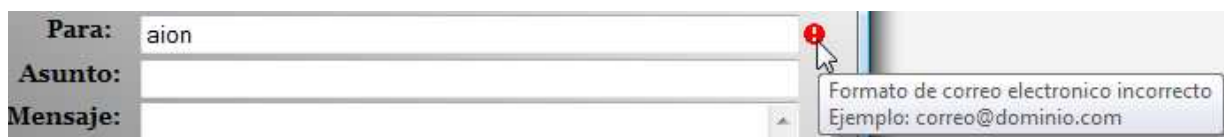


Figura 5.129 Formato de correo incorrecto

Asimismo en el TextBox *Asunto* es necesario especificar el mismo, de lo contrario nos aparecerá un mensaje de advertencia parecido al de la Figura 5.128.

La parte de *Mensaje* y *Adjuntar archivo* no son obligatorias. Por lo tanto si se han llenado los campos obligatorios y se han puesto los datos correctos, tal como nos muestra la Figura 5.130 y damos clic en *Enviar* el sistema muestra un mensaje indicándonos que el correo se ha enviado satisfactoriamente, cerrando la ventana de envío de correo y regresando a la sala correspondiente.

Si damos clic en el botón *Cancelar* cierra la ventana de envío de correo, cancela la acción y regresa a la sala correspondiente.

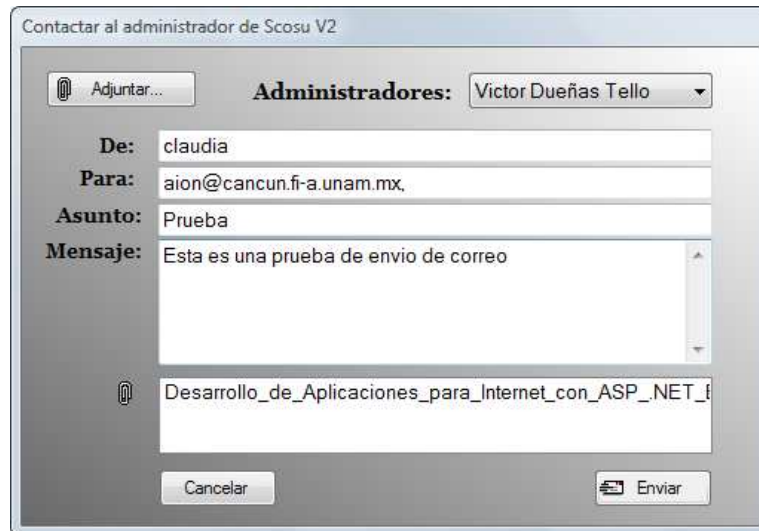


Figura 5.130 Ventana de envío de correo con datos correctos

El correo llegara con una dirección donde especifica desde que sala se envió (por ejemplo Sala3@cancun.fi-a.unam.mx) junto con el nombre de quien lo hizo y los demás datos colocados.

5.1.2.1.20 Acerca de SCOSU v2

Si seleccionamos la opción *Acerca de SCOSU v2* del menú *?*, el sistema nos mostrara una ventana como la de la Figura 5.131, donde presenta detalles del sistema y de su creación.



Figura 5.131 Acerca de

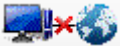




Para regresar damos clic en el botón *Aceptar* cerrando la ventana y volviendo a la sala correspondiente.

5.1.2.1.21 Barra de herramientas

SCOSU v2 cuenta con una barra de herramientas, Figura 5.132, donde la mayoría de sus elementos no son más que accesos directos a las funciones del sistema que más utiliza el usuario, cuyas opciones se encuentran en los diferentes menús. A continuación se especifica cada uno de ellos:



Figura 5.132 Barra de herramientas

-  **Modo desconectado:** Acceso directo a la opción *Modo desconectado* del menú *Herramientas*, véase el punto 5.1.2.1.16 *Modo desconectado*.
-  **Sancionados:** Acceso directo a la opción *Sancionados* del menú *Ver*, véase el punto 5.1.2.1.12 *Ver Sancionados*.
-  **Sancionar Alumno:** Acceso directo a la opción *Sancionar Alumno* del menú *Herramientas*, véase el punto 5.1.2.1.14.1 *Sancionar un alumno cuando NO se encuentra utilizando un equipo*.
-  **Restaurar Sistema:** Acceso directo a la opción *Restaurar Sistema* del menú *Herramientas*, véase el punto 5.1.2.1.15 *Restaurar Sistema*.
-  **Buscar alumno:** Esta opción se explica a detalle en el siguiente punto.

5.1.2.1.22 Buscar alumno

Existen momentos en que es necesario saber si cierto alumno se encuentra ocupando una computadora en algún aula de la sala, para esos casos SCOSU v2 cuenta con la función de buscar al alumno en todas la computadoras que se encuentren en estado Ocupada, podemos encontrar esta función en la barra de herramientas, Figura 5.133. Cuenta con dos opciones, la primera es buscar al alumno mediante su nombre parcial o completo, la segunda es buscarlo mediante su número de cuenta.



Figura 5.133 Búsqueda de alumno

5.1.2.1.22.1 Búsqueda de alumno mediante su nombre

Esta función es la que encontraremos por default en la búsqueda, también podemos elegirla dando clic sobre el botón de *Buscar alumno* y seleccionando la opción *Por nombre*. Con esta opción nos aparecen tres TextBox donde colocaremos el nombre, apellido paterno o materno respectivamente, si no se identifica cual TextBox le corresponde a cada caso, solo basta con colocar el puntero del mouse sobre ellos para que muestre un mensaje de a cual le corresponde, Figura 5.134.

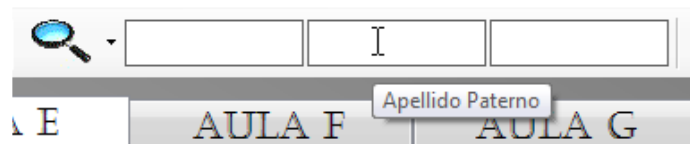


Figura 5.134

El nombre del alumno pueden ser parcial o completo, dependiendo de la información que se coloque SCOSU v2 mostrara todas las coincidencias que encuentre, por lo que es recomendable que sean lo más preciso posible.

Después de haber colocado el nombre completo o parcial del alumno damos *Enter*, dando como resultado una lista con todas las coincidencias que el sistema encontró, Figura 5.135, mostrando el número de cuenta y nombre completo de todos los alumnos encontrados, así como la computadora que están ocupando y el aula donde se encuentra.

Opción 1.- Nombre completo →



Opción 2.- Nombre parcial →



Resultado de las dos opciones

Alumno encontrado				
	Num. Cuenta	Nombre	Computadora	Aula
▶	306171473	JOSE EDUARDO AVILES LOPEZ	71	F
	407007961	JOSE DANIEL FALCON RESENDIZ	44	F
	99287753	JOSE CARLOS RAMIREZ HERNANDEZ	43	F
	304040003	JOSE MANUEL VELAZQUEZ REYNA	2	E
	304184880	JOSE LUIS VASQUEZ SANTOS	3	E
*				

Regresar

Figura 5.135 Resultados de la búsqueda José

Si se coloca el nombre de un alumno que no se encuentra ocupando una computadora de la sala en ese momento, el sistema nos muestra un mensaje indicando la situación, Figura 5.136.



Figura 5.136

5.1.2.1.22.2 Búsqueda de alumno mediante su número de cuenta

Si queremos utilizar esta opción damos clic en el botón *Buscar alumno* y seleccionamos la opción *Por número de cuenta*, mostrándonos únicamente un TextBox, Figura 5.137.



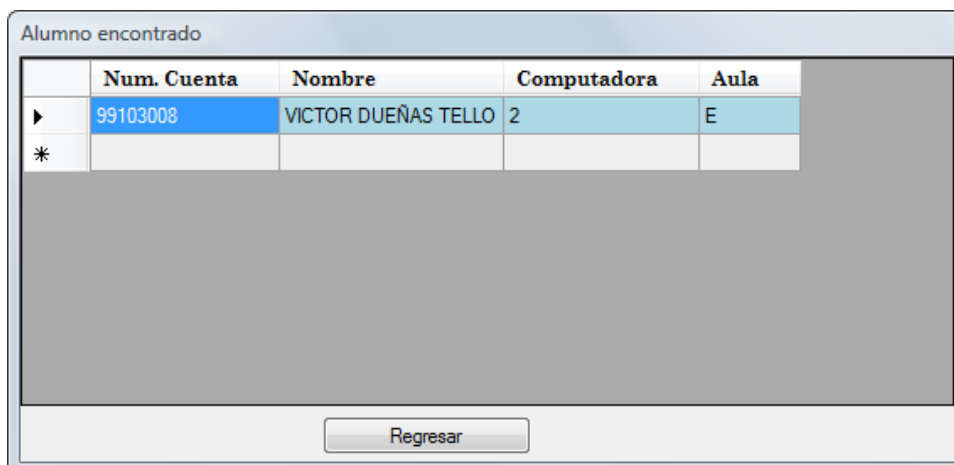
Figura 5.137 Búsqueda por número de cuenta

El TextBox solo acepta números, en él debemos colocar los 9 dígitos correspondientes del número de cuenta, si no lo escribimos completo nos muestra un mensaje de advertencia indicando el error, Figura 5.138.



Figura 5.138

Si el número de cuenta es incorrecto o no se encuentran coincidencia, el sistema nos muestra un mensaje como el de la Figura 5.136. Entonces si el número de cuenta esta completo y es correcto, SCOSU v2 muestra al alumno encontrado, mostrándonos además su nombre completo, la computadora que está utilizando y el aula donde se encuentra, Figura 5.139.



	Num. Cuenta	Nombre	Computadora	Aula
▶	99103008	VICTOR DUEÑAS TELLO	2	E
*				

Regresar

Figura 5.139 Resultado de la búsqueda por número de cuenta

La diferencia entre buscar al alumno por nombre o por número de cuenta es que, por el primero la lista mostrada tendrá varios resultados por todas las coincidencias que encontrara el sistema, por el segundo solo mostrara al alumno buscado, pero será necesario contar con el número de cuenta de dicho alumno, un dato más difícil de saber.

5.1.2.1.23 Vista previa

Las salas 1, 2 y 3 cuentan con una vista previa de las aulas que contienen, mostrando sus respectivas computadoras y el estado en que se encuentran. La vista previa se actualiza conforme se va cambiando el estado de las maquinas.



Figura 5.140 Vista previa de la sala 3

La sala 4 no tiene vista previa debido a que solo cuenta con un aula.

Si pasamos el puntero del mouse sobre cualquier computadora de la vista previa, nos mostrara un mensaje indicando de qué computadora se trata, Figura 5.141.

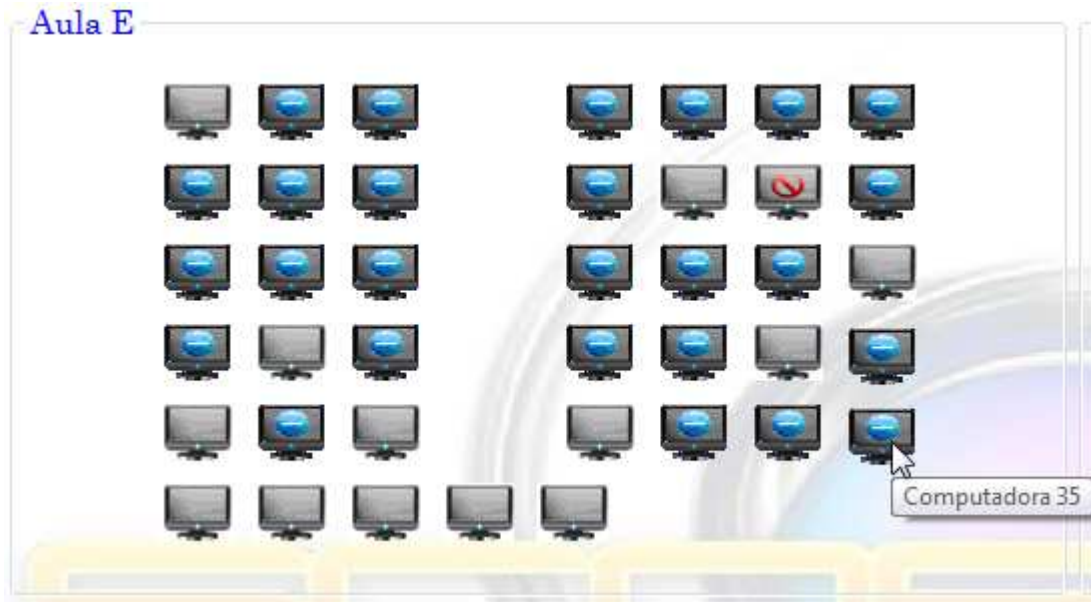


Figura 5.141

5.1.2.2 Manual del usuario de SCOSU v2 Modulo del Jefe

5.1.2.2.1 Entrando al sistema

Como ya habíamos mencionado anteriormente SCOSU v2 Modulo del Jefe se instalara solamente en las maquinas de los jefes de sala, por lo cual cada jefe podrá acceder únicamente a la sala que le corresponde seleccionando su sala e introduciendo su contraseña como se muestra en la siguiente figura.

Contraseña

SCOSU v2

Única
Unidad de Servicios de Cómputo Académico
Facultad de Ingeniería

Autor: Victor Dueñas Tello

Modulo del Jefe

Sala: Sala 1

Contraseña: *****

Aceptar Salir

Sala seleccionada Contraseña

Figura 5.142 Form de autenticación de usuario.

En caso de no haber seleccionado alguna de las cuatro salas, en un costado de la casilla se colocara un símbolo de interrogación que al colocarnos sobre el nos mandara una advertencia de que seleccionemos alguna de las salas para poder continuar, de igual manera si no hemos seleccionado alguna sala y no hemos introducido una contraseña no nos dejara avanzar a menos que seleccionemos una sala.

Si bien has seleccionado alguna sala, pero la contraseña fue mal tecleada o incorrecta nos manda un mensaje de alerta para verificar nuestros datos ingresados.

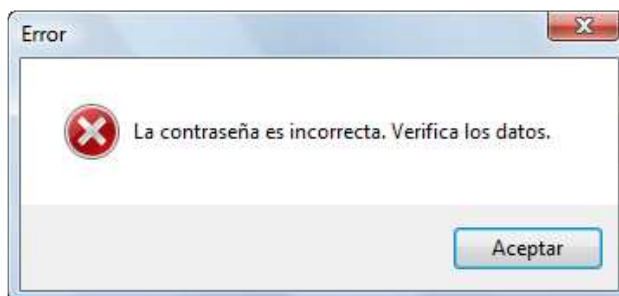


Figura 5.143 Mensaje de contraseña Incorrecta

Si los datos fueron ingresados correctamente pasaremos al siguiente Form, Figura 5.144, el cual es el escenario donde trabajara el jefe de la sala donde cuenta con una serie de menús, que más adelante se explicaran uno por uno.



Figura 5.144 Escenario principal de SCOSU v2 Modulo del Jefe

5.1.2.2 Reportes

Para seleccionar esta opción seleccionamos *Reporte* del menú *Archivo*.

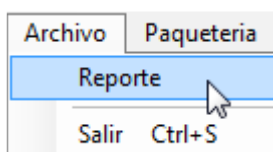


Figura 5.145 Menú Archivo

Nos muestra como su nombre lo dice el reporte de los usuarios que han utilizado las salas de cómputo de UNICA, ya sea por día o en general, de cada uno de los alumnos de la facultad.

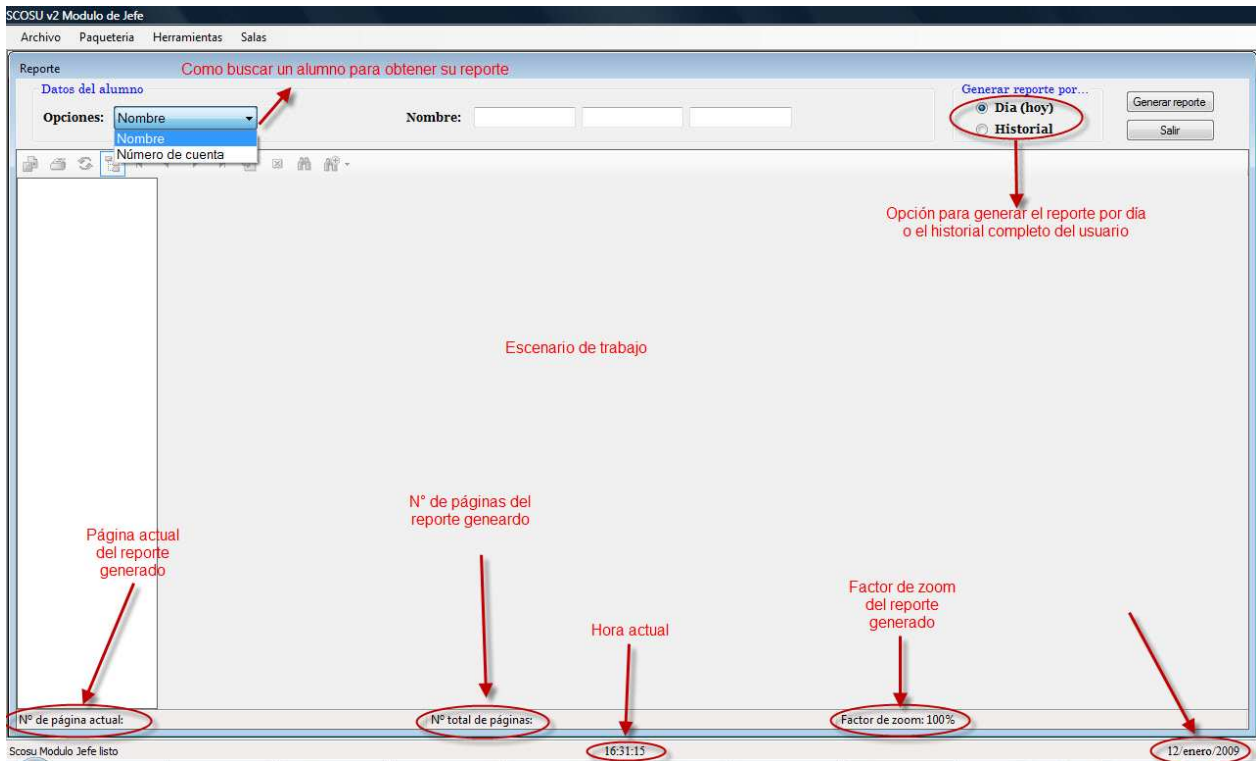


Figura 5.146 Escenario principal de la generación de Reportes.

5.1.2.2.1 Reportes por nombre y número de cuenta

Dentro de este tenemos dos Formas de obtener el reporte, uno de ellos es por nombre del alumno, donde debemos de ingresar el nombre completo del alumno del cual deseamos obtener su reporte, así como si lo deseamos obtener por día (fecha actual) o bien el historial completo.

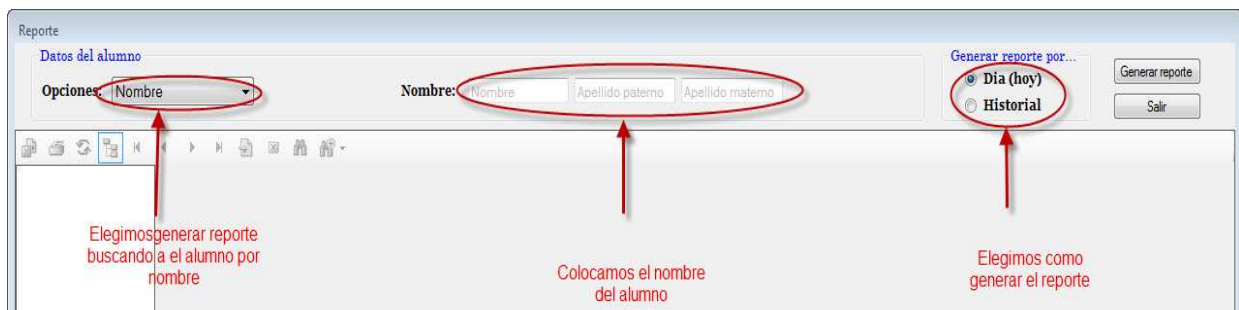


Figura 5.147 Generar Reporte por nombre.

En caso de no colocar el nombre completo del alumno el sistema muestra un mensaje de advertencia indicando el error cometido, Figura 5.148.

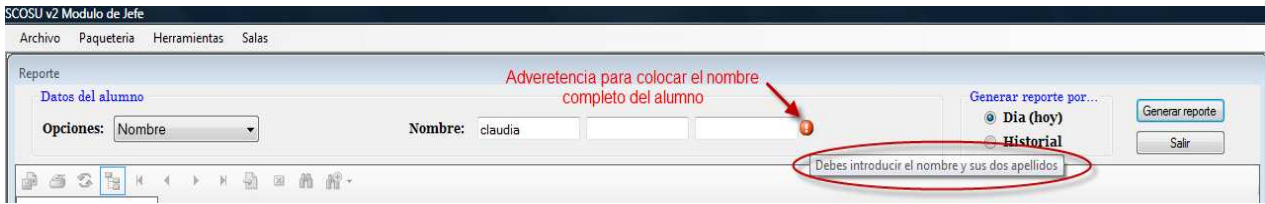


Figura 5.148 Advertencia para colocar el nombre completo del alumno.

Ahora bien si se ha colocado el nombre completo del alumno y oprimimos el botón de generar reporte sin haber elegido una opción de cómo generarlo ya sea por día o bien el historial completo del alumno, el sistema no muestra un mensaje de advertencia indicando que seleccionemos alguna opción, Figura 5.149.

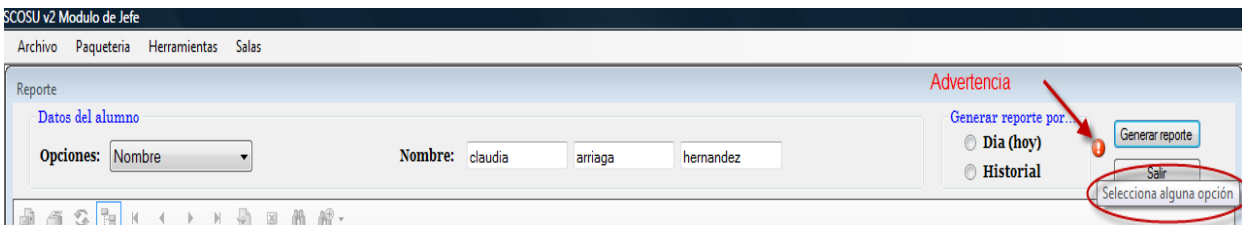


Figura 5.149 Advertencia por no elegir una opción para generar el reporte.

Por otra parte si elegimos generar reporte por el número de cuenta también nos muestra una advertencia en caso de que no sea correcto, al igual que si no se ha elegido una opción de generar reporte ya sea por historial o bien por el día actual.

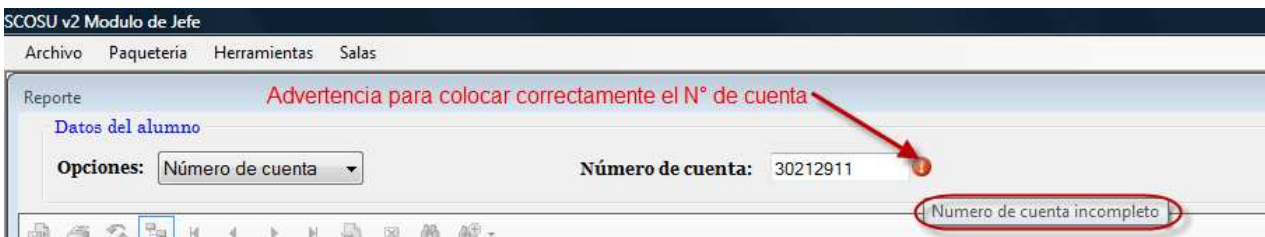


Figura 5.150 Advertencia de número de cuenta.

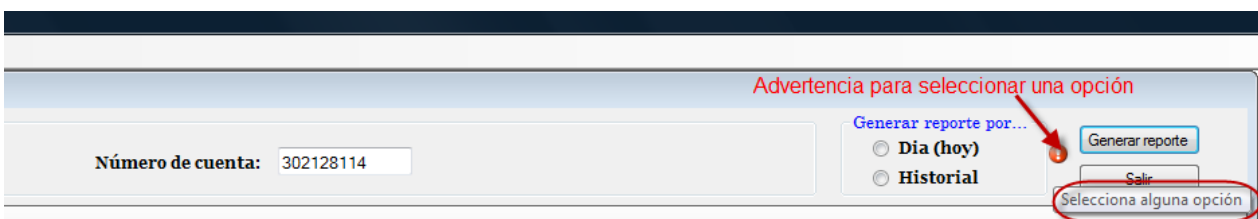


Figura 5.151 Advertencia de elegir una opción de reporte.

5.1.2.2.2 Reportes por día e historial

Si hemos elegido generar historial completo del alumno, nos mostrara un reporte completo del número total de veces que ha ingresado el alumno a las salas de computo de UNICA, el tiempo total en minutos del uso de los equipos así como el informe detallado por día, el cual se desglosa en el tiempo total de su estancia en salas mostrado en minutos, el número de veces que ingreso, la hora de entrada y salida, en que salas ingreso, en que aula y que equipos fueron los que utilizo, Figura 5.152.

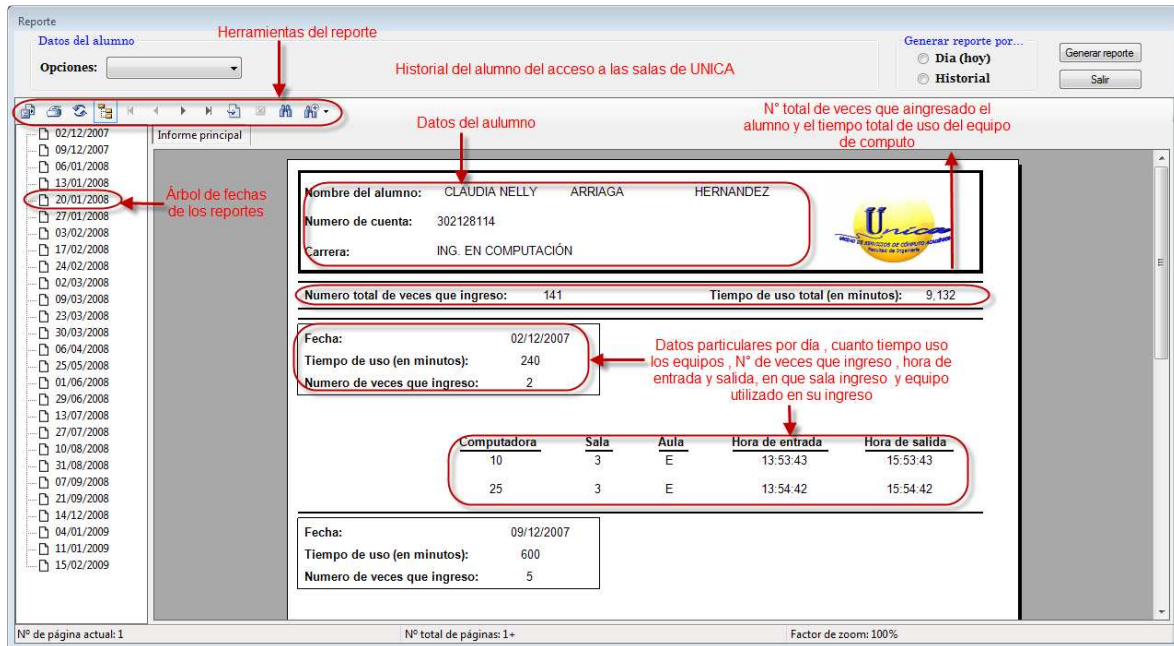


Figura 5.152 Generar reporte por historial.

También dentro de este nos podemos desplazar por la fecha en que ingreso el alumno y ver fechas del reporte en particular, Figura 5.153.

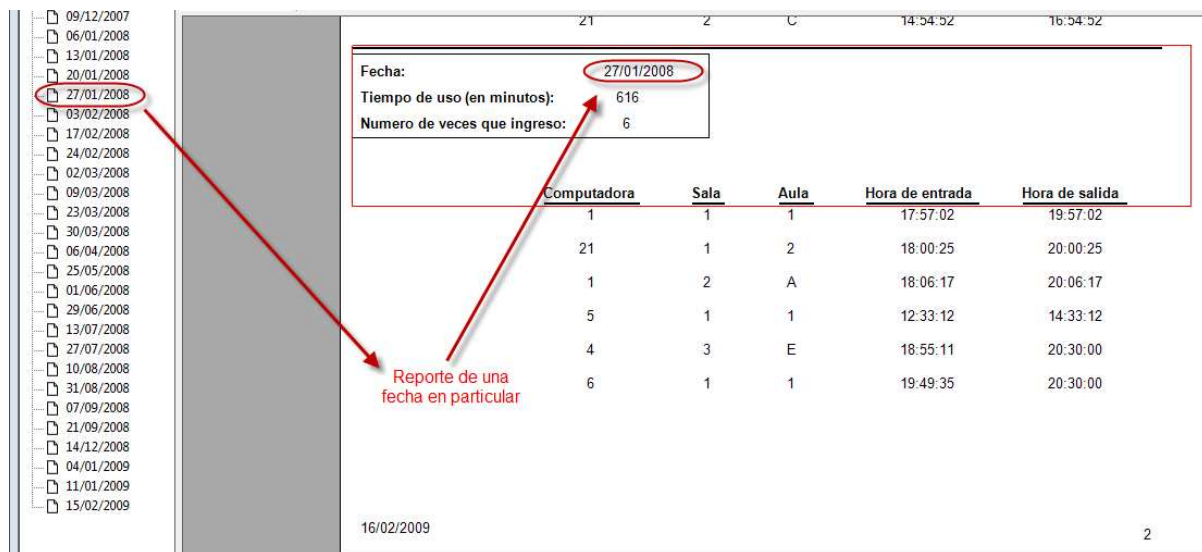


Figura 5.153 Desplazándonos dentro del historial completo del reporte.

Ahora bien, Si elegimos la opción día, solo nos mostrara el reporte si el alumno ha ingresado en la fecha actual, con las misma características que el generado por la opción historial, Figura 5.152, solo que en este caso no podremos desplazarnos por la fechas ya que solo genera la actual.

Si en cualquiera de las dos opciones no existen registros del alumno el sistema nos mostrara un mensaje indicando que no existen datos para realizar el informe, Figura 5.154.

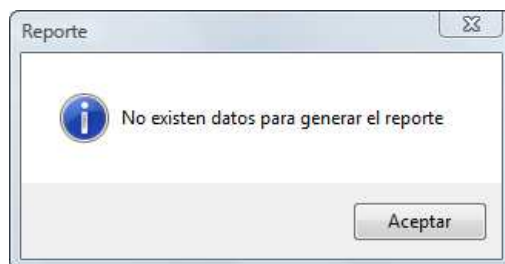



Figura 5.154

5.1.2.2.3 Herramientas del reporte

Una vez realizado el reporte contamos con diferentes herramientas (Figura 5.155) para realizar varias acciones, las cuales describiremos a continuación.



Figura 5.155 Herramientas del Reporte.

- **Guardar** : Opción que nos permite guardar el reporte generado en diferentes formatos, como doc, pdf, xls y más.

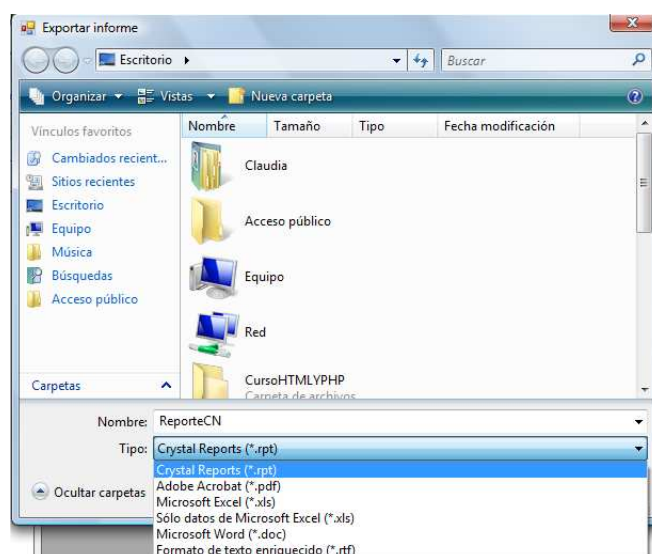










Figura 5.156 Guardar Reporte.

- **Imprimir**  : Imprime nuestro reporte.
- **Actualizar**  : Actualiza dicho reporte.
- **Árbol del Reporte**  : Nos muestra el árbol de reportes.
- **Ir a la primera página**  : Nos lleva a la primera página.
- **Ir a la página anterior**  : Nos lleva una página anterior.
- **Ir a la página siguiente**  : Nos lleva a la página siguiente.
- **Ir a la última página**  : Nos lleva a la última página.
- **Ir a una página en específico**  : Nos lleva a una página que el usuario especifica.

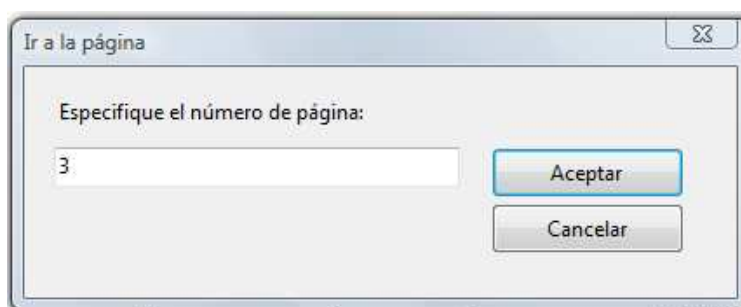




Figura 5.157 Ir a alguna página.

- **Cerrar vista actual**  : Cierra la vista actual.
- **Buscar texto en específico**  : Busca algún texto en específico dentro del reporte.

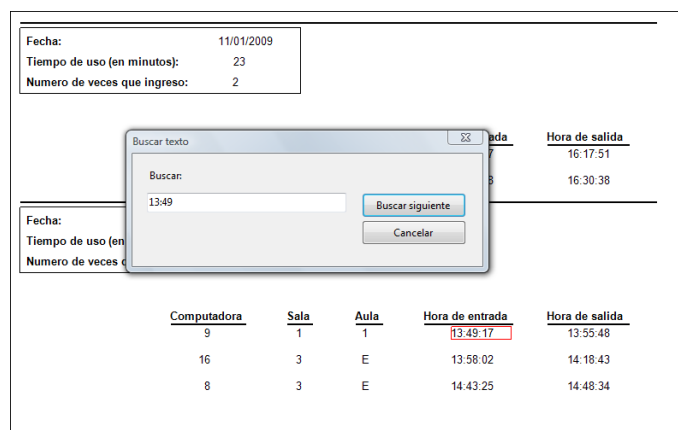



Figura 5.158 Buscar texto.

- **Zoom** : Factor de zoom del documento y en la opción de personalizar el usuario coloca el factor de zoom que desea.

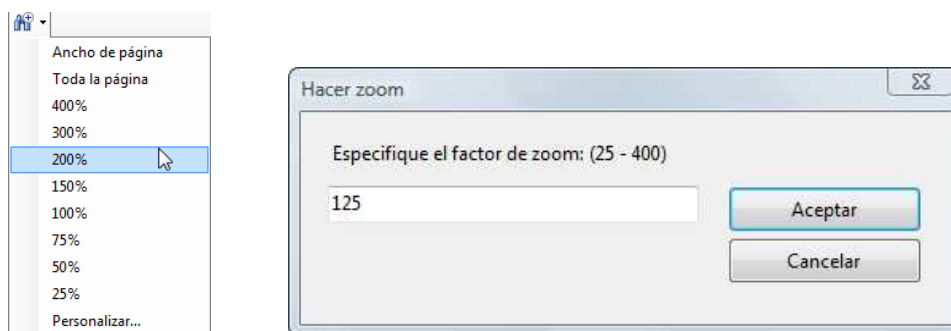


Figura 5.159 Factor de Zoom.

Por último si queremos salir damos clic en el botón *Salir*.

5.1.2.2.3 Salir del sistema

Para salir del sistema seleccionamos la opción Salir del menú archivo, Figura 5.160.

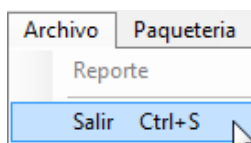


Figura 5.160

Nos mostrara un mensaje preguntando si en verdad deseamos salir, si la respuesta es Sí, el sistema se cierra, de lo contrario se cancela la acción.

5.1.2.2.4 Actualizar software

Para entrar a esta opción seleccionamos *Actualizar software* del menú *Paquetería*, mostrándonos una ventana como la de la Figura 5.162.

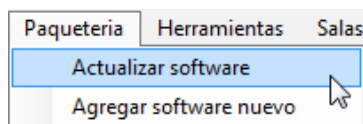


Figura 5.161 Menú Paquetería

Nos permite actualizar el software que se está utilizando actualmente así como agregar más paquetería a la actual dándonos una vista previa del mismo, para ello debemos de elegir el aula del cual queremos actualizar el software así como el sistema operativo de la misma, para que nos active los datos de dicha aula y poder realizar nuestras actualizaciones.

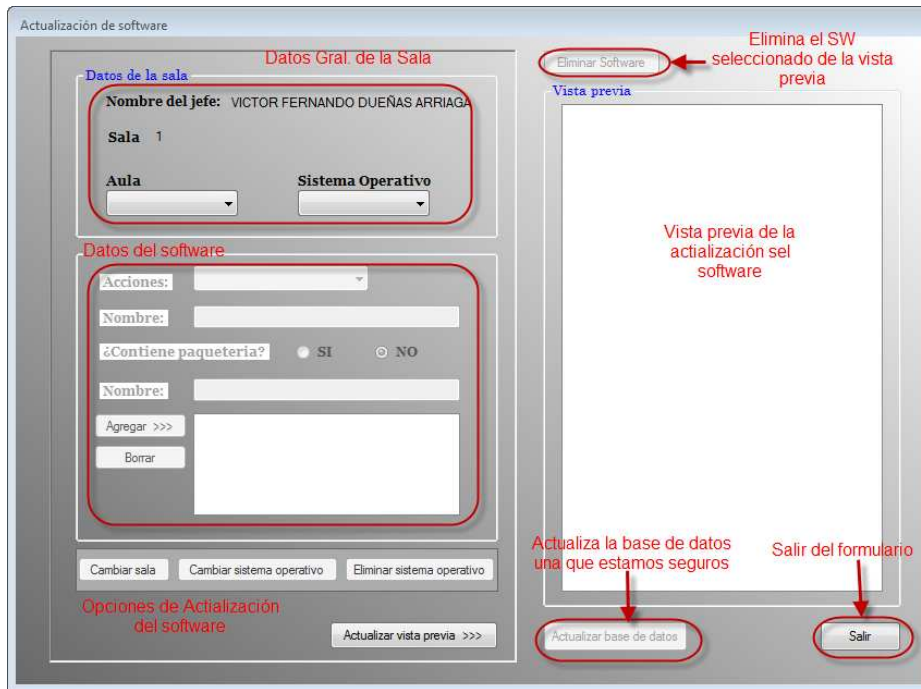


Figura 5.162 Escenario de actualización del software.

Una vez elegido el aula y el sistema operativo se activan los datos del software donde debemos elegir que es lo que queremos realizar en el ComboBox *Acciones*, en él tenemos que existen dos opciones *Actualizar Software* actual y otro de *Agregar Software* al actual, Figura 5.163.

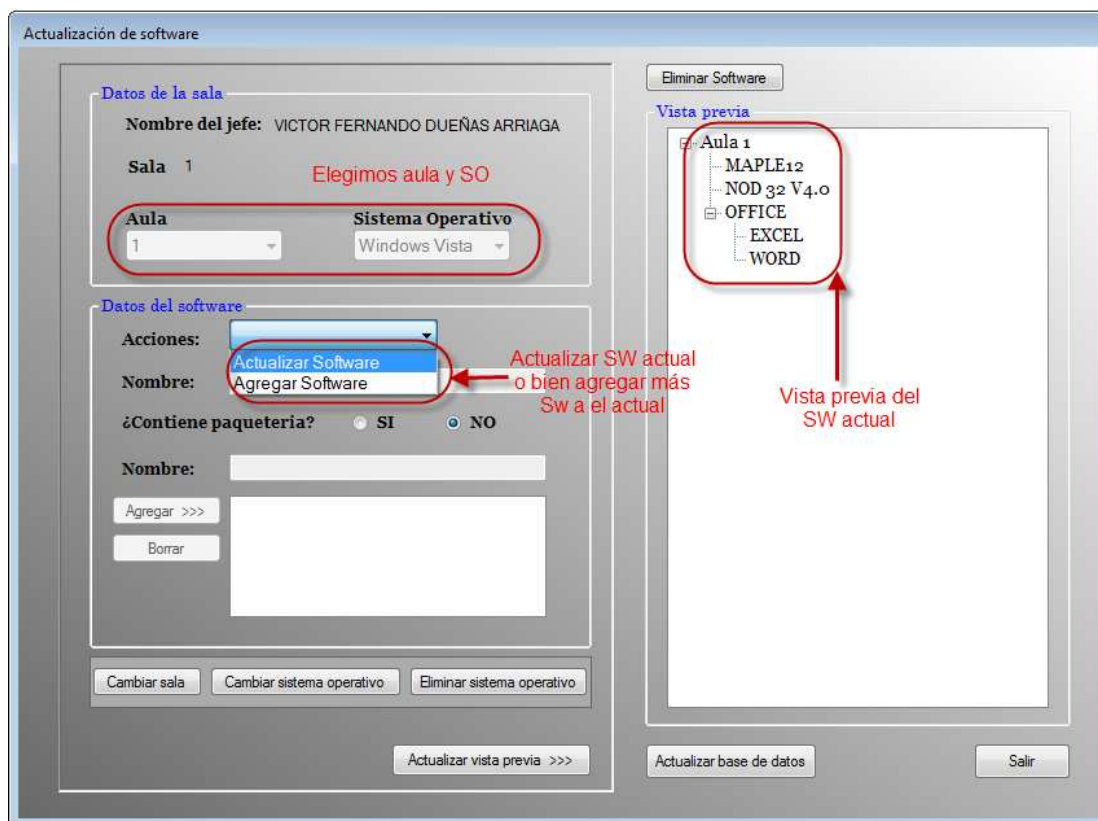


Figura 5.163 Actualizar o agregar software al actual.

Si elegimos dentro de las acciones la de *Actualizar Software* nos pedirá que seleccionemos alguno de los que se encuentran en la vista previa, es decir, el que queremos actualizar, el software mostrado es con el que cuenta actualmente el aula elegida.

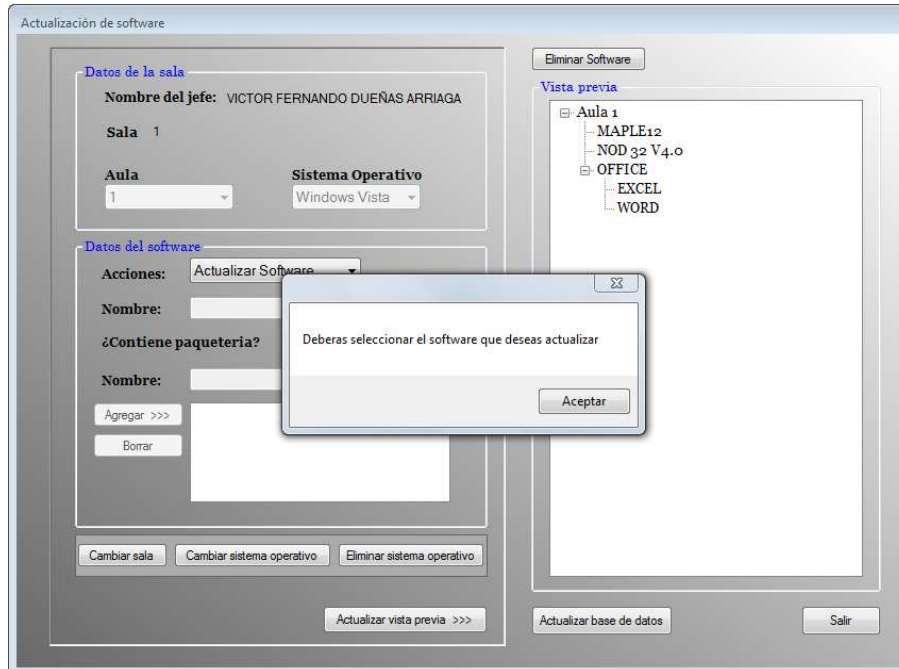


Figura 5.164 Actualizar SW

En caso de haber elegido alguna paquetería de la vista previa, nos mandara un mensaje de que tenemos que seleccionar dentro de la lista el software principal que queremos actualizar y no la paquetería.

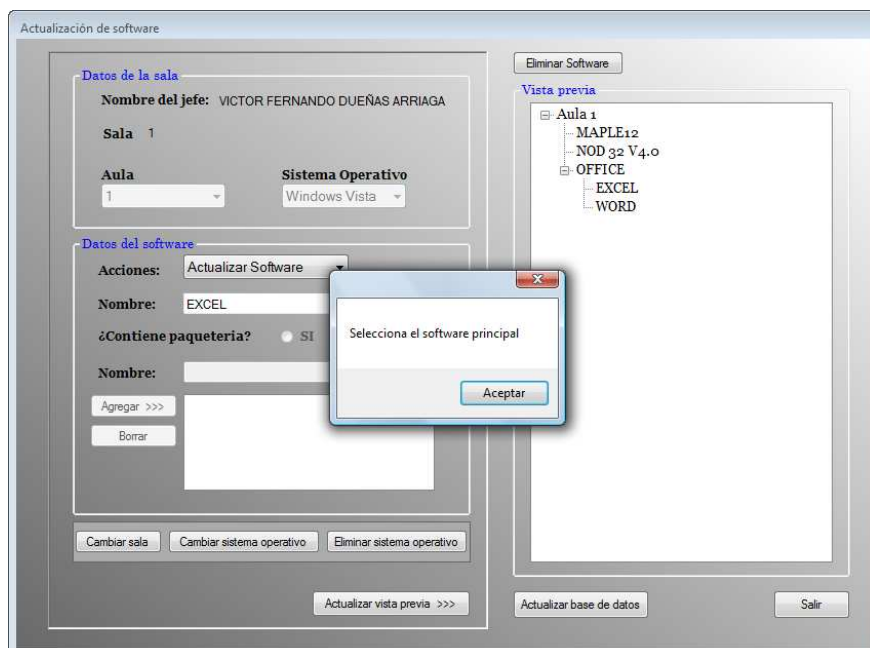


Figura 5.165 Actualizar software principal

Una vez seleccionado el software principal, podremos actualizarlo como lo deseemos, ya se cambiarle el nombre, quitarle alguna paquetería o bien agregársela como se muestra el siguiente ejemplo:

A) Primero que nada elegimos el aula y el Sistema Operativo, así como el software que se desea actualizar, en este caso elegimos OFFICE.

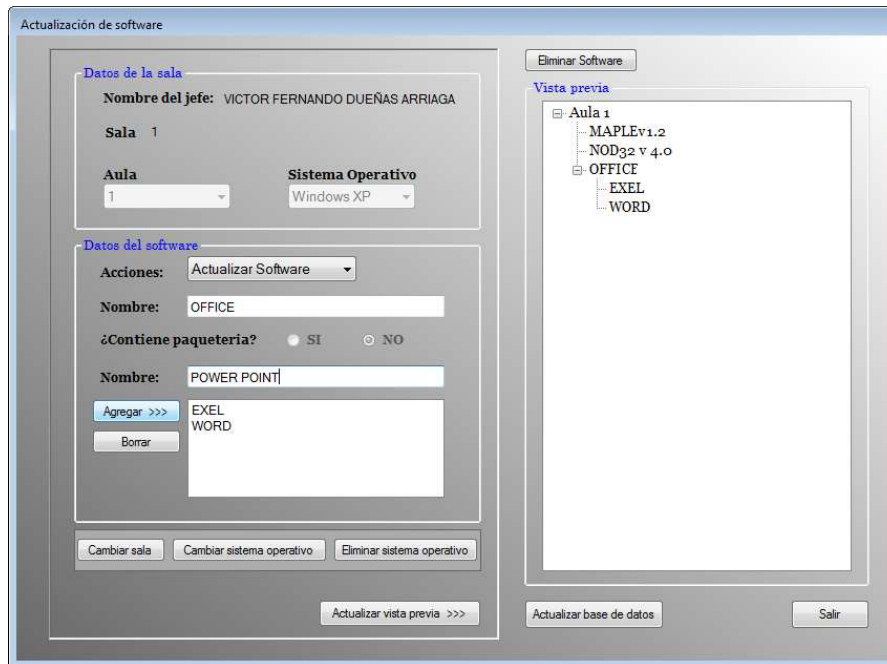
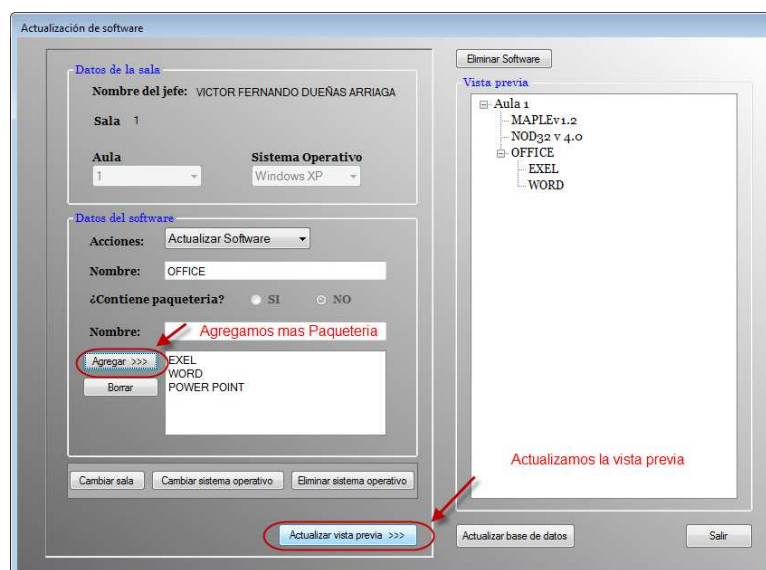


Figura 5.166 Ejemplo Inciso A)

B) Posteriormente agregamos una paquetería por ejemplo POWER POINT y le damos agregar en la parte inferior izquierda y damos actualizar vista previa, para que nos muestre los cambios.



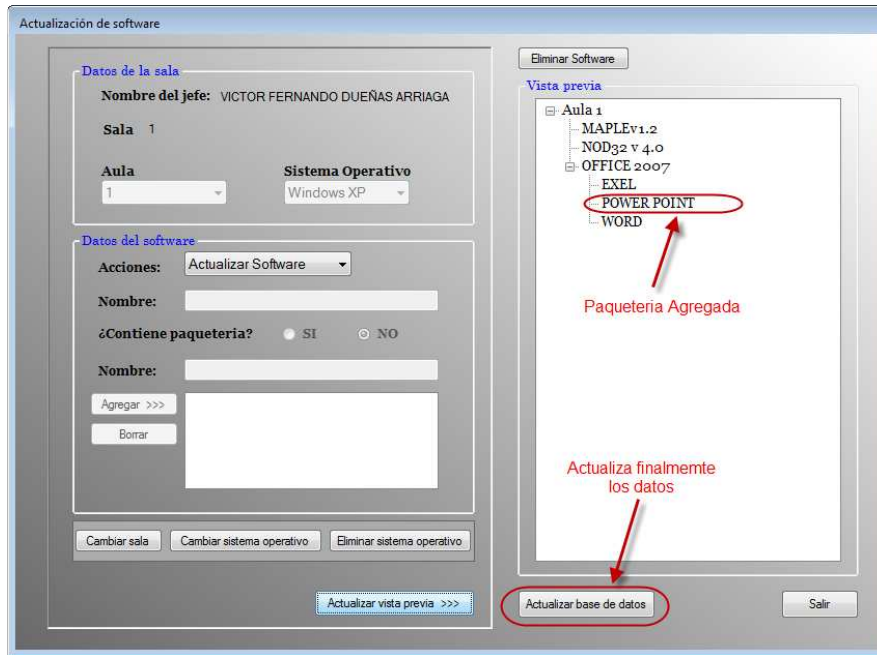


Figura 5.167 Inciso B) Muestra los cambios realizados.

C) Finalmente le damos clic al botón *Actualizar base de datos*, para que realice los cambios realizados.

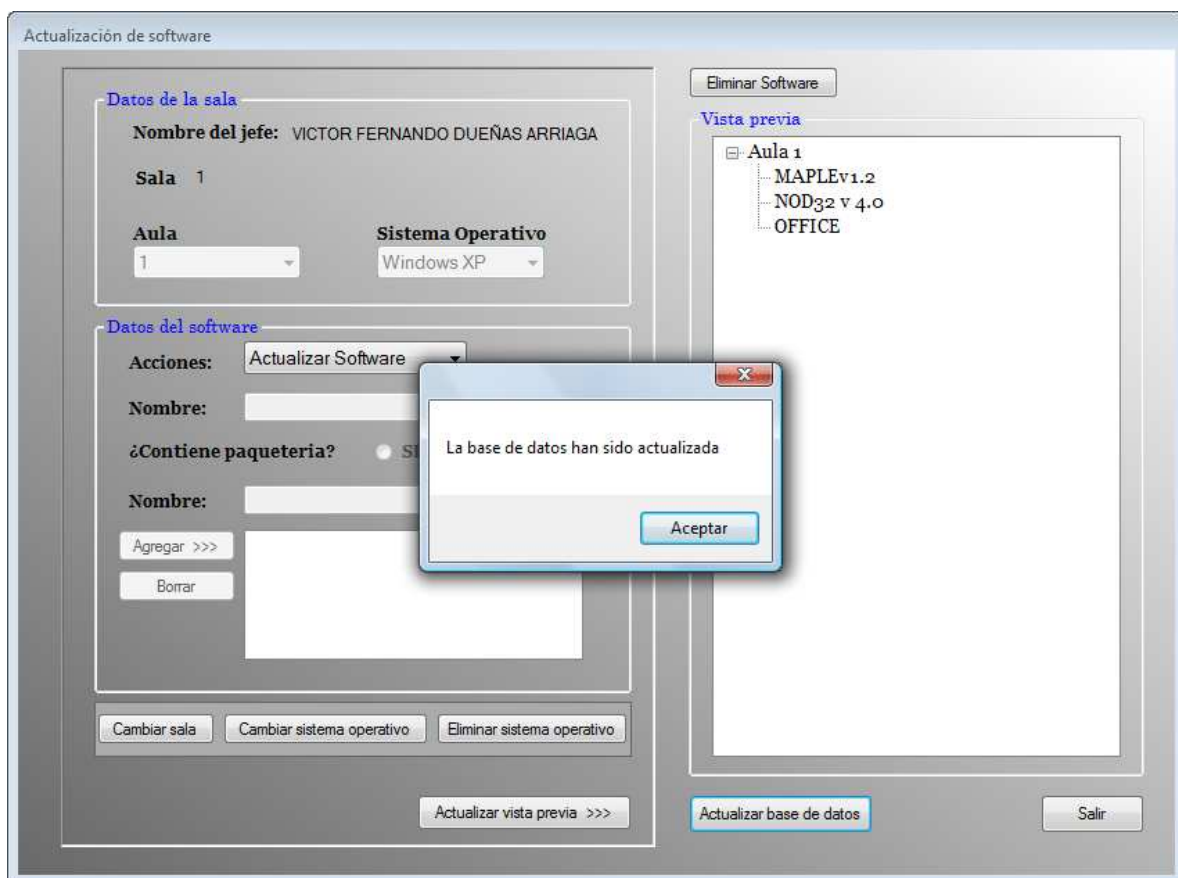


Figura 5.168 Inciso C) Muestra mensaje de envió de datos(Base actualizada).

En caso de querer agregar más software, lo que realizaremos en este caso es seleccionar dentro de acciones la opción de *Agregar Software*. Por ejemplo:

- A) Una vez seleccionada la opción de *Agregar Software* debemos de colocar el nombre del software nuevo, puesto que las demás opciones quedan deshabilitadas hasta que agreguemos su nombre, el este caso solo se habilitan el TextBox *Nombre* y si contienen paquetería o no, en caso de q nos saltemos la opción de introducir el nombre, nos mandara un mensaje de advertencia indicando que aun no hemos colocado el nombre del software nuevo, Figura 5.169.

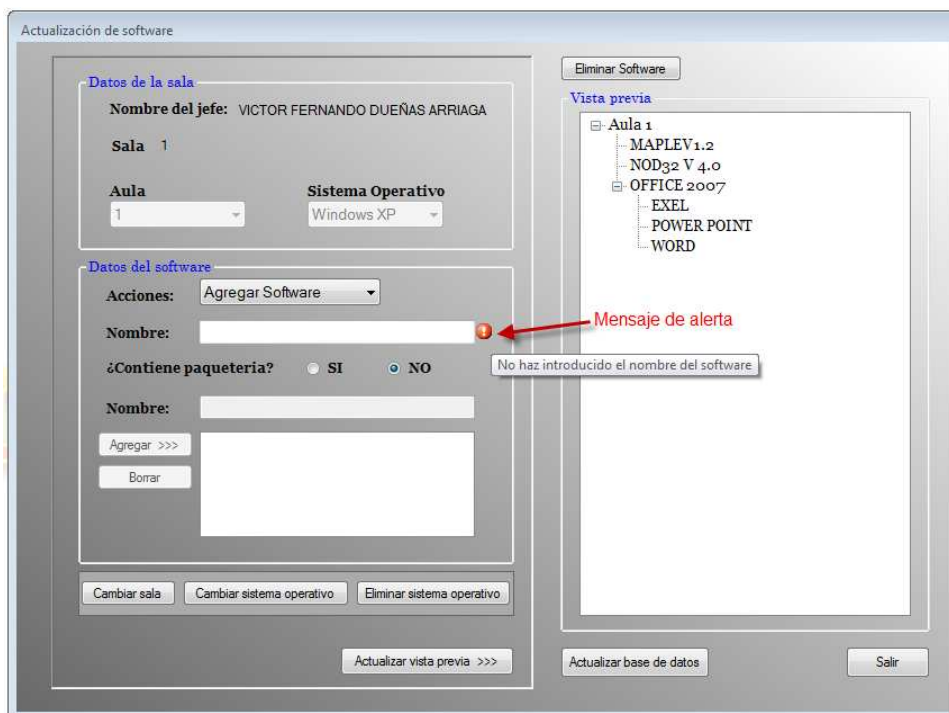


Figura 5.169 Mensaje de error

- B) Después de colocar el nombre hay que seleccionar si contiene o no paquetería, en caso de contenerla, hay que agregarla colocando el nombre y con el botón de *Agregar* se agrega la misma, Figura 5.170, o bien si nos equivocamos seleccionamos el software en la lista que nos muestra en la parte inferior y la borramos dando clic en el botón *Borrar*, Figura 5.171, pero si queremos borra algún software de la lista sin haberlo elegido, el sistema muestra un mensaje de advertencia indicando el error.

En caso de no contener paquetería nos saltamos al siguiente punto.

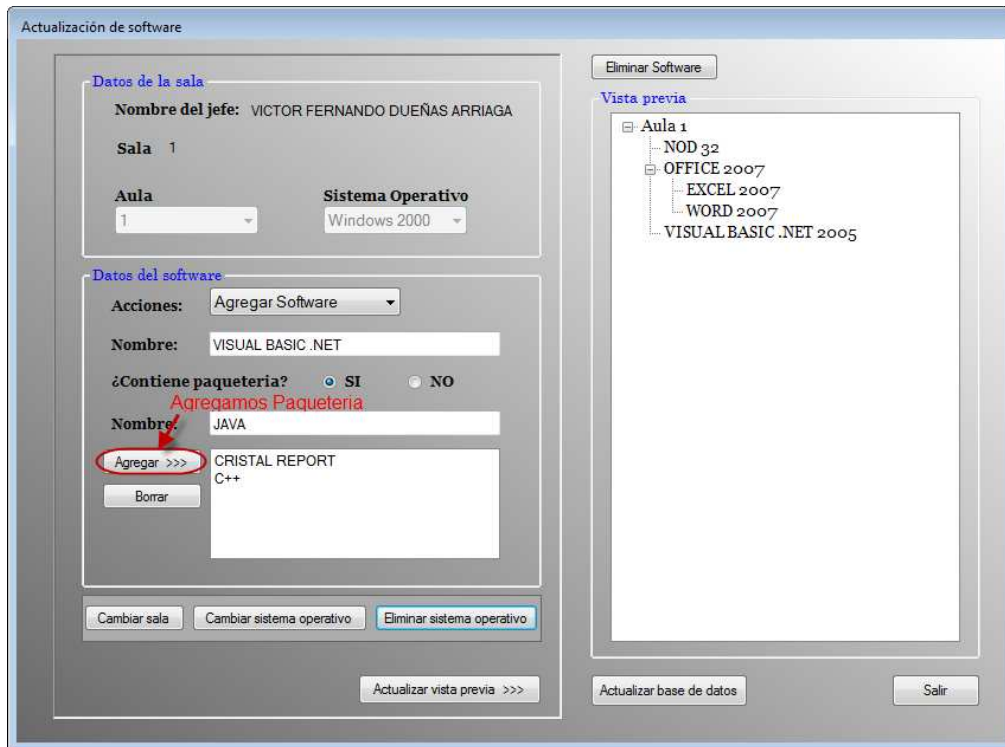


Figura 5.170 Agregar paquetería

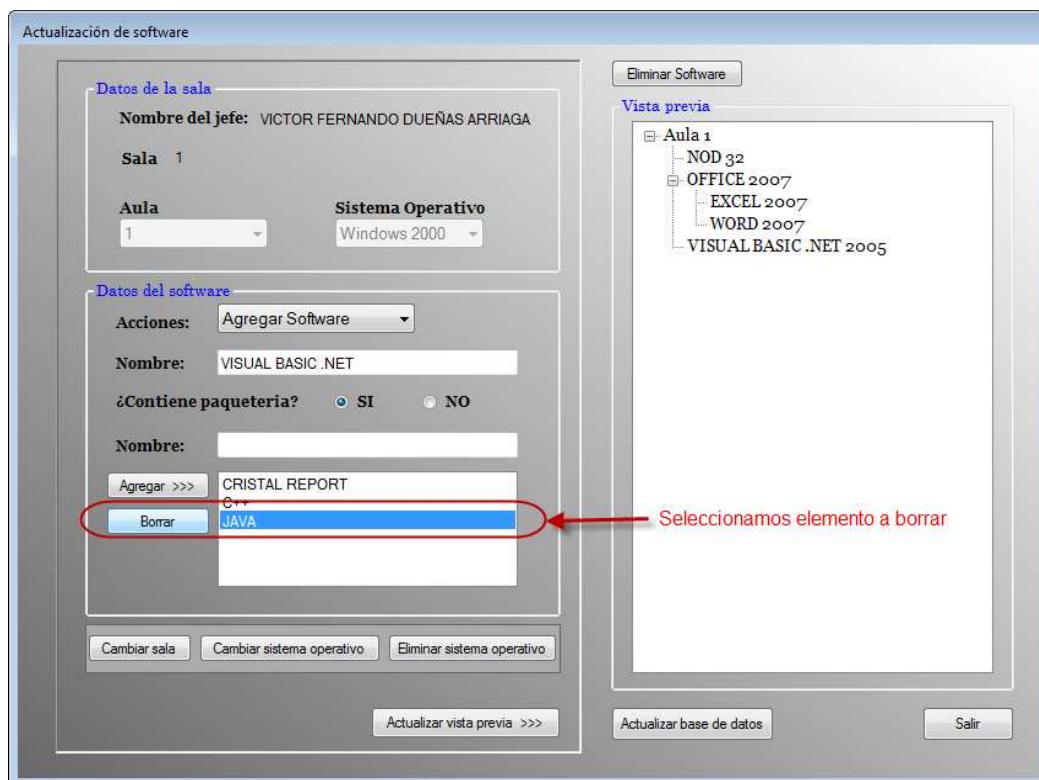


Figura 5.171 Elemento a borrar

C) Finalmente damos clic en el botón *Actualizar vista previa* y *Actualizar base de datos*, Figura 5.172, para hacer validos los cambios realizados, mostrándonos el siguiente mensaje, Figura 5.173.

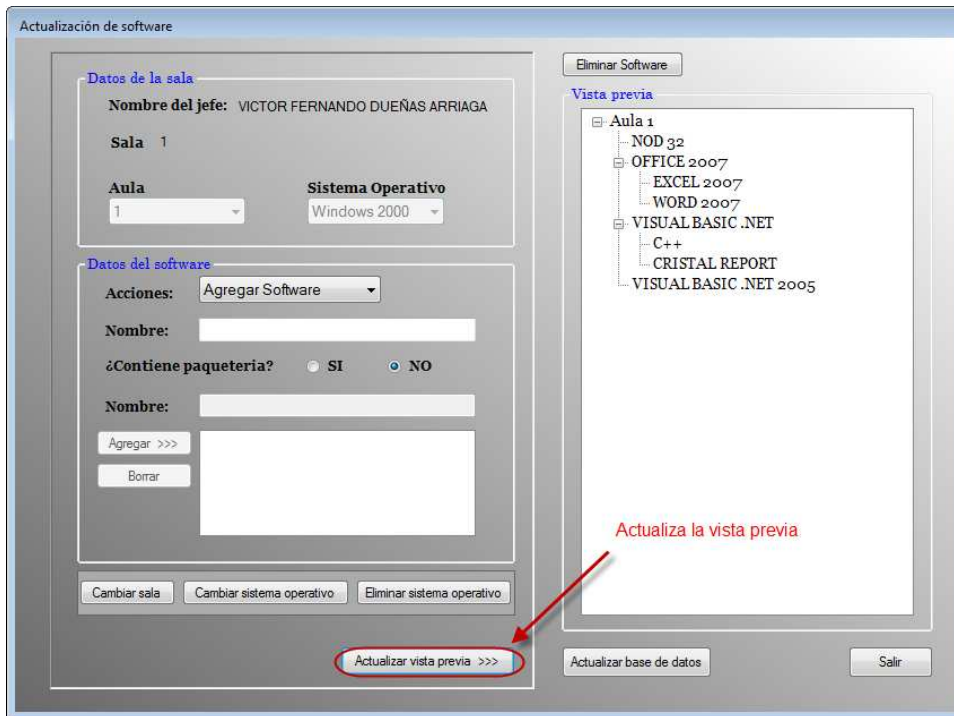


Figura 5.172 Vista previa actualizada

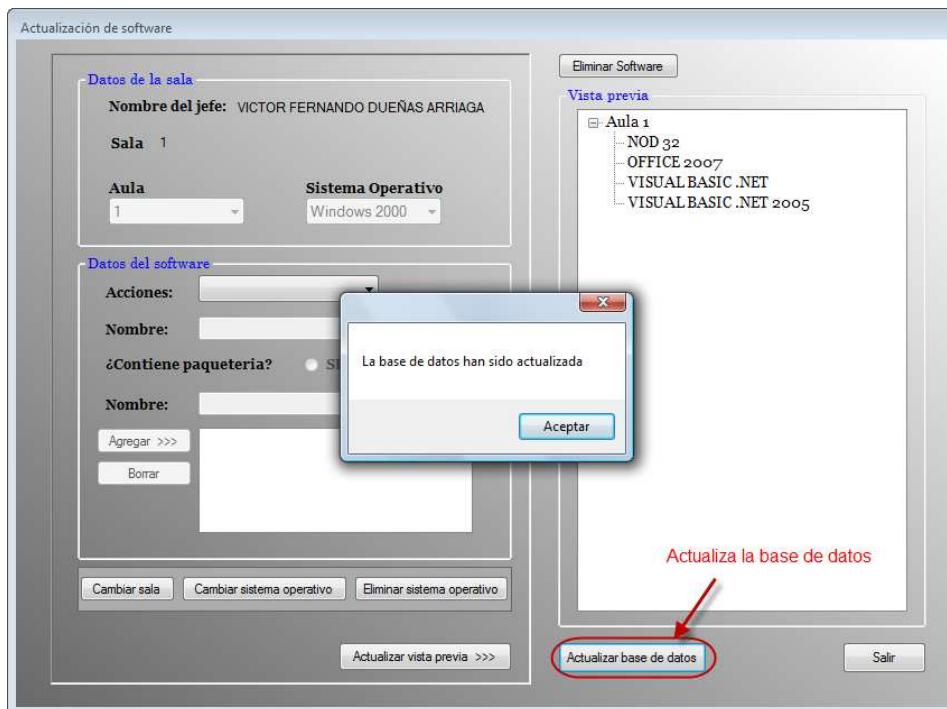


Figura 5.173 Actualiza la base de datos mostrando un mensaje al usuario.

Si por otra parte quisiéramos eliminar algún elemento que tengamos en la vista previa por ejemplo, en el caso anterior contamos con dos software llamados VISUAL BASIC .NET, podríamos eliminar alguno de ellos, esto lo realizamos seleccionando el que queramos eliminar y posteriormente damos clic sobre el botón *Eliminar Software* y automáticamente nos actualiza la vista previa con el cambio realizado y por lo tanto, lo único que hay q hacer es actualizar la base de datos.

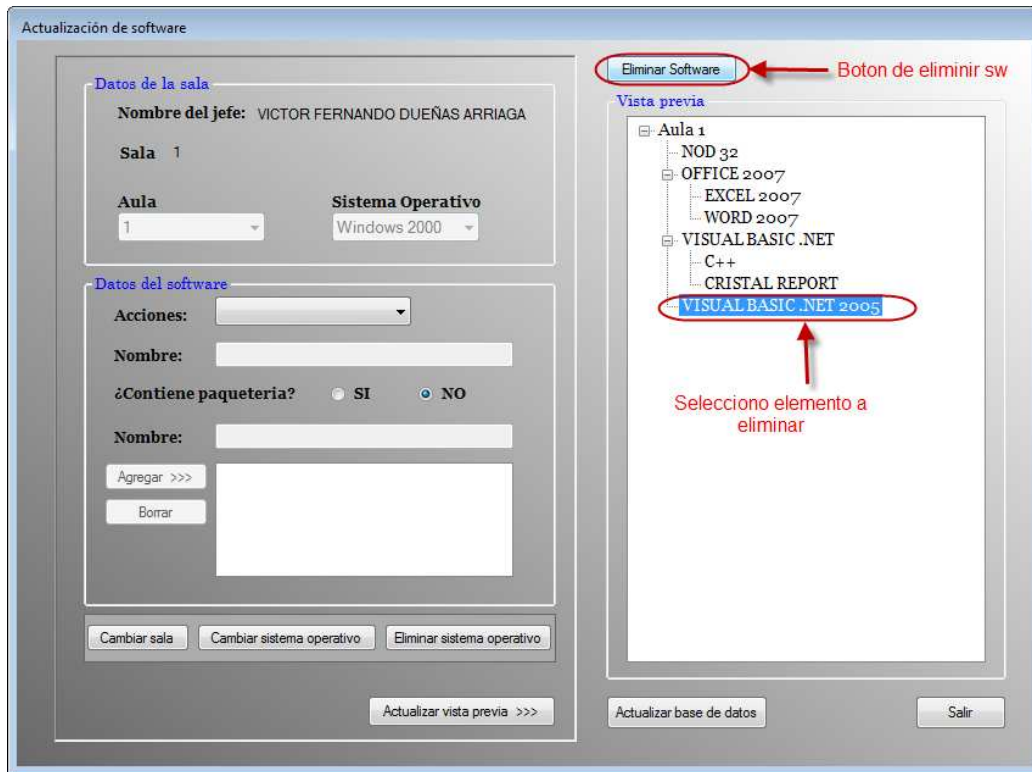


Figura 5.174 Eliminar elemento de software

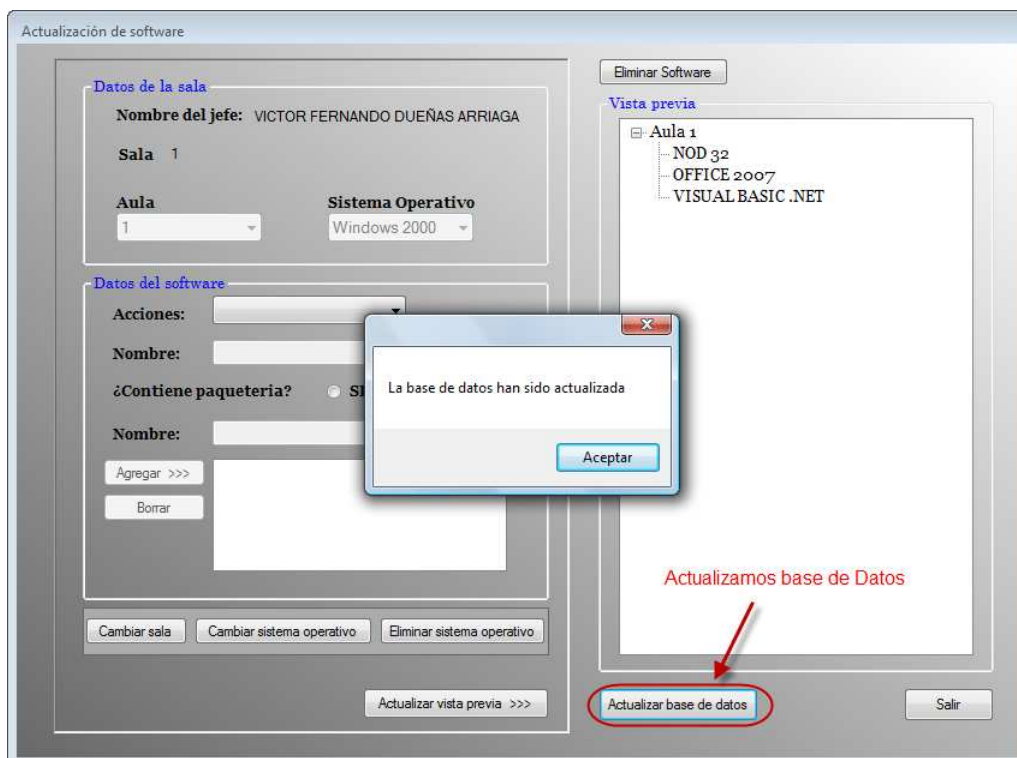


Figura 5.175 Actualizar base de datos.

Finalmente si al actualizar o bien agregar software nos equivocamos de Aula o de sistema operativo, lo único que hay que hacer es dar clic en la parte inferior izquierda en lo botones de *Cambiar sala* o *Cambiar sistema operativo* y nos mostrara un mensaje como el siguiente:

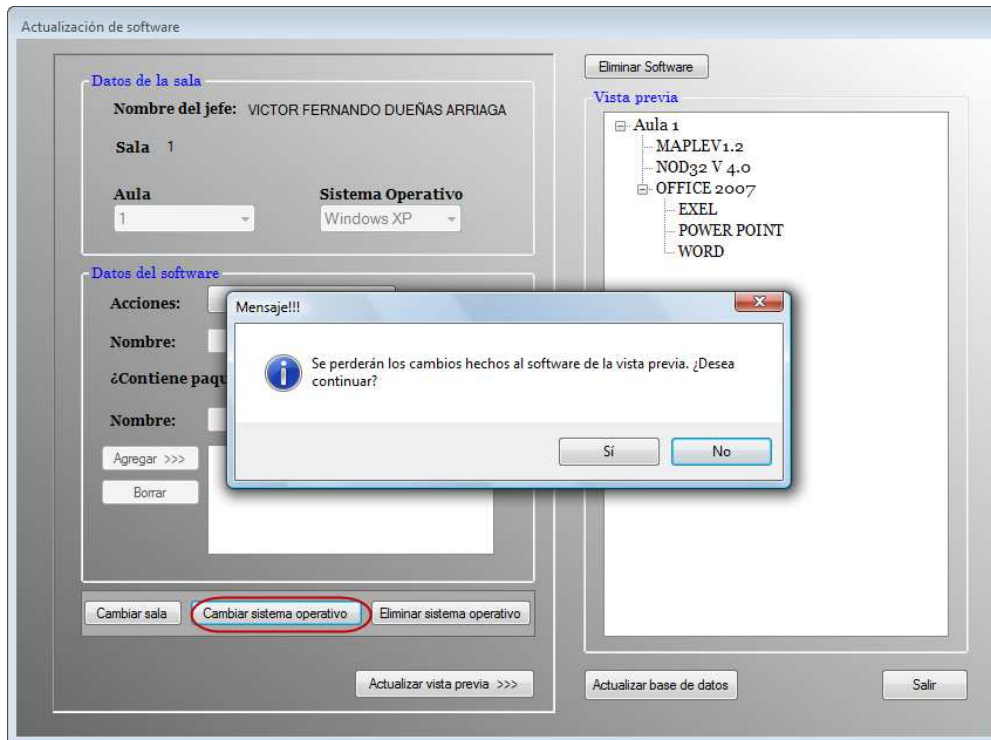


Figura 5.176 Mensaje para cambiar Sistema operativo o cambiar sala.

En caso de querer eliminar algún sistema operativo del aula, lo que hay que hacer es dar clic en el botón de *Eliminar sistema operativo* donde antes de eliminarlo nos mandara un mensaje de advertencia que nos dice si en verdad deseamos eliminar el sistema operativo.

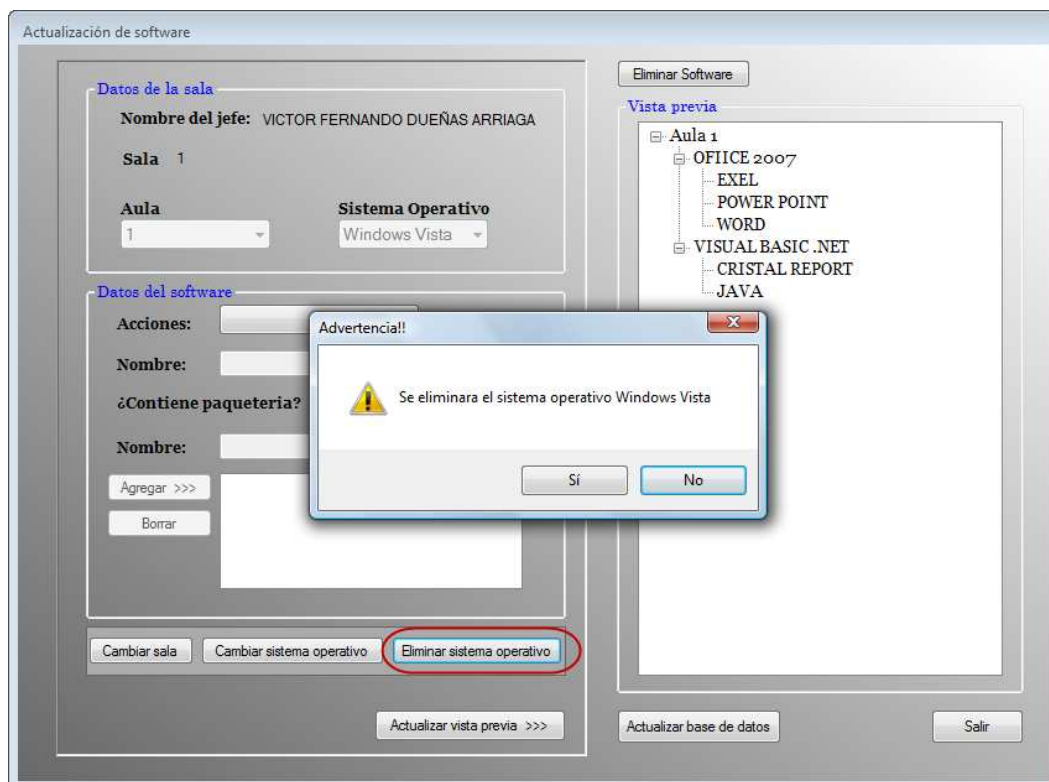


Figura 5.177 Eliminar sistema operativo

5.1.2.2.5 Agregar software nuevo

Esta función, nos sirve para restablecer por completo toda la paquetería de una sala, es decir, si ya teníamos cierta paquetería en alguna de las aulas con algún sistema operativo lo cambia por completo, borra lo que tenía anteriormente y coloca lo que se está agregando en ese momento. Para poder ingresar seleccionamos la opción *Agregar software nuevo* del menú *Paquetería*, Figura 5.178.

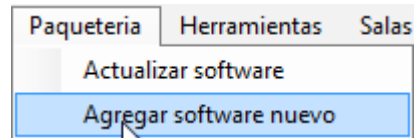


Figura 5.178

Una vez seleccionada la opción nos muestra una ventana como la de la siguiente figura.

 A screenshot of the 'Agregar software nuevo' window. The window is divided into several sections:

- Datos de la sala / Datos Gral. de la Sala:** Includes a text field for 'Nombre del jefe' (VICTOR FERNANDO DUEÑAS ARRIAC), a text field for 'Sala' (1), a dropdown for 'Aula', and a dropdown for 'Sistema Operativo'.
- Datos del software / Datos del Software:** Includes a text field for 'Nombre:', radio buttons for '¿Contiene paquetería?' (SI and NO), another text field for 'Nombre:', and buttons for 'Agregar >>>' and 'Borrar'.
- Vista previa:** A large white area with the text 'Vista previa del software nuevo' and 'Agrega los nuevos cambios en el software'.
- Opciones para cambiar software / Datos almacenados:** Includes buttons for 'Cambiar aula', 'Cambiar sistema operativo', and 'Quitar Software'. Below this is a section for 'Datos almacenados' with a text field and a 'Vista de los datos almacenados' label.
- Buttons:** 'Aceptar' and 'Salir' buttons are located at the bottom right.

 Red arrows point to the 'Agregar a la vista previa >>>' button, the 'Aceptar' button, and the 'Salir' button.

Figura 5.179 Escenario principal de Agregar software nuevo.

Este Form tiene básicamente el mismo funcionamiento que el Form de *Actualizar software*, pero en este caso solo agrega el software restaurando por completo la paquetería. Por ejemplo cambiemos la configuración actual del software de la sala 1 en el aula 1 con sistema operativo Windows 2000, dicha configuración la podemos ver en con la opción *Actualizar software* tal como lo muestra la Figura 5.180.

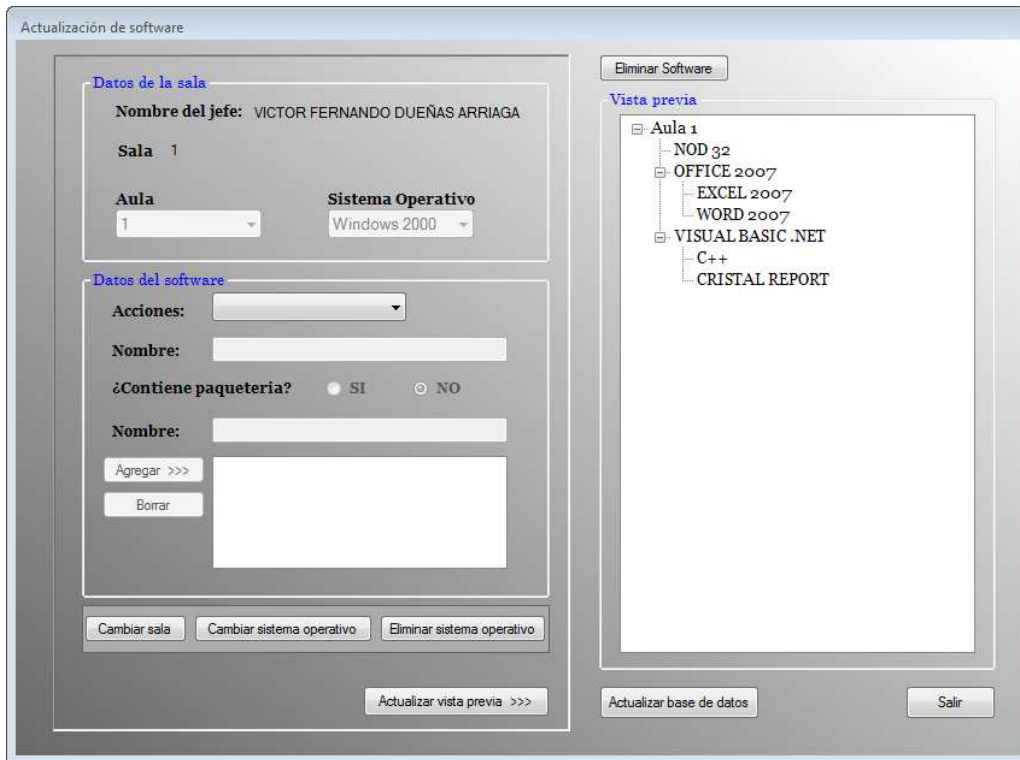


Figura 5.180 Configuración de la sala1 aula 1 con S.O. Windows 2000.

A) Primero que nada elegimos el aula 1 con sistema operativo Windows 2000 y empezamos a agregar el software y su paquetería en caso de que contenga. Si queremos agregar el sistema operativo antes de haber seleccionado el aula del ComboBox *Aula* o empezar a agregar la paquetería antes de colocar el nombre del software principal en el TextBox *Nombre*, nos mandara un mensaje de alerta, ya sea para agregar el aula (Figura 5.181) o bien, para colocar el nombre del software (Figura 5.182).

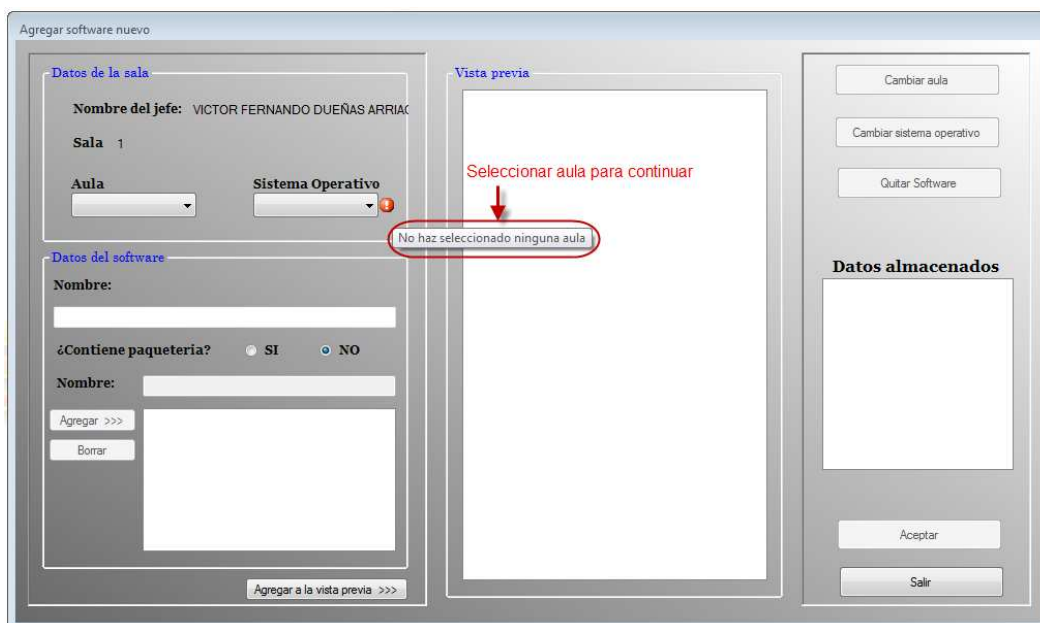


Figura 5.181 Seleccionar aula para continuar.

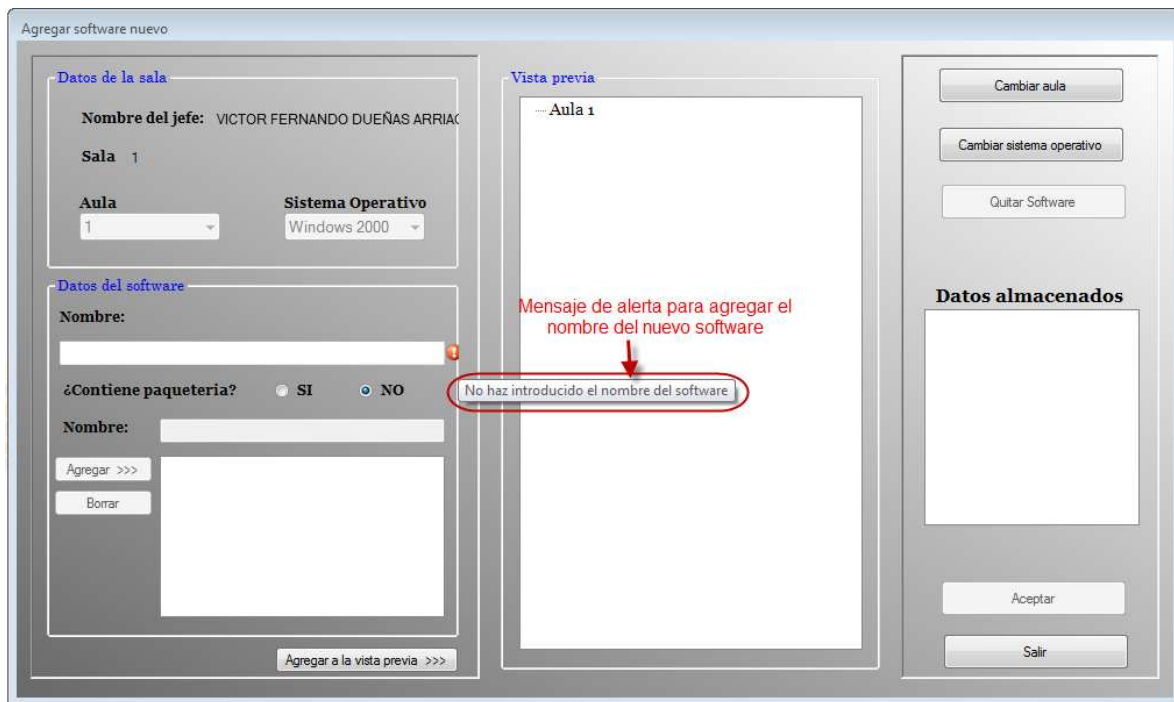


Figura 5.182 Mensaje para agregar en nombre del nuevo software antes de continuar.

En la siguiente figura vemos como se ha colocado la paquetería correspondiente del software Office 2003 una vez seleccionado el sistema operativo Windows 2000.

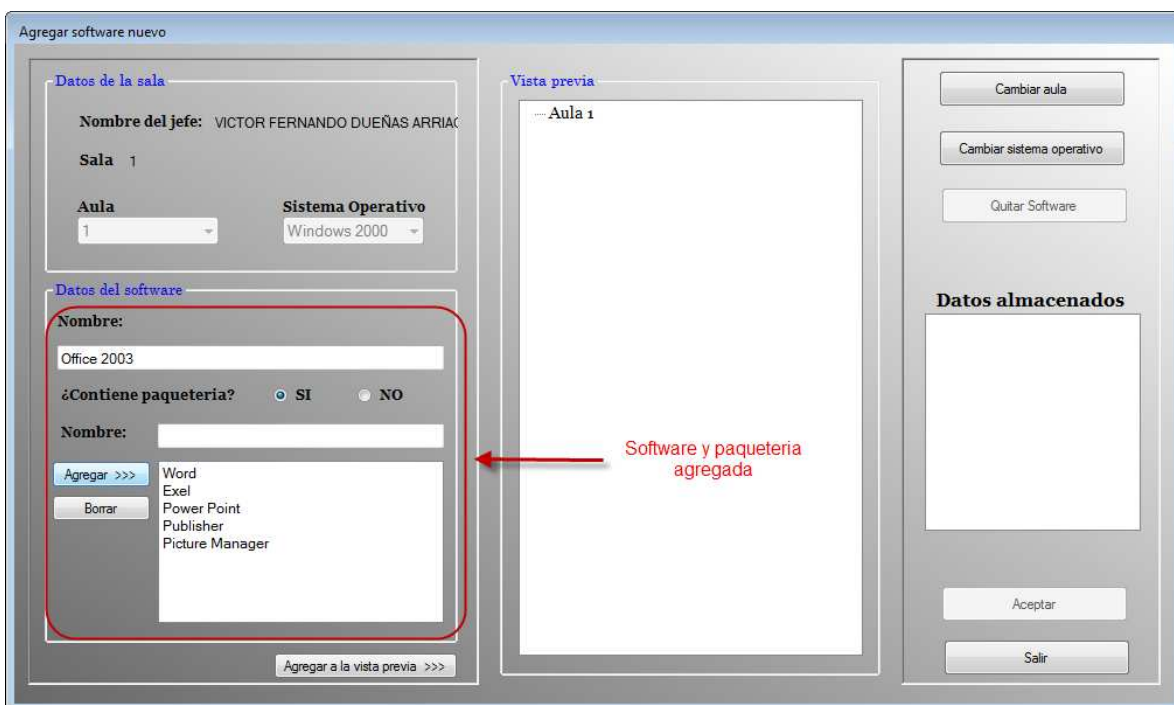


Figura 5.183 Software agregado

B) Después de haber colocado el software damos clic en el botón *Agregar a la vista previa* para poder visualizarlo, Figura 5.184, y así sucesivamente con el nuevo software que deseamos agregar.

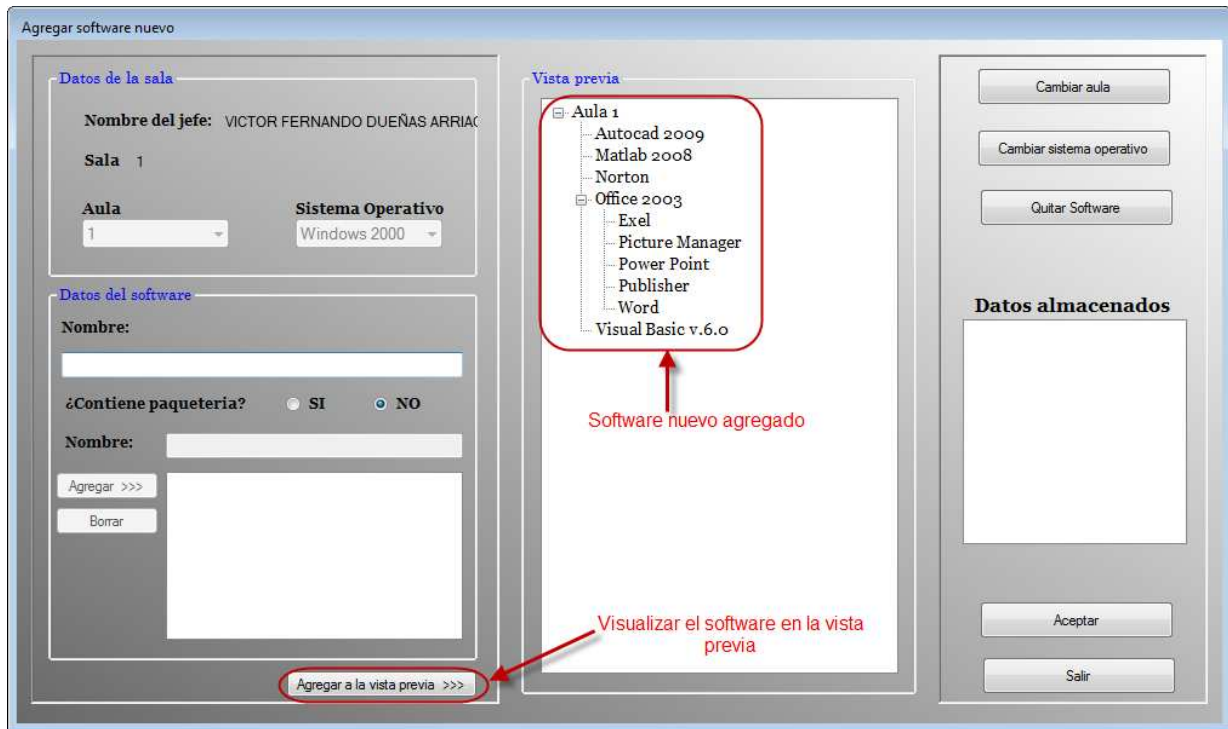


Figura 5.184 Nuevo software agregado.

C) Finalmente si deseamos que se apliquen los cambios realizados damos clic en el botón de *Aceptar* mostrándonos un mensaje de que si ya existían datos anteriormente estos serán eliminados para insertar los nuevos, Figura 5.185 y nos mostrara en el cuadro de *Datos almacenados* el aula con el sistema operativo que le colocamos, Figura 5.186.

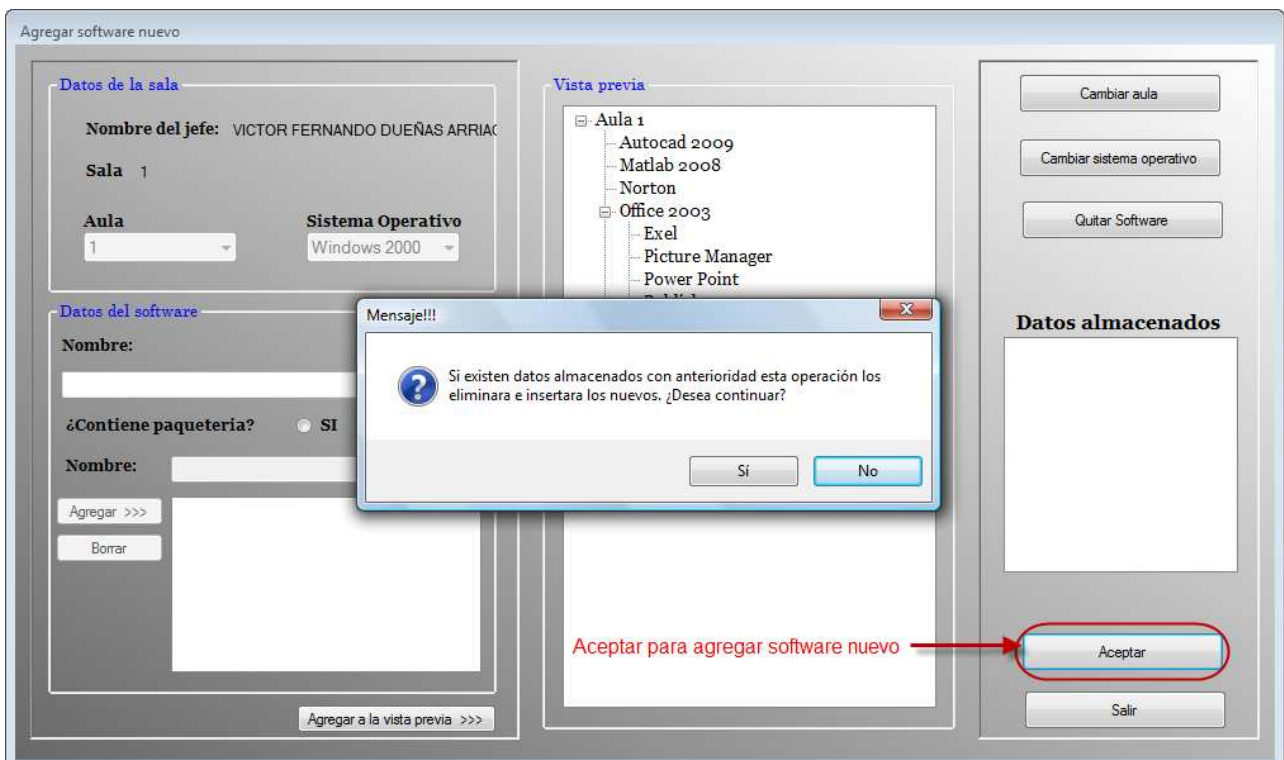


Figura 5.185 Agregando software nuevo

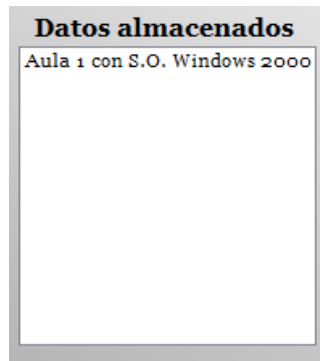


Figura 5.186 Datos almacenados.

D) Finalmente podemos revisar si se realizaron los cambios seleccionando la opción *Actualizar software*, en el cual elegiremos el aula junto con el sistema operativo correspondiente, Figura 5.187.

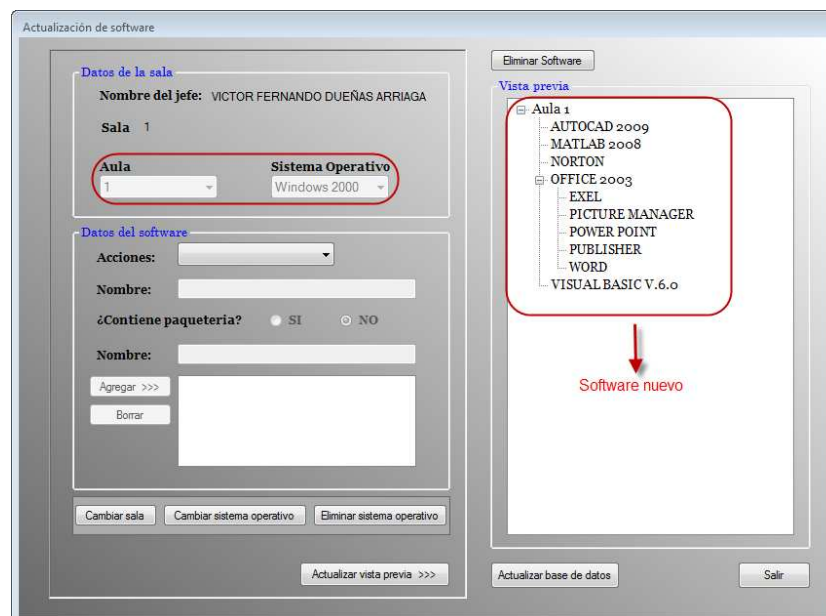


Figura 5.187 Software Nuevo en la sala 1, aula 1, con sistema operativo Windows 2000.

5.1.2.2.6 Cursos

Si queremos utilizar esta función tenemos que seleccionar la opción *Cursos* del menú *Herramientas*, Figura 5.188.

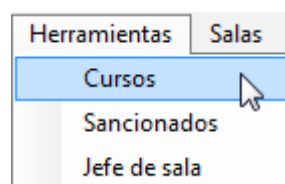


Figura 5.188

Enseguida nos muestra un Form con los cursos que se encuentran programados en todas las salas de única, dicho Form contiene una tabla con el identificador de curso, el nombre del curso a impartir, la fecha de inicio y de terminación del mismo, la hora de inicio y de fin, en que sala y aula se impartirá.

Id	Curso	Fecha de inicio	Fecha de fin	Hora de inicio	Hora de fin	Sala	Aula
39	ADO .NET	02/09/2008	09/09/2008	19:00:00	20:00:00	3	H
40	JSP'S	03/09/2008	12/09/2008	14:00:00	16:00:00	3	H

Figura 5.189 Cursos que se impartirán.

A parte de mostrar los cursos tiene la opción de eliminar el curso cuando este termine o antes, si es necesario. Para eliminar uno de los cursos que nos muestra en la tabla, lo que debemos hacer es seleccionar la casilla correspondiente y dar clic en el botón Eliminar como se muestra a continuación.

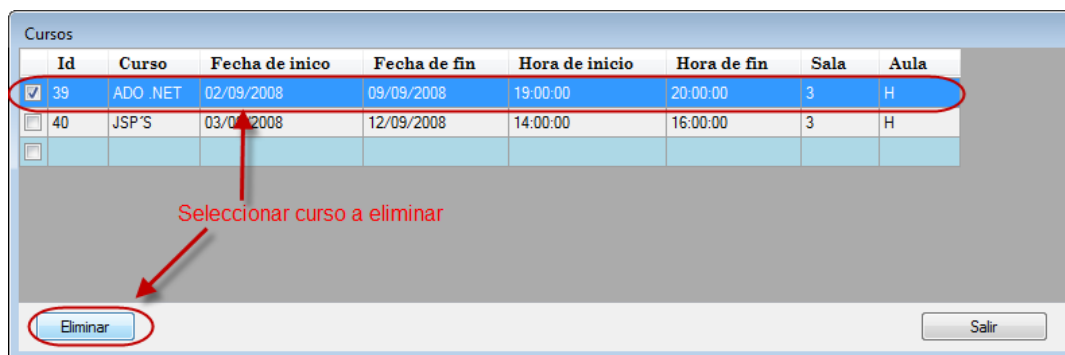


Figura 5.190 Eliminar curso.

Una vez que deseamos eliminar el curso nos mostrara la advertencia de que si queremos eliminar o no el curso, si la respuesta es Sí, procederá a eliminarlo, en caso contrario se cancela la acción.

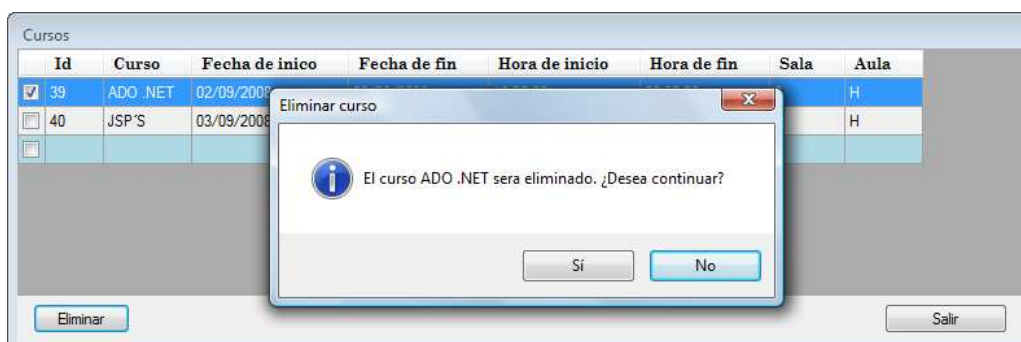


Figura 5.191 Mensaje de advertencia.

5.1.2.2.7 Sancionados

Para utilizar esta función seleccionamos la opción *Sancionados* del menú *Herramientas*, Figura 5.192.

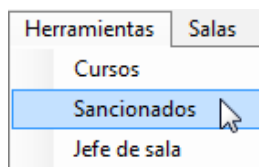


Figura 5.192

Nos muestra un Form con los alumnos que se encuentran sancionados, mostrándonos información como el número de cuenta, nombre y carrera a la que pertenece el alumno sancionado, el motivo por el cual fue sancionado, el responsable o persona quien sanciona, la fecha de la sanción y la terminación de la misma (Devolución), Figura 5.193.

Num. Cuenta	Nombre	Carrera	Motivo	Responsable	Sanción	Devolución
<input checked="" type="checkbox"/> 409004038	CARLOS XAVIER JIMENEZ LOPEZ	ING. PETROLERA	PRESTAMO (USUARIO)	GONZALO DEL SEMSTRE	09/02/2009	04/05/2009
<input type="checkbox"/> 306036682	ANA JESSICA TORRES PEREZ	ING. PETROLERA	PRESTAMO (DUEÑO)	GONZALO DEL SEMESTYRE	09/02/2009	04/05/2009
<input type="checkbox"/> 305141189	ROBERTO DANIEL REYNA AYALA	ING. EN COMPUTACIÓN	YOUTUBE, JUEGOS O VIDEOS	GERARDO	23/02/2009	16/03/2009
<input type="checkbox"/> 306113981	ALDO JAIR ORTEGA BELLO	ING. EN COMPUTACIÓN	MAS DE 15 HOJAS	ESPE	25/02/2009	11/03/2009
<input type="checkbox"/> 99051435	JUAN CARDENAS HERNANDEZ	ING. EN COMPUTACIÓN	AMBOS LADOS DE LA HOJA	BRENDA	25/02/2009	11/03/2009
<input type="checkbox"/> 303076300	RICHARD EMERSON LAZCANO BARAJAS	ING. EN COMPUTACIÓN	YOUTUBE, JUEGOS O VIDEOS	GERARDO	27/02/2009	20/03/2009
<input type="checkbox"/> 304188802	JORGE GARCIA GOMEZ	ING. CIVIL	COLOR, REHUSO, ACETATO	BRENDA	04/03/2009	18/03/2009
<input type="checkbox"/> 304156249	FERNANDO JIMENEZ TOPETE	ING. PETROLERA	YOUTUBE, JUEGOS O VIDEOS	BRENDA	04/03/2009	25/03/2009
<input type="checkbox"/> 301108225	JORGE ROBERTO TORRES JIMENEZ	ING. INDUSTRIAL	AMBOS LADOS DE LA HOJA	INGRID	06/03/2009	20/03/2009
<input type="checkbox"/> 306662166	RICARDO FERAT DURAN	ING. INDUSTRIAL	PRESTAMO (DUEÑO)	BRENDA	09/03/2009	01/06/2009

Figura 5.193 Formulario Sancionado.

También cuenta con la opción de eliminar un sancionad, es decir, quitar la sanción antes de que termine, lo cual quedara a consideración del jefe de la sala. Para eliminar un alumno sancionado que nos muestra en la tabla, debemos seleccionar la casilla correspondiente y dar clic en el botón Eliminar mostrándonos un mensaje preguntándonos si en verdad lo deseamos eliminar, si la respuesta es Sí, procede con a eliminarlo, en caso contrario cancela la acción, Figura 5.194.

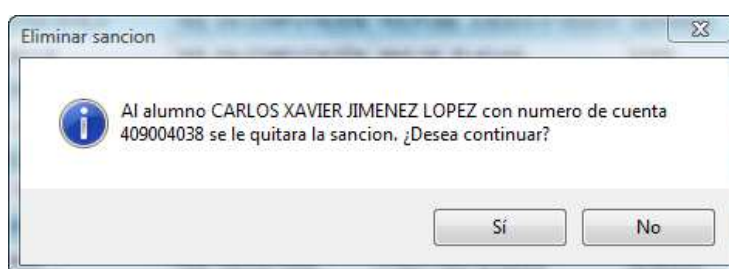


Figura 5.194 Mensaje de advertencia

5.1.2.2.8 Jefe de salas

Opción que muestra los datos generales del jefe de la sala. Para entrar seleccionamos *Jefe de sala* del menú *Herramientas*, Figura 5.195.

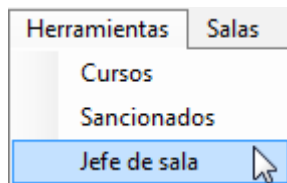


Figura 5.195

Este Form cuenta con la opción de actualizar los datos del jefe de la sala y de salir del formulario.

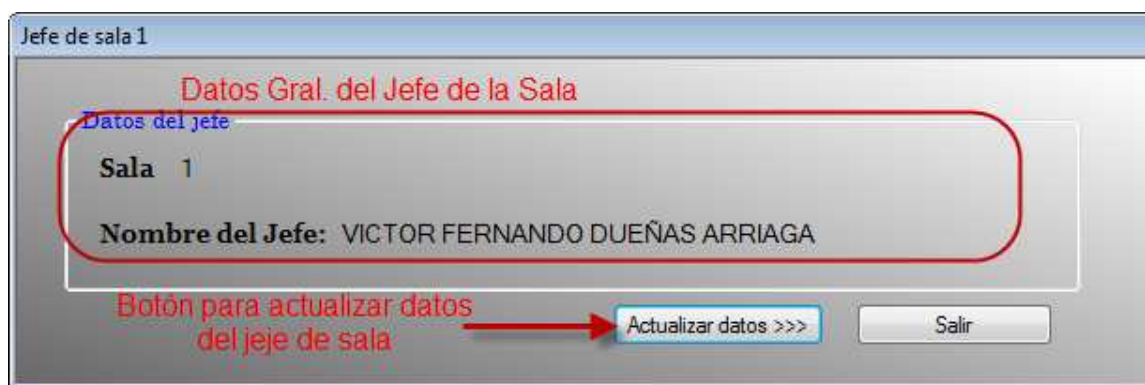


Figura 5.196 Submenú Jefe de sala.

Si queremos actualizar los datos del jefe de la sala, solo hay que dar clic en el botón *Actualizar datos* y nos pedirá la contraseña actual para mostrarnos la parte donde cambiaremos los datos.

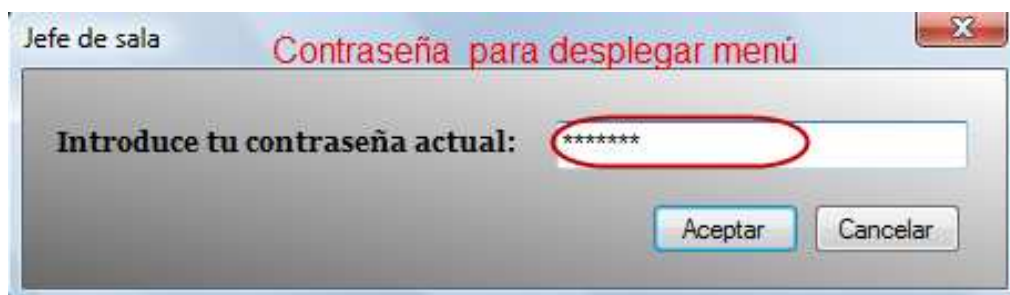


Figura 5.197 Ingresar contraseña del jefe de sala.

Una vez ingresado la contraseña correctamente nos desplegara dicho menú, el cual muestra los datos que podemos actualizar del jefe de sala, como lo es su nombre completo y la contraseña del mismo.

The screenshot shows a web application window titled 'Jefe de sala 1'. It contains two main sections. The top section, titled 'Datos Gral. del Jefe', has a sub-header 'Datos del jefe' and contains the text 'Sala 1' and 'Nombre del Jefe: VICTOR FERNANDO DUEÑAS ARRIAGA'. Below this are two buttons: 'Regresar <<<' and 'Salir'. The bottom section, titled 'Actualizar datos', has a sub-header 'Datos que se decean actualizar del Jefe de sala' and contains five input fields: 'Apellido paterno:', 'Apellido materno:', 'Nombre:', 'Contraseña:', and 'Confirmar contraseña:'. To the right of these fields are two buttons: 'Registrar' and 'Actualizar'.

Figura 5.198 Actualizar datos del jefe de sala.

Cabe mencionar que en los campos que pertenecen al nombre del jefe no permite agregar números solo letras y en caso de no haber llenado los campos correspondientes, y dar clic en actualizar o registrar, El sistema nos muestra un mensaje de advertencia indicando el error cometido, Figura 5.199.

This screenshot shows the same 'Actualizar datos' form as in Figure 5.198, but with the input fields filled with text: 'Apellido paterno: Dueñas', 'Apellido materno: Tello', and 'Nombre: Victor'. The 'Contraseña:' and 'Confirmar contraseña:' fields are empty. A red error message 'Mensaje para llenar los campos vacios' is displayed, with an arrow pointing to a red-bordered box containing the text 'Debes llenar el campo'. The 'Registrar' button is highlighted with a dashed border, and the 'Actualizar' button is disabled.

Figura 5.199 Llenar campos completamente.

Además la nueva contraseña será confirmada en el campo *Confirmar contraseña*, en caso de no ser iguales el sistema mostrara un mensaje indicando el error cometido.

Entonces si los datos fueron llenados correctamente al dar clic en *Actualizar* y *Registrar*, nos mostrara un mensaje confirmando que los datos han sido actualizados, Figura 5.200.

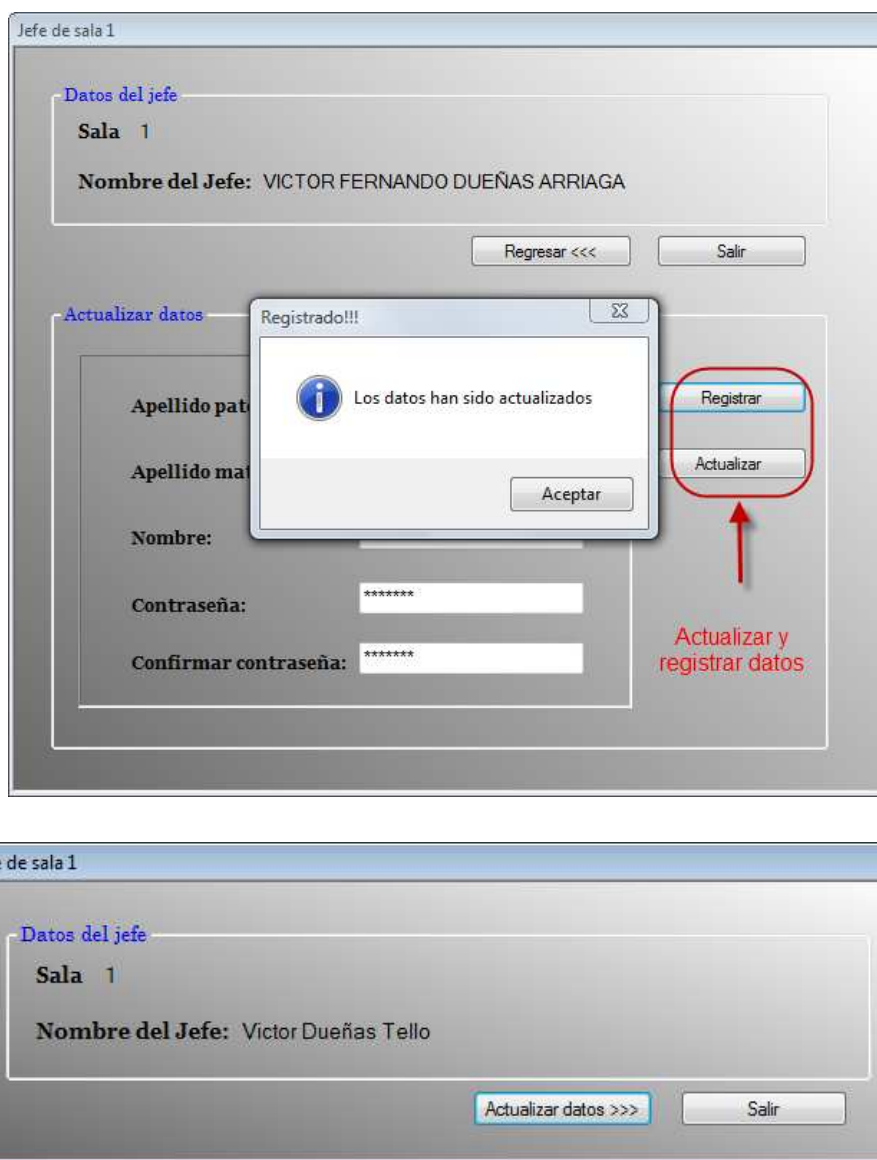


Figura 5.200 Datos actualizados.

5.1.2.2.9 Sala 3

En ciertas ocasiones es necesario saber el estado en que se encuentran las computadoras de las diferentes salas de cómputo de UNICA, para ello SCOSU v2 Modulo del Jefe cuenta con cuatro Form, los cuales representan a las cuatro salas de cómputo. Para fines de este manual solo se mencionara el funcionamiento del Form que representa la Sala 3, puesto que los demás funcionan de la misma manera.

Para poder ingresar seleccionamos la opción *Sala 3* del menú *Salas*, Figura 5.201.

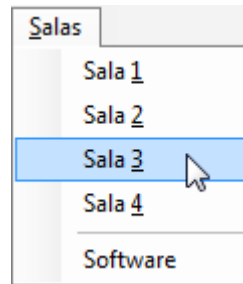


Figura 5.201

Mostrándonos la siguiente ventana.



Figura 5.202 Estados de los equipos de la Sala 3

Se puede observar los diferentes estados de las computadoras, los cuales se actualizan cada 30 segundos. Si queremos saber el número de equipo colocamos el puntero del mouse sobre dicho equipo, Figura 5.203.



Figura 5.203 Número de equipo

El número de equipo aparecerá de la misma manera para la Sala 1 y 2, en la Sala 4 no aparece dado que ya están presentes desde el inicio.

Para salir damos clic sobre el botón *Salir* con lo cual el sistema nos mostrar un mensaje preguntando que si en realidad queremos hacerlo, si la respuesta es Sí, se cierra el Form en caso contrario se cancela la acción.

5.1.2.2.10 Software

Esta opción nos muestra un Form en donde podemos hacer una búsqueda rápida del software con el que se cuenta en las salas, como se muestra en la siguiente figura.

Busqueda de Software

Introduce el nombre del software que deseas buscar (completo o parcial)

Aceptar

Resultados

Software	Sala	Aula	Sistema operativo
----------	------	------	-------------------

Salir

Buscar software

Area de resultados obtenidos de la busqueda

Figura 5.204 Búsqueda de software

Para poder ingresar seleccionamos la opción Software del menú Salas, Figura 5.205.

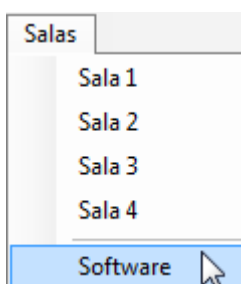


Figura 5.205

Si queremos realizar la búsqueda de algún software debemos introducir su nombre ya sea completo o parcial y dar clic en el botón *Aceptar*. Si no se coloca el nombre y damos clic en *Aceptar* el sistema muestra un mensaje de advertencia indicando que debe de introducirse el nombre del software. Por otra parte si no se encuentran coincidencias en la base de datos, es decir, no existe al archivo, nos mostrara un mensaje indicándolo, Figura 5.206.

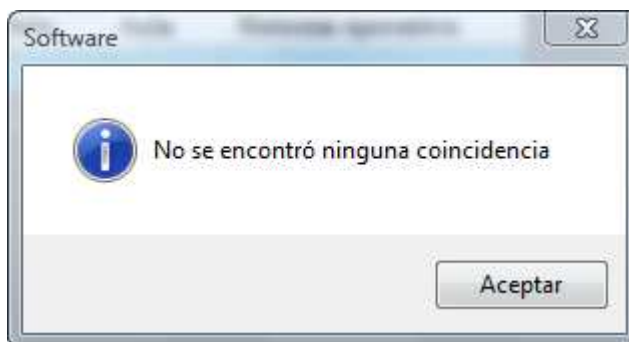


Figura 5.206

Una vez colocado el software, nos mostrara los resultados en una tabla, la cual contiene datos como el nombre de todos los software que coincidieron con el texto que introdujimos, la sala donde se encuentra, el aula donde está instalado y en qué sistema operativo esta, Figura 5.207.

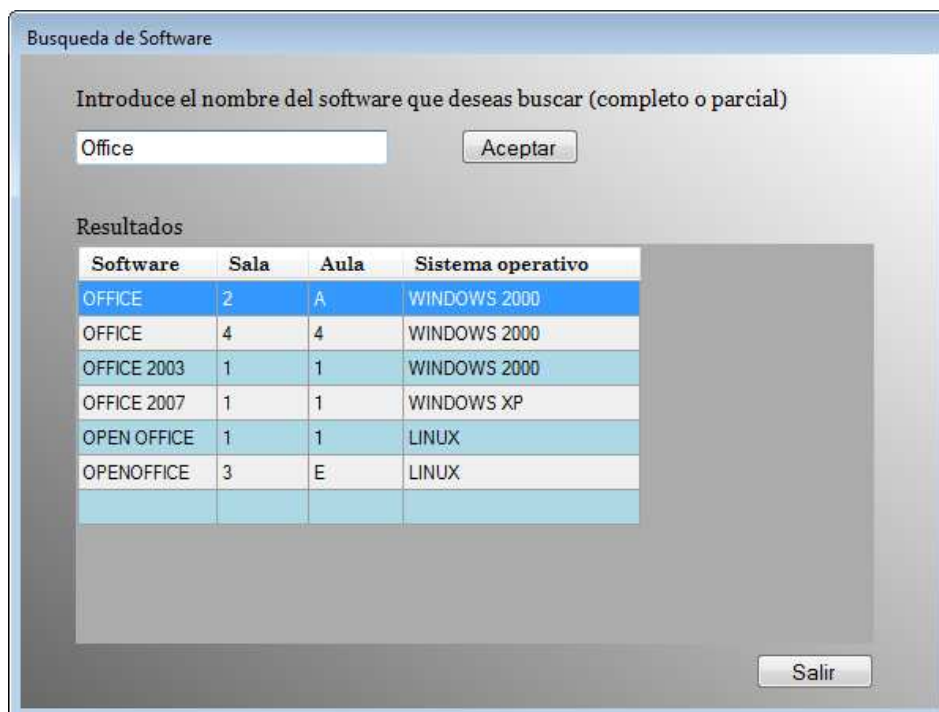


Figura 5.207 Buscar Software

Para salir damos clic sobre el botón *Salir* cerrando el Form actual.

5.2 Capacitación

Una correcta capacitación resultara en un correcto funcionamiento del sistema, para el caso de SCOSU v2 la capacitación se realiza mediante dos métodos, el primero se realiza con un curso donde se explica su funcionamiento y dependiendo de la persona a la que se dirija, el curso será a nivel administrativo o a nivel usuario; el segundo es mediante los manuales con los que cuenta SCOSU v2 (manual de usuario y de administrador) donde podrá consultar cualquier duda que tenga.

5.2.1 Capacitación del administrador

El curso dirigido al administrador del sistema tiene una duración de cinco días hábiles de tres horas cada uno. En él se ven, aspectos como el paradigma de la programación orientada a objetos y el funcionamiento de los eventos, los dos pilares de la programación de SCOSU v2, el modelado del sistema, el esquema de la base de datos así como su diseño y ADO .Net como acceso a los registros de la base de datos.

Una vez terminado el curso el administrador del sistema puede realizar modificaciones o ampliar las funciones de SCOSU v2, si así se requiere, además podrá consultar el manual del programador en caso de que se tenga alguna duda.

5.2.2 Capacitación de los usuarios

La capacitación está dirigida a los dos tipos de usuarios del sistema:

- **Personal de salas:** Estos usuarios son los responsables de utilizar el sistema SCOSU v2, para poder asignar computadoras a los alumnos que así lo soliciten, sancionarlos en caso de que cometan alguna falta al reglamento de UNICA, apartar las aulas para los cursos, entre otras cosas.
- **Jefes de sala:** Son los que llevan el control en cada una de las salas de UNICA y son los encargados de utilizar el sistema SCOU v2 Modulo del Jefe que les permite realizar funciones exclusivas de su cargo, colocar la paquetería de las salas, quitar sanciones a los alumno antes de tiempo y varias funciones más.

El curso varia en tiempo, dependiendo de a qué tipo de usuario se esté dando. Para el caso del Personal de salas, el curso tiene una duración máxima de tres horas, donde se explican todas las funciones con las que cuenta SCOSU v2 y que hacer en caso de que hubiera alguna falla, ya sea por error humano o por el propio sistema.

Además el usuario cuenta con un manual que podrá consultar en caso de que tenga alguna dificultad con el sistema (*véase el punto 5.1.2.1.18 Ayuda de SCOSU v2*), el cual podemos observar en el apartado 5.1.2.1 Manual del usuario de SCOSU v2.

El curso de los jefes de sala es de dos horas, en él, se abarcan todas las funciones de SCOSU v2 Modulo del Jefe, además de contar con su manual en caso de cualquier duda, dicho manual se encuentra en el apartado 5.1.2.2 Manual del usuario de SCOSU v2 Modulo del Jefe.

5.3 Instalación

La instalación la realiza el administrador del sistema antes de que comience el nuevo semestre o antes en caso de que sea necesario, en cada una de las cuatro salas de UNICA y el las computadoras de los jefes de sala.

SCOSU v2 se instalara en los equipos asignados para el control de salas y SCOSU v2 Modulo del Jefe en el equipo del jefe de sala, dichos equipos deberán de contar con los requerimientos mínimos señalados en el Capitulo 2 en el apartado 2.6 Infraestructura.

Para llevar a cabo la instalación del sistema se debe de seguir el procedimiento indicando en el apartado 5.1.1 Manual de instalación. En cuanto a la base de datos es necesario depurarla por lo menos cada semestre tomando en cuenta los siguientes puntos:

- Tabla *alumno* se limpia.
- Tabla *alumno_modulo* se limpia.
- Tabla *alumno_sancionado* se conservan los datos.
- Tabla *cancelar_sancion* se limpia.
- Tabla *computadora* se conservan los datos, solo se coloca el campo *id_estado* de todos los registros a 1 y el campo de descripción se limpia también.
- Tabla *computadora_so* se queda como esta, no se deberá de mover los registros puesto que son los sistemas operativos que se han agregado a las diferentes salas.
- Tabla *curso* conserva sus datos.
- Tabla *jefe_sala* se queda con sus datos.

- Tabla *modulo* se queda con sus datos.
- Tabla *registro_historico* se limpia.
- Tabla *reporte* se limpia.
- Tabla *resp_sancion* se queda con sus datos.
- Tabla *sala* se queda con sus datos.
- Tabla *sancion* conserva sus datos.
- Tabla *software* conserva sus datos.
- Tabla *tipo_estado* se queda con sus datos.
- Tabla *tipo_so* se queda con sus datos.
- Reiniciar las secuencias
 - id_cancelar_seq
 - id_curso_seq
 - id_resp_seq

5.4 Mantenimiento

Después de instalar el sistema SCOSU v2 en sus dos módulos se deberá de mantener en correcto funcionamiento, comprobando regularmente la aplicación, buscando condiciones de error y eventos inusuales. La revisión del sistema la realiza el administrador cada mes, garantizando su correcto funcionamiento.

Aun así, por muy cuidadoso que se sea programando la aplicación e identificando errores, habrá ocasiones en que surgirá algún problema propio del sistema u ocasionado por un problema externo, como puede ser la conexión al servidor, para estos casos SCOSU v2 cuenta con la opción de poder contactar al administrado vía correo electrónico mediante su función *Contactar al Administrador*, (para más detalles vea el apartado 5.1.2.1.19 Contactar al Administrador), donde podrá informar al administrador del error ocurrido y si es el caso, hasta adjuntar una imagen mostrando dicho error. Con esta opción se espera una óptima y rápida solución para los problemas que puedan surgir en el futuro.

CONCLUSIONES Y RESULTADOS

Conclusiones

Se cumplieron ampliamente los objetivos y resultados planteados.

SCOSU v2 es la evolución del Sistema de Control de Salas de UNICA (SCOSU). Realicé esta nueva versión utilizando el diseño conceptual, lógico y físico como parte de los métodos utilizados, con lo cual se lograron excelentes resultados.

Utilizando el estándar tecnológico de desarrollo de software de UNICA realicé el modelado del sistema con UML, debido a que nos permite representar visualmente las reglas de creación, estructura y comportamiento de un grupo relacionados de objetos y clases, nos ayuda a visualizar de manera eficiente la complejidad del sistema y nos permite realizar de una manera más ágil los cambios y nuevas especificaciones que surjan en un futuro, además una vez realizado el modelado, la programación se realiza enfocada a dicho modelo, ahorrando tiempo y esfuerzo.

Concluyo que:

- Creé un sistema capaz de controlar el acceso de los alumnos a las cuatro diferentes salas de computo de UNICA, solucionado los problemas del anterior sistema y obteniendo mejores resultados.
- Realicé un buen diseño de la base, con lo cual se evito la redundancia de datos, optimizando la consulta y modificación de registros por parte de SCOSU v2 y eliminando los errores generados por estas acciones.
- Aproveché las capacidades de la clase DataSet y así creé el Modo desconectado, el cual nos permite seguir utilizando a SCOSU v2 en caso de que se pierde la conexión con los servidores o exista una falla con la red, creando una base de datos interna alojada en el sistema.
- Gracias a que el sistema se dividió en dos módulos se repartieron las tareas de acuerdo al cargo que tienen en la sala, el personal de salas se encarga del acceso de los alumnos utilizando SCOSU v2 y el jefe de la sala realiza funciones que necesitan de su criterio utilizando SCOSU v2 Modulo del Jefe.
- El sistema es capaz de crear reportes detallados de los alumnos que han ingresado a las salas de computo de UNICA, estos reportes tienen datos relevantes como el

número de veces y las fechas en que ha ingresado a las salas en el semestre, el tiempo que utilizo el equipo cada vez que ingreso, el tiempo total de uso y más.

- Con la ayuda de SCOSU v2 Modulo del Jefe, el jefe de la sala puede registrar, dar de alta o cambiar el software que se requiera en sus salas.
- Se redujo el tiempo de atención al alumno agilizando su entrada a las salas de cómputo de UNICA.
- Restringe la entrada a los usuarios debido a que solo podrán ingresar a la sala que les corresponde, validando la ip del equipo donde se encuentra instalado el sistema.
- Los datos obtenidos y guardados en la base de datos son más detallados, un ejemplo de ello es que se sabe exactamente el tiempo (en minutos) que utilizo una computadora el alumno, esto ayuda a que las estadísticas solicitadas por el Departamento de Servicios Académicos (DSA) sean más precisas.
- El haber contado con el lenguaje UML para el modelado del sistema fue sumamente importante, gracias a él, la integración de SCOSU v2 con otros sistemas será una tarea más fácil de realizar.
- Realicé de forma detallada la documentación necesaria para que los futuros usuarios y administradores comprendan el sistema, la cual está conformada por los manuales de instalación, usuario y programador.
- El sistema posee su propio envío de correo electrónico para tener una constante comunicación con el administrador y poder informarle cualquier imprevisto.
- SCOSU v2 cuenta con una interfaz moderna y amigable facilitando su uso.
- Se integro el Sistema de Monitoreo de las Salas (SMS) con SCOSU v2.

Acciones a corto plazo

Una vez que se cumplieron con los requerimientos solicitados, el sistema entro en vigor instalándolo en las cuatro salas de cómputo de UNICA durante un periodo de prueba, lo cual trajo que los usuarios solicitaran algunas funciones extras, dichas funciones ya fueron tomadas en cuenta y se podrán implementar fácilmente dadas las características de SCOSU v2, además de contar con la documentación necesaria (manuales, modelados, diagramas).

SCOSU v2 se está integrando a otros sistemas con la finalidad de conseguir datos más detallados del uso de las salas de cómputo de UNICA, como es el Sistema de Control de Impresiones (SCI) el cual se encarga de administrar el número de impresiones que realizan los alumnos, los datos que obtiene van desde el número de impresiones que realiza un alumno a lo largo del semestre hasta el total de impresiones realizadas en cada una de las salas de cómputo de UNICA.

Se está capacitando a otros becarios de UNICA como administradores de SCOSU v2 para tener un mejor margen de respuesta si se llegara a presentar algún inconveniente.

Bibliografía

BARRAGÁN Paz, María del Rosario. Mayo 2005. Estándar de herramientas tecnológicas para el desarrollo de sistemas de carácter institucional de UNICA. México.

BLANCO, Luis M. 2002. Programación en Visual Basic .Net. Madrid, España. Ed. Grupo Eidos. 725 págs.

BLANCO, Luis M. 2003. Crystal Reports para Visual Studio .Net (Visual Basic .Net). Madrid, España. Ed. Grupo Eidos. 109 págs.

CHARTE, Francisco O. 2002. Base de datos con Visual Basic .Net. Madrid, España. Ed. Anaya Multimedia. 526 págs.

GUEVARA, Marco; Flores, César. 2002. AllFusion Erwin Data Modeler. Ed. Libros Digitales .Net. 67 págs.

PRESSMAN, Roger S. 2002. Ingeniería del Software: un enfoque práctico. 5ª.edición. España. Ed. Mcgraw-Hill. 601 págs.

SCHMULLER, Joseph. 2000. Aprendiendo UML en 24 horas. México. Ed. Prentice Hall. 404 págs.

SOM, Guillermo; Zorrilla, Unai; Serrano, Jorge. Mayo 2006. Visual Basic 2005. Baracaldo, España. 446 págs.

Mesografía

Wikipedia. Herramienta CASE.

<http://es.wikipedia.org/wiki/CASE>

Importancia de UML

<http://www.osmosislatina.com/lenguajes/uml/basico.htm>