

UNIVERSIDAD DE CUENCA

FACULTAD DE INGENIERIA

ESCUELA DE INFORMATICA

**DESARROLLO DE UN SISTEMA DE SEGURIDAD Y
CONTROL DE ACCESO PARA UN CENTRO DE CÓMPUTO.**

**Tesis previa la obtención del título de
Ingeniero de sistemas**

DIRECTOR:

Ing. María Fernanda Granda.

AUTOR:

Ricardo G. Guanga Cadme

**Cuenca – Ecuador
2005**

AGRADECIMIENTO

Mis sinceros agradecimientos a todas aquellas personas quienes colaboraron en la realización de esta tesis, de manera especial a la Ingeniera María Fernanda Granda.

DEDICATORIA

Esta tesis va dedicada a toda mi familia, y de manera especial a mi Padre y a la memoria de Madre.

La tesis aquí presentada fue realizada bajo la dirección del
Ing. María Fernanda Granda

Cada una de las ideas, criterios y opiniones vertidas en esta tesis
son de absoluta y exclusiva responsabilidad del autor

INDICE

1	INTRODUCCION	1
1.1	ANTECEDENTES	1
1.2	JUSTIFICACIÓN	2
1.3	OBJETIVOS	3
1.3.1	Objetivo General	3
1.3.2	Objetivos Específicos	3
2	MARCO TEÓRICO.....	5
2.1	SITUACION ACTUAL DEL CENTRO DE COMPUTO	5
2.1.1	Estructura de Salas y Equipos de Computacion	5
2.2	SISTEMAS DE IDENTIFICACION	9
2.2.1	Métodos de Identificación	9
2.2.2	Tecnología de Tarjetas	9
2.3	TECNICAS DE BLOQUEO DEL COMPUTADOR	12
2.3.1	Autenticación de Windows.....	13
2.3.2	Bloquear el Escritorio de Windows.....	18
3	ANALISIS DEL SISTEMA	24
3.1	INTRODUCCIÓN	24
3.2	DICCIONARIO DE DATOS.....	25
3.3	DIAGRAMA DE PAQUETES	26
3.4	DIAGRAMAS DE CASOS DE USO	29
3.4.1	Subsistema Administración del Centro de Cómputo.....	29
3.4.2	Subsistema Acceso al Sistema.....	42
3.4.3	Subsistema Comunicación.....	47
3.4.4	Subsistema Cliente	49
3.5	DIAGRAMA DE CLASES INICIAL	52
3.6	DIAGRAMAS DE INTERACCION	53
3.6.1	Gestión de Usuarios.....	53
3.6.2	Gestión de Máquinas o Computadores.....	57
3.6.3	Gestión Salas y Reservasiones.	60
3.6.4	Subsistema Acceso al Sistema.....	67
3.6.5	Subsistema Comunicación (Cliente – Servidor).....	74
3.6.6	Subsistema Cliente	78
4	DISEÑO DEL SISTEMA.....	78
4.1	INTRODUCCIÓN	78
4.1.1	Decisiones de Alto Nivel.....	78
4.2	DICCIONARIO DE CLASES	80
4.2.1	Subsistema Administración del Centro de Cómputo.....	80
4.2.2	Subsistema Acceso al Sistema.....	83
4.2.3	Subsistema Comunicación.....	84
4.2.4	Subsistema Cliente	84
4.3	DISEÑO DEL SUBSISTEMA ADMINISTRACIÓN DEL CENTRO DE COMPUTO....	86
4.3.1	Arquitectura del Subsistema.....	86

4.3.2	Diagrama de Clases del Subsistema.	87
4.4	DISEÑO DEL SUBSISTEMA ACCESO AL SISTEMA.	88
4.4.1	Arquitectura del Subsistema.	88
4.4.2	Diagrama de Clases del Subsistema.	88
4.5	DISEÑO DEL SUBSISTEMA COMUNICACION.	89
4.5.1	Arquitectura del Subsistema.	89
4.5.2	Diagrama de Clases del Subsistema.	89
4.6	DISEÑO DEL SUBSISTEMA CLIENTE.	90
4.6.1	Arquitectura del Subsistema.	90
4.6.2	Diagrama de Clases del Subsistema.	90
4.7	DISEÑO DE LA BASE DE DATOS.	91
5	IMPLEMENTACION DEL SISTEMA	93
5.1	HERRAMIENTAS UTILIZADAS.....	93
5.1.1	Microsoft.NET.....	93
5.1.2	Otras Herramientas Utilizadas.....	109
5.2	DIAGRAMAS DE IMPLEMENTACION	110
5.2.1	Diagrama de Componentes.....	110
5.2.2	Diagrama de Despliegue.....	111
6	MANUAL DEL USUARIO	113
6.1	SISTEMA DE SEGURIDAD Y CONTROL DE ACCESO (SERVIDOR).113	
6.2	INICIANDO CON EL SISTEMA	116
6.3	MENUS DEL SISTEMA	126
6.4	HERRAMIENTAS	140
7	CONCLUSIONES Y RECOMENDACIONES	144
7.1	CONCLUSIONES.	144
7.2	RECOMENDACIONES.....	145
8	BIBLIOGRAFIA	147

1. INTRODUCCION



1 INTRODUCCION

1.1 ANTECEDENTES

En la actualidad los Centros Educativos como universidades, colegios, escuelas, se ven en la obligación de capacitar al alumnado en el área de informática por lo que han tenido que crear los Centros de Cómputo o Laboratorios de Computación para así dotar de todos los servicios que la rama de la computación ofrece a estudiantes, profesores y personal administrativo.

En la Facultad de Ingeniería de la Universidad de Cuenca se instaló un Centro de Computo, el mismo que funciona desde hace algunos años en el cuál se ha puesto a disposición de los estudiantes toda la infraestructura necesaria para que puedan desarrollar sus actividades, pero debido a que no ha existido una adecuada organización en el Centro de computo no existe un control sobre los equipos, así como tampoco un registro de las personas autorizadas a utilizar los mismos, lo que ocasionaba que cualquier persona ingrese a las instalaciones, motivos que han sido causa para que existan daños en los equipos, y hasta pérdidas; y al no existir control alguno tampoco se ha podido establecer responsables de estas anomalías.

Por lo indicado anteriormente el Laboratorio de Computación de Ingeniería se ve en la necesidad de poseer un Sistema para controlar los horarios de clases de cada una de las Salas del Centro de Cómputo, así como también permitir a profesores o alumnos realizar reservaciones de determinadas Salas ya sea para sustentaciones de trabajos o recibir clases y así tener el control que se necesita.

En el mercado existe gran variedad de software, como por ejemplo aplicaciones para el control de un Cyber Café, los mismos que poseen un software de bloqueo en cada cliente y controlado desde un servidor central (aplicación Cliente – Servidor).



Así mismo existe un software para controlar el acceso de los estudiantes a ciertas salas de cómputo que son dedicadas únicamente para el estudiantado, es decir dichas salas no poseen un horario de clases.

Conocidos los paquetes de software existente en el mercado es claro determinar que no se adaptan completamente a los requerimientos y necesidades del Centro de Cómputo de la Facultad de Ingeniería, sin mencionar el costo de cada uno de estos paquetes de software, y el costo de mantenimiento que implicaría la adquisición de uno de estos.

Con los antecedentes mencionados, la Directora del Centro de Cómputo Ing. María Fernanda Granda planteó la necesidad de elaborar un sistema que cubra dichas necesidades.

1.2 JUSTIFICACIÓN

El Centro de Cómputo de la Facultad de Ingeniería requiere de un mejor control de los estudiantes que ingresan a sus instalaciones a través de:

- Identificación de los alumnos que van a utilizar un computador. (solamente estudiantes de la Facultad de Ingeniería)
- Un sistema que bloquee los computadores permitiendo su uso únicamente a personal autorizado.

Para lograr lo anterior y también para optimizar la administración de este local es necesario un sistema de control de máquinas, horarios y reservaciones de las salas del Laboratorio de Cómputo.



1.3 OBJETIVOS

1.3.1 Objetivo General

Desarrollar un nuevo Sistema Informático para el Control de Acceso y Seguridad para el Centro de Cómputo de la Facultad de Ingeniería de la Universidad de Cuenca.

1.3.2 Objetivos Específicos

Los Objetivos específicos que se persiguen son los siguientes:

- Estudiar las tecnologías para la identificación y control de acceso para seleccionar la más conveniente.
- Incorporar un sistema de bloqueo en los equipos de computación.
- Implementar un sistema para el control de horarios de las Salas del Centro de Cómputo.

2. MARCO TEORICO



2 MARCO TEÓRICO

2.1 SITUACION ACTUAL DEL CENTRO DE COMPUTO

Actualmente el Centro de Cómputo no dispone de ningún sistema de control de acceso al Centro de Cómputo, pudiendo cualquier persona, sea estudiante o no, ingresar y ocupar cualquiera de los equipos de computación en cualquier sala.

2.1.1 Estructura de Salas y Equipos de Computacion

El Laboratorio de Cómputo dispone de las siguientes instalaciones:

- Cuatro salas de máquinas o computadores.
- Una oficina para el Operador o Utilitario.
- Una oficina para el Director o Administrador.
- Una sala para los Servidores y armarios.
- Baños.

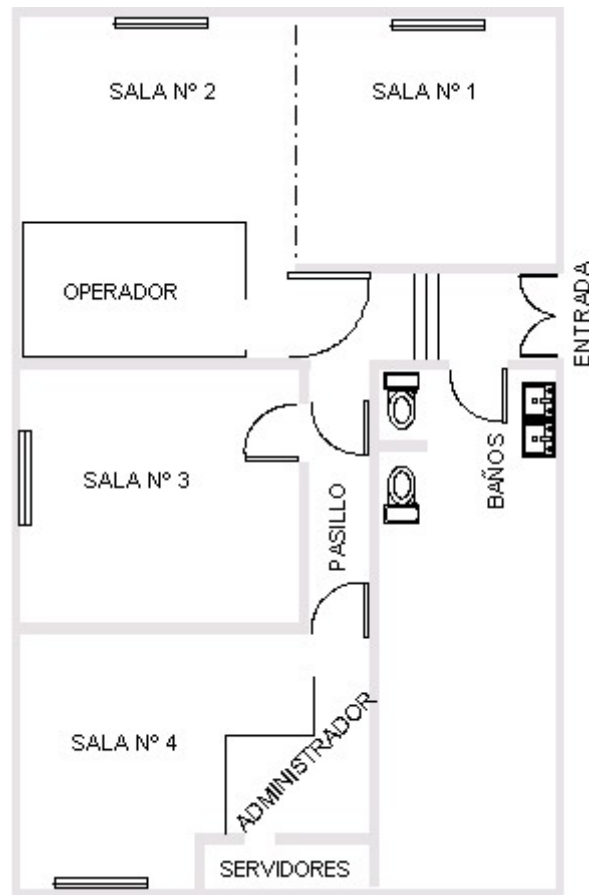


figura 2.1 Estructura del Centro de Cómputo

Características de la Red y Configuración de los Equipos.

La red instalada en el Centro de Cómputo es una red LAN Ethernet de 100 Mbps y con acceso a Internet. Los ordenadores poseen una dirección IP asignada dinámicamente a través del protocolo DHCP. En el entorno de red el nombre asignado a cada máquina es la combinación de la palabra “ccing-” seguido del número de la sala mas un guión “-“ y finaliza con el número de la máquina. Así por ejemplo el nombre del computador #12 de la sala #1 tendrá asignado el nombre “ccing-1-12”.

Ha sido una normativa del Centro de Cómputo tener instalado en todos los computadores un Sistema Operativo basado en la plataforma NT; actualmente todas las máquinas tienen instalado Windows XP. La seguridad más importante ofrecida por este Sistema Operativo es el acceso al mismo mediante la



validación del par login/password, pero para este caso dicha seguridad no es aplicable debido a que se está utilizando un esquema basado en un login genérico (estudiante con privilegios limitados) por puesto de trabajo debido a la dificultad de poseer una cuenta por usuario en cada computador.

Cada una de las salas estan equipadas como se muestra en la tabla 1 y su disposición se presentan en las figuras 2 y 3:

SALA / EQUIPO	SALA N° 1	SALA N° 2	SALA N° 3	SALA N° 4
COMPUTADOR	16	12	15	12
IMPRESORA	1	1	1	1

Tabla 2.1 Número de Equipos por Salas

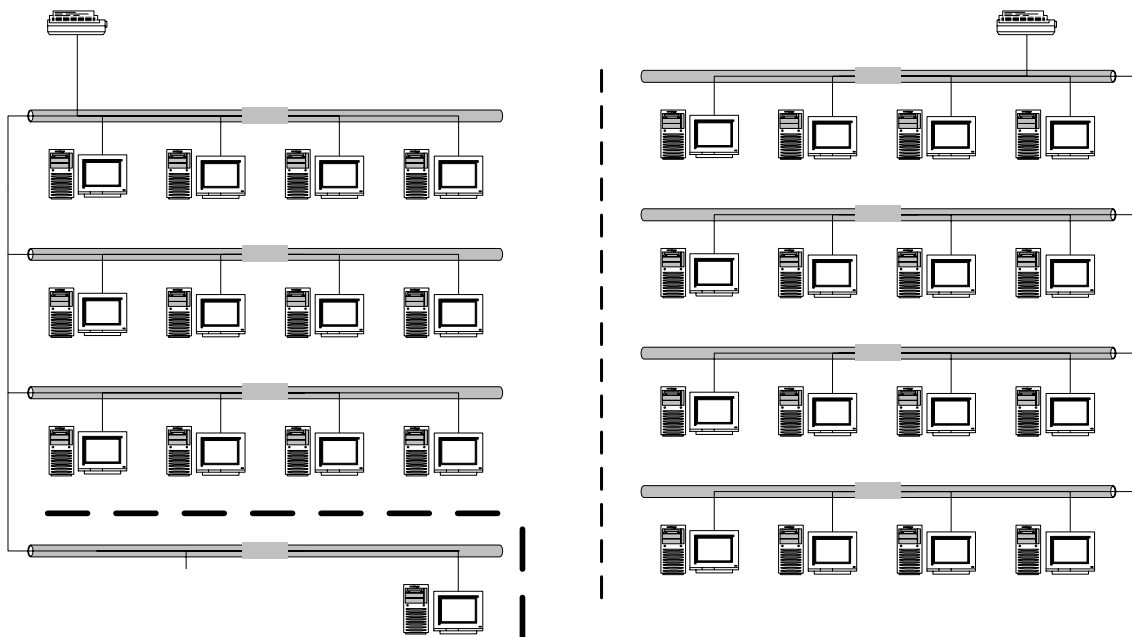


figura 2.2 Disposición de los Equipos en las Salas 1 y 2

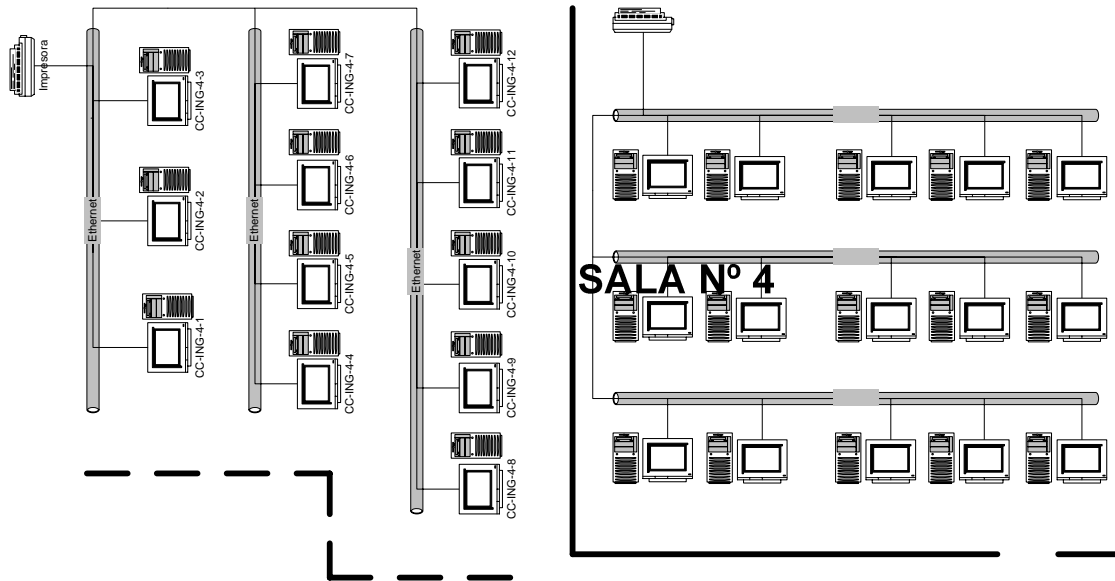


figura 2.3 Disposición de los Equipos en las Salas 3 y 4

ADMINISTRADOR



2.2 SISTEMAS DE IDENTIFICACION

A través de la historia el ser humano y las sociedades han tenido la necesidad de controlar el acceso a ciertas áreas y lugares. Esta necesidad es motivada inicialmente por temor a que personas inescrupulosas o delincuentes puedan robar, extraer o hacer mal uso de quipos sin autorización.

2.2.1 Métodos de Identificación

Una parte importante de la seguridad hace referencia al asunto de la identificación personal para controlar el acceso a determinados lugares. La informática aborda este tema atendiendo a tres modelos de identificación:

- *Primer Nivel:* Identificación en base a algo que se tiene, por ejemplo una credencial con una fotografía, con una banda magnética, una tarjeta inteligente, una llave, etc.
- *Segundo Nivel:* Identificación en base a algo que se sabe; por ejemplo una clave de acceso o palabra de paso (password), un número de acceso PIN (Personal Identificación Number), etc. Este ha sido el método tradicional más comúnmente empleado.
- *Tercer Nivel:* Identificación en base a algo que se es o algo que se hace. Es la denominada tecnología biométrica.

2.2.2 Tecnología de Tarjetas

A continuación se listarán las principales tarjetas existentes en el mercado conjuntamente con la tecnología utilizada y sus principales usos:

Tarjetas de Código de Barras:



Es una tarjeta de apariencia similar a la magnética, pero en lugar de banda lleva impreso un código de barras que no es más que un conjunto de barras y espacios que contienen información codificada (figura 1.4).



figura 2.4 Código de barras

Algunas de las ventajas de esta tarjeta son:

- Al pasarla por el lector, no existe rozamiento, sólo hay un haz de luz que lee el código en cuestión, con lo cual su vida útil es levemente mayor.
- El costo de las tarjetas es similar a las magnéticas.
- Se imprimen a bajo costo
- Los equipos de lectura de código de barras son fáciles de usar.

La función de los lectores de código de barras es leer la información codificada en las barras y espacios del símbolo, y enviarla a un decodificador que finalmente la envía a la computadora o terminal. Los lectores producen una señal digital pura de las barras y espacios.

Tarjetas de Banda Magnética:

La tecnología de banda magnética utiliza cargas electromagnéticas para codificar o grabar información en una cinta pegada a una tarjeta en su parte posterior. Las tarjetas de identificación están hechas de un material de poliéster con una banda magnética de alta coercitividad.

El tamaño o las dimensiones de la tarjeta han sido estandarizadas por el ANSI (American National Standard Institute) y por el ISO (International Standards Organization) para facilitar la manipulación y almacenamiento de las mismas. La franja magnética existente en la tarjeta posee tres pistas (cada una de unos 2.5 mm de anchura) con diferente densidad de información. Esta se mide en



bits por pulgada. Cada una de estas pistas tiene usos y formatos independientes entre si (figura 1.5).

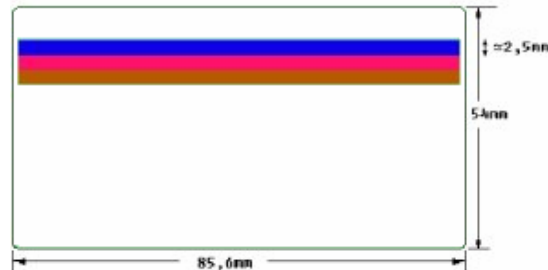


figura 2.5 Banda Magnética

Ventajas:

- Esta tecnología está ampliamente desarrollada y difundida.
- El costo del equipamiento necesario es relativamente bajo, sobre todo teniendo en cuenta su alta durabilidad.
- El tipo de material de la banda magnética previene alteraciones maliciosas o accidentales en la información siendo también inmune a la posibilidad de ser borrada en forma accidental. Los imanes no pueden dañar la información inclusive con contacto directo.

El lector de banda magnética consiste de una cabeza de grabación magnética, la cual puede leer y grabar información en la cinta. Los datos grabados en la cinta están en código binario que posteriormente son convertidos a caracteres alfanuméricos utilizando un formato de datos de alto nivel como son el ISO BCD o el ALPHA.

Tarjetas de Chip o Tarjetas Inteligentes:

Actualmente este tipo de tarjetas se encuentra muy difundido y son tarjetas plásticas del tamaño de una tarjeta de crédito que contiene relativamente grandes cantidades de información dentro de un micro-chip.



Hay dos tipos de tarjetas inteligentes. El primer tipo de tarjeta únicamente contiene memoria. Estas tarjetas son usadas para almacenar información tal como valores de dinero que el usuario puede gastar en una variedad de transacciones. El segundo tipo de tarjetas son verdaderas tarjetas inteligentes donde un microprocesador está impreso en la tarjeta; por lo que esta tarjeta tiene la capacidad de tomar decisiones sobre los datos almacenados en la tarjeta.

El tipo de microprocesador se presenta en dos versiones: La versión de contacto y la versión sin contacto. Ambos tipos tienen el microprocesador acoplado en la tarjeta, sin embargo la versión sin contacto no tiene los contactos dorados visibles en la tarjeta. La tarjeta sin contacto utiliza una tecnología para pasar datos entre la tarjeta y el lector, sin que ningún contacto físico sea realizado.

La desventaja más grande hoy en día de las tarjetas inteligentes es crear un sistema con tarjetas inteligentes. El costo por tarjeta individual sigue siendo muy superior comparado con el costo de una tarjeta de banda magnética.

Tarjetas Ópticas:

Las tarjetas de memoria óptica usan una tecnología similar a la usada por CDs. Un panel de un material sensible al láser está laminado en la tarjeta y es usado para almacenar la información.

Una de sus principales características es su capacidad de almacenamiento de información entre 4 y 6.6 MB.

2.3 TECNICAS DE BLOQUEO DEL COMPUTADOR

Actualmente locales comerciales como Cyber Cafes o cualquier otra entidad que ponga a disposición salas con equipos de computación para su uso público



requieren de un sistema para bloquear el acceso al computador como un medio de seguridad, para de esta forma evitar que cualquier persona pueda utilizar una determinada computadora sin autorización (arbitrariamente).

Hoy por hoy no existen técnicas especializadas y/o definidas para bloquear un ordenador que sea de uso público.

En el transcurso de la investigación y del análisis para el desarrollo de un sistema de bloqueo de un computador de uso público se encontraron básicamente dos formas de bloqueo, en lo que al Sistema Operativo Microsoft Windows con tecnología NT se refiere. Estas formas de bloquear una PC se definen a continuación:

2.3.1 Autenticación de Windows

El acceso a los recursos de una máquina de Windows NT es controlado por un ID de seguridad (SID). Todos los recursos tienen una máscara de acceso que es aplicado sobre el SID para determinar si el SID tiene permitido acceder al recurso en cuestión. Cada usuario en un computador con Windows NT tiene asignado un access token (ficha de acceso) conteniendo su SID. El access token del usuario es comparado con la máscara de el recurso. El usuario únicamente tiene acceso si el SID tiene permisos para acceder al recurso.

WinLogon

El Logon es ejecutado por medio del proceso de login o autenticación interactivo (Winlogon). Winlogon es un proceso confiable para manejar la interacción con el usuario respecto a la seguridad. Este proceso es el primero en ejecutarse en el sistema y realiza las siguientes tareas:

- Crea una estación de windows para representar el teclado, el ratón y el monitor.



- Crea tres escritorios (desktops), un escritorio de aplicación (usado por el usuario), un escritorio de Winlogon (usado por WinLogon para mostrar la pantalla de Login o autenticación) y un escritorio de protector de pantalla (para ejecutar el protector de pantalla). Únicamente el proceso Winlogon tiene acceso al escritorio WinLogon. Esto quiere decir que siempre que el escritorio WinLogon esté activo, ningún otro proceso tiene acceso a los datos asociados con el escritorio.
- Registra una – secuencia de atención segura – (SAS), como la secuencia de tecla de método abreviado (CTRL+ALT+DEL).
- Luego de que el usuario ingresa la contraseña, el WinLogon envía la información a la Autoridad de Seguridad Local (LSA), el cual verifica la contraseña generando el token (ficha) de el usuario. Este token de acceso entonces es usado por el Winlogon para crear el Shell del usuario.

Los aspectos de identificación y autenticación de el winlogon son implementados como una Librería de enlace dinámico (DLL) reemplazable llamado GINA.

GINA

La palabra GINA es un acrónimo para “Graphical Identification and Authentication”, que es una librería de enlace dinámico (DLL) que forma parte del sistema operativo Windows. El GINA es un componente DLL que implementa las políticas de autenticación de el modelo de acceso interactivo. Este archivo es cargado, durante el proceso de arranque del sistema, por la aplicación Winlogon.exe. Una vez cargado esta librería controla las siguientes funciones^[1]:



- Monitoreo del SAS (Secure Attention Sequence): El GINA es responsable del reconocimiento de una – secuencia de atención segura – (SAS), monitoreando eventos SAS y notificando a Winlogon cuando un SAS ha ocurrido. Puede haber más que un SAS definido y el conjunto de SAS puede cambiar con el tiempo. Por ejemplo puede estar presente un conjunto de SAS cuando Winlogon se encuentra en estado logged-off y otro conjunto cuando su estado es logged-on. Winlogon provee servicios para asistir al GINA en el uso del SAS estándar CTRL+ALT+DEL. [1]

[1] <http://msdn.microsoft.com>, MsGina

- Procesando el SAS: Una de las razones para modularizar el acceso a windows (Winlogon) y hacer del GINA un componente reemplazable es proveer mecanismos de identificación y autenticación alternativas. Para hacer esto el GINA debe presentar todas las interfaces de usuario que son el resultado del reconocimiento de un evento SAS. Cuando ningún usuario está logeado (logged on), el GINA es responsable de presentar las opciones de identificación y autenticación así como también cualquier otra opción permitida. Cuando un usuario está logeado (logged on), el GINA es responsable de presentar las opciones necesarias al usuario así como también la de tomar cualquier acción que se crea conveniente. Por ejemplo, en un sistema que incluya una Tarjeta Inteligente, puede ser apropiado automáticamente bloquear la estación de trabajo si el usuario remueve la Tarjeta Inteligente [1].
- Activación del Shell: Cuando un usuario realiza una identificación satisfactoria, el GINA trabaja conjuntamente con el Winlogon.exe para crear los procesos iniciales para ese usuario y para asignar los procesos que el símbolo del acceso de usuario (user's access token) obtuvo del WinLogon.exe. Este proceso debe iniciar el shell predeterminado para el usuario. Normalmente, userinit.exe es iniciado como el proceso inicial. Este programa se ejecuta en el escritorio dentro del contexto del nuevo usuario logeado. Userinit.exe prepara o instala el ambiente del usuario,



esto incluye: restaurar las conexiones de red, cargar la configuraciones del perfil del usuario (color, fuente, protectores de pantalla, etc.). El shell estándar para Windows NT es Explorer.exe. Este programa administra el escritorio, barra de tareas, etc. Una vez que el Shell es creado, el resto de los procesos creados por el usuario heredan automáticamente el ambiente que userinit.exe configuró [1].

El Winlogon y el GINA tienen un protocolo para su interacción. El Winlogon llama a la librería GINA siempre que necesite ejecutar una operación; GINA puede ejecutar estas actividades y enviar una respuesta de retorno al proceso Winlogon.

[1] <http://msdn.microsoft.com>, MsGina

Arquitectura de la Autenticación de Windows:

Durante el encendido del computador, el proceso Winlogon.exe es iniciado. Este proceso se mantiene ejecutándose en background durante todo el tiempo que el Sistema Operativo está cargado.

Cuando un usuario presiona CTRL+ALT+DEL (emite un evento SAS), el proceso Winlogon llama al GINA para controlar el proceso de identificación y autenticación del usuario. GINA presenta en pantalla un cuadro de control de acceso (Logon) para que el usuario lo llene. Usando este cuadro de diálogo GINA adquiere la información que necesita para validar al usuario.

GINA entonces contacta tanto al Directorio Activo o al Controlador de Dominio. Luego de que GINA ha validado el username y contraseña, éste retorna un token y el control a el proceso WinLogon que a su vez inicia un shell de nivel de usuario usando los permisos de éste, y entonces crea el ambiente del usuario según las configuraciones del usuario autenticado.



Una vez que el Shell del usuario y el ambiente es establecido, WinLogon pasa el control del Shell del usuario. Todos estos pasos de la autenticación de Windows se representan en la figura 2.6:

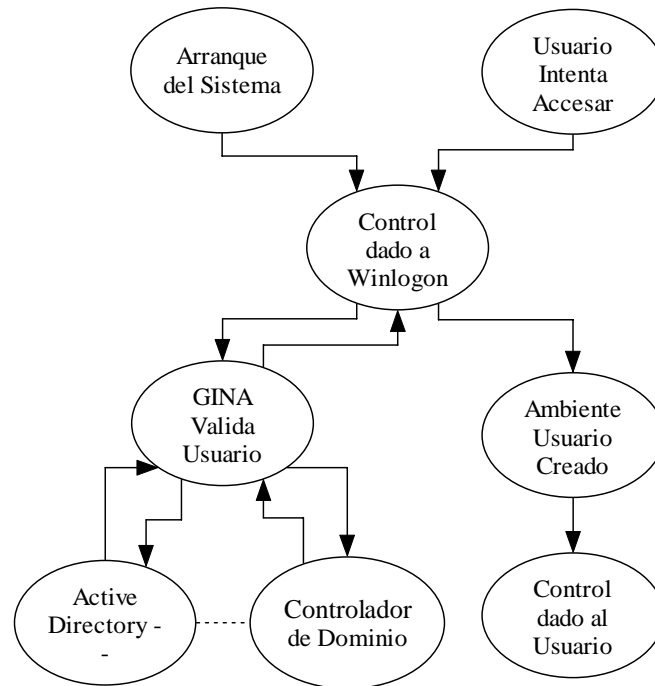


figura 2.6 Arquitectura de la Autenticación de Windows

Esta forma de bloquear el computador por medio de la Autenticación de Windows es muy conveniente y segura. Para personalizar este tipo de Autenticación la única forma como se mencionó anteriormente es desarrollando una nueva librería de enlace dinámico que reemplace al archivo MSGina.dll de Windows.



2.3.2 Bloquear el Escritorio de Windows

Esta forma de bloquear un computador es la más difundida e implementada en los software para Cyber Cafes y este bloqueo es menos complejo que el anterior, sin embargo posee su grado de dificultad, debido a que se requiere la utilización de las librerías API y/o los Hooks (Ganchos) de Windows.

API de Windows

Las API de Windows son bibliotecas de vínculos dinámicos (DLL) que forman parte del sistema operativo Windows. Se utilizan para realizar tareas cuando resulta difícil escribir procedimientos equivalentes.

La ventaja de utilizar las API de Windows en el código es que pueden ahorrar tiempo porque contienen numerosas funciones útiles ya escritas y listas para utilizar. La desventaja es que puede resultar difícil trabajar con las API de Windows y pueden ser implacables cuando las cosas van mal.

Las bibliotecas de vínculos dinámicos (DLL) de Windows representan una categoría especial de interoperabilidad. Las API de Windows no utilizan código administrado, no tiene bibliotecas de tipos incorporadas y utilizan tipos de datos que son diferentes a los utilizados con Visual Studio .NET y su lenguaje C#. Debido a estas diferencias y a que las API de Windows no son objetos COM, la interoperabilidad con las API de Windows y la plataforma .NET se lleva a cabo utilizando invocación de la plataforma, o PInvoke. Invocación de la plataforma es un servicio que permite al código administrado llamar a funciones no administradas implementadas en archivos DLL.

Las llamadas de API de Windows eran una parte importante de la programación con Visual Basic en el pasado, pero rara vez son necesarias con C# .NET. Es decir el .Net Framework implementa gran parte de los API de



Windows internamente para ser utilizados a través del código administrado. Es por esto que el .Net Framework es relativamente grande, pero no tan grande como para abarcar todos los API de Windows. Por ejemplo una de las áreas especiales de los API de Windows que el .NET Framework no cubre es la del acceso al Hardware.

Ganchos de Windows (Hooks)

Los ganchos de Windows pueden considerarse uno de los rasgos más poderosos de Windows. Con ellos podemos monitorear el tráfico de mensajes en el sistema y procesar ciertos tipos de mensajes antes de estos lleguen al proceso designado. Para esto se crea una función filtro llamada también procedimiento del gancho, que será llamado cada vez que ocurra un evento que sea de nuestro interés. Hay dos tipos de ellos: ganchos locales y ganchos remotos.

- ***Ganchos Locales*** atraparán eventos que ocurrirán en el propio proceso.
- ***Ganchos Remotos*** atraparán eventos que ocurrirán en otro(s) proceso(s). Hay dos tipos de ganchos remotos
 - ***específicos-a-un-hilo*** [*thread-specific*] atrapan eventos que ocurren en un hilo específico de otro proceso. En pocas palabras, quieres observar un hilo específico en un proceso específico.
 - ***ancho-de-sistema*** [*system-wide*] atrapa todos los eventos de todos los hilos en todos los procesos en el sistema.

Cuando se establece un gancho, Windows crea una estructura de datos en la memoria, que contiene información sobre el gancho, y lo agrega a una lista enlazada de ganchos existentes. El gancho nuevo es agregado en frente de los ganchos antiguos. Cuando un evento ocurre, si se instala un gancho local, la



función filtro, declarada en nuestro programa, es llamada de una manera directa. Pero si es un gancho remoto, el sistema debe inyectar el código para el procedimiento del gancho dentro del espacio de direcciones del(os) otro(s) proceso(s). Y el sistema puede hacer eso sólo si la función reside en una DLL. De esta manera, si se quiere usar un gancho remoto, el procedimiento del gancho debe residir en una DLL.

Hay dos excepciones a esta regla: ganchos de grabación diaria [journal record] y ganchos de ejecución diaria [journal playback]. El procedimiento de gancho para estas dos clases debe residir en el hilo que instala los ganchos. La razón por la que debe ser así, es porque ambos ganchos tienen que ver con la intercepción de bajo-nivel de los eventos de entrada del hardware. Los eventos de entrada deben ser grabados/ejecutados [recorded/playbacked] en el orden que aparecen. Si el código de estos dos ganchos está en un DLL, los eventos de entrada pueden dispersarse entre varios hilos y es imposible saber su orden. Así que la solución es: el procedimiento de gancho de esos dos ganchos deben estar solamente en un hilo, en el hilo que instala los ganchos.

A continuación se listarán los tipos de ganchos existentes [2]:

- WH_CALLWNDPROC nos permite monitorear mensajes enviados a los procedimientos de windows. El sistema llama al gancho WH_CALLWNDPROC antes de pasar el mensaje al procedimiento de la ventana receptora, es decir cuando SendMessage es llamado.
- WH_CALLWNDPROCRET similar al anterior pero es llamado después de que el procedimiento de la ventana a procesado el mensaje o en otras palabras cuando regresa el mensaje SendMessage
- WH_GETMESSAGE utilizado para monitorear mensajes ejecutado cuando GetMessage o PeekMessage son llamados
- WH_KEYBOARD llamado cuando GetMessage o PeekMessage regresan WM_KEYUP o WM_KEYDOWN desde la cola de mensajes



- WH_MOUSE llamado cuando GetMessage o PeekMessage regresan un mensaje de ratón desde la cola de mensajes

[2] <http://www.allapi.net>, Windows Hooks



- WH_HARDWARE llamado cuando GetMessage o PeekMessage regresa algún mensaje hardware que no está relacionado con el teclado ni con el ratón.
- WH_MSGFILTER llamado cuando una caja de diálogo, el menú o la barra de pergamino [scrollbar] está apunto de [is about] procesar un mensaje. Este gancho es local. Se usa específicamente para esos objetos que tienen sus propios bucles de mensajes internos.
- WH_SYSMSGFILTER lo mismo que WH_MSGFILTER pero de ancho de sistema [system-wide]
- WH_JOURNALRECORD llamado cuando Windows regresa mensajes desde hardware input queue
- WH_JOURNALPLAYBACK llamado cuando es solicitado un evento del hardware desde la cola de entrada del sistema.
- WH_SHELL llamado cuando algo interesante sobre el shell ocurre, tal como cuando la barra de tareas necesita redibujar su botón.
- WH_CBT usado específicamente para entrenamiento [computer-based training (CBT)].
- WH_FOREGROUNDIDLE usado internamente por Windows. Poco usada para aplicaciones generales
- WH_DEBUG usado para depurar el procedimiento de enganche

Para instalar un gancho, se llama a la función API de Windows SetWindowsHookEx que tiene la siguiente sintaxis:

SetWindowsHookEx(HookType:DWORD,pHookProc:DWORD,hInstance:DWORD, ThreadID:DWORD)

- HookType es uno de los valores en la lista de arriba, e.g., WH_MOUSE, WH_KEYBOARD
- pHookProc es la dirección del procedimiento de gancho que será llamada para procesar los mensajes para el gancho específico. Si el



gancho es remoto, debe residir en una DLL. Si no, debe estar en el proceso.

- `hInstance` es el manejador de instancia de la DLL en la cual reside el procedimiento de gancho. Si el gancho es local, este valor debe ser `NULL`
- `ThreadID` es el ID del hilo para el cual se quiere instalar el gancho que lo espíe. Este parámetro es el que determina si el gancho es local o remoto. Si este parámetro es `NULL`, Windows interpretará el gancho como un gancho remoto de ancho de sistema [system-wide] que afecta todos los hilos del sistema. Si se quiere especificar el ID de un hilo en tu propio proceso, este hilo es local. Si se especifica el ID del hilo de otro proceso, el gancho es thread-specific remote one. Hay dos excepciones a esta regla: `WH_JOURNALRECORD` y `WH_JOURNALPLAYBACK` siempre son ganchos de ancho de sistema [system-wide] locales que no se requieren que estén en una DLL. Y `WH_SYSMMSGFILTER` siempre es un gancho remoto de ancho de sistema [system-wide]. Realmente es idéntico al gancho `WH_MSGFILTER` con `ThreadID==0`.

Si la llamada tiene éxito, el valor retornado es el manejador del procedimiento del gancho. Si no, regresa `NULL`. Se debe guardar el manejador del gancho para poder desinstalarlo después.

Para desinstalar el gancho se debe llamar a la función `API UnhookWindowsHookEx` que sólo acepta un parámetro, el manejador del gancho que se desea desinstalar. Si la llamada tiene éxito, regresa un valor diferente a cero. En caso contrario, regresa un valor igual a cero.

Pasos para el bloque del escritorio

El bloqueo de escritorio consiste en impedir el acceso al escritorio de windows e impedir la ejecución de aplicaciones. Durante la investigación para lograr este objetivo se encontraron los siguientes pasos a ejecutar:



- Ocultar el Escritorio de Windows: Este proceso consiste en ocultar el escritorio de windows, ocultar la barra de tareas y el botón de Inicio. En este proceso se utilizan las API de Windows.
Pero solo realizando esta operación no es suficiente debido a que el botón de inicio todavía está activo al presionar la tecla de windows o al hacer doble click en el área del escritorio. Para evitar esto se realiza el siguiente paso.
- Deshabilitar las teclas del sistema: Las teclas del sistema son todas las combinaciones de teclas especiales que el sistema operativo usa para intercambiar entre aplicaciones o ejecutar el Administrador de Tareas. Para lograr este proceso se utilizan los Hooks de Windows.
- Finalmente colocar un formulario que abarque todo el escritorio de Windows en forma de papel tapiz.

3. ANALISIS DEL SISTEMA



3 ANALISIS DEL SISTEMA

3.1 INTRODUCCIÓN

En este capítulo abordaremos el Análisis del sistema aplicando la Metodología RUP (Proceso Unificado de Desarrollo), basada en la notación UML; para lo cual se seguirán los siguientes pasos:

- La realización de un diccionario de datos exponiendo aquellas expresiones que resulten importantes para este proyecto.
- La subdivisión del Sistema en subsistemas mediante el diagrama de paquetes.
- El análisis de cada subsistema mediante la técnica de casos de uso.
- En base a los casos de uso, se obtiene un diseño inicial del diagrama de clases para cada subsistema en el que sea necesario.
- La realización de un análisis dinámico del sistema, presentando los diagramas de secuencia y de colaboración de cada caso de uso.

De esta manera se habrá analizado el sistema completamente y se puede pasar a la fase de diseño del sistema.



3.2 DICCIONARIO DE DATOS

Nombre	API
Descripción:	(Application Program Interface). Conjunto de convenciones internacionales que definen cómo debe invocarse una determinada función de un programa desde una aplicación. Cuando se intenta estandarizar una plataforma, se estipulan unos APIs comunes a los que deben ajustarse todos los desarrolladores de aplicaciones. Herramientas de programación para rutinas, protocolos y software.

Nombre	Socket
Descripción:	Número de identificación compuesto por dos números: La dirección IP y el número de puerto TCP. En la misma red, el número IP es el mismo, mientras el número de puerto es el que varía.

Nombre	Interfaz
Descripción:	Todo el conjunto de botones, etiquetas, imágenes, etc. que forman la interfaz con la que el usuario interactúa con el sistema.

Nombre	Clave
Descripción:	La clave personal de cada alumno que le permitirá acceder al sistema.



3.3 DIAGRAMA DE PAQUETES

El sistema consta de dos partes bien definidas, cada una de las cuales será considerada como un subsistema aparte.

En primer subsistema es el encargado de la Administración del Centro de Cómputo. Por otro lado el subsistema encargado de establecer la comunicación entre cliente y servidor, para el intercambio de comandos o instrucciones, lo denominaremos Comunicación. Por otra parte el subsistema Cliente, como su nombre lo indica se encuentra en el computador Cliente y es el encargado de bloquear/desbloquear el computador. El subsistema de Acceso al Sistema es el encargado de realizar la identificación del usuario y actividades como la selección de un computador. Finalmente se tiene el subsistema de Migración de datos que es el encargado de migrar datos desde un archivo de Excel a la Base de Datos del Sistema. (*figura 3.1*).

Los subsistemas de Administración del Centro de Cómputo, Acceso al Sistema, Comunicación y Migración de Datos se encuentran ubicados en el Servidor, mientras que los subsistemas Cliente y Comunicación se encuentran en el Cliente, teniendo así una arquitectura Cliente – Servidor.

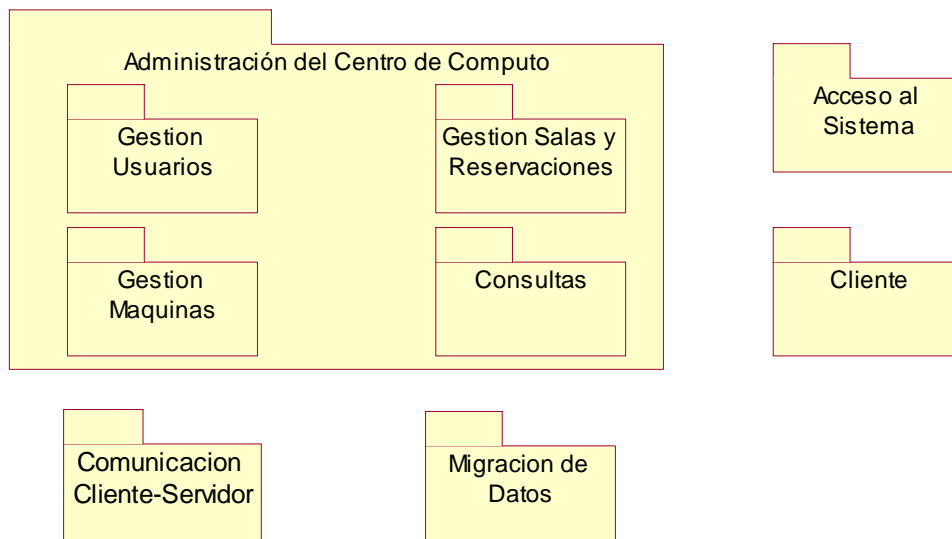


figura 3.1 Diagrama de Paquetes del Sistema

Dentro de Administración del Centro de Cómputo se han considerado varios subsistemas, los mismos que describimos a continuación:

- **Gestión de Usuarios:** Los usuarios se identificarán en el sistema mediante tarjetas magnéticas. El Sistema debe proporcionarle al administrador mecanismos para crear un nuevo usuario, así como también mecanismos para modificar y eliminar a un usuario.
- **Gestión de Máquinas o Computadores:** El Sistema debe suministrar mecanismos para crear una nueva máquina, así como también mecanismos para modificar y eliminar una máquina o computador.
- **Gestión de Salas y Reservas:** Este subsistema facilita mecanismos para el mantenimiento de salas, el mantenimiento de horarios y el también procesos para el mantenimiento de reservas.
- **Consultas:** permite al usuario realizar consultas sobre consumos.



A continuación estudiaremos cada subsistema por separado y comenzaremos por los casos de uso.



3.4 DIAGRAMAS DE CASOS DE USO

Los diagramas de casos de uso reflejan de forma simple los requerimientos de un sistema. Su utilidad es descomponer las necesidades del sistema que se va a analizar en acciones sencillas que son realizadas por una serie de actores que representan a los usuarios del sistema.

3.4.1 Subsistema Administración del Centro de Cómputo.

3.4.1.1 Gestión de Usuarios

En el diagrama de casos de uso se tiene un único actor participante que es el administrador.

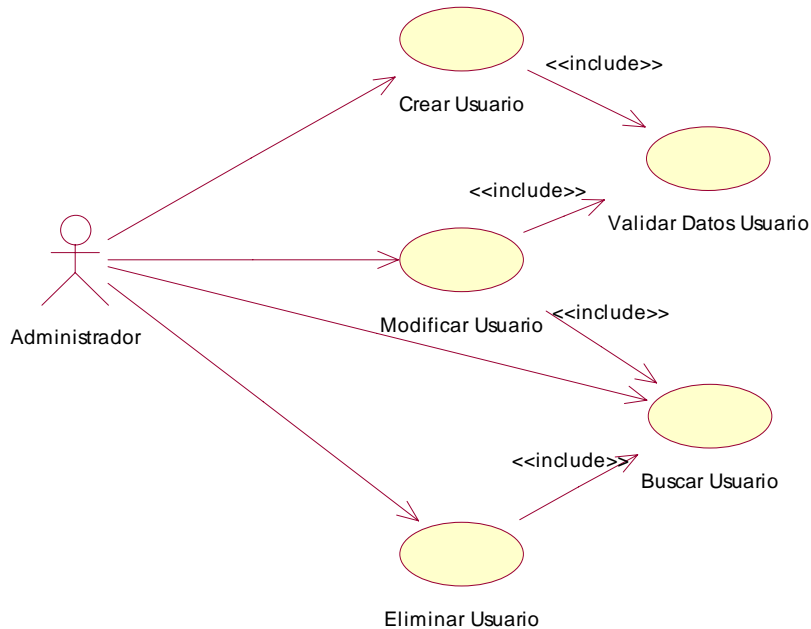


figura 3.2 Diagrama de Casos de Uso Gestión de Usuarios

A continuación se pasa a describir cada uno de los casos de uso anteriores mediante el empleo de las plantillas.



Caso de Uso Crear Usuario

Nº 1	Crear Usuario
Objetivo	Crear un nuevo Usuario (Alumno, Profesor ó Administrador) para el Centro de Cómputo
Actores	Administrador
Precondiciones	El usuario está identificado en el Sistema como Administrador.
Pasos	1. A: Introducir los datos (ID, Cédula, Apellidos, Nombres, Teléfono, Dirección). Como identificador del usuario para su acceso al sistema se tomará la cédula. Seleccionar el Tipo de Usuario: si es Alumno ingresar los datos de su Escuela y Año que cursa. Finalmente se debe asignar un número de horas de consumo. 2. S: Caso de Uso <i>Validar Datos Usuario</i>
Postcondiciones	El usuario queda registrado en el sistema
Variaciones	2.a. Los datos del usuario son incorrectos: 2.a.1. Indicar error. 2.a.2. Finalizar caso de uso.
Extensiones	
Cuestiones	

Caso de Uso Validar Datos Usuario

Nº 2	Validar Datos Usuario
Objetivo	Validar la información del usuario para efectos de su registro en el sistema y verificar que el número de cédula del usuario sea único
Actores	Interfaz Mantenimiento Usuarios (IMU), Servicio Validación Datos (SVD)
Precondiciones	
Pasos	1. IMU: Enviar petición de validación de los datos. 2. SVD: Valida que los campos ID, Cédula, Nombres, Apellidos, Tipo de Usuario y Horas Asignadas sean válidos y no nulos. 3. SVD: Verifica que el número de cédula sea único. 4. IMU: Recibe respuesta de datos correctos
Postcondiciones	Los datos del usuario son válidos
Variaciones	2.a. Los datos del usuario son incorrectos: 2.a.1. Indicar error.



	<p>2.a.2. Finalizar caso de uso.</p> <p>3.a. La cédula del usuario no es única:</p> <p>3.a.1. Indica error.</p> <p>3.a.2. Finalizar caso de uso.</p>
Extensiones	
Cuestiones	

Caso de Uso Eliminar Usuario

Nº 3	Eliminar Usuario
Objetivo	Eliminar a un usuario del Sistema
Actores	Administrador
Precondiciones	Debe haberse dado el c.d.u. <i>Crear Usuario</i>
Pasos	<p>1. S: Caso de Uso <i>Buscar Usuario</i>.</p> <p>2. S: Efectuar la eliminación del usuario en el sistema.</p>
Postcondiciones	El usuario queda eliminado de sistema
Variaciones	<p>1.a. El usuario no es encontrado</p> <p>1.a.1. Indicar usuario no encontrado</p> <p>1.a.2. Finalizar caso de uso</p>
Extensiones	
Cuestiones	

Caso de Uso Buscar Usuario

Nº 4	Buscar Usuario
Objetivo	Dado un criterio de búsqueda, Cédula, apellidos y/o nombres el sistema busca y carga los datos del usuario
Actores	Administrador
Precondiciones	
Pasos	<p>1. A: Seleccionar un criterio de búsqueda e Ingresar el campo pedido (Cédula, apellidos y/o nombres)</p> <p>2. S: Buscar y cargar los datos del usuario</p>
Postcondiciones	
Variaciones	<p>2.a. El usuario no encontrado</p> <p>2.a.1. Indicar usuario no encontrado</p>



	2.a.2. Finalizar caso de uso
Extensiones	
Cuestiones	

Caso de Uso Modificar Usuario

Nº 5	Modificar Usuario
Objetivo	Modificar los datos de un usuario del Centro de Cómputo
Actores	Administrador
Precondiciones	Debe haberse dado el c.d.u. <i>Crear Usuario</i>
Pasos	<ol style="list-style-type: none">1. S: Caso de Uso <i>Buscar Usuario</i>.2. A: Introducir los datos a modificar3. S: Caso de Uso <i>Validar Datos Usuario</i>.4. S: Actualizar datos del usuario.
Postcondiciones	Los datos del usuario quedan modificado en el sistema
Variaciones	
Extensiones	
Cuestiones	



3.4.1.2 Gestión de Máquinas o Computadores

En el diagrama de casos de uso se tiene un único actor participante que es el administrador.

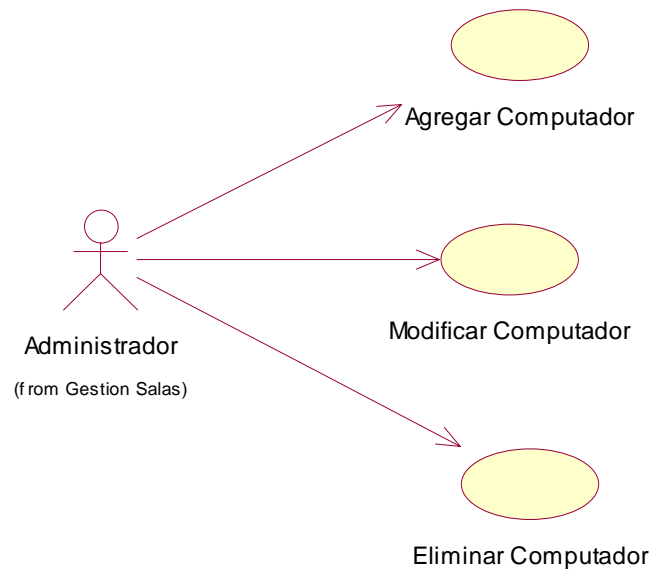


figura 3.3 Diagrama de Casos de Uso Gestión de Computadores.

A continuación se pasa a describir cada uno de los casos de uso anteriores mediante el empleo de las plantillas.

Caso de Uso Agregar Computador

1	Agregar Computador
Objetivo	Agregar un nuevo computador a una determinada Sala del Centro de Cómputo.
Actores	Administrador
Precondiciones	
Pasos	1. A: Introducir información del Nombre Dirección MAC, Dirección IP, Procesador, Memoria, Disco Duro y asignar o seleccionar la Sala a la que pertenece. 2. S: El sistema registra la información
Postcondiciones	La nuevo computador queda registrado en el sistema



Variaciones	
Extensiones	
Cuestiones	

Caso de Uso Modificar Computador

2	Modificar Computador
Objetivo	Modificar la información de un computador
Actores	Administrador
Precondiciones	Debe haberse dado el caso de uso <i>Agregar Computador</i>
Pasos	1 A: Seleccionar Sala 2 S: Listar Computadores de la sala 3 A: Seleccionar computador a modificar 4 S: Presenta la información del computador 5 A: Introducir los datos a modificar 6 S: Actualizar los datos del computador
Postcondiciones	Registro del computador queda actualizado en el sistema
Variaciones	
Extensiones	
Cuestiones	

Caso de Uso Eliminar Computador

3	Eliminar Computador
Objetivo	Eliminar un computador de una Sala del Centro de Cómputo
Actores	Administrador
Precondiciones	Debe haberse dado el caso de uso <i>Agregar Computador</i>
Pasos	1 A: Seleccionar Sala 2 S: Listar Computadores de la Sala 3 A: Seleccionar computador a modificar 4 S: Presentar información del computador en pantalla 5 A: Confirmar Eliminación 6 S: Realizar la eliminación del computador
Postcondiciones	Registro del computador queda eliminado del sistema



Variaciones	
Extensiones	
Cuestiones	



3.4.1.3 Gestión de Salas y Reservaciones

En el diagrama de casos de uso el principal actor es el administrador que tiene la responsabilidad de crear, modificar o eliminar salas del centro de cómputo, así como también la de crear, modificar o eliminar horarios. Para el caso de uso de reservaciones participan el administrador y el usuario (estudiante o profesor).

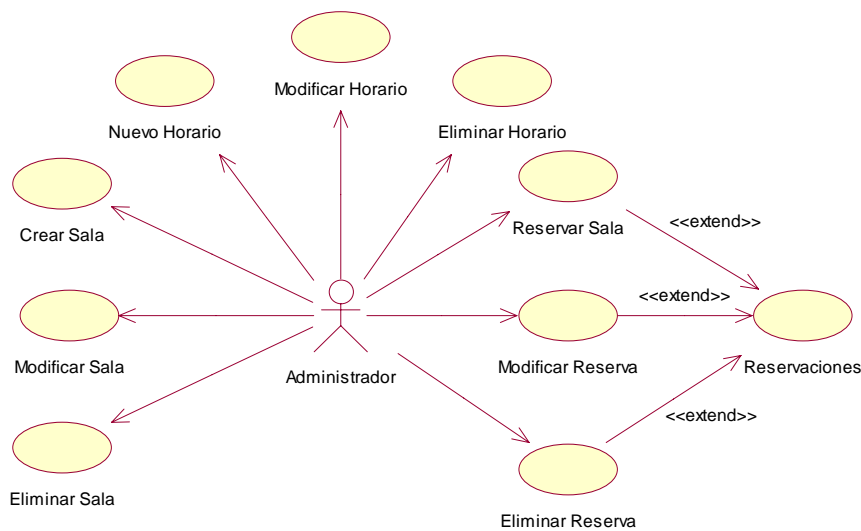


figura 3.4 Diagrama de Casos de Uso Gestión de Salas y Reservaciones

A continuación se describen cada uno de los casos de uso:

Caso de Uso Crear Sala

1	Crear Sala
Objetivo	Agregar una nueva sala al Centro de Cómputo
Actores	Administrador
Precondiciones	
Pasos	1 A: Introducir el nombre y las aplicaciones instaladas. 2 S: El sistema registra la información
Postcondiciones	La nueva sala queda registrada en el sistema
Variaciones	
Extensiones	



Cuestiones	
-------------------	--

Caso de Uso Modificar Sala

2	Modificar Sala
Objetivo	Modificar la información de una sala
Actores	Administrador
Precondiciones	Debe haberse dado el caso de uso <i>Crear Sala</i>
Pasos	1 A: Seleccionar Sala 2 S: Presenta la información de la Sala 3 A: Introducir los datos a modificar 4 S: Actualizar los datos de la Sala
Postcondiciones	Registro de la Sala queda actualizado en el sistema
Variaciones	
Extensiones	
Cuestiones	

Caso de Uso Eliminar Sala

3	Eliminar Sala
Objetivo	Eliminar una Sala del Centro de Cómputo
Actores	Administrador
Precondiciones	Debe haberse dado el caso de uso <i>Crear Sala</i>
Pasos	1 A: Seleccionar Sala 2 S: Presentar información de la sala en pantalla 3 A: Confirmar Eliminación 4 S: Realizar la eliminación de la Sala
Postcondiciones	Registro de la sala queda eliminado del sistema
Variaciones	
Extensiones	
Cuestiones	

Caso de Uso Nuevo Horario

4	Nuevo Horario
----------	---------------



Objetivo	Asignar un Nuevo Horario de Clase a una Sala del Centro de Cómputo.
Actores	Administrador
Precondiciones	
Pasos	<ol style="list-style-type: none">1 A: Introducir la Hora Inicio, Hora Final, Día, Asignatura, Profesor y Sala2 S: Verificar que los datos introducidos sean correctos.3 S: Verificar que el horario de clase de la sala esté disponible.4 S: El sistema registra la información
Postcondiciones	La nueva sala queda registrada en el sistema
Variaciones	<ol style="list-style-type: none">2.a. Los datos del Horario de Clase son incorrectos:<ol style="list-style-type: none">2.a.1. Indicar error.2.a.2. Finalizar caso de uso.3.a. El horario de Clase no está disponible o existe cruce de horas:<ol style="list-style-type: none">3.a.1. Indicar error.3.a.2. Finalizar caso de uso.
Extensiones	
Cuestiones	

Caso de Uso Modificar Horario

5	Modificar Horario
Objetivo	Modificar la información de un Horario de Clase
Actores	Administrador
Precondiciones	Debe haberse dado el caso de uso <i>Nuevo Horario</i>
Pasos	<ol style="list-style-type: none">1 A: Seleccionar Sala y Horario de Clase2 S: Presenta la información del Horario de Clase.3 A: Introducir los datos a modificar.4 S: Verificar que los datos introducidos sean correctos.5 S: Verificar que el horario de Clase esté disponible6 S: Actualizar los datos del Horario en el Sistema
Postcondiciones	Horario de Clase queda actualizado en el sistema
Variaciones	<ol style="list-style-type: none">4.a. Los datos del Horario de Clase son incorrectos:<ol style="list-style-type: none">4.a.1. Indicar error.4.a.2. Finalizar caso de uso.



	5.a. El horario del Horario no está disponible o existe cruce de horas: 5.a.1. Indicar error. 5.a.2. Finalizar caso de uso.
Extensiones	
Cuestiones	

Caso de Uso Eliminar Horario

6	Eliminar Reservación
Objetivo	Eliminar una Reservación de una Sala
Actores	Administrador
Precondiciones	Debe haberse dado el caso de uso <i>Nuevo Horario</i>
Pasos	1 A: Seleccionar Sala y Horario. 2 S: Presenta la información del Horario de Clase. 3 A: Confirmar Eliminación 4 S: Eliminar el Horario de Clase
Postcondiciones	Horario de Clase queda eliminado del sistema
Variaciones	
Extensiones	
Cuestiones	

Caso de Uso Reservar Sala

7	Reservar Sala
Objetivo	Realizar una nueva reservación de una sala del Centro de Cómputo
Actores	Administrador
Precondiciones	
Pasos	1. A: Introducir la Fecha Inicial, la Fecha Final, Hora Inicial, Hora Final Descripción, Responsable y la Sala a Reservar 2. S: Verificar que los datos introducidos sean correctos. 3. S: Verificar que el horario de la sala a reservar esté disponible. 4. S: El sistema registra la información



Postcondiciones	La nueva sala queda registrada en el sistema
Variaciones	<p>2.a. Los datos de la reservación son incorrectos:</p> <p style="padding-left: 40px;">2.a.1. Indicar error.</p> <p style="padding-left: 40px;">2.a.2. Finalizar caso de uso.</p> <p>3.a. El horario a reservar no está disponible o existe cruce de horas:</p> <p style="padding-left: 40px;">3.a.1. Indicar error.</p> <p style="padding-left: 40px;">3.a.2. Finalizar caso de uso.</p>
Extensiones	
Cuestiones	

Caso de Uso Modificar Reserva

8	Modificar Reserva
Objetivo	Modificar la información de una reservación
Actores	Administrador
Precondiciones	Debe haberse dado el caso de uso <i>Reservar Sala</i>
Pasos	<ol style="list-style-type: none"> 1. A: Seleccionar Sala y Reservación. 2. S: Presenta la información de la Reservación. 3. A: Introducir los datos a modificar. 4. S: Verificar que los datos introducidos sean correctos. 5. S: Verificar que el horario de la sala a reservar esté disponible 6. S: Actualizar los datos de la Sala
Postcondiciones	Registro de la Reservación queda actualizado en el sistema
Variaciones	<p>4.a. Los datos de la reservación son incorrectos:</p> <p style="padding-left: 40px;">4.a.1. Indicar error.</p> <p style="padding-left: 40px;">4.a.2. Finalizar caso de uso.</p> <p>5.a. El horario a reservar no está disponible o existe cruce de horas:</p> <p style="padding-left: 40px;">5.a.1. Indicar error.</p> <p style="padding-left: 40px;">5.a.2. Finalizar caso de uso.</p>
Extensiones	
Cuestiones	



Caso de Uso Eliminar Reserva

9	Eliminar Reservación
Objetivo	Eliminar una Reservación de una Sala
Actores	Administrador
Precondiciones	Debe haberse dado el caso de uso <i>Reservar Sala</i>
Pasos	<ol style="list-style-type: none">1. A: Seleccionar Sala y Reservación.2. S: Presenta la información de la Reservación.3. A: Confirmar Eliminación4. S: Eliminar la Reservación
Postcondiciones	Registro de la reservación queda eliminado del sistema
Variaciones	
Extensiones	
Cuestiones	



3.4.2 Subsistema Acceso al Sistema.

Los usuarios se identificarán en el sistema mediante tarjetas magnéticas. Una vez identificados, el usuario podrá consultar su consumo, revisar el estado de las máquinas así como también habilitar o deshabilitar un computador.

En el diagrama de casos de uso se tienen tres actores usuario, profesor y alumno. Profesor y Alumno mantienen una relación de generalización con Usuario.

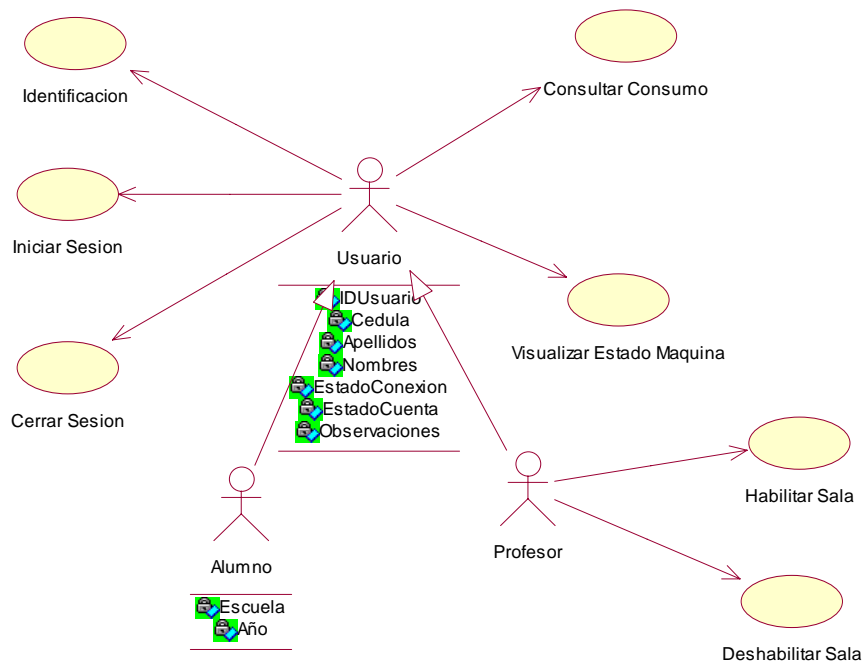


figura 3.5 Diagrama de Casos de Uso Acceso al Sistema

A continuación se describen cada uno de los casos de uso:

Caso de Uso Identificar Usuario

1	Identificar Usuario
Objetivo	Decodificar la información de la tarjeta y verificar la información del usuario
Actores	Usuario



Precondiciones	
Pasos	<ol style="list-style-type: none">1. A: Pasar tarjeta por el lector o introducir su número de cédula y su clave.2. S: Verificar Datos3. S: Informar del resultado
Postcondiciones	El usuario es identificado y permitido su acceso al sistema
Variaciones	<ol style="list-style-type: none">3.a. Usuario Incorrecto o no válido<ol style="list-style-type: none">3.a.1 Indicar Error3.a.2 Finalizar Caso de Uso
Extensiones	
Cuestiones	

Caso de Uso Visualizar Estado Máquina

2	Visualizar Estado Máquina
Objetivo	Visualizar en pantalla el estado de los computadores de una sala determinada
Actores	Usuario
Precondiciones	Debe haberse dado el caso de uso <i>Identificación</i>
Pasos	<ol style="list-style-type: none">1 A: Seleccionar Sala.2 S: Visualizar en pantalla el estado de los computadores
Postcondiciones	
Variaciones	
Extensiones	
Cuestiones	

Caso de Uso Iniciar Sesión

3	Habilitar Máquina
Objetivo	Habilitar o desbloquear un computador para poder utilizarlo
Actores	Usuario
Precondiciones	Debe haberse dado el caso de uso Visualizar Estado Máquina
Pasos	<ol style="list-style-type: none">1 A: Seleccionar Computador (Doble Clic).2 A: Introducir Minutos a ocupar.



	<p>3 S: Modificar el estado del Computador a NO Disponible(Ocupado).</p> <p>4 S: Crear un registro de consumo en la base de datos.</p> <p>5 S: Enviar Instrucción Inicio de Sesión de Consumo al computador seleccionado a través del Subsistema de Comunicación para Desbloquear la Máquina.</p> <p>6 S: Cerrar sesión de Acceso al Sistema.</p>
Postcondiciones	Estado de Computador cambiado e instrucción enviada a subsistema de comunicación.
Variaciones	
Extensiones	
Cuestiones	

Caso de Uso Cerrar Sesión

4	Deshabilitar Máquina
Objetivo	Deshabilitar un Computador para que quede bloqueado.
Actores	Usuario
Precondiciones	Estar Identificado en el Sistema como Administrador.
Pasos	<p>1 A: Seleccionar Computador.</p> <p>2 A: Clic en la opción Cerrar Sesión.</p> <p>3 S: Modificar el estado del Computador a Disponible.</p> <p>4 S: Eliminar el registro de consumo y generar un registro de histórico.</p> <p>5 S: Enviar Instrucción de Cerrar Sesión de Consumo al computador seleccionado a través del subsistema de Comunicación para Bloquear el Computador.</p>
Postcondiciones	Estado del Computador actualizado a disponible. Instrucción enviada al Subsistema de Comunicación.
Variaciones	
Extensiones	
Cuestiones	

Caso de Uso Consultar Consumo



5	Consultar Consumo
Objetivo	Mostrar el Consumo de Horas Realizadas por un usuario y Horas disponibles.
Actores	Usuario
Precondiciones	Haberse dado el caso de uso <i>Identificación</i>
Pasos	1 A: Seleccionar Consultar Consumo 2 S: Mostrar información de Consumo.
Postcondiciones	
Variaciones	
Extensiones	
Cuestiones	

Caso de Uso Habilitar Sala

6	Habilitar Sala
Objetivo	Habilitar o desbloquear los computadores de una sala.
Actores	Profesor
Precondiciones	Debe haberse dado el caso de uso <i>Identificación</i>
Pasos	1. A: Pasar tarjeta por el lector. 2. S: Verificar Datos 3. S: Verificar Horario de Clases y/o Reservas 4. S: Modificar el estado de los Computadores de la sala a NO Disponible(Ocupado) 5. S: Crear un registro de consumo en la base de datos. 6. S: Enviar instrucción de Iniciar Sesión de Consumo al a todos los computadores de la sala seleccionada a través del Subsistema de Comunicación.
Postcondiciones	Estado de los computadores de la Sala quedan Actualizados a Ocupado.
Variaciones	
Extensiones	
Cuestiones	



Caso de Uso Deshabilitar Sala

7	Deshabilitar Sala
Objetivo	Deshabilitar o bloquear los computadores de una sala.
Actores	Usuario
Precondiciones	Estar Identificado en el Sistema como Administrador o Profesor.
Pasos	<ol style="list-style-type: none">1. A: Seleccionar Sala.2. A: Clic en la opción Deshabilitar Sala.3. S: Modificar el estado del Computador a Disponible.4. S: Eliminar el registro de consumo y generar un registro de histórico5. S: Enviar instrucción de Cerrar Sesión de Consumo a todos los computadores de la sala seleccionada a través del Subsistema de Comunicación
Postcondiciones	Estado de los computadores de la sala actualizado a disponible.
Variaciones	
Extensiones	
Cuestiones	



3.4.3 Subsistema Comunicación

En el diagrama de casos de uso el único actor es la aplicación.

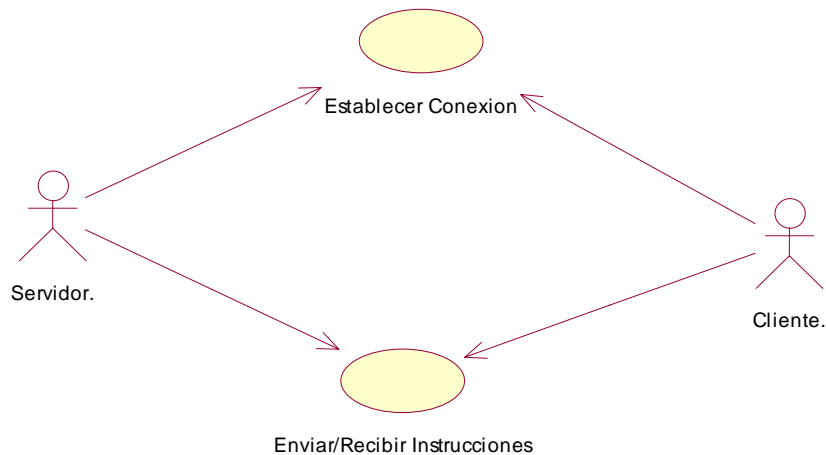


figura 3.6 Diagrama de Casos de Uso Comunicación

Caso de Uso Establecer Conexión

1	Establecer Conexión
Objetivo	Establecer una conexión entre la aplicación Servidor con la aplicación Cliente
Actores	Servidor (S) y Cliente (C)
Precondiciones	
Pasos	<ol style="list-style-type: none">1. S: Servidor se mantiene escuchando por peticiones de clientes.2. C: Envía una solicitud de conexión.3. S: Acepta la petición de conexión.4. S: Crea un registro con información del cliente (socket, dirección IP, etc.).5. S: Crea un hilo para manejar esta conexión independientemente.
Postcondiciones	Conexión establecida entre el servidor y el cliente.
Variaciones	<ol style="list-style-type: none">3.a. La petición ha sido rechazada:<ol style="list-style-type: none">2.a.1. Indicar error.



	2.a.2. Finalizar caso de uso.
Extensiones	
Cuestiones	

Caso de Uso Enviar / Recibir Instrucciones

2	Enviar / Recibir Instrucciones
Objetivo	Realizar un intercambio de instrucciones (conversación) entre la aplicación cliente y la aplicación servidor.
Actores	Servidor (S) y Cliente (C)
Precondiciones	
Pasos	<p>Servidor envía instrucciones (datos) al Cliente:</p> <ol style="list-style-type: none"> 1. S: Servidor envía una Instrucción. 2. C: Cliente recibe la instrucción. 3. C : Generar evento de recepción de datos (OnArrivalData). <p>Cliente envía instrucciones (datos) al Servidor:</p> <ol style="list-style-type: none"> 1. C: Cliente envía una instrucción. 2. S: Servidor recibe una la instrucción. 3. S: Generar evento de recepción de datos (OnArrivalData).
Postcondiciones	Intercambio de instrucciones realizado.
Variaciones	<ol style="list-style-type: none"> 1.a. Error en el envío de datos: <ol style="list-style-type: none"> 1.a.1. Indicar error. 1.a.2. Finalizar caso de uso. 2.a. Error en la recepción: <ol style="list-style-type: none"> 1.a.1. Indicar error. 1.a.2. Cerrar conexión cliente. 1.a.3. Finalizar caso de uso.
Extensiones	
Cuestiones	



3.4.4 Subsistema Cliente

En el diagrama de casos de uso el único actor es la aplicación.

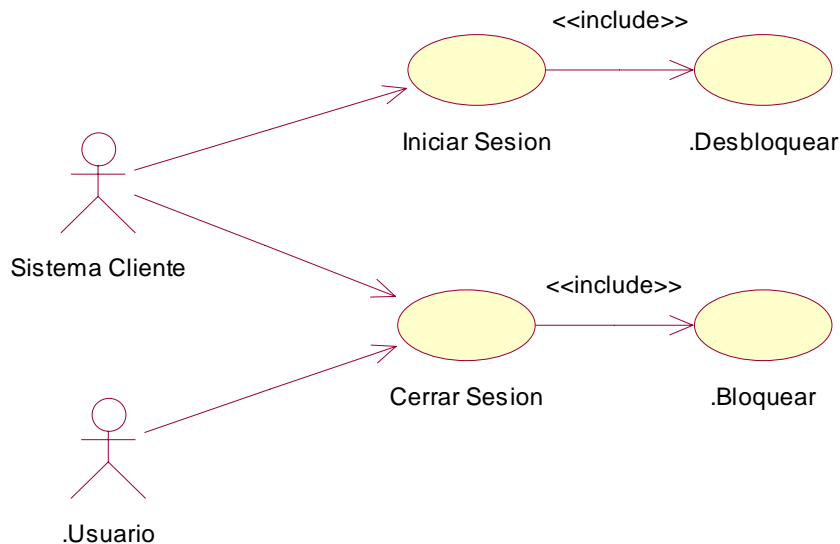


figura 3.7 Diagrama de Casos de Uso Cliente

Caso de Uso Iniciar Sesión

1	Iniciar Sesión
Objetivo	Recibir una orden desde el Servidor para que inicie una sesión.
Actores	Comunicación
Precondiciones	Debe haber una conexión preestablecida entre la aplicación servidor con la aplicación cliente.
Pasos	1. S: Caso de Uso Desbloquear. 2. S: Iniciar cronómetro local. 3. S: Visualizar en pantalla el cuadro de diálogo. 4. S: Ocultar formulario que cubre el escritorio.
Postcondiciones	Computador habilitado.
Variaciones	2.a. Al cumplirse el tiempo reservado se da el caso de uso Cerrar Sesión.
Extensiones	
Cuestiones	



Caso de Uso Cerrar Sesión

2	Cerrar Sesión
Objetivo	Recibir una orden desde el servidor para cerrar una sesión.
Actores	Comunicación
Precondiciones	Debe haber una conexión preestablecida entre la aplicación servidor con la aplicación cliente. Debe haberse dado el caso de uso Iniciar Sesión.
Pasos	<ol style="list-style-type: none">1. S: Caso de uso Bloquear.2. S: Finalizar contador.3. S: Ocultar Formulario Principal.4. S: Mostrar Formulario de Bloqueo.
Postcondiciones	Computador bloqueado.
Variaciones	
Extensiones	
Cuestiones	

Caso de Uso Bloquear

3	Bloquear.
Objetivo	Inhabilitar el computador.
Actores	Aplicación Cliente.
Precondiciones	
Pasos	<ol style="list-style-type: none">1. S: Bloquear escritorio de Windows2. S: Bloquear teclado.3. S: Bloquear Administrador de Tareas de Windows.
Postcondiciones	Computador inhabilitado.
Variaciones	
Extensiones	
Cuestiones	

Caso de Uso Desbloquear



4	Desbloquear.
Objetivo	Habilitar el computador.
Actores	Aplicación Cliente.
Precondiciones	
Pasos	<ol style="list-style-type: none">1. S: Habilitar escritorio de Windows2. S: Habilitar teclado.3. S: Habilitar Administrador de Tareas de Windows
Postcondiciones	Computador habilitado.
Variaciones	
Extensiones	
Cuestiones	



3.5 DIAGRAMA DE CLASES INICIAL

Aquí se presentan los diagramas de clases propuestos inicialmente. Se ha realizado únicamente este diagrama para el subsistema de Administración del Centro de Cómputo, debido a que el resto de subsistemas no poseen muchas clases, los mismos que se analizarán el capítulo de diseño con más profundidad y detalle.

El siguiente diagrama de clases se ha elaborado a partir de los casos de uso definidos para el subsistema Servidor:

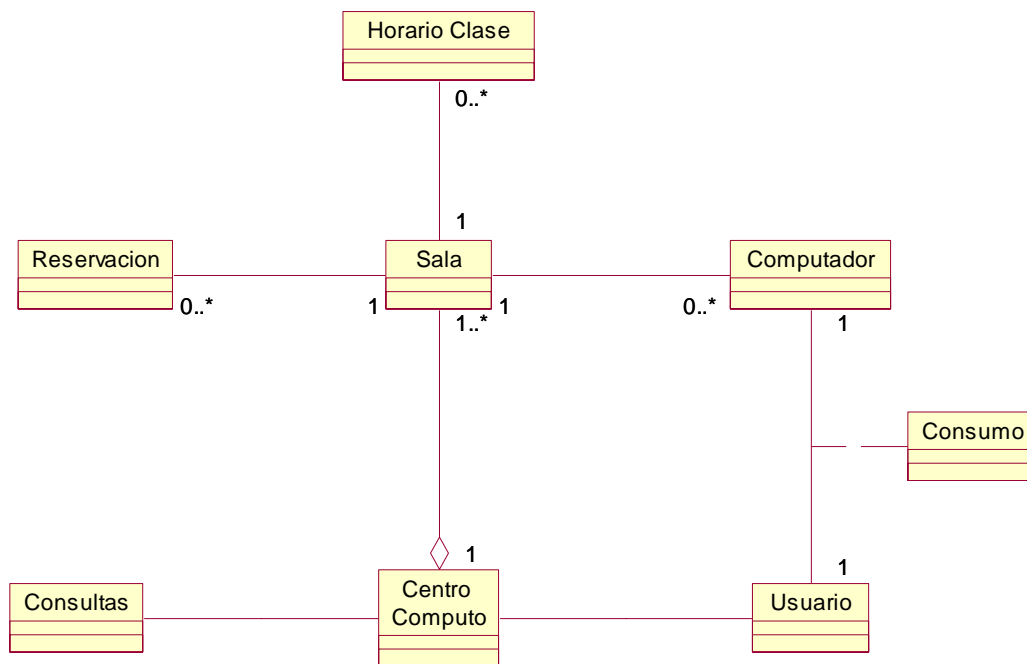


figura 3.8 Diagrama de Clases Subsistema Administración Centro Cómputo

A continuación veremos los diagramas de secuencia y colaboración de los distintos casos de uso.



3.6 DIAGRAMAS DE INTERACCION

En este apartado se expondrá los diagramas de secuencia y colaboración que representan a los distintos casos de uso. Estos diagramas muestran la interacción de las distintas clases, así como el paso de mensajes y la secuencia de mensajes que produce cada uno de los casos de uso.

3.6.1 Gestión de Usuarios

A continuación se exponen los diagramas de secuencia y colaboración de los casos de uso correspondientes al subsistema de gestión de usuarios.

Caso de Uso Crear Usuario

Diagrama de Secuencia:

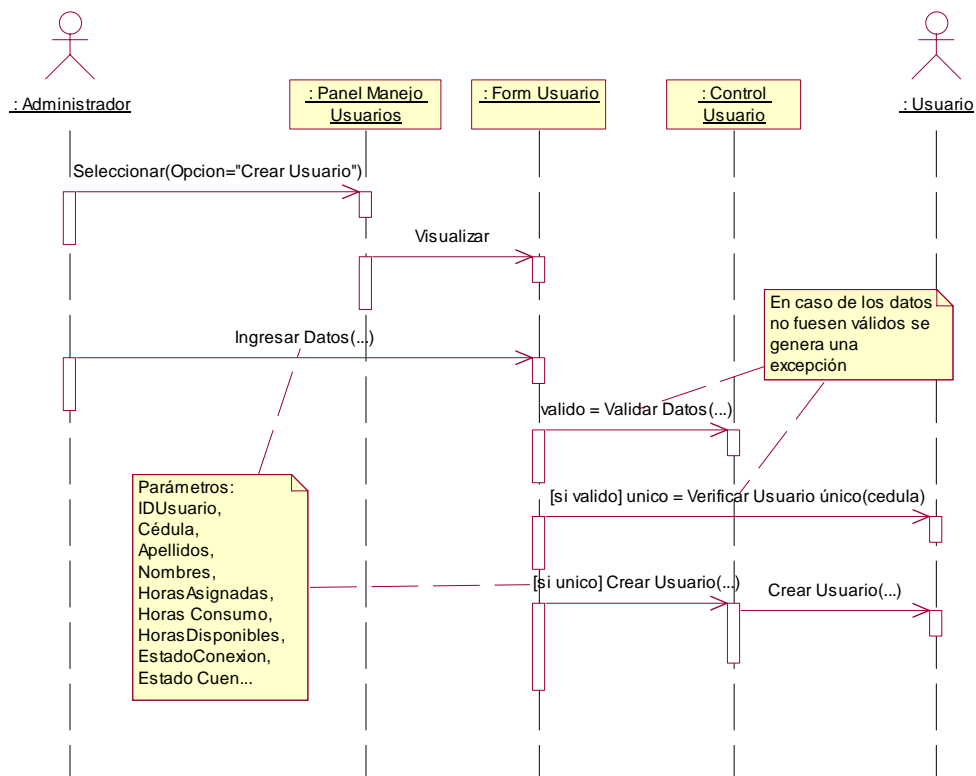


figura 3.9 Diagrama de Secuencias Crear Usuario



Diagrama de Colaboración:

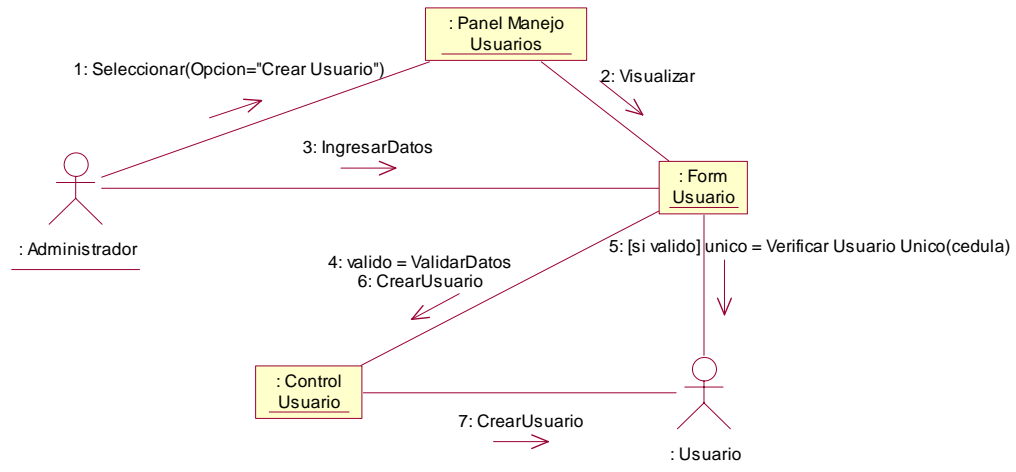


figura 3.10 Diagrama de Colaboración Crear Usuario

Caso de Uso Modificar Usuario

Diagrama de Secuencia:

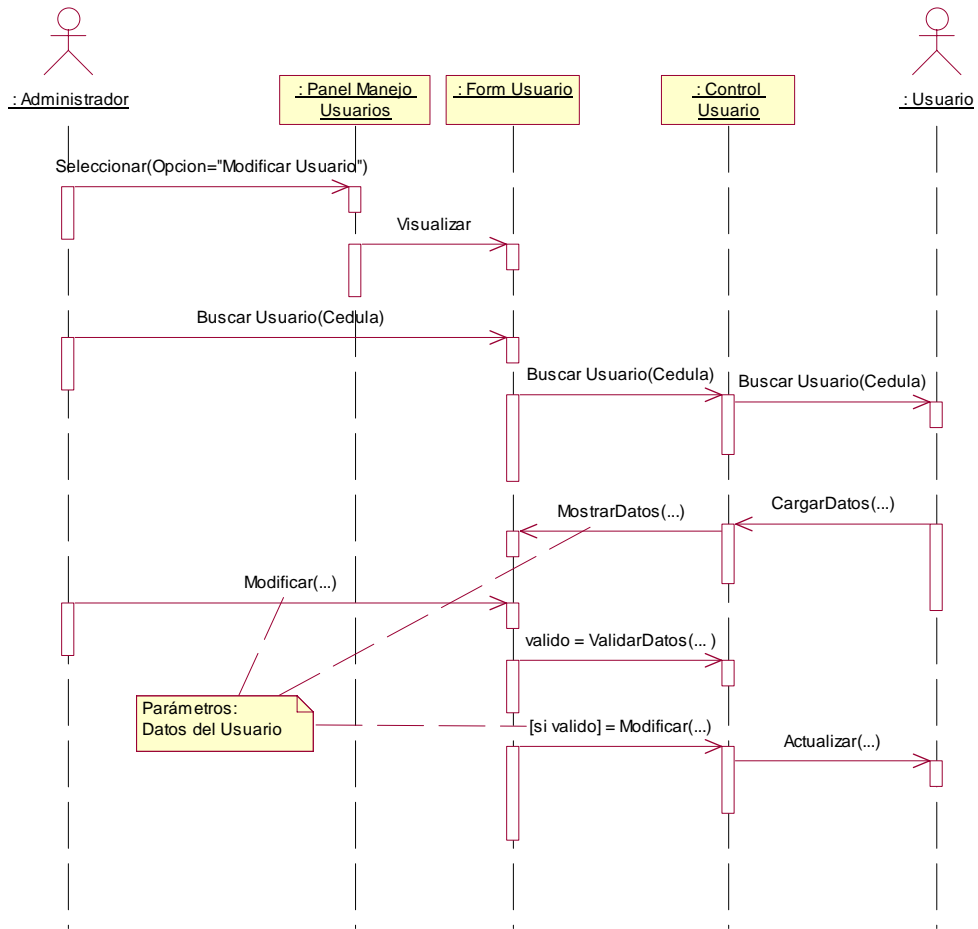


figura 3.11 Diagrama de Secuencia Modificar Usuario

Diagrama de Colaboración:

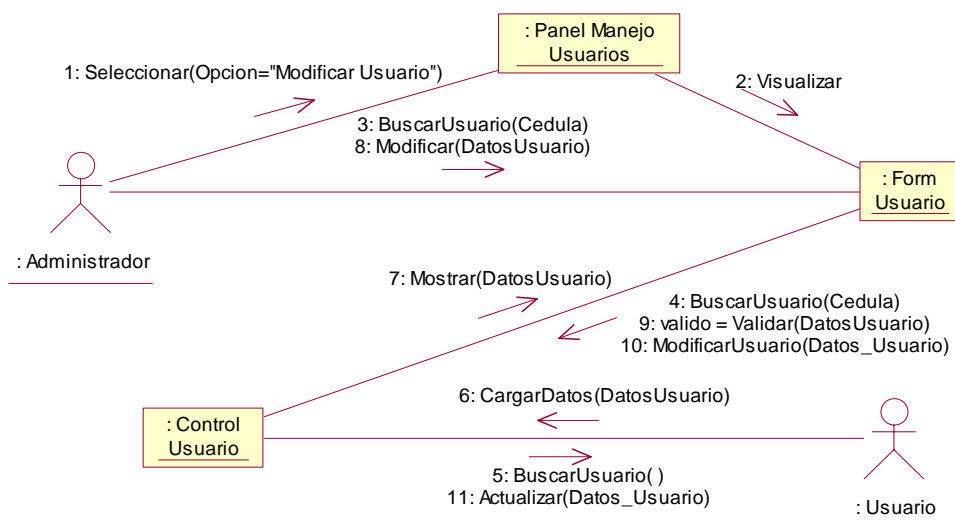


figura 3.12 Diagrama de Colaboración Modificar Usuario



Caso de Uso Eliminar Usuario

Se omiten los diagramas de este caso de uso por ser prácticamente similar al caso de uso Modificar Usuario, diferenciándose únicamente por la actividad de Validar Datos.



3.6.2 Gestión de Máquinas o Computadores.

Caso de Uso Agregar Computador.

Diagrama de Secuencia:

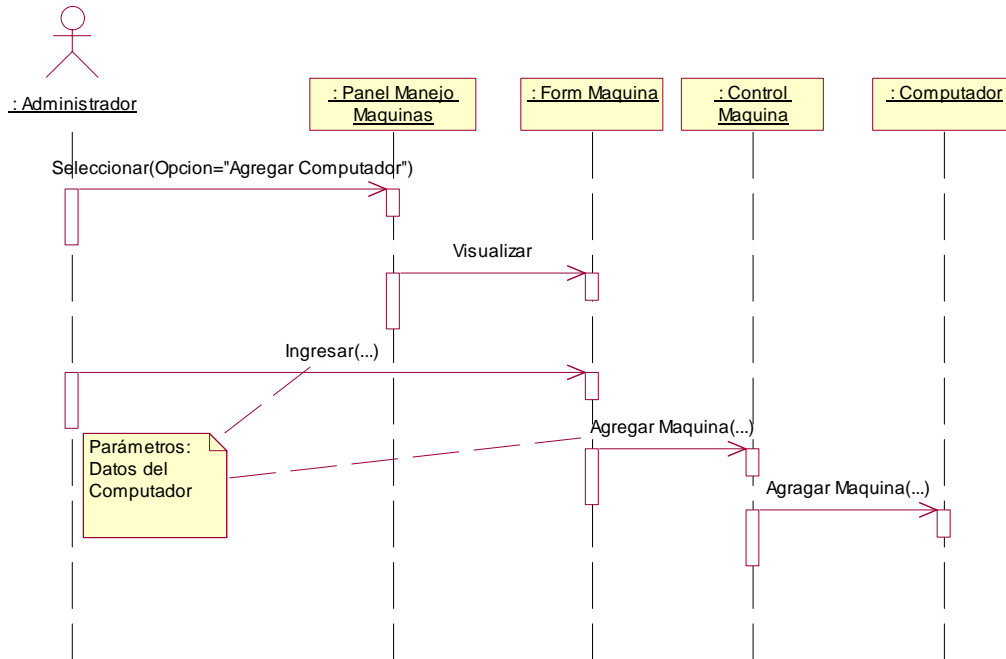


figura 3.13 Diagrama de Secuencia Agregar Computador

Diagrama de Colaboración:

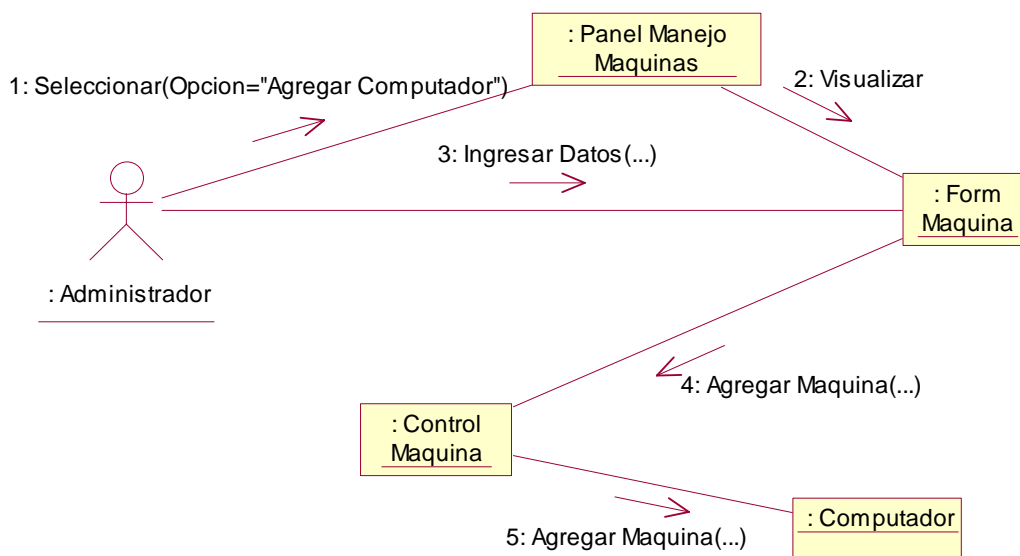




figura 3.14 Diagrama de Colaboración Agregar Computador

Caso de Uso Modificar Computador.

Diagrama de Secuencia:

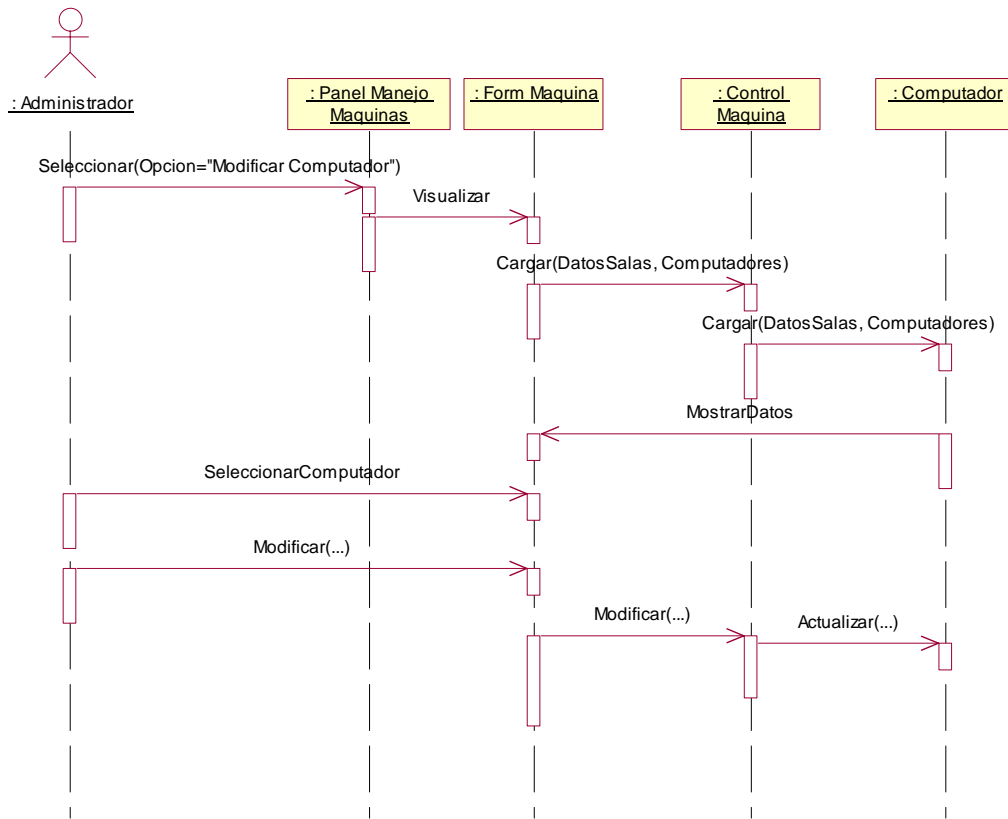


figura 3.15 Diagrama de Secuencia Modificar Computador

Diagrama de Colaboración:

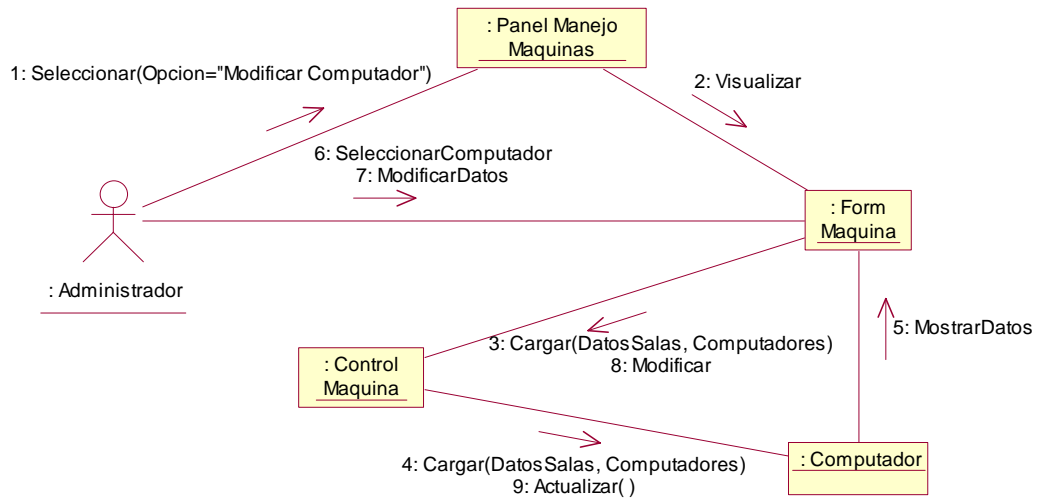


figura 3.16 Diagrama de Colaboración Modificar Computador

Caso de Uso Eliminar Computador.

Se omiten los diagramas de este caso de uso por ser análogo al caso de uso Modificar Computador.



3.6.3 Gestión Salas y Reservaciones.

Los diagramas para los casos de uso de crear sala, modificar sala y eliminar sala, se omitirán por ser análogos a los diagramas de Gestión de Máquinas y no aportan ningún aspecto nuevo.

Reservaciones

Caso de Uso Crear Reservación.

Diagrama de Secuencia:

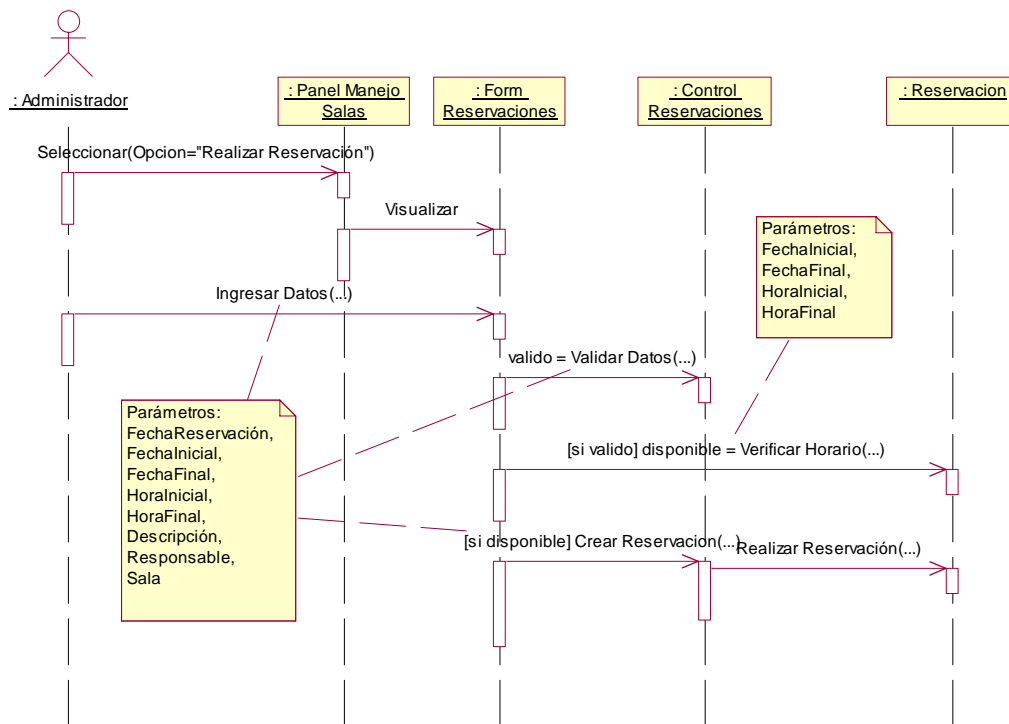


figura 3.17 Diagrama de Secuencia Crear Reservación



Diagrama de Colaboración:

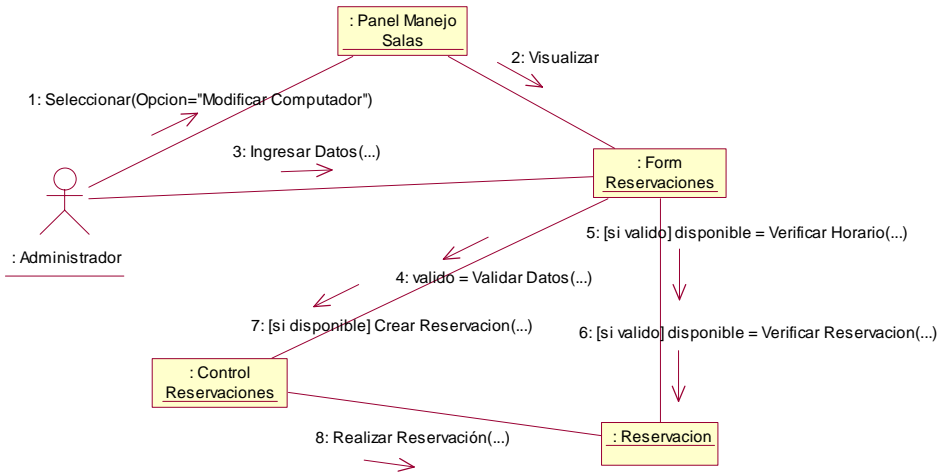


figura 3.18 Diagrama de Colaboración Crear Reservación

Caso de Uso Modificar Reservación.

Diagrama de Secuencia:

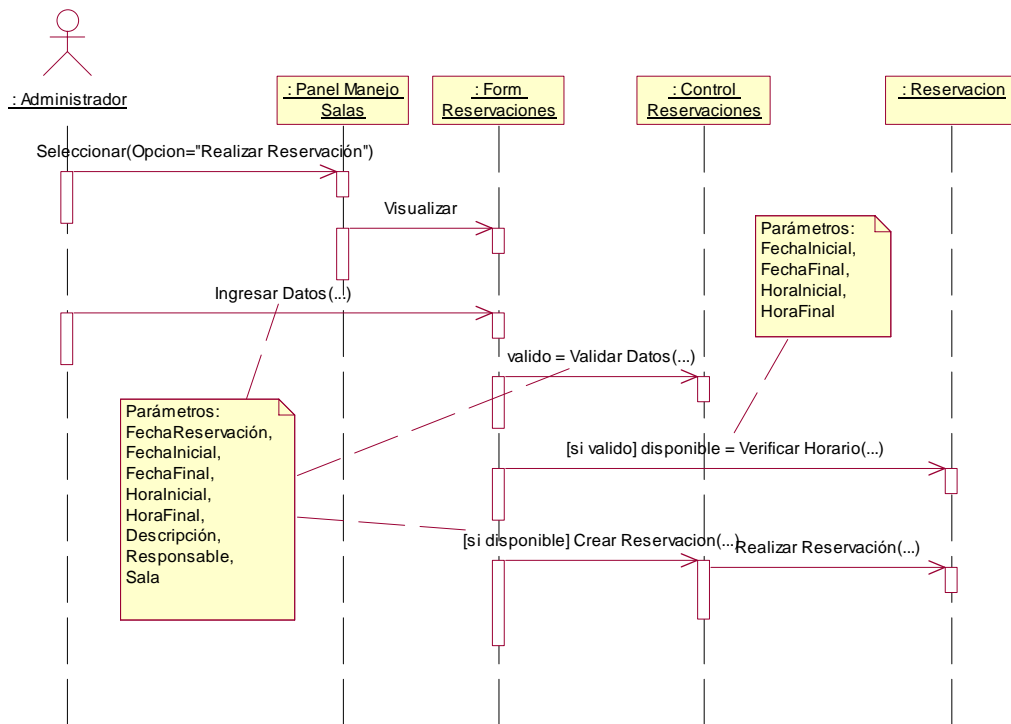




figura 3.19 Diagrama de Secuencia Modificar Reservación

Diagrama de Colaboración:

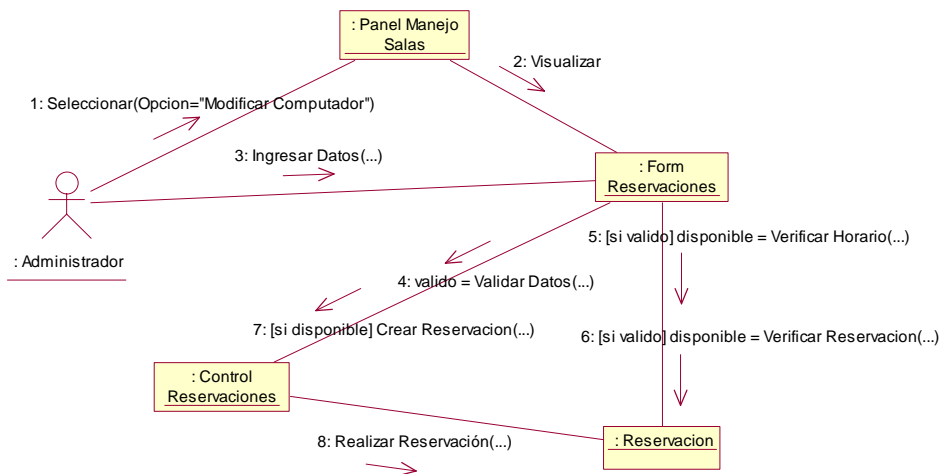


figura 3.20 Diagrama de Colaboración Moficiar Reservación

Caso de Uso Eliminar Reservación.

Se omiten los diagramas de este caso de uso por ser análogo al caso de uso Modificar Reservación.



Horarios

Caso de Uso Nuevo Horario.

Diagrama de Secuencia:

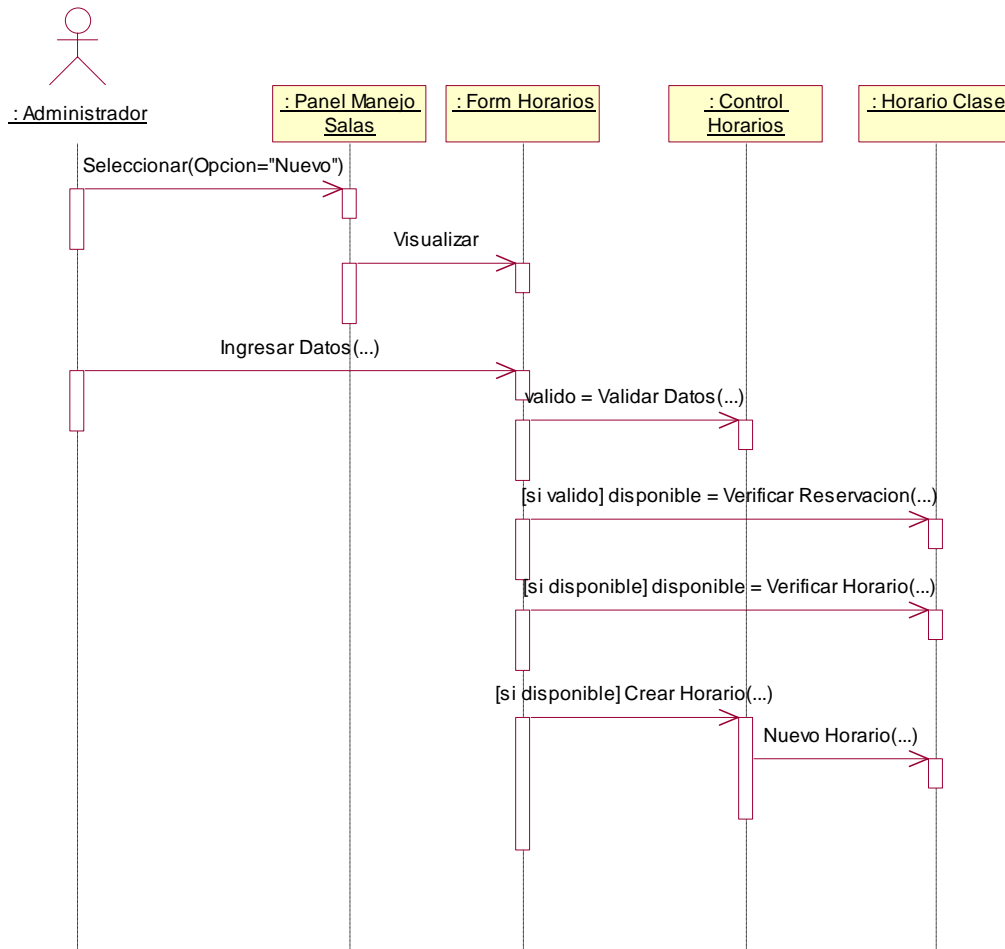


figura 3.21 Diagrama de Secuencia Nuevo Horario



Diagrama de Colaboración:

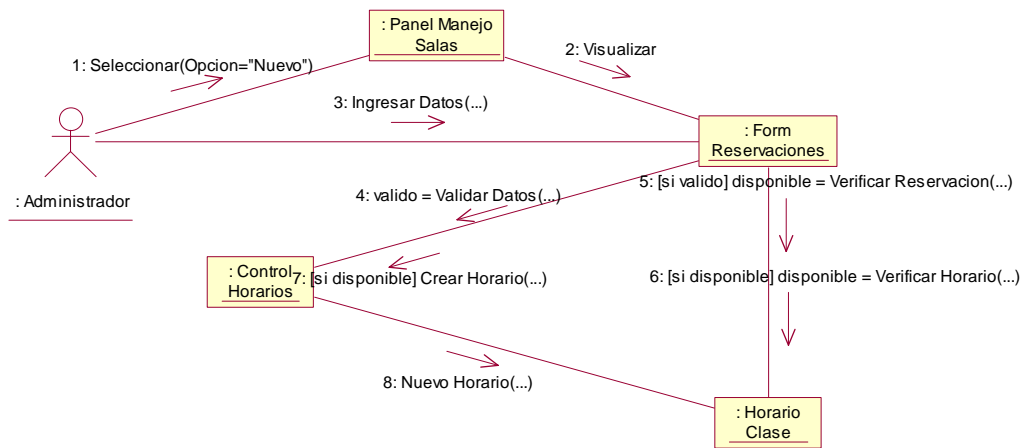


figura 3.22 Diagrama de Colaboración Nuevo Horario

Caso de Uso Modificar Horario.

Diagrama de Secuencia:

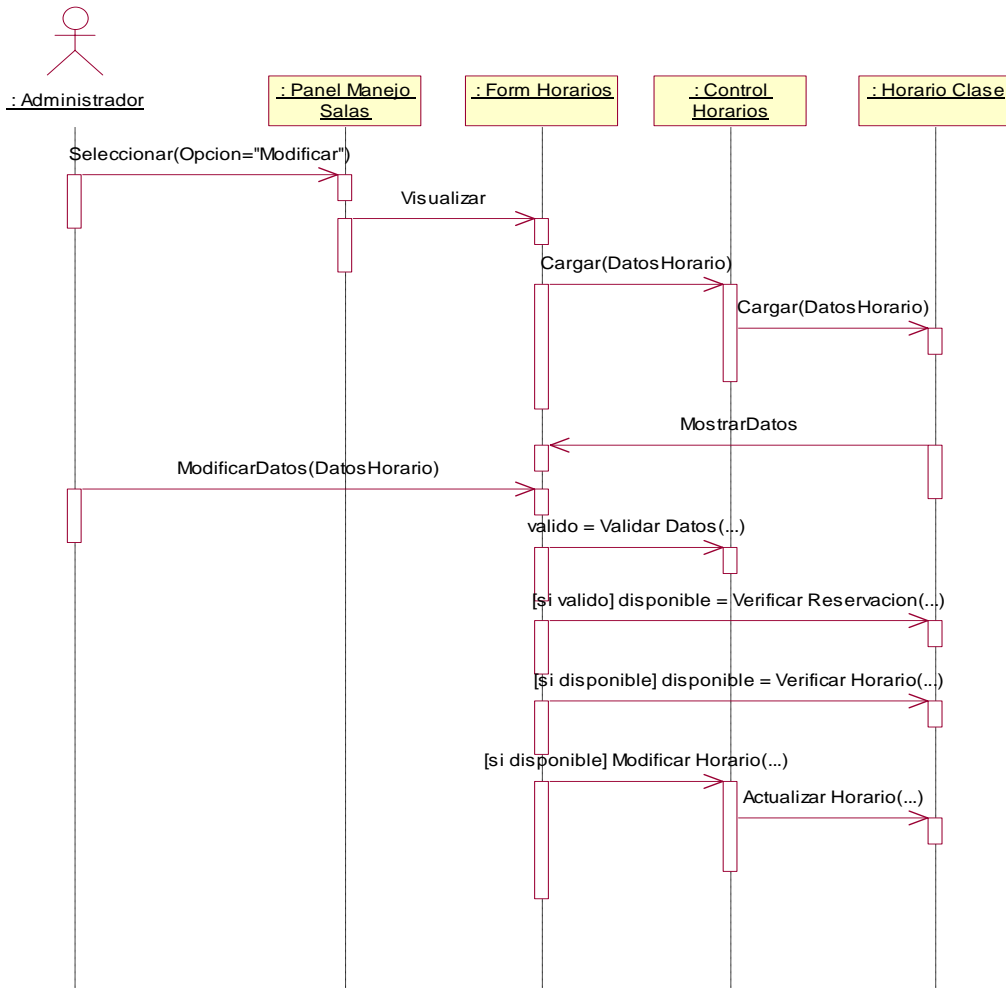


figura 3.23 Diagrama de Secuencia Modificar Horario

Diagrama de Colaboración:

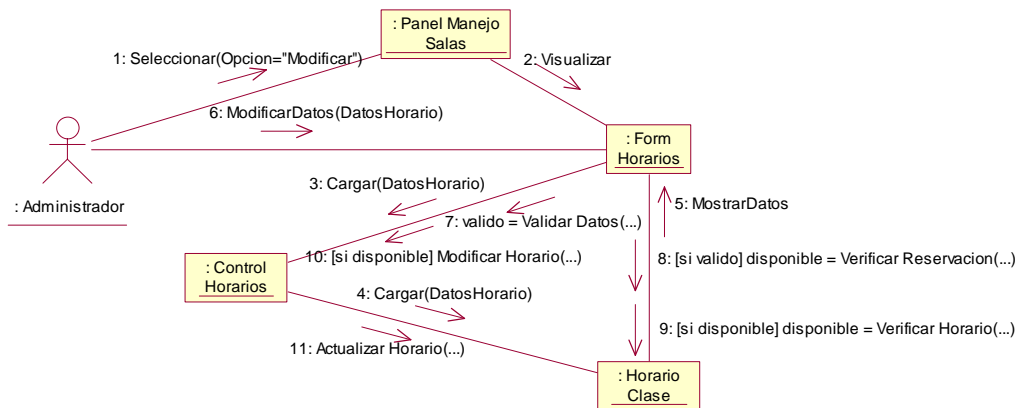




figura 3.24 Diagrama de Colaboración Modificar Horario

Caso de Uso Eliminar Horario.

Se omiten los diagramas de este caso de uso por ser similar al caso de uso Modificar Horario.



3.6.4 Subsistema Acceso al Sistema

A continuación se exponen los diagramas de secuencia y colaboración de los casos de uso correspondientes al subsistema de acceso al sistema.

Caso de Uso Identificación

Diagrama de Secuencia:

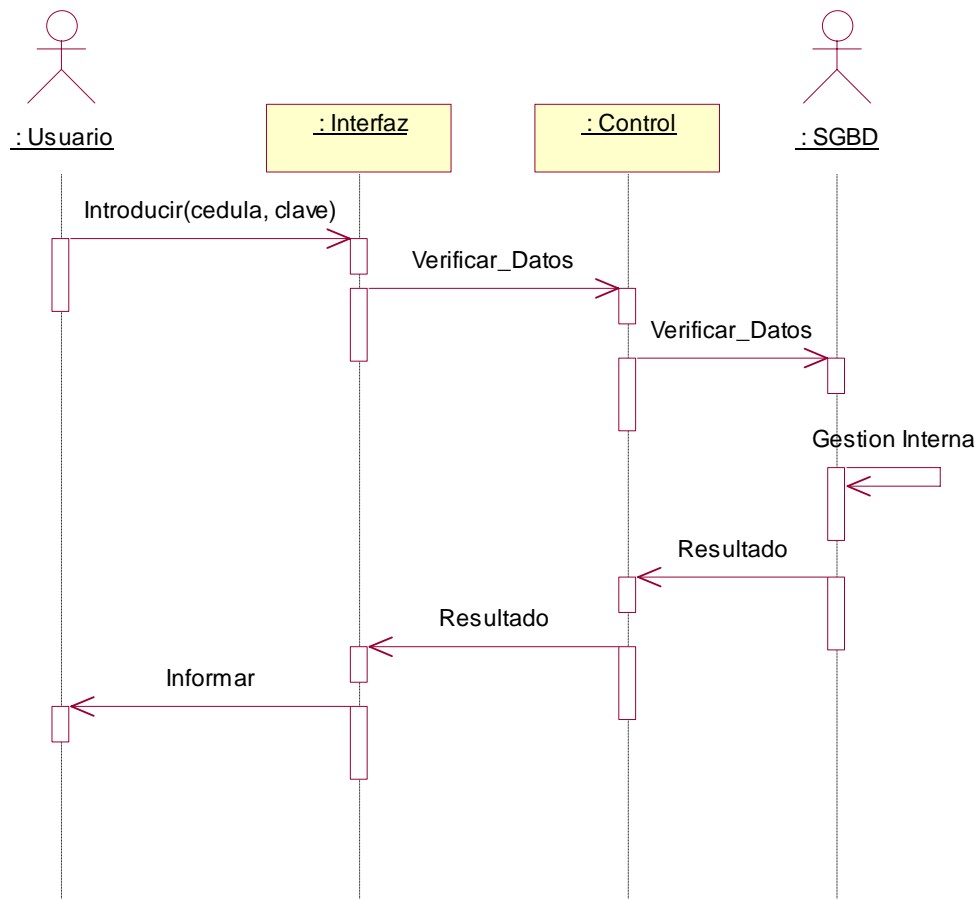


figura 3.25 Diagrama de Secuencias Identificación



Diagrama de Colaboración:

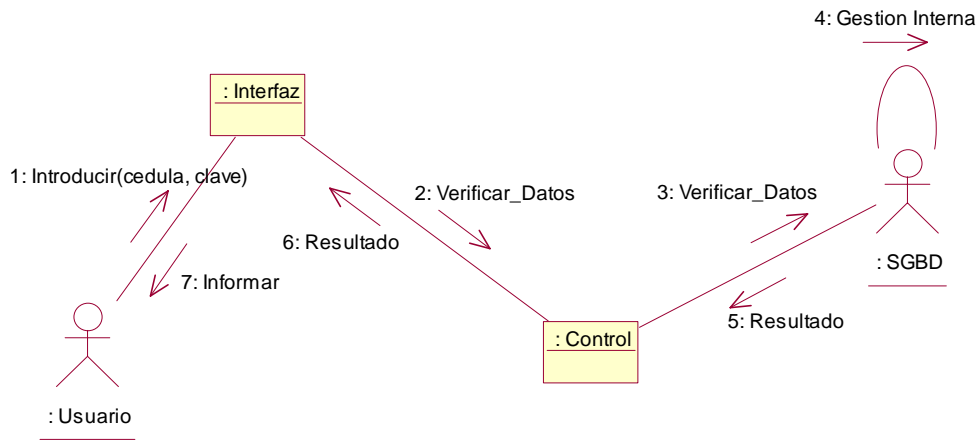


figura 3.26 Diagrama de Colaboración Identificación

Caso de Uso Visualizar Estado Máquina

Diagrama de Secuencia:

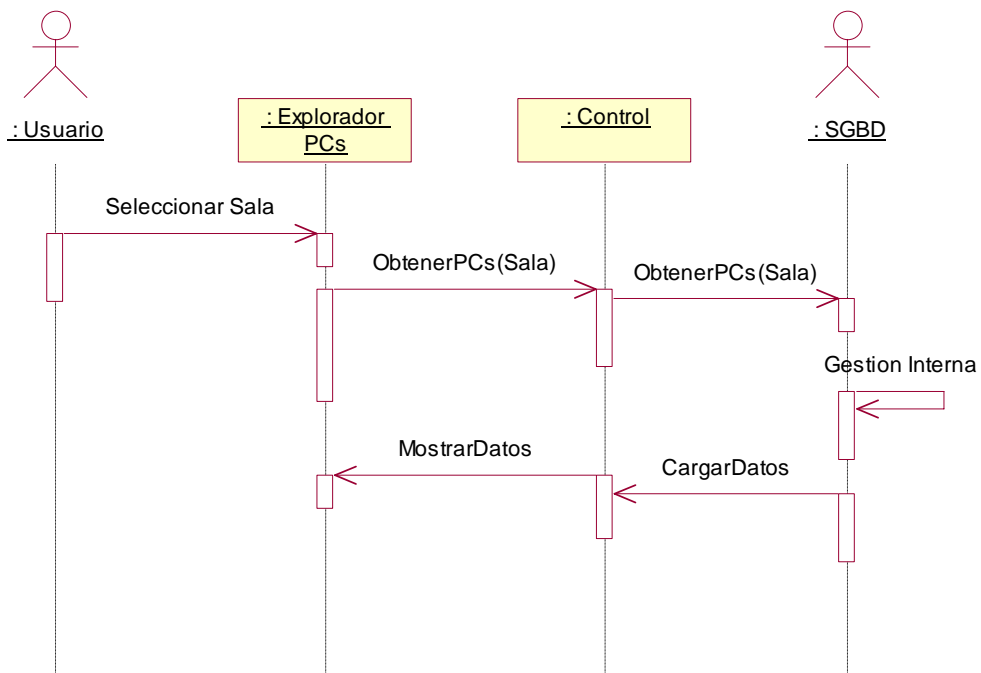


figura 3.27 Diagrama de Secuencias Visualizar Estado Máquina



Diagrama de Colaboración:

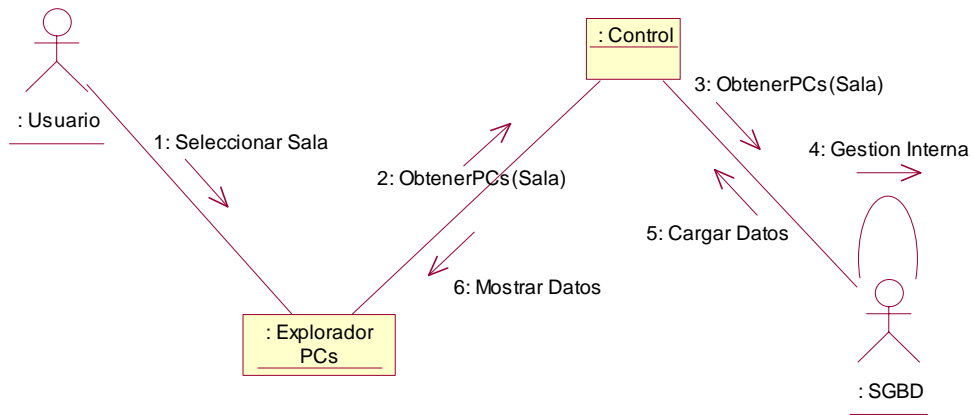


figura 3.28 Diagrama de Colaboración Visualizar Estado de un Computador

Caso de Uso Iniciar Sesión Consumo

Diagrama de Secuencia:

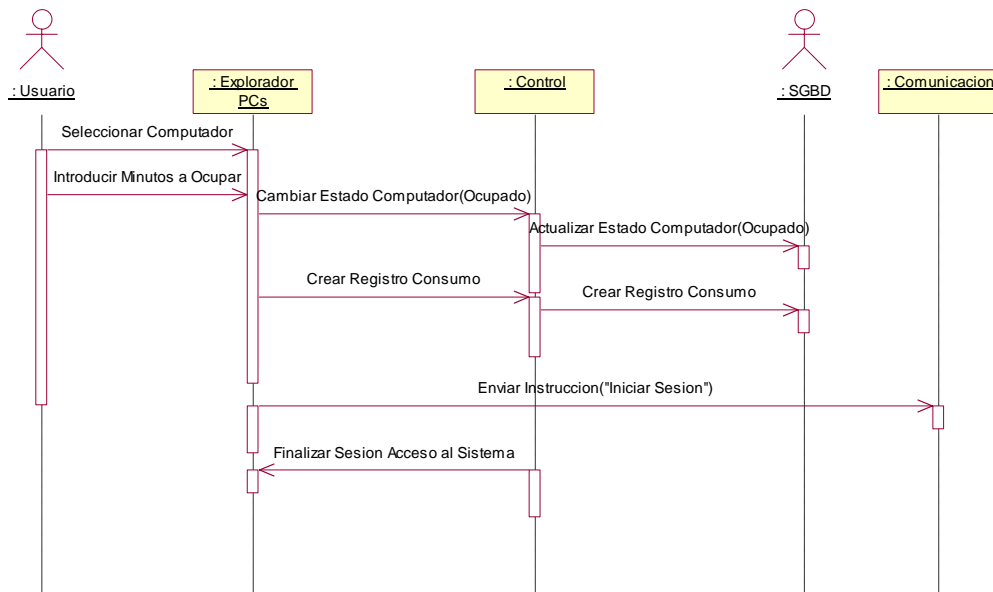


figura 3.29 Diagrama de Secuencias Caso de Uso Iniciar Sesión Consumo



Diagrama de Colaboración:

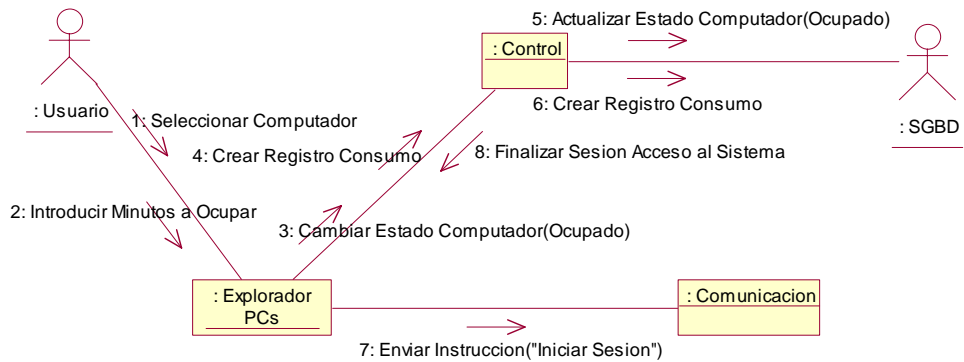


figura 3.30 Diagrama de Colaboración Iniciar Sesión Consumo

Caso de Uso Cerrar Sesión Consumo

Diagrama de Secuencia:

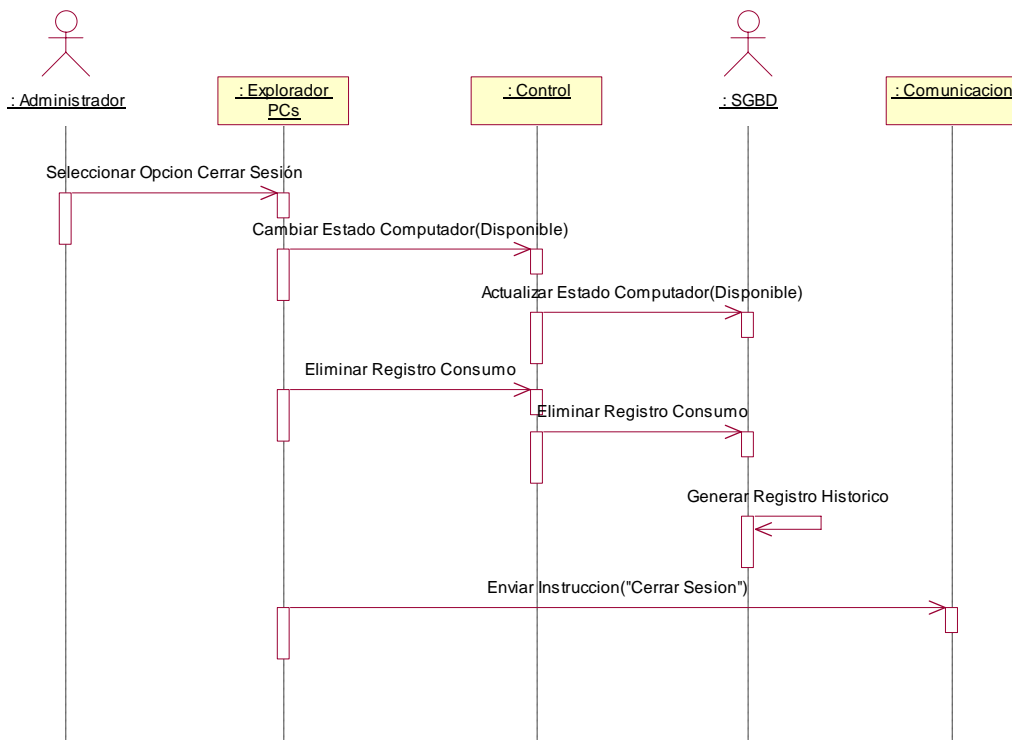


figura 3.31 Diagrama de Secuencias Caso de Uso Cerrar Sesión Consumo



Diagrama de Colaboración:

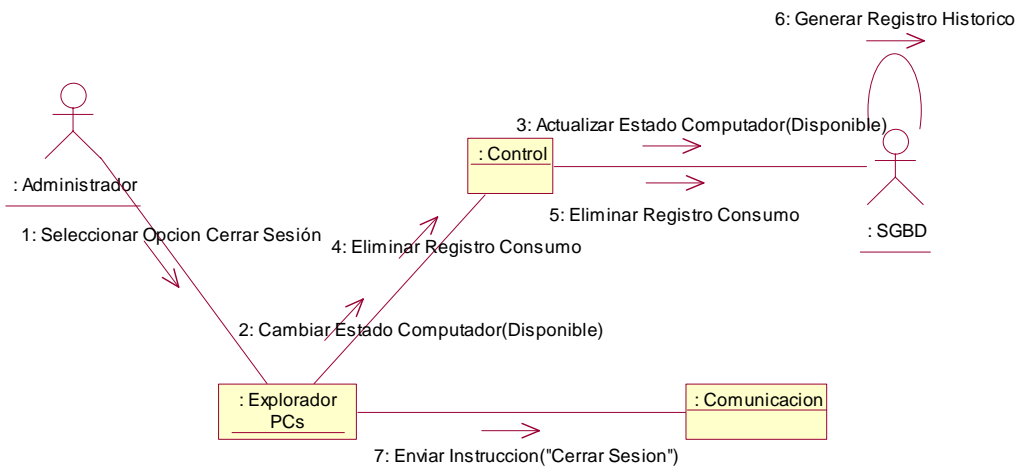


figura 3.32 Diagrama de Colaboración Cerrar Sesión Consumo

Caso de Uso Habilitar Sala

Diagrama de Secuencia:

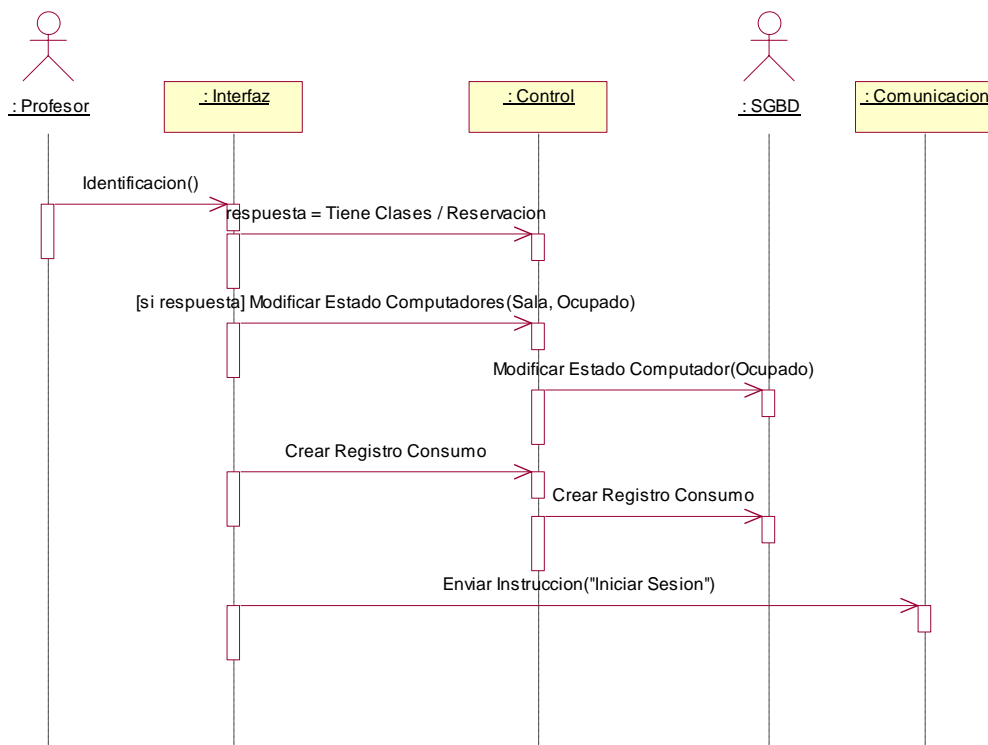


figura 3.33 Diagrama de Secuencias Caso de Uso Habilitar Sala



Diagrama de Colaboración:

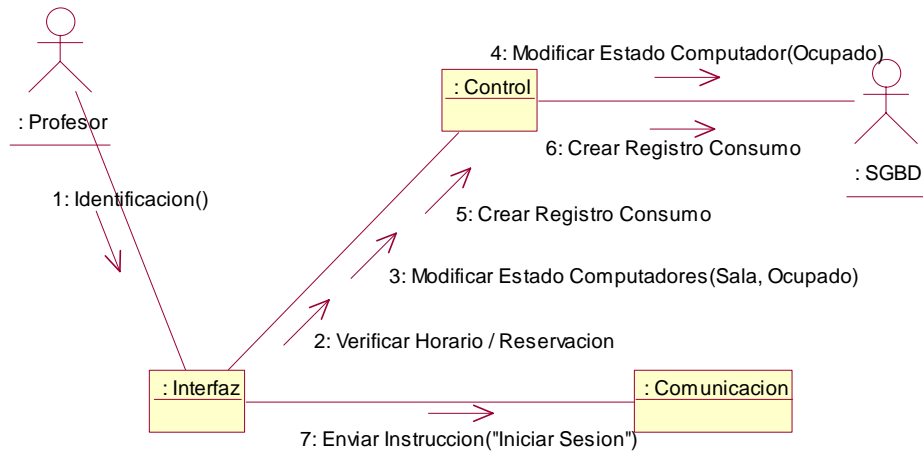


figura 3.34 Diagrama de Colaboración Habilitar Sala

Caso de Uso Deshabilitar Sala

Diagrama de Secuencia:

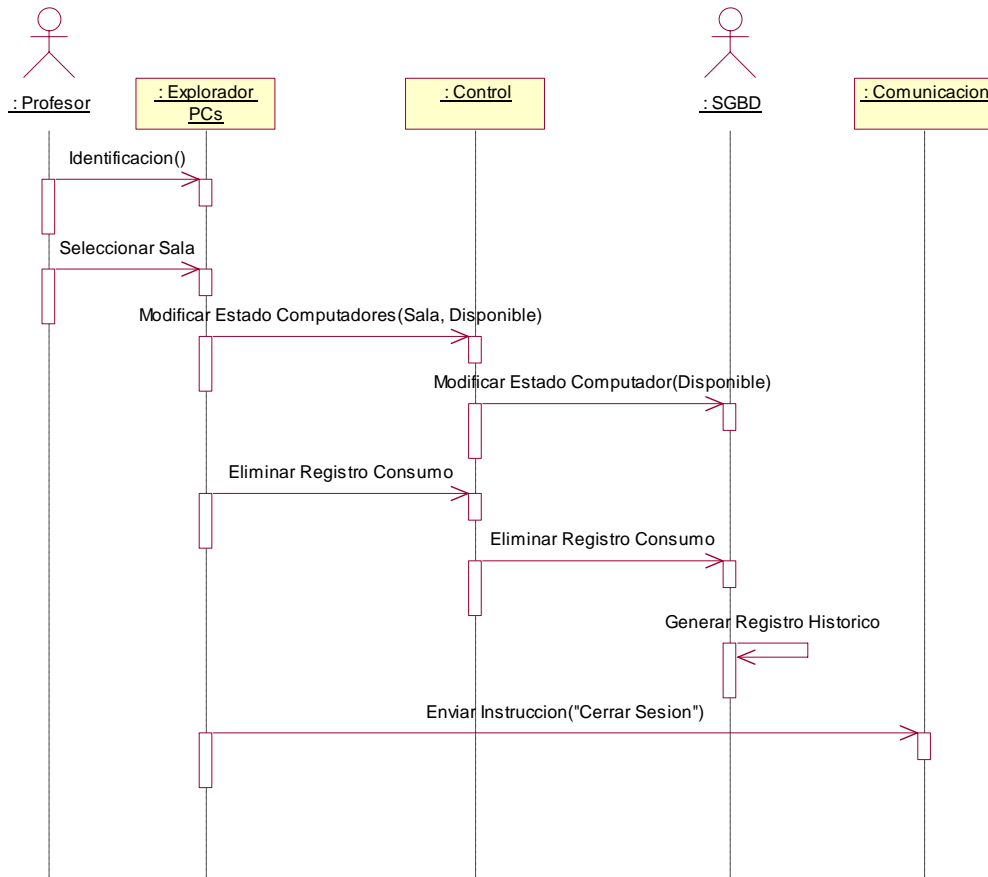


figura 3.35 Diagrama de Secuencias Caso de Uso Dehabilitar Sala

Diagrama de Colaboración:

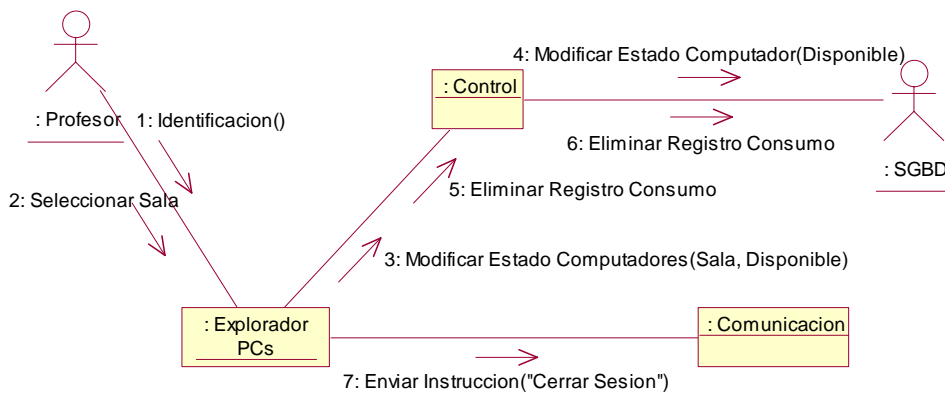


figura 3.36 Diagrama de Colaboración Deshabilitar Sala



3.6.5 Subsistema Comunicación (Cliente – Servidor)

A continuación se exponen los diagramas de secuencia y colaboración de los casos de uso correspondientes al subsistema de comunicación entre el cliente y el servidor.

Caso de Uso Establecer Conexión

Diagrama de Secuencia:

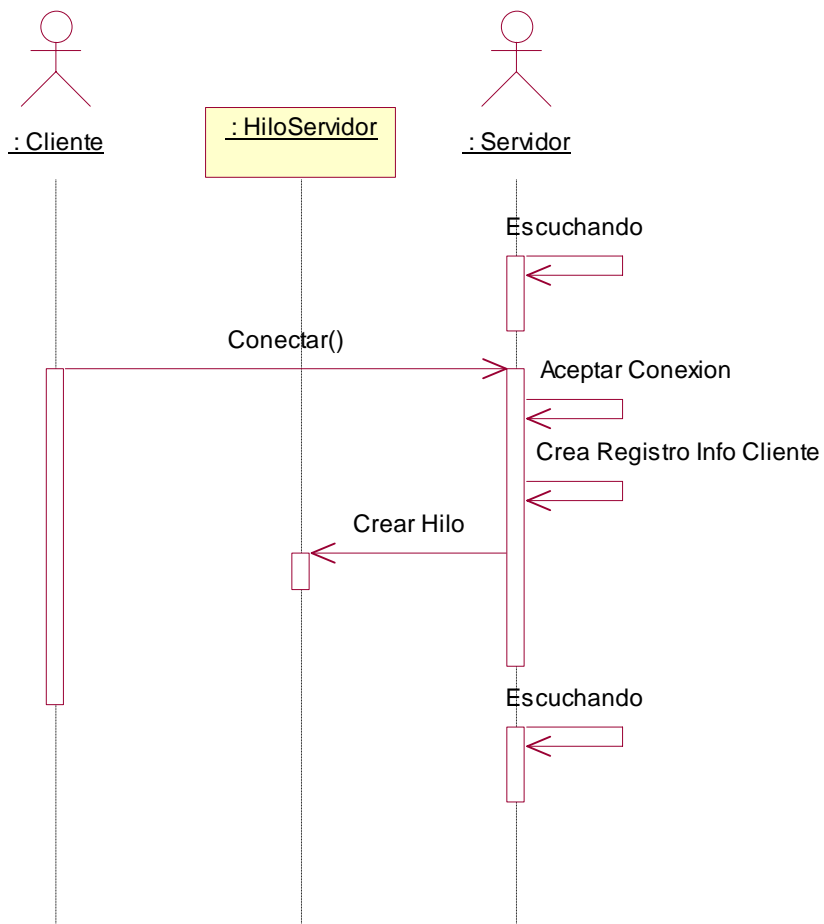


figura 3.37 Diagrama de Secuencias Establecer Conexión



Diagrama de Colaboración:

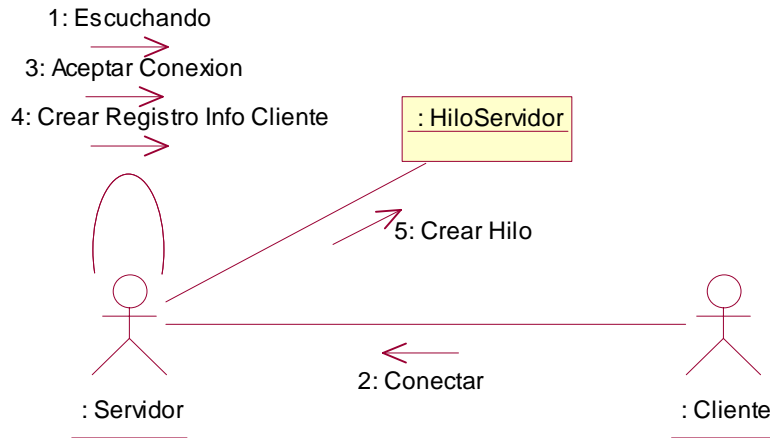


figura 3.38 Diagrama de Colaboración Establecer Conexión

Caso de Uso Enviar / Recibir Datos

Diagrama de Secuencia:

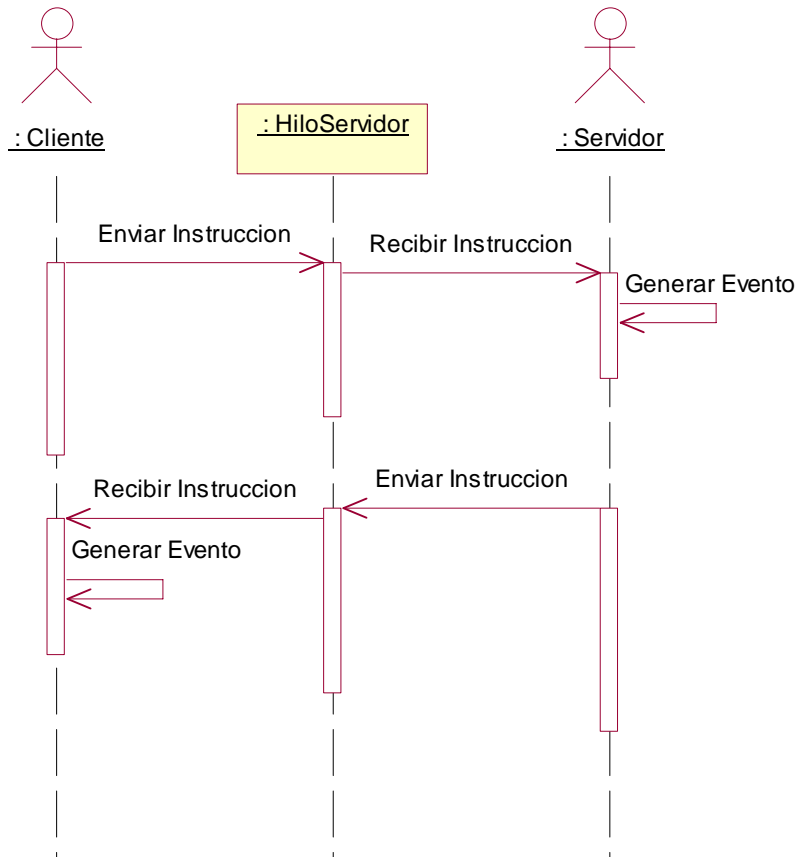
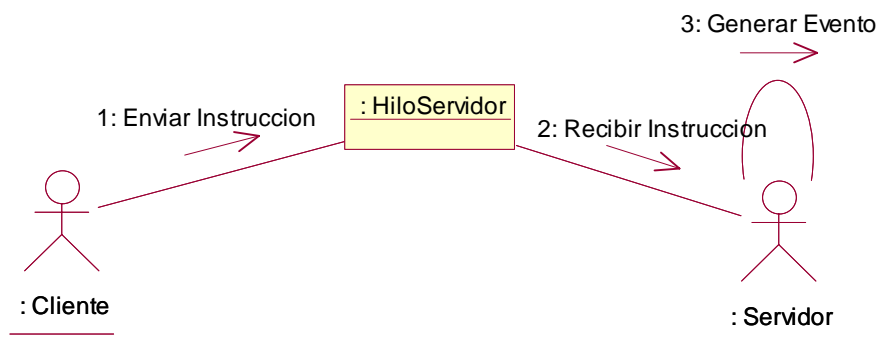


figura 3.39 Diagrama de Secuencias Enviar / Recibir Instrucciones

Diagrama de Colaboración:



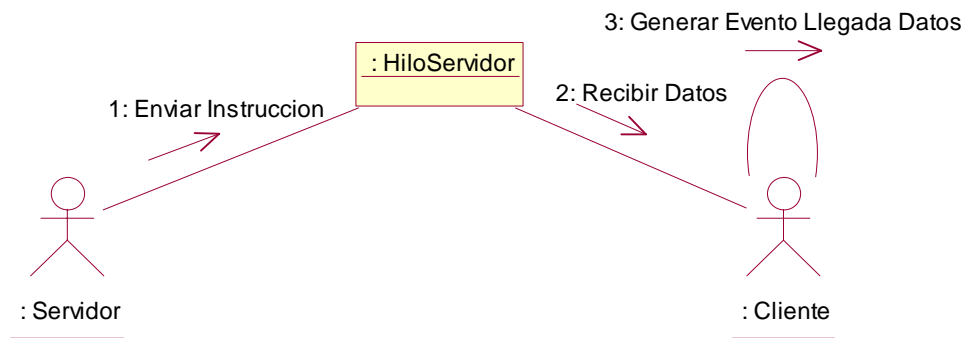


figura 3.40 Diagrama de Colaboración Enviar / Recibir Instrucciones



3.6.6 Subsistema Cliente

A continuación se exponen los diagramas de secuencia y colaboración de los casos de uso correspondientes al subsistema de cliente.

Caso de Uso Iniciar Sesión

Diagrama de Secuencia:

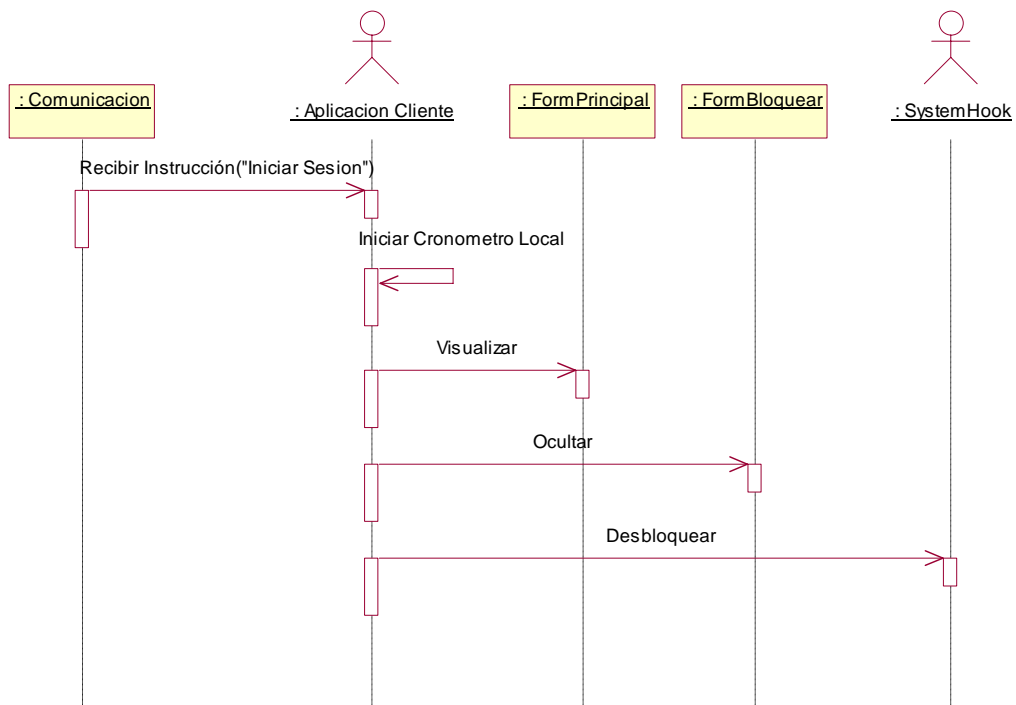


figura 3.41 Diagrama de Secuencias Iniciar Sesión



Diagrama de Colaboración:

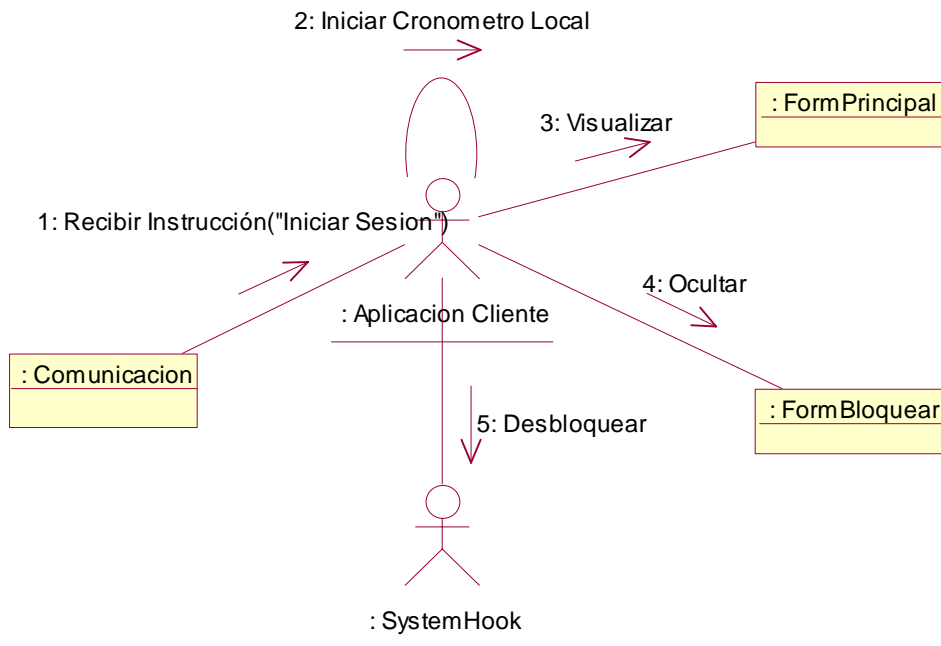


figura 3.42 Diagrama de Colaboración Iniciar Sesión

Caso de Uso Cerrar Sesión

Diagrama de Secuencia:

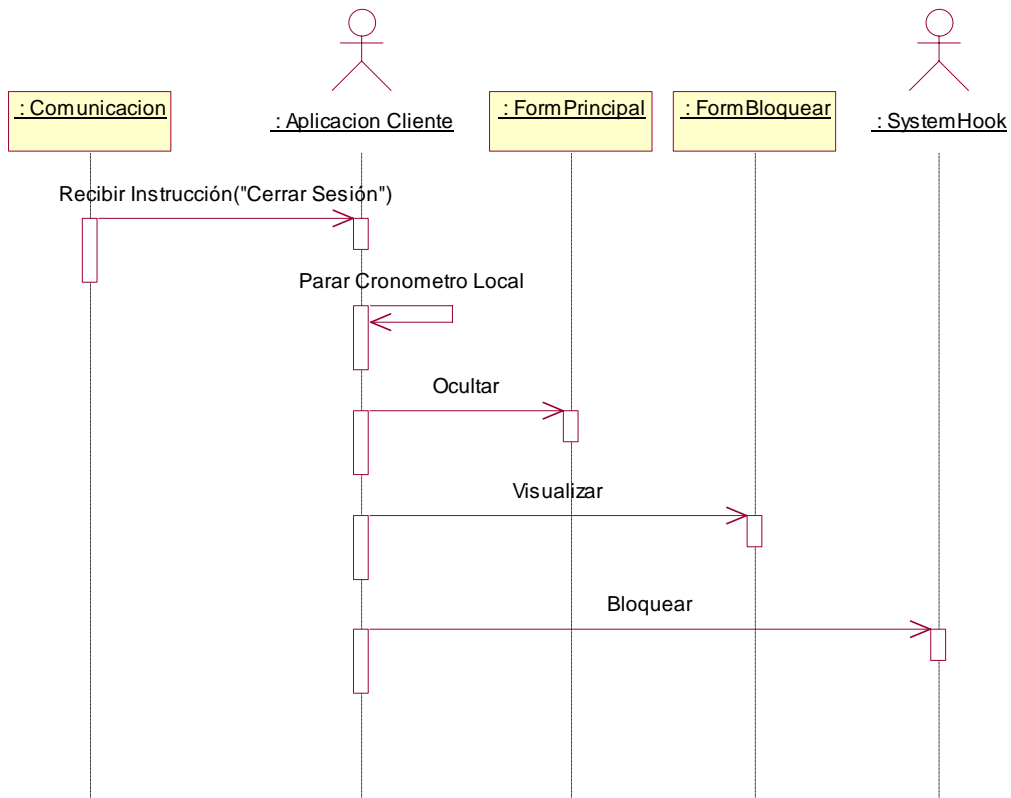


figura 3.43 Diagrama de Secuencias Cerrar Sesión

Diagrama de Colaboración:

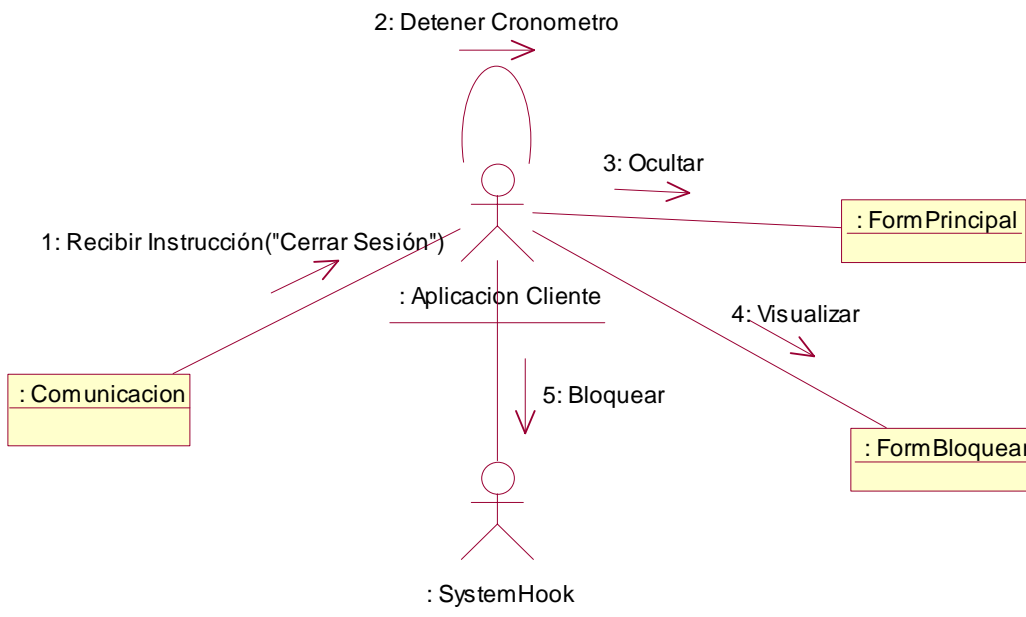


figura 3.44 Diagrama de Colaboración Cerrar Sesión



Caso de Uso Bloquear

Diagrama de Secuencia:

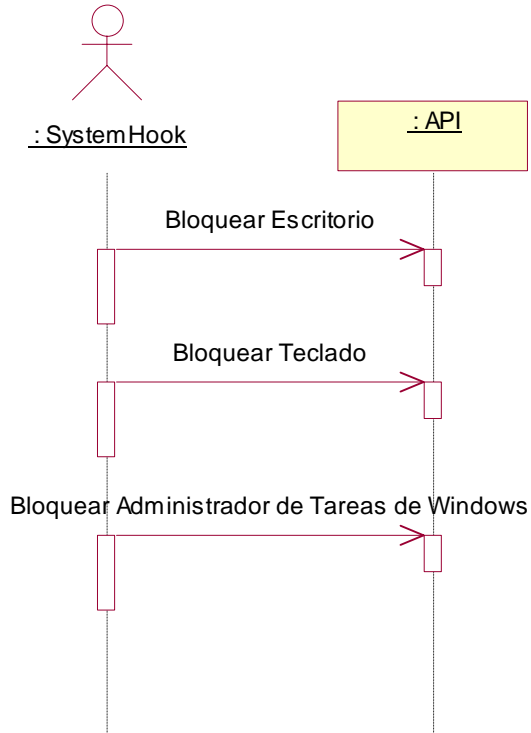


figura 3.45 Diagrama de Secuencias Bloquear

Caso de Uso Desbloquear

Diagrama de Secuencia:

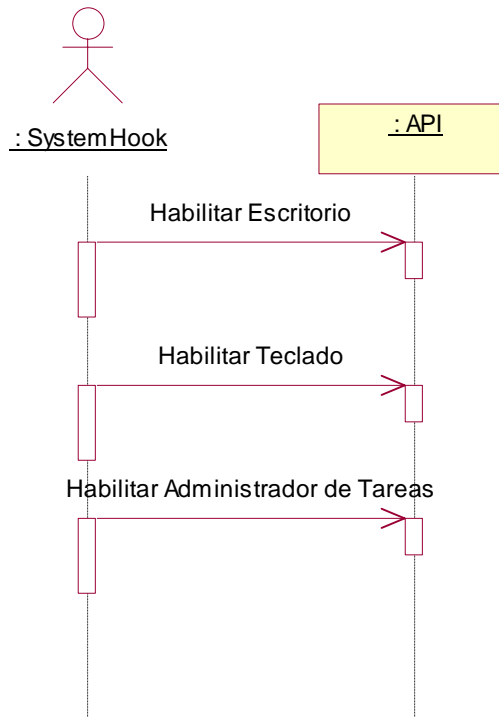


figura 3.46 Diagrama de Secuencias Caso de Uso Desbloquear.

4. DISEÑO DEL SISTEMA



4 DISEÑO DEL SISTEMA

4.1 INTRODUCCIÓN

En este apartado se presentará en detalle todo el diseño del sistema, de sus distintos subsistemas. Abordaremos los siguientes aspectos:

- Diccionario de Clases del Sistema.
- Diseño del Subsistema Administración del Centro de Cómputo..
- Diseño del Subsistema de Comunicación.
- Diseño del Subsistema Cliente.

4.1.1 Decisiones de Alto Nivel

Conforme se iba diseñando el proyecto, se han presentado diversos problemas que conforme al avance del proyecto se han resuelto.

- **La problemática de los bloqueos.**

Uno de los principales problemas encontrados al diseñar la aplicación fue el hecho de implementar una técnica de bloqueo en los computadores clientes; debido a la falta de información sobre este tema. El tiempo consumido para la investigación de este tema fue extenso y finalmente se definieron, básicamente, dos tipos de bloqueos como se definen en el capítulo uno.

El primero de ellos, la Autenticación de Windows se descartó por dos motivos principales:

- Obtener un estudio muy completo sobre la Arquitectura de Seguridad de Windows especialmente en lo que se refiere a token, paquetes de autenticación y las temas relacionados.



- Poseer amplios conocimientos sobre la programación en el lenguaje C++, debido a que la librería GINA interactúa directamente con algunos procesos de windows, se vuelve indispensable la programación en bajo nivel para el desarrollo de una librería personalizada GINA de Windows.

- **La Comunicación entre los Subsistemas.**

El siguiente paso a decidir fue la forma de realizar la comunicación entre los subsistemas Servidor y Cliente. Al principio no se tenía una idea concreta o un panorama claro de cómo realizar este módulo, para poder determinar el consumo de minutos del estudiante. En el transcurso de la investigación se determinó que la comunicación entre el Cliente y el Servidor tenía que ser: confiable, permanente y bidireccional. La plataforma de desarrollo .NET pone a disposición de los programadores dos tecnologías para el desarrollo de aplicaciones Cliente – Servidor en una red de área local.

- Sockets.
- Net Remoting.

Para este caso se optó por la primera opción, siendo su programación de mas bajo nivel que NetRemoting.

Teniendo en cuenta el número de máquinas que se van a conectar al sistema, es fácilmente predecible que se realizarán accesos de forma simultánea, por lo cual, la aplicación servidor ha de diseñarse de forma que permita accesos simultáneos. Es decir, será un servidor multithread.



4.2 DICCIONARIO DE CLASES

El diccionario de clases contiene información de las principales clases que se van a implementar en el sistema.

Las clases se definirán de en base a la siguiente plantilla:

Clase:	Nombre de la Clase
Descripción	Descipción relacionada con la clase

4.2.1 Subsistema Administración del Centro de Cómputo

Clase:	frmPrincipal
Descripción	Esta clase es el formulario principal del sistema y se deriva del librería Windows.System.Forms

Clase:	frmUsuarios
Descripción	Clase formulario que representa a la capa de interfaz para la gestión de usuarios.

Clase:	frmBuscarUsuario
Descripción	Formulario para permitir al usuario realizar una búsqueda de un estudiante o profesor.

Clase:	frmPrnSala
Descripción	Formulario utilizado para la gestión de salas.

Clase:	frmPrnComputador
Descripción	Clase formulario perteneciente a la capa de interfaz, para permitir la gestión de computadores o máquinas.



Clase:	frmPrnHorario
Descripción	Formulario de la capa interfaz que permite la gestión de los horarios.

Clase:	frmPeríodo
Descripción	Formulario utilizado para realizar el ingreso o modificación de un período.

Clase:	frmPrnReservación.
Descripción	Formulario para controlar la interfaz de la gestión de reservaciones.

Clase:	cls_ctrGestionUsuarios
Descripción	Clase encargada de realizar consultas con respecto a usuarios, como por ejemplo búsquedas de usuarios y para el control de usuarios.

Clase:	cls_ctrGestionSalas
Descripción	Clase encargada para el control de salas y el control de computadores.

Clase:	cls_ctrGestionHorarios
Descripción	Clase encargada de realizar el control de horarios y el control del período lectivo.

Clase:	cls_ctrGestionRsvnes
Descripción	Clase que se ocupa del control de reservaciones.

Clase:	cls_ctrUsuario
Descripción	Clase encargada de la validación de los datos del usuario.



Clase:	cls_ctrSala
Descripción	Clase que controla la validación de los datos de la sala.

Clase:	cls_ctrComputador
Descripción	Clase para validar los datos de un computador.

Clase:	cls_ctrHorario
Descripción	Clase que se ocupa de validar los datos de los horarios.

Clase:	cls_ctrReservacion
Descripción	Clase que realiza la validación de los datos de reservaciones.

Clase:	cls_accUsuario
Descripción	Clase que proporciona métodos para el acceso a la base de datos en lo que respecta a la gestión de usuarios.

Clase:	cls_accSala
Descripción	Clase que permite el acceso a la base de datos en lo que se refiere al mantenimiento de salas..

Clase:	cls_accComputador
Descripción	Clase que realiza el acceso a la base de datos para la administración de computadores.

Clase:	cls_accHorario
Descripción	Esta clase permite el acceso a la base de datos para el manejo de horarios.

Clase:	cls_accPeriodo
Descripción	Clase para realizar el acceso a la base de datos para el



	manejo del período.
--	---------------------

Clase:	cls_accReservación
Descripción	Clase para realizar el acceso a la base de datos para la administración de reservaciones.

Clase:	cs_Configuración
Descripción	Clase para establecer algunas configuraciones del sistema como por ejemplo la cadena de configuración para el acceso a la base de datos.

4.2.2 Subsistema Acceso al Sistema

Clase:	frmLogin
Descripción	Clase formulario para la identificación de un usuario.

Clase:	frmExploradorPCs
Descripción	Formulario tipo explorador de windows para visualizar las salas y los computadores con su respectivo estado (ocupado, disponible).

Clase:	cls_ctrGestionConsumos
Descripción	Clase para el control de consumo de los usuarios.

Clase:	cls_ctrConsumo
Descripción	Clase para realizar la validación de los consumos realizados por los usuarios.

Clase:	cls_accConsumo
Descripción	Clase que suministra de métodos para el acceso a la base de datos en lo que se refiere a consumos.



4.2.3 Subsistema Comunicación

Clase:	AsyncSocketCliente
Descripción	Clase que se encuentra en lado del servidor. Viene a ser como una instancia de un cliente conectado. Contiene métodos para la recepción de datos y para cerrar el socket; así como también contiene métodos especiales para generar eventos cuando llegan datos, cuando sucede un error y cuando el socket se cierra.

Clase:	SocketServidor
Descripción	Clase principal del servidor para manejar las conexiones entrantes. Contiene métodos para iniciar el socket servidor, crear registro de las conexiones entrantes o remover del registro las conexiones que se cierran, así como también métodos especiales para generar eventos de cuando una conexión cliente se cierra o se establece, eventos de cuando llegan datos o sucede algún error y eventos de cuando se inicia o para el socket servidor.

Clase:	SocketCliente
Descripción	Clase para realizar la conexión con el cliente. Tiene las funciones principales para establecer una conexión con el socket servidor.

4.2.4 Subsistema Cliente

Clase:	frmPrincipal
Descripción	Formulario principal de la aplicación cliente.



Clase:	frmPantallaBloq
Descripción	Formulario utilizado para cubrir toda la pantalla cuando se bloquea el computador.

Clase:	clsConfig
Descripción	Clase para guardar y obtener información, sobre la configuración de la aplicación cliente, desde el archivo de configuración.

Clase:	clsRegistro
Descripción	Clase para acceder al registro de windows.

Clase:	clsWin32API
Descripción	Clase que contiene las declaraciones de funciones para acceder a la librería SystemHook.

Clase:	SystemHook
Descripción	Librería con algunos métodos para realizar el bloqueo del escritorio y el teclado.



4.3 DISEÑO DEL SUBSISTEMA ADMINISTRACIÓN DEL CENTRO DE COMPUTO.

4.3.1 Arquitectura del Subsistema.

La arquitectura del subsistema está dividida en cuatro capas lógicas:

- **Capa Interfaz.**- Permite que los usuarios tengan acceso a la aplicación. Esta capa está conformada por formularios de windows y archivos de código que implementan el control de eventos para los distintos controles.
- **Capa Fachada del Negocio.**- Ofrece interfaces con la capa Interfaz para el control de usuarios, el control y exploración de salas, también para el control de horarios y reservaciones.
- **Capa Reglas del Negocio.**- Esta capa contiene la implementación de las reglas y la lógica empresariales. Las reglas de empresa realizan tareas tales como la validación de los datos de usuarios, salas, computadores, etc.
- **Capa de Acceso a Datos.**- Esta capa como su nombre lo indica proporciona métodos para el acceso a la base de datos a la capa de Negocio (Fachada y Reglas).

Además de las cuatro capas lógicas definidas, este subsistema contiene también funciones compartidas encapsuladas en un proyecto denominado Común. Este proyecto Común contiene conjunto de datos, conocidos como DataSet, que se utilizan para pasar información entre las distintas capas. Además contiene información sobre la configuración del subsistema.



4.3.2 Diagrama de Clases del Subsistema.

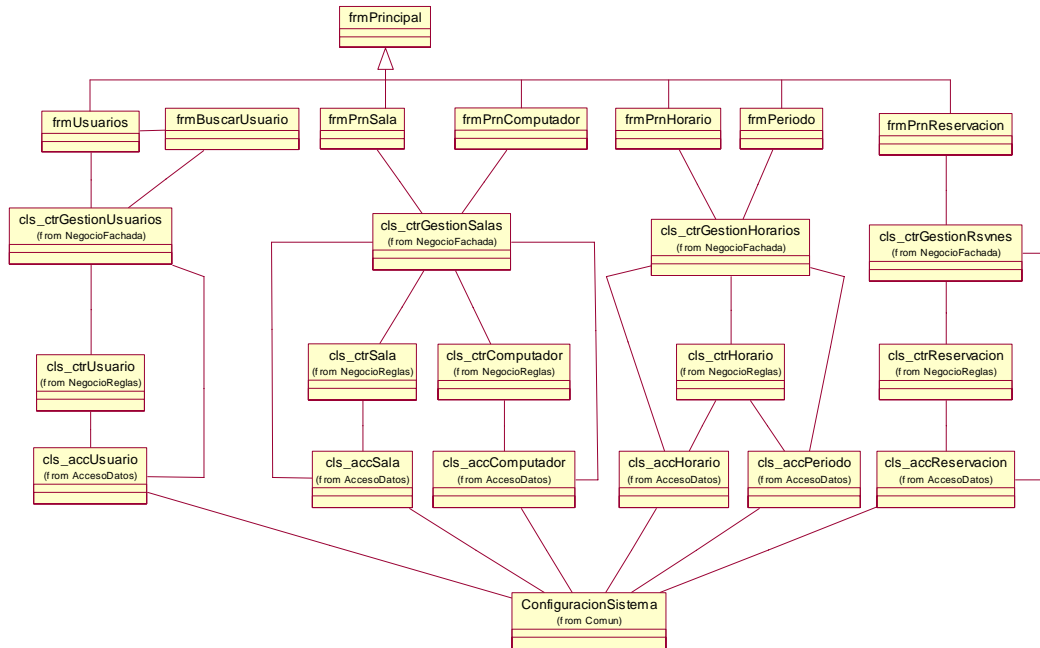


figura 4.1 Diagrama Clases del Subsistema Administración del Centro de Cómputo.



4.4 DISEÑO DEL SUBSISTEMA ACCESO AL SISTEMA.

4.4.1 Arquitectura del Subsistema.

La arquitectura del subsistema es igual a la definida en el subsistema anterior, es decir, consta de cuatro capas lógicas: Interfaz, Fachada del Negocio, Reglas del Negocio y Acceso a Datos.

4.4.2 Diagrama de Clases del Subsistema.

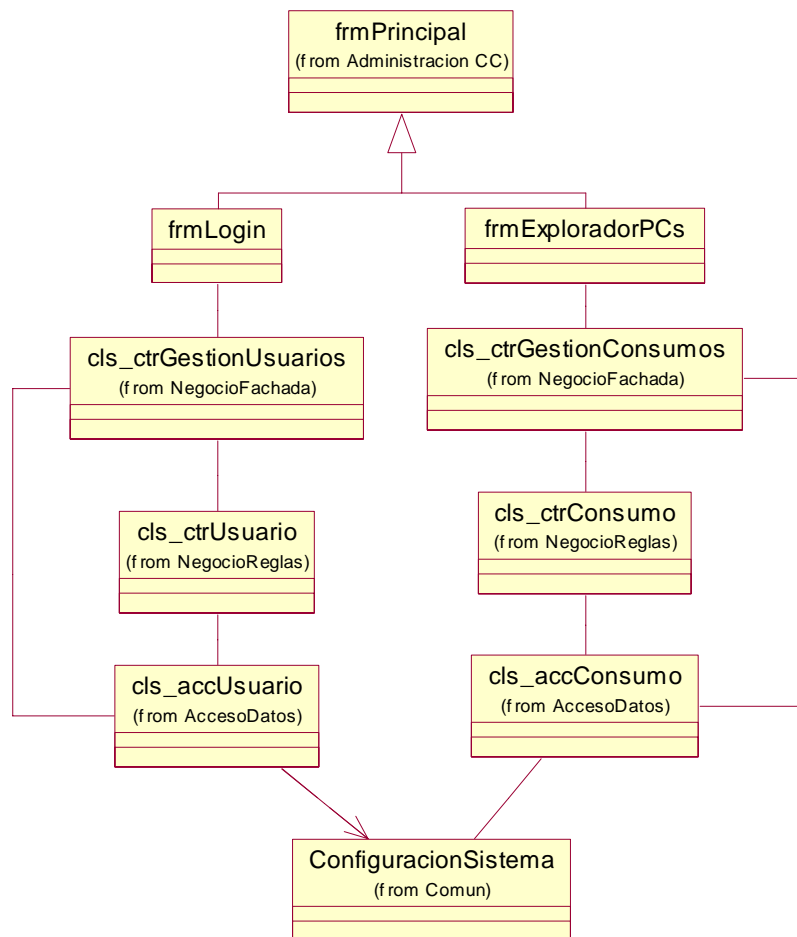


figura 4.2 Diagrama Clases del Subsistema Acceso al Sistema.



4.5 DISEÑO DEL SUBSISTEMA COMUNICACION.

4.5.1 Arquitectura del Subsistema.

El subsistema tiene una arquitectura cliente – servidor.

4.5.2 Diagrama de Clases del Subsistema.

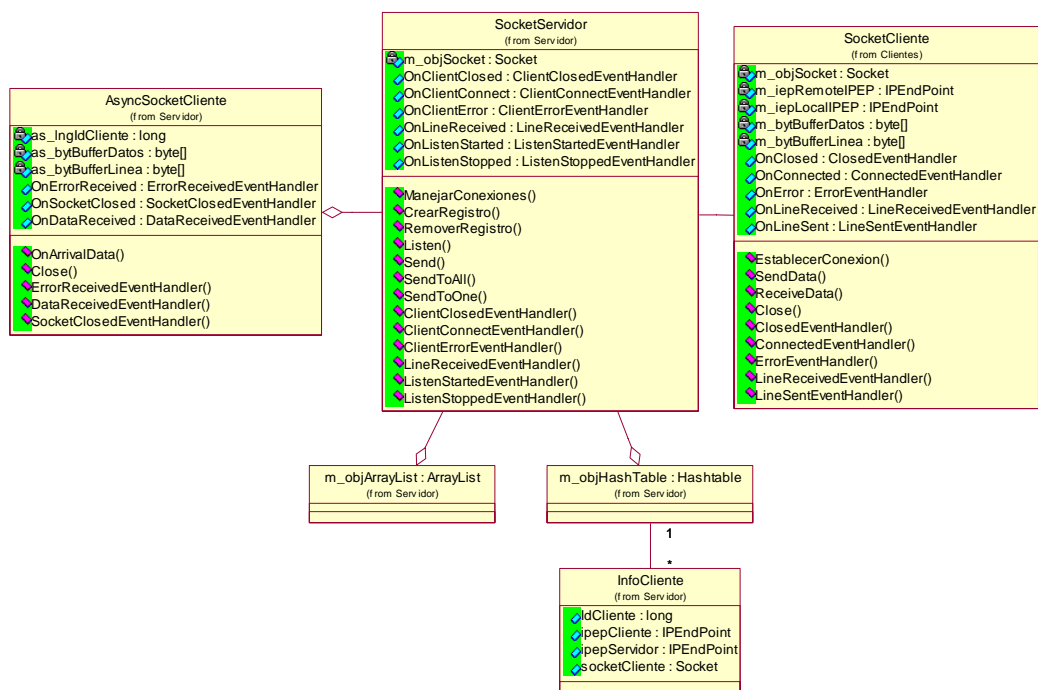


figura 4.3 Diagrama Clases del Subsistema Comunicación.



4.6 DISEÑO DEL SUBSISTEMA CLIENTE.

4.6.1 Arquitectura del Subsistema.

El subsistema no está dividido en capas lógicas. Este es el subsistema que se instala en el computador cliente.

4.6.2 Diagrama de Clases del Subsistema.

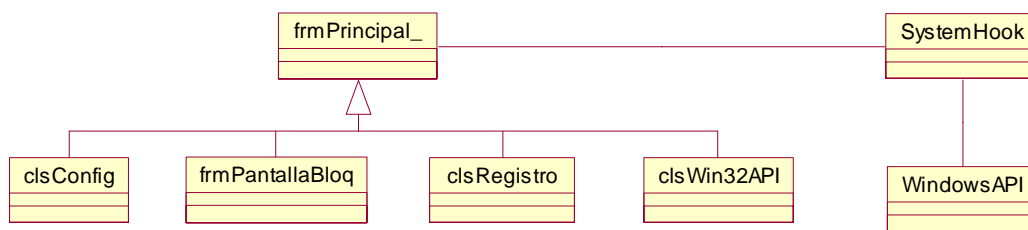


figura 3.4 Diagrama Clases del Subsistema Cliente.



4.7 DISEÑO DE LA BASE DE DATOS.

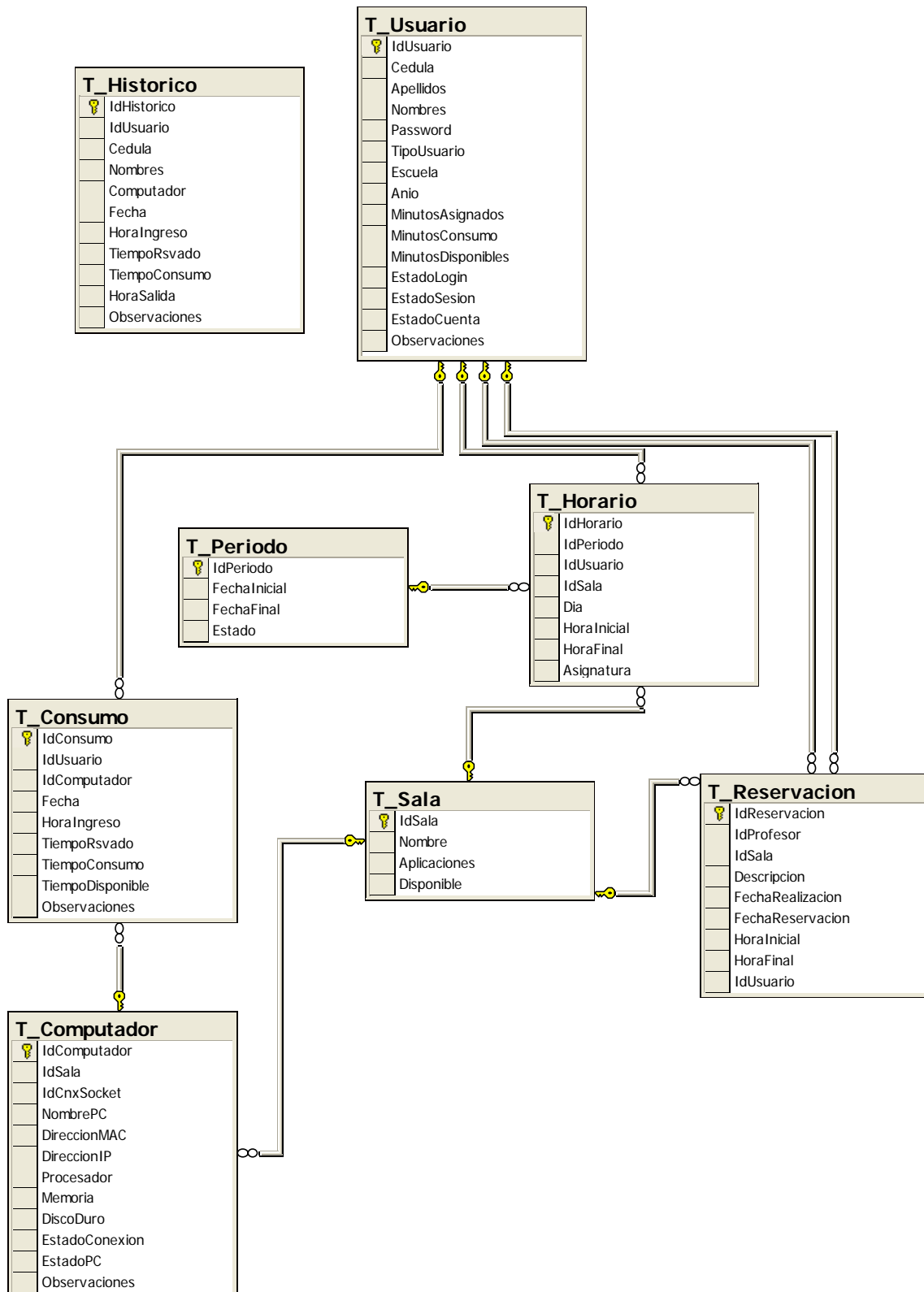


figura 4.5 Diagrama Clases del Subsistema Administración del Centro de Cómputo.

5. IMPLEMENTACION DEL SISTEMA



5 IMPLEMENTACION DEL SISTEMA

5.1 HERRAMIENTAS UTILIZADAS

5.1.1 Microsoft.NET

Microsoft.NET es el conjunto de nuevas tecnologías en las que Microsoft ha estado trabajando durante los últimos años con el objetivo de obtener una plataforma sencilla y potente para distribuir el software en forma de servicios que puedan ser suministrados remotamente y que puedan comunicarse y combinarse unos con otros de manera totalmente independiente de la plataforma, lenguaje de programación y modelo de componentes con los que hayan sido desarrollados. Ésta es la llamada plataforma .NET, y a los servicios antes comentados se les denomina servicios Web.

Para crear aplicaciones para la plataforma .NET, tanto servicios Web como aplicaciones tradicionales (aplicaciones de consola, aplicaciones de ventanas, servicios de Windows NT, etc.), Microsoft ha publicado el denominado kit de desarrollo de software conocido como .NET Framework SDK, que incluye las herramientas necesarias tanto para su desarrollo como para su distribución y ejecución y también ha publicado Visual Studio.NET, que permite hacer todo lo anterior desde una interfaz visual basada en ventanas.

El concepto de Microsoft.NET también incluye al conjunto de nuevas aplicaciones que Microsoft y terceros han desarrollado para ser utilizadas en la plataforma .NET. Entre ellas podemos destacar aplicaciones desarrolladas por Microsoft tales como Windows.NET, Visual Studio.NET, MSN.NET, Office.NET, y los nuevos servidores para empresas de Microsoft (SQL Server.NET, Exchange.NET, etc.).



5.1.1.1 .NET Framework

.NET Framework constituye la 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 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); 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 4.1* muestra un diagrama con la distribución de elementos dentro del entorno de .NET Framework.

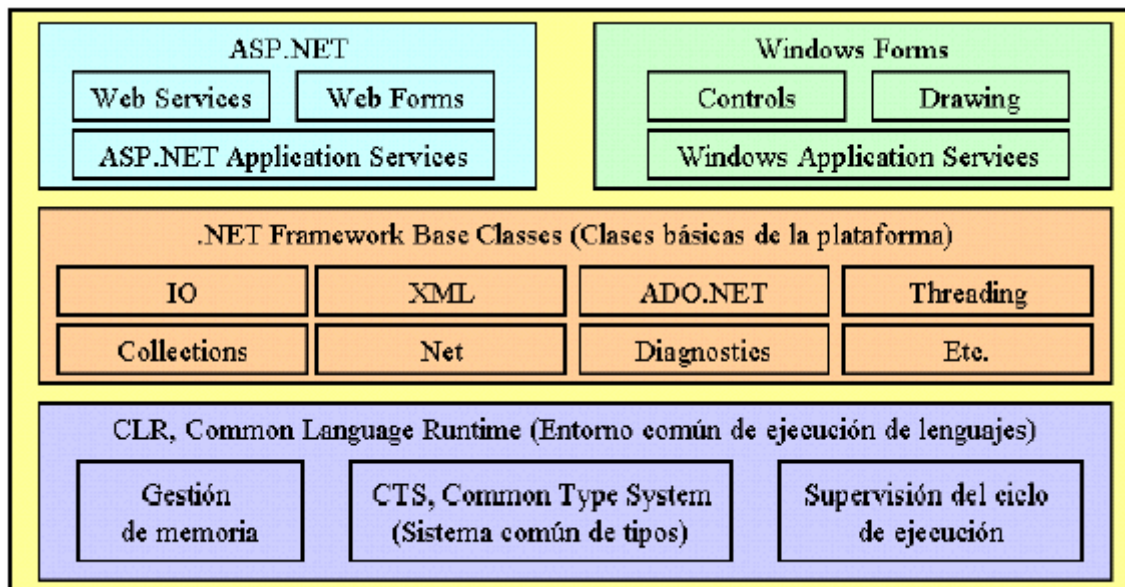


Figura 5.1 Esquema de componentes dentro de la plataforma .Net Framework



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

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, etc.

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, etc.

El .NET Framework no está orientado únicamente para la programación en Internet por lo que pone a disposición los denominados Windows Forms, la nueva generación de formularios, con características avanzadas y mejoradas.

El CLR, Common Language Runtime

El Entorno de Ejecución Común de Lenguajes o CLR (Common Language Runtime), representa el alma de .NET Framework y es el encargado de la ejecución del código de las aplicaciones.

A continuación se enumeran algunas de las características de este componente de la plataforma:

- Proporciona un desarrollo de aplicaciones más sencillo y rápido gracias a que gran parte de las funcionalidades que tradicionalmente debía de crear el programador, vienen implementadas en el entorno de ejecución.
- Administra el código en tiempo de ejecución, en todo lo referente a su carga, disposición en memoria, recuperación de memoria no utilizada a través de un recolector de memoria, etc.



- Implementa características de gestión a bajo nivel (administración de memoria, por ejemplo), que en ciertos lenguajes, eran labor del programador.
- Proporciona un sistema común de tipos para todos los lenguajes del entorno.
- Gestiona la seguridad del código que es ejecutado.
- Dispone de un diseño abierto a lenguajes y herramientas de desarrollo creadas por terceros fabricantes.
- Facilita enormemente la distribución e instalación de aplicaciones, ya que en teoría, es posible instalar una aplicación simplemente copiando los ficheros que la componen en uno de los directorios del equipo en el que se vaya a ejecutar, eliminando los temibles conflictos de versiones entre librerías, problema conocido también con el nombre de Infierno de las DLL.

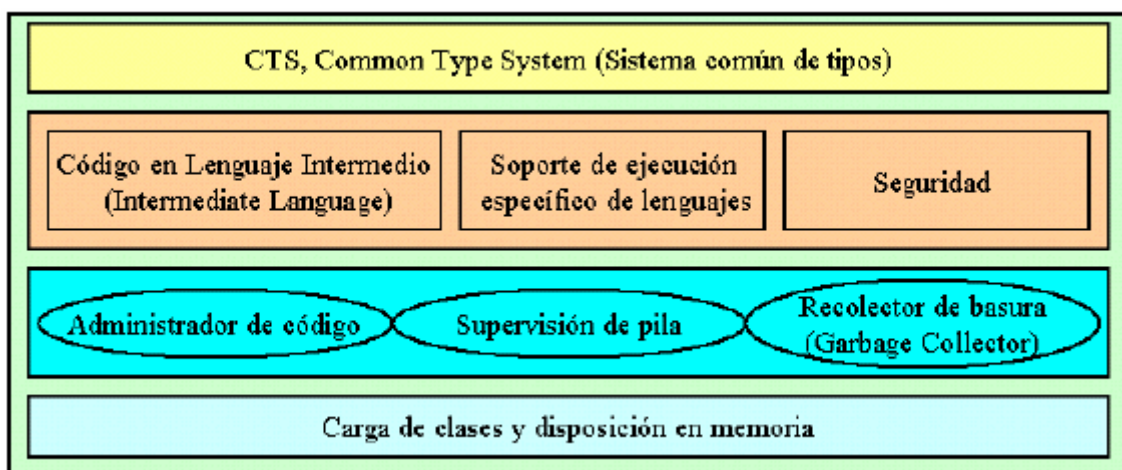


Figura 5.2 Esquema de elementos dentro del CLR.



El CTS, Common Type System

El Sistema Común de Tipos o CTS (Common Type System), es el mecanismo del CLR que permite definir el modo en que los tipos serán creados y manipulados por el entorno de ejecución de .NET Framework.

Entre las funcionalidades que comprende, podemos destacar la integración de código escrito en diferentes lenguajes; optimización del código en ejecución; un modelo de tipos orientado a objetos, que soporta múltiples lenguajes; y una serie de normas que aseguran la intercomunicación entre objetos.

El sistema común de tipos (CTS), como hemos indicado, permite definir o diseñar el modo en cómo el código de la aplicación será ejecutado, pero no se encarga directamente de su ejecución; dicho de otro modo, el CTS le dice al CLR cómo quiere que sea ejecutado el código.

Un ejemplo de las ventajas del CTS, consiste en que desde un lenguaje como VB.NET, podemos instanciar un objeto de una clase escrita en otro lenguaje como C#; y al hacer una llamada a uno de los métodos del objeto, no es necesario realizar conversiones de tipos en los parámetros del método, funcionando todo de forma transparente.

Al mencionar el sistema de tipos de la plataforma .NET, podemos pensar, que se trata sólo del conjunto de tipos de datos con que podemos declarar variables en nuestro código; sin embargo, el CTS, va mucho más allá y se extiende a cualquier elemento que pueda ser ejecutado dentro del entorno.

Por tal motivo, en el contexto de .NET Framework, un tipo se puede definir como una entidad de código ejecutada dentro del CLR; entendiendo por entidad de código, aquella a partir de la cual creamos una instancia y manejamos posteriormente en el programa como un objeto.

De todo lo anterior podemos obtener dos conclusiones:



- Todas las implementaciones de clases, interfaces, estructuras, etc., ya sean nativas de la plataforma o creadas por el programador, se pueden considerar tipos válidos de .NET.
- Todos los tipos que manipulamos dentro de .NET Framework son objetos.

Microsoft Intermediate Language (MSIL)

Todos los compiladores que generan código para la plataforma .NET no generan código máquina para CPUs x86 ni para ningún otro tipo de CPU concreta, sino que generan código escrito en el lenguaje intermedio conocido como Microsoft Intermediate Language (MSIL). El CLR da a las aplicaciones la sensación de que se están ejecutando sobre una máquina virtual, y precisamente MSIL es el código máquina de esa máquina virtual. Es decir, MSIL es el único código que es capaz de interpretar el CLR, y por tanto cuando se dice que un compilador genera código para la plataforma .NET lo que se está diciendo es que genera MSIL.

MSIL ha sido creado por Microsoft tras consultar a numerosos especialistas en la escritura de compiladores y lenguajes tanto del mundo académico como empresarial. Es un lenguaje de un nivel de abstracción mucho más alto que el de la mayoría de los códigos máquina de las CPUs existentes, e incluye instrucciones que permiten trabajar directamente con objetos (crearlos, destruirlos, inicializarlos, llamar a métodos virtuales, etc.), tablas y excepciones (lanzarlas, capturarlas y tratarlas)

El compilador de C# compila directamente el código fuente a MSIL, también Microsoft ha desarrollado nuevas versiones de sus lenguajes Visual Basic (Visual Basic.NET) y C++ (C++ con extensiones gestionadas) cuyos compiladores generan MSIL, y se ha desarrollado un intérprete de JScript (JScript.NET) que genera código MSIL. Ahora, también hay numerosos terceros que han anunciado estar realizando versiones para la plataforma .NET



de otros lenguajes como APL, CAML, Cobol, Eiffel, Fortran, Haskell, Java (J#), Mercury, ML, Mondrian, Oberon, Oz, Pascal, Perl, Python, RPG, Scheme y Smalltalk.

La principal ventaja del MSIL es que facilita la ejecución multiplataforma y la integración entre lenguajes al ser independiente de la CPU y proporcionar un formato común para el código máquina generado por todos los compiladores que generen código para .NET. Sin embargo, dado que las CPUs no pueden ejecutar directamente MSIL, antes de ejecutarlo habrá que convertirlo al código nativo de la CPU sobre la que se vaya a ejecutar. De esto se encarga un componente del CLR conocido como compilador JIT (Just-In-Time) o jitter que va convirtiendo dinámicamente el código MSIL a ejecutar en código nativo según sea necesario.

5.1.1.2 ADO.NET

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. 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.

ADO .NET se puede definir como:

- Un conjunto de interfaces, clases, estructuras y enumeraciones que permiten el acceso a datos desde la plataforma .NET de Microsoft.



- La evolución lógica del API ADO tradicional de Microsoft.
- Permite un modo de acceso desconectado a los datos, los cuales pueden provenir de múltiples fuentes de datos, de diferente arquitectura de almacenamiento.
- Soporta un completo modelo de programación y adaptación, basado en el estándar XML.

5.1.1.3 El Lenguaje de Programación C#

C# (leído en inglés “C Sharp” y en español “C Almohadilla”) es el nuevo lenguaje de propósito general diseñado por Microsoft para su plataforma .NET. Sus principales creadores son Scott Wiltamuth y Anders Hejlsberg, éste último también conocido por haber sido el diseñador del lenguaje Turbo Pascal y la herramienta RAD Delphi.

Aunque es posible escribir código para la plataforma .NET en muchos otros lenguajes, C# es el único que ha sido diseñado específicamente para ser utilizado en ella, por lo que programarla usando C# es mucho más sencillo e intuitivo que hacerlo con cualquiera de los otros lenguajes ya que C# carece de elementos heredados innecesarios en .NET. Por esta razón, se suele decir que C# es el lenguaje nativo de .NET

La sintaxis y estructuración de C# es muy similar a la C++, ya que la intención de Microsoft con C# es facilitar la migración de códigos escritos en estos lenguajes a C# y facilitar su aprendizaje a los desarrolladores habituados a ellos. Sin embargo, su sencillez y el alto nivel de productividad son equiparables a los de Visual Basic.



Un lenguaje que hubiese sido ideal utilizar para estos menesteres es Java, pero debido a problemas con la empresa creadora del mismo -Sun-, Microsoft ha tenido que desarrollar un nuevo lenguaje que añadiese a las ya probadas virtudes de Java las modificaciones que Microsoft tenía pensado añadirle para mejorarlo aún más y hacerlo un lenguaje orientado al desarrollo de componentes.

En resumen, C# es un lenguaje de programación que toma las mejores características de lenguajes preexistentes como Visual Basic, Java o C++ y las combina en uno solo. El hecho de ser relativamente reciente no implica que sea inmaduro, pues Microsoft ha escrito la mayor parte de la BCL usándolo, por lo que su compilador es el más depurado y optimizado de los incluidos en el .NET Framework SDK.

Características de C#

Con la idea de que los programadores más experimentados puedan obtener una visión general del lenguaje, a continuación se recoge de manera resumida las principales características de C#. Algunas de las características aquí señaladas no son exactamente propias del lenguaje sino de la plataforma .NET en general. Sin embargo, también se comentan aquí también en tanto que tienen repercusión directa en el lenguaje, aunque se indicará explícitamente cuáles son este tipo de características cada vez que se toquen:

- **Sencillez:** C# elimina muchos elementos que otros lenguajes incluyen y que son innecesarios en .NET. Por ejemplo:
 - El código escrito en C# es autocontenido, lo que significa que no necesita de ficheros adicionales al propio fuente tales como ficheros de cabecera o ficheros IDL
 - El tamaño de los tipos de datos básicos es fijo e independiente del compilador, sistema operativo o máquina para quienes se



compile (no como en C++), lo que facilita la portabilidad del código.

- No se incluyen elementos poco útiles de lenguajes como C++ tales como macros, herencia múltiple o la necesidad de un operador diferente del punto (.) acceder a miembros de espacios de nombres (::)
- **Modernidad:** C# incorpora en el propio lenguaje elementos que a lo largo de los años ha ido demostrándose son muy útiles para el desarrollo de aplicaciones y que en otros lenguajes como Java o C++ hay que simular, como un tipo básico decimal que permita realizar operaciones de alta precisión con reales de 128 bits (muy útil en el mundo financiero), la inclusión de una instrucción foreach que permita recorrer colecciones con facilidad y es ampliable a tipos definidos por el usuario, la inclusión de un tipo básico string para representar cadenas o la distinción de un tipo bool específico para representar valores lógicos.
- **Orientación a objetos:** Como todo lenguaje de programación de propósito general actual, C# es un lenguaje orientado a objetos, aunque eso es más bien una característica del CTS que de C#. Una diferencia de este enfoque orientado a objetos respecto al de otros lenguajes como C++ es que el de C# es más puro en tanto que no admiten ni funciones ni variables globales sino que todo el código y datos han de definirse dentro de definiciones de tipos de datos, lo que reduce problemas por conflictos de nombres y facilita la legibilidad del código.

C# soporta todas las características propias del paradigma de programación orientada a objetos: encapsulación, herencia y polimorfismo.



En lo referente a la encapsulación es importante señalar que aparte de los típicos modificadores `public`, `private` y `protected`, C# añade un cuarto modificador llamado `internal`, que puede combinarse con `protected` e indica que al elemento a cuya definición precede sólo puede accederse desde su mismo ensamblado.

Respecto a la herencia -a diferencia de C++ y al igual que Java- C# sólo admite herencia simple de clases ya que la múltiple provoca más quebraderos de cabeza que facilidades y en la mayoría de los casos su utilidad puede ser simulada con facilidad mediante herencia múltiple de interfaces. De todos modos, esto vuelve a ser más bien una característica propia del CTS que de C#.

Por otro lado y a diferencia de Java, en C# se ha optado por hacer que todos los métodos sean por defecto sellados y que los redefinibles hayan de marcarse con el modificador `virtual` (como en C++), lo que permite evitar errores derivados de redefiniciones accidentales. Además, un efecto secundario de esto es que las llamadas a los métodos serán más eficientes por defecto al no tenerse que buscar en la tabla de funciones virtuales la implementación de los mismos a la que se ha de llamar. Otro efecto secundario es que permite que las llamadas a los métodos virtuales se puedan hacer más eficientemente al contribuir a que el tamaño de dicha tabla se reduzca.

- **Orientación a componentes:** La propia sintaxis de C# incluye elementos propios del diseño de componentes que otros lenguajes tienen que simular mediante construcciones más o menos complejas. Es decir, la sintaxis de C# permite definir cómodamente propiedades (similares a campos de acceso controlado), eventos (asociación controlada de funciones de respuesta a notificaciones) o atributos (información sobre un tipo o sus miembros)



- **Gestión automática de memoria:** Como ya se comentó, todo lenguaje de .NET tiene a su disposición el recolector de basura del CLR. Esto tiene el efecto en el lenguaje de que no es necesario incluir instrucciones de destrucción de objetos. Sin embargo, dado que la destrucción de los objetos a través del recolector de basura es indeterminista y sólo se realiza cuando éste se active –ya sea por falta de memoria, finalización de la aplicación o solicitud explícita en el fuente-, C# también proporciona un mecanismo de liberación de recursos determinista a través de la instrucción `using`.

- **Seguridad de tipos:** C# incluye mecanismos que permiten asegurar que los accesos a tipos de datos siempre se realicen correctamente, lo que permite evita que se produzcan errores difíciles de detectar por acceso a memoria no perteneciente a ningún objeto y es especialmente necesario en un entorno gestionado por un recolector de basura. Para ello se toman medidas del tipo:
 - Sólo se admiten conversiones entre tipos compatibles. Esto es, entre un tipo y antecesoros suyos, entre tipos para los que explícitamente se haya definido un operador de conversión, y entre un tipo y un tipo hijo suyo del que un objeto del primero almacenase una referencia del segundo (downcasting). Obviamente, lo último sólo puede comprobarlo en tiempo de ejecución el CLR y no el compilador, por lo que en realidad el CLR y el compilador colaboran para asegurar la corrección de las conversiones.

 - No se pueden usar variables no inicializadas. El compilador da a los campos un valor por defecto consistente en ponerlos a cero y controla mediante análisis del flujo de control del fuente que no se lea ninguna variable local sin que se le haya asignado previamente algún valor.



- Se comprueba que todo acceso a los elementos de una tabla se realice con índices que se encuentren dentro del rango de la misma.
- Se puede controlar la producción de desbordamientos en operaciones aritméticas, informándose de ello con una excepción cuando ocurra. Sin embargo, para conseguirse un mayor rendimiento en la aritmética estas comprobaciones no se hacen por defecto al operar con variables sino sólo con constantes (se pueden detectar en tiempo de compilación)
- A diferencia de Java, C# incluye delegados, que son similares a los punteros a funciones de C++ pero siguen un enfoque orientado a objetos, pueden almacenar referencias a varios métodos simultáneamente, y se comprueba que los métodos a los que apunten tengan parámetros y valor de retorno del tipo indicado al definirlos.
- Pueden definirse métodos que admitan un número indefinido de parámetros de un cierto tipo, y a diferencia lenguajes como C/C++, en C# siempre se comprueba que los valores que se les pasen en cada llamada sean de los tipos apropiados.
- **Instrucciones seguras:** Para evitar errores muy comunes, en C# se han impuesto una serie de restricciones en el uso de las instrucciones de control más comunes. Por ejemplo, la guarda de toda condición ha de ser una expresión condicional y no aritmética, con lo que se evitan errores por confusión del operador de igualdad (==) con el de asignación (=); y todo caso de un switch ha de terminar en un break o goto que indique cuál es la siguiente acción a realizar, lo que evita la ejecución accidental de casos y facilita su reordenación.
- **Sistema de tipos unificado:** A diferencia de C++, en C# todos los tipos de datos que se definan siempre derivarán, aunque sea de manera



implícita, de una clase base común llamada `System.Object`, por lo que dispondrán de todos los miembros definidos en ésta clase (es decir, serán “objetos”)

A diferencia de Java, en C# esto también es aplicable a los tipos de datos básicos. Además, para conseguir que ello no tenga una repercusión negativa en su nivel de rendimiento, se ha incluido un mecanismo transparente de boxing y unboxing con el que se consigue que sólo sean tratados como objetos cuando la situación lo requiera, y mientras tanto puede aplicárseles optimizaciones específicas.

El hecho de que todos los tipos del lenguaje deriven de una clase común facilita enormemente el diseño de colecciones genéricas que puedan almacenar objetos de cualquier tipo.

- **Extensibilidad de tipos básicos:** C# permite definir, a través de estructuras, tipos de datos para los que se apliquen las mismas optimizaciones que para los tipos de datos básicos. Es decir, que se puedan almacenar directamente en pila (luego su creación, destrucción y acceso serán más rápidos) y se asignen por valor y no por referencia. Para conseguir que lo último no tenga efectos negativos al pasar estructuras como parámetros de métodos, se da la posibilidad de pasar referencias a pila a través del modificador de parámetro `ref`.
- **Extensibilidad de operadores:** Para facilitar la legibilidad del código y conseguir que los nuevos tipos de datos básicos que se definan a través de las estructuras estén al mismo nivel que los básicos predefinidos en el lenguaje, al igual que C++ y a diferencia de Java, C# permite redefinir el significado de la mayoría de los operadores -incluidos los de conversión, tanto para conversiones implícitas como explícitas- cuando se apliquen a diferentes tipos de objetos.



Las redefiniciones de operadores se hacen de manera inteligente, de modo que a partir de una única definición de los operadores ++ y -- el compilador puede deducir automáticamente como ejecutarlos de manera prefijas y postifja; y definiendo operadores simples (como +), el compilador deduce cómo aplicar su versión de asignación compuesta (+=) Además, para asegurar la consistencia, el compilador vigila que los operadores con opuesto siempre se redefinan por parejas (por ejemplo, si se redefine ==, también hay que redefinir !=)

También se da la posibilidad, a través del concepto de indizador, de redefinir el significado del operador [] para los tipos de dato definidos por el usuario, con lo que se consigue que se pueda acceder al mismo como si fuese una tabla. Esto es muy útil para trabajar con tipos que actúen como colecciones de objetos.

- **Extensibilidad de modificadores:** C# ofrece, a través del concepto de atributos, la posibilidad de añadir a los metadatos del módulo resultante de la compilación de cualquier fuente información adicional a la generada por el compilador que luego podrá ser consultada en tiempo ejecución a través de la librería de reflexión de .NET . Esto, que más bien es una característica propia de la plataforma .NET y no de C#, puede usarse como un mecanismo para definir nuevos modificadores.
- **Versionable:** C# incluye una política de versionado que permite crear nuevas versiones de tipos sin temor a que la introducción de nuevos miembros provoquen errores difíciles de detectar en tipos hijos previamente desarrollados y ya extendidos con miembros de igual nombre a los recién introducidos.

Si una clase introduce un nuevo método cuyas redefiniciones deban seguir la regla de llamar a la versión de su padre en algún punto de su código, difícilmente seguirían esta regla miembros de su misma signatura definidos en clases hijas previamente a la definición del mismo



en la clase padre; o si introduce un nuevo campo con el mismo nombre que algún método de una clase hija, la clase hija dejará de funcionar. Para evitar que esto ocurra, en C# se toman dos medidas:

- Se obliga a que toda redefinición deba incluir el modificador `override`, con lo que la versión de la clase hija nunca sería considerada como una redefinición de la versión de miembro en la clase padre ya que no incluiría `override`. Para evitar que por accidente un programador incluya este modificador, sólo se permite incluirlo en miembros que tengan la misma signatura que miembros marcados como redefinibles mediante el modificador `virtual`. Así además se evita el error tan frecuente en Java de creerse haber redefinido un miembro, pues si el miembro con `override` no existe en la clase padre se producirá un error de compilación.
- Si no se considera redefinición, entonces se considera que lo que se desea es ocultar el método de la clase padre, de modo que para la clase hija sea como si nunca hubiese existido. El compilador avisará de esta decisión a través de un mensaje de aviso que puede suprimirse incluyendo el modificador `new` en la definición del miembro en la clase hija para así indicarle explícitamente la intención de ocultación.
- **Eficiente:** En principio, en C# todo el código incluye numerosas restricciones para asegurar su seguridad y no permite el uso de punteros. Sin embargo, y a diferencia de Java, en C# es posible saltarse dichas restricciones manipulando objetos a través de punteros. Para ello basta marcar regiones de código como inseguras (modificador `unsafe`) y podrán usarse en ellas punteros de forma similar a cómo se hace en C++, lo que puede resultar vital para situaciones donde se necesite una eficiencia y velocidad procesamiento muy grandes.



- **Compatible:** Para facilitar la migración de programadores, C# no sólo mantiene una sintaxis muy similar a C, C++ o Java que permite incluir directamente en código escrito en C# fragmentos de código escrito en estos lenguajes, sino que el CLR también ofrece, a través de los llamados Platform Invocation Services (PInvoke), la posibilidad de acceder a código nativo escrito como funciones sueltas no orientadas a objetos tales como las DLLs de la API Win32. Nótese que la capacidad de usar punteros en código inseguro permite que se pueda acceder con facilidad a este tipo de funciones, ya que éstas muchas veces esperan recibir o devuelven punteros.

5.1.2 Otras Herramientas Utilizadas

5.1.2.1 Rose

Herramienta CASE que facilita el análisis y el diseño de aplicaciones. Permite crear diagramas que modelen un sistema desde distintos puntos de vista, facilitando el desarrollo de sistemas.



5.2 DIAGRAMAS DE IMPLEMENTACION

5.2.1 Diagrama de Componentes

Son uno de los dos tipos de diagramas que aparecen cuando se modelan los aspectos físicos de los sistemas orientados a objetos. Un diagrama de componentes presenta la organización y las dependencias entre un conjunto de componentes y se utiliza para modelar la vista de implementación estática de un sistema. Esto implica modelar los elementos físicos que residen en un nodo, tales como ejecutables y bibliotecas.

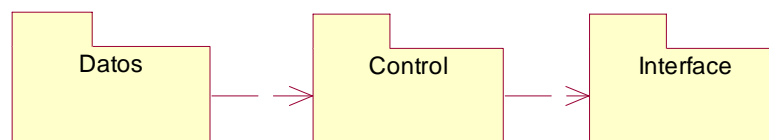


Figura 5.3 Jerarquía de Componentes.

- **Componentes de Datos**

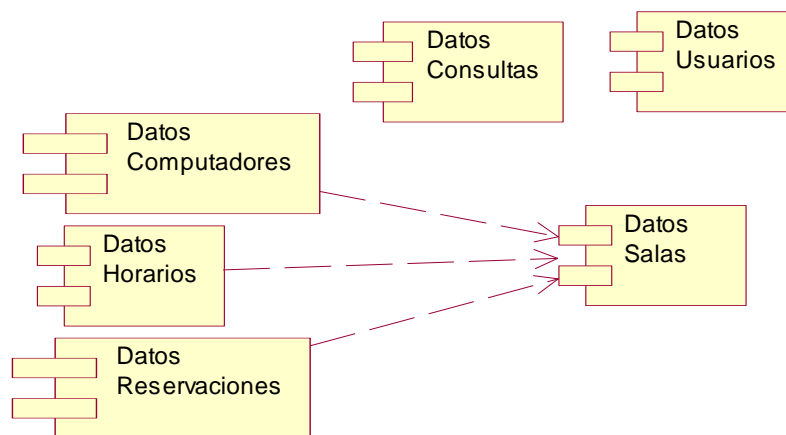


Figura 5.4 Componentes de Datos.



- **Componentes de Control**

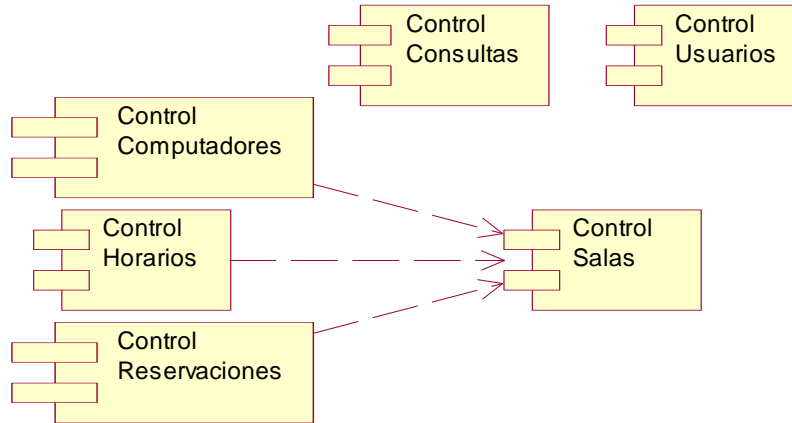


Figura 5.5 Componentes de Control.

- **Componentes de Interface**

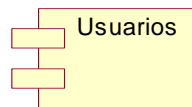


Figura 5.6 Componentes de Interface.

5.2.2 Diagrama de Despliegue

Los diagramas de despliegue son uno de los dos tipos de diagramas que aparecen cuando se modelan los aspectos físicos de los sistemas orientados a objetos. Un diagrama de despliegue muestra la configuración de los nodos que participan en la ejecución y de los componentes que residen en ellos.

Los diagramas de despliegue muestran el hardware físico y las redes utilizadas por una aplicación. Una plataforma o nodo puede incluir clases o componentes. Las líneas conectan plataformas para indicar enlaces de red utilizadas por el sistema.



La aplicación creada para este proyecto es un sistema servidor instalado en un servidor que puede o no incluir la base de datos. Por otra parte el subsistema cliente se encontrará instalado en una estación conectado a la red local.

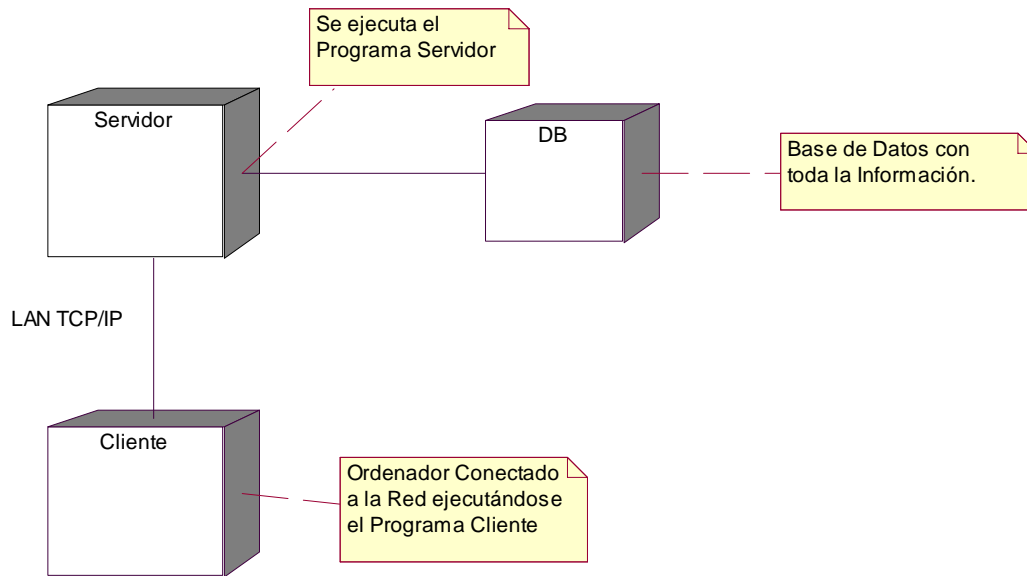


Figura 5.7 Diagrama de Despliegue.

6. MANUAL DEL USUARIO



6 MANUAL DEL USUARIO

6.1 SISTEMA DE SEGURIDAD Y CONTROL DE ACCESO (SERVIDOR).

BIENVENIDA

Sistema de Seguridad y Control de Acceso está diseñado para controlar el acceso al centro de cómputo por parte de los estudiantes y profesores



CARACTERISTICAS

- Subsistema Servidor.
 - Monitoreo de los computadores conectados en la red.
 - Medición y control del tiempo consumido por cada usuario.
 - Bloqueo y Desbloqueo de estaciones de trabajo dependiendo de, si el usuario está o no registrado.
 - Habilitar / Deshabilitar salas si un profesor tiene clases.
 - Reportes de consumo por estaciones, usuario y fecha.
 - Explorador de Estaciones de Trabajo.
 - Gestión de Salas del Centro de Cómputo.
 - Gestión de Estaciones de Trabajo.
 - Gestión de Horarios y Reservaciones.
 - Gestión de Usuarios.
 - Reportes de Horarios, Reservaciones y Usuarios.
 - Migrar datos de los estudiantes desde la base de datos central de la Universidad hacia la base de datos del sistema.
 - Migrar datos de computadores desde un archivo de Excel según un formato establecido (Orden de las columnas) a la base de datos del sistema.

- Subsistema Cliente.
 - Cierre de la sesión por el propio usuario cuando éste ya no desea seguir haciendo uso del computador.
 - Módulo de seguridad para el bloqueo de restricciones a nivel del Registro de Windows. Ello permite bloquear ciertas partes de la plataforma Windows para minimizar desconfiguraciones por parte de los usuarios que ingresan al Centro de Cómputo.
 - Conexión automática al Subsistema Servidor cuando se produce una desconexión por diferentes causas como desconexión del cable de red.



REQUERIMIENTOS DEL SISTEMA

Requerimientos mínimos para el Subsistema Servidor.

- Se requiere disponer de cualquier versión del Sistema Operativo Windows (NT/2000/XP/2003).
- Utilización de tarjeta de red de 10/100 mbps (mega bits por segundo) o superior para un mejor desempeño del sistema.
- Procesador Intel Pentium 4 1.5 Ghz o compatible, Memoria de 256MB. Recomendado para soportar un aproximado de 60 a 70 conexiones.

Requerimientos mínimos para el Subsistema Cliente.

- Se requiere disponer de cualquier versión del Sistema Operativo Windows (NT/2000/XP/2003).
- Utilización de tarjeta de red de 10/100 mbps (mega bits por segundo) o superior para un mejor desempeño del sistema.
- Procesador Intel Pentium III 900 Mhz, Memoria de 128MB.



6.2 INICIANDO CON EL SISTEMA

INSTALACION DE LA APLICACIÓN SERVIDOR.

- a) Antes de la Instalación revise los requerimientos del sistema.
- b) Instalación del MSDE

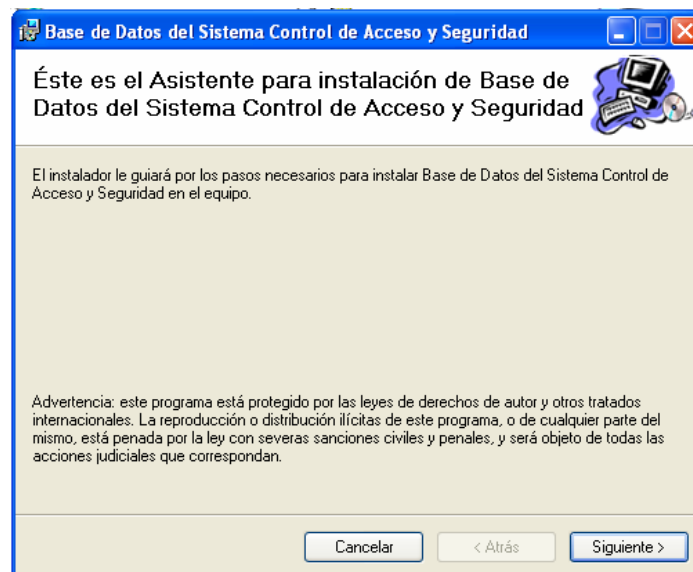
Este es el motor de la Base de Datos SQL Server y tiene que estar instalado antes de instalar la Base de Datos del Sistema de Seguridad y Control de Acceso. Esta aplicación se encuentra dentro de la carpeta MSDE (Microsoft Data Engine).

- c) Instalación de la Base de Datos.

La base de datos del Sistema de Seguridad y Control de Acceso se la puede instalar en el mismo computador donde se instalará la aplicación Servidor o en otro computador.

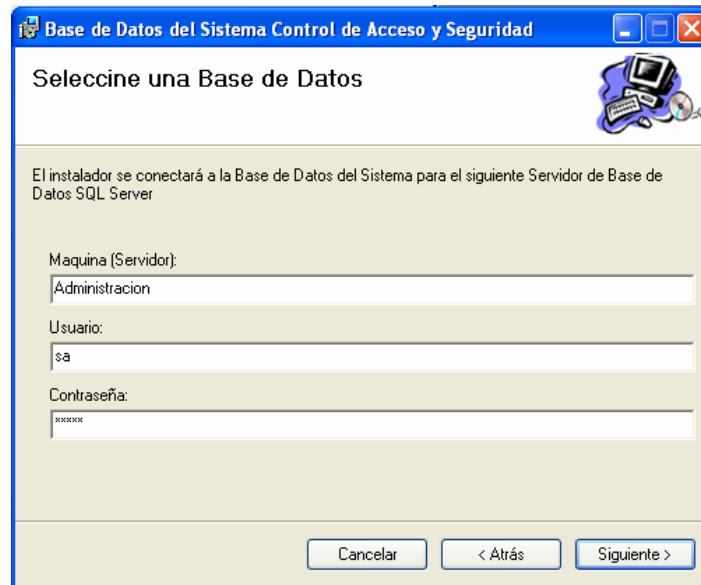
Los pasos a seguir para la instalación son los siguientes:

- En la carpeta “Base de Datos”, del CD de Instalación, hacer doble clic en el ejecutable “DataBase Sistema Control y Seguridad.exe”.
- Luego presione Siguiente.

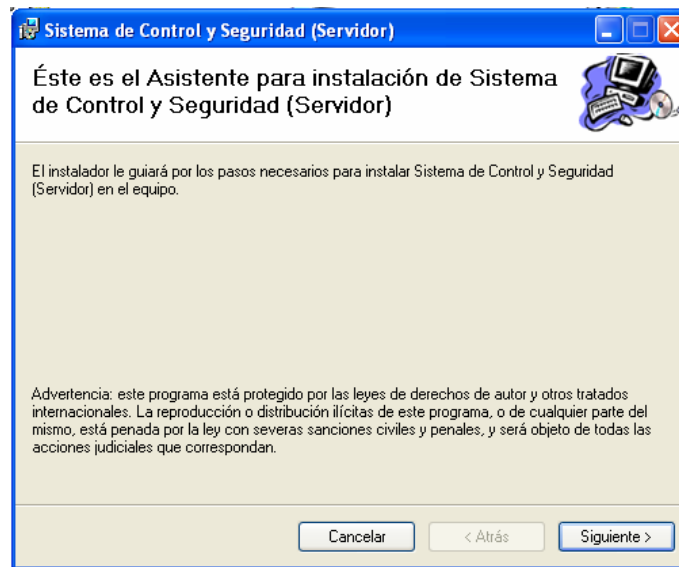




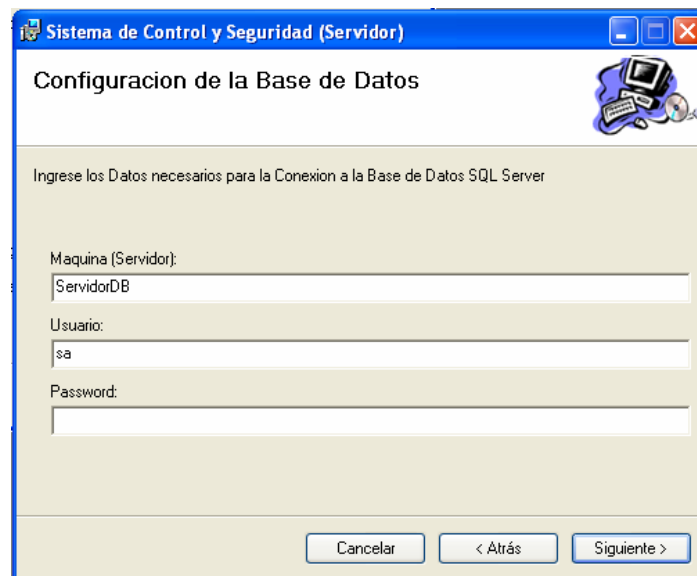
- En el siguiente cuadro de diálogo se debe ingresar el Nombre del Computador en el que se está instalando la Base de Datos, el nombre del Usuario con el que se conectará a la base de datos conjuntamente con la Contraseña y finalice presionando Siguiente.



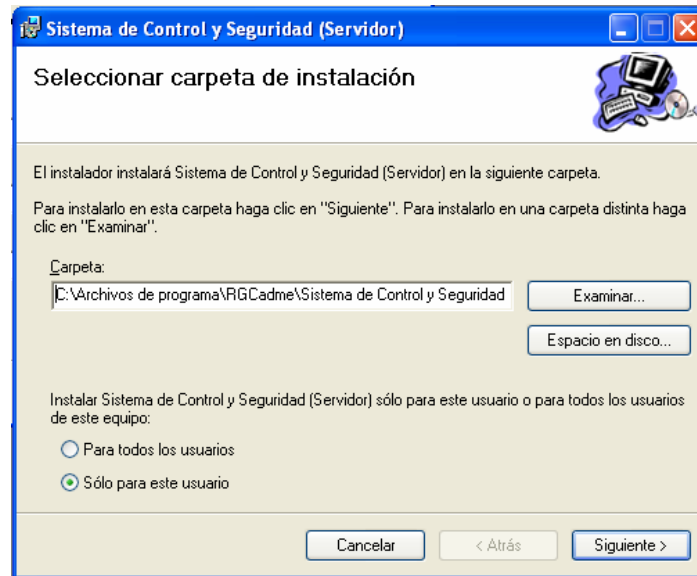
- En el transcurso de la instalación si los datos ingresados anteriormente son incorrectos y no se consigue una conexión a la Base de Datos, la instalación se cancelará. Finalmente presione Cerrar para terminar con la instalación.
- d) Instalación de la Aplicación Sistema de Seguridad y Control de Acceso.
Para instalar el Sistema de Seguridad y Control de Acceso se tiene que seguir los siguientes pasos:
- En la carpeta “Sistema de Control y Seguridad (Servidor)”, hacer doble clic en el archivo ejecutable.



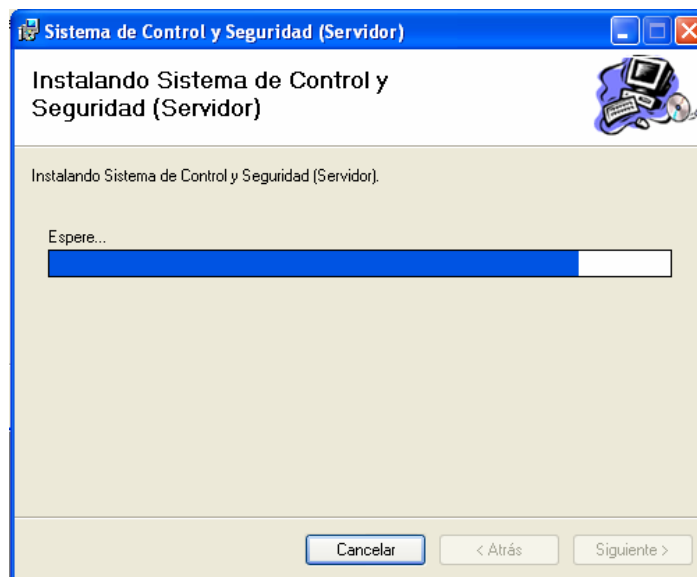
- Presionar Siguiente para continuar.



- Se tiene que ingresar los datos necesarios para realizar la conexión a la base de datos del sistema. Primero se ingresa el nombre de la máquina donde se instaló se la base de datos, luego el nombre de usuario y finalmente la contraseña. Termine presionando siguiente.



- Luego se tiene que ingresar la carpeta en donde se instalará el Sistema y presione Siguiente.



- Presione Cerrar para terminar con la instalación

e) Instalación de la Aplicación Migrar Datos de Estudiantes.

Para instalar el Sistema de Seguridad y Control de Acceso se tiene que seguir los siguientes pasos:

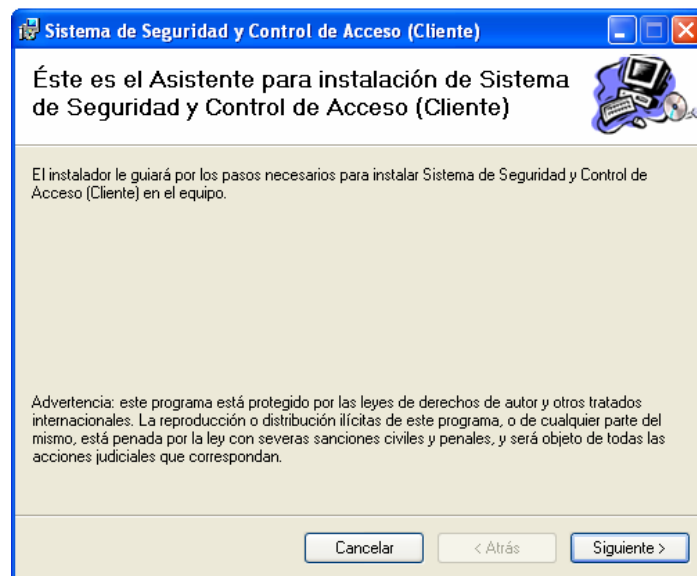
- En la carpeta "Migrar Datos de Estudiantes", hacer doble clic en el archivo ejecutable.
- Presionar Siguiente para continuar.



- Se tiene que ingresar los datos necesarios para realizar la conexión a la base de datos del sistema. Primero se ingresa el nombre de la máquina donde se instaló se la base de datos, luego el nombre de usuario y finalmente la contraseña. Termine presionando siguiente.
- Luego se tiene que ingresar el Nombre de la base de datos del Sistema de Matricula Diferenciada.
- Luego se tiene que ingresar la carpeta en donde se instalará esta Aplicación y presione Siguiente.
- Presione Cerrar para terminar con la instalación.

INSTALACION DE LA APLICACIÓN CLIENTE.

- En la carpeta “Sistema de Seguridad y Control de Acceso (Cliente)”, hacer doble clic en el archivo ejecutable.



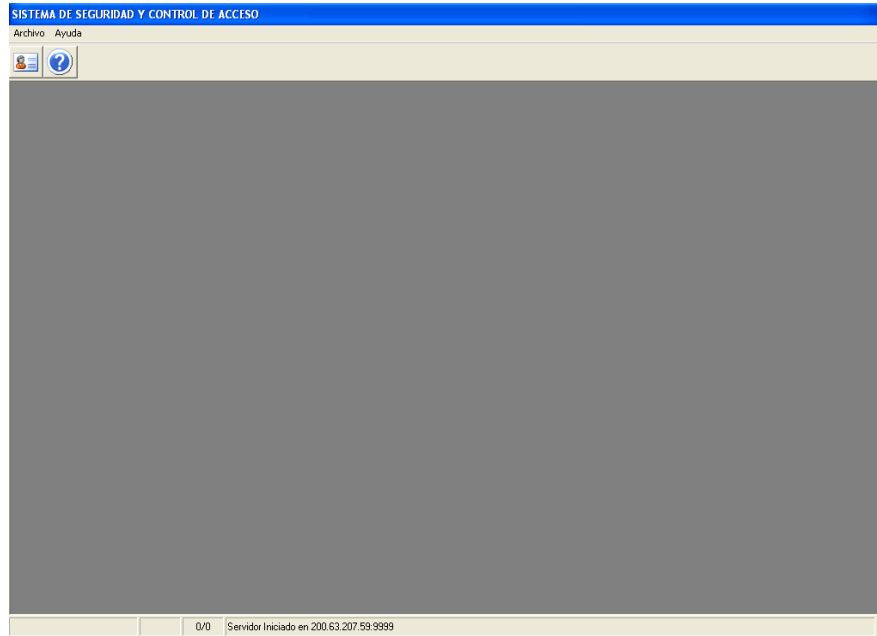
- Presionar Siguiente para continuar.
- Se tiene que ingresar los datos necesarios para realizar la conexión con la Aplicación Servidor; estos datos son: el número de puerto (el número predeterminado es 9999) y la dirección IP del Servidor.
- Para terminar con la instalación hacer clic en el botón cerrar.



INTRODUCCION

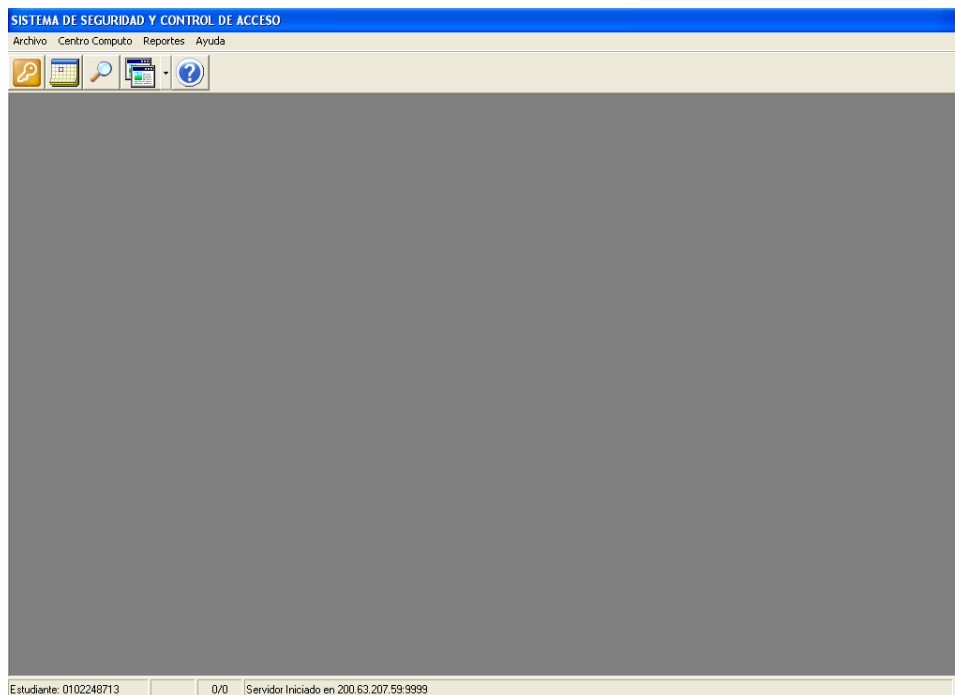
Subsistema Servidor

El Subsistema servidor es fácil de usar, la ventana principal de la aplicación es presentada a continuación.



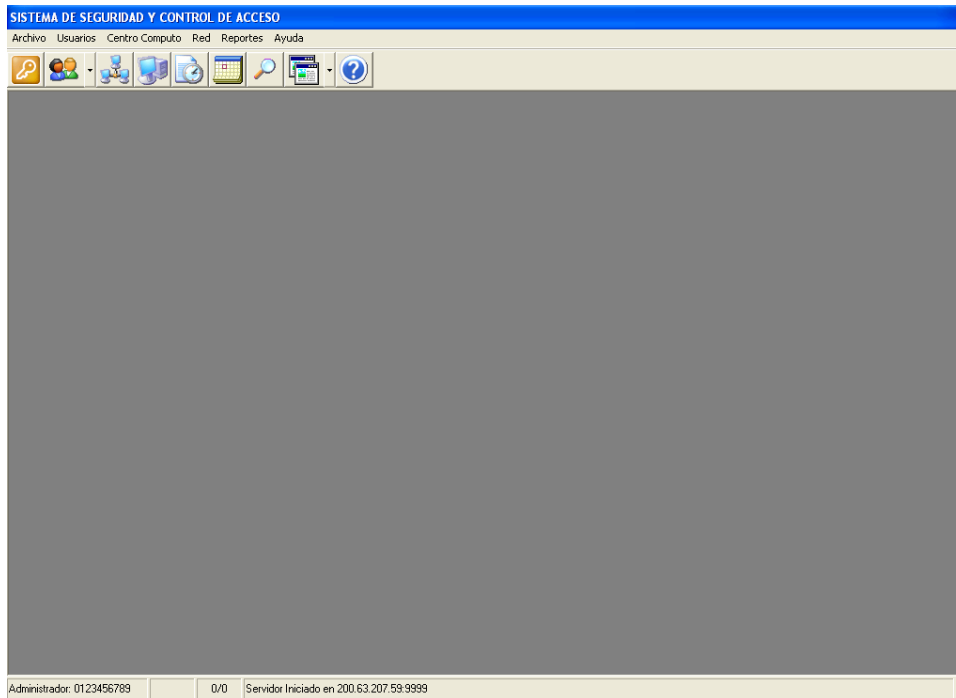
Esta ventana es presentada cuando ningún usuario ha ingresado al sistema servidor.

Cuando un usuario estudiante ó profesor se registra se presenta la siguiente ventana:





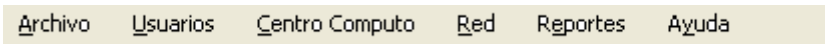
Cuando un usuario Administrador se registra se presenta la siguiente pantalla:





DESCRIPCION DE LA BARRA DE MENUS

La barra de menús posee las siguientes opciones como se observa en la figura:



El menú [Archivo] incluye las siguientes opciones:

- Iniciar:* Autenticarse en el sistema.
- Cambiar Contraseña:* Permitir al usuario cambiar su contraseña.
- Cerrar Sesión:* Cerrar la sesión en el sistema.
- Salir:* Salir del Sistema.

El menú [Usuarios] posee las siguientes opciones:

- Crear:* Crear un nuevo usuario.
- Modificar:* Modificar un usuario.
- Eliminar:* Eliminar a un usuario.

El menú [Centro Computo] incluye las siguientes opciones:

- Salas...:* Gestión de salas.
- Computadores...:* Gestión de estaciones de trabajo.
- Horarios:* Este menú a su vez incluye las siguientes opciones:
 - Período:* Este menú posee las siguientes opciones:
 - Nuevo:* Crear un nuevo Período.
 - Modificar:* Modificar un Período.
 - Cerrar:* Cierra un período.
 - Horarios...:* Gestión de Horarios.
- Reservaciones:* Gestión de Reservaciones.
- Explorador...:* Presenta en pantalla el explorador de estaciones de trabajo, como el explorador de Windows.

El menú [Red] incluye las siguientes opciones:

- Iniciar Servidor:* Inicia el servidor para escuchar peticiones de conexión.
- Parar Servidor:* Detiene el servidor.

El menú [Reportes] incluye las siguientes opciones:



- Horarios:* Reportes de horarios.
Reservaciones: Ejecuta el reporte de reservaciones.
Consumos x Usuario: Realizar el reporte de consumos por un usuario determinado.
Consumos x Estación: Reporte de Consumos por Estación.

El menú [Ayuda] incluye las siguientes opciones:

- Contenido:* Abre el archivo de ayuda.
Acerca de...: Presenta información de la versión.



DESCRIPCION DE LA BARRA DE HERRAMIENTAS:

La barra de herramientas contiene las siguientes opciones como se presenta en la figura:



Botón Iniciar / Cerrar Sesión: Iniciar una sesión en el sistema servidor o cerrar.



Botón Usuarios: Crear, modificar o eliminar a un usuario.



Botón Salas: Administración de salas.



Botón Computadores: Administración de estaciones de trabajo.



Botón Horarios: Administración de horarios de clases de las salas.



Botón Reservaciones: Administración de reservaciones de las salas.



Botón Explorador: Similar al explorador de Windows para explorar las estaciones de trabajo.



Botón Reportes: Reportes de Horarios y Reservaciones.



Botón Ayuda: Abrir el archivo de ayuda.



6.3 MENUS DEL SISTEMA

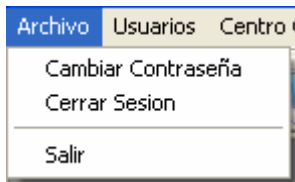
El MENU [Archivo]

Cuando ningún usuario está autenticado el menú [Archivo] presenta las siguientes opciones:



Login: Ingresar al sistema mediante el número de cédula y una contraseña.

Cuando un usuario está dentro del sistema el menú [Archivo] presenta las siguientes opciones:



Cambiar Contraseña: Cambiar la contraseña de un usuario.

Cerrar Sesión: Cerrar la sesión en el sistema.

Salir: Salir del sistema. Esta opción se presenta cuando el usuario es Administrador, caso contrario no está presente.

Menú [Login]:

Se ingresa el Nombre de Usuario y la Contraseña para entrar al Sistema. El sistema verificará el tipo de usuario. Si es administrador se habilitarán todas las opciones disponibles. Si es un usuario estudiante se habilitarán algunas opciones. De la misma manera si es un usuario Profesor se restringe el acceso a algunas opciones, pero adicionalmente se comprueba si éste tiene clases o no.

Menú [Cambiar Contraseña]:



Cambiar Contraseña

Actual Contraseña:

Nueva Contraseña:

Confir. Contraseña:

Para cambiar la contraseña se elige esta opción. Se debe ingresar la contraseña anterior, la nueva contraseña y la confirmación de la nueva contraseña.

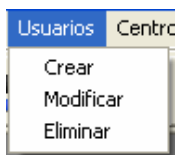
Menú [Cerrar Sesión]:

Cerrar la sesión en el sistema. Cuando se selecciona un computador en el Explorador de Estaciones la sesión es cerrada automáticamente.

Menú [Salir]:

Salir del Sistema Servidor. Esta opción está presente siempre y cuando un usuario esté autenticado en el sistema como administrador.

EI MENU [Usuarios]



Crear: Crear un nuevo usuario en el sistema.

Modificar: Modificar los datos de un usuario en el sistema.

Eliminar: Eliminar a un usuario del sistema.

Menú [Crear]:



Usuarios - [Nuevo]

Codigo:

Cedula: 0102248713

Apellidos: Perez

Nombres: Juan

Password:

Confir. Password:

Tipo Usuario: Estudiante

Cuenta :

Escuela: Civil Ilimitado

Año: 5

Horas Asignadas: 80

Tiempo Consumo: 0:00:00

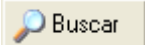
Tiempo Disponibles: 80:00:00

Observaciones:

Aceptar Cerrar

Se ingresan todos los datos necesarios para la creación de un nuevo usuario; pudiendo ser Administrador, Estudiante o Profesor.

Menú [Modificar]:

Al seleccionar esta opción se presenta el mismo cuadro de diálogo que en el menú [Crear] pero con un botón adicional  para buscar a un usuario al cual se va a modificar. Cuando se presiona este botón se presenta el siguiente cuadro de diálogo:

Buscar Usuarios

Buscar por: Nombres

Apellidos: perez

Nombres:

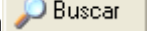
Id	Cedula	Apellidos	Nombres	Tipo	Escuela	Año
3	010356448	Perez	Juan	2	Civil	5

Aceptar Limpiar Cerrar

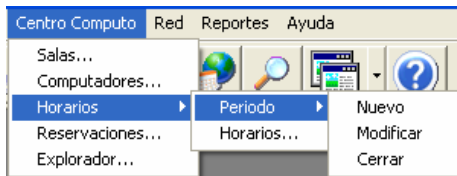
Los criterios de búsqueda permitidos son por Código, Cédula y Nombres.



Menú [Eliminar]:

Para la eliminación de un usuario previamente se deberá buscarlo con el botón , similar a la modificación de un usuario.

El MENU [Centro Computo]



Salas...: Se realiza la creación, modificación y eliminación de una sala.

Computadores...: Principalmente se realiza la creación, modificación y eliminación de un computador.

Periodo:

Nuevo: Crear un nuevo período lectivo.

Modificar: Modificar el nuevo período lectivo.

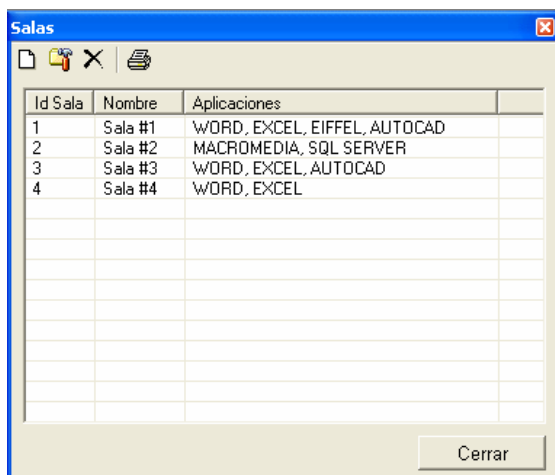
Cerrar: Cerrar el nuevo período lectivo.

Horarios...: Se realiza la creación, modificación y eliminación de un horario de clases de una sala.

Reservaciones...: Se realiza la creación, modificación y eliminación de una reservación de una sala.


Explorador...: Se realiza la revisión o exploración de las estaciones de trabajo en las diferentes salas.

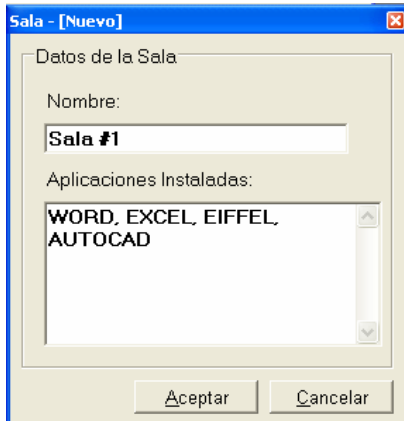
Menú [Salas]:



Gestionar las salas de un Centro de Cómputo. Lista las salas existentes en el Centro de Cómputo y la barra de herramientas tiene los siguientes botones:



 Botón [Nuevo]:




Datos de la Sala

Nombre:
Sala #1


Aplicaciones Instaladas:
WORD, EXCEL, EIFFEL,
AUTOCAD

Aceptar Cancelar


Para crear una nueva sala se tiene que ingresar el Nombre de la sala y las Aplicaciones Instaladas.

 Botón [Modificar]:

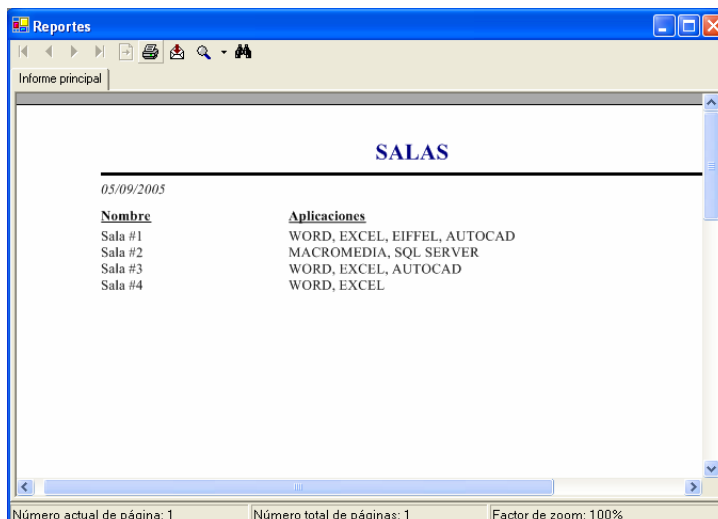
Previamente se debe seleccionar una sala de la lista. Se carga el mismo cuadro de diálogo de Botón [Nuevo] con los datos de la sala a modificar.

 Botón [Eliminar]:

Previamente se debe seleccionar una sala de la lista. Se carga el mismo cuadro de diálogo de Botón [Nuevo] con los datos de la sala a modificar.

 Botón [Imprimir]:

Se imprime las salas visualizadas en la lista.



Reportes

Informe principal

SALAS

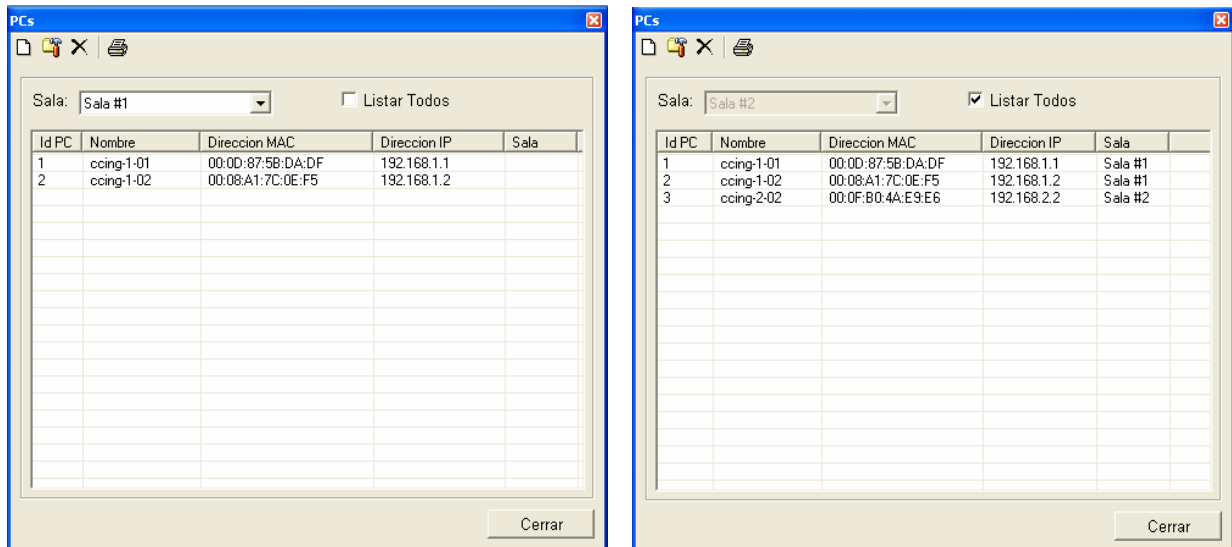
05/09/2005

<u>Nombre</u>	<u>Aplicaciones</u>
Sala #1	WORD, EXCEL, EIFFEL, AUTOCAD
Sala #2	MACROMEDIA, SQL SERVER
Sala #3	WORD, EXCEL, AUTOCAD
Sala #4	WORD, EXCEL

Número actual de página: 1 Número total de páginas: 1 Factor de zoom: 100%




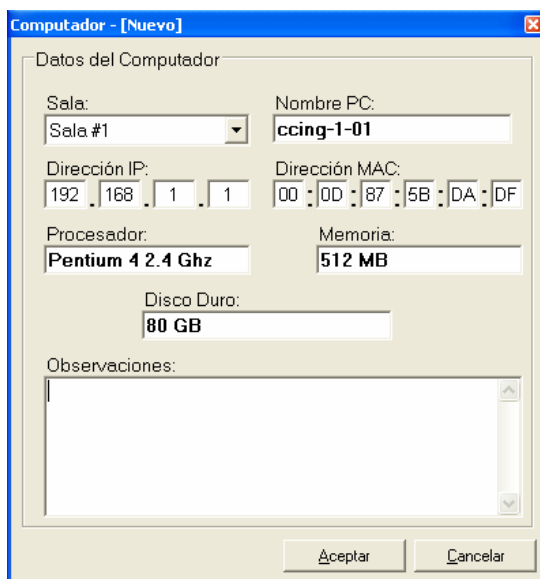
Menú [Computadores...]



Gestión de computadores de las salas del Centro de Cómputo. Lista los computadores por salas o lista la totalidad de los computadores existentes o ingresados en el sistema.

La barra de Herramientas incluye los siguientes botones:

 Botón [Nuevo]:





Para crear una nueva estación u ordenador se tiene que seleccionar la sala al que va a pertenecer este ordenador, el nombre de la estación (tiene que ser único), la Dirección IP (se valida que sea única), la Dirección MAC (valor tiene ser único), el Procesador, Memoria, Disco Duro y por último ciertas observaciones o comentarios.



Botón [Modificar]:

Previamente se debe seleccionar un computador de la lista. Se carga el mismo cuadro de diálogo de Botón [Nuevo] con los datos del ordenador a modificar.



Botón [Eliminar]:

Previamente se debe seleccionar un ordenador de la lista. Se carga el mismo cuadro de diálogo de Botón [Nuevo] con los datos del computador a modificar.



Botón [Imprimir]:

Se imprime los computadores visualizadas en la lista.

Reportes

Informe principal

COMPUTADORES

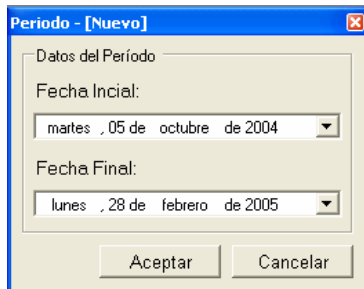
05/09/2005

NombrePC	DireccionMAC	DireccionIP	Procesador	Memoria	DiscoDuro
Sala: Sala #1					
ccing-1-01	00:0D:87:5B:DA:DF	192.168.1.1	Pentium 4 2.4	512 MB	80 GB
ccing-1-02	00:08:A1:7C:0E:F5	192.168.1.2	Pentium 4 2.6	512 MB	80 GB

Número actual de página: 1 | Número total de páginas: 1 | Factor de zoom: 100%

Menú [Periodo]

[Periodo – Nuevo]



Crea un nuevo período lectivo. Se tiene que ingresar una fecha inicial y una fecha final.

[Período – Modificar]:

Se carga el mismo cuadro de diálogo que [Nuevo] con los datos del período a modificar.

[Período – Cerrar]

Se carga el mismo cuadro de diálogo que [Nuevo] con los datos del período a Cerrar. Cuando se cierra un período se vacían los datos de período, horarios, reservaciones y usuarios estudiantes. Así como también el histórico.

Menú [Horarios...]



Gestionar los horarios de clases de cada sala del Centro de Cómputo. Se listan los horarios de clases por sala. Si no existen salas no se habilita la gestión de horarios.

La barra de herramientas incluye las siguientes opciones:



Botón [Nuevo]:



Horario - [Nuevo]

Datos del Horario

Día:
Lunes

Hora Inicio: 7:00 Hora Final: 9:00

Asignatura:
Leng. de Programacion

Profesor:
Arbito Gerardo

Aceptar Cancelar

Para crear un nuevo horario de clases se tiene que seleccionar la sala al que va a pertenecer este horario, seleccionar el día, ingresar la hora de inicio, la hora final, escribir la asignatura y por último seleccionar un profesor. En este combo se listan únicamente profesores. Si no existen profesores no se permite el ingreso de un horario.



Botón [Modificar]:

Previamente se debe seleccionar un horario de la lista. Se carga el mismo cuadro de diálogo de Botón [Nuevo] con los datos del horario a modificar.



Botón [Eliminar]:

Previamente se debe seleccionar un horario de la lista. Se carga el mismo cuadro de diálogo de Botón [Nuevo] con los datos del horario a modificar.



Botón [Imprimir]:

Se imprime el horario de la sala seleccionada. En el caso de que se active la casilla de verificación [Imprimir Todas las Salas] se imprimirán los horarios de clase de todas las salas.



Hora	Lunes	Martes	Miercoles	Jueves	Viernes	Sabado
7:00:00 9:00:00	Leng. de Programacion Arbito Gerardo	Leng. de Programación Arbito Gerardo		Ingenieria Software Espinoza Mauricio	Prog. Orientada Objetos Vanegas Pablo	
9:00:00 11:00:00	Sist. Oper. I Granda Maria Fernanda	Prog. Orientada Objetos Vanegas Pablo	Expresión Gráfica León Luis	Sist. Oper. II Parra Otto	Base de Datos II Mendez Lucia	
11:00:00 13:00:00		Expresión Gráfica León Luis	Expresión Gráfica León Luis	Expresión Gráfica León Luis	Expresión Gráfica León Luis	
15:00:00 17:00:00	Expresión Gráfica León Luis	Org. Datos y Archivos Mendez Lucia	Org. Datos y Archivos Mendez Lucia			
17:00:00 19:00:00	Utilitarios Jimenez Juan	Base de Datos II Mendez Lucia	Base de Datos II Mendez Lucia	Leng. de Programación Espinoza Mauricio		

Menú [Reservaciones...]:

Fecha Real...	Fecha Reservacion	Hora Inicial	Hora Final
05/09/2005	miércoles, 01 de diciembre de 2004	07:00	09:00

Gestionar las reservaciones de cada sala del Centro de Cómputo. Se listan las reservaciones por sala. Si no existen salas no se habilita la gestión de reservaciones.

La barra de herramientas incluye las siguientes opciones:



Botón [Nuevo]:



Reservaciones - [Nuevo]

Datos de la Reservación:

Fecha Reservación:
miércoles, 01 de diciembre de 2004

Hora Inicio: 7:00 Hora Final: 9:00

Descripción:
Examen Prog. Orientación Objetiva

Profesor:
Vanegas Pablo

Aceptar Cancelar

Para crear una nueva reservación se tiene que seleccionar la sala a la que va a pertenecer esta reservación, seleccionar la fecha, ingresar la hora de inicio, la hora final, escribir una descripción y por último seleccionar un profesor responsable. En este combo se listan únicamente profesores. Si no existen profesores no se permite el ingreso de la reservación; sin embargo el que realiza la reservación puede ser un estudiante.



Botón [Modificar]:

Previamente se debe seleccionar una reservación de la lista. Se carga el mismo cuadro de diálogo de Botón [Nuevo] con los datos de la reservación a modificar.



Botón [Eliminar]:

Previamente se debe seleccionar una reservación de la lista. Se carga el mismo cuadro de diálogo de Botón [Nuevo] con los datos de la reservación a modificar.

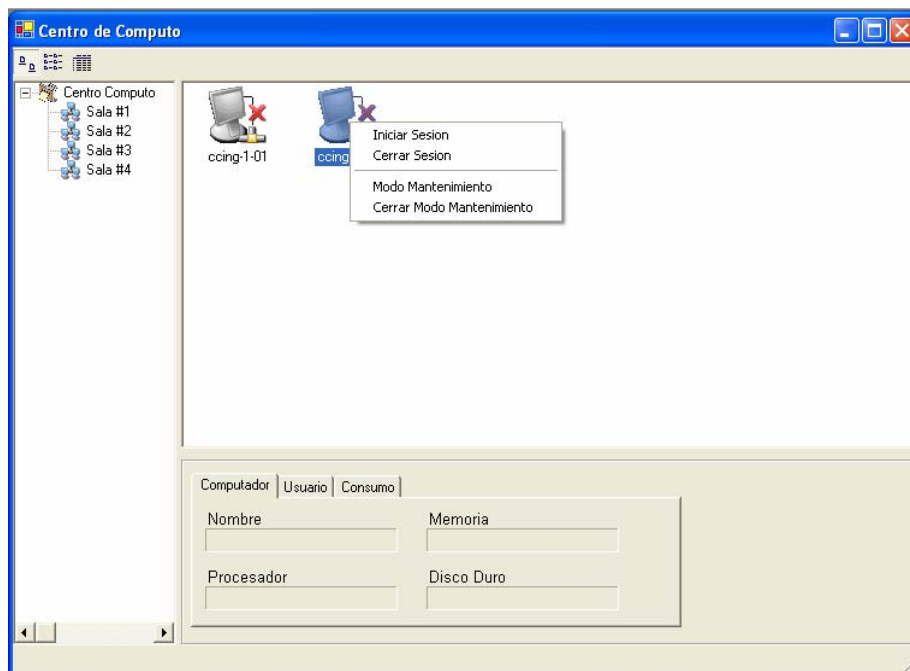
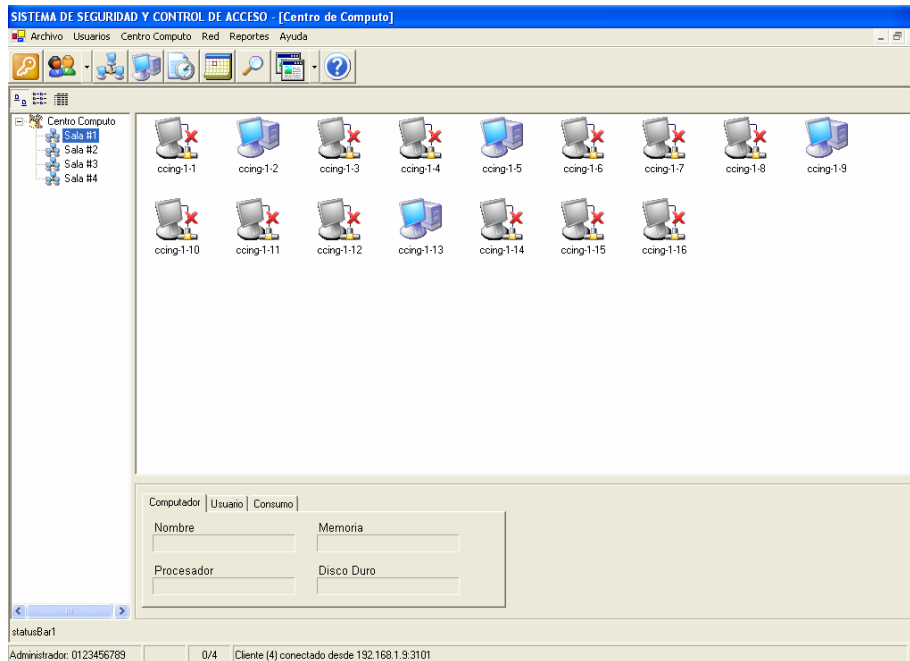


Botón [Imprimir]:

Ver la opción [Reportes – Reservaciones].

Menú [Explorador...]

Desde el explorador de computadores se puede navegar por las salas del centro de cómputo y observar los computadores y su estado (disponibles, ocupados, etc.). Así mismo desde el explorador se puede seleccionar un computador para utilizar. Si es administrador puede habilitar una máquina en modo mantenimiento. Los estudiantes únicamente pueden iniciar una sesión



EI MENU [Red]





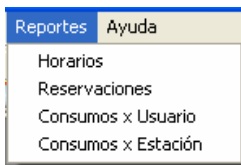
Menú [Iniciar Servidor]:

Iniciar el Servidor para comenzar a escuchar peticiones de clientes.

Menú [Parar Servidor]:

Detener el servidor para que no acepte conexiones o peticiones de clientes.

EI MENU [Reportes]



Horarios: Imprime un reporte de todos los horarios de clases.

Reservaciones: Imprime un reporte de las reservaciones de las salas.

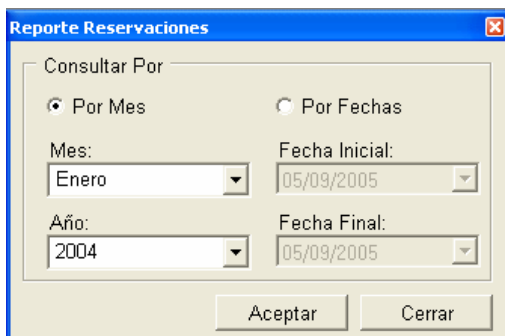
Consumos x Usuario: Imprime un reporte del consumo de un usuario específico.

Consumos x Estación: Imprime un reporte del consumo de una estación o computador específico.

Menú [Horarios]:

Imprime un reporte de horarios de clases de todas las salas del Centro de Cómputo. Listando la hora inicial, hora final, el día, el profesor responsable y la asignatura. Ver Menú [Horarios...], botón [Imprimir].

Menú [Reservaciones]



Imprimir [Por Mes]: Imprime las reservaciones de un mes. Especificar el mes y el año.

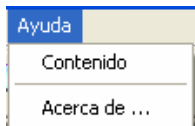


Imprimir [Por Fechas]: Imprime las reservaciones en un intervalo de fechas. Especificar la fecha inicial y la fecha final.

The screenshot shows a window titled 'Reportes' with a toolbar and a main content area. The content area displays a report titled 'RESERVACIONES DE SALAS' for the month of December 2004. The report includes a table with columns for reservation date, start time, end time, and user ID. The reservation is for 'Sala #1' (VanegasPablo) on 01/12/2004 from 0:00:00 to 9:00:00, with user ID 012345678. The status is 'Examen Prog. Orien Obj'. The report was generated on 05/09/2005. The status bar at the bottom shows 'Número actual de página: 1', 'Número total de páginas: 1', and 'Factor de zoom: 100%'.

Mes: Diciembre Año: 2004		05/09/2005			
Sala #1	Fecha Reservacion	Horainicial	HoraFinal	Usuario	
VanegasPablo Examen Prog. Orien Obj	01/12/2004	0:00:00	7:00:00	9:00:00	012345678

El MENU [Ayuda]



Menú [Contenido]:

Presenta en pantalla el archivo de ayuda de la aplicación.

Menú [Acerca de...]



6.4 HERRAMIENTAS

MIGRAR DATOS ESTUDIANTES.

Esta herramienta está diseñada para cargar datos de los estudiantes matriculados desde la base de datos de la Universidad hacia la base de datos del sistema.

The screenshot shows the 'Migrar Datos de Estudiantes' window. At the top, there is a dropdown menu for 'Facultad:' set to 'INGENIERIA' and a 'Cargar Datos' button. Below this is a large, empty table area. To the right of the table, there is a 'Horas Asignadas' input field with the value '80', a 'Migrar Datos' button, and a 'Salir' button. At the bottom left, there is a 'Registro:' label and an empty text input field.

The screenshot shows the 'Migrar Datos de Estudiantes' window with a populated table. The table has columns for 'Cedula', 'Apellidos', 'Nombres', 'Escuela', and 'Curso'. The data is as follows:

Cedula	Apellidos	Nombres	Escuela	Curso
0104242987	DUCHIMAZA PENARANDA	ANGEL PATRICIO	CIVIL	3
0104420096	BARRERA CRESPO	PEDRO DAVID	CIVIL	1
0105233514	PELAEZ CENTENO	BLASCO ANTONIO	ELECTRICA	1
1400634521	SUQUI MOROCHO	NORMA ESTHER	CIVIL	1
0104807961	PAUTA SUAREZ	JOHANNA CRISTINA	SISTEMAS	1
0105232425	PELAEZ CENTENO	CARLOS FABIAN	ELECTRICA	1
1400554539	ORELLANA JARA	DIEGO FERNANDO	CIVIL	1
1104495609	ESPINOZA MORA	JOSE LUIS	ELECTRICA	1

Below the table, there is a 'Registros: 882' label. To the right, there is a 'Horas Asignadas' input field with the value '80', a 'Migrar Datos' button, and a 'Salir' button. At the bottom left, there is a 'Registro:' label and an empty text input field.



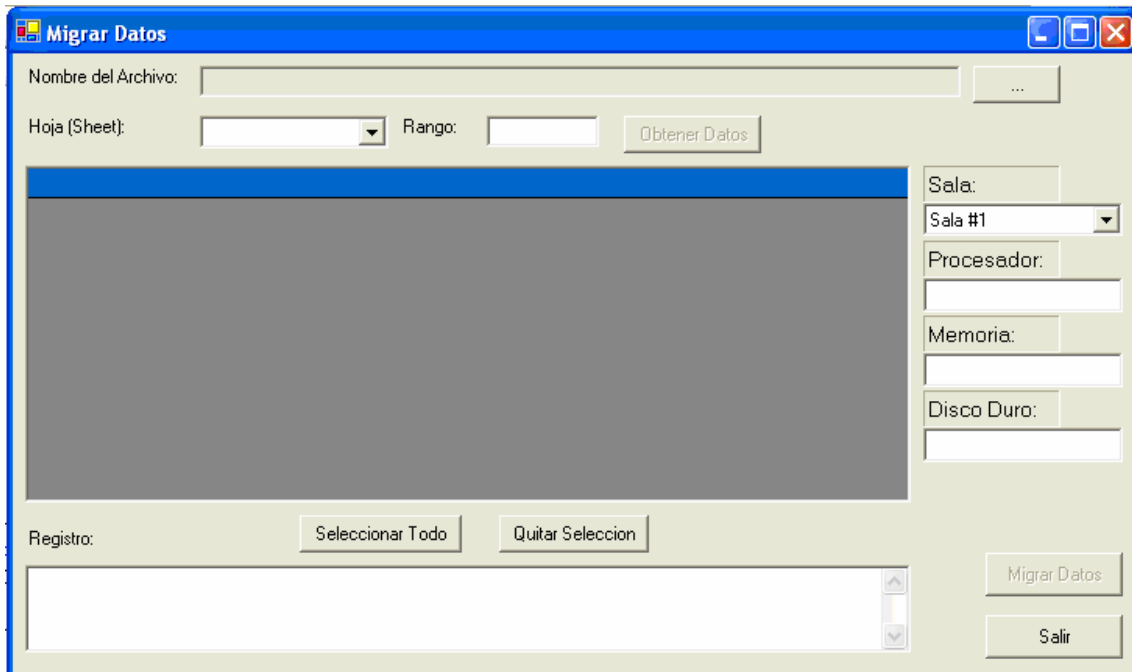
Cedula	Apellidos	Nombres	Escuela	Año
0104242987	DUCHIMAZA PENARANDA	ANGEL PATRICIO	CIVIL	3
0104420096	BARRERA CRESPO	PEDRO DAVID	CIVIL	1
0105233514	PELAEZ CENTENO	BLASCO ANTONIO	ELECTRICA	1
1400634521	SUQUI MOROCHO	NORMA ESTHER	CIVIL	1
0104807961	PAUTA SUAREZ	JOHANNA CRISTINA	SISTEMAS	1
0105232425	PELAEZ CENTENO	CARLOS FABIAN	ELECTRICA	1
1400554539	ORELLANA JARA	DIEGO FERNANDO	CIVIL	1
1104495609	ESPINOZA MORA	JOSE LUIS	ELECTRICA	1

Pasos para migrar Datos:

- Al momento de cargar esta herramienta se cargan las facultades en la lista de facultades.
- Seleccionar una facultad.
- Ingresar criterios para filtrar los datos. Estos filtros pueden ser:
 - Cedula
 - Apellidos
 - Nombres
 - Escuela
 - Curso
- Clic en el botón [Cargar Datos] y se listarán todos los datos que cumplan los criterios ingresados.
- Debe ingresar el número de horas que se asignarán a los estudiantes listados en la grilla.
- Presionar el botón de Migrar Datos.
- Una barra de estado indicará el porcentaje del proceso ejecutado.

MIGRAR DATOS COMPUTADOR

Esta herramienta está diseñada para migrar datos de computadores desde un archivo Excel a la base de datos de la aplicación.



Pasos para Migrar los Datos:

- Seleccionar el archivo de Excel.
- Seleccionar la hoja de Excel que contiene los datos y seguidamente ingresar el rango en la hoja que contiene los datos. Esta hoja de Excel estrictamente debe poseer el siguiente formato
 - Columna 1: Código del computador.
 - Columna 2: Dirección MAC del Computador.
 - Columna 3: Nombre del Computador.
 - Columna 4: Sala a la que pertenece el computador.
 - Columna 5: Dirección IP del Computador.
- Clic en el botón [Obtener Datos], se listarán los datos en la grilla.
- Seleccionar los computadores a migrar a través de la columna [Estado] de la grilla.
- Seleccionar la Sala a la que va a migrar los datos del computador (combo Salas)
- Ingresar datos del Procesador, Memoria y Disco Duro de los computadores a migrar.
- Clic en botón [Migrar Datos].



Migrar Datos

Nombre del Archivo: C:\Documents and Settings\Administrador.TESIS-RGC\Mis documentos\Tesis\datos_computador.xls

Hoja (Sheet): Hoja3 Rango: A3:e58

	F1	F2	F3	F4	F5	Estado
	35	00:11:5B:E6:	CCING-3-07	sala3	192.168.3.7	<input checked="" type="checkbox"/>
	36	00:11:5B:E6:	CCING-3-08	sala3	192.168.3.8	<input checked="" type="checkbox"/>
	37	00:11:5B:E6:	CCING-3-09	sala3	192.168.3.9	<input checked="" type="checkbox"/>
	38	00:11:5B:E6:	CCING-3-10	sala3	192.168.3.10	<input checked="" type="checkbox"/>
	39	00:11:5B:C7:	CCING-3-11	sala3	192.168.3.11	<input checked="" type="checkbox"/>
▶	40	00:11:5B:E4:	CCING-3-12	sala3	192.168.3.12	<input checked="" type="checkbox"/>
	41	00:11:5B:E6:	CCING-3-13	sala3	192.168.3.13	<input checked="" type="checkbox"/>
	42	00:11:5B:E4:	CCING-3-14	sala3	192.168.3.14	<input checked="" type="checkbox"/>
	43	00:11:5B:E4:	CCING-3-15	sala3	192.168.3.15	<input checked="" type="checkbox"/>
	44	00:11:11:3B:	CCING-4-01	sala4	192.168.4.1	<input type="checkbox"/>

Registro:

Ingresado ID: 21\n\r\nComputador Ingresado ID: 22\n\r\nComputador Ingresado ID: 23\n\r\nComputador Ingresado ID: 24\n\r\nComputador Ingresado ID: 25\n\r\nComputador Ingresado ID: 26\n\r\nComputador Ingresado ID: 27\n\r\nComputador Ingresado ID: 28\n\r\nComputador Ingresado ID: 29\n\r\n

Sala: Sala #3

Procesador: **Pentium 4 2.4 GHz**

Memoria: **512MB**

Disco Duro: **80 Gb**

7. CONCLUSIONES Y RECOMENDACIONES



7 CONCLUSIONES Y RECOMENDACIONES

7.1 CONCLUSIONES.

- Con el bloqueo de los computadores (Subsistema Cliente) se ha conseguido que únicamente los estudiantes que estén registrados en el sistema puedan ocupar los equipos de computación, lo que ha permitido racionalizar y optimizar el uso de los computadores; esto es que los estudiantes utilicen únicamente un computador a la vez y no varios.
- El módulo de monitorización de salas, del Subsistema Servidor, permite al Administrador del sistema, llevar un mejor control del uso de equipos en base a la verificación del estado en el que se encuentran los computadores, esto es ocupado (sesión iniciada), disponible (sesión cerrada) o en modo mantenimiento (desbloqueada).
- La obtención de datos de estudiantes matriculados en la facultad de Ingeniería se facilitó al hacer uso del Subsistema de “Migración de Datos de Estudiantes”, con lo que se pudo utilizar la información del Sistema de Matricula Diferenciada.
- El software para gestionar las salas del centro de cómputo permite al administrador del Centro de Cómputo llevar, de una manera automatizada, la información de los equipos de computación, horarios de clases y reservaciones correspondientes a las salas del Laboratorio de Cómputo.
- Como consecuencia de una metodología orientada a objetos se han creado algunos componentes que podrían ser reutilizados en futuros proyectos de esta índole, tales como los componentes para la comunicación cliente – servidor que pueden ser utilizados en el desarrollo de una aplicación para el monitoreo de una red, por ejemplo.



- Para el control de acceso se ha utilizado la tecnología de tarjetas magnéticas principalmente por el bajo costo de los lectores y de las tarjetas y por la facilidad de codificación.

7.2 RECOMENDACIONES.

- Al Sistema, paulatinamente, se puede agregar nuevas funcionalidades, más específicamente en la parte del sistema que corresponde a la comunicación cliente – servidor. Estas nuevas funcionalidades pueden ser órdenes enviados desde el servidor hacia el cliente, como por ejemplo enviar un comando de “Apagar” a los computadores de una determinada sala.
- El sistema es una aplicación para Windows y utiliza formularios de Windows y el diseño del sistema se basó en la arquitectura de tres capas, es así que es factible modificar la capa de interfaz windows correspondiente al subsistema de gestión del centro de cómputo (gestión de usuarios, gestión de salas, gestión de horarios y reservaciones) a una interfaz web, para aumentar la funcionalidad de este subsistema, y poder gestionar remotamente las salas del Centro de Cómputo desde cualquier computador que posea internet.

8. BIBLIOGRAFIA



8 BIBLIOGRAFIA

1. RUMBAUGH J, JACOBSON I, BOOCH G., “*El Proceso Unificado de Desarrollo de Software*”, Addison – Wesley.
2. RUMBAUGH J, JACOBSON I, BOOCH G., “*El Lenguaje Unificado de Modelado*”, Addison – Wesley, 1998, 482 pg.
3. MUELLER, J., “*Net Framework Solutions – in search of the lost Win32 API*”, Sybex, 2002, 424 pg.
4. PETZOLD, Ch., “*Programming Microsoft Windows with C#*”, Microsoft Press, 2002, 1071 pg.
5. BLUM R., “*C# Network Programming*”, Sybex, 2003, 647 pg.
6. REID F., “*Network programming in .NET*”. Elsevier Digital Press, 2004, 541 pg.
7. KIMMEL P., “*Advanced C# Programming*”, McGrawHill / Osborne, 2002, 551 pg.
8. ARCHER T., “*Inside C#*”, Microsoft Press, 2001.
9. TITUS T., “*C# Threading Handbook*”, Apress, 2004, 288 pg.
10. PETZOLD Ch., “*Programming Windows*”, Microsoft Press, 1998.
11. KULINIEWICZ P., “*Windows API Reference*”, 2001
12. Jelsoft Enterprises Ltd.”NET / C#”. 2005. www.codeguru.com.



13. “Programación en Castellano”. 2005. www.programacion.com

14. KPD-Team. “Your #1 source of information about programming the windows API”. 2002. www.allapi.net

15. Guillermo Som. “La Web del Visual Basic, C#, .NET y más...”.
www.elquille.info