



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

**FACULTAD DE ESTUDIOS SUPERIORES
"ARAGON"**

**"SISTEMA ADMINISTRADOR DE CENTROS DE CÓMPUTO
PARA NIVEL BACHILLERATO SOBRE PLATAFORMA .NET"**

TESIS

**QUE PARA OBTENER EL TÍTULO DE INGENIERO EN
COMPUTACIÓN**

PRESENTAN

**ROSARIO GARCÍA LUCERO
HERNÁNDEZ MARTÍNEZ HÉCTOR**

ASESOR: M. en C. MARCELO PÉREZ MEDEL



México 2006



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

CONTENIDO

Indice

1. Problemática

Presentación del caso por estudiar

Descripción de los procesos del Centro

Proceso 1. Entrada de usuarios y asignación de Equipo

Proceso 2. Salida de usuarios

Proceso 3. Alta de un equipo

Proceso 4. Baja de un equipo

Proceso 5. Prestamos

Proceso 6. Horarios

Proceso 7. Servicios

Diagramas de los procesos del Centro

Diagrama de actividades atender usuario

Diagrama de actividades servicio-sala

Diagramas de actividades de alta de un equipo

Diagramas de actividades de baja de un equipo

Diagramas de actividades de préstamos

Diagramas de actividades de horarios

Problemática

Algunas alternativas para su solución

Alternativa 1: Comprar un software que automatice los procesos

Arroba

Ádroba

Cyber Admin Control

RentPC

Caffe-Control

Alternativa 2: Que el centro de cómputo siga funcionando igual tratando de subsanar los problemas que existen, sin automatizar

Alternativa 3: Solo automatizar algunos procesos

Alternativa 4: Hacer un software

2. Perfil tecnológico

Metodologías para el desarrollo de software

Metodología tradicional o en cascada

Metodologías recientes

Otras metodologías

Metodología Delphi

Características de la consulta

Metodología eXtrema

- Planificación
- Diseño, desarrollo y pruebas
- Valores y principios de la programación extrema
- Metodología GRAPPLE
 - Recopilación de necesidades
 - Análisis
 - Diseño
 - Desarrollo
 - Distribución

Herramienta no computacional: Lenguaje Unificado de Modelado (UML)

Herramientas computacionales para el desarrollo de software

- Rational Rose
- SELECT Enterprise
- Visual UML
- UML Modeler
- Microsoft Office Visio 2003

Lenguajes de Programación

- Programación Estructurada
- Programación Orientada a objetos
 - Visual Basic 6
 - Java
 - Visual Basic .NET (VB.NET)

Gestores de Base de Datos

- Microsoft Office Access
- Microsoft SQL Server 2000

Metodología, herramienta y lenguaje de programación, ideales para el desarrollo de Software

3. Recopilación de las necesidades del sistema

Necesidades propias para el sistema

- Administradores
- Usuarios
- Sanciones
- Identificación de un usuario
- Préstamo de Equipo
- Ubicación (sala)
- Servicios
- Equipos
- Reportes

Personas implicadas en el sistema

Diagramas de responsabilidades

- Diagrama de responsabilidades atender usuarios
- Diagrama de responsabilidades Servicio-Sala
- Diagrama de responsabilidades de alta de un equipo
- Diagrama de responsabilidades de baja de un equipo

Diagrama de responsabilidades de préstamos
Diagrama de responsabilidades de horarios

Diagramas generales de los casos de uso de SACC

Factores primordiales que influyen en la decisión de las elecciones de las herramientas

Selección de Herramientas para el desarrollo del sistema

¿Que metodología elegir?

¿Que herramientas elegir?

Herramienta computacional: Microsoft Office Visio

Lenguaje de programación

Gestor de Bases de Datos Microsoft SQL Server 2000

4. Análisis y Diseño

Desarrollo de los Casos de Uso

Casos de Uso de SACC

Autenticación

Dar de alta un Administrador

Eliminar un Administrador

Editar el registro de un Administrador

Cambio de contraseña

Realizar reportes

Dar de alta un servicio

Dar de baja un servicio

Editar el registro de un servicio

Ver todos los servicios

Crear relación Equipo-Servicio

Eliminar relación Equipo-Servicio

Crear relación Ubicación (sala)-Servicio

Eliminar relación Ubicación (sala)-Servicio

Dar de Alta (equipos o CPU's o monitores o teclados o mouse o extras y/o ubicaciones)

Dar de baja (CPU's o monitores o teclados o mouse, extras)

Editar el registro de (equipos o CPU's o monitores o teclados o mouse o extras y/o ubicaciones)

Eliminación (equipos o CPU's o monitores o teclados o mouse o extras y/o ubicaciones)

Ver todos (equipos, CPU's, monitores, teclados, mouse, extras u ubicaciones)

Alta de un usuario

Edición de usuario

Eliminación de usuario

Búsqueda de usuario

Dar de alta una sanción

Editar una sanción

Dar de baja una sanción

Ver todas las sanciones

Modelo de la estructura de clases de SACC

Desarrollo del diagrama de clases inicial

Llenado de las clases

Diseño de SACC
Componentes del sistema

De los casos de uso al Diseño de las interfaces
Interfaces de SACC
Diccionario del modelo de clases

5. Implementación

Orientación a las interacciones y cambios de estado

Componente UML: Base de Datos de SACC

Diagrama de secuencias
Diagrama de secuencias de instancias de Ocupación (Préstamo de Equipo)
Diagrama de secuencias genérico de Alta de Equipo
Diagrama de secuencias genérico de Alta de Usuario

Instalación y distribución de SACC

6. Propuestas y resultados

Políticas oficiales de desempeño documentado en la ENP 9

Pruebas de SACC

Comprobación del cumplimiento de los casos de uso de SACC

Posibles líneas de desarrollo

Conclusiones

Anexos

Bibliografía

Introducción

Actualmente, el avance tecnológico se ha venido caracterizando tanto por el uso de los equipos de cómputo, el de Internet y el de los dispositivos móviles. Ya que gracias al servicio de Internet ya no sólo se pueden tener conectados equipos de cómputo, sino que también se puede conectar otros dispositivos cliente móviles como Pocket PC's, Palm's, teléfonos celulares, etc.

El principal objetivo de un equipo de cómputo es el de procesar de manera eficaz y rápida la información, mientras que el de Internet es el de mantener a dichos equipos conectados ya sea con otros equipos de cómputo o dispositivos móviles para poder compartir información entre ellos, proporcionando así a las personas de diversos lugares -como por ejemplo los de bibliotecas, escuelas, empresas y hogares- acceso a información de calidad que eduque, informe y entretenga.

Debido al aumento de la necesidad de los recursos anteriormente mencionados, éstos comienzan a ser accesibles al público. Por ende, actualmente en el desarrollo de aplicaciones ya no solo se piensa que la aplicación debe ejecutarse en una computadora, sino que también pueden en los dispositivos, pero eso no es todo, sino que ahora el usuario no es el que se tiene que acoplar a las aplicaciones, sino ahora son las aplicaciones al usuario, un claro ejemplo de esto es el servicio de .Net Passport de Microsoft y que muchos inconscientemente utilizamos, este servicio que desde 1999 permite entre otras cosas que el usuario se autentique una sola vez y luego acceda a sitios que hagan uso de este servicio sin necesidad de volver a identificarse.

Para que los estudiantes tengan acceso a equipos de cómputo y a Internet, la Universidad Nacional Autónoma de México (UNAM) cuenta con espacios generalmente denominados Centros de Cómputo para que los alumnos utilicen tales servicios como apoyo en el desarrollo de sus materias, prácticas, tareas y sobretodo para que logren una educación sólida.

Pero la tarea no termina ahí, sino que se pretende ofrecer un buen servicio, y así aprovechar al máximo los recursos con los que cuentan tales Centros.

Por experiencia, si se presta un poco de atención en los procedimientos de algunos de estos Centros de Cómputo, se podrá apreciar que a lo que muchos interesa, no es dar un buen servicio, sino contar con mejores equipos de cómputo, esto se debe a que no se obtienen ganancias monetarias directas, ni se paga más a los encargados del centro de cómputo de las escuelas por la cantidad de los alumnos que atiendan, ni se ven los gastos que implica una actualización de equipos. Sin embargo, operan y el servicio es eficiente, pero lo que se pretende con este trabajo es que se aproxime a ser eficaz, para poder brindar un mejor servicio a más alumnos con los recursos con que ya se cuenta.

Pero, ¿Por qué no se ha puesto atención en brindar mejores servicios?, la respuesta es por falta de tiempo, ya que la mayoría es usado para realizar manualmente tareas del centro de cómputo. Entonces, el hecho de crear un sistema que ayude llevar a cabo algunas de estas tareas, implicará, que los encargados tengan más tiempo para otras ocupaciones que no pueden ser automatizadas, como por ejemplo la atención a los alumnos, atención a equipos (instalaciones, configuraciones), etc.

Es por ello que este trabajo está enfocado al desarrollo de un Sistema que ayude a administrar las tareas para un centro de cómputo, específicamente para la Escuela Nacional Preparatoria Pedro de Alba, plantel 9.

El desarrollo del presente trabajo, se hará bajo la novedosa plataforma de Microsoft: “.NET Framework.”, que es un nuevo modelo de desarrollo de aplicaciones basado en Internet, además de tener otras tantas características que se mencionarán en este trabajo.

En el primer capítulo se presentará la forma en que se llevan a cabo las diferentes actividades dentro del centro de cómputo, además de conocer quienes llevan a cabo tales tareas, y con todo ello, analizar los problemas que implica al realizarlo de dicha manera además de presentar diversas opciones que se tienen para la solución de tales problemas.

En el segundo capítulo se verá el perfil tecnológico mencionando algunas metodologías existentes, herramientas computacionales y no computacionales, lenguajes de programación así como también algunos gestores de bases de datos. Además de analizar tales instrumentos para el desarrollo de soluciones, evitando gastos mayores.

En el tercer capítulo se analizarán los requerimientos para el desarrollo del Sistema Administrador de Centros de Cómputo, donde se menciona que factores se deben considerar para la recopilación de las necesidades del futuro sistema. Además, de mencionar cuales son los requerimientos propios del sistema, y de distinguir a los usuarios finales. También se da conocer una idea de lo que el sistema se espera que realice, especificando que herramientas de las mencionadas en el capítulo anterior conviene utilizar.

En el cuarto capítulo se llevará a cabo el análisis y el diseño para el desarrollo del sistema, en este, se realizará un análisis de las clases y objetos relacionados con el desarrollo de SACC (Sistema Administrador de Centros de Cómputo), dando una visión general de las tablas que más adelante conformarán la base de datos, además de analizar como se llevan paso a paso las tareas mencionadas en el capítulo tres.

En el quinto capítulo se verá la forma en que se implementará el sistema, presentando la base de datos con la que se trabajará, además de hacer la demostración de cómo se lleva a cabo la codificación de SACC usando como apoyo los diagramas de secuencia, es decir, la forma en que los componentes del sistema interactúan.

Finalmente, en el sexto capítulo, se mencionarán algunas propuestas a futuro para el sistema y por supuesto los resultados obtenidos al poner a trabajar a SACC.

1. Problemática

Presentación del caso por estudiar

Debido al gran avance tecnológico y a la rapidez con la que éste se viene gestando, el uso de la computadora hoy en día es imprescindible. En este sentido, contar con un centro de cómputo en el interior de las escuelas de la UNAM¹, es de mucha ayuda para los alumnos y en general para el personal ya sea para prácticas, para realizar tareas, para apoyar materias curriculares e incluso cursos extracurriculares.

Este trabajo está enfocado a un centro de cómputo de nivel bachillerato, simplemente porque por experiencia laboral se tiene más conocimiento de su estructura, funcionamiento, constitución, necesidades, etc., y, siendo más específicos, el escenario será el centro de cómputo de la Escuela Nacional Preparatoria (ENP) Plantel 9 "Pedro de Alba", ya que el encargado ha decidido incorporar un sistema que le ayude a disminuir su carga de trabajo, **automatizando** ciertos procesos para mejorar la calidad de los servicios que brinda, optimizar los recursos con los que cuenta, y, sobre todo, ahorrar tiempo y demás recursos como son los económicos, materiales y humanos, invirtiéndolos en otras funciones que benefician al Centro, a los alumnos y a todos los usuarios en general.

¹ Universidad Nacional Autónoma de México

En cuanto a administración, el objetivo de un centro de cómputo es gestionar de manera eficaz los recursos con los que se cuenta para satisfacer las necesidades de los usuarios, quienes en este caso son los alumnos o externos². El servicio es eficiente cuando se cumple con dicho objetivo, y es eficaz cuando, de igual forma, se cumple con el objetivo pero se gastan mínimamente los recursos.

Como es sabido, la materia prima de un centro de cómputo es la información además de los recursos humanos, materiales y financieros, que trabajando en conjunto, logran su principal objetivo. Es por ello que el manejo de la información es de gran importancia.

Por dichas razones es importante realizar un análisis cualitativo de cómo se llevan a cabo las funciones dentro del centro de cómputo para poder así, evaluar el servicio que actualmente se brinda a los alumnos, y visualizar de mejor manera las desventajas.

Este plantel de nivel medio superior de la UNAM está conformado por más de un centro de cómputo, con ubicaciones distintas, en los cuales se ofrecen servicios diversos, a saber:

1. La impartición de clases.
2. Realización de prácticas.
3. Servicio de paquetería (hoja de cálculo, procesador de textos, presentaciones, bases de datos, enciclopedia, traductor, etc.).
4. Servicio de impresión.
5. Servicio de Internet, entre otros.

Casi siempre se puede encontrar un centro dentro de la biblioteca, otro junto a los salones de clases y otro más en los laboratorios de investigación, siendo en promedio tres por escuela.

Estos Centros ofrecen sus servicios a diferentes tipos de usuarios, algunos se usan sólo para dar clases o para realizar actividades en las que únicamente participan profesores y otros dan servicio solamente a los alumnos. Aunque se debe tener en cuenta que dependiendo de las políticas de cada centro se pueden presentar administrativos, alumnos o académicos de otros planteles.

Algo más que caracteriza a estos centros es que trabajan bajo la plataforma Windows (básicamente Windows 98/Millennium/NT Server/2000/XP), debido a que los alumnos y profesores usan programas que trabajan bajo la plataforma Windows, sólo por mencionar algunos ejemplos, se tienen las materias curriculares (como informática) que están basadas en programas como Excel, Word, Power Point, varias materias extracurriculares (opciones técnicas en computación, en contabilidad, en histopatología) que igualmente trabajan con programas como COI, NOI y Visual Basic, además de que los equipos con los que se cuenta tienen licencia Windows.

A continuación, se presentará una descripción detallada de la forma en la que han llevado a cabo sus funciones las personas que colaboran en el centro de cómputo.

² Externos se refiere a los todos académicos y administrativos.

Descripción de los procesos del Centro

Primeramente, para que el centro de cómputo inicie su servicio, se debe contar con la presencia de un encargado, usualmente un Técnico Académico además de equipos en funcionamiento.

El técnico es la persona que se ocupa de las actividades administrativas que se realizan dentro del centro de cómputo.

Para ofrecer alguno de los servicios que brinda el centro de cómputo, por ejemplo:

- Impresión
- Internet
- Aplicaciones en general
- Uso de salas completas para clases, etc.

se sigue una secuencia de pasos para proveerlos.

Además, el encargado cuenta con tarjetones numerados, cada uno de éstos coincide con otro colocado en cada equipo para identificarlo. También cuenta con una libreta o un formato que imprime a diario con la fecha del día, para el registro de los usuarios del centro de cómputo.

La forma en que se agrupan las actividades que se llevan a cabo en un proceso está relacionada con el nombre que llevan por título.

Proceso 1. Entrada de usuarios y asignación de Equipo

Cuando un usuario llega al centro de cómputo y hace la petición de alguno de los servicios antes mencionados, si a consideración del encargado en turno, el usuario ha mostrado un buen comportamiento dentro del Centro en visitas anteriores, se realiza lo siguiente:

El encargado empieza su tarea ubicando si hay equipos libres que tengan el servicio solicitado o uno similar; una vez que los ubica, pide una identificación al usuario como garantía del equipo y así se continúa con el registro.

En caso de no haber equipos libres se le indica al usuario que regrese en un lapso de aproximadamente 20 minutos o si el equipo no cubre sus necesidades, se le indica que no hay equipo. Si el alumno no lleva identificación alguna se le indica que no puede hacer uso del centro de cómputo pero en caso de que el alumno sea un visitante frecuente y el encargado lo ubique se puede hacer una excepción y dejarlo pasar. Si el alumno va acompañado de un compañero que sí porte una credencial, y utilizarán el mismo equipo entonces se le hace a éste la indicación de que quedará como responsable de las acciones del otro.

Paralelamente al registro del usuario el encargado le indica la ubicación y número del equipo que se le asignará.

Así el usuario realiza su registro manualmente, proporcionando algunos datos, por mencionar los más relevantes:

- Nombre completo
- Número de Cuenta
- Aplicación a usar
- Hora de entrada
- Equipo y/o sala asignado, etc.

Una vez completado el registro, el usuario o los usuarios ocupan el equipo y el encargado lo considera como ocupado colocando la identificación del usuario(s) junto con el tarjetón que identifica al equipo. Por ende, todo aquel tarjetón que no cuenta con una identificación se considera como equipo para préstamo (no ocupado).

Proceso 2. Salida de usuarios

Cuando un usuario desea retirarse, pasa a recoger su identificación y registrar su salida, asentando la hora en la que deja el equipo.

Una vez que esto pasa, el equipo se considerará como desocupado.

Al final del día el encargado guarda en un lugar seguro el registro de los alumnos.

Proceso 3. Alta de un equipo

Cuando se asigna equipo nuevo al Centro o es necesario conocer en su totalidad el que ya posee, el encargado realiza un inventario.

Para esto, el encargado revisa todos los dispositivos y toma nota del tipo (mouse, monitor, CPU, teclado, etc.), de los números de inventario y de serie, de la marca, del modelo, y en ocasiones, según sea necesario, de otras características como ubicación, especificaciones técnicas, estado físico, etc., las cuales enlista manualmente en una hoja.

Posteriormente, si existe algún formato para su entrega, sólo se vacían los datos y en caso de no ser así, la mayoría de las veces el encargado realiza el formato de presentación en el programa que desee (Excel, Word, etc.), como lo considere necesario.

Proceso 4. Baja de un equipo

En caso de ser necesario dar de baja un Equipo por estados tales como: falla física irreparable, obsolescencia, u otro, el encargado toma las características necesarias del inventario ya almacenado, lo coteja con el equipo, realiza un reporte escrito de petición de baja y lo entrega a la Unidad de Resguardos.

Proceso 5. Prestamos

En caso de que se presten Equipos a un departamento externo al centro de cómputo, en el inventario se marcan como prestados, el lugar al que se prestó y el responsable de dicho préstamo.

Proceso 6. Horarios

Para llevar un control de los horarios en los que se encuentra ocupada determinada sala, ya sea por clases o eventos especiales, el encargado de horarios de asignaturas del plantel proporciona tal información al encargado del centro de cómputo. Esta información se pone a la vista de los usuarios para que la consideren.

Proceso 7. Servicios

Para ubicar los servicios con los que cuenta el centro de cómputo, primero se analiza si en el equipo se va a realizar una instalación nueva; si es así al inicio se hace la relación de los servicios que se instalarán; en caso de que el equipo ya cuente con una instalación, sólo se visita la sala y se hace una relación de los servicios que se encuentran en cada equipo; para ambas situaciones dichas relaciones se ponen a la vista del encargado.

Diagramas de los procesos del Centro

Una vez que se conoce la forma en que se llevan a cabo los diferentes procesos dentro del centro de cómputo se pueden representar en los Diagramas de Actividades.

Un Diagrama de Actividades modela la manera en que se presentan los procesos ya descritos.

A continuación se verán tales Diagramas.

Diagrama de actividades atender usuario

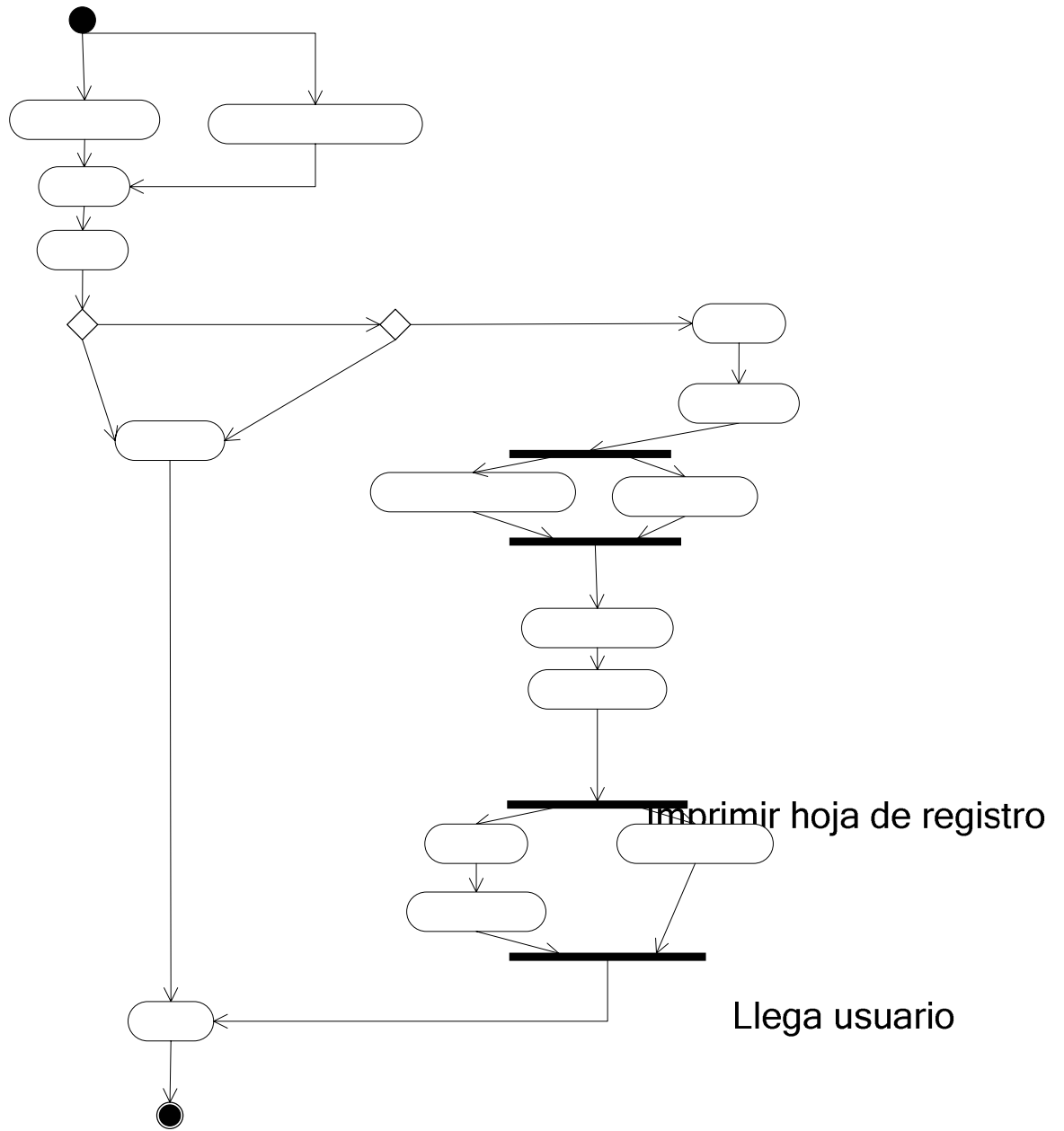


Figura 1. Diagrama de Actividades del proceso "Atender Usuario".

[Ha mostr

Diagrama de actividades servicio-sala

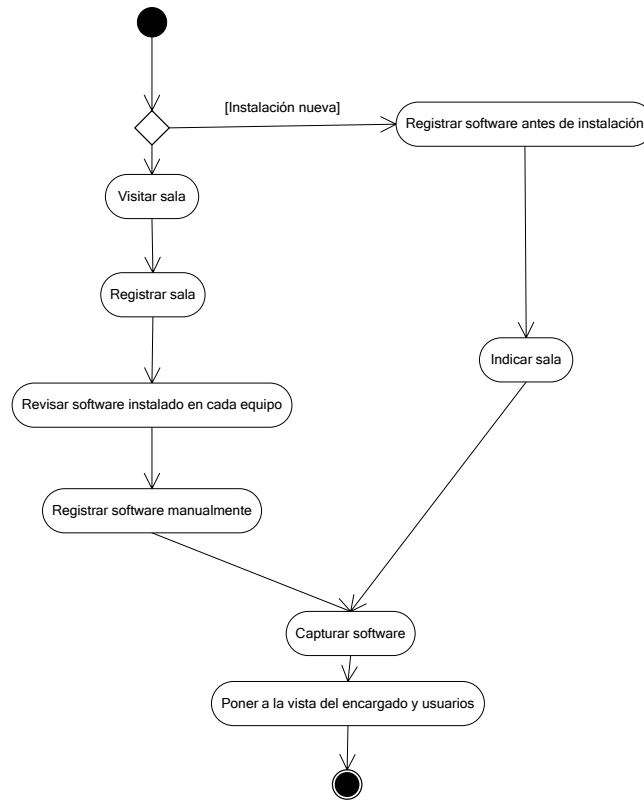


Figura 2. Diagrama de Actividades del proceso "Servicio-Sala".

Diagramas de actividades de alta de un equipo

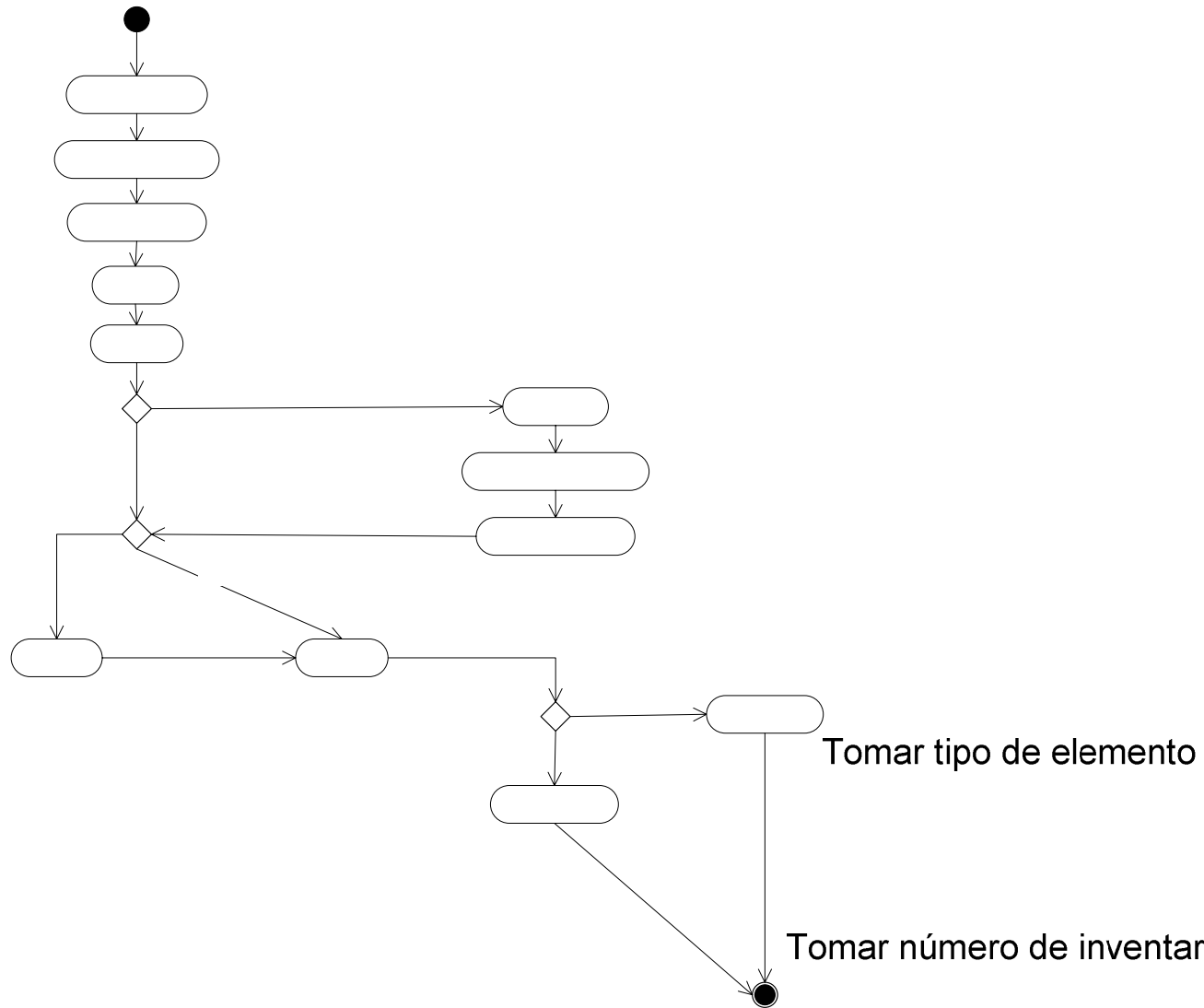


Figura 3. Diagrama de Actividades del proceso "Alta de un Equipo".

Tomar tipo de elemento

Tomar número de inventar

Tomar número de serie

Tomar marca

Tomar modelo

Diagramas de actividades de baja de un equipo

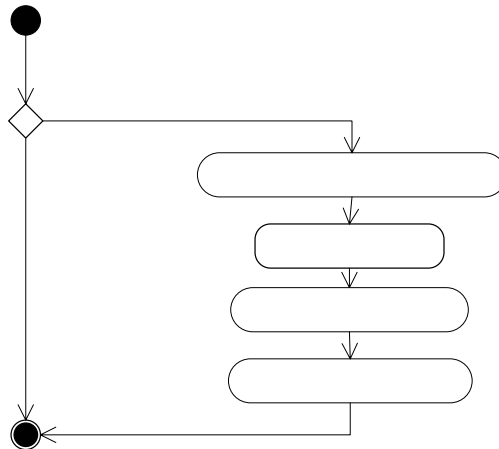


Figura 4. Diagrama de Actividades del proceso "Baja de un equipo"

Diagramas de actividades de préstamos

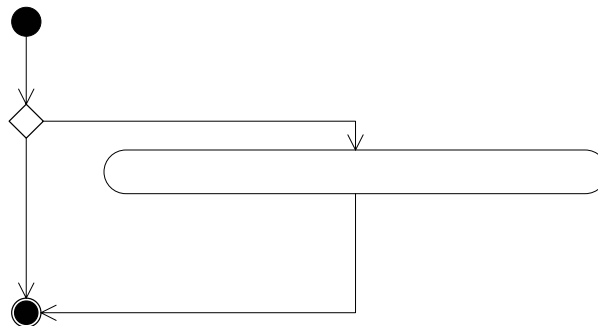


Figura 5. Diagrama de Actividades del proceso "Préstamos"

Diagramas de actividades de horarios

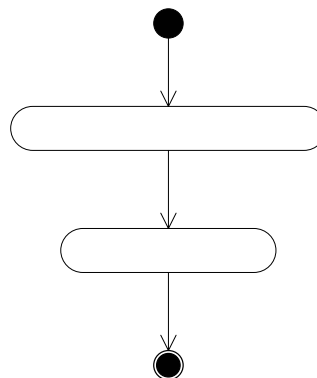


Figura 6. Diagrama de Actividades del proceso "Horarios"

Una vez observados y analizados los diagramas de actividades y los de responsabilidades se obtiene un panorama general del caso por estudiar, además de que ello ayudará a detectar algunos de los problemas que se desprenden de la forma en que realizan su trabajo los encargados.

Problemática

Cualitativamente el servicio que presta el centro de cómputo a los alumnos del plantel es eficiente ya que como se mencionó, el Centro logra su objetivo. Sin embargo, se presentan los siguientes inconvenientes en los que podemos observar un desgaste innecesario de algunos recursos:

- Cuando el alumno o externo pide un servicio, si es que ya ha visitado el centro de cómputo, es difícil recordar a los que han sido sancionados, ya que generalmente sólo residen en la mente del encargado o algunas veces los puede registrar en un papel. No obstante, hace falta un procedimiento de registro de sanciones.
- Al ubicar el equipo con el servicio solicitado o similar, a veces no se sabe con exactitud en cuáles lo podemos encontrar, ya que a veces no todos tienen los mismos servicios por no contar con las mismas características. Es decir, no se cuenta con una relación de los servicios o si se cuenta con ella no está actualizada.
- Como el Centro consta de varias salas, es difícil encontrar a simple vista qué equipos están libres para buscar en ellos el servicio. Siendo precisamente ese el problema: no se conoce exactamente cuantas máquinas están libres para servicio.
- Otro de los problemas que se desprende al ubicar el servicio en los equipos es que a veces por descuido y por no contar con una relación, se busca en los que tienen fallas y por lo tanto se asignan en mal estado.
- Debido a que no se tiene un buen control, en cuanto al manejo de las credenciales y a que algunos alumnos o externos no dejan identificación no se sabe si el equipo ya ha sido asignado.
- El registro de ingreso y egreso de los alumnos o externos al Centro se hace de forma manual en una libreta o en una hoja que se imprime a diario con la fecha del día, ocasionando que el registro sea muy lento y un gasto innecesario de papel.
- Otro inconveniente en el registro de los alumnos es que en ocasiones, no se revisa que los datos sean verídicos o estén escritos correctamente (por no contar con una identificación) o que haya omisión de datos, y que no tengan abreviaciones, etc.
- Debido a las diferentes caligrafías cuando se necesita obtener estadísticas de los registros de uso de equipo algunos datos no son legibles, obteniendo así, resultados no fidedignos.

- En ocasiones no se registra a profesores o externos, debido a que, a veces los profesores se ofenden al solicitarles identificación o no la traen. Además, ellos cuentan con un Centro en donde se les brinda servicio exclusivo.
- Una vez que se registra el alumno o externo la asignación de equipos también es manual y se ha venido haciendo sin un método bien establecido, acarreado que el tiempo sea mayor y pueda aumentar si se asigna un equipo equivocado, por ejemplo equipo ya prestado, con fallas, etc.
- Cuando se desocupa una máquina suele suceder que se sigue considerando como prestada, ya que como se mencionó hay alumnos o externos que no dejan credencial y no hay una forma de distinguir con certeza un equipo ocupado de un desocupado.
- No se sabe exactamente cuanto tiempo lleva un usuario en un equipo. Esto se debe a que el usuario puede registrar su hora de entrada tomando la hora de un reloj y la de salida con otro. Cuando se requiere realizar estimaciones de tiempo de uso de equipo de los alumnos los datos cuentan con un rango elevado de incertidumbre.
- El registro de los usuarios al estar hecho sólo una vez y en papel si le llegase a ocurrir un accidente no habría forma de recuperarlo.
- Cada vez que se pide reporte de inventarios, se pierde tiempo al sacar números y demás características necesarias manualmente que posteriormente son capturados a un formato especial para su entrega –si es que lo hay-, incrementando aún más el tiempo de entrega al no haber uno, lo que trae como consecuencia, a veces, la suspensión del servicio, por el tiempo que se lleva en realizarlo.
- Para prestar un equipo a otro departamento dentro de la escuela no se cuenta con un formato específico y se pierde tiempo al realizarlo o suele olvidarse a qué personas se prestó.
- No se sabe quién hace el movimiento de equipo de una sala a otra debido a que hay varios encargados en el transcurso del día, acarreado, muchas veces, pérdida de equipo.
- Cuando se pierde un dispositivo o se daña es difícil recordar quién ocupó el equipo un determinado día.
- No se lleva un registro de cuántas hojas imprime un usuario en un determinado periodo, por lo que el servicio ni es equitativo ni es viable en términos económicos.

Otro problema independiente de la forma en que llevan a cabo las actividades los técnicos, es que a lo largo del ciclo escolar, se presentan eventos que requieren de todo el equipo disponible de un Centro, como, por ejemplo, la Evaluación de Conocimientos de los Alumnos de Nuevo Ingreso, y, principalmente, el Instrumento de Apoyo a la Superación Académica (IASA), antes Cuestionario de Actividades Docentes, etc., dando como resultado que los

alumnos vayan a otro Centro, saturándolo y por ende, crezca la necesidad de dar más servicio con la misma cantidad de equipos y de personas que lo operan, que por lo general es un encargado por turno.

Algunas alternativas para su solución

Todos los problemas anteriormente descritos implican pérdida de tiempo, el cual podría ser utilizado en diversas tareas relacionadas con el centro, además de pérdidas económicas. Pensemos tan sólo en cuánto ahorraría la preparatoria anualmente en el mantenimiento a equipos, que se pueden evitar mediante un nuevo y mejor control, y cuánto, en gastos excesivos de materiales –como el papel-; pudiendo redistribuirlos en otros ámbitos de la comunidad universitaria, además de brindar un servicio más equitativo, rápido, eficaz y sistemático, superando con ello los problemas antes citados.

A continuación se presentan ciertas alternativas que se tienen para obtener alguna solución.

Alternativa 1: Comprar un software que automatice los procesos

Una de las soluciones para llevar a cabo, es comprar un software que ayude a realizar tales tareas de una manera más rápida y sencilla.

Para esto se buscó software relacionado con la administración de centros de cómputo, sin embargo son muy pocos los que hay en México y son orientados al uso de Cafés Internet cuyo objetivo es el control de recursos y cobro de horas.

Algunos de los principales software se mencionan a continuación:

Arroba.

Es un software del grupo Execom, el cual es capaz de controlar un Cibercafé y es compatible con MaxiComercio, un programa que se usa como servidor y permite la facturación, compras, control de bancos y manejo de almacenes.

Una de las desventajas que se puede apreciar es que este software no funciona en un ambiente donde no se cuenta con una red, y en el caso a estudiar sólo algunas están conectadas en una Intranet. Otra de las desventajas, es que este paquete viene con dos productos, los cuales se venden por separado. Esto causa el incremento de su costo y en caso de comprarse no se usarían por completo y sólo ocuparían más espacio en el disco duro, lo que trae como consecuencia sólo gastos innecesarios de los recursos con los que cuenta el centro de cómputo.

Ádroba.

Es una versión reciente de Arroba formado por cuatro aplicaciones principales, que son:

1. Un servidor que garantiza la estabilidad y comunicaciones entre los equipos.
2. Small BackOffice que es compatible con MaxiComercio, permite generar reportes y administra el comportamiento administrativo del Café Internet, es decir, es básicamente un módulo de comunicación entre Ádroba y MaxiComercio.
3. Un cliente, que da seguridad al equipo por medio de perfiles, ya que bloquea el acceso al equipo, es decir, no permite el uso del equipo sin una contraseña de administrador, además de que restringe el acceso al registro de Windows y a programas críticos.
4. Punto de venta que permite la realización de todo lo relacionado con el dinero: cobro de cuentas, ventas, ingresos/egresos de efectivo, cortes de caja, etc. Y se pueden realizar desde cualquier equipo conectado a la red, contando con los permisos adecuados.

Cuenta con otro software de tarjetas de prepago y promociones asociadas, los usuarios pueden acceder a tarifas preferenciales, entre otras prestaciones. También se pueden establecer combinaciones de tarifas; lleva un registro de trabajos de impresión.

Ádroba es compatible con MaxiComercio, ya que opera en conjunto y es opcional.

El precio de la versión estándar es de \$900 y \$2,500 para la edición Premium que incluye MaxiComercio.

Los inconvenientes que presenta este software es que trabaja en red, que se tiene que instalar en un servidor y, además, en el cliente; pero si, por alguna razón, el lugar se queda sin red, el sistema ya no funcionará. Por lo que no es una buena opción, ya que se cuenta con una red local pequeña, que no abarca todos los equipos o clientes.

Al trabajar con perfiles restrictivos, es como trabajar con los perfiles que también ofrecen los Sistemas Operativos NT/2000/XP/2003, por lo tanto si se desean implementar no es necesario comprarlo, por que Windows por si sólo ofrece más ventajas de las que puede brindar este programa.

Además de que la parte de prepago probablemente se pueda usar como control de acceso de usuarios, pero las partes de cobro en este caso no son útiles.

Cyber Admin Control

Este software ayuda a la administración de cobros de tiempo, ventas, control de scanner, impresiones, quemadores de CD, la renta de equipos desconectados, como equipos con Linux y consolas como Play Station, además de restringir el acceso a equipos, descargas de archivos y controlar estadísticas de clientes.

Lleva un registro de las operaciones realizadas en el Café Internet gracias a un servidor Web. Es decir, los equipos cliente le informan al servidor si están en línea, bloqueadas, su tiempo de uso, qué artículos, consumibles y servicios, ha comprado, mostrando cuánto debe el usuario en un ticket.

También permite agregar los empleados que uno desee con su propia contraseña para una mejor seguridad en el control del negocio.

Genera reportes por rango de fechas, de horas, empleados, productos y por equipo, además de poderse exportar a Excel.

El programa tiene un costo de \$3,000 si se descarga de Internet que incluye soporte técnico vía correo electrónico o por teléfono, y si se adquiere el CD que contiene la documentación sobre la instalación, tips, forma de uso y soporte ilimitado vía correo electrónico, se obtienen por el precio de \$3,500.

Este software puede ser útil en este caso, sin embargo la parte de cobros no sería utilizada, la parte de reportes no cubre totalmente las tareas llevadas a cabo en el Centro, además de que el gran inconveniente es que todos los equipos deben contar con red.

Este programa no tiene control de usuarios, que es algo muy importante para el centro de cómputo, por lo que no ayudaría a resolver el mayor problema que se tiene.

RentPC

Este software es otra alternativa, cuyas funciones principales son la de no permitir el uso del equipo sin autorización.

Este programa no requiere conexión de red, ya que trabaja de forma local, es decir, que el número de equipos dentro del establecimiento será el número de veces que se tenga que instalar dicho software.

Tiene dos tipos de usuarios: uno que es el administrador y tienen un control ilimitado del software y otro que es el encargado con tareas limitadas.

El programa puede iniciarse manual o automáticamente y, para usar un equipo, es necesario que se introduzca la contraseña de un encargado. Al iniciar el uso del equipo se da la oportunidad de lanzar una aplicación, como el navegador de Internet o paquetería, para reducir el tiempo de espera al consumidor del servicio.

Una vez iniciado el uso del equipo, el usuario podrá ver qué tiempo lleva y cuánto le falta para cubrir una hora, y antes de llegar a este tiempo se lanza una alarma, además de saber la cantidad de dinero a pagar.

Después se da clic en terminar, y en limpiar para borrar los datos y que esté preparado para la siguiente renta. Cada computadora guarda el registro de las rentas y puede ser manipulado por un administrador.

RentPC tiene un precio de \$960 y no depende del número de equipos en los que se desee instalar.

Este software trabaja sin conexión a red, sin embargo hay que instalarlo en cada equipo y si por algún motivo el equipo cliente falla, se podría perder información del registro, es decir, toda la información sobre los usuarios de ese equipo, además de que como se observó es un programa que sólo se dedica al manejo de dinero y en este caso no es nada útil.

Caffe-Control

Este software no sólo está orientado a Cafés Internet, sino también a billares y centros de juego.

Dentro del sistema se puede tener acceso a reportes de cortes de caja, concentrado de ventas, reportes de utilidad bruta, detalles de las rentas, rangos de rentas por cliente.

Se puede dar de alta a distintos operadores dando a cada uno privilegios.

Cuenta con un sistema de ventas para otros productos y servicios, como botanas, disquetes, impresiones, etc. El cliente puede notificar al encargado que necesita ayuda e, incluso mediante un sonido, llamarlo.

El programa también contabiliza la renta de dispositivos que no se conectan a la red, como consolas de juegos y mesas de billar.

Su costo en versión Web es de \$749, la básica viene con un CD y tiene un costo de \$999, la versión PRO ofrece un sistema de prepago en el que el cliente puede consultar tiempo usado, tiempo restante, hora de inicio y termino, cuesta \$1,499, la versión completa de \$2,499 viene con facturación de productos, como impresiones, escaneos, etc.

El problema con este programa es que la mayoría de los módulos serían desperdiciados, por lo que cubre muy pocas de las tareas que se desean mejorar, como el control de usuarios.

Concluyendo que la alternativa de comprar un software no es viable, ya que ninguno de los programas arriba mencionados cubre totalmente con las tareas que se tratan de automatizar. Aún cuando el costo sea relativamente bajo, se tendrían que comprar varios de ellos para cubrir con la mayoría de las tareas, dando como consecuencia que el espacio que ocuparían en disco duro sea elevado, además de que para que los técnicos lo manejen adecuadamente se deben familiarizar con el manejo de todos y debido a que los programas no manejan interfaces similares, la curva de aprendizaje sería mayor y causaría mucho tedio en ellos, llegando seguramente a un nivel elevado de rechazo.

Alternativa 2: Que el centro de cómputo siga funcionando igual tratando de subsanar los problemas que existen, sin automatizar.

Otra de las alternativas, es la de que el Centro siga funcionando en algunos aspectos de igual forma y en los que se tiene un control aunque sea manual, tratar de mejorarlo.

Algunas de las ventajas que se pueden mencionar son:

- No se invertirían muchos recursos, ni económicos, ni humanos y mucho menos materiales.
- No se pierde tiempo en entrenamiento para el uso de algún software.

Algunas de las desventajas que trae consigo, son:

- Puede ocurrir que se llegue al mismo descontrol, en caso de que un técnico falte y se ponga un sustituto.
- El avance general de mejora y de la calidad de servicios es imperceptible.
- El tiempo para mejorar y realizar estos procesos aumentarían, llegando probablemente a la desidia.
- Si las adecuaciones son mal implementadas pueden llegar a empeorar la situación del centro, dando como resultado que los estudiantes universitarios al ver un servicio deficiente probablemente abandonen el Centro y se desperdicie todo ese equipo.

Por lo tanto esta otra alternativa tiene más desventajas que la de comprar un software.

Alternativa 3: Solo automatizar algunos procesos.

Otra alternativa es que sólo se automaticen algunos procesos que se consideren los más importantes.

Esto implicaría pérdida de tiempo en la programación ya que posiblemente cada técnico en turno maneje o tenga muy bien controlado el registro de usuarios, otro el asunto de los inventarios, etc., dando como consecuencia que cada Técnico piense que el proceso que domina es el más importante.

Por diferencia de criterios puede llegarse nuevamente al descontrol, ya que si un técnico piensa que un proceso no es importante no lo va a usar, provocando un desperdicio de tiempo de programación, además de que si ya se programaron unos procesos, sería mejor programar la mayoría para cubrir más de las tareas administrativas.

Por tales razones esta alternativa sugiere la realización de un software que cubra casi al 100% con las necesidades del centro de cómputo.

Alternativa 4: Hacer un software

Como podemos observar, las alternativas anteriores presentan algunos inconvenientes. El hecho de tratar de automatizar la mayoría de las tareas posibles dentro del Centro es muy favorecedor, ya que traerá varias ventajas, como son:

- En cuanto a los beneficios monetarios, estos se hacen indirectamente a la UNAM ya que se pueden redireccionar a cubrir otras necesidades que presente el centro de cómputo, un ejemplo es: al tener más tiempo él o los encargados, lo podrán invertir en un mejor mantenimiento de los dispositivos y, por ende, no se tendrá que invertir para reponer tales componentes, sino que, esos recursos se podrán utilizar para comprar otras cosas e implicará más equipo a disposición de los alumnos.
- La administración sería centralizada debido a que sólo se instalaría en un equipo o, en su defecto, en un número igual al número de técnicos, ya que cada quien hará los reportes correspondientes a su turno, además de que el software proporcionaría un estándar de reportes, registro de alumnos, etc.
- La curva de aprendizaje³ sería mínima a las alternativas anteriores, debido a la experiencia y conocimientos de los técnicos, además de que sólo tendrían que aprender a usar un producto para hacer la mayoría de las tareas.
- No se necesitaría forzosamente un equipo con Sistema Operativo Windows de servidor, es decir se puede instalar en un Sistema Operativo Windows de cliente.
- No se necesitaría más infraestructura de la que se tiene, es decir, el sistema se hará con base a los recursos con los que cuenta el Centro, considerando que éste no cuenta con red para todos los equipos y además no se espera contratar a más técnicos para su manejo.
- El tiempo de espera de los usuarios para ser atendidos disminuiría considerablemente, debido a que el registro ya no sería manual. Se pretende que el sistema cuente con un almacén de datos con los alumnos inscritos en la preparatoria, y con ello su registro sea mucho más rápido al no tener que capturar sus datos, sólo será búsqueda, comprobación y asignación.
- Los reportes llevarían menos tiempo de realización y estarían al día debido a que los datos se irán almacenando en el sistema. Cuando estos, se necesiten sólo bastará realizar una búsqueda de los datos requeridos, verificar e imprimirlos, además de que los formatos ya estarán integrados en el software.
- Se sabría qué servicios y hardware se tiene, lo que ayudaría a la optimización de los recursos y un mejor control del inventario, pues los datos se irán almacenando en el sistema, por lo que estarán disponibles en cualquier instante ya sea para su consulta, reporte o cualquier situación que requiera de dichos datos.
- Los técnicos tendrían más tiempo para realizar otras acciones que favorezcan el buen funcionamiento del Centro y principalmente para auxiliar a los alumnos, profesores y usuarios en general, en sus dudas y peticiones de material y equipo.

³ La curva de aprendizaje es el tiempo que tarda un usuario en aprender a usar un software.

- Aun cuando el costo de los productos anteriormente mencionados es relativamente bajo, adquirirlos implicaría un gasto adicional, mientras que para el desarrollo del software no hay tal, ya que:
 1. El costo de programación. No hay costo monetario de programación, pues las personas involucradas para su desarrollo serían los mismos técnicos o administradores del lugar, por lo tanto no se invertirá en su desarrollo, ya que se haría en el mismo horario de trabajo aprovechando los tiempos muertos y sólo recibirán el sueldo acostumbrado.
 2. El costo de las herramientas para la programación es nulo, pues los equipos del centro de cómputo cuentan con Licencias del Sistema Operativo Windows, Microsoft Office 2003, Visual Studio .Net, ya que son las utilizadas por los técnicos cuando se requiere de pequeños sistemas para la Escuela Nacional Preparatoria 9.

Además de que el desarrollo de un sistema tratará de cubrir casi al 100% las necesidades del centro de cómputo y, principalmente como ya se mencionó, se tratará de aprovechar al máximo los recursos económicos, materiales y humanos con los que se cuenta.

Hay que hacer énfasis aquí, en que uno de los recursos con los que cuenta el centro de cómputo y que tiene un peso muy importante en la realización de un software, es como ya se mencionó, que los equipos trabajan con sistemas operativos Windows y por lo tanto el sistema a desarrollar tendrá que ejecutarse bajo tal plataforma.

Considerando las razones anteriormente expuestas, además de que los desarrolladores están íntimamente ligados a la problemática y serán los futuros usuarios finales, la opción más factible es la realización de un software.

2

2. Perfil tecnológico

Al tomar la decisión de generar un software, lo que se espera es obtener un sistema en funcionamiento y que cumpla con las necesidades del usuario final, pero antes de ello, se debe tener muy claro cuál es el objetivo de tal y por su puesto la manera en que se llevará a cabo.

Para saber cuál es esa forma, hay se deben varios aspectos, los cuales, en conjunto constituyen el perfil tecnológico. Este comprende metodologías existentes, lenguajes de programación, gestores de bases de datos y demás herramientas para el desarrollo de software, pero lo principal en este contexto es saber con que opciones se cuenta para la realización del sistema.

Para comprender mejor tal perfil, se deben conocer ciertos conceptos, que a continuación se presentan:

Ciclo de vida del desarrollo de sistemas: Es una serie de fases que los analistas de sistemas, los ingenieros de software, los programadores y los usuarios finales, consideran para la elaboración de un proyecto. Estas fases generalmente son: El análisis, el diseño, la implementación, las pruebas y el mantenimiento.

Uno de los principales objetivos de este ciclo es el de definir las etapas a seguir para el desarrollo de un sistema, además de especificar las actividades a ser ejecutadas por cada uno de los integrantes de un grupo de desarrollo, entre otras.

Otros conceptos también importantes son:

Método: Es la forma de llevar a cabo las tareas en el desarrollo de un software.

Técnica: Es un método que aplica herramientas y reglas específicas para completar una o más fases del ciclo de vida del desarrollo de sistemas.

Herramientas: Son las aplicaciones de apoyo necesarias para automatizar las prácticas en el desarrollo de Software.

Metodología: Son diversas etapas bien definidas por algún autor, en donde están bien establecidas las reglas, métodos, herramientas y productos resultantes para el desarrollo de un software.

A continuación se mencionarán algunas herramientas y metodologías que ayudan en el desarrollo de software.

Metodologías para el desarrollo de software

Anteriormente las metodologías para la elaboración de software eran muy rigurosas, pero se suceden que se desarrollan otras mejores, que tratan de reparar errores –si es que los hay- y traen más opciones de trabajo.

Para algunos crear un software suena muy fácil, sin embargo ¿Cuántas de esas personas tienen un método bien establecido para hacerlo?

En seguida se resumirán algunas metodologías, con la finalidad de conocer sus características, para así poder decidir sobre cual trabajar.

Metodología tradicional o en cascada

Inicialmente en el desarrollo de software era muy común ver trabajos que se basaban en el llamado método tradicional o en cascada, aunque hay que destacar que no porque sean de antaño quiera decir que ya no sean útiles.

En esta metodología las etapas se dan en lapsos de tiempo bien definidos, siempre una después de otra, es decir hasta cuando se complete una se podrá continuar con la siguiente. Las etapas que tiene son: el análisis, el diseño, la codificación y por último la distribución.

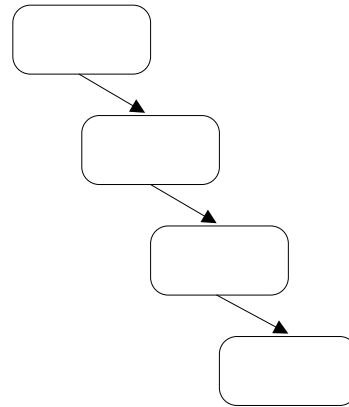


Figura 7. Método en cascada o tradicional del desarrollo de software

La forma de realizar un proyecto de la manera como se muestra en la figura 7, traía o trae varias desventajas para los que todavía la utilizan, como son las siguientes:

- Provoca que las tareas se realicen de manera individual, es decir, ni el analista tiene comunicación con el diseñador, o el diseñador con el programador, etc., aunque no hay que descartar la posibilidad de existir una muy buena comunicación entre ellos, esto dependerá del grupo de desarrolladores.
- Reduce la comprensión del problema, ya que no hay que olvidar que la comprensión evoluciona al mismo tiempo que el sistema.
- Los resultados de cada etapa en ocasiones suelen no utilizarse completamente o como debieran, un ejemplo de ello es si por alguna razón un diseñador encuentra errores en el análisis al no haber comunicación entre ellos o no pudiese darse después de terminada una fase, suele pasar, que no hay forma de estar en contacto nuevamente con él, y al querer evitarse retardos, reparaciones o peor aún, rehacer el trabajo de la etapa anterior, la salida más viable, es continuar con los resultados que se tienen, con errores, provocando que el sistema a desarrollar tenga elevadas posibilidades de fracaso.
- Puede ocurrir que el líder o responsable del proyecto equivocadamente piense que la programación lleva la mayor parte del tiempo y lo peor es que se lo asigna, siendo que en realidad, la parte más complicada de este proceso es el análisis seguido del diseño, pues teniendo muy buenos resultados en estas etapas la codificación es sencilla y rápida.

Metodologías recientes

En contraste con el método en cascada, actualmente se tiende al desarrollo de software con una amplia colaboración entre las fases y no sólo eso, sino que además con el usuario final, por ejemplo, los analistas y diseñadores, se reúnen con el programador, para reafirmar sus ideas, y los programadores a su vez interactúan con los analistas y diseñadores para compartir sus puntos de vista, pero todos tienen contacto con el usuario final, probablemente unos más

que otros como los analistas y diseñadores. Estas relaciones traen buenas consecuencias, la principal es que con ello se logra un sistema fiable que cubre las necesidades del usuario final, por ende es menos propenso al abandono.

Es por ello que a continuación se resumirán ciertas características de algunas de ellas.

Otras metodologías

Existen otras metodologías que sólo apoyan al desarrollo de proyectos, y entre ellas se pueden mencionar:

Metodología Delphi

Esta metodología, permite realizar un análisis prospectivo con ayuda de un grupo de expertos o personas seleccionadas que son consultados por medio de un cuestionario estructurado, coordinado y analizado por un grupo asesor y líderes del proyecto.

En la realización de este proceso a los participantes se les consulta por correo postal o electrónico y su identidad es mantenida en anonimato por los líderes del proyecto, hasta terminar con las consultas.

Este tipo de comunicación evita que los consultados tengan cualquier tipo de influencia por parte de los demás participantes, además de no inhibirse por la imposición de criterios de los entrevistados más concedores.

Por medio de las respuestas provenientes de los expertos o personas seleccionadas, se trata de llegar a un acuerdo de opiniones, para esto, se toman los resultados y a partir de ellos se obtienen diferencias, tendencias, deseos y expectativas. Para llegar a dicho acuerdo primero se define un perfil y las respuestas serán consideradas, siempre y cuando sean relevantes, todo esto para lograr una reducción del grado de error derivado de las predicciones.

Debido a que en este método se realizan varias rondas, al final de cada una de ellas, los resultados se analizan y se regresan a cada participante en la ronda siguiente, con la finalidad de se vea la opinión personal comparada con la opinión del grupo y tenga la oportunidad de reflexionar sus respuestas para consolidar un acuerdo.

El acuerdo se obtiene mediante un procedimiento de valoración de respuestas, medidas de agregación y eliminación de respuestas extremas, para su realización se deben establecer las reglas desde el inicio.

El número de rondas va a depender de las necesidades de los líderes del proyecto, es decir, cuando se desea llegar al consenso se recomienda hacer tantas rondas como sea necesario hasta llegar al acuerdo, pero si no se alcanza, entonces se debe interrumpir hasta alcanzar una estabilidad estadística en las respuestas.

Características de la consulta

El diseño de las consultas y la elección de las personas elegidas o expertos que participarán, son seleccionados por los líderes del proyecto en conjunto con el grupo asesor.

Debido a la comunicación que existe entre el grupo asesor y líderes del proyecto, se generan las siguientes ventajas:

- Los cuestionarios cumplen los objetivos del proyecto.
- Los cuestionarios son entendibles y aseguran neutralidad, evitando que algunas respuestas tengan más peso que otras, es decir, evitar que los líderes del proyecto manipulen los resultados.
- De igual manera se evita la manipulación en la elección de las personas que participarán en las rondas.

Aunque estos valores no se apoyan en argumentos decisivos, se sugiere que las dimensiones idóneas sean entre 7 y 30 participantes.

Para una mejor interpretación y representación gráfica de los resultados de cada ronda, las respuestas se agrupan del siguiente modo:

Unanimidad. Cuando todos los participantes efectúan una opinión idéntica.

Consenso. Cuando al menos las 9/10 partes de los participantes realizan la misma predicción sin alcanzar la unanimidad.

Mayoría. Cuando la misma predicción es realizada por una amplia mayoría de los participantes; esto es, entre más de 2/3 y menos de 9/10 partes.

Discordancia o discrepancia se considera para el resto de situaciones posibles; es decir, cuando una misma respuesta no es compartida por, al menos, las 2/3 partes de los participantes.

Metodología eXtrema

La eXtreme Programming (Programación extrema), es una metodología surgida de la experiencia en el desarrollo de software de sus autores.

Es única y compacta, aunque no se base en principios nuevos, es una nueva forma de desarrollar software.

En el proceso de desarrollo extremo, el cliente se encuentra mucho más cercano, puede cambiar de opinión sobre la marcha bajo la condición de estar siempre a disposición del equipo de desarrollo para resolver las dudas que vayan surgiendo.

El documento de los requerimientos no se hace inicialmente, pues se van definiendo durante el desarrollo del proyecto y son redactados por el propio cliente en tarjetas, es decir, nadie del equipo de desarrollo lo realiza.

En el lugar de la documentación de requerimientos, están las historias de usuarios que constan de un listado de las características que se desea que el sistema tenga finalmente y consta de dos fases:

- a) En la primera, el cliente describe con sus propias palabras las características del sistema y el responsable de la codificación le informa la dificultad técnica de cada una, su costo y le asigna alguna duración. Al final el cliente las ordena por prioridades.
- b) En la segunda fase se eligen cada una de las historias que se implementarán (primera iteración) y se dividen en tareas para su realización.

El proceso de creación de estas historias no es el mismo que el de los casos de uso de UML (Lenguaje Unificado de desarrollo) aunque sean muy parecidos.

El tiempo de desarrollo para cada historia de los usuarios deberá estar comprendida entre una y tres semanas idealmente (cinco días de cada semana). Cuando una historia es de duración menor, se reúnen varias para cubrir tal tiempo, pero en caso de que estas rebasen el tiempo, entonces deberán ser divididas.

En dichos tiempos establecidos, se toma un tiempo de holgura⁴ para ajustar el tiempo ideal al real, que son tomados de la experiencia de proyectos anteriores.

Planificación

En este punto se planifican las entregas, cada entrega es resultado de una iteración donde todo el equipo de desarrollo participó, y es una nueva versión para el cliente. En estas iteraciones es conveniente, introducir las partes esenciales del sistema, y entregarlas a tiempo, ya que esto hará que se tenga más tiempo para trabajar con ellas, por lo tanto es recomendable hacer entregas muy frecuentes, ya que ayudarán a encontrar más rápido los errores y el tiempo de reparación se reducirá.

Para después de cada tres o cinco iteraciones es recomendable volver a revisar las historias de los usuarios y renegociar los tiempos.

Hay que planificar también cada una de las iteraciones, esto es lo que se conoce como planificación iterativa. En esta planificación se consideran las historias primordiales y las que no han sido aceptadas en las iteraciones pasadas.

Como se puede apreciar el diseño que se sigue es continuo, es ágil, no se mira demasiado adelante y tampoco se implementan tareas no programadas, a esto se le conoce también como programación just-in-time (programación en el instante).

⁴ Espacio de tiempo suficiente para que en caso de ser necesario se termine una tarea. Dicho de otra forma es un tiempo que se deja en una fase de un proyecto para solucionar imprevistos.

Una recomendación a la hora del desarrollo es concentrarse en el proyecto con los requisitos necesarios, hacerlo bien, y al último poner los detalles que se consideren.

Esta forma de realizar un software hace que exista mucha comunicación entre los participantes (que es una característica de la XP), y a la vez por su carácter informal se hace agradable.

Diseño, desarrollo y pruebas

En la programación extrema todas las tareas se tratan de desarrollar de manera rápida, sin interrupciones y con la finalidad correcta.

El diseño debe ser lo más simple posible y se establece que todas las tareas deben ser revisadas y mejoradas de manera continua, según se vayan añadiendo funcionalidades al sistema durante todo su desarrollo.

El término metáfora es muy importante en la programación extrema, el objetivo es mejorar la comunicación entre los integrantes del equipo de desarrollo al crear una visión global y común del sistema que se pretende desarrollar. La metáfora debe ser expresada en términos sencillos y conocidos por todo el grupo de desarrolladores. Además de incluir las principales clases y patrones que se usarán en el sistema.

Al realizar la metáfora, el mejor apoyo es el conocimiento de los nombres elegidos durante el proyecto para los módulos, sistemas, clases, métodos, etc., ya que si se realiza una buena elección, hará que tales módulos sean claros, reusables y simples, tres conceptos de gran importancia para la programación extrema.

Para el diseño de las partes más complejas, se llevan a cabo reuniones donde participan los desarrolladores, aquellos que tienen más experiencia y los clientes. En estas reuniones se usan un tipo de tarjetas llamadas CRC (Class, Responsibilities and Collaboration - Clases, Responsabilidades y Colaboración) que tienen como objetivo facilitar la comunicación y documentar los resultados. Para cada una de las clases identificadas, se utiliza una tarjeta de estas, en ella se especifica su finalidad y las interacciones que tienen con otras clases. Estas tarjetas bien elaboradas suelen inducir a un diseño más obvio. Estas sesiones son una simulación de las interacciones que se están trabajando.

Otro término clave e innovador en la programación extrema, son las pruebas unitarias, que son pruebas sencillas que se realizan sobre un pequeño segmento de código, suficiente para que se verifique que se cumplan los requerimientos del cliente. Estas se repiten cada vez que se pasa la prueba.

Realizar estas pruebas puede resultar engorroso, sin embargo, la ventaja es que se hacen una vez y luego se pueden reutilizar muchas veces. Todas las acciones anteriores fomentan la compartición de código, una de las prácticas en la XP que permiten acelerar el desarrollo.

Otro de sus beneficios, es que ayuda a dar prioridades, comprueba la evolución del desarrollo, ofrece realimentación inmediata, se logra un código

simple y funcional de manera rápida, y ya no es forzoso contar con equipos independientes en el desarrollo.

Cuando se note complejidad en la implementación de las pruebas, se deben considerar aún más necesarias, incluso, es muy probable que lo que se intenta probar no es lo suficientemente sencillo y entonces deba rediseñarse.

Estas pruebas no se deben confundir con las de aceptación antes mencionada, ya que estas últimas son realizadas por el cliente.

Otro concepto importante es la integración. En la programación extrema se le denomina integración continua, es decir, se va realizando con pequeños fragmentos de código, evitando la integración final. Las ventajas de este enfoque es que permite la realización de pruebas completas, la pronta detección de problemas de incompatibilidad, y evita la participación necesaria de un equipo independiente de integración.

En el desarrollo de programación extrema deberá existir una versión siempre integrada. La sincronización y comunicación entre los desarrolladores hará que las modificaciones necesarias se realicen sobre la última versión, asegurando que no se trabaje sobre una versión vieja (más antigua que el código actual).

Encontrarse con un código obsoleto es muy probable, sin embargo la programación extrema utiliza la refactorización para evitar esto. Refactorizar consiste básicamente en quitar redundancia, eliminar funcionalidad que no se usa o fortalecer diseños viejos, incluso es usada en las pruebas unitarias.

Al refactorizar, se debe contemplar el tipo de código que se tiene, he ahí la importancia de que sea simple, ya que esto ayudará ahorrar tiempo cuando se lleva a cabo y aumentará la calidad del sistema.

Como se menciona la compartición de código en la programación extrema es muy importante, pues al ser de todos, cualquier desarrollador puede realizar cambios, corregir errores, refactorizar o mejorar cualquier parte del sistema en cualquier momento. Además, es una forma para que todo el equipo se haga responsable del sistema y como consecuencia cuando un desarrollador deje el proyecto –posibilidad que se presenta- no sea un hecho catastrófico.

Un importante y controvertido término en la programación extrema es la programación por parejas, ya que una pareja de desarrolladores comparten equipo, realizan de forma continua la revisión de código y diseño, y para hacerlo se van rotando.

El hecho de que dos personas trabajen en forma unida, generan probablemente la misma cantidad de código, pero de mayor calidad. Sin embargo esta es la práctica que más evasivas provoca por parte de jefes y de los propios programadores.

Valores y principios de la programación extrema

Los autores de XP afirman que para obtener diseños, códigos simples, software con éxito y sobretodo clientes contentos, se utilicen una serie de valores y principios. Son cuatro los valores en la programación extrema: comunicación, simplicidad, realimentación y coraje. Este último valor implica que cuando se llegue a tener en funcionamiento el software se mejore. Aunque algunos añaden un quinto valor que es la humildad.

Los principios fundamentales se apoyan en los valores y también son cuatro. Se busca realimentación veloz, modificaciones incrementales, trabajo de calidad y mucha simplicidad. Estos son resultado de la unión entre los valores y las prácticas.

Las prácticas de la programación extrema son las siguientes:

- El juego de la planificación (the planning game).
- Pequeñas entregas (small releases).
- Metáfora (metaphor).
- Diseño simple (simple design).
- Pruebas (testing).
- Refactorización (refactoring).
- Programación por parejas (pair programming).
- Propiedad colectiva (collective ownership).
- Integración continua (continuous integration).
- 40 horas semanales (40-hour week).
- Cliente en casa (on-site customer).
- Estándares de codificación (coding standards).

Se recomienda que estas prácticas se apliquen de dos en dos, pero esto debe ser incremental, es decir, tratar de aplicarlas todas, ya que el objetivo final es ese, pues representan un conjunto completo.

Metodología GRAPPLE

GRAPPLE (Guías para la Ingeniería para la Aplicación Rápida), es un conjunto de ideas de varias personas. Los tres grandes amigos Rumbaugh, Booch, Ivar Jacobson, que como vimos anteriormente cada uno contaba con su metodología cuyas ideas son similares a GRAPPLE, y no está por demás saber que se unieron para crear el Proceso Racional Unificado (UML).

Esta metodología maneja segmentos en el lugar de fases, para eliminar la idea de que una fase debe completarse antes de iniciar la otra. GRAPPLE tiene cinco segmentos. Donde cada una consta de diversas acciones y cada acción dará como resultado un producto de trabajo, y es responsabilidad de un integrante del grupo desarrollador.

GRAPPLE se encausa a los sistemas orientados a objetos. Por ello, las acciones dentro de cada segmento se orientan a crear resultados de naturaleza orientada a objetos.

Los segmentos son:

1. Recopilación de necesidades
2. Análisis
3. Diseño
4. Desarrollo
5. Distribución

Recopilación de necesidades

Este segmento es de primordial importancia. Se debe comenzar por conocer los procesos del dominio, en especial los que se tratarán de mejorar con el sistema. Estos se pueden obtener por medio de entrevistas que se realizan entre el analista y el cliente. Una ventaja obtenida con estas entrevistas, es que el analista se familiarizará con el vocabulario manejado por el cliente.

En este segmento se realiza un análisis del dominio, es decir, se trata de comprender al máximo el entorno del cliente, sobre el que se trabajará, tratando solo conceptos que son del sistema que se pretende generar. El producto de este segmento son los diagramas de actividades que captan los pasos y puntos decisivos en estos procesos.

Durante la entrevista lo que se debe capturar son los sustantivos y verbos que se presenten, ya que unos se convertirán en clases y otros en atributos, que servirán para obtener el producto, que es un diagrama de clases de alto nivel.

Para descubrir las necesidades del dominio, no solo basta con tales entrevistas, sino que es recomendable hacer uso de las llamadas sesiones JAD (Desarrollo Conjunto de Aplicaciones), ya que las entrevistas pueden llevar demasiado tiempo dependiendo de la complejidad del sistema. En dichas sesiones, se reúne a los clientes potenciales y a los miembros del equipo de desarrollo, donde ambos exponen sus puntos de vista en beneficio de todos. En esta sesión se puede comenzar con la refinación de clases.

El resultado de las sesiones JAD y las entrevistas, es un diagrama de paquetes, donde cada paquete agrupa un conjunto de casos de uso, que son funciones del sistema.

Otra parte importante, es la identificación de los sistemas cooperativos, es decir, analizar de qué sistemas dependerá y cuales dependerán de éste.

Cuando se finalice esta sesión, se puede esperar la aprobación del cliente o los costos, pero el resultado depende de lo que espera obtener el cliente.

Análisis

En esta parte se profundizan las necesidades del sistema aumentando la comprensión de este.

Después se prosigue con la comprensión del uso del sistema, para esto, en una sesión JAD se trabaja para identificar a los actores que inician un caso de uso y cuales son los que lo terminan, que pueden ser personas u otros sistemas.

Aquí el equipo de desarrollo intenta realizar nuevos casos de uso. El resultado, son los diagramas de casos de uso.

Después se trata de hacer realidad los casos de uso, para ello se sigue en contacto con el cliente, el objetivo es obtener la secuencia de pasos en cada caso de uso, sin embargo esta resulta la más compleja para algunos clientes, pues se les pide que enumeren o expresen paso a paso los procesos.

En esta sesión JAD, se debe continuar con la depuración de los diagramas de clases, llenando los nombres de las asociaciones, multiplicidades, generalizaciones y agregaciones.

Una asociación es la forma en que se relacionan dos clases, entre estas clases puede haber más de una asociación.

La multiplicidad es cuando varios objetos de la misma clase se relacionan con un objeto de otra clase.

La generalización es la herencia que puede existir en los casos de uso.

Una agregación es una forma de asociación en la cual un objeto esta conformado por varios objetos.

Además de analizar los cambios de estado de los objetos en el proceso, el producto de este trabajo es un diagrama de estados.

Una vez definidos los diagramas de casos de uso y de clases, se puede definir la forma en que los objetos se comunican, para obtener un diagrama de secuencias y de colaboraciones e incluir los cambios de estado.

Al tiempo de realizar los pasos anteriores, el diseñador del sistema descubre los detalles específicos de la integración con los sistemas cooperativos.

Diseño

En esta parte se trabaja con los resultados del segmento del análisis. En el análisis y el diseño se realizan las revisiones pertinentes hasta lograr que se complete el diseño, incluso algunas metodologías combinan al diseño y el análisis en una sola fase.

Los programadores tomarán los diagramas de clases y generaran los diagramas de objetos necesarios. Darán vida a los diagramas de objetos mediante el análisis de cada operación y el desarrollo de un diagrama de actividades correspondiente. Los diagramas de actividades fungirán como la base de gran parte del código en el segmento de desarrollo.

Ahora toca trabajar con los diagramas de componentes, aquí los programadores son los que juegan un papel importante. La tarea es visualizar los componentes que resultarán del siguiente segmento y mostrar las dependencias

entre ellos. Un componente en UML es una parte física del sistema que se encuentra en la computadora, este puede ser un archivo ejecutable, de datos, etc.

Una vez completados los diagramas, el diseñador del sistema empezará a planear la distribución e integración con sistemas cooperativos. Creará un diagrama de distribución que muestre el lugar donde se encontrarán los componentes.

Para el diseño de la interfaz de usuario, hay otra sesión JAD. La interfaz deberá permitir la consumación de todos los casos de uso. Para ello un analista de GUI (Interfaz grafica de usuario) deberá trabajar con los usuarios finales para desarrollar los modelos en papel, de las pantallas correspondientes a los grupos de casos de uso. Los usuarios representarán con papeletas los componentes de dicha pantalla como los botones, casillas de verificación, listas desplegables, menús, etc.

Por ultimo corresponde realizar las pruebas de diseño, en donde se evalúa si el software hace lo que se pretendía que hiciera (esto se especifica en los casos de uso).

Los especialistas en la documentación trabajan en conjunto con los diseñadores para comenzar a generar un panfleto de la documentación y llegar a una estructura de alto nivel para cada documento.

Desarrollo

Esta parte está a cargo de los programadores, que con un buen análisis y diseño, este segmento se realiza rápido y sin ningún problema.

Con los diagramas de clases, de objetos, de actividades y de componentes, los programadores generarán el código del sistema.

Las personas dedicadas a realizar las pruebas (no los desarrolladores) ejecutarán unas secuencias de comandos de prueba para evaluar si el código hace lo que se pretende. Los resultados de las pruebas son el producto de tal trabajo. Esta parte se realiza, realimentando la generación de código y se puede continuar hasta que todas las pruebas sean superadas.

Finalmente el especialista en GUI genera las interfaces de usuarios ya aprobadas y las conecta con el código. Las pruebas siguientes aseguran que las interfaces funcionen adecuadamente.

Así hay que recordar que durante el desarrollo, los especialistas en documentación trabajan en paralelo con los programadores para asegurar la entrega oportuna de toda la documentación.

Distribución

Cuando el sistema es terminado, se distribuye en el hardware adecuado y se integra con los sistemas cooperativos.

El diseñador creará un plan que incluya los pasos a seguir en caso de que el sistema falle. Además de generar una copia de seguridad.

También el diseñador se ve implicado en la tarea de distribuir el sistema en los equipos requeridos.

Finalmente el equipo de desarrollo verifica el sistema instalado, se ejecuta y se checa si las alternativas de recuperación funcionan, así dependiendo de estos resultados se ve si conviene realizar una depuración.

De los segmentos anteriores, los resultados del primer hasta el tercer segmento, conforman un documento que se puede entregar al cliente y a los desarrolladores. Una vez completados todos los segmentos el documento se convierte en uno donde se define el sistema.

GRAPPLE es una metodología hecha para trabajar con UML, ya que como se puede observar en cada resultado se espera el diseño de ciertos diagramas, y estos son de UML.

Herramienta no computacional: Lenguaje Unificado de Modelado (UML)

Antes de involucrarnos con UML, hay que señalar que algunos autores lo consideran como una metodología, y algunos otros como sólo una herramienta que ayuda a alguna metodología como es el caso de GRAPPLE y muchas otras metodologías.

Esto se basa en las siguientes ideas:

“El UML (Lenguaje Unificado de Modelado) es una de las herramientas más emocionantes en el mundo actual en el desarrollo de sistemas. Esto se debe a que permite a los creadores de sistemas generar diseños que capturen sus ideas en una forma convencional y fácil de comprender para comunicarlas a otras personas”⁵

“Para diseñar software orientado a objetos con éxito, tenemos que incorporar una metodología contrastada. Esta metodología debe contener un conjunto de normas y notaciones que permitan transformar los requisitos lógicos del sistema en el código utilizado para implementarlo. Esta metodología también debe ser coherente con los conceptos asociados a los diversos lenguajes de programación orientada a objetos (POO). Una de las metodologías de diseño más utilizadas actualmente en POO es el Lenguaje de Modelado Unificado (UML)”⁶

Este lenguaje es muy conocido pues unifica los métodos de Grady Booch, James Rumbaugh OMT (Object Modeling Technique) y Ivar Jacobson. Además de ser una herramienta que **no requiere de un equipo de cómputo** para su uso y

⁵ Aprendiendo UML en 24 Horas, Joseph Schmuller, Ed. Pearson Educación, Pág. 5.

⁶ Introducción orientada a objetos con Visual Basic .Net, Dan Clark, Ed. ANAYA Multimedia, Pág. 39.

los conceptos que utiliza son soportados por muchas metodologías y herramientas computacionales actuales, de las que se mencionaron.

UML se caracteriza por ser un **lenguaje de modelado** para especificar, construir, visualizar, documentar los elementos de un sistema de software y sobre todo fácil de entender, cuya notación se ha convertido en un estándar en el mundo de los desarrolladores de sistemas.

Hay que tener muy claro que solo es una herramienta de modelado, es independiente de los lenguajes de programación y de la metodología de desarrollo, soporta Programación Orientada a Objetos (POO) y reutilización.

A continuación se describirán brevemente algunos de los diagramas del UML. De los más representativos dentro de las fases del ciclo de vida de un proyecto.

Como primer diagrama se tiene el diagrama de actividades muestra los pasos y puntos de decisión dentro del comportamiento de un proceso. En la siguiente figura (Figura 8) se puede observar la representación general. Estos diagramas permiten identificar a las personas involucradas en el dominio a analizar con sus respectivos procesos.

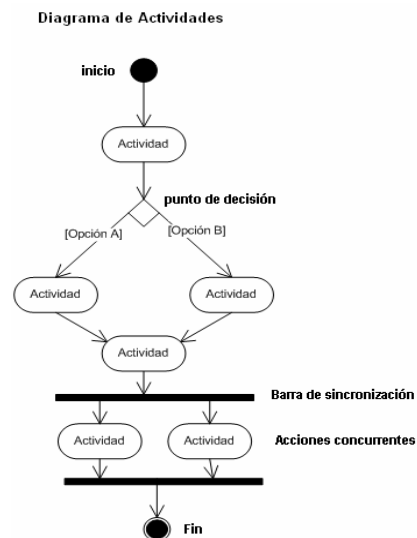


Figura 8. Vista general de un diagrama de actividades

Las clases, también son una parte importante dentro del desarrollo orientado a objetos, ya que muestran las entidades de un sistema o dominio y la forma en que se relacionan entre si. En UML cada clase se representa con un rectángulo con su nombre, como se puede apreciar en la siguiente figura.

Clase

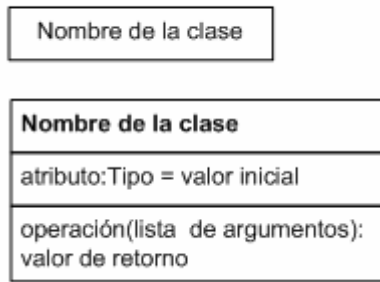


Figura 9. Representación de una clase en UML

UML trabaja con diferentes tipos de asociaciones entre sus clases, estas son las tripartitas y/o múltiples. Una multiplicidad es la relación de muchas clases entre sí, en las tablas 1 y 2, se puede observar la forma en que se representan en UML y la forma de interpretar su simbología.

Relación	Se lee:
1 a 1	Relación de uno a uno
1 a *	Relación de uno a muchos
1 a 1..*	Relación de uno a 1 o muchos
1 a 0,1	Relación de uno a 0 o a uno

Tabla1. Tipo de relaciones en UML

Símbolo	Se lee:
*	Más o muchos
,	ó
..	ó

Tabla2. Simbología de relaciones en UML

Una asociación tripartita indica que hay tres clases asociadas. Este tipo de asociación se modela con un rombo en UML.

Un ejemplo, se observa cuando un usuario puede necesitar más de un servicio y utilizar el mismo equipo, esto en un modelo, luciría de la siguiente manera:

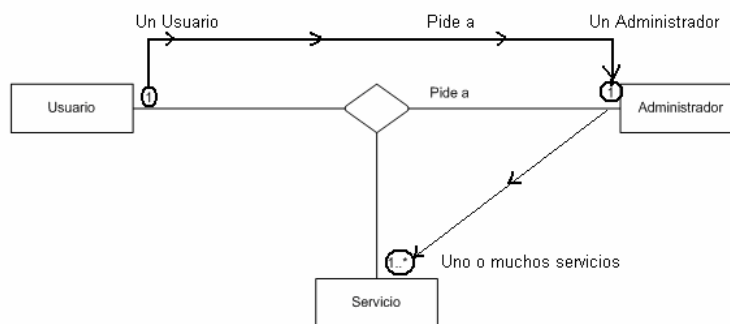


Figura 10. Ejemplo de una relación tripartita con asociaciones múltiples.

En este tipo de asociaciones, las multiplicidades indican cuantos objetos de dos clases están involucradas y una tercera permanece constante.

Otro punto importante dentro del análisis o software a desarrollar son los casos de uso pues muestran la secuencia de pasos para lograr determinada actividad dentro del dominio y servirán a los diagramas de secuencias. Cada caso de uso aparece como una elipse y cada actor como una figura representativa.

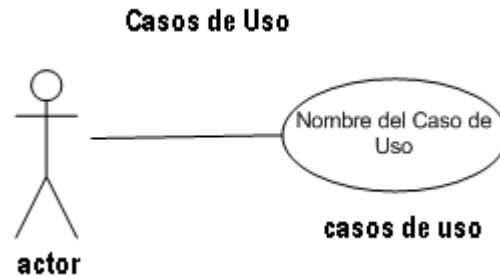


Figura 11. Representación de los casos de uso en UML

Los casos de uso, como se mencionó son de gran ayuda para el desarrollo de los diagramas de secuencias, ya que representan la forma en que interaccionan los objetos entre sí, al paso del tiempo. Los objetos se colocan en la parte superior y el tiempo procede desde la parte superior hacia la inferior del diagrama. Las flechas denotan mensajes que van de objeto a objeto. Este tipo de diagramas ayudan a la codificación y a la creación de las interfaces del futuro sistema.

En los casos de uso se pueden presentar dos tipos de relaciones, una de ellas es la inclusión (incluye o usa) y la otra es la extensión (amplía o extiende).

Se dice que hay una inclusión cuando un caso de uso incluye a otro, es decir, que el caso de uso incluido tiene que ejecutarse como condición previa.

Una relación amplía o que extiende, se presenta entre dos casos de uso cuando, en función de una condición, un caso de uso amplía el comportamiento del caso de uso inicial.

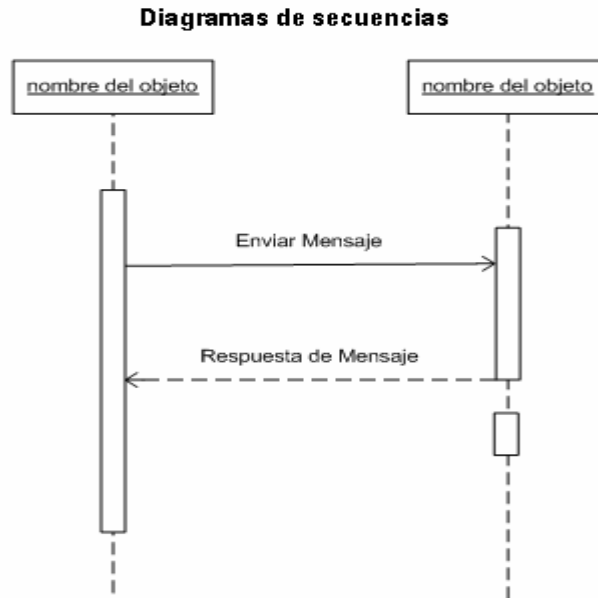


Figura 12. Representación de los diagramas de secuencias en UML

Los diagramas y símbolos que se acaban de mencionar son de mucha ayuda para el modelado de las ideas de un proyecto. Estos símbolos y diagramas en conjunto son una notación estándar fácil de comprender tanto para los desarrolladores como para los usuarios finales.

Herramientas computacionales para el desarrollo de software

En el capítulo anterior se menciona que el Centro trabaja bajo la plataforma Windows, por lo que solamente se mencionarán algunas herramientas y lenguajes de programación, que funcionen sobre la misma plataforma.

Todas las metodologías aludidas, tienen su forma de proseguir para el desarrollo de un software, sin embargo hay que hacer uso de algunas herramientas para obtener los resultados esperados en menos tiempo. Los diagramas a realizar durante el desarrollo de un sistema pueden hacerse con un lápiz y papel, pero la desventaja es que las modificaciones se vuelven tediosas y su presentación dista mucho de lo que se puede hacer con una herramienta computacional.

En seguida, se presentarán algunas de las herramientas computacionales más usuales en el desarrollo del software, que ayudan en la creación de modelos -principalmente modelos UML-.

Rational Rose

Esta software es líder en el mercado en herramientas de modelado, este es un producto de la empresa que da empleo a los tres amigos (Rumbaugh, Jacobson, Booch), es por ello que se enfoca estrictamente al UML.

Al ejecutar tal programa se presenta un panel en la izquierda que es el examinador, y en la ventana de la derecha que generalmente es de mayor tamaño o al menos se puede modificar de tamaño, tenemos la ventana de diagramación; en ella se encuentra una paleta de iconos UML para diagramas de clases, el cual es el tipo de diagrama que se abre de forma predeterminada.

El examinador esta organizado de acuerdo a vistas: vista de caso de uso, vista lógica, vista de componentes y vista de distribución. Para crear un tipo de diagrama, se debe hacer clic con el botón derecho en las vistas, y las opciones relacionadas con ellas aparecerán en el menú. Seleccionando una, aparecerá una nueva ventana de diagramación, también hay una paleta de iconos adecuada para el diagrama a realizar.

Rational Rose tiene algo de dinamismo en sus iconos, ya que agrega distintivos visuales, en sus diagramas.

SELECT Enterprise

Es una herramienta de modelado para toda una empresa, como su nombre lo indica, ya que automáticamente se conecta con un servidor donde es accesible para los modeladores y los usuarios de los modelos de toda una empresa. La ventaja de estas características es que se lleva un control de las personas que acceden a los modelos y permite llevar un control de las versiones de los modelos.

Cuando se ejecuta este programa, la ventana principal de SELECT cargará una parecida a la de Rational Rose, pero hasta ahí queda la similitud. En el esquema de SELECT Enterprise, UML es una de los varios paquetes de modelado importantes, ya que este permite utilizar un conjunto de símbolos que no son de UML para dibujar modelos de datos y generar modelos de proceso de negocios.

La ventana principal de SELECT consta de cuatro pestañas, en su examinador que se presenta en la parte izquierda de la pantalla. La primera contiene un diccionario del modelo que se realiza, la segunda que permite crear los diagramas de modelado, la tercera ayuda a la organización de los diagramas, y la cuarta ofrece ayuda.

Uno de sus funciones exclusivas de este programa es que al generar un diagrama de clases, SELECT permite crear un diagrama de estados sólo si se crea una clase por adjuntarle.

También permite vincular diagramas y se puede llevar a cabo un informe que muestre que diagramas están vinculados con otro en particular.

Visual UML

Al ejecutar esta herramienta, se ve un panel de tipo explorador y examinador del extremo izquierdo de la ventana, que sirven para crear y administrar los diagramas. Es una herramienta muy amigable pues con unos clics y un poco de conocimiento de UML se pueden crear los diagramas, todos los tipos de diagramas están en el nivel superior.

Al crear un diagrama aparece la paleta asociada a esta, estas paletas brindan cierta facultad de utilizar imágenes que nos son de UML, dado que permite dibujar recuadros y líneas en el diagrama.

UML Modeler

Es una buena herramienta CASE para aprender UML Modeler de la compañía Objectteering. Permite crear diagramas UML sin saturar con la multitud de funciones avanzadas.

Al ejecutar este programa, se tiene una ventana, muy similar a las de las herramientas anteriores. Cuenta con un panel izquierdo dividido en dos ventanas, una donde aparece de manera organizada en forma de árbol los diagramas desarrollados en el proyecto –con nombre del proyecto-.

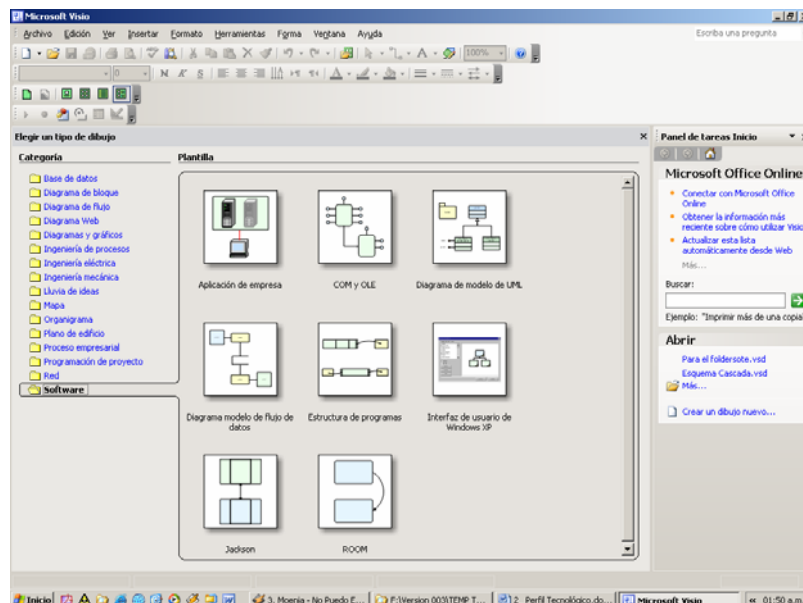
Cuenta con 3 pestañas: diagramas, ítems -símbolos-, documentación.

En la parte inferior hay un panel en el que se indica el estado de o que se esta realizando.

Del lado derecho aparece una pantalla principal del editor.

Microsoft Office Visio 2003

Cuando se ejecuta Microsoft Visio por primera vez, se abre una plantilla en el panel principal. Con ella se abren varias galerías de símbolos en el panel izquierdo, estas tienen los estilos, opciones de configuración y herramientas necesarias para crear el diagrama elegido.



También se abre un entorno de dibujo –es la parte principal de trabajo-, esta, incluye menús, barras de herramientas, galerías de símbolos con formas y un panel de tareas situado a la derecha de la página de dibujo. Para crear un diagrama es suficiente con arrastrar las formas desde las galerías de símbolos

hasta la página de dibujo que se encuentra del lado derecho de la pantalla. Estas formas son específicas para el diagrama a realizar, y muchas de ellas tienen un comportamiento integrado ajustado al comportamiento del dibujo o diagrama a realizar.

La página de dibujo, representa la página impresa e incluye una cuadrícula que ayuda a colocar las formas.

El panel de tareas se emplea para acceder rápidamente a distintos tipos de información mientras se trabaja o para colaborar con otros miembros del equipo.

Se puede obtener rápidamente respuesta a asuntos relacionados con Microsoft Office Visio escribiendo una pregunta o una palabra clave en el cuadro Escriba una pregunta, situado en el lado derecho de la barra de menús. También se pueden ver temas de Ayuda de Visio en el panel de tareas, situado a la derecha de la página de dibujo.

Unas de las características de Microsoft Office Visio es que se pueden compartir diagramas, publicar diagramas en Web, agregar diagramas a archivos creados con otros programas de Microsoft Office System, Crear formas con formato, galerías, crear un logotipo y guardarlo en una galería de símbolos y así crear una plantilla, etc.

Cuenta con una extensa variedad de categorías, para el desarrollo de Bases de datos, diagramas de bloques, diagramas de flujo, diagramas Web, diagramas y gráficos, ingeniería de procesos, ingeniería eléctrica, ingeniería mecánica, lluvia de ideas, mapa, organigrama, plano de edificio, proceso empresarial, programación de proyecto, red y software.

Aunque todas las herramientas aquí mencionadas trabajan de manera distinta, tienen características en común. Cada una permite la diagramación flexible al crear un vínculo entre dos elementos, todas cuentan con extensas herramientas y cuadros de diálogo para sus modificaciones.

Lenguajes de Programación

Un lenguaje de programación está formado por un conjunto de instrucciones (comprensibles por los humanos) que representan órdenes que se le dan a una computadora.

Programación Estructurada

La programación estructurada presenta inconvenientes similares al desarrollo en cascada, además de los que se listan a continuación.

- Los programas son más difíciles de mantener, ya que es difícil modificar las funciones actuales sin que se produzca un efecto adverso en toda la funcionalidad del sistema.

- La programación no se adapta bien al trabajo en equipo. Los programadores tienen que conocer todos los aspectos del funcionamiento del programa, y no pueden concentrar sus esfuerzos en un solo aspecto del sistema.
- Es difícil traducir modelos de trabajo reales en modelos de programación.
- Su funcionamiento es correcto de modo aislado, pero no se integra bien con otros sistemas.

Aunado a estos inconvenientes, el avance tecnológico ha orillado a que los desarrolladores se vean obligados a ofrecer aplicaciones con características semejantes al sistema operativo en el que se van a ejecutar. Por ejemplo, aplicaciones en donde las interfaces sean más intuitivas y amigables, es decir, se está tendiendo a modelos de aplicaciones distribuidos donde la lógica del programa, la interfase del usuario y la base de datos están íntimamente ligadas y se acceden a través de Internet o intranet.

Programación Orientada a objetos

En contraste con la programación estructurada se presenta la programación orientada a objetos.

Actualmente la programación orientada a objetos ha adquirido mucha importancia en el desarrollo del software. Principalmente el desarrollo con el lenguaje de programación Visual Basic pues se ha convertido en uno de los principales lenguajes de programación para aplicaciones Windows, además de su buen contrincante Java.

Los lenguajes de programación orientados a objetos empezaron a gestarse a mediados de los 60's con un lenguaje de programación llamado Smalltalk. A pesar de que los desarrolladores de software no adoptaron mayoritariamente estos tempranos avances, las metodologías y lenguajes orientados a objetos siguieron evolucionando. Algunos lenguajes como C++ se hicieron muy populares durante los 90's, especialmente con la aparición de Java. En el 2002 con la última versión de Visual Studio, Microsoft presentó un nuevo lenguaje de programación orientado a objetos, C#, y rehizo Visual Basic como verdadero lenguaje de POO.

Algunas de las grandes ventajas de los lenguajes orientados a objetos son:

- La transición de los modelos de trabajo reales a los modelos de implementación de software son más intuitivos.
- El mantenimiento y la implementación de cambios en los programas es más eficiente y rápida.
- El trabajo en equipo permite crear sistemas de software de un modo más eficiente, gracias a la especialización de cada individuo en una parte concreta del sistema.

- La integración con sistemas informáticos distribuidos⁷.
- La integración con los Sistemas Operativos actuales como por ejemplo Windows XP Professional, Windows Vista, Windows Server 2003, también es mejor.
- Permite crear interfaces gráficas de usuario, más intuitivas.

A continuación se hará una breve descripción de algunos lenguajes visuales orientados a objetos, y que actualmente en el mercado son muy utilizados, en el desarrollo de aplicaciones.

Visual Basic 6

Un lenguaje de programación robusto con la comunidad de desarrolladores más extensa y activa del mundo. Visual Basic 6 viene incluido en Visual Studio 6 de Microsoft, permite desarrollar robustas aplicaciones de 32 bits ya que es un lenguaje compatible solo con Windows. Fue diseñado para facilitar el desarrollo de aplicaciones en un entorno grafico (GUI-Graphical User Interface).

Sin embargo presenta algunos inconvenientes, ya que muchos consideraban que contaba con sintaxis de objetos, sin embargo no es un verdadero lenguaje orientado a objetos. Esta versión de Visual no dispone de funciones tan básicas de POO como la herencia, y sobrecarga de métodos. Microsoft reconoció este inconveniente y desarrollo desde cero a Visual Basic.Net, un verdadero lenguaje de programación OO.

Java

Fue diseñado por la compañía de Sun Microsystems, con el propósito de crear un lenguaje que pudiera funcionar en redes computacionales heterogéneas y que fuera independiente de la plataforma en la que se vaya a ejecutar. Esto significa que un programa de java puede ejecutarse en cualquier máquina o plataforma.

Es un lenguaje orientado a objetos, su sintaxis es similar a la de C y C++, cualquier computadora puede compilar un programa java que tenga el interprete de java. Corre en una maquina virtual, por lo tanto es un lenguaje interpretado.

Es un lenguaje que soporta múltiples hilos de procesamiento (threads), es decir, es un lenguaje que puede ejecutar diferentes líneas de código al mismo tiempo.

Visual Basic .NET (VB.NET)

⁷ Un sistema informático distribuido esta basado en el modelo de computación distribuida que es aquel que utiliza sus módulos diseminados en distintos equipos de cómputo.

Es una versión de Visual Basic enfocada al desarrollo de aplicaciones sobre plataforma .NET.

Al desarrollar aplicaciones en cualquier lenguaje de Visual Studio .Net, se trabajará con objetos, métodos, propiedades y eventos. En seguida se definirán estos conceptos:

Un objeto se define como se especifica en la lengua tradicional, es decir, los objetos son todas las cosas que nos rodean, por ejemplo: Héctor, Lucero, Jesús, águila, halcón, zanahorias, manzanas, ratón, teclado, etc.

Un **objeto** es un elemento programable que se usa para hacer una aplicación, por ejemplo en Visual Basic puede ser un formulario, y en la vida cotidiana un reloj, por ejemplo. Técnicamente un objeto es la instancia de una clase, por ejemplo, mi compañero y yo somos instancias de la clase persona.

Una **clase** es un "molde" que a partir del cual se pueden crear objetos.

Para poder interactuar con tales objetos se hace uso de las propiedades, métodos y eventos.

Las **propiedades** de un objeto, son sus características. Por ejemplo, las características de un formulario de Visual Basic pueden ser altura, ancho y color, mientras que para el reloj, pueden ser también la altura, el ancho y el color.

Un **método** son las cosas que los objetos que pueden hacer, por ejemplo los métodos para el formulario de Visual Basic son la de mostrarse y ocultarse. Para el reloj sus métodos serían el poder modificar la hora, minutos y/o segundos.

Los **eventos** de un objeto son las acciones a las que puede responder. Por ejemplo cuando un usuario cierra el formulario, el formulario responde a un evento llamado closed que ejecuta el código que el programador puso en el evento. En el reloj, produce sonidos respondiendo al evento alarma.

El desarrollo en .NET, es un novedoso modelo de computación distribuida llamado servicios Web basados en XML (Lenguaje de Mercado Extensible)., en la que el elemento integrador es Internet y se le da forma gracias al aumento de equipos conectados a Internet.

El objetivo es poder comunicar entre si, sistemas heterogéneos dentro y fuera de una empresa y que esta comunicación sea independiente del sistema operativo y del lenguaje o modelo de programación. Para esto se crearon estándares que permitan hacer uso de los servicios Web basados en XML:

Algunos de los estándares son:

- **XML (Lenguaje de Mercado Extensible)**. Como formato universal para el marcaje de textos (para la representación de datos).
- **SOAP (Protocolo simple de Acceso a objetos)**. Este es un protocolo que permite mover los datos entre las aplicaciones y sistemas. Este es el mecanismo por medio del cual los servicios Web son invocados e interactúan.

- **UDDI (Descubrimiento, Descripción e Integración Universal).** Este es un lenguaje que permite publicar, encontrar y usar los servicios Web basados en XML, es como la página amarilla. UDDI es también un servicio Web.
- **WSDL (Lenguaje de descripción de Servicios Web).** Lenguaje por medio del cual los servicios describen entre otras cosas que hacen o cuál es su función.

La plataforma .NET no es un solo producto. Es un conjunto de productos. Desde sistemas operativos como Windows XP, servidores de aplicaciones como SQL Server 2000, productos de oficina como Office XP, herramientas de desarrollo como Visual Studio .Net (que es donde se ubica Visual Basic .NET), Clientes inteligentes como PocketPC (PC de bolsillo), SmartPhone (Teléfonos inteligentes), TabletPC, Xbox, PC's, NoteBooks, y hasta servicios Web provistos por Microsoft como .NET Passport (actualmente Password Network).

Tanto la invocación de los servicios como su ejecución pueden ser hechas en cualquier dispositivo o sistema operativo Windows y accedido desde Internet.

El componente principal de la plataforma .NET, es el .NET Framework, que a su vez su componente fundamental es el CLR (Common Lenguaje Runtime). Este componente administra el código que será ejecutado y forma una capa de extracción entre el código y el sistema operativo.

Sobre el CLR se encuentra la biblioteca de clases base de .NET Framework, estas clases e interfaces base sirven de base a los programadores para realizar sus aplicaciones, componentes y controles.

Una vez que se sabe un poco acerca de .NET y donde se ubica el lenguaje de programación Visual Basic .NET, se mencionará un poco acerca de él.

Visual Basic .NET es un lenguaje de programación orientado a eventos con nuevas características orientadas a objetos, y como novedades más importantes podemos citar la posibilidad de definir ámbitos de tipo, clases que pueden derivarse de otras mediante herencia, sobrecarga de métodos, nuevo control estructurado de excepciones o la creación de aplicaciones con múltiples hilos de ejecución.

Otras de sus particularidades son:

- Cuenta con un IDE (Entorno de Desarrollo Integrado), que hace que la programación sea muy sencilla, además de ser el mismo para los demás lenguajes compatibles con .NET.
- Diseño de controles de usuario para aplicaciones Windows y Web.
- Crear librerías de controles
- Programación de bibliotecas de clase.

- El .NET Framework cuenta con un conjunto de clases ADO.NET (implementadas en sus clases base), que son las responsables de la administración de los datos. También cuenta con un proveedor OLEDB.
- Posibilidad de conectarse con diferentes tipos de bases de datos, como por ejemplo Oracle, SQL Server, Access, etc.
- Envío de datos vía documentos XML que es un formato universal para representar datos.
- Generación de reportes basados en Crystal Reports a partir de información obtenida de orígenes de datos (archivos de texto, bases de datos, etc.)
- Creación de Web Services que son unidades de código, basados en XML, en donde cada una hace una tarea en particular. Con estos servicios se busca crear aplicaciones centradas en el usuario, es decir hacer que la información del usuario no este relacionada con ningún dispositivo, lenguaje, etc.
- Aplicaciones de consola. Se puede escribir código de Visual Basic .NET en un editor de textos y compilarlos mediante línea de comando.
- Crear aplicaciones para dispositivos móviles (Pocket PC) que soporten el Compact .Net framework que es una versión reducida del framework original.
- Posibilidad de crear módulos en diferentes lenguajes .Net, gracias al Framework, ya que Microsoft desarrollo lenguajes compatibles con .NET Framework. Además de Visual Basic, estos lenguajes son C#, C++, JScript y J#. Para unirse a estos lenguajes, otros proveedores han desarrollado versiones compatibles con .NET Framework de muchos lenguajes que gozan de gran popularidad, como Perl y SmallTalk.
- Integración completamente con Internet, ya que uno de los objetivos primordiales de .NET es aprovechar el uso de Internet y las redes en las aplicaciones, pues cuenta con numerosas clases integradas en el .NET Framework, como System.Web, System.Net, System.Net.Sockets, System.Web.Services.
- Convierte a los objetos en cajas negras donde lo que interesa es lo que hace y no cómo esta implementado, esto permite la reutilización de objetos, pues el código de tal objeto no se tiene que volver a escribir cada vez que se quiere usar.

La importancia de conocer las metodologías y herramientas anteriormente mencionadas es simplemente porque las metodologías no trabajan de manera aislada, sino que van acompañadas al menos de una herramienta computacional de modelado y un proceso de desarrollo (método). La herramienta computacional de modelado ayuda al uso de la notación (principalmente grafica) que los métodos usan para expresar sus diseños en cada etapa. En el proceso se incluyen las etapas que se irán realizando para obtener dichos resultados.

Gestores de Base de Datos

Ya se mencionó sobre las metodologías, herramientas computacionales, lenguajes de modelado y lenguajes de programación, sin embargo no es suficiente para el desarrollo de la aplicación, ya que falta hablar de las Bases de Datos, una parte importante en los sistemas, pues su función principal es la de almacenar los datos que interactuarán con el sistema o software.

Los gestores que se mencionarán son Access y SQL Server, esto se manejará así, porque se asegura plena compatibilidad con Windows, además de que no se invertirá en otras herramientas porque como ya se mencionó en el capítulo I, se cuenta con estas.

Microsoft Office Access

"Microsoft Access es un sistema gestor de bases de datos relacionales (SGBD)."⁸

En la terminología propia de las bases de datos hay tres conceptos claves dentro de las tablas: campo, registro y dato.

Un campo es cada uno de los tipos de datos que se van a usar. Se hace referencia a los campos por su nombre.

Un registro está formado por el conjunto de información en particular.

Un dato es la intersección entre un campo y un registro.

Se puede crear un proyecto de Microsoft Access que sea fácil de conectar a una base de datos de Microsoft SQL, o utilizar el Asistente de Microsoft SQL Server para crear una base de datos de SQL Server y un proyecto de Access a la vez. Se puede hacer una conexión desde Access para que mediante las herramientas de SQL se pueda ver, crear, modificar y eliminar tablas, vistas, procedimientos almacenados y diagramas de base de datos.

También se puede crear una nueva base de datos de Access a partir de datos de un programa de Microsoft Office System, como por ejemplo un archivo de texto o de hoja de cálculo: Simplemente se abre el archivo, y Microsoft Access crea automáticamente una base de datos de Access y vincula el archivo. Además de que se puede importar o vincular datos de Microsoft Outlook o Microsoft Exchange.

Microsoft SQL Server 2000

Es un motor de bases de datos que también comercializa Microsoft. Éste posee varias aplicaciones para la administración de las bases de datos, entre ellas se encuentra el Enterprise Manager y el Query Analyzer.

⁸ Ayuda de Microsoft Office Access.

Cuenta con las siguientes características:

- Un motor de programación
- Funciones de alerta de administrador
- Operaciones de control entre múltiples servidores
- Permisos a usuarios
- Crear secuencias de comandos
- Crear respaldos de Bases de Datos
- Administrar tablas, vistas, procedimientos almacenados, índices, reglas, etc.
- Brinda autenticación de usuarios por medio del sistema operativo (Windows NT/2000/2003), o por medio de él (autenticación SQL).

Además de ser **un servidor .Net Corporativo**, es decir, es soportado nativamente por la plataforma. Net, pues provee la infraestructura para su construcción, instalación y operación.

Metodología, herramienta y lenguaje de programación, ideales para el desarrollo de Software

Para poder definir qué metodología, herramienta y lenguaje de programación son ideales para un buen desarrollo de software y de buena calidad, se debe tener en cuenta, que la clave está en tener una excelente comprensión del problema a resolver.

Además que para realizar tal elección, se dependerá del líder del proyecto, de la experiencia del líder del proyecto, de las características del equipo de desarrollo, de las necesidades del sistema, de la experiencia en determinados lenguajes de programación y de otros tantos factores. No obstante, que una metodología sea muy útil sólo para algunos desarrolladores no quiere decir que sea inservible para otros.

En cuanto a la herramienta a elegir, inicialmente se debe analizar exhaustivamente si se adapta a nuestras necesidades y si es lo suficientemente flexible, pues muchas de las funciones avanzadas de las herramientas de desarrollo, son difíciles de utilizar por lo que llega a pasar que se dedica más tiempo a aprender a manejarlo que a diagramar el diseño en sí.

Pero entonces, ¿Qué persona es la elegida para tomar tal decisión? la respuesta es: la que cuente con más experiencia dentro del grupo de desarrolladores o el líder del proyecto. Persona que en base a su experiencia y a los requerimientos del futuro software puede elegir con fundamentos el enfoque idóneo para cada proyecto. Su trabajo es asesorar al equipo de desarrollo sobre qué metodologías, generación de documentación, lenguajes de programación, herramientas de desarrollo, etc., que en el caso del desarrollo del sistema para el centro de cómputo, seremos nosotros.

Ahora que se conocen con qué opciones se cuentan para el desarrollo del software que se pretende crear, es necesario conocer los requerimientos, para así, poder decidir las opciones sobre las que se trabajarán.

3

3. Recopilación de las necesidades del sistema

La obtención de requerimientos del futuro sistema permitirá obtener las características necesarias que deberá poseer el nuevo sistema. Este es el primer paso y el más importante en el inicio del desarrollo del sistema, aunque en algunas metodologías no la definen como tal, siempre se deben considerar, ya que son las bases para la construcción del mismo.

Necesidades propias para el sistema

La parte técnica para el centro de cómputo, es diseñar el sistema a las necesidades de los usuarios, no a la tecnología, insistiendo que para esto es muy importante la ayuda de los usuarios finales, en este caso son los alumnos de la Preparatoria 9, y principalmente del administrador del centro de cómputo.

Se requiere de un sistema que como se mencionó ayude a disminuir el trabajo de los encargados o administradores de Centros de Cómputo para la Escuela Nacional Preparatoria Plantel 9 "Pedro de Alba" y se brinde un mejor servicio a los alumnos.

Con el sistema se pretende agilizar el proceso de entrada y salida de los usuarios, la asignación de equipo y sala a los usuarios, lograr una optimización de los recursos del Centro con el apoyo de los resultados obtenidos por los diversos reportes que generará el sistema.

Sólo por mencionar algún ejemplo, se logrará una optimización del espacio de almacenamiento en el equipo, puesto que con el registro de uso de servicios se pueden calcular estadísticas para saber cual está siendo utilizado con mayor frecuencia y, así, se puede decidir que solo determinado software sea instalado en los equipos, logrando con ello mayor espacio de almacenamiento y rendimiento.

El sistema será operado por los encargados del centro de cómputo (CC), en donde cada uno tendrá una cuenta de administrador⁹.

Dicha cuenta deberá tener un nombre de usuario, nombre completo del Administrador y una contraseña (la contraseña no podrá ser vacío o nulo) para evitar que alguien no autorizado tenga acceso al sistema.

Además, de las siguientes características en cuanto a los:

Administradores

El sistema vendrá configurado por default con una cuenta con contraseña de un administrador llamada SACCNET que no se podrá borrar, esto para evitar que el sistema se quede sin administradores.

Existirá un módulo donde se podrá dar de alta o de baja a los administradores.

Para dar de alta a un administrador en el sistema se requerirá que forzosamente otro administrador lo registre; de igual forma, para poder dar de baja a un administrador se requerirá de otro.

Se podrá cambiar la contraseña y para esto será necesaria la contraseña anterior.

Se contará con la posibilidad de editar una cuenta de administrador y para ello también se requerirá la contraseña.

Con esto se pretende evitar que un administrador pueda actuar en nombre de otro ya que cada movimiento que realice un administrador se registrará en una bitácora que se imprimirá con su nombre.

Usuarios

En esta parte del sistema, se permitirá agregar (dar de alta) y quitar (dar de baja) a un usuario (alumno o externo) además de permitir la edición y la búsqueda.

Como se cuenta con el registro de cada uno de los alumnos que están activos en el plantel, para ello al buscarlo y no encontrarlo en dicho registro se podrá agregar al alumno pero será necesario conocer su número de cuenta que

⁹ Administrador es toda aquella persona que usa el sistema sin ninguna restricción.

será usado como ID de usuario, nombre completo, grupo, además de contar con un campo de observación.

En caso de ser externo (personas que no sean alumnos) al darlo de alta se creará un ID único de usuario, para ello será necesario su nombre completo y procedencia, además de contar con un campo para observaciones.

Para dar de baja a un usuario (alumno o externo) es suficiente con conocer su ID de usuario.

En dicho registro no deberán existir ID's de usuario repetidos.

Para editar los datos de un usuario es necesario conocer su ID de usuario o nombre.

La búsqueda de un usuario puede realizarse, conociendo sólo el ID de usuario o nombre completo.

Sanciones

También se registrará a los usuarios que hayan sobrepasado o violado las reglas impuestas por el centro de cómputo.

Así el sistema, en el caso de que un usuario esté sancionado y solicite servicio, enviará un mensaje de SANCIONADO y no podrá registrar ningún movimiento hasta que un administrador la anule.

En esta parte el sistema deberá poder dar de alta una sanción, editarla, anularla, además de poder consultarlas todas.

Identificación de un usuario

Cuando un usuario solicita servicio, deberá entregar una identificación ya sea con su número de cuenta o nombre completo para que quede como garantía del equipo.

En esta parte el sistema deberá poder dar de alta, baja, editar o ver todos los tipos de identificaciones.

Préstamo de Equipo

Una vez que el sistema verifica si el usuario está sancionado y se busca el usuario solicitado mostrando automáticamente su ubicación y luego si hay equipos libres en dicha ubicación.

En caso de haber lugar, se le asigna uno, y se registrará en un almacén de datos que se llamará ocupación, donde se registrará ID de usuario, ID de identificación, ID de Equipo, servicio, fecha con hora de entrada y administrador que lo esta prestando, y es cuando entonces se marcará el equipo como ocupado, hasta que sea asignado a dos usuarios.

Cuando un administrador registre la salida de un usuario se buscará en el registro de ocupación, se marcará su hora de salida, y se registrará el histórico, usuario, ID, ubicación, servicio, fecha y hora de entrada y fecha y hora de salida.

Como existe la posibilidad de prestar una ubicación (sala) completa a un profesor o alumno, ya sea para clases, conferencias, prácticas, exposiciones, etc., el sistema permitirá asignar varios equipos a un solo usuario, que habitualmente son profesores del mismo plantel.

Además de que podrán consultar los prestamos de equipo.

En caso de ser necesario se podrá cancelar algún préstamo.

Ubicación (sala)

El sistema permitirá dar de alta, baja, edición y ver todas las ubicaciones. Considerando una sala como un conjunto de máquinas que pueden o no compartir el mismo servicio.

Servicios

El sistema permitirá dar de alta, baja, edición de los datos que identificarán el servicio con el que cuentan los equipos. Además de poder ver todos los servicios.

Equipos

Esta parte del sistema se dividirá en cinco sub-módulos que serán:

- Monitor
- Teclado
- Mouse
- CPU
- Extras (cañón, impresora, scanner, cámaras digitales, bocinas, etc.).

Un equipo puede tener más de un extra asignado.

Cada una de las anteriores contará con una parte en donde se harán observaciones.

Un equipo se considera disponible cuando cuente por lo menos con Monitor, Teclado, Mouse, CPU, en caso de que alguno de los componentes no funcione o falle, se avisará con un mensaje en el instante de asignarlo a un usuario.

Cada uno de los dispositivos o equipos debe de tener altas, bajas, la posibilidad de editarlos y eliminarlos.

Dar de baja un equipo, hace referencia

Reportes

El sistema podrá imprimir ya sea en pantalla o en impresora diversos reportes tales como:

- De usuarios
- Del histórico
- De sanciones
- De equipo (teclado, mouse, monitor, CPU, extras)
- De servicios
- De ubicaciones
- De equipo-servicio (de los servicio que hay en cada equipo).

Todos deben poder imprimirse en rangos por fecha, por servicio, aplicación, usuario, por máquina, por inventario, por equipo, por estado, por administrador, por número de cuenta, etc.

Se podrá exportar a un archivo de Word o Excel, PDF y RTF.

Para terminar con las necesidades propias del sistema se pondrá un nombre al sistema. Analizando todas las tareas que se pretende ayude el sistema, se observa que se orienta a ayudar a realizar las tareas administrativas que se presentan en el centro de cómputo, por lo tanto se llamará "Sistema Administrador de Centros de Cómputo (SACC)".

Este nombre se podrá complementar con la frase "para nivel bachillerato", ya que aunque está enfocado específicamente a las tareas del centro de cómputo de la ENP 9, la mayoría de los Centros al menos los de nivel bachillerato, trabajan de una manera muy parecida.

Una vez concluida esta parte es conveniente conocer a las personas van a interactuar con el, directa o indirectamente, para poder continuar con los casos de uso, en donde tales actores juegan un papel importante,

Personas implicadas en el sistema

En esta parte se identificará a la gente que tiene relación directa o indirecta con el futuro sistema, y se mencionará cuál es la importancia de ellas. Considerando que son personas que sólo están relacionadas con el sistema y no con el desarrollo.

En base a los diagramas de actividades del capítulo 1, se pueden realizar los diagramas de responsabilidades para poder identificar a los actores principales que intervienen en los procesos del centro de cómputo.

Diagramas de responsabilidades

Los diagramas de responsabilidades lucirán de la siguiente manera:

Diagrama de responsabilidades atender usuarios.

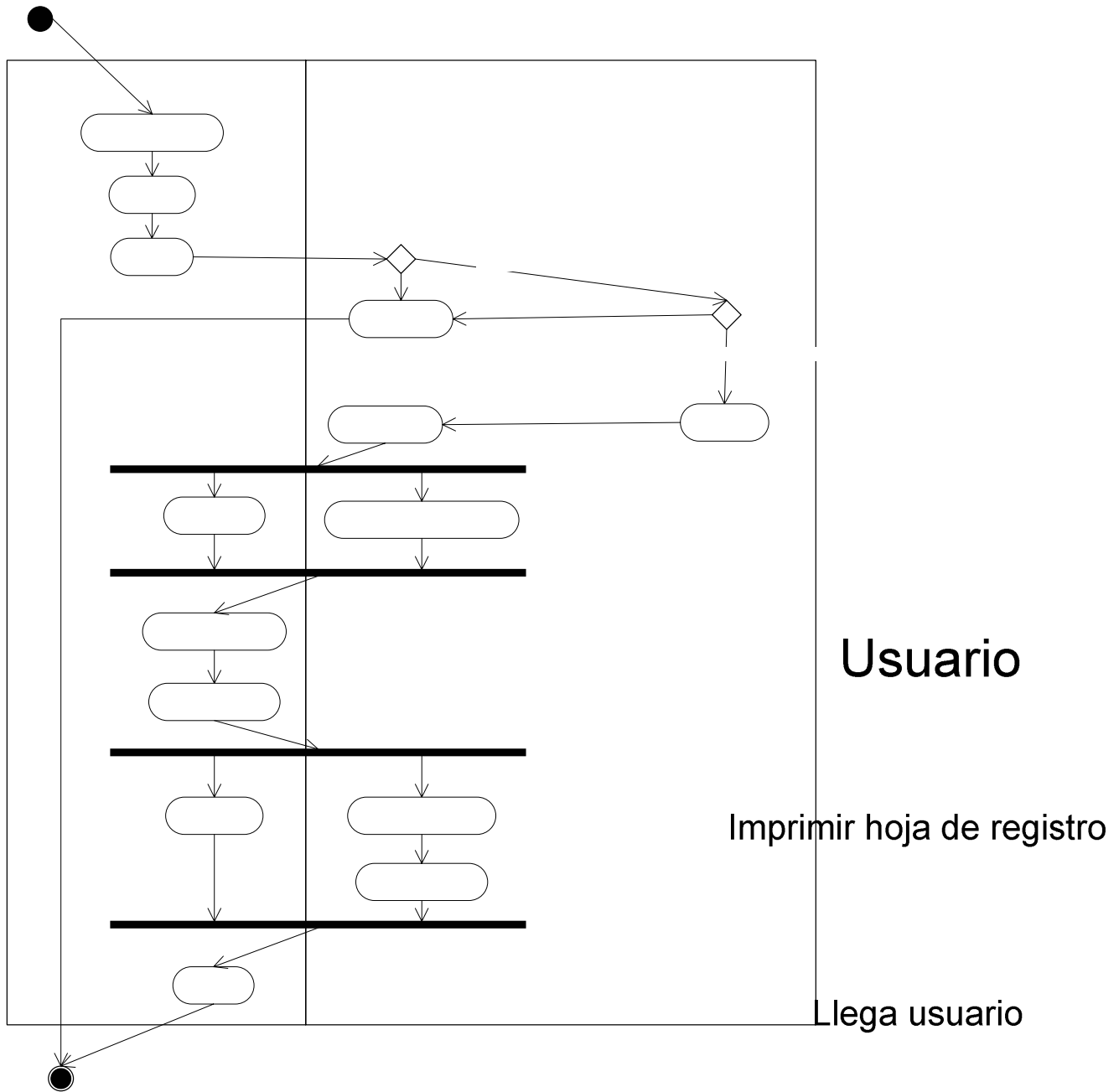


Figura 13. Diagrama de Responsabilidades del proceso "Atender Usuario".

Pide servicio

Diagrama de responsabilidades de alta de un equipo

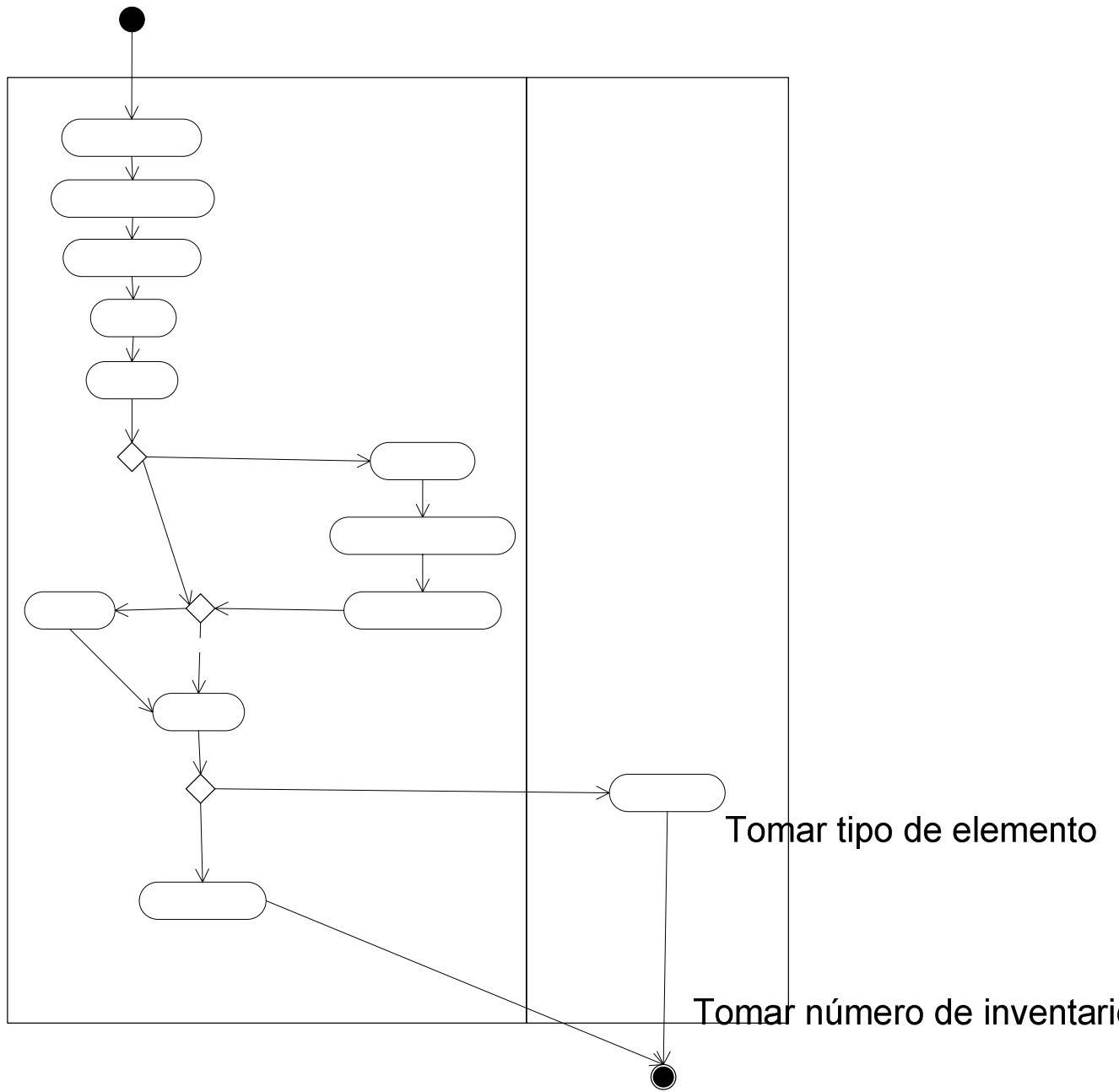


Figura 15. Diagrama de Responsabilidades del proceso "Alta de un equipo"

Tomar tipo de elemento

Tomar número de inventari

Tomar número de serie

Tomar marca

Tomar modelo

Diagrama de responsabilidades de baja de un equipo

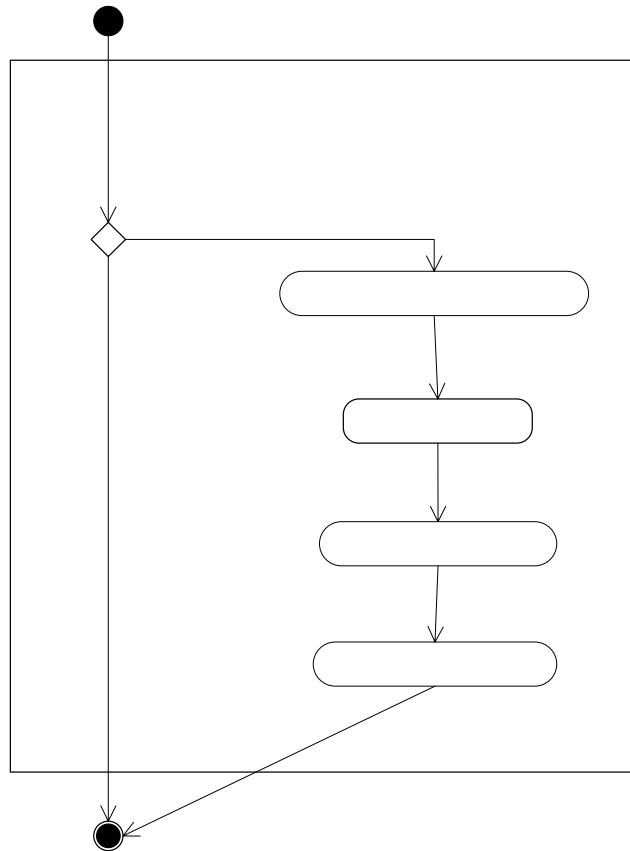


Figura 16. Diagrama de Responsabilidades del proceso "Baja de un equipo"

Diagrama de responsabilidades de préstamos

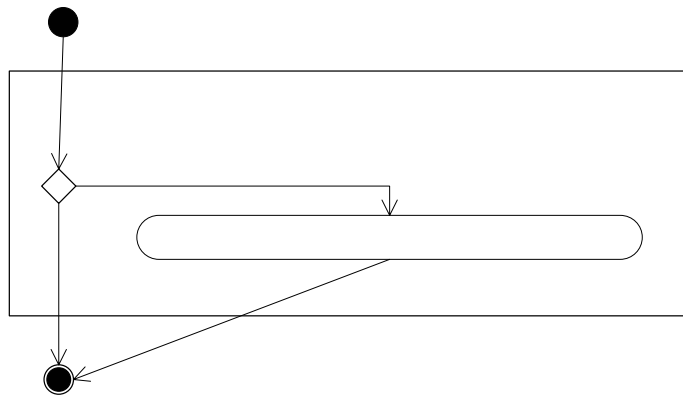


Figura 17. Diagrama de Responsabilidades del proceso "Préstamos"

Diagrama de responsabilidades de horarios

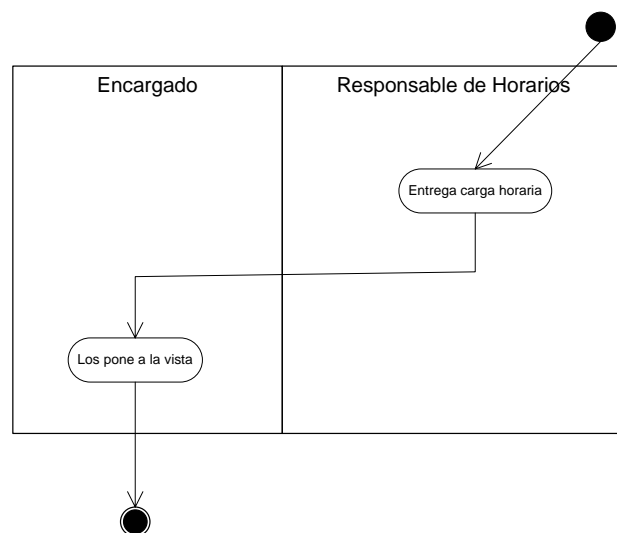


Figura 18. Diagrama de Responsabilidades del proceso "Horarios"

Como se puede apreciar en tales diagramas (figuras 13, 14, 15, 16, 17 y 18) la mayoría de las tareas son llevadas a cabo por el encargado, esta es una muy buena demostración de la carga de trabajo para los técnicos en turno.

Encargado, Técnico o Administrador. Persona responsable del buen funcionamiento del centro de cómputo, con autorización para realizar actividades que favorezcan al Centro.

Usuarios (alumnos y no alumnos). Hacen uso de los diversos servicios del centro de cómputo.

El siguiente paso es identificar las actividades que llevan a cabo dentro del Centro (casos de uso) y se pretenden automatizar.

Las piezas esenciales para que funcione el sistema, son precisamente tales actores. A partir de estas, se puede producir un diagrama con paquetes que muestre tales piezas. Donde cada paquete deberá contener casos de uso o las acciones que llevan a cabo esas personas.

En donde cada paquete debe tener el conjunto de casos de uso de una manera organizada.

La siguiente figura nos muestra las piezas fundamentales para el sistema.

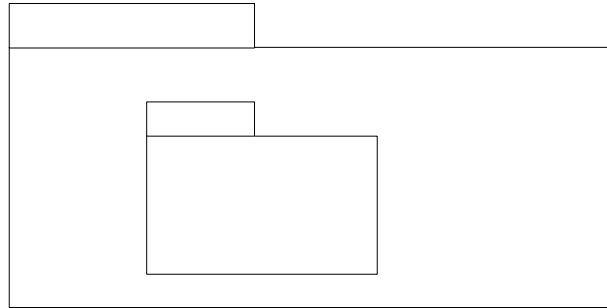


Figura 19. Piezas principales del Sistema SACC.

Las acciones de cada paquete se muestran a continuación:

Para el paquete Administrador:

Préstamo

- Préstamo de Equipo
- Cancelación de préstamos
- Terminación de préstamos
- Ver todos los préstamos

Usuario

- Dar de alta un usuario
- Editar un registro de un usuario
- Dar de baja de un usuario
- Buscar usuarios.

Identificaciones

- Dar de alta una identificación
- Editar una identificación
- Dar de baja una identificación
- Ver todas las identificaciones

Sanciones

- Dar de alta una sanción
- Editar una sanción
- Dar de baja una sanción
- Ver todas las sanción

Administradores

- Dar de alta un administrador
- Editar un registro de un administrador
- Eliminar un administrador
- Cambiar contraseña

Ubicación

- Dar de alta una ubicación
- Editar una ubicación
- Dar de baja una ubicación
- Ver todas las ubicación

Servicios

- Dar de alta un servicio
- Editar un servicio
- Dar de baja un servicio
- Ver todos los servicios

Equipo (CPU o monitor o mouse o teclado o extras)

- Dar de alta un equipo(CPU o monitor o mouse o teclado o extras)
- Editar un equipo(CPU o monitor o mouse o teclado o extras)
- Dar de baja un equipo(CPU o monitor o mouse o teclado o extras)
- Eliminar un equipo(CPU o monitor o mouse o teclado o extras)
- Ver todos los equipo(CPU o monitor o mouse o teclado o extras)

Como se puede observar, el paquete Administrador es el de mayor actividad, y una de las razones más importantes de la existencia del sistema SACC.

Por lo tanto el diagrama de paquetes de funcionalidad será:

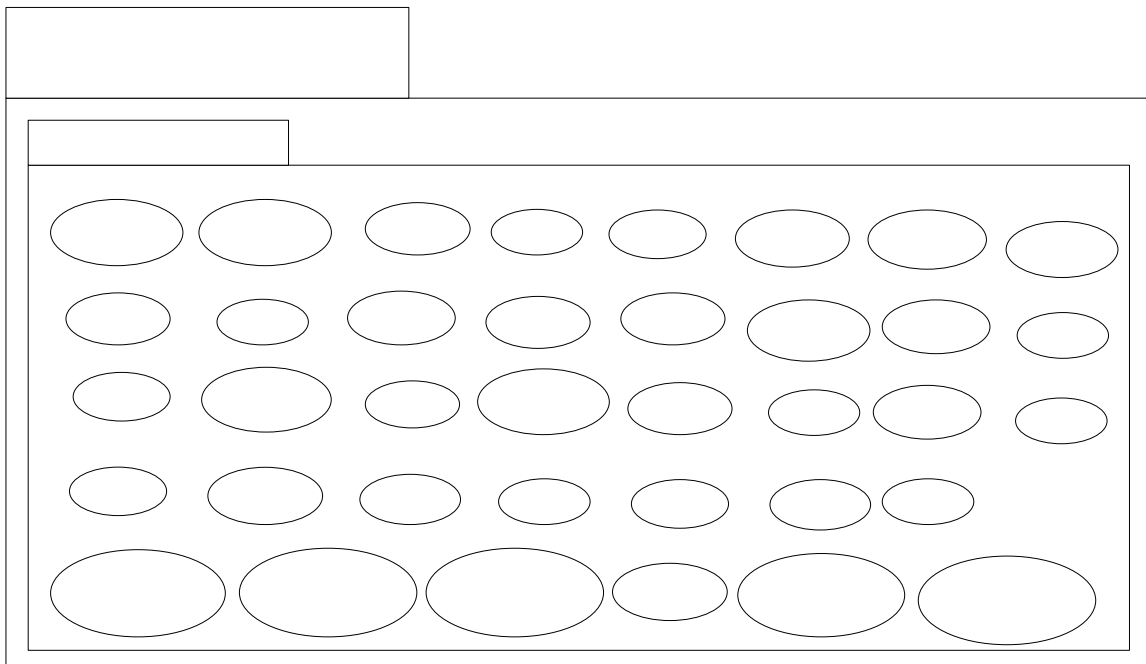


Figura 20. Piezas principales del Sistema SACC.

Una vez identificados a los actores principales que utilizarán el sistema con sus respectivas actividades, se pueden elaborar los diagramas generales de casos de uso de SACC que ayudarán a comprender la forma en que el sistema se comportará desde el punto de vista del usuario, además de indicar los límites entre el sistema y el mundo exterior.

Para organizar los casos de uso, se agruparan en un "paquete" los casos de uso que se relacionen, como se muestra en la figuras 21 a 27.

Diagramas generales de los casos de uso de SACC

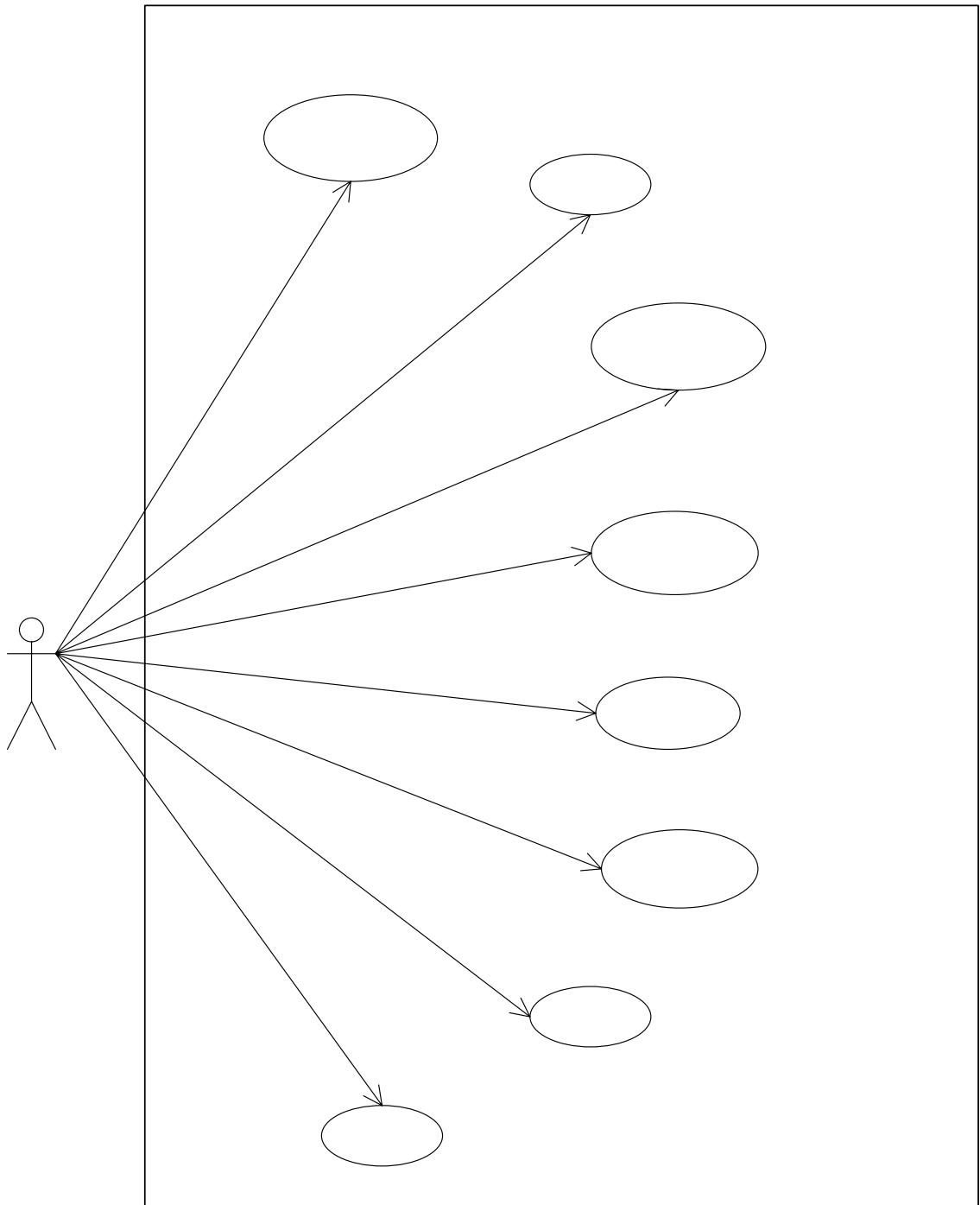


Figura 21. Casos de Uso general de SACC.

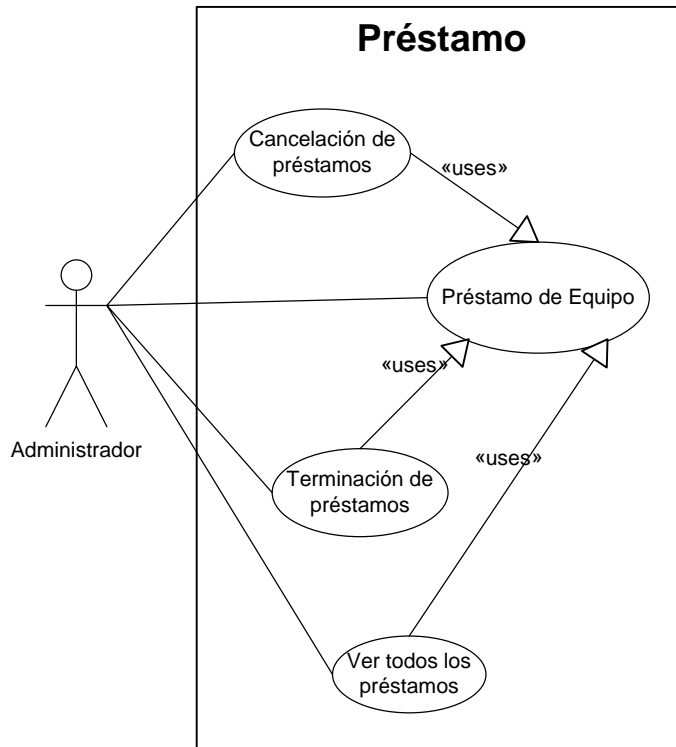


Figura 22. Casos de Uso relacionados con Préstamo de SACC.

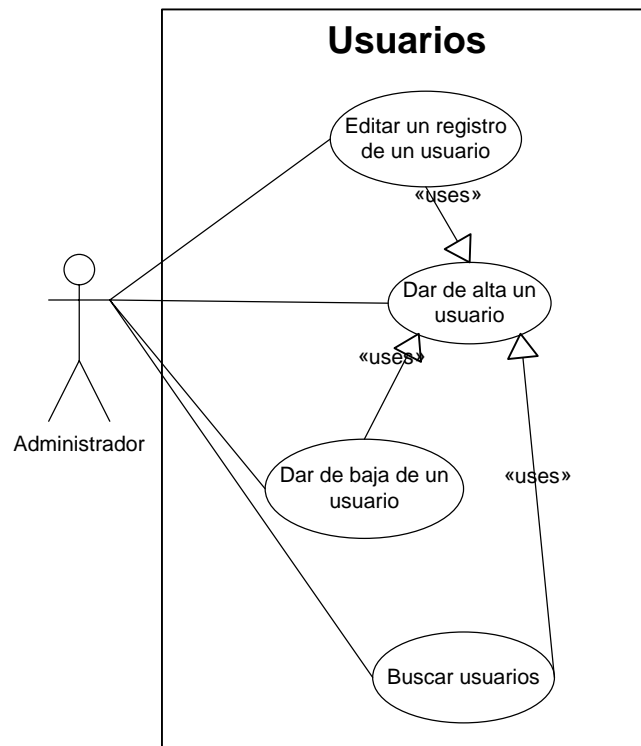


Figura 23. Casos de Uso relacionados con Usuarios de SACC.

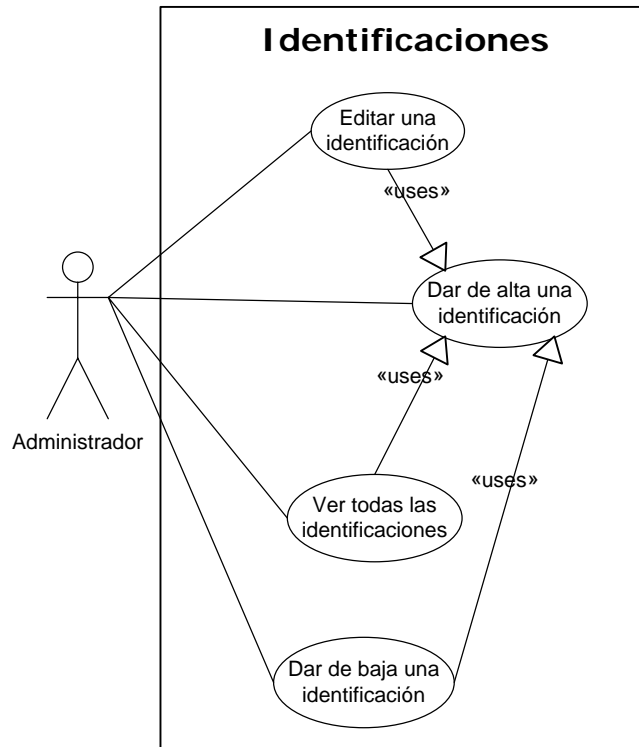


Figura 24. Casos de Uso relacionados con las identificaciones de SACC.

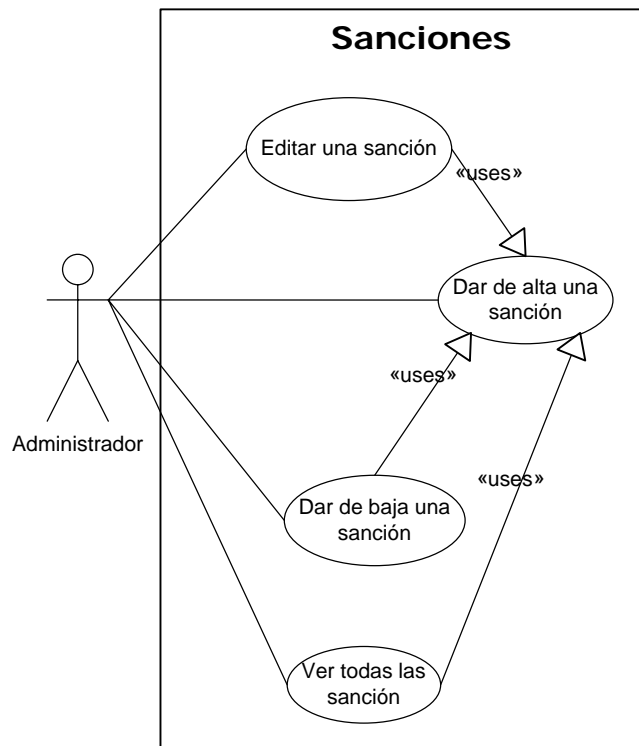


Figura 25. Casos de Uso relacionados con Sanciones de SACC.

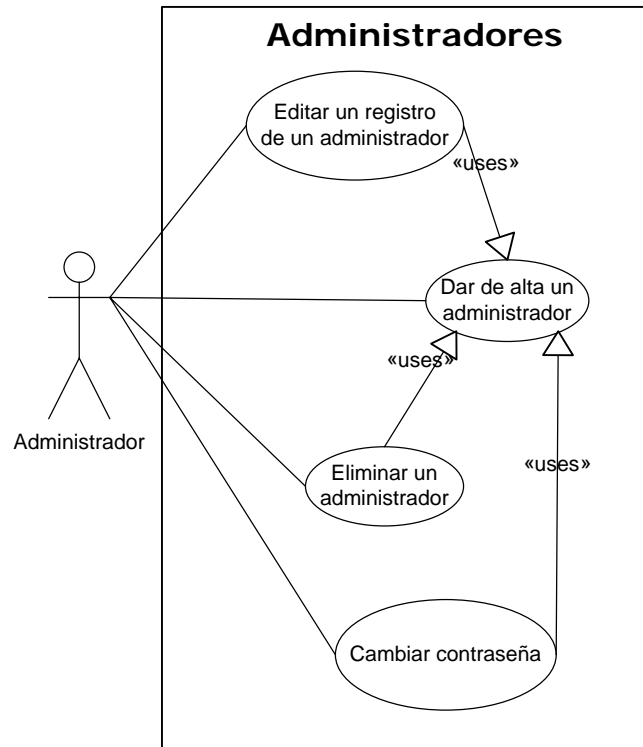


Figura 26. Casos de Uso relacionados con los Administradores de SACC.

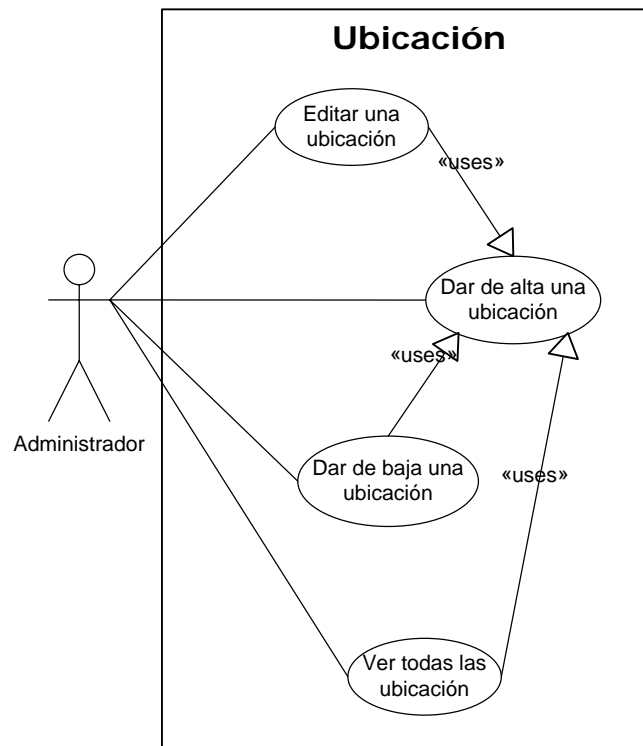


Figura 27. Casos de Uso relacionados con las ubicaciones de SACC.

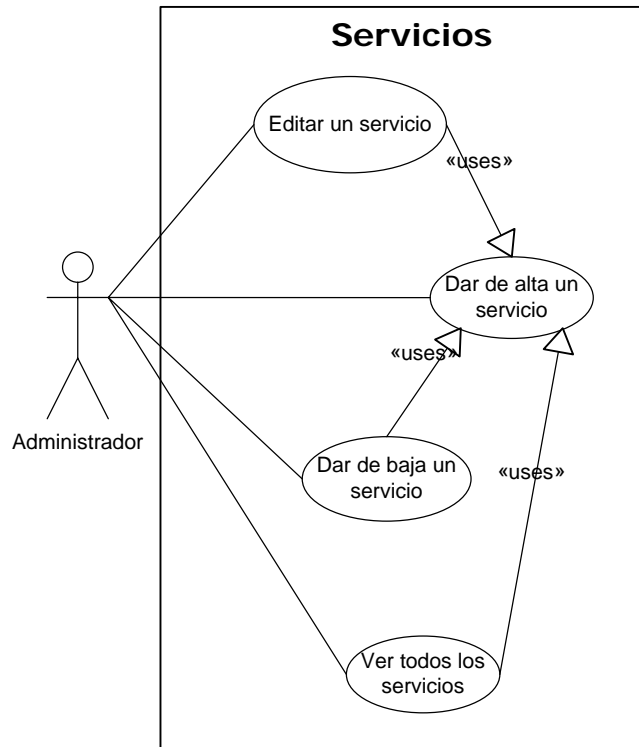


Figura 28. Casos de Uso relacionados con los servicios de SACC.

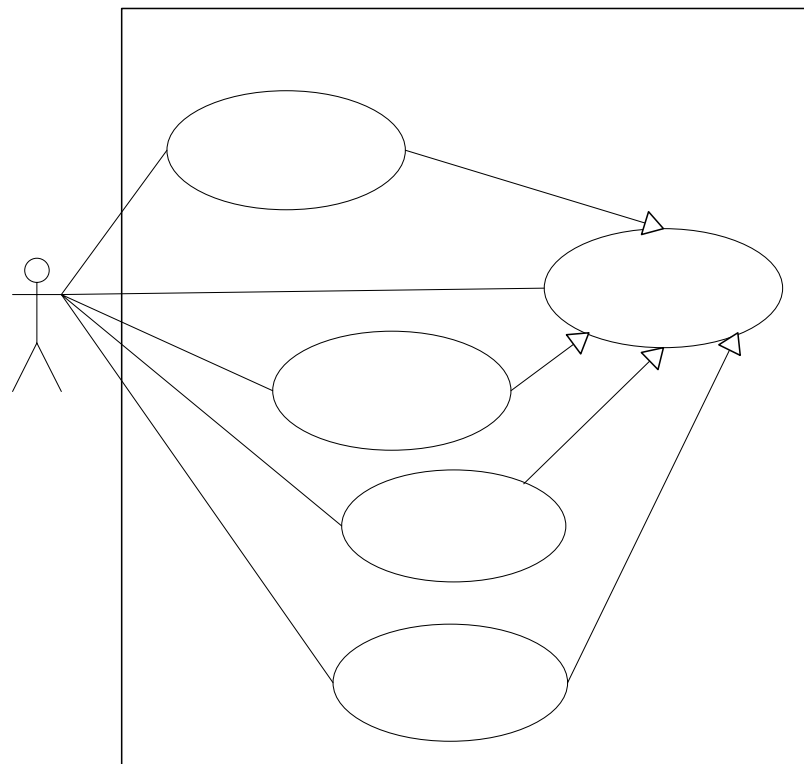


Figura 29. Casos de Uso relacionados con el Equipo de SACC.

Después de analizar los requerimientos propios del sistema y los casos de uso, se logrará que el desarrollo de SACC resulte más manejable, desarrollando

la aplicación en diferentes fases, es decir, que gracias a la forma en que se organizaron los casos de uso, se puede realizar primero la relacionada con los usuarios, después el registro de salida, etc., aunque no es forzoso realizarlo así.

Una vez identificados a los usuarios principales con sus respectivas actividades (casos de uso), se debe definir cuál metodología y cuáles de las herramientas mencionadas en el capítulo anterior cubren los requerimientos para el desarrollo del sistema.

Factores primordiales que influyen en la decisión de las elecciones de las herramientas

En los capítulos anteriores, se mencionan algunos puntos clave para el desarrollo del sistema, sin embargo, en esta parte se deben dejar bien definidos, para así poder decidir que opciones de las ya mencionadas, son las más convenientes para trabajar con ellas.

Los detalles importantes que se deben considerar, son:

1. ¿Qué sistemas operativos están instalados en los equipos en donde se ejecutará el futuro sistema?

El Centro de cómputo, cuenta con equipos que trabajan con la plataforma Windows (98/NT4/2000/XP). El sistema en el que se instalará la aplicación tiene Windows XP.

2. ¿Cuáles son las especificaciones de hardware de los equipos (CPU, memoria, espacio en disco duro, etc.) donde se ejecutará el futuro sistema?

Celeron a 2.6 GHz

256 MB RAM

Disco Duro de 40 GB

3. ¿Los equipos cliente tienen acceso a Internet?

No todos los equipos cuentan con acceso a Internet, sin embargo, todos tienen la posibilidad de estar conectados en una red local, pues todas cuentan con tarjeta de red.

Un punto importante que no debe pasar desapercibido para tal elección, es considerar los recursos de Software con los que ya cuenta en este caso el centro de cómputo, y la experiencia del grupo desarrollador.

Recordando cuales son estas herramientas, encontramos a Microsoft Office 2003, Sistemas Operativos Windows 9x/NT, y herramientas de desarrollo como Visual Studio 6.0, Visual Studio.Net, SQL 2000 Server, Borland C.

Selección de Herramientas para el desarrollo del sistema

¿Que metodología elegir?

Las opciones de las metodologías que se presentaron en el capítulo anterior, tienen como finalidad, modelar el problema que se presenta para comprenderlo mejor. Actualmente el desarrollo de software con orientación a objetos esta siendo muy utilizado, ya que se modela el problema en términos de tipos de objetos y esto permite que cuando el problema real cambia, el sistema a desarrollar sea más fácil de modificar, lo que es una gran ventaja ya que disminuye la posibilidad de abandono.

Es por ello que se elegirá una metodología Orientada a Objetos, además de que por experiencia se ha comprobado que las grandes ventajas que se mencionan acerca de la programación orientada a objetos son de gran utilidad para que el sistema, no se deje en el abandono y tenga éxito.

Algunas de las características a las que se hace referencia, son:

- Reutilización de código
- Facilidad para dar mantenimiento
- Facilidad de modificación, aún cuando ya este terminado el proyecto.

Entonces, bien se podría elegir una metodología como la de los tres grandes amigos que crearon el Proceso Racional Unificado (Rumbaugh, Jacobson, Booch), y que, como se observo en el capítulo anterior cada uno tiene su propia metodología, sin embargo cada una de ellas, no cubren todo el ciclo de vida de un proyecto, y aunque es cierto que unas fases son más importantes que otras, todas, hasta el mantenimiento merecen ser consideradas. Es por ello, que se eligió GRAPPLE, una metodología cuyas ideas no son originales, sino una condensación de ideas de varias personas y se asemejan mucho a las de los tres amigos, además de que cubre todas las fases importantes en el desarrollo de software y fue implantada considerando UML para los resultados de cada uno de sus segmentos.

Además de dichas razones se presentan:

- La metodología GRAPPLE es orientada a objetos.
- Es secuencial en el sentido de que el primer segmento es la de recopilación de necesidades, seguida por el análisis, el diseño, desarrollo y distribución, donde todos los segmentos son importantes.
- Es flexible a las modificaciones, ya que cualquier modificación que se desee realizar en cualquiera de los segmentos, se puede hacer regresando a los segmentos anteriores, en donde los cambios son minúsculos.
- Está metodología es soportada por UML.

- Da oportunidad -con un poco de creatividad-, de agregar ideas propias respecto a lo que funcionará en una organización en particular, puede sustraer los pasos incluidos que no funcionen.
- Los segmentos de GRAPPLE van de lo general a lo específico. Comienza con la asimilación conceptual del dominio, trasciende a la funcionalidad de alto nivel, profundiza en los casos de uso, depura los modelos, diseña, desarrolla y distribuye el sistema.
- Dedicar tanto tiempo como sea necesario en el análisis y diseño, para que la codificación se realice sin problemas.
- Cada uno de sus segmentos de GRAPPLE consta de diversas acciones, y en cada uno da por resultado un producto de trabajo. Los diagramas de UML son esos productos de trabajo.

¿Que herramientas elegir?

Una vez elegida la metodología GRAPPLE, automáticamente se acepto incluso de UML como herramienta de apoyo, pues como se mencionó en cada segmento de esta metodología se obtiene un diagrama que corresponde a este.

UML con sus diagramas toca todas las fases del ciclo de vida de un proyecto, es decir, para cada fase, se obtienen determinados diagramas, los cuales son útiles para las siguientes fases a desarrollar.

Además de ser el Lenguaje más utilizado en la industria de desarrollo de software, ofrece las siguientes ventajas; que ayudan a que el sistema o software a desarrollar tenga éxito.

- Sus diagramas ofrecen una notación estándar para diseños sólidos en la industria del software.
- Es un enlace entre quienes tienen la idea (usuarios finales) y el o los desarrolladores, ya que captura las ideas que se tienen sobre el futuro sistema para poderlas transmitir o comunicarlas al grupo desarrollador.
- Cada diagrama tiene diferentes fines dentro del proceso de desarrollo del sistema.
- Debido a que UML es un lenguaje, cuenta con reglas para combinar su simbología.
- Modela o describe lo que el sistema hará, pero no dice como implementarlo.
- Los diagramas más comunes del UML, como el de actividades, de clases, de secuencias, de colaboraciones, etc., permiten que la comprensión del sistema a desarrollar sea más fácil, y las modificaciones sean igualmente sencillas.

- Dedicar el tiempo necesario en las primeras fases (análisis y diseño) para que la codificación sea la parte más sencilla.
- Es independiente de los lenguajes de programación.
- La simbología que maneja, es sencilla de comprender.
- La curva de aprendizaje es muy baja.
- Es independiente a la herramienta computacional (Visio, Rational Rose, UML Modeler, etc.) con la que se deseen crear los diagramas, siempre y cuando sean orientadas al desarrollo de software.
- Sus diagramas son sencillos de comprender por los usuarios finales.

Herramienta computacional: Microsoft Office Visio

Existen diversos fabricantes que cuentan con paquetes que permiten generar diagramas UML. Entre el más usado está Rational Rose, aunque tiene otros dignos contendientes.

Sin embargo para aprovechar al máximo con los recursos que se tienen se usará Microsoft office Visio 2003, además de que:

- Es fácil de usarse, ya que se maneja como cualquier programa de Office.
- Es compatible con los demás programas de Microsoft Office.
- Se pueden importar diagramas a un procesador de textos como Microsoft Office Word sin ningún problema.
- Se pueden editar los diagramas hechos en Microsoft Office Visio, desde Microsoft Office Word.
- Es sencillo de instalar.
- Ocupa poco espacio en disco duro.
- Maneja todos los diagramas de UML.
- La creación de los diagramas es muy sencilla, ya que sólo basta con arrastrarlos y soltarlos al panel de trabajo.
- Los símbolos de UML están organizados, según los o el diagrama que se desee realizar. Por ejemplo para realizar un diagrama de actividades, todos los símbolos necesarios están en el mismo grupo, así, como también para hacer los diagramas de casos de uso, y los de secuencias, etc.

Lenguaje de programación

Lo preferible al desarrollar un sistema es que sea lo más intuitivo y sencillo posible, para el usuario final, así la curva de aprendizaje será muy baja. Para lograr tal objetivo habrá que consultar varias veces al usuario final, para que quede como el desee. Lo ideal para hacer modificaciones después de terminar un sistema es desarrollarlo en un lenguaje orientado a objetos.

Esta es una de las razones por las que se seleccionó un lenguaje orientado a objetos.

Además de que la programación orientada a objetos (POO – Object Oriented Programming) es un método de desarrollo en el que la estructura de los programas se basa en la interacción entre objetos para la realización de una tarea. Si prestamos atención en cómo realizamos nuestras tareas cotidianas, observamos que interactuamos con un mundo orientado a objetos.

Es por ello, que se optará por realizar la parte de codificación con Visual Basic .NET, pues además de ser un recurso con el que ya se cuenta, tiene muchas ventajas, que a continuación se mencionan:

- Es un novedoso modelo de computación distribuida en Internet. Aún así, no es un punto crítico si no se cuenta con Internet, ya que Internet se usa para que la información pueda ser accedida desde cualquier equipo independientemente del hardware y el Sistema Operativo que se tenga, pero los sistemas creados en .Net seguirán funcionando.
- Quita las barreras existentes entre distintas aplicaciones, entre sistemas, entre los sistemas y la gente que los usa, entre organizaciones, ya que la plataforma .NET, brinda la posibilidad de programar en diversos lenguajes soportados por .NET para el desarrollo de un proyecto, además de que brinda la posibilidad de ejecutar cualquier aplicación independientemente del Sistema Operativo, etc.
- Utiliza los servicios Web como un medio para poder conectar diferentes sistemas operativos, dispositivos físicos, información y usuarios. Un ejemplo muy claro de estos servicios son los teléfonos celulares que pueden navegar en Internet, y que hacen uso de código o servicios que se encuentran en algún servidor fuera o dentro de la empresa, en cualquier sistema operativo y plataforma de hardware.
- Se aprovecha el incremento de la capacidad de los nuevos dispositivos móviles llamados "Smart Devices." Para que los usuarios hagan uso de la funcionalidad que tienen los servicios Web con interfaces cada vez más sencillas.
- El framework se instala como un componente aparte en Windows 2000, mientras que en Windows XP y las futuras versiones de Windows –como Windows Vista- lo incorporan directamente en el sistema operativo, como por ejemplo Windows Server 2003 o Windows .NET CE. Para los dispositivos móviles se llama .Net Compact Framework.

- Permite integrar múltiples lenguajes¹⁰. ya que además de los lenguajes nativos de Visual Studio .NET hay muchos otros como por ejemplo PERL, Cobol, RPG.
- Evita reescribir el software para adaptarlo a las distintas plataformas tanto de software como de hardware.
- No es necesario tener instalado Visual Studio .NET en el equipo cliente, para poder ejecutar una aplicación hecha en este.
- Para ejecutar una aplicación, sólo es suficiente con instalar el .NET Framework en el equipo cliente.
- Cuenta con un IDE (Interfaz de Desarrollo Integrado), para que los programadores logren realizar aplicaciones con mayor facilidad y además estas sean eficaces, ya que incluye:
 - Un editor cuya función es realizar comprobaciones sintácticas, auto-completar y resalte en color.
 - Un mismo IDE para todos los lenguajes .NET.
 - Un sistema de depuración exhaustivo en donde se pueden establecer puntos de interrupción, avanzar paso a paso por el código, además de ver y modificar el valor de las variables.
 - Un sistema de documentación de ayuda integrado.
 - Desarrollo de GUI (Interfaces Gráficas de Usuario) simplificado mediante operaciones de arrastrar y soltar.
 - Edición XML y HTML.
 - Herramientas de distribución automatizada que se integran con Windows Installer.
 - Cuenta con la interfaz completamente personalizable y ampliables.
- Encapsula la complejidad, es decir en un programa se usan objetos y no se tiene que escribir su código, por ejemplo cuando en un formulario se usa un control, se escribe código en el evento clic y se ejecuta el programa. Durante la ejecución al hacer clic en el control, el control responde al evento clic y ejecuta el código asociado, disminuyendo así, el tiempo de programación.

Con Visual Basic .NET, los programadores pueden seguir aprovechando sus conocimientos y capacidades para crear la próxima generación de aplicaciones y servicios Web XML.

¹⁰ Especificación aprobada como estándar por la ECMA (European Computer Manufacturers Association), una organización Europea en la que se analizan muchos de los estándares más comunes en la industria del software

Gestor de Bases de Datos Microsoft SQL Server 2000

Se utilizará Microsoft SQL Server 2000, ya que es un servidor .Net Corporativo, es decir, es soportado nativamente por la plataforma. Net, pues provee la infraestructura para su construcción, instalación y operación y ya se cuenta con el software.

Además de aprovechar al máximo las características de la plataforma .NET, ya que sobre las clases base del .NET Framework, se encuentran implementadas un conjunto de clases como ADO.NET. ADO.NET utiliza los datos con un formato de estructura XML. Utilizando este tipo de estructura resulta más sencillo transferir información el Protocolo de Transferencia de Hipertexto (HTTP) y compartir datos entre sistemas heterogéneos.

Microsoft ofrece compatibilidad con varios proveedores. A los datos almacenados en Microsoft SQL Server 7.0 y versiones posteriores pueden accederse mediante el proveedor de datos SQL.

Por lo tanto para la realización de SACC, en resumen se utilizará lo siguiente:

- Programación Orientada a Objetos
- Metodología GRAPPLE
- UML
- Microsoft Office Visio
- Visual Basic .NET –que viene el Visual Studio .Net-
- SQL Server 2000
- Sistema Operativo Windows
-

4

4. Análisis y Diseño

Un dominio es todo aquel espacio en el que se presenta una problemática, en este caso es el centro de cómputo de la Preparatoria 9 "Pedro de Alba".

Después de determinar los requisitos del sistema y los usuarios que interactuarán con el sistema, el siguiente paso es desarrollar detalladamente los casos de uso.

Desarrollo de los Casos de Uso

Al terminar esta fase se contará con una clara idea de lo que el sistema realizará desde la perspectiva de los usuarios finales. El desarrollo de SACC se orientará en los Casos de Uso, debido a que estos indicarán las rutas de acceso al programa.

Aunque no hay una forma específica para establecer un análisis para los casos de uso, los puntos listados a continuación dan un panorama general para un desarrollo claro y preciso.

Recordando qué, un caso de uso es una colección de situaciones en donde cada una de ellas es una secuencia de pasos. Para cada escenario de un caso de uso, se debe mencionar:

- Una breve descripción del escenario
- Las conjeturas del escenario
- El actor inicial de caso de uso
- Las condiciones previas al caso de uso
- Los pasos relacionados con el sistema en el escenario
- Las condiciones resultantes
- El actor beneficiado

Cada uno de los paquetes mencionados en el capítulo anterior, se analizarán por separado, es decir, cada caso se tratará con el equipo involucrado, ya que cada miembro del equipo cuenta con el mejor conocimiento de los procesos a su cargo. Sólo por mencionar un ejemplo el de los administradores se tratará con los administradores, etc.

Esta parte es la más compleja para los usuarios finales, ya que se les pide tomar algunas consideraciones de lo que será el diseño del sistema.

Casos de Uso de SACC

Para iniciar a trabajar con SACC, se debe autenticar el administrador. El caso de uso es:

Autenticación

Descripción:

Cuando un técnico o encargado llega al centro de cómputo, deberá autenticarse para poder trabajar con el sistema Administrador de Centros de Cómputo.

Conjeturas:

Se asume la existencia de una interfaz que permite introducir el nombre de usuario (login) y la contraseña (password) del técnico o encargado en turno.

Condición previa:

Iniciar SACC.

Condición resultante:

Autenticar al Administrador para que pueda hacer uso de SACC.

El actor beneficiado es:

El técnico o encargado.

Los pasos del caso son:

1. Se inicia SACC.
2. Se activa la interfaz para autenticar al técnico o encargado en turno.
3. El Administrador introduce su nombre de usuario y su contraseña.
4. Se activan las opciones de entrar o cancelar.

Una vez autenticado el administrador para iniciar con las actividades de SACC. Se analizarán los casos de uso enfocados a los usuarios, que a la vez se subdivide en los siguientes casos de uso:

Dar de Alta un Administrador
Eliminar un Administrador
Editar el registro de un Administrador
Cambio de contraseña

Dar de alta un Administrador

El sistema contará con una Base de Datos que contendrá a los administradores dados de alta en el sistema.

Descripción:

Registrar o agregar a la Base de Datos de los Administradores de SACC, a un nuevo Administrador.

Conjeturas:

Se asume que existe una interfaz la cual permite a un administrador dar de alta a otro Administrador, es por ello como ya se mencionó, SACC cuenta con una contraseña de Administrador por default.

Condición previa:

Que se necesite registrar a un nuevo Administrador en el Sistema.

Condición resultante:

Dar de alta o agregar a un administrador y quede registrado en la Base de Datos Administradores de SACC.

El actor beneficiado es:

Administrador

Los pasos para el caso de uso son:

1. Activar la interfaz para dar de alta a un administrador.
2. Se activa la interfaz para dar de alta a un administrador.
3. El administrador captura el login, Nombre completo del nuevo administrador y una contraseña –longitud máxima de 10 caracteres-, y confirma la contraseña.
4. Dar clic en el botón Agregar un administrador.
5. El sistema registra al Administrador en la Base de Datos Administradores.

Eliminar un Administrador

Este caso auxilia a la seguridad de SACC, para cuando un administrador ya no vaya a utilizar el sistema.

Descripción:

Dar de baja o eliminar de la Base de Datos Administradores a un administrador.

Conjeturas:

Asumimos que el administrador ya esta registrado y que hay una interfaz que permite dar de baja a un administrador.

Condición previa:

Que el administrador exista en la Base de Datos Administradores y se desee borrar dicho registro.

Condición resultante:

Eliminar el registro de un administrador en la Base de Datos administradores y quede actualizada dicha Base, exceptuando la cuenta con la que viene por default el sistema.

El actor beneficiado es:

Administrador.

Los pasos para el caso de uso son:

1. Activar la interfaz que permita eliminar el registro en la Base de Datos Administradores.
2. Aparece la interfaz que permita eliminar el registro en la Base de Datos Administradores.
3. SACC trae todos los login de los administradores ya registrados en una interfaz.
4. El administrador selecciona el que desea eliminar.
5. SACC trae el login y nombre del administrador a eliminar.
6. El administrador acepta para realizar la eliminación.
7. El administrador confirma si desea eliminar o no al administrador seleccionado.
8. El sistema actualiza la Base de Datos administradores.

Editar el registro de un Administrador

Este caso ayuda a la modificación del registro de un administrador.

Descripción:

Modificar el registro de un administrador de la Base de Datos Administradores.

Conjeturas:

Asumimos que el administrador ya esta registrado y que hay una interfaz que permite editar el registro de un administrador.

Condición previa:

Que el administrador exista en la Base de Datos Administradores y se desee modificar dicho registro.

Condición resultante:

Modificar el registro de un administrador en la Base de Datos administradores y quede actualizada dicha Base, exceptuando la cuenta con la que viene por default el sistema.

El actor beneficiado es:

Administrador.

Los pasos para el caso de uso son:

1. Activar la interfaz que permita editar el registro en la Base de Datos Administradores.
2. Aparece la interfaz que permita editar el registro en la Base de Datos Administradores.

3. SACC trae todos los login de los administradores ya registrados en una interfaz.
4. El administrador selecciona el que desea editar.
5. SACC trae el login y nombre del administrador a editar.
6. El administrador acepta para realizar la edición.
7. El sistema actualiza la Base de Datos administradores.

Cambio de contraseña

Para el cambio de contraseña por seguridad, para realizarlo es necesario conocer la contraseña anterior.

Descripción:

Cambiar la contraseña del Administrador actual de SACC.

Conjeturas:

Se asume que el administrador esta registrado ya en la Base de Datos administradores y que hay una interfaz que permite cambiar la contraseña de un administrador actual.

Condición previa:

Que el administrador exista en la Base de Datos Administradores y que haya un administrador en sesión actualmente.

Condición resultante:

Cambiar la contraseña de un administrador.

El actor beneficiado es:

Administrador.

Los pasos para el caso de uso son:

1. Activar la interfaz para cambiar la contraseña.
2. Se activa la interfaz para cambiar la contraseña.
3. El administrador capturará la contraseña anterior.
4. El administrador capturará una contraseña nueva y la confirmará volviendo a capturarla.
5. Dar clic en el botón cambiar contraseña.
6. El sistema cambia la contraseña del administrador.

Alguna de las tareas del administrador es realizar diversos tipos de reportes o Informes, como por ejemplo acerca de los usuarios, inventarios, salas y servicios, etc., los cuales cubren determinados periodos.

Realizar Reportes

Realizar reportes

Descripción:

Generar reportes o informes de usuarios, inventario, salas, servicio, etc., que están registrados en SACC, indicando los parámetros necesarios.

Conjeturas:

Se asume que existe una interfaz para cada uno de los tipos de reportes que se pueden realizar.

También que existe una interfaz la cual permite elegir los parámetros deseados para generar el reporte solicitado y una Bases de Datos que contienen información referente a estos.

Condición previa:

Que exista la necesidad de conocer diferentes datos acerca de los usuarios, equipo, salas, servicios, etc.

Condición resultante:

Generar un reporte de usuarios, Equipos, Salas, Servicios, etc., con los parámetros deseados.

El actor beneficiado es:

El administrador.

Los pasos del Caso de Uso son:

1. Activar la interfaz según el tipo de reporte deseado.
2. Se activa la interfaz según el tipo de reporte deseado.
3. Se activa la interfaz para elegir los parámetros del reporte deseado.
4. El administrador elige y/o captura los parámetros deseados.
5. El sistema genera el reporte.

Ahora se analizaran los casos de uso relacionados con el Equipo del centro de cómputo.

Los casos correspondientes serán:

Dar de alta un servicio

Eliminar un servicio

Editar el registro de un servicio

Ver todos los servicios

Crear relación Equipo-Servicio

Eliminar relación Equipo-Servicio

Crear relación Ubicación (sala)-Servicio

Eliminar relación Ubicación (sala)-Servicio

Dar de alta un servicio.

Como mencionamos, el centro de cómputo puede en cualquier instante dar de alta un servicio.

Descripción:

Dar de alta un servicio del centro de cómputo para contar con una relación actualizada de los servicios con los que cuenta el centro de cómputo.

Conjeturas:

Se asume que hay una interfaz la cual permitirá dar de alta los servicios que hay dentro del centro de cómputo.

También asumimos que hay una Base de Datos de servicios, la cual va almacenar dichos datos.

Condición previa:

Requerir la relación actualizada de los servicios con los que cuenta el centro de cómputo y necesitar dar de alta un servicio no existente en la respectiva Base de Datos.

Condición resultante:

Que queden registrados los servicios en la respectiva Base de Datos, y mantener actualizado a SACC.

El actor beneficiado es:

Administrador

Los pasos para el caso de uso son:

1. Activar la interfaz que permita dar de alta un servicio.
2. Se activa la interfaz que permita dar de alta un servicio.
3. El administrador captura el servicio a dar de alta y si es necesario alguna observación.
4. El administrador acepta dar de alta el servicio.
5. SACC lo agrega en la Base de Datos servicios.

Dar de baja un servicio

Descripción:

Dar de baja un servicio del centro de cómputo para contar con una relación actualizada de los servicios con los que cuenta el centro de cómputo.

Conjeturas:

Se asume que hay una interfaz la cual permitirá dar de baja los servicios que hay dentro del centro de cómputo.

También asumimos que existe una Base de Datos de los servicios, que ya contiene dichos datos.

Condición previa:

Que el registro del servicio exista en la Base de Datos respectiva y se desee eliminar.

Condición resultante:

Que sean eliminadas, los servicios dentro de las respectivas Bases de Datos de SACC.

El actor beneficiado es:

Administrador

Los pasos para el caso de uso son:

1. Activar la interfaz que permita dar de baja un servicio.
2. Se activa la interfaz que permite dar de baja un servicio.
3. El administrador elige el servicio que desea dar de baja.
4. El sistema trae un resumen del servicio a dar de baja, mostrando el nombre del servicio y sus observaciones.
5. El administrador elige el nombre del servicio que se va a dar de baja.
6. El administrador acepta la baja del servicio.
7. SACC elimina el registro de la respectiva Base de Datos.

Editar el registro de un servicio

Descripción:

Modificar el registro de un servicio.

Conjeturas:

Se asume que hay una interfaz que permite editar el registro de un servicio. También asumimos que hay una Base de datos que tiene ya registrados tales datos.

Condición previa:

Que se desee modificar el registro de un servicio que ya existe en la respectiva Base de Datos.

Condición resultante:

Que se modifique el registro del servicio deseado.

El actor beneficiado es:

Es el administrador.

Los pasos para el caso de uso son:

1. Activar la interfaz que permite editar los registros de los servicios.
2. Se activa la interfaz que permite editar los registros de los servicios.
3. El administrador elige el servicio que desea editar.
4. El administrador da clic en editar.
5. Se activan los campos nombre del servicio y observaciones del servicio seleccionado.
6. Se acepta la edición.
7. SACC actualiza el registro en la respectiva Base de Datos.

Ver todos los servicios

Este caso permite visualizar todos los servicios que hay en el centro de cómputo.

Descripción:

Ver todos los servicios que se dan en el centro de cómputo.

Conjeturas:

Se asume que hay una interfaz la cuál permitirá ver todos los servicios que hay dentro del centro de cómputo. También se asume que hay una Base de Datos con tales servicios.

Condición previa:

Verificar la relación actualizada de los servicios con los que cuenta el centro de cómputo.

Condición resultante:

Ver los servicios registrados en SACC.

El actor beneficiado es:

Administrador

Los pasos para el caso de uso son:

1. Activar la interfaz que permita ver los servicios.
2. Se activa la interfaz que permita ver los servicios.
3. El administrador elige el servicio que desea observar.
4. SACC muestra el nombre del servicio y sus respectivas observaciones.
5. El administrador acepta para terminar de ver el servicio.

Crear relación Equipo-Servicio

Descripción:

Crear la relación entre un equipo y un servicio. Esto con la finalidad de realizar más rápido la búsqueda de los servicios en los equipos del centro de cómputo.

Conjeturas:

Se asume la existencia de la Base de Datos equipo-servicio. Además de que existe una interfaz la cual permite crear una relación entre equipo-servicio.

Condición previa:

Contar con la Bases de Datos equipo-servicio. Además de tener la necesidad de agregar un servicio en uno de los equipos del centro de cómputo.

Condición resultante:

Crear la relación, equipo-servicio.

El actor beneficiado es:

Administrador

Los pasos para el caso de uso son:

1. Activar la interfaz para crear la relación equipo-servicio.
2. Se activa la interfaz para crear la relación equipo-servicio.
3. SACC trae los equipos que hay en el centro de cómputo.
4. El administrador elige un equipo.
5. SACC trae los servicios que hay en el centro de cómputo.
6. El administrador elige un servicio.
7. El administrador acepta.
8. SACC guarda la relación en una Base de Datos llamada equipo-servicio.

Eliminar relación Equipo-Servicio

Descripción:

Eliminar la relación entre equipo-servicio. Con la finalidad de contar con una relación actualizada de los servicios que se encuentran en determinado equipo.

Conjeturas:

Asumimos la existencia de la Base de Datos equipo-servicio, además de que existe una interfaz la cual permite eliminar una relación entre un equipo-servicio.

Condición previa:

Contar con las relaciones equipo-servicio.

Condición resultante:

Eliminar la relación equipo-servicio.

El actor beneficiado es:

Administrador

Los pasos para el caso de uso son:

1. Activar la interfaz para eliminar la relación equipo-servicio.
2. Se activa la interfaz para eliminar la relación equipo-servicio.
3. SACC trae los equipos que ya cuentan con una relación.
4. El administrador selecciona el equipo, al que se eliminará la relación.
5. SACC trae los servicios que tienen relación con el equipo que se eligió.
6. El administrador selecciona uno de los servicios.
7. El administrador acepta, para eliminar la relación.
8. SACC actualiza las relaciones en la respectiva Base de Datos.

Crear relación Ubicación (sala)-Servicio

Descripción:

Crear la relación entre una ubicación o sala y un servicio. Esto con la finalidad de ubicar más rápido los servicios en los equipos del centro de cómputo.

Conjeturas:

Se asume la existencia de la Base de Datos ubicación-servicio. Además de que existe una interfaz la cual permite crear una relación entre ubicación-servicio.

Condición previa:

Contar con la Bases de Datos ubicación-servicio. Además de tener la necesidad de agregar un servicio en una de las ubicaciones del centro de cómputo.

Condición resultante:

Crear la relación ubicación-servicio.

El actor beneficiado es:

Administrador

Los pasos para el caso de uso son:

1. Activar la interfaz para crear la relación ubicación-servicio.
2. Se activa la interfaz para crear la relación ubicación-servicio.
3. SACC trae las ubicaciones que hay en el centro de cómputo.
4. El administrador elige una ubicación.
5. SACC trae los servicios que hay en el centro de cómputo.
6. El administrador elige un servicio.
7. El administrador acepta.
8. SACC guarda la relación en una Base de Datos llamada ubicación-servicio.

Eliminar relación Ubicación (sala)-Servicio

Descripción:

Eliminar la relación entre una ubicación-servicio. Con la finalidad de contar con una relación actualizada de los servicios que se encuentran en determinada ubicación.

Conjeturas:

Asumimos la existencia de la Base de Datos ubicación-servicio, además de que existe una interfaz la cual permite eliminar una relación entre una ubicación-servicio.

Condición previa:

Contar con las relaciones ubicación-servicio.

Condición resultante:

Eliminar la relación ubicación-servicio.

El actor beneficiado es:

Administrador

Los pasos para el caso de uso son:

1. Activar la interfaz para eliminar la relación ubicación-servicio.
2. Se activa la interfaz para eliminar la relación ubicación-servicio.
3. SACC trae las ubicaciones que ya cuentan con una relación.
4. El administrador selecciona una ubicación, al que se le eliminará la relación.
5. SACC trae los servicios que tienen relación con la ubicación que se eligió.
6. El administrador selecciona uno de los servicios.
7. El administrador acepta, para eliminar la relación.
8. SACC actualiza las relaciones en la respectiva Base de Datos.

Ahora se analizarán los casos de uso enfocados al Equipo que hay dentro del centro de cómputo. Hay que considerar que el equipo esta conformado de varios dispositivos, los cuales deberán tener las mismas características en el sistema.

Por lo tanto los casos son:

Dar de Alta (equipos o CPU's o monitores o teclados o mouse o extras y/o ubicaciones).

Eliminación (equipos o CPU's o monitores o teclados o mouse o extras y/o ubicaciones).

Dar de baja (CPU's o monitores o teclados o mouse, extras).

Editar el registro de (equipos o CPU's o monitores o teclados o mouse o extras y/o ubicaciones).

Ver todos (equipos o CPU's o monitores o teclados o mouse o extras y/o ubicaciones).

Dar de Alta (equipos o CPU's o monitores o teclados o mouse o extras y/o ubicaciones).

Es importante hacer énfasis en la importancia que representa un inventario actualizado del centro de cómputo. Una de las más complejas e importantes tareas para un administrador de un centro de cómputo es la realización del inventario, en donde SACC lo irá creando con las altas de los equipos en el sistema.

Descripción:

Dar de alta en la respectiva Base de Datos de SACC, los equipos o CPU's o monitores o teclados o mouse y/o extras que hay dentro del centro de cómputo, con algunas características como: Número de equipo, Marca, Modelo, Procesador, Memoria, Número de Inventario, Número de Serie, Tipo, Observaciones, Ubicación -sala-, estado –sirve o no-.

Conjeturas:

Asumimos que existe una Base de Datos en la que se van a guardar las características necesarias de los Equipos.

Condición previa:

Requerir de un mayor control sobre los Equipos que hay dentro del centro de cómputo.

Condición resultante:

Registrar todos los Equipos que existen dentro del centro de cómputo.

El actor beneficiado es:

El administrador

Los pasos del Caso de Uso son:

1. Activar la interfaz para dar de alta al respectivo componente.
2. Aparece la interfaz para dar de alta al respectivo componente.
3. El administrador captura las características necesarias en el sistema.
4. El administrador da de alta al respectivo componente en el sistema.
5. El sistema guarda tal información en la respectiva Base de Datos.

Dar de baja (CPU's o monitores o teclados o mouse, extras).

Descripción:

Para dar de baja un componente ya sea por petición, mal estado, obsolescencia, etc., no es necesario capturar todas características -ya que deberá encontrarse en la respectiva Base de Datos, sólo será necesario asegurarse de qué componente es, cotejando los datos y tomando algunas características y eliminarla.

Conjeturas:

Asumimos que el componente que se quiere dar de baja existe en la Base de Datos correspondiente. También asumimos que hay una interfaz que permite hacerlo.

Condición previa:

Que el componente que se desea dar de baja exista en la respectiva Base de Datos Equipos.

Condición resultante:

Es que el componente se elimine de la respectiva Base de Datos.

El actor beneficiado es:

El administrador.

Los pasos del Caso de Uso son:

1. Activar la interfaz para dar de baja un CPU o monitor o teclado o mouse o extras.
2. Se activa la interfaz para dar de baja un CPU o monitor o teclado o mouse o extras.
3. El administrador captura las características necesarias.
4. El administrador da de baja un CPU u monitor o teclado o mouse o extras.
5. El sistema borra el registro de la respectiva Base de Datos para tener a SACC actualizado.

Nota: El mouse, teclado, monitor, y demás dispositivos que componen al equipo de cómputo tendrán el mismo número que el CPU, entendiendo a este como el número de equipo.

Editar el registro de (equipos o CPU's o monitores o teclados o mouse o extras y/o ubicaciones).

Descripción:

Poder editar los datos de los equipos o CPU's o monitores o teclados o mouse o extras y/o ubicaciones que se encuentran registrados en la respectiva Base de Datos.

Conjeturas:

Asumimos que hay una interfaz que permite editar algún registro en la Base de Datos respectiva.

Condición previa:

Que exista el registro respectivo en la Base de Datos.

Condición resultante:

Modificar alguno de los datos en la Base de Datos respectiva.

El actor beneficiado es:

El administrador.

Los pasos del Caso de Uso son:

1. Activar la interfaz que nos permita editar algún registro en la Base de Datos respectiva.
2. Aparece la interfaz que nos permita editar algún registro en la Base de Datos respectiva.
3. El administrador edita las características necesarias.
4. El sistema actualiza la Base de Datos respectiva.

Eliminación (equipos o CPU's o monitores o teclados o mouse o extras y/o ubicaciones).

Descripción:

Poder eliminar permanentemente los datos de los equipos o CPU's o monitores o teclados o mouse o extras que se encuentran registrados en la respectiva Base de Datos.

Conjeturas:

Asumimos que hay una interfaz que permite eliminar algún registro en la Base de Datos respectiva.

Condición previa:

Que exista el registro respectivo en la Base de Datos.

Condición resultante:

Eliminar algún dato en la Base de Datos respectiva.

El actor beneficiado es:

El administrador.

Los pasos del Caso de Uso son:

1. Activar la interfaz que permita eliminar permanentemente algún registro en la Base de Datos respectiva.
2. Aparece la interfaz que permita eliminar algún registro en la Base de Datos respectiva.
3. El administrador elimina el registro necesario.
4. El sistema actualiza la Base de Datos respectiva.

Ver todos (equipos, CPU's, monitores, teclados, mouse, extras u ubicaciones).

Este caso permite visualizar todos los equipos o CPU's o monitores o teclados o mouse o extras y/o ubicaciones que hay en el centro de cómputo.

Descripción:

Ver todos los equipos o CPU's o monitores o teclados o mouse o extras y/o ubicaciones que hay en el centro de cómputo.

Conjeturas:

Se asume que hay una interfaz la cuál permitirá ver todos los equipos o CPU's o monitores o teclados o mouse o extras y/o ubicaciones que hay en el centro de cómputo. También se asume que existe una Base de Datos con los respectivos registros.

Condición previa:

Verificar la relación actualizada de los equipos o CPU's o monitores o teclados o mouse o extras y/o ubicaciones con los que cuenta el centro de cómputo.

Condición resultante:

Ver los equipos o CPU's o monitores o teclados o mouse o extras y/o ubicaciones registrados en SACC.

El actor beneficiado es:

Administrador

Los pasos para el caso de uso son:

1. Activar la interfaz que permita ver los equipos o CPU's o monitores o teclados o mouse o extras y/o ubicaciones que hay en el centro de cómputo.
2. Se activa la interfaz que permita ver los equipos o CPU's o monitores o teclados o mouse o extras y/o ubicaciones que hay en el centro de cómputo.
3. El administrador elige los elementos que desea observar.
4. SACC muestra el nombre del o los elementos y sus respectivas observaciones.

Los casos de uso para el administrador son:

Alta de un usuario
Edición de usuario
Eliminación de usuario
Búsqueda de usuario

Alta de un usuario

Si el alumno o el usuario externo, no se encuentra en la respectiva Base de Datos, se podrán agregar.

Descripción:

Dar de alta o agregar un usuario en la respectiva Base de Datos, para que pueda hacer uso de los servicios que se ofrecen.

Conjeturas:

Asumimos que existe una interfaz la cual permite dar de alta a un usuario.

Condición previa:

Agregar un usuario en la respectiva Base de Datos para que pueda hacer uso de los servicios que ofrece el centro de cómputo.

Condición resultante:

Que el futuro usuario quede registrado en la Base de Datos de usuarios.

El actor beneficiado es:

El administrador

Los pasos del Caso de Uso son:

1. Activar la interfaz para dar de alta a un usuario.
2. Se activa la interfaz para dar de alta a un usuario.
3. El Administrador captura el Nombre, Grupo y Número de Cuenta del alumno o en su caso el nombre del usuario externo, lugar de procedencia, etc.
4. Se activa la opción para registrar al usuario.
5. El sistema registra al usuario en la Base de Datos.

Edición de usuario

Este caso ayuda a hacer alguna modificación necesaria a ciertos registros de la Bases de Datos de usuarios.

Descripción:

Cambiar algún registro de la Base de Datos usuarios que ya esté registrado.

Conjeturas:

Se asume que el usuario ya esta registrado en la respectiva Base de Datos y se desea modificar algún dato.

También asumimos que hay una interfaz que permite editar un registro en tal Base de Datos.

Condición previa:

Exista el registro en la Base de Datos y que por alguna razón se desee modificar.

Condición resultante:

Modificar el registro de un usuario para tener los datos actualizados en la Base de Datos.

El actor beneficiado es:

El administrador.

Los pasos del Caso de Uso son:

1. Activar la interfaz que permita editar los datos de algún usuario.
2. Aparece la interfaz que permita editar los datos de algún usuario.
3. El administrador captura el número de cuenta del alumno o nombre del externo.
4. El sistema trae todos los datos del alumno o externo con tal número de cuenta o nombre, ya registrados en la Base de Datos respectiva.
5. Se activa una interfaz con tales registros.
6. El administrador navega entre los datos y hace la modificación al registro deseado.
7. El sistema actualiza la Base de Datos necesaria.

Eliminación de usuario

Este caso ayuda a eliminar ciertos registros de la Bases de Datos de usuarios.

Descripción:

Eliminar algún registro que ya se encuentre registrado de la Base de Datos usuarios.

Conjeturas:

Se asume que el usuario ya está registrado en la Base de Datos y se desea eliminar el registro.

También asumimos que hay una interfaz que lo permite hacer.

Condición previa:

Que exista el registro en la Base de Datos y que por alguna razón se desee eliminar.

Condición resultante:

Eliminar el registro de un usuario para tener los datos actualizados en la Base de Datos.

El actor beneficiado es:

El administrador.

Los pasos del Caso de Uso son:

1. Activar la interfaz que permita eliminar el registro de un usuario.
2. Aparece la interfaz que permita eliminar el registro de un usuario.
3. El sistema trae todos los datos del usuario de la Base de Datos.
4. El administrador hace la eliminación del registro deseado.
5. El sistema actualiza la Base de Datos necesaria.

Búsqueda de usuario

Con ayuda de SACC la búsqueda de información de un usuario es más sencilla y más rápida.

Descripción:

Buscar un registro en la Base de Datos usuarios.

Conjeturas:

Se asume que el usuario ya esta registrado en la respectiva Base de Datos y se desea ver sus datos.

También asumimos que hay una interfaz que permite realizar la búsqueda.

Condición previa:

Que exista el registro en la Base de Datos y que por alguna razón se deseen conocer sus datos.

Condición resultante:

Buscar el registro de un usuario para tener los datos actualizados en la Base de Datos.

El actor beneficiado es:

El administrador.

Los pasos del Caso de Uso son:

1. Activar la interfaz que permita buscar los datos de algún usuario.
2. Aparece la interfaz que permita buscar los datos de algún usuario.
3. El administrador captura el número de cuenta o nombre del usuario.
4. El sistema trae todos los datos del usuario.
5. El administrador los consulta.

Un complemento del registro de un usuario y muy importante es el manejo de sanciones, ya que de ello dependerá si puede realizarlo.

Dar de alta una sanción

Editar una sanción

Dar de baja una sanción

Ver todas las sanciones

Dar de alta una sanción

Antes de poder dar servicio a un usuario, es necesario conocer si en visitas anteriores al centro de cómputo ha mostrado buen comportamiento.

Descripción:

Registrar o dar de alta una sanción.

Conjeturas:

Asumimos que existe una Base de Datos en la que se registraran las sanciones.

También asumimos que se cuenta con un interfaz para dar de alta alguna sanción.

Condición previa:

Se desea sancionar a un usuario para que no tenga derecho a préstamo de servicio.

Condición resultante:

Registrar una sanción a un usuario.

El actor beneficiado es:

Al administrador.

Los pasos del Caso de Uso son:

1. Activar la interfaz para dar de alta una sanción.
2. Se activa la interfaz para dar de alta una sanción.
3. El Administrador captura los datos necesarios para registrar la sanción.

4. El sistema registra la sanción en la Base de Datos.

Editar una sanción

Se tendrá la posibilidad realizar cualquier modificación a la sanción de un usuario.

Descripción:

Editar una sanción.

Conjeturas:

Asumimos que se tiene registrada una sanción y que se desea realizar una modificación a tal registro.

Condición previa:

Que el usuario tenga una sanción y se desee modificar.

Condición resultante:

Modificar una sanción.

El actor beneficiado es:

Administrador.

Los pasos para el caso de uso son:

1. Activar la interfaz que permita editar alguna sanción.
2. Aparece la interfaz que permita editar alguna sanción.
3. El administrador captura el identificador del usuario.
4. El administrador hace la modificación correspondiente.
5. El sistema actualiza la Base de Datos necesaria.

Dar de baja una sanción

Descripción:

Dar de baja una sanción que ya se encuentre registrada.

Conjeturas:

Se asume que la sanción ya está registrada en la Base de Datos y se desea dar de baja.

También asumimos que hay una interfaz que lo permite hacer.

Condición previa:

Que exista el registro en la Base de Datos y que por alguna razón se desee dar de baja.

Condición resultante:

Dar de baja una sanción para tener los datos actualizados en la Base de Datos.

El actor beneficiado es:

El administrador.

Los pasos del Caso de Uso son:

1. Activar la interfaz que permita dar de baja la sanción de un usuario.
2. Aparece la interfaz que permita dar de baja la sanción de un usuario.
3. El sistema trae todos los datos de la sanción.
4. El administrador hace la eliminación del registro deseado.
5. El sistema actualiza la Base de Datos necesaria.

Ver todas las sanciones

Descripción:

Ver todos las sanciones de los usuarios.

Conjeturas:

Se asume que hay una interfaz la cuál permitirá ver todas las sanciones. También se asume que existe una Base de Datos con los respectivos registros.

Condición previa:

Verificar la relación de sanciones que tienen los alumnos.

Condición resultante:

Ver todas las sanciones que se tienen registradas en SACC.

El actor beneficiado es:

El administrador

Los pasos para el caso de uso son:

1. Activar la interfaz que permita ver las sanciones de los usuarios.
2. Se activa la interfaz que permita ver las sanciones de los usuarios.
3. El administrador elige la sanción a consultar.
4. SACC muestra todos los datos sobre la sanción.

Para finalizar, es importante ver que en las conjeturas de los casos de uso, se hicieron algunas consideraciones sobre el futuro diseño del Sistema.

Aunque en todos los casos de uso se asume la existencia de una interfaz, hay que notar que aún no se especifica como lucirán, ni como se llevarán a cabo los procesos y mucho menos los detalles técnicos. El detalle es que las conjeturas ayudan a dar forma a tales interfaces.

Modelo de la estructura de clases de SACC

Una vez identificados y desarrollados los casos de uso, se puede iniciar a identificar las clases que muy probablemente se incluirán en el sistema. Para realizarlo habrá que distinguir los **sustantivos** de los procesos y casos de uso del centro de cómputo. Algunos de esos sustantivos se usarán como clases y otros como atributos, y posteriormente figuraran en un modelo llamado Modelo de Clases Inicial.

Es muy importante seguir en contacto con el usuario final, ya que permitirá afinar los procesos y casos de uso.

A continuación se mencionarán algunos sustantivos elegidos en forma singular por factibilidad y posteriormente si es necesario se agregarán otros. Aquí son indispensables el sentido común y la experiencia ya que ayudarán a la elección de los sustantivos.

Algunos sustantivos, son:

Access, aplicación, asignaturas, clases, CPU, datos, dispositivos, elementos, encargado, responsable de horarios, equipo, especificaciones técnicas, estado físico, eventos, Excel, falla física irreparable, formato de entrega, hoja de registro, hoja en blanco, hora de entrada, hora de salida, horarios, identificación, impresión, Internet, inventario, libreta de registro, marca, modelo, monitor, mouse, No. De Cta., No. de inventario, No. de Serie, nombre, obsolescencia, paquetería, plantel, programa, registro, reporte de petición, reporte de usuarios, sala, servicio, software, tarjetones numerados, técnico, ubicación, unidad de resguardos, usuario, vistas, Word.

Desarrollo del diagrama de clases inicial

Para el desarrollo del diagrama de clases inicial, se debe realizar una depuración en los sustantivos ya elegidos, ya que unos se usarán como clases, algunos como atributos y algunos otros simplemente se eliminarán.

Pero ¿Qué es una clase y un atributo? Una clase es como una plantilla para fabricar objetos, imaginémoslo como un gran molde para hacer galletas con el que podemos hacer muchas de ellas. Un atributo es una característica de una clase, un claro ejemplo de este concepto es: la forma de la clase galleta.

Considerando tales conceptos y después de hacer una depuración los sustantivos principales hasta ahora serán:

Usuario, encargado, ubicación, equipo, servicio, administrador, alumno, identificación, sanción, ocupación.

Ya que otros tantos sustantivos, se adecuan más a ser atributos, como por ejemplo: número de cuenta, número de equipo, nombre completo, hora de entrada, hora de salida.

Al valerse del sentido común para la eliminación de los demás que no entran en estas categorías, se debe tener especial cuidado. Sólo por mencionar algunos sustantivos eliminados como lo es "departamento externo", que al no ser parte física del centro de cómputo se puede eliminar, "plantel" es otro de ellos, ya que sus actividades son otras y el análisis sólo se enfocará a las actividades que se realizan dentro de este.

Ahora si se analiza la palabra encargado y técnico, se observa que sus tareas desde cualquier perspectiva se enfoca a administrar la mayor parte de las tareas que surgen dentro del centro de cómputo, por lo tanto se puede cambiar por el sustantivo Administrador.

Recordando la descripción de los procesos, se hizo mención del registro de los usuarios, en donde además de obtener algunos datos personales, también se adquieren los referentes al servicio a utilizar se puede deducir que todos estos datos son para alguna tarea específica que no se menciona, y es muy probable que se utilice para realizar diversos tipos de reportes, estadísticas, etc. incluso que en base a ello se pueda sancionar a los usuarios, por lo tanto podemos agregar una clase llamada reportes o informes.

Ahora, enfocando la vista en el sustantivo Inventario, que es el que guarda información de los equipos podría entrar en la clase reporte, por lo tanto el sustantivo inventario se puede eliminar.

También si se analiza el sustantivo Software se puede ver que es muy específico debido a que es uno de los servicios que brinda el centro de cómputo, por tal motivo puede sustituirse por uno más conveniente como por ejemplo, el sustantivo servicio.

Finalmente luego de filtrar, agregar y depurar la lista de sustantivos, ésta quedaría de la siguiente manera:

Administrador, usuario, monitor, equipo, servicio, identificación, sanción, ocupación, teclado, mouse, CPU, extras, ubicación.

Estos sustantivos conformarán el diagrama de Clases Inicial y para ello se usará la notación Camello.

La notación Camello consiste en escribir la primera letra del nombre de una clase en mayúsculas, en caso de que el nombre tuviese más de una palabra entonces se sigue la misma regla, pero sin dejar espacios en blanco entre ellas.

Por lo tanto, el diagrama de clases inicial para el dominio del centro de cómputo, se verá de la siguiente manera:

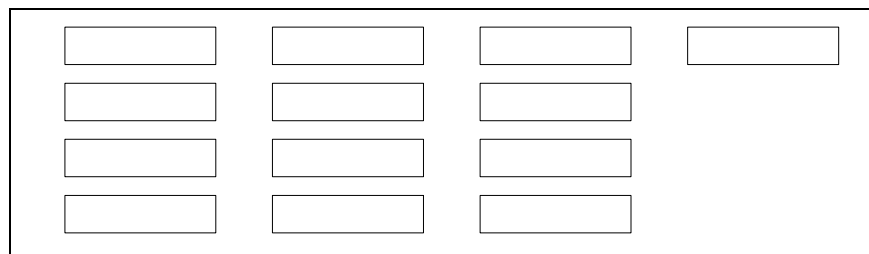
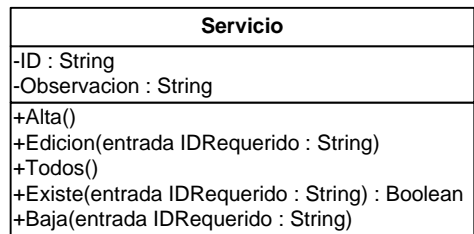
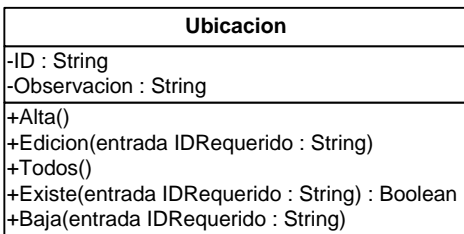
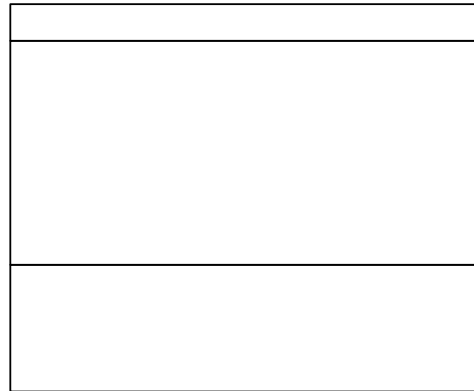
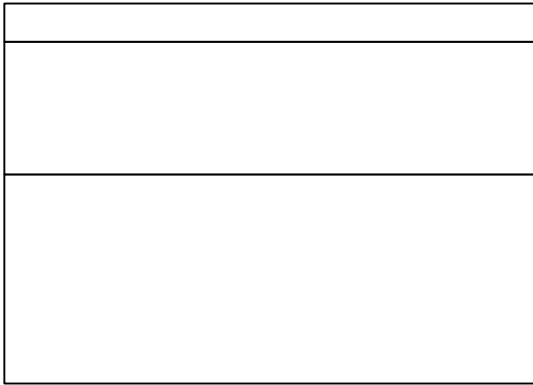


Figura 30. Diagrama de Clases Inicial para el dominio centro de cómputo

Llenado de las clases

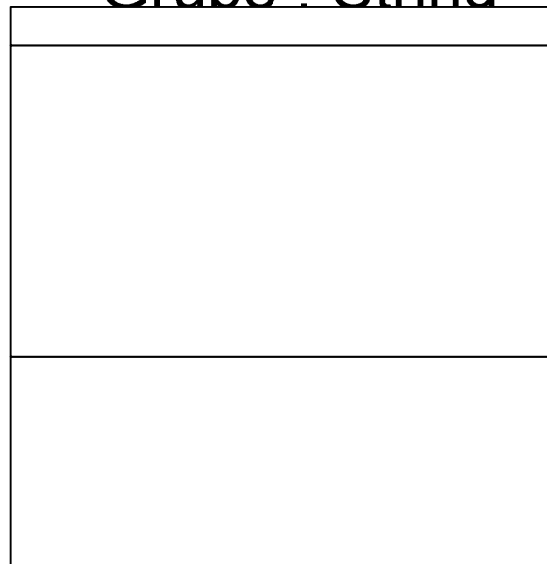
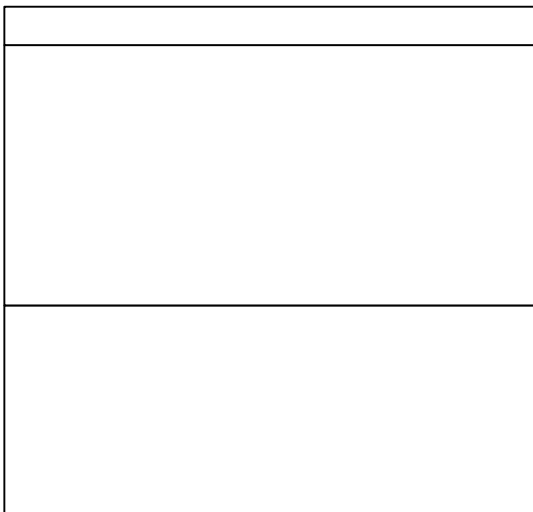
Para dar cuerpo a tales clases con las que se ha venido trabajando, se debe llevar a cabo su llenado colocando sus atributos y métodos u operaciones.

Recordando que las clases son: Administrador, usuario, monitor, equipo, servicio, identificación, sanción, ocupación, teclado, mouse, CPU, extras, ubicación. A continuación se presentan con sus respectivos atributos y métodos.



-Nombre : String

-Grupo : String



String
String

da IDRe
IDRequ
(entrada
IDReque

+ExisteN(entrada Nombre

+Baja(entrada IDRequerido

+Todos()

Mouse	
-ID : Integer -Marca : String -Modelo : String -Serie : String -Inventario : String -Funcional : Boolean -Baja : Boolean -Observacion : String	
+Alta() +Edicion(entrada IDRequerido : Integer) +Existe(entrada IDRequerido : Integer) : Boolean +ExisteI(entrada InventarioRequerido : String) : Boolean +ExisteS(entrada SerieRequerido : String) : Boolean +Todos() +BajaP(entrada IDRequerido : Integer) : Boolean +Eliminar(entrada IDRequerido : Integer) : Boolean	

Monitor	
-ID : Integer -Marca : String -Modelo : String -Serie : String -Inventario : String -Funcional : Boolean -Baja : Boolean -Observacion : String	
+Alta() +Edicion(entrada IDRequerido : Integer) +Existe(entrada IDRequerido : Integer) : Boolean +ExisteI(entrada InventarioRequerido : String) : Boolean +ExisteS(entrada SerieRequerido : String) : Boolean +Todos() +BajaP(entrada IDRequerido : Integer) : Boolean +Eliminar(entrada IDRequerido : Integer) : Boolean	

Sancion	
-ID : Integer -Usuario : String -Administrador : String -Observacion : String -Inicio : Date -Termino : Date -Vigente : Boolean	
+Alta() +Edicion(entrada UsuarioRequerido : String) +Existe(entrada UsuarioRequerido : String) : Boolean +Baja(entrada UsuarioRequerido : String) : Boolean +Todos()	

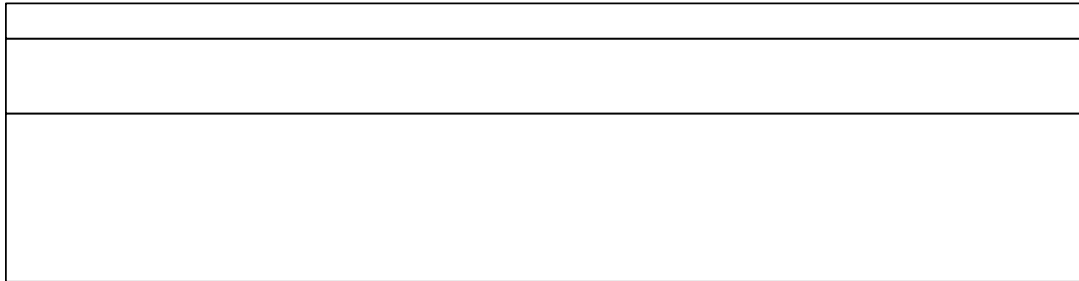
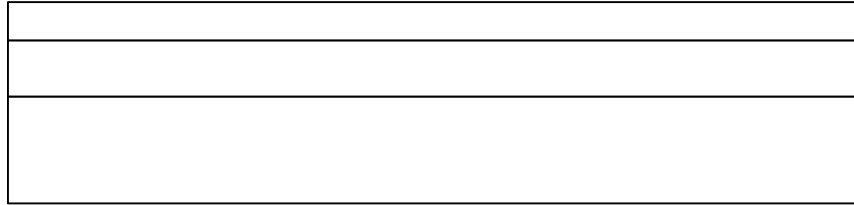


Figura 31. Clases con atributos y métodos.

Diseño de SACC

El diseño del Sistema SACC se basará en los casos de uso. Para ello es conveniente conocer primero sus componentes, puesto que son los que le darán movimiento al sistema.

Componentes del sistema

Para identificar fácilmente los componentes de SACC se usarán las conjeturas de los casos de uso.

Como se recordará, los casos de uso fueron divididos en tareas comunes, como por ejemplo en tareas relacionadas con el administrador, con los usuarios, con los equipos, los reportes, etc.

Uno de los componentes muy importantes y que se mencionan en la mayoría de las conjeturas son las interfaces. Por buena costumbre, organización y estandarización con aplicaciones Windows, es preferible la creación de una interfaz principal, que además ayudará a organizar las demás interfaces.

Otro de los componentes, también muy importantes, ya que almacenarán algunos de los datos con los que trabajará SACC, como los Servicios, Ubicaciones, Equipos, Usuarios, etc., son las Bases de Datos.

Más específicamente, si se comienza con los casos de uso del paquete administrador; se puede apreciar que el sistema necesitará interfaces para autenticar al administrador, para verificar el número de cuenta del alumno o nombre del usuario externo, agregar o editar los registros, para la verificación de las sanciones, para capturar los servicios, ubicar los equipos libres, etc.; además de una interfaz principal en la que se engloben los diferentes casos de uso.

-Equipo : Integer
-Servicio : String

+Alta()
+Eliminar(entrada
+Todos()
+Existe(entrada E

-ID : String
-Nombre : String
-Contraseña : String

+Alta()
+Edicion(entrada IDRequer
+CambioContraseña(entrac
+Existe(entrada IDRequeric

+Todos()

+Eliminar(entrada IDReque

Para terminar, con los componentes de SACC, del lado del Hardware se requerirá una Computadora personal de escritorio cuyas características sean las siguientes:

Celeron a 1.2 GHz
256 MB de RAM
Espacio en Disco Duro de 40 GB

En caso de no tener instalado SQL Server en este equipo se deberá contar con un servidor, cuyas características dependerán del Sistema Operativo de servidor que tenga, como por ejemplo, si contará con:

Sistema Operativo Windows 2000 Server:

Requerimientos mínimos:

Pentium a 133MHz
64 MB Memoria RAM
Disco Duro con 1GB de espacio libre

Requerimientos recomendados:

Pentium III a 500MH
128 MB Memoria RAM
Disco Duro con 3GB de espacio libre

Además de estos requerimientos, es necesario: Unidad de CD-ROM, Tarjeta de Red, Tarjeta de video SVGA, Ratón, Teclado.

Sistema Operativo Windows Server 2003:

Requerimientos mínimos:

Pentium a 133 MHz
128 MB Memoria RAM
Disco Duro con 1.5 GB de espacio libre

Requerimientos recomendados

Pentium III a 550 MHz
256 MB Memoria RAM
Disco Duro con 3.0 GB de espacio libre

Además de estos requerimientos, es necesario: Unidad de CD-ROM, Tarjeta de Red, Tarjeta de video SVGA, Ratón, Teclado.

De los casos de uso al Diseño de las interfaces.

Los casos de uso ayudan a visualizar la forma en como funcionará el sistema, es por ello que se dice que la interfaz ayuda en la implementación de los casos de uso.

Para esto se usarán los diagramas de secuencias que se desarrollaran en el capítulo siguiente, que girándolos se podrán observar en tres dimensiones, obteniendo así la interfaz del usuario. Como se puede observar en las Figuras 32 y 33.

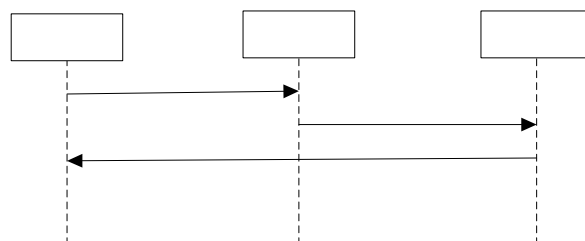


Figura 32. Diagrama de secuencias, visto en dos dimensiones.

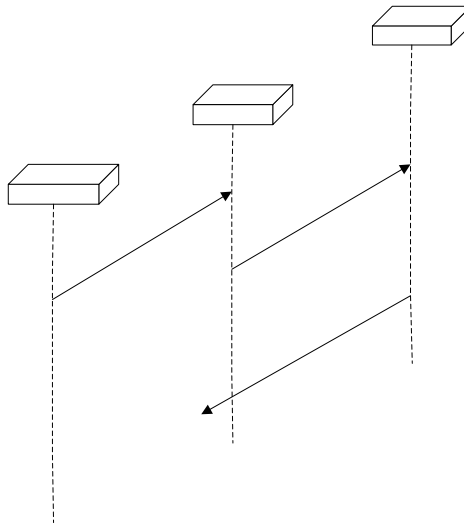


Figura 33. Diagrama de secuencias visto en tres dimensiones.

Para el desarrollo de las interfaces se necesitan conocer algunos principios para su construcción.

Luego de mucha experiencia con interfaces WIMP (Ventanas, Iconos, Menús y Dispositivos apuntadores), se pueden encontrar diversos principios generales. He aquí los principales:

1. Comprender las tareas que el usuario tiene que llevar a cabo.
2. Siempre incluir la posibilidad que el usuario cancele una acción luego de que la haya iniciado.
3. Brindar al usuario diferentes formas de realizar cada acción.
4. Los componentes relacionados deberán aparecer uno junto de otro.
5. No utilizar palabras muy técnicas.
6. Evitar el uso de fuentes cursivas y decorativas.
7. Intentar mantener el tamaño de los botones.
8. Alinear los componentes y campos de datos de izquierda a derecha.
9. En caso de tener un botón como predeterminado, habrá que destacarlo colocándolo como el primero en el conjunto.
10. Y especialmente, basarse en estándares del sistema operativo Windows.

Interfaces de SACC

El hecho de haber dividido a los casos de uso en grupos, ayudará ahora a construir la interfaz principal, que servirá para organizar las demás interfaces.

La pantalla principal servirá al usuario final para navegar por cada uno de los grupos de casos de uso, al igual que se pueda navegar dentro de un mismo grupo.

Así, considerando la forma en que agrupamos nuestros casos y los principios de construcción de las interfaces, las demás interfaces lucirán, de la siguiente manera.

La siguiente pantalla muestra una idea de lo que es el inicio de SACC, que es la interfaz que llevará al Administrador por las demás interfaces.



Figura 34. Interfaz principal de SACC.

A continuación se muestran sólo las interfaces principales, considerando que las demás son semejantes a estas.

Por ejemplo, la interfaz de Usuarios tendrá que incluir todos los casos relacionados con este grupo y su interfaz lucirá de la siguiente manera:

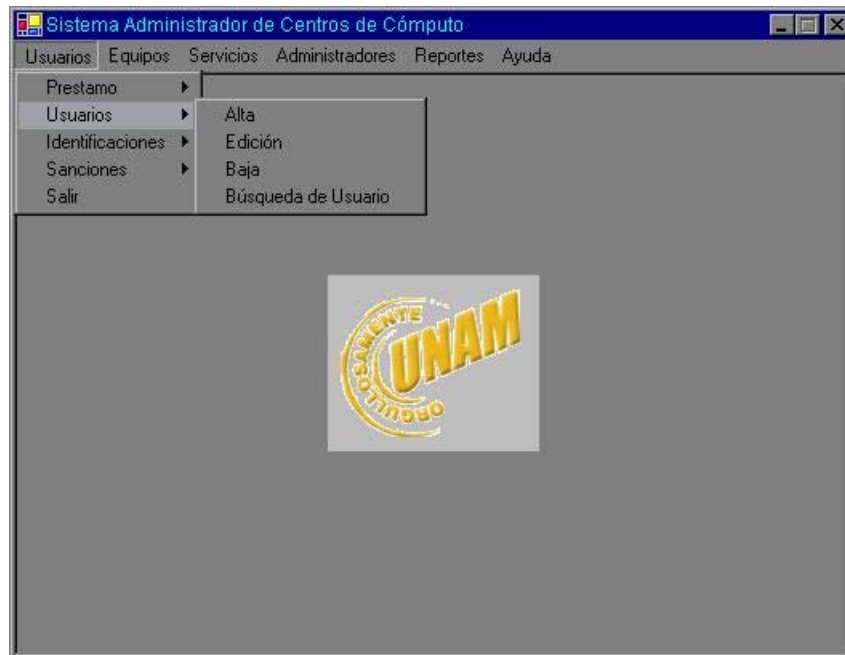


Figura 35. Interfaz principal de SACC.

Otro grupo es el de Equipo, donde la interfaz es la siguiente:



Figura 36. Interfaz principal de SACC.

Otro grupo sería el que hace referencia a los Servicios:



Figura 37. Interfaz principal de SACC.

Otro más para lo relacionado con los Administradores:

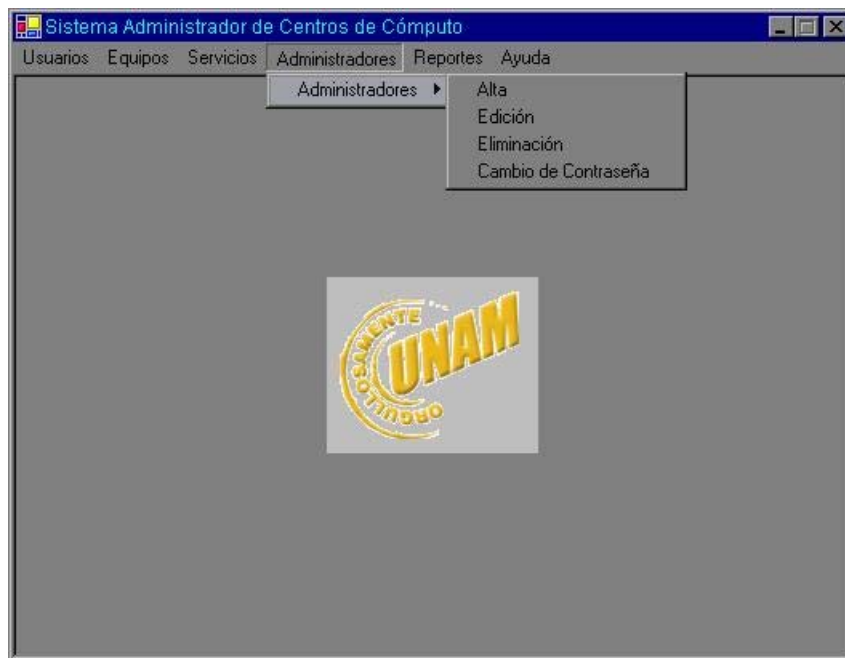


Figura 38. Interfaz principal de SACC.

Otro es el relacionado con los Reportes, donde la interfaz es la siguiente:

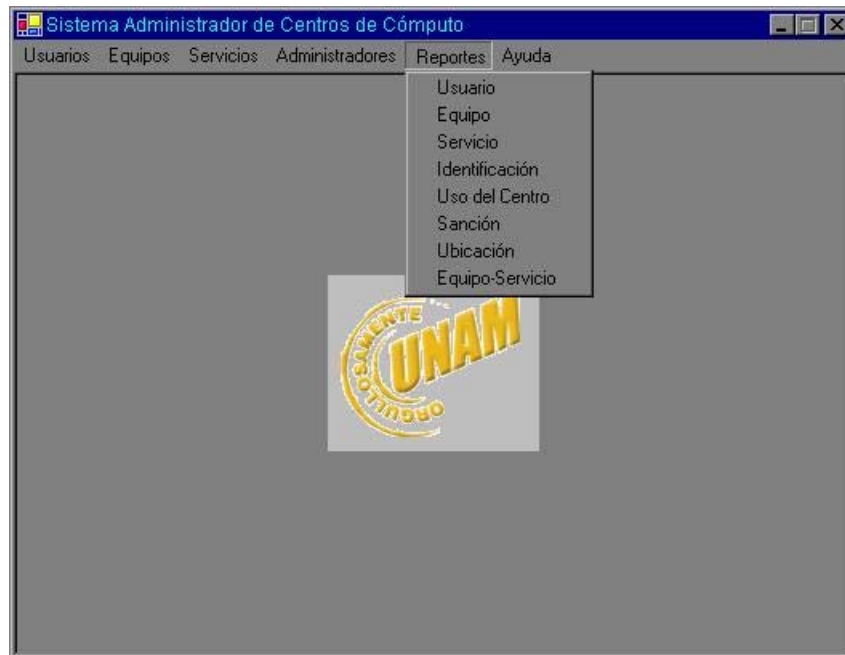


Figura 39. Interfaz principal de SACC.

Diccionario del modelo de clases

Este diccionario, está hecho para dejar más claro algunas palabras que se utilizaron en los modelos de clases presentados, ya que pueden ser algo vagos.

CONCEPTO	DEFINICIÓN
Equipo:	Computadora Personal
Administrador:	Persona que se encarga de realizar las diversas tareas que se presentan en el centro de cómputo.
Reporte:	Información sobre las actividades que se basan en datos que organizados de cierta forma permiten tomar decisiones de una manera más sencilla.
Paquetería:	Conjunto de programas diseñado para trabajar conjuntamente y satisfacer la mayoría de las necesidades del trabajo de oficina. Con frecuencia incluyen un procesador de textos, una hoja de cálculo, una base de datos y un programa de presentaciones.
Aplicaciones específicas:	Diversos programas, orientados a el diseño gráfico, a la programación, a circuitos eléctricos, multimedia, etc.
Dispositivos:	Todo aquel aparato que se puede conectar a la computadora y que de manera conjunta, ayuda a que ésta realice diversas funciones, como por ejemplo el ratón, el teclado, el monitor, las bocinas, etc.
Especificaciones técnicas:	Características técnicas de un Equipo de Cómputo, como velocidad del procesador, cantidad de memoria RAM, tamaño del disco duro, etc.
Ocupación:	Clase encargada de guardar los usuarios que están dentro del centro de cómputo.

5

5. Implementación

Una vez analizadas las clases de SACC, los casos de uso y las interfaces preliminares, se puede comenzar con la codificación.

Los casos de uso serán refinados con el desarrollo de los diagramas de secuencias.

Y estos a su vez con ayuda de la base de datos serán la base para desarrollar el código.

Orientación a las interacciones y cambios de estado

Para realizar los diagramas de secuencias es necesario conocer que componentes UML se usaran y sus interacciones, es decir, que en base a los casos de uso, se modelará la forma en que los componentes de SACC se comunican y la forma y/o momento en que cambian de estado, esta información servirá de mucho para la codificación, además de permitir ver más a detalle los casos de uso.

Componente UML: Base de Datos de SACC

Si recuerda que los componentes son una parte física del sistema que se encuentra en la computadora como por ejemplo los archivos ejecutables, base de datos, etc. Entonces uno de los componentes de mayor importancia en SACC es la base de datos, ya que es la que guardará toda la información de los equipos, ubicaciones, usuarios, etc.

Esta base de datos esta conformada de varias tablas las cuales se relacionan como se muestra en la Figura 40. Cada una de estas tablas está íntimamente ligada a las clases, debido a que cada una de las clases tendrá que comunicarse con la base de datos.

Por ejemplo, la clase usuario tiene procedimientos como Alta, Baja y Edición, que para poder llevar a cabo dichas acciones se debe realizar una conexión a dicha Base de Datos.

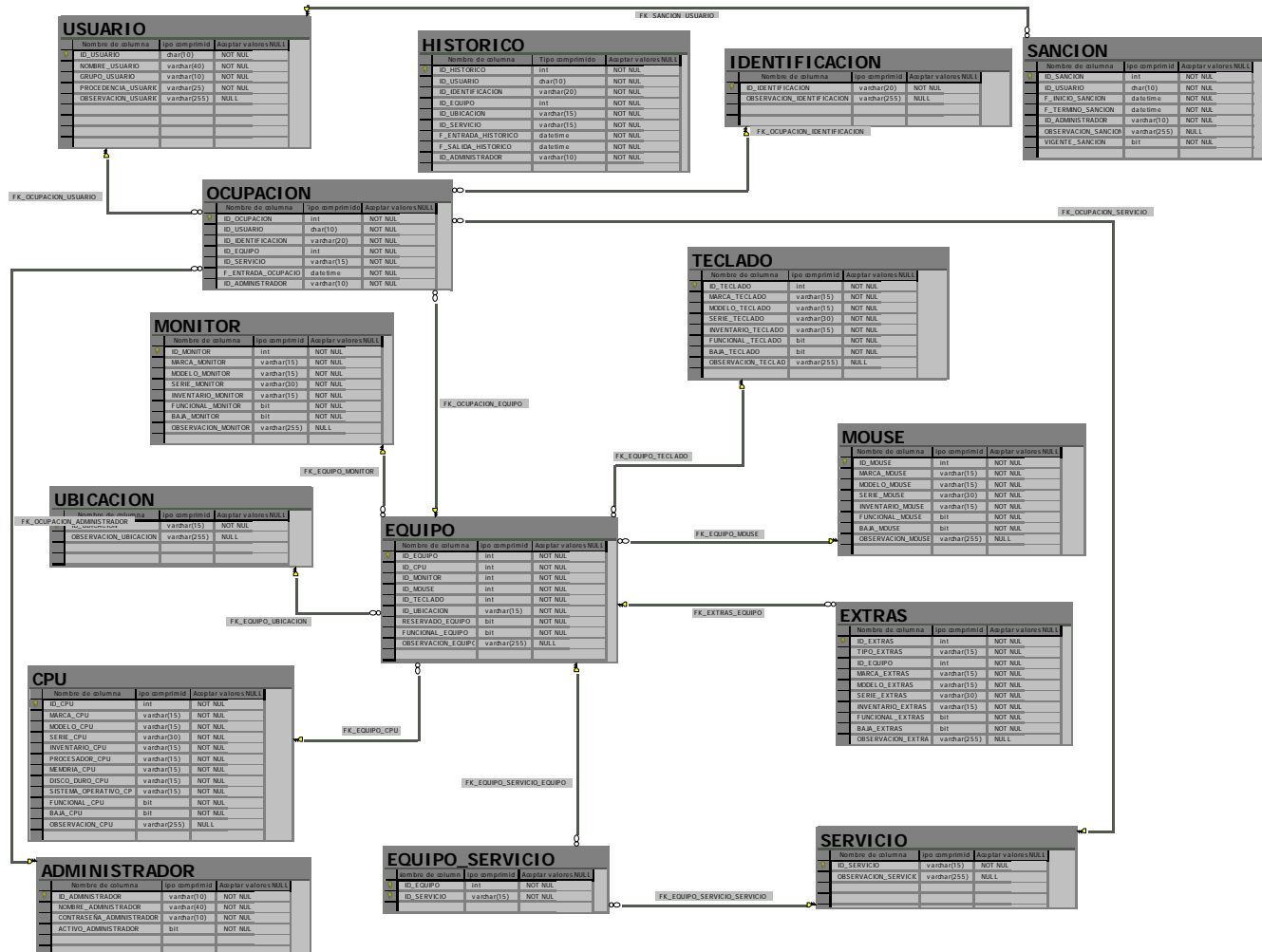


Figura 40. Relaciones de la base de datos de SACC.

Una vez que se conoce la Base de datos, se desarrollarán los diagramas de secuencias, que son simplemente los casos de uso pero representados de una manera más detallada como se podrá observar en las figuras 41, 42 y 43.

Diagrama de secuencias

Hay dos tipos de diagramas de secuencias, los de instancias y los genéricos. El diagrama de instancias sólo se enfoca a un escenario (una instancia), como por ejemplo verificar número de cuenta y en cambio, los diagramas de secuencias genéricos, considera todos los escenarios de un caso de uso al momento de crear el diagrama.

Por lo tanto se podrá generar un diagrama de secuencias genérico a partir de uno de instancias. Además de que los diagramas de secuencias en el desarrollo de una aplicación mayor, implicaría más tiempo para su realización.

A continuación se realizarán algunos de los diagramas de secuencias más importantes de SACC.

En los diagramas podrá observar que se abarcan procesos completos de cuando se realiza un préstamo, se da de alta un Equipo, y cuando se da de alta un equipo, del centro de cómputo.

Diagrama de secuencias de instancias de Ocupación (Préstamo de Equipo)

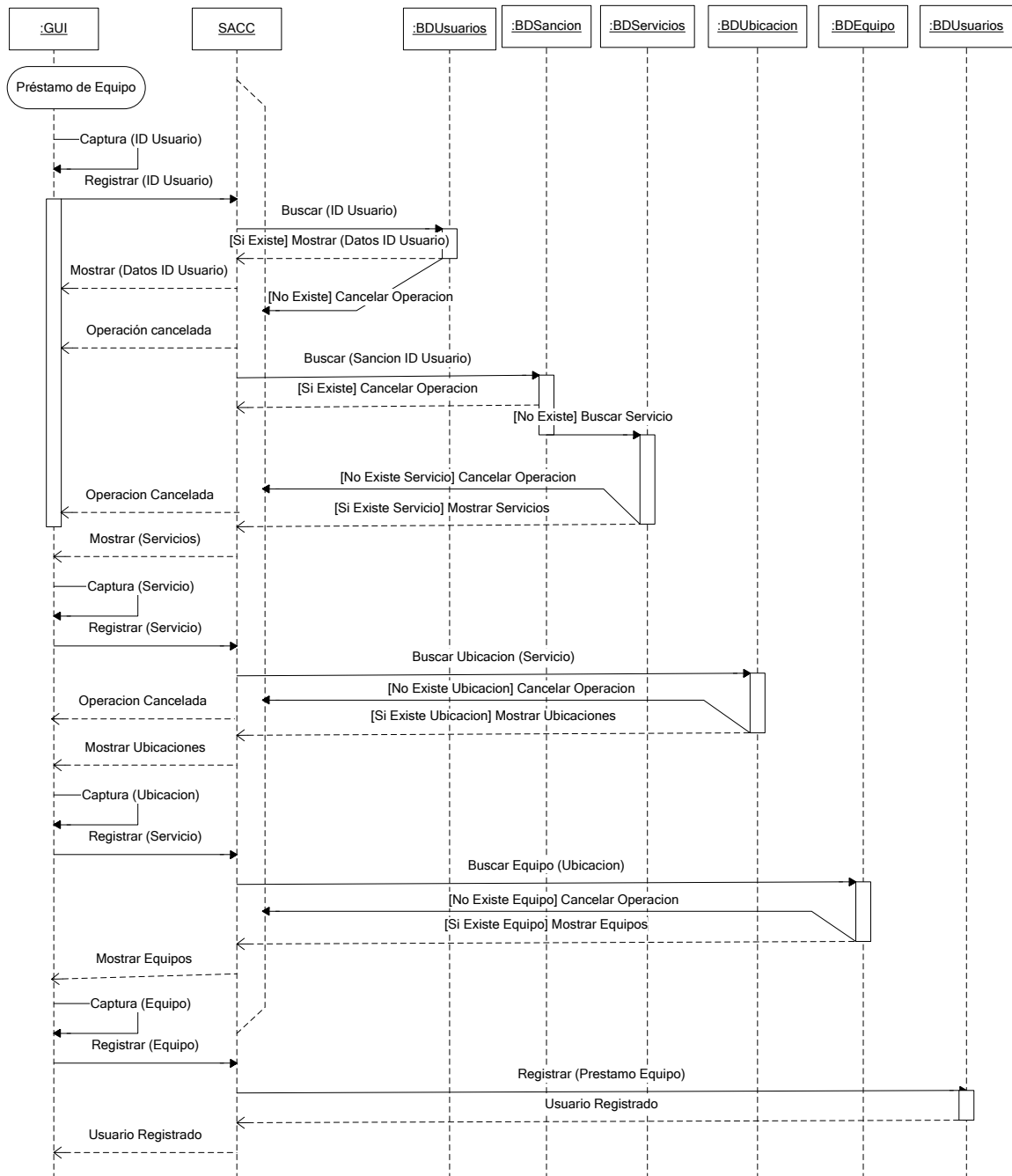
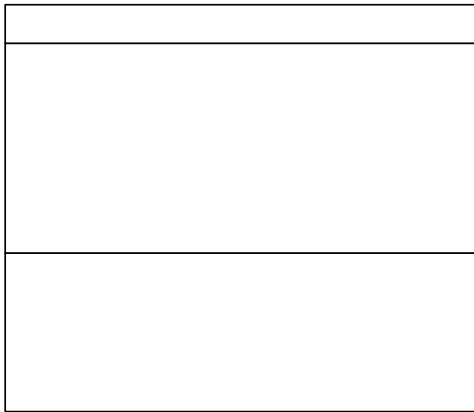


Figura 41. Diagrama de secuencias genérico de Préstamo de Equipo de SACC.



Una vez que desarrollado el diagrama de secuencias, la interfaz y la clase, la codificación resulta mucho más sencilla.

La siguiente parte de código solo es una muestra de las propiedades y métodos que quedaron definidos en el capítulo anterior en el diagramas de clases.

```
Public Class Ocupacion
Private intID, intEquipo, intOcupantes As Integer
Private strUsuario, strIdentificacion, strServicio, strAdministrador As String
Private datEntrada As DateTime
```

```
Public IDs(100000) As Integer
Public EQUIPOs(100000) As Integer
Public SERVICIOs(100000) As String
Public USUARIOs(100000) As String
Public IDENTIFICACIONes(100000) As String
Public ADMINISTRADORes(100000) As String
Public ENTRADAs(100000) As DateTime
Public Encontrados As Integer
Public CancelaError As Boolean
Public TerminaError As Boolean
```

```
Public ReadOnly Property ID() As Integer
End Property
```

```
Public Property Usuario() As String
End Property
```

```
Public Property Identificacion() As String
End Property
```

```
Public Property Equipo() As Integer
End Property
```

```
Public Property Servicio() As String
End Property
```

Ocupacion

-ID : Integer

-Equipo : Integer

-Ocupantes : Integer

-Usuario : String

-Identificacion : String

-Servicio : String

-Administrador : String

-Entrada : Date

+Alta()

+Cancela(entrada UsuarioRequerido

+Todos()

+ExisteE(entrada EquipoRequerido :

+ExisteU(entrada UsuarioRequerido :

+Terminar(entrada UsuarioRequerido

```
Public Property Entrada() As DateTime
End Property

Public Property Administrador() As String
End Property

Public Sub Alta()
End Sub

Public Sub Cancela(ByVal UsuarioRequerido As String)
End Sub

Public Sub Todos()
End Sub

Public Function ExisteE(ByVal EquipoRequerido As Integer) As Boolean
End Function

Public Function ExisteU(ByVal UsuarioRequerido As String) As Boolean
End Function

Public Sub Terminar(ByVal UsuarioRequerido As String)
End Sub

End Class
```

Una vez desarrollados tales métodos, se puede generar la interfaz definitiva para este caso de prestar Equipo con ayuda de SACC.

Con ayuda de la base de datos y el código de las clases ya desarrollados, además de la ayuda del usuario final se construyo la interfaz para cada una de las tareas que realizará SACC.

La forma en como se comporta dicha interfaz con la base de datos, esta definida en los diagramas de secuencias.

The image shows a Windows-style dialog box titled "Prestar Equipo". It contains the following fields and controls:

- Usuario:** A text input field followed by a "Buscar Usuario" button.
- Identificación:** A dropdown menu.
- Nombre:** A text input field.
- Grupo:** A text input field.
- Procedencia:** A text input field.
- Observaciones:** A large text area for notes.
- Servicio:** A dropdown menu.
- Ubicación:** A dropdown menu.
- Equipo:** A dropdown menu.
- Buttons:** "Aceptar" and "Cancelar" buttons at the bottom right.

A continuación se muestran otros de los diagramas más importantes de SACC, que de igual forma muestran la evolución de los diagramas a código.

Diagrama de secuencias genérico de Alta de Equipo

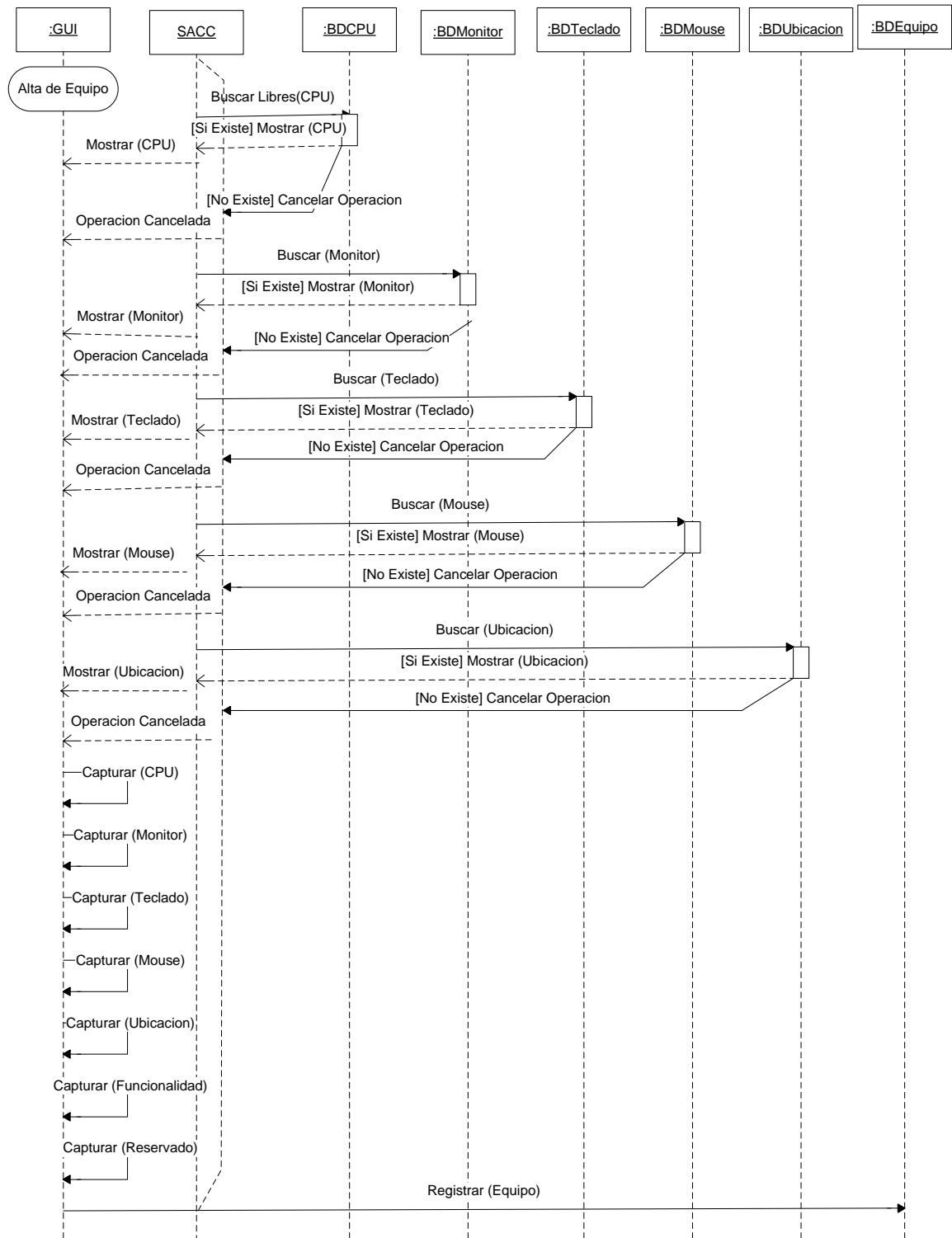


Figura 42. Diagrama de secuencias genérico de Alta de Equipo de SACC.

Equipo
-ID : Integer -CPU : Integer -Teclado : Integer -Mouse : Integer -Monitor : Integer -Ubicacion : String -Observacion : String -Reservado : Boolean -Funcional : Boolean
+Alta() +Edicion(entrada IDRequerido: Integer) +Todos() +Existe(entrada IDRequerido: Integer) : Boolean +Eliminar(entrada IDRequerido: Integer) : Boolean

```
Public Class Equipo
Private intID, intCPU, intTeclado, intMouse, intMonitor As Integer
Private strUbicacion, strObservacion As String
Private bolReservado, bolFuncional As Boolean
```

```
Public IDs(1000) As Integer
Public CPUs(1000) As Integer
Public MONITORes(1000) As Integer
Public MOUSEs(1000) As Integer
Public TECLADOs(1000) As Integer
Public UBICACIONes(1000) As String
Public RESERVADOs(1000) As Boolean
Public FUNCIONALes(1000) As Boolean
Public OBSERVACIONes(100) As String
Public Encontrados As Integer
```

```
Public BajaError As Boolean
```

```
Public Property ID() As Integer
End Property
```

```
Public Property CPU() As Integer
End Property
```

```
Public Property Monitor() As Integer
End Property
```

```
Public Property Mouse() As Integer
End Property
```

```
Public Property Teclado() As Integer
End Property
```

```
Public Property Ubicacion() As String
End Property
```

```
Public Property Reservado() As Boolean
```


End Property

Public Property Funcional() As Boolean
End Property

Public Property Observacion() As String
End Property

Public Sub Alta()
End Sub

Public Sub Edicion(ByVal IDRequerido As Integer)
End Sub

Public Sub Todos()
End Sub

Public Function Existe(ByVal IDRequerido As Integer) As Boolean
End Function

Public Function Eliminar(ByVal IDRequerido As Integer) As Boolean
End Function

The image shows a Windows-style dialog box titled "Alta de Equipo". It features a title bar with standard minimize, maximize, and close buttons. The main area contains several input fields: a text box for "Equipo", and dropdown menus for "CPU", "Monitor", "Teclado", "Mouse", and "Ubicacion". To the right of the "Equipo" field are two checkboxes labeled "Funcional" and "Reservado". At the bottom of the dialog is a large text area for "Observaciones". Two buttons, "Aceptar" and "Cancelar", are located at the bottom right of the dialog.

Figura 43. Interfaz final de Alta de Equipo.

Diagrama de secuencias genérico de Alta de Usuario

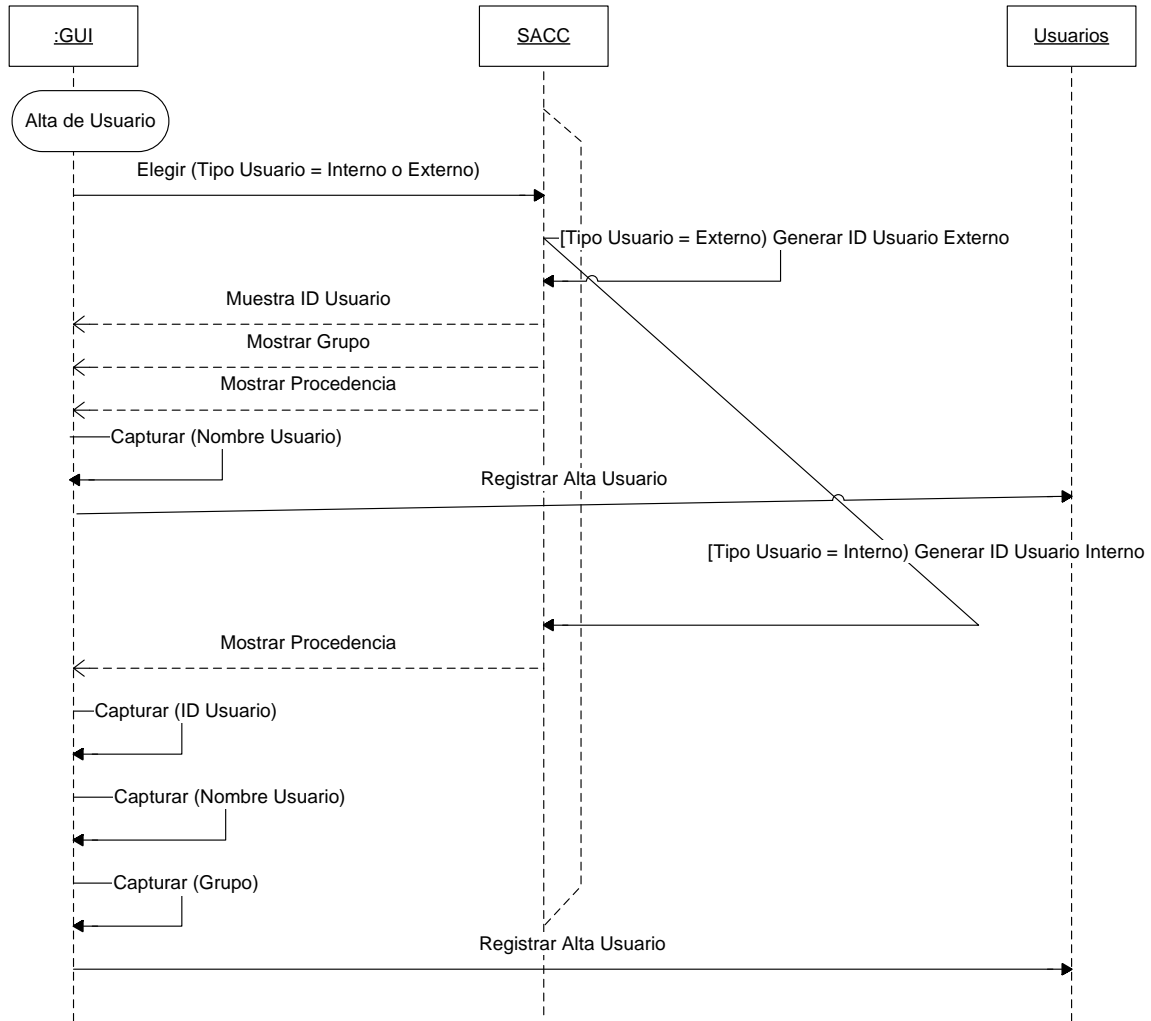
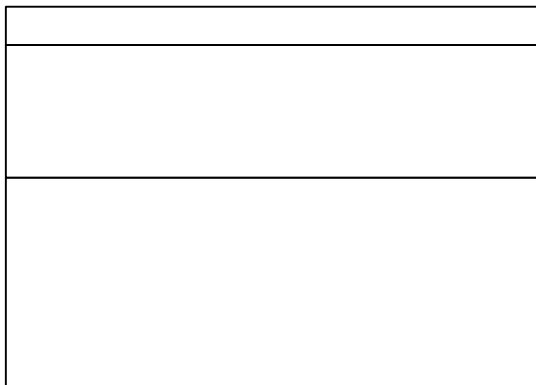


Figura 43. Diagrama de secuencias genérico para dar de Alta a un Usuario SACC.



```
Public Class Usuario  
Private strID, strNombre, strGrupo, strProcedencia, strObservacion As String
```

```
Public IDs(100000) As String  
Public NOMBRES(100000) As String  
Public GRUPOs(100000) As String  
Public PROCEDENCIAS(100000) As String  
Public OBSERVACIONes(100000) As String  
Public Encontrados As Integer  
Public BajaError As Boolean
```

```
Public Property ID() As String  
End Property
```

```
Public Property Nombre() As String  
End Property
```

```
Public Property Grupo() As String  
End Property
```

```
Public Property Procedencia() As String  
End Property
```

```
Public Property Observacion() As String  
End Property
```

```
Public Sub Alta()  
End Sub
```

```
Public Sub BuscarID(ByVal IDRequerido As String)  
End Sub
```

```
Public Sub Edicion(ByVal IDRequerido As String)  
End Sub
```

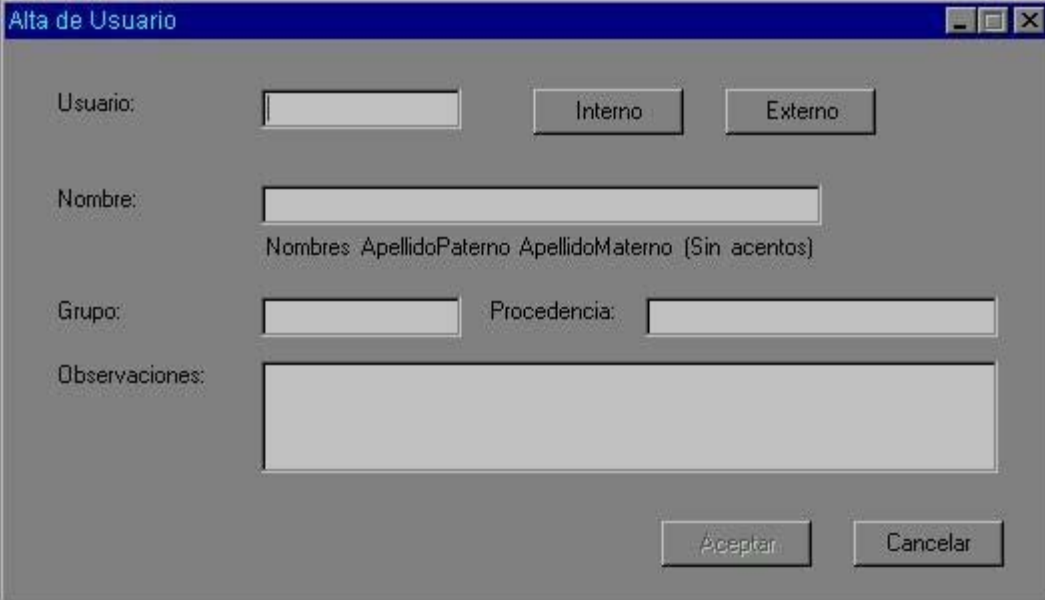
```
Public Sub BuscarNombre(ByVal NombreRequerido As String)  
End Sub
```

```
Public Function Existe(ByVal IDRequerido As String) As Boolean  
End Function
```

```
Public Function ExisteN(ByVal NombreRequerido As String) As Boolean  
End Function
```

```
Public Sub Baja(ByVal IDRequerido As String)  
End Sub
```

```
Public Sub Todos()  
End Sub
```



Alta de Usuario

Usuario:

Nombre:
Nombres ApellidoPaterno ApellidoMaterno (Sin acentos)

Grupo: Procedencia:

Observaciones:

Figura 44. Interfaz final de Alta de Usuario.

Instalación y distribución de SACC

Una vez terminada la aplicación habrá que llevarse a cabo la instalación en el equipo en donde se ejecutará.

Pero antes habrá que tener la versión liberada del programa, esto con la finalidad de poder distribuir la aplicación para instalarla en el equipo en el que usará, cuyas características se mencionaron anteriormente.

Para instalar la aplicación es recomendable crear un proyecto en Visual Studio .Net que cree un pequeño programa que permitirá distribuir la aplicación.

Ésta no es la única forma de realizarlo, ya que se puede simplemente copiar los archivos ejecutables en el lugar deseado, aunque en la práctica se recomienda que se instalen y desinstalen utilizando las herramientas de instalación distribuida que contiene Visual Studio .Net y que usa los servicios de Windows Installer, además de estar aprovechando las ventajas que tiene la suite de desarrollo.

Para poder instalar la aplicación en el equipo no es necesario tener instalado el lenguaje con el que se desarrollo, que en este caso fue Visual Basic .Net, sólo es suficiente con el .Net Framework que es necesario para que se puedan ejecutar las aplicaciones desarrolladas bajo tal plataforma.

El Framework se puede descargar gratis desde la página de Microsoft.com, además de los más recientes Sistemas Operativos ya lo contienen.

Una vez instalado el framework, sólo se tiene que ejecutar la aplicación, en caso de que no se cuente con el, la misma aplicación no se podrá instalar hasta que se cumpla con tal requisito.

En caso de que se desee instalar la aplicación en un equipo cuyos recursos de hardware sean menores a los que se describieron anteriormente, el requisito mínimo necesario para que se pueda ejecutar es tener instalado un sistema Operativo Windows 98 o superior y contar con un poco más de los requerimientos mínimos para cada Sistema Operativo

6

6. Propuestas y resultados

Una vez que se cuenta con un sistema que ayudará al administrador del centro de cómputo de la ENP 9 a realizar su trabajo, lo que corresponde ahora es, analizar el comportamiento de SACC ya puesto en acción, para saber si realmente se cumplen con los objetivos esperados.

Políticas oficiales de desempeño documentado en la ENP 9

En la Escuela Nacional Preparatoria 9 Pedro de Alba, se cuenta con el siguiente documento oficial, en el que se señalan algunas políticas que se deben aplicar dentro del centro de cómputo, para el buen uso de sus servicios, tanto de parte de los alumnos, como de los profesores y de los mismos encargados del Centro.

Tal documento, dice lo siguiente:

Escuela Nacional Preparatoria Plantel 9 "Pedro de Alba" Secretaría General
Reglamento para el laboratorio de cómputo 1. OBJETIVOS GENERALES

El reglamento del laboratorio de cómputo normará las actividades académicas que se realicen en sus instalaciones y su aplicación tendrá como objetivos principales:

Preservar en buen estado las instalaciones y mantener en óptimas condiciones todos los equipos de cómputo, para beneficio de la comunidad del plantel.

Dar a conocer los derechos y obligaciones de los usuarios del laboratorio de cómputo, así como de las responsabilidades del personal académico.

Normar el uso y funcionamiento de todos los espacios y equipos que conforman el laboratorio de cómputo.

Garantizar la impartición de las asignaturas y opciones técnicas que requieren del uso de los recursos de cómputo, de acuerdo a los planes y programas de estudio vigentes.

Dotar a los estudiantes de un ambiente favorable, en el que puedan realizar sus prácticas de clase y elaborar trabajos escolares.

Favorecer las funciones de docencia, asesoría, actualización y capacitación.

II. DEL LABORATORIO DE CÓMPUTO

Está integrado por:

Salas COM1 y COM2. Se usan de manera prioritaria para impartir las asignaturas de "informática" e "informática aplicada a la ciencia y a la industria".

Salas OTC I y OTC II. Espacios para los alumnos inscritos en la opción técnica de computación. Clases de apoyo con software didáctico para asignaturas del área experimental.

Sala de Usuarios. Lugar donde los alumnos acuden en sus horas libres a realizar trabajos ó prácticas de cómputo.

Horario de funcionamiento:

Lunes a Viernes de 07:50 a 21:10 horas.

En cada sala se mostrará en la entrada el horario de actividades específico y el software instalado en las computadoras.

Servicios que se ofrecen:

Docencia de las materias de cómputo que integran el mapa curricular de los alumnos inscritos en el plantel, docencia de las materias de cómputo no curriculares complementarias al mapa curricular de los alumnos inscritos en el plantel, docencia en actividades académicas de breve temporalidad que requieran de equipo de cómputo, espacio de difusión académica y cultural así como servicios básicos de cómputo para labores propias de los académicos y alumnos

que lo requieran.

Servicios de impresión de acuerdo a los horarios y normas establecidas en este reglamento.

Servicios de actualización al personal institucional que así lo requiera, previa solicitud por escrito, programándolo de acuerdo a los requerimientos, protocolos y tiempos disponibles tanto del personal como del Laboratorio.

Apoyo técnico necesario para la instalación, uso y manejo de los equipos, accesorios y dispositivos periféricos solicitados para apoyo académico.

Servicios de telecomunicación digital para actividades académicas.

III. DE LOS USUARIOS

Son usuarios del laboratorio:

Los alumnos inscritos formalmente en el plantel.

El personal académico que para sus labores docentes requieran el uso de equipo o servicios de cómputo.

El personal académico que para sus labores institucionales requieran el uso del equipo o servicios de cómputo.

El personal administrativo que para sus labores institucionales requieran el uso del equipo o servicios de cómputo.

Personal de la Universidad Nacional Autónoma de México que tenga una actividad institucional concreta así como objetiva y que haya sido asignado el plantel.

No son usuarios del laboratorio de cómputo:

Alumnos de la Universidad Nacional Autónoma de México, que no se encuentren inscritos en el plantel.

Personal ajeno a la Universidad Nacional Autónoma de México que en su calidad de proveedor o contratista que se encuentre en el plantel.

Persona alguna ajena a la Universidad Nacional Autónoma de México.

Obligaciones

Conocer y cumplir el presente reglamento.

Depositar en los casilleros respectivos la mochila o algún otro artículo o prenda no necesario en las salas de cómputo.

Entregar credencial vigente de acceso al plantel para utilizar un equipo de cómputo en la sala de usuarios.

Mantener el orden al momento de ingresar al laboratorio para evitar poner en

riesgo tanto a los usuarios como al equipo.

Todo usuario que ingrese al laboratorio de cómputo deberá de hacerlo con tranquilidad, orden y cuidando mantener cerrada la puerta principal.

Ingresar solo con documentos de trabajo, el resto de sus cosas deberá depositarlas a la entrada del laboratorio de cómputo en el lugar destinado para ello.

Si el ingreso del alumnos es posterior al del grupo en el que cursa la asignatura, deberá depositar sus pertenencias en la estantería abierta, quedando bajo su propio cuidado y responsabilidad.

No consumir bebidas y alimentos en cualquier aula del laboratorio de cómputo.

Revisar las condiciones del equipo asignado, en caso de encontrar fallas, daños (raspaduras, manchas, rayas, etc.) o equipo faltante (mouse, teclado monitor, CPU) reportarlo de inmediato al responsable del laboratorio de cómputo en turno.

Por ningún motivo debe intentar hacer reparaciones de software, hardware, desarmar equipos o desconectar cables. En caso necesario se de acudir responsable del laboratorio de cómputo en turno.

Abstenerse de instalar cualquier tipo de software. Utilizar únicamente los paquetes de software con que cuenta el laboratorio.

No se debe desconectar, mover un cambiar equipo.

Queda estrictamente prohibido cambiar la configuración del equipo de cómputo y de los programas contenidos en el mismo, así como copiar, borrar archivos y programas ajenos. Toda modificación del sistema será realizada bajo supervisión del encargado.

Queda prohibido a los usuarios del laboratorio de cómputo la utilización de software alguno para fines de ocio. Si el usuario es sorprendido cometiendo esta falta los implementos técnicos y personales que haya utilizado serán canalizados a la Oficina Jurídica del Plantel o destruidos en su presencia.

Se restringe enfáticamente la utilización de software comercial para juegos, estos deberán de instalarse previa comunicación con el responsable del laboratorio de cómputo en turno y autorización de la Secretaría General del Plantel, dado que deberá tener relación directa con algún objetivo de la materia que el docente imparte, no se deberá permitir que los alumnos utilicen este tipo de software son supervisión docente.

Cuidar de la integridad de su información, así como también del equipo a utilizar, en caso de encontrar el equipo infectado de virus, deberá reportarlo de inmediato al responsable del laboratorio en turno.

El usuario que dañe intencionalmente el equipo de cómputo, será el responsable directo de reponer el equipo, mobiliario o cualquier otro dispositivo en caso de deterioro, pérdida total o parcial. Si es el caso deberá restituirlo por otros con las mismas características y marca. Independientemente de que será remitido con la

autoridad correspondiente del plantel.

Guardar buen comportamiento y disciplina durante su permanencia en el laboratorio.

Mantener limpia su área de trabajo.

El uso de equipo de cómputo es exclusivamente para fines didácticos o institucionales, por lo que se prohíbe el uso para trabajos personales o de terceros con fines de lucro.

Los usuarios que vengan en grupo deberán esperar afuera del laboratorio a su profesor, y podrán ingresar hasta que éste se los indique.

Es deber y responsabilidad del usuario respaldar su información, ya que el laboratorio no es responsable de la pérdida de la misma. Todo trabajo almacenado en el espacio de disco duro será eliminado periódicamente para evitar virus y saturación de espacio.

El apagado de los equipos debe hacerse mediante software, cuidando seguir los procedimientos correctos.

Cuando se requiera, registrarse en los sistemas de control vigentes para cada uno de los espacios a utilizar como un requisito indispensable para tener derecho a hacer uso de los servicios del Laboratorio.

Cooperar con el personal del laboratorio de cómputo para la consecución de los fines de servicio y conservación del equipo, tratando con cortesía y respeto tanto a los profesores como a los responsables del laboratorio de cómputo.

La sala de usuarios se utilizará `por hora de clase y solamente se permitirá continuar trabajando otra hora de clase si la demanda de equipos de computo lo permite.

Se permitirá preferentemente sólo un usuario por equipo de cómputo y sin excepción se registrarán todos los usuarios que ingresen a la sala de usuarios.

Para el uso del servicio de impresión

Traer le papel bond necesario para sus trabajos y en buen estado para obtener impresiones únicas de los mismos sin exceder un máximo de páginas.

No podrá realizar impresiones en hojas especiales (cebolla, albanene, etiquetas, acetatos, plastificado, engrapado, etc.) que pudiesen bloquear o dañar las impresoras.

No podrá realizar impresiones que tengan gráficos o dibujos y que aumenten la densidad de impresión de tonos grises y oscuros.

Para el uso del servicio de INTERNET:

El servicio de Internet que ofrece de manera gratuita a todos los usuarios reconocidos por el presente reglamento, quienes deberán acatar de manera

puntual las siguientes disposiciones.

Ser considerado con los demás usuarios y utilizar el Internet y los equipos o un máximo de 1 hora por usuario a la semana y sólo para fines académicos, en ningún caso podrá utilizarse éste recurso para fines de ocio o entretenimiento personal (chats, talks, write, bbs, ICQ, juegos, etc.), salvo los ordenados por profesores en casos grupales y estando presente el titular.

Queda estrictamente prohibido utilizar este servicio con fines de lucro.

Queda totalmente prohibido el uso de sistemas para comunicación en línea¹¹ o círculos de conversación. A excepción de que en las asignaturas se haga mención práctica sobre la utilización e idioma de estos, en cuyo caso deberá estar siempre el profesor presente y supervisado.

Es altamente penado realizar prácticas de crackeo¹² en daño de la institución o de otros servicios y visitar páginas de contenido erótico o pornográfico.

Derechos:

Todos los usuarios del laboratorio de cómputo tendrán los siguientes derechos:

Hacer uso de los servicios de cómputo, respetando las disposiciones contempladas en el presente reglamento y las prioridades establecidas.

Ser atendido con cortesía y respeto por el personal del laboratorio.

Hacer uso del software existente.

Utilizar los recursos del laboratorio, bajo los linchamientos y normas establecidas por el presente reglamento.

Formular sugerencias que estimen oportunas para mejor la calidad del servicio que se brinda.

Ser informado sobre los horarios y planificación de uso, en medida de los posible a través de anuncios del Laboratorio o directamente con el responsable del laboratorio de cómputo en turno.

Participar en las actividades organizadas con carácter general y académico.

La asignación de aula y horarios para clases regulares a los docentes queda fuera del alcance de esta normatividad, por ser función de las autoridades correspondientes del plantel. La asignación de aula y horarios para clase esporádicas o de breve temporalidad deberá de ser coordinada por el responsable del laboratorio de cómputo en e turno correspondiente, no debiendo de afectar de ninguna manera as actividades regulares docentes propias del laboratorio de cómputo.

¹¹ Chats, Talks, etc.

¹² Violación conciente y directa a la integridad de información y recursos computables para fines de lucro personal.

IV. ATRIBUCIONES Y RESPONSABILIDADES DE LOS ENCARGADOS DEL LABORATORIO, PROFESORES Y AUTORIDADES.

Vigilar la ejecución y observación de las normas del presente reglamento y aplicar, en el ámbito de su competencia, las sanciones derivadas.

Autoridades

Las políticas y lineamientos en lo relativo a las generalidades funcional del laboratorio de cómputo estará sujeto a la responsabilidad de la autoridad universitaria correspondiente de acuerdo a las jerarquías y niveles correspondientes.

En caso de que algún equipo falle en su funcionamiento o haya sido dañado, las autoridades serán responsables de agilizar y supervisar el servicio de reparación o mantenimiento que corresponda.

Proporcionar el material y los servicios para la impartición de las asignaturas que requieren el uso de los recursos de cómputo, de acuerdo a los planteles y programas de estudio vigentes.

Las autoridades del plantel deberán de cuidar que siempre que un docente en el laboratorio de cómputo imparte clase exista un responsable del laboratorio de cómputo.

Responsable del laboratorio de cómputo

El responsable del laboratorio de cómputo es el técnico académico adscrito al laboratorio de cómputo del plantel es el docente a cargo de supervisar las actividades propias del laboratorio de cómputo, así como de velar por el buen funcionamiento y óptimo desempeño.

El responsable del laboratorio de cómputo como profesionista del área deberá de condensar y aplicar las medidas necesarias dentro del laboratorio de cómputo para poder mantener un nivel funcional óptimo con los recursos, espacios y equipo con los que cuenta el laboratorio de cómputo.

La implementación de políticas y lineamientos de la funcionalidad general del laboratorio de cómputo definido por alguna autoridad correspondiente de la Universidad Nacional Autónoma de México es obligación directa del responsable del laboratorio de cómputo y cuenta con el apoyo de profesores y autoridades del plantel.

El responsable del laboratorio de cómputo puede decidir directamente sobre cualquier asunto de funcionalidad particular y de control dentro del laboratorio de cómputo.

Ejecutar las actividades técnicas, de control y funcionamiento del laboratorio.

Control y mantener en operación y buenas condiciones todo el equipo e instalaciones asignadas para brindar el servicio para el que está destinado el

Laboratorio.

Reportar inmediatamente a la Secretaría General o a la Oficina Jurídica del plantel a los usuarios que estén cometiendo una falta a este reglamento.

Informar de la manera convenida a la Secretaría General de las situaciones relevantes que se presenten en su turno.

Comunicar y hacer notar a los docentes en el laboratorio de cómputo de la debida asignación, utilización y tiempo de uso del aula en la que estén laborando.

En caso de que un usuario esté realizando actividades no escolares, el encargado podrá pedirle el usuario que deje el equipo.

Desalojar el laboratorio cuando sea solicitado para realizar la limpieza, o dar mantenimiento a los equipos de cómputo.

Respetar su horario de actividades.

Tratar con cortesía y respeto a los profesores, alumnos, empleados y demás usuarios del laboratorio.

Velar por el adecuado uso del equipo y mobiliario del laboratorio para evitar daños y/o descomposturas, para ello contará con el apoyo de los profesores.

Permitir que los usuarios accedan a los servicios que soliciten, siempre y cuando cumpla con los protocolos, espacios y tiempos que le sea indicado por el responsable del laboratorio de cómputo en turno.

En la medida de los espacios, recursos y seguridad del laboratorio de cómputo el docente que imparte clases regulares en el laboratorio de cómputo puede solicitar a la Secretaría General del plantel, un espacio para uso personal, pero al término de cada ciclo escolar deberá limpiarlo y retirar los artículos que se hubiesen acumulado.

Supervisar las salas de cómputo antes y al terminar una clase, para reportar daños por parte de los integrantes de algún grupo recabando la firma de conformidad de los profesores.

El responsable del laboratorio de cómputo así como los docentes y autoridades del plantel deberán de cuidar que pasillos y accesos al laboratorio de cómputo estén totalmente despejados de personal ocioso, equipo, artículos de los usuarios u objeto alguno que pudiese comprometer la seguridad del laboratorio de cómputo y los usuarios en caso de emergencia.

A solicitud de un docente en el laboratorio de cómputo el responsable del laboratorio de cómputo evaluará e implementará soluciones a problemas específicos dentro del ámbito del laboratorio de cómputo, previa autorización de la Secretaria General del Plantel.

Profesores

El docente en el laboratorio de cómputo es el profesor que requiere de un aula del laboratorio de cómputo para funciones docentes frente a grupo. Es responsabilidad directa del docente la disciplina así como la seguridad de los usuarios, equipo físico y equipamientos lógicos que requiera en un aula del laboratorio de cómputo.

Respetar su horario y aula asignada en el laboratorio.

No es permisible que un docente en el laboratorio de cómputo pueda impartir clase en dos aulas de manera simultanea. El responsable del laboratorio de cómputo puede hacer al docente las observaciones necesarias para poder detener esta práctica, sí el docente continua con esta actitud el responsable del laboratorio de cómputo deberá de comunicarlo a la Secretaría General del plantel.

El profesor asignado a un grupo deberá verificar ocular y superficialmente el equipo antes y después de su hora, en compañía del técnico académico en turno, en caso de encontrar alguna situación anómala, deberá hacerla notar a la brevedad al encargado del laboratorio, en caso contrario el profesor será el responsable de los daños.

Solicitar el apoyo del encargado del laboratorio cuando desconozca la utilización de alguna herramienta de cómputo.

Mantener el orden y supervisar que los alumnos hagan uso adecuado de programas y equipo de cómputo, en las sesiones reservadas para sus cursos.

Reportar inmediatamente, cuando algún equipo de cómputo falle, al responsable del laboratorio de cómputo en turno, así como auxiliar en las actividades que conduzcan a la pronta reparación del equipo dañado, utilizando para ello los formatos disponibles en la Secretaría General del Plantel.

Informar de la manera convenida a la Secretaría General del plantel correspondiente de las situaciones relevantes que se presenten en el laboratorio de cómputo.

Solicitar que sus alumnos al término de su clase y con tiempo desalojen la sala para que los usuarios de la siguiente sesión puedan trabajar. Así como verificar que todos y cada uno de los componentes del equipo a él equipo asignado estén completos.

Abstenerse él y sus alumnos de intercambiar partes o componentes de los equipos a él confiados, ya que en estos casos, será responsable de los daños ocasionados a los equipos.

Permitir que el responsable del laboratorio de cómputo en turno revise e inspeccione el estado y condición de los equipos para la siguiente clase el principio y final de su clase, para así deslindar responsabilidades.

Por cuestiones de metodología pedagógica el docente puede realizar pequeñas adaptaciones y distribuciones temporales de equipo y mobiliario del aula del laboratorio de cómputo que utilice, debiendo de restablecer en su totalidad

modificación alguna antes de que termine su tiempo de uso en el aula en ese mismo evento y previo acuerdo y supervisión del responsable del laboratorio de cómputo en turno.

En caso de que algún docente en el laboratorio de cómputo requiera de la instalación de software o información en los equipos que utilizará deberá de presentar su solicitud de instalación junto con los requerimientos e insumos necesarios cuando menos con una semana de anticipación.

Si desea realizar revisión de trabajos o programas, coordinar el grupo o seccionarlo para mantener el control y disciplina. Si es necesario realizar estas actividades los viernes por la mañana o el lunes por la tarde, dado que el centro de cómputo tiene la menor actividad.

V. DE LAS SANCIONES

Las sanciones aplicables a quienes violen las disposiciones de este reglamento o cometan daños o cualquier acto ilícito en el laboratorio de cómputo son, de acuerdo a la falta cometida será la que dicten las autoridades correspondientes, considerando ante todo la gravedad del hecho constitutivo de la infracción, de sus consecuencias, así como la intencionalidad y reincidencia de la fracción, todo ello en el marco de la legislación universitaria.

Las llamadas de atención podrán ser aplicadas por el profesor o responsable del laboratorio de cómputo en turno, quien las registrará y reportará a la autoridad correspondiente del plantel.

La acumulación de llamadas de atención por un usuario, ameritará la suspensión temporal del servicio del laboratorio de cómputo.

El usuario será suspendido definitivamente del laboratorio de acuerdo a los criterios:

Reincidencia en el mal comportamiento.

Daños o alteraciones a las instalaciones materiales y/o equipo ocasionados con dolo o negligencia.

Robo de material y/o equipo.

Actos que pongan en peligro la seguridad de terceros por negligencia dentro del laboratorio.

En todos los casos anteriores se turnará inmediatamente el caso a la autoridad correspondiente del plantel, que sancionará de acuerdo con lo estipulado en la legislación universitaria.

TRANSITORIOS:

Cualquier sanción no prevista en el presente reglamento será resuelto por el encargado del laboratorio en el ámbito de su competencia o bien por la

autoridad del plantel que corresponda.

La autoridad del plantel correspondiente se reserva el derecho de modificar, adecuar o actualizar el presente reglamento de acuerdo a las necesidades del plantel y teniendo siempre que considerar la previa revisión colegiada a las modificaciones que desee implementar y las observaciones emanadas por el H. Consejo Interno del plantel.

El presente reglamento entra en vigor a partir de la fecha de firma y su vigencia depende de la aprobación del H. Consejo Interno del plantel.

Sin embargo, tales procedimientos no se respetan al cien por ciento ya que, o no son conocidos por todos o no se tiene el suficiente tiempo para llevarlos a cabo o algunos son muy restrictivos.

Para definir el desempeño del sistema, se identificarán algunos criterios de desempeño que se tengan establecidos en la escuela. Los criterios empleados para medir o evaluar los servicios que brinda el centro de cómputo, se hacen en base a la cantidad de usuarios que lo visitan, por ende, esto quiere decir que entre más usuarios entren al centro, el servicio es bueno. Sin embargo el hecho de que se tengan muchos usuarios no implica que el servicio sea eficaz.

Pruebas de SACC

Debido a que el desarrollo de SACC se hizo con programación orientada a objetos, esto ayudó a que las modificaciones se fueran realizando al mismo tiempo de desarrollarse la aplicación, provocando así que, al concluir el proceso de desarrollo, las modificaciones que desearán llevar a cabo fueran diminutas.

Es por ello que las pruebas durante el ciclo de vida de la aplicación juegan un papel importante y Visual Studio .NET proporciona tales características, sólo por mencionar una, cuenta con un conjunto eficaz e integrado de herramientas de prueba, que ayudan a identificar errores significativos lo antes posible, con lo que se disminuye el impacto de los problemas en el resultado global.

Comprobación del cumplimiento de los casos de uso de SACC

El cumplimiento de los casos de uso se hizo al 100%, ya que como se acaba de mencionar, conforme se avanzaba en el desarrollo del sistema, se probaban cada una de ellos, módulo por módulo, línea por línea.

Para verificar el cumplimiento de los casos de uso, se llevaron a cabo diversas pruebas entre las que destacan las siguientes:

1. Para el registro del usuario en los campos donde se tenía que capturar sólo letras, se introdujeron números.
2. Una vez iniciada una acción, se cancelaba.

3. La aplicación, se instaló en diferentes equipos.
4. Se navego entre cada uno de sus menús.
5. Se verifico la secuencia del foco al navegar entre las opciones del sistema.
6. La aplicación se ejecutó varias veces en un mismo equipo.
7. Se registraron 20 usuarios.
8. Se registraron 10 usuarios en uso de equipo.
9. Se inventariaron 15 equipos.
10. Se exportaron diversos reportes a archivos .doc, .xls, .pdf y .rft.
11. Se dejaron campos vacíos cuando se registraba un usuario, equipo, servicio, etc.

Al obtener los resultados de un modulo, se continuaba con el siguiente. Una vez terminada esta fase, se puede afirmar que el sistema tiene un excelente desempeño. Debido a que las pruebas se superaron y que además el tiempo en que se realizan las diferentes actividades fue menor que cuando se realizaban de forma manual.

Según algunas pruebas comparativas, de la forma en que se hacían las diferentes acciones, se obtuvieron los siguientes resultados:

Acción	Tiempo	
	Forma manual	Con ayuda de SACC
Registro de un usuario	3 minutos	Menos de un minuto
Realizar un inventario	20 horas	18 horas
Realizar un reporte	8 horas	1 hora
Asignar un equipo	3 minutos	Menos de 5 segundos
Reconocer la falla de un equipo	5 minutos	Menos de un minuto
Ubicar un equipo libre	2 minutos	Menos de 5 segundos

Es por ello, que se dice que el sistema ayudará a reducir el tiempo para efectuar las tareas del administrador, que era uno de los principales objetivos de SACC.

Como se aprecia, el hecho de que se haya reducido el tiempo, implica que el servicio mejorará, ya que una de las características que debe tener un servicio, es calidad, rapidez y sobretodo que sea económico, pero lo económico aquí no se aplica para los alumnos, pero si las otras dos características.

La calidad del servicio viene de la mano con la rapidez con la que se sirve. Por ejemplo al obtener más rápido los reportes que se generarán con ayuda de SACC, se podrá saber a tiempo el estado de los equipos y por ende se podrán corregir más pronto, ya sea dándolo de baja, reinstalándolo, llamando a las

personas indicadas para que lo reparen –ya que no se permite abrir equipos-, etc., según sea el caso.

Del mismo modo, con ayuda de los reportes de sanciones se logrará dar servicio a aquellos alumnos que realmente hagan uso de los servicios de una manera adecuada.

También con ayuda de los reportes se podrá saber que servicios son los más utilizados y cuáles no, así se puede optimizar el espacio de los discos duros por ejemplo, en base a ello se instalarán sólo las aplicaciones necesarias o de mayor uso.

Con todo esto, y otras características más que ofrece SACC se logrará que el servicio sea eficaz.

Posibles líneas de desarrollo

Como se puede observar SACC, tiene algunos detalles. En cuanto a la realización del inventario probablemente se lleve a cabo en el mismo tiempo en que se realizaba manualmente, ya que de una u otra forma, se tendrá que capturar la información a SACC, ya que por la distribución y lejanía de los equipos resulta casi imposible ir capturando al mismo tiempo de obtener sus características del equipo.

Aunque esta tarea si se puede llevar a cabo, se necesitará de más de una persona, cosa que en el centro es poco probable, pues solo hay un encargado por turno, y el hecho de contar con el auxilio de una segunda persona -no remunerada por cierto-, se puede provocar la creación de un inventario de mala calidad.

Otra de las desventajas del módulo de inventarios es que, si no se capturan correctamente los datos de los equipos, en el futuro se tendrán los mismos problemas que si se realizará a mano.

Gracias a todas estas desventajas, se ha pensado en el desarrollo de un módulo, que ayude a terminar con tales problemas.

El módulo está pensado para que permita imprimir los números de inventario, en formato de código de barras, con la finalidad de que pueda leerse con un lector de códigos de barras, es decir que sólo bastará con capturar los números de inventario y se imprimirán en un formato de código de barras junto con otra información importante como número de serie y número de equipo, así como ubicación.

El hecho de que se puedan tener tales datos en código de barras permitirá que las capturas o verificaciones de datos posteriores sean más sencillas.

Sin embargo para la primera captura de tales datos no existe escapatoria, pero como se dice por ahí, si se realiza bien la primera vez, no se tendrán problemas las siguientes veces que se desee utilizar.

La lectura de códigos de barras, ayudará también a obtener los datos de los alumnos de una manera más rápida, ya que como se puede apreciar en la siguiente figura, la credencial de los alumnos de este plantel tiene un código de barras, y con ello se puede obtener sus datos.

Por lo tanto, un lector de códigos de barras, será muy útil para la utilización de SACC.

Otra nueva línea de desarrollo para SACC, es en cuanto el uso de algún dispositivo móvil. El hecho de que SACC fuese desarrollado en la plataforma .Net, fue porque si se cuenta con un dispositivo de estos, y con acceso a Internet por ambas partes, es decir, tanto el dispositivo como el equipo en donde se encuentra la aplicación, podrá ser usada desde tal dispositivo, siempre y cuando se cuente con tal servicio.

Para que esta parte sea posible, sólo será suficiente crear la interfaz para tal dispositivo. Como por ejemplo una Pocket PC (PC de bolsillo), pudiendo ser manipulado el sistema entonces, desde tal dispositivo y agregando las ventajas que ofrece un lector de códigos de barras, se obtendrán muy buenos resultados, puesto que se podrá realizar el inventario sin necesidad de que alguien los capture en el sistema, ya que con el lector de códigos de barras conectado a la Pocket PC, el administrador podrá desplazarse con ellos e ir realizando el inventario, además de evitar errores de captura.

Conclusiones

Como se pudo observar en el transcurso de este trabajo, el hecho de llevar a cabo el desarrollo de un sistema, en este caso SACC, no es un proceso aislado, ya que no sólo hubo que averiguar los recursos con los que contaba el Centro de cómputo, si no que también hubo que pensar en la metodología a utilizar, en las herramientas computacionales, en la gente que participaría en el desarrollo, en el lenguaje de programación, en los usuarios finales, en el tiempo que se le dedicaría a determinada fase, en la plataforma donde se implementaría, etc.

Actualmente no sólo se debe poner especial atención en los beneficios del sistema, sino también en su implementación futura, para asegurar que el tiempo de vida del sistema sea aún mayor, es decir, se deben considerar los cambios en la tecnología, que hoy día son muy rápidos tanto en Hardware como en Software.

El hecho de tener la posibilidad de usar una aplicación en un dispositivo que no sea un equipo de cómputo como por ejemplo una PocketPC, TabletPC, SmartPhone, etc., resulta una experiencia muy agradable para el usuario que interactúa con ella, debido a que tales dispositivos tienen las ventajas de movilidad, tamaño y conectividad, además de que con el avance tecnológico tales dispositivos, económicamente hablando, comienzan a estar más al alcance de cualquier persona.

Para el desarrollo de aplicaciones se debe tener muy presente el uso de diagramas para la representación de los resultados de cada una de las fases según la metodología elegida, pues servirá como documentación del desarrollo de la aplicación de una forma gráfica, además de que ayudará a los usuarios y desarrolladores futuros, que no tuvieron que ver con el perfeccionamiento del sistema. Para elegir la metodología ideal para el desarrollo de cualquier sistema, dependerá de la experiencia de los líderes del proyecto.

Otra de las ventajas acerca del uso de los diagramas durante la realización de un sistema es que gracias a ellos la codificación se torna más sencilla y por ende el desarrollo tarda menos tiempo, y se tiene más tiempo para las correcciones.

En cuanto al software y plataforma sobre la que se desarrolle la aplicación, siempre se da, si no es que la mayoría de las veces que, para sacar mayor provecho a los recursos con los que se cuenta es preferible usarlo en la plataforma para la que fue creada, es decir, si se trabaja con la plataforma .Net, es preferible manejar software de Microsoft, ya que se sacará mayor provecho a lo beneficios que ofrece.

Una de las principales ideas que tuvo Microsoft al realizar el Framework, que es el corazón de la plataforma .Net, es que los desarrolladores sólo se preocupen por satisfacer las necesidades del usuario, y no por la plataforma en la que se trabaje –fuera esta Windows NT/98/ME/XP/2003 o Linux-, ni tampoco sobre el dispositivo donde se ejecute.

El hecho de que el Framework fuera ideado para trabajar con el Sistema Operativo Microsoft, no indica que las aplicaciones hechas sobre tal plataforma no funcionen en Linux o en Windows 2003 Mobile (PocketPC), ya que este

modelo de desarrollo se hizo pensando en romper barreras de software y hardware.

Para la ejecución de las aplicaciones en Linux, hay un proyecto llamado MONO, el cual puede hacer uso del Framework de Microsoft pero bajo tal plataforma.

Ahora la obtención de aplicaciones muy elaboradas ya no es sólo una idea, sino que ahora se podrán ejecutar en cualquier plataforma, dando como resultado que se puedan poner a trabajar en conjunto los programadores independientemente de la plataforma en la que especialicen, dando así como producto grandes aplicaciones que satisfagan completamente al usuario.

Es por ello que SACC será de gran utilidad no sólo en el centro de cómputo para el que fue hecho, sino también para otros que pertenezcan a la UNAM debido a que maneja números de cuenta. sin embargo este modulo puede ser sustituido o eliminado para ser utilizado en otros centro de cómputo que no pertenezcan a la UNAM.

En cuanto al módulo para la realización de inventarios, este sería de gran utilidad si SACC se ejecutara en una PocketPC con un lector de código de barras, ya que reduciría el tiempo de realización de este, y no sólo se podrá usar esta parte para un centro de cómputo, sino que con unos pequeños cambios se puede utilizar para realizar cualquier tipo de inventario, como por ejemplo el de una tienda comercial.

Por lo tanto con el desarrollo SACC los tiempos de atención hacia los usuarios son más cortos, el reporte de equipos con fallas es más rápido, la entrega de inventario es más eficiente, se tiene mejor aprovechamiento de los recursos de hardware, software, etc. Por lo que el servicio en el centro de cómputo tiende a ser eficaz, cumpliéndose cabalmente las necesidades del usuario final que en este caso son los Encargados.

Anexos

Sistema Administrador de Centros de Cómputo SACC

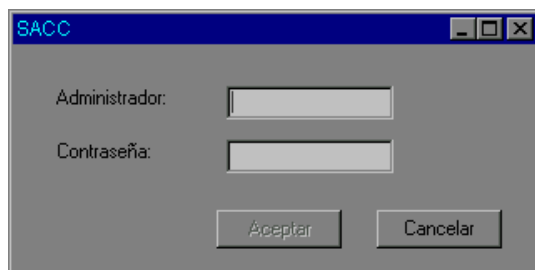
Manual de Usuario

Desarrolladores:

Hernández Martínez Héctor
Rosario García Lucero

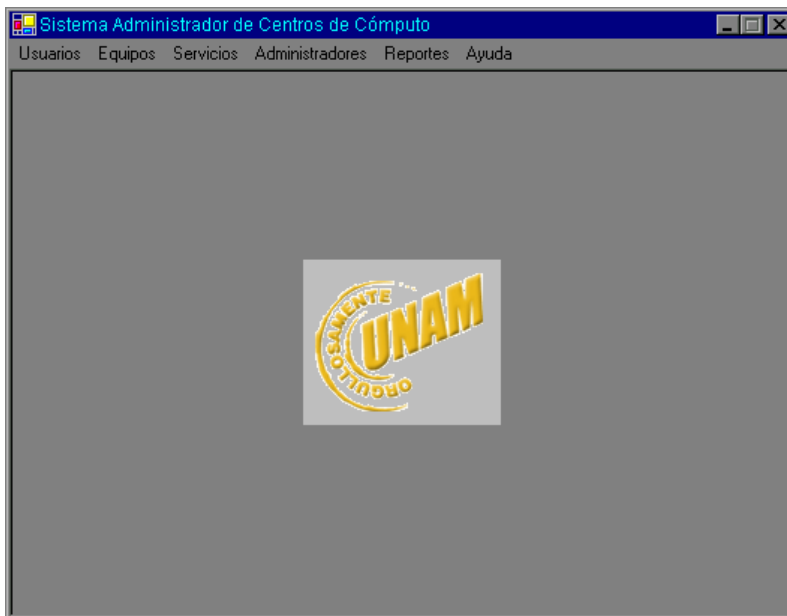
Cómo entrar

Cuando el sistema se inicia aparece una pantalla que pide que el administrador se identifique. Si el nombre de usuario y/o la contraseña no son correctos el sistema no permitirá el acceso.



Pantalla Principal

Una vez autenticado el usuario aparecerá la pantalla principal la cual se muestra a continuación:



En esta pantalla está el menú de acciones el cuál nos permitirá realizar todas las operaciones en las que nos auxiliará SACC.

Las actividades están divididas de la siguiente manera:

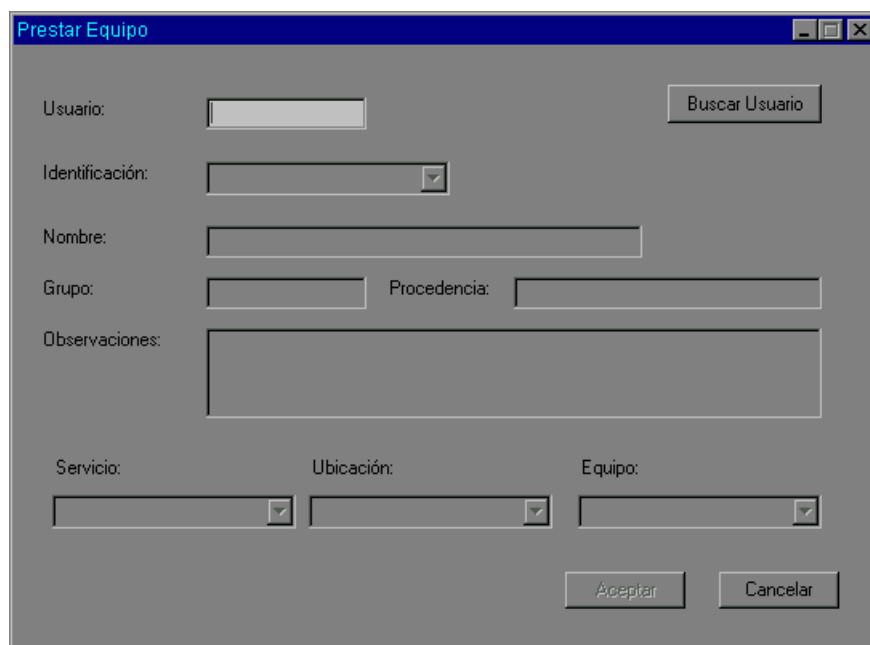
Menú Usuarios

Submenú Préstamos

Este menú contiene todo lo relacionado al Préstamo de equipo a usuarios.

Contiene las siguientes opciones:

Préstamo de equipo. En este menú se puede prestar equipo a usuarios que no estén sancionados.



The screenshot shows a Windows-style dialog box titled "Prestar Equipo". It contains the following fields and controls:

- Usuario:** A text input field with a "Buscar Usuario" button to its right.
- Identificación:** A dropdown menu.
- Nombre:** A text input field.
- Grupo:** A text input field.
- Procedencia:** A text input field.
- Observaciones:** A large text area for notes.
- Servicio:** A dropdown menu.
- Ubicación:** A dropdown menu.
- Equipo:** A dropdown menu.
- Buttons:** "Aceptar" and "Cancelar" buttons at the bottom right.

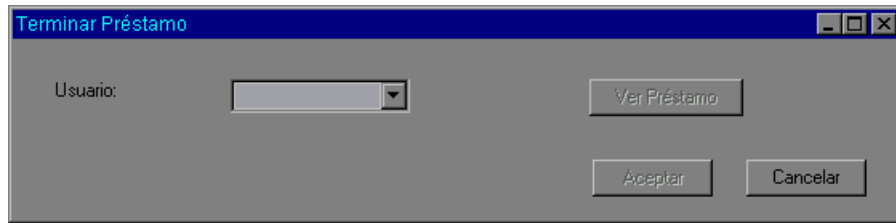
Cancela Préstamo. Este menú cancela el préstamo de un equipo, es decir el préstamo no se registrará en el Histórico de Usuarios.



The screenshot shows a Windows-style dialog box titled "Cancela Préstamo". It contains the following fields and controls:

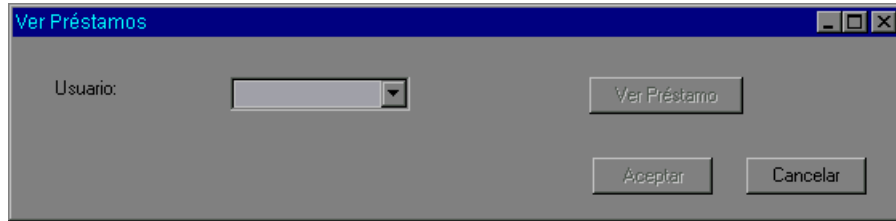
- Usuario:** A dropdown menu.
- Ver Préstamo:** A button to the right of the user selection.
- Buttons:** "Aceptar" and "Cancelar" buttons at the bottom right.

Termina Préstamo. Este menú es usado cuando un usuario termina



The screenshot shows a dialog box titled "Terminar Préstamo". It contains a "Usuario:" label followed by a dropdown menu. To the right of the dropdown is a button labeled "Ver Préstamo". Below these elements are two buttons: "Aceptar" and "Cancelar".

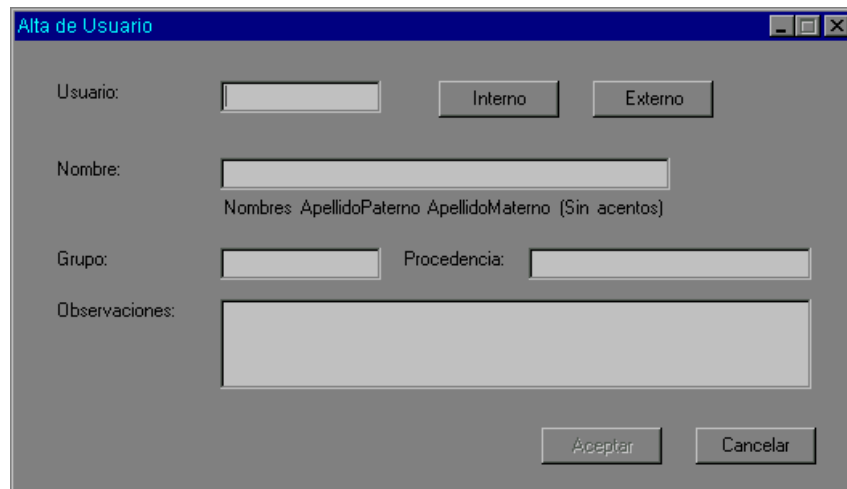
Ver todos. Este menú se pueden ver todos los prestamos actuales del equipo.



The screenshot shows a dialog box titled "Ver Préstamos". It contains a "Usuario:" label followed by a dropdown menu. To the right of the dropdown is a button labeled "Ver Préstamo". Below these elements are two buttons: "Aceptar" and "Cancelar".

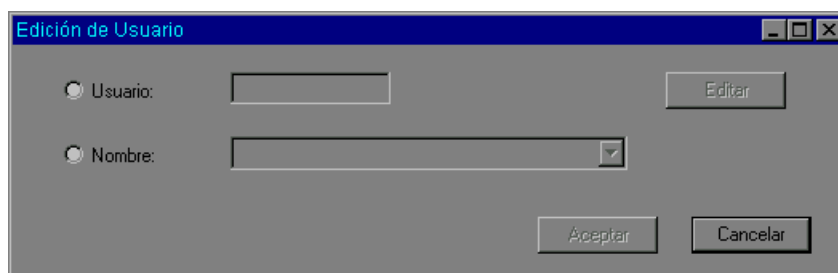
Submenú Usuarios

Alta. En este menú se dan de alta los usuarios que tendrán derecho a préstamo de equipo.



The screenshot shows a dialog box titled "Alta de Usuario". It contains several input fields and buttons. At the top, there is a "Usuario:" label followed by a text input field and two buttons: "Interno" and "Externo". Below this is a "Nombre:" label followed by a text input field. Underneath the name field is the text "Nombres ApellidoPaterno ApellidoMaterno (Sin acentos)". Below that are two more text input fields: "Grupo:" and "Procedencia:". At the bottom is a large text area for "Observaciones:". At the very bottom are two buttons: "Aceptar" and "Cancelar".

Edición. Aquí se editan los datos de los usuarios.

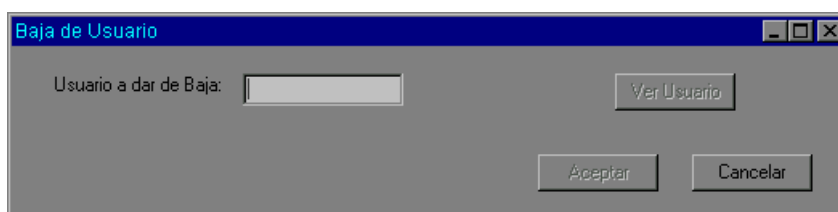


Edición de Usuario

Usuario:

Nombre:

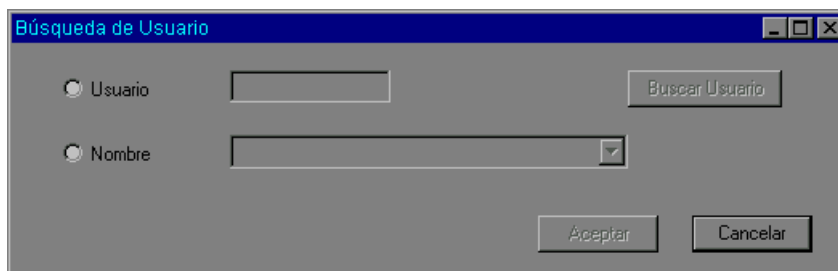
Baja. En este menú se dan de baja los usuarios.



Baja de Usuario

Usuario a dar de Baja:

Búsqueda de usuarios. En este menú se pueden consultar los datos de los usuarios.



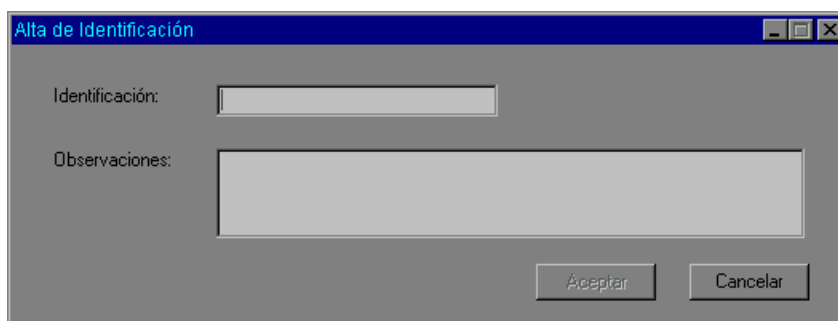
Búsqueda de Usuario

Usuario

Nombre

Submenú Identificaciones

Alta. Aquí se dan de alta los tipos de identificación que pueden presentar los usuarios cuando se registran para un préstamo.



Alta de Identificación

Identificación:

Observaciones:

Edición. En este menú se editan los tipos de identificación.



Edición de Identificación

Identificación:

Baja. En este menú se eliminan los tipos de identificaciones.



Baja de Identificación

Identificación:

Ver todos. Este menú ofrece la posibilidad de ver todos los tipos de identificaciones que puede presentar un usuario.

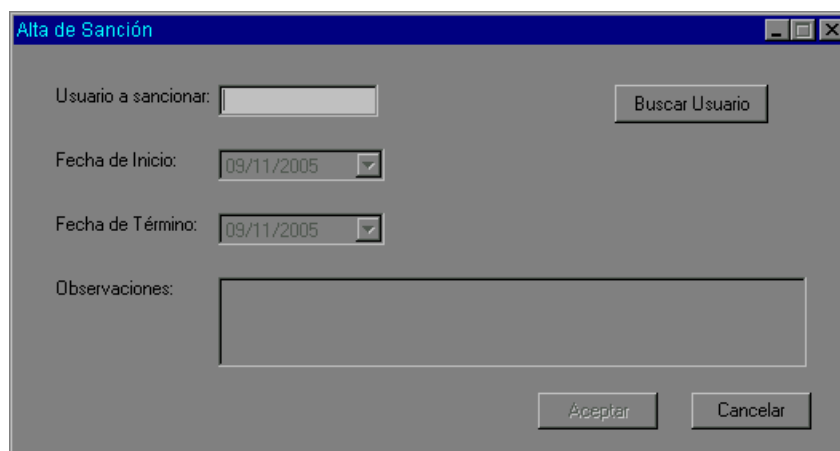


Ver Identificación

Identificación:

Submenú Sanciones

Alta. En este menú se registran las sanciones a los alumnos.



Alta de Sanción

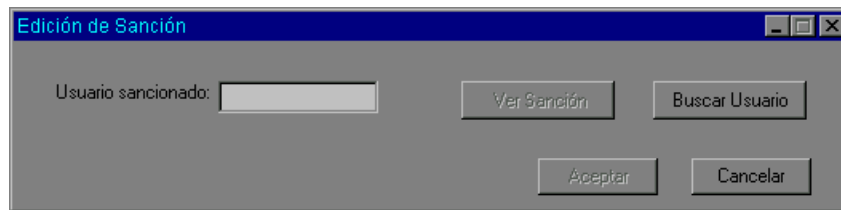
Usuario a sancionar:

Fecha de Inicio:

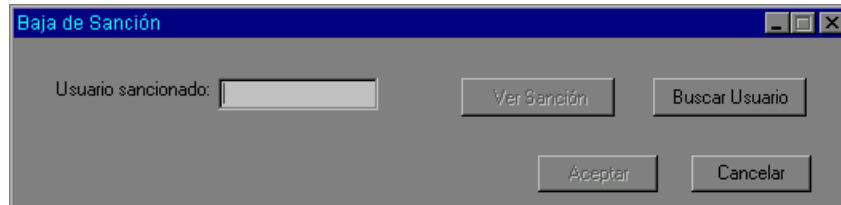
Fecha de Término:

Observaciones:

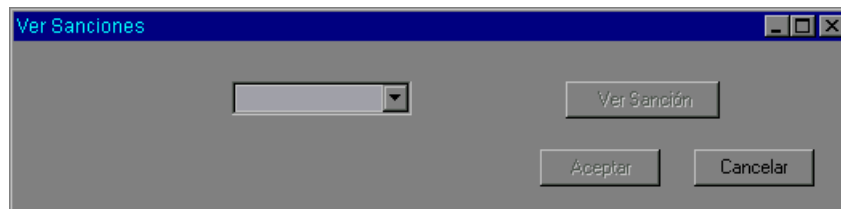
Edición. En este menú se editan las sanciones que tienen los usuarios.



Baja. Aquí se dan de baja las sanciones.

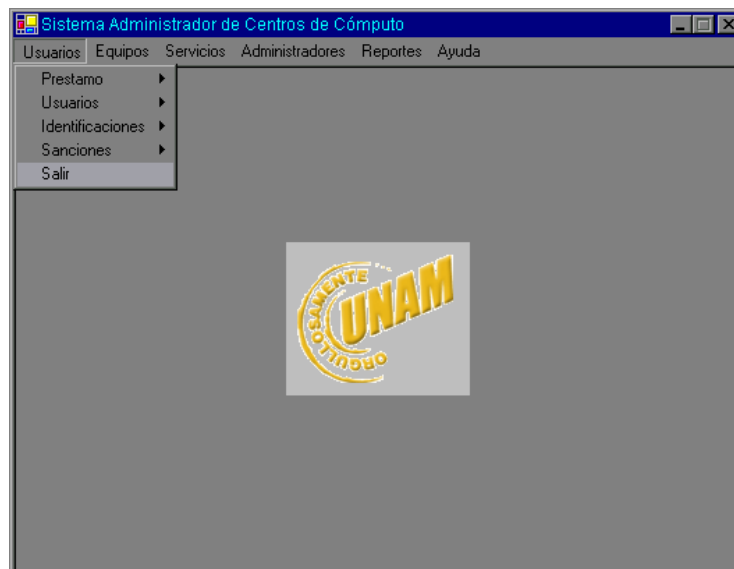


Ver todos. En este menú se pueden ver todos las sanciones vigentes.



Submenú Salir

Este submenú permite cerrar el programa.



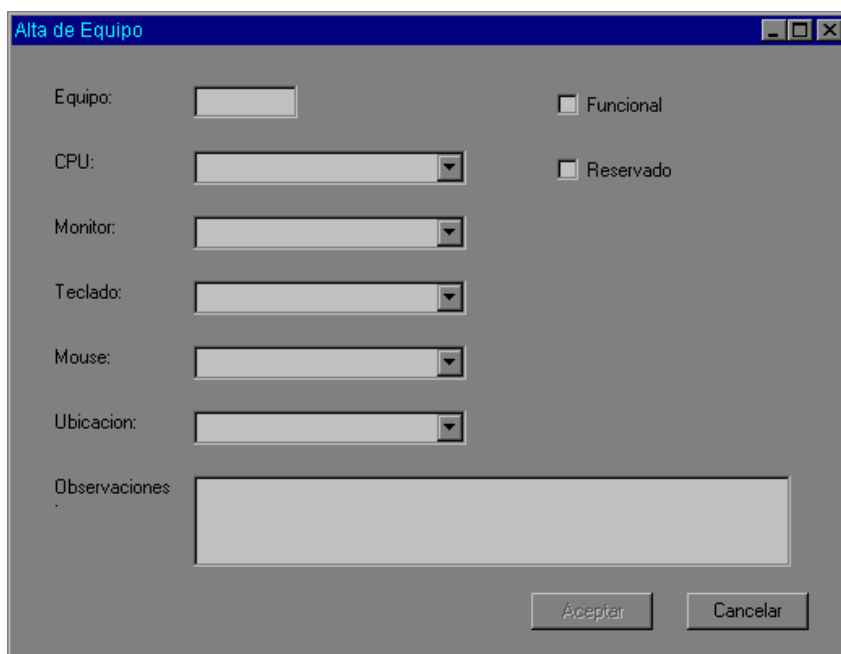
Menú Equipo

Este menú contiene todo lo relacionado a los equipos de cómputo que se encuentran dentro del Centro.

Contiene las siguientes opciones:

Submenú Equipo

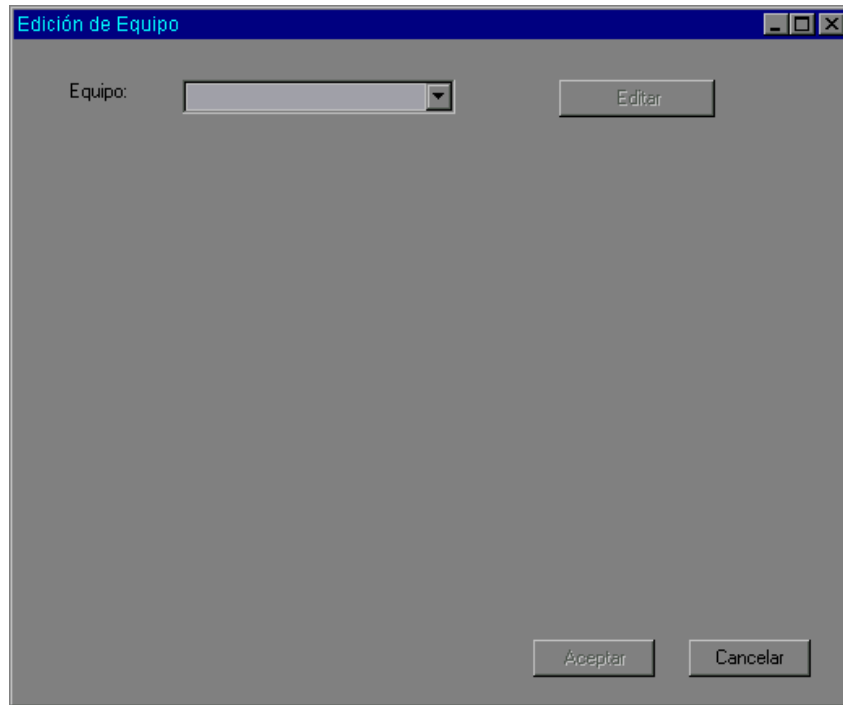
Alta. En este menú se dan de alta los equipos que se encuentran dentro del Centro de Cómputo.



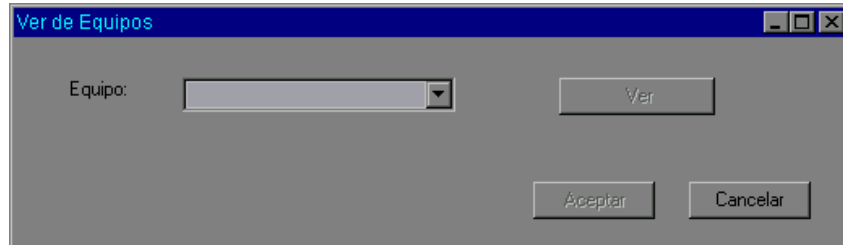
The image shows a Windows-style dialog box titled "Alta de Equipo". The dialog has a blue title bar with standard window controls. The main area is gray and contains the following elements:

- Equipo:** A text input field.
- CPU:** A dropdown menu.
- Monitor:** A dropdown menu.
- Teclado:** A dropdown menu.
- Mouse:** A dropdown menu.
- Ubicacion:** A dropdown menu.
- Funcional:** A checkbox.
- Reservado:** A checkbox.
- Observaciones:** A large text area for notes.
- Buttons:** "Aceptar" and "Cancelar" buttons at the bottom right.

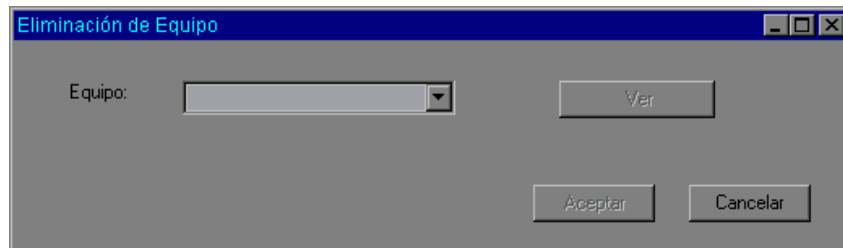
Edición. Aquí se editan los datos de los equipos.



Ver todos. En este menú se pueden ver todos los equipos.

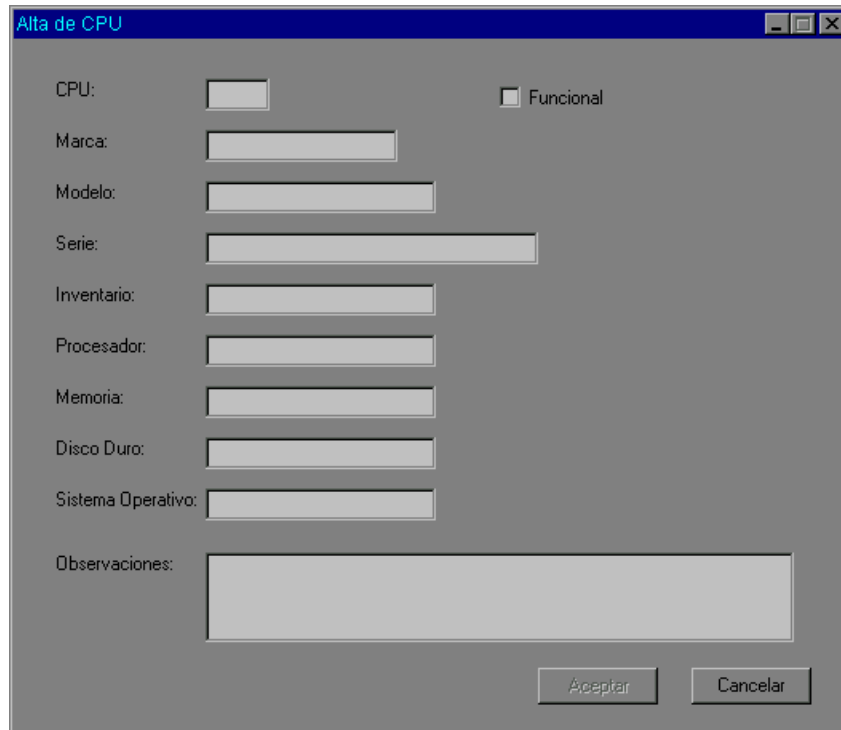


Eliminación. En este menú se pueden eliminar los datos de los equipos.



Submenú CPU

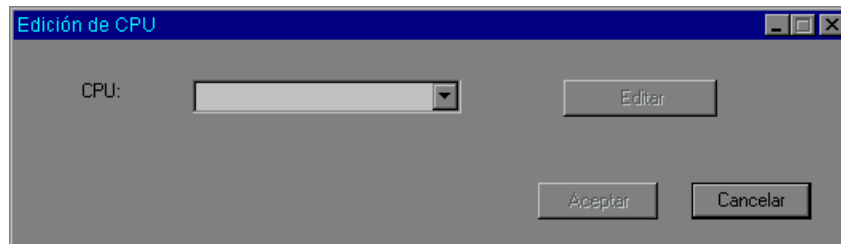
Alta. En este menú se dan de alta los CPU's que se encuentran dentro del Centro de Cómputo.



The 'Alta de CPU' dialog box contains the following fields and controls:

- CPU: Funcional
- Marca:
- Modelo:
- Serie:
- Inventario:
- Procesador:
- Memoria:
- Disco Duro:
- Sistema Operativo:
- Observaciones:
- Buttons: Aceptar, Cancelar

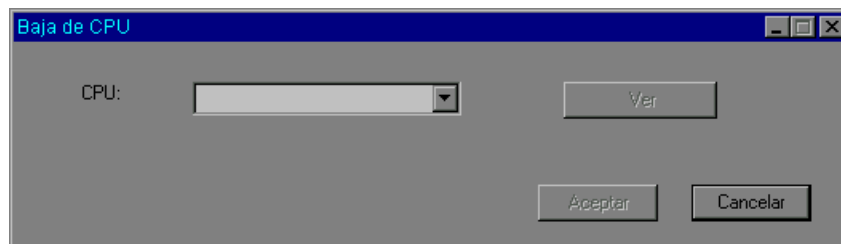
Edición. Aquí se editan los datos de los CPU's.



The 'Edición de CPU' dialog box contains the following fields and controls:

- CPU:
- Buttons: Aceptar, Cancelar

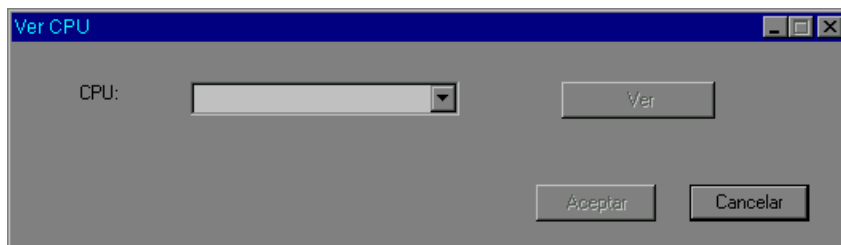
Baja. En este menú se dan de baja los CPU's.



The 'Baja de CPU' dialog box contains the following fields and controls:

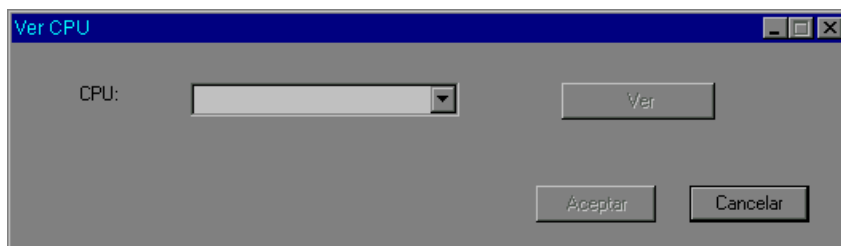
- CPU:
- Buttons: Aceptar, Cancelar

Ver todos. En este menú se pueden consultar los datos de todos los CPU's que hay dentro del Centro.



The screenshot shows a dialog box titled "Ver CPU". It features a "CPU:" label followed by a dropdown menu. To the right of the dropdown is a "Ver" button. Below the dropdown and "Ver" button are two buttons: "Aceptar" and "Cancelar".

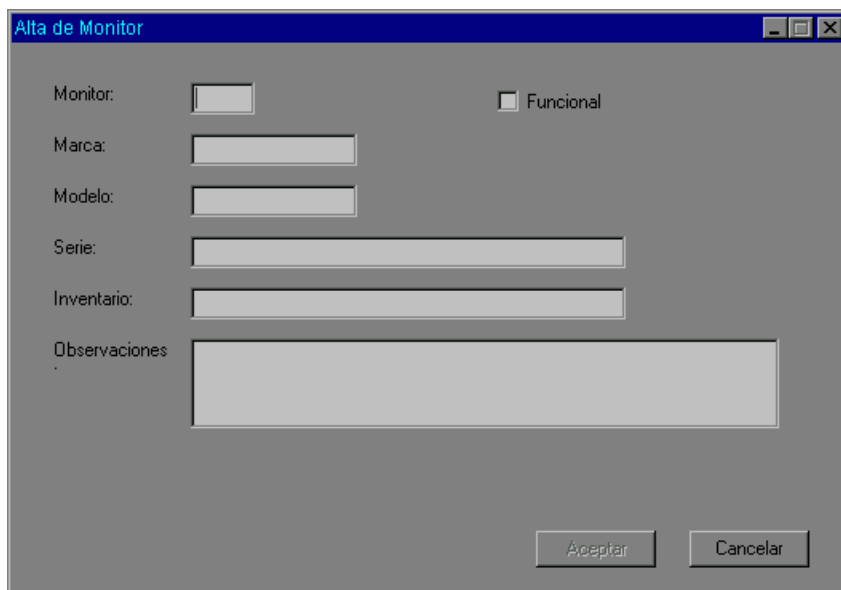
Eliminación. En este menú se pueden eliminar los datos de los CPU's, siempre y cuando no este siendo utilizado por un Equipo.



This is an identical screenshot of the "Ver CPU" dialog box as shown above, featuring a "CPU:" dropdown, a "Ver" button, and "Aceptar" and "Cancelar" buttons.

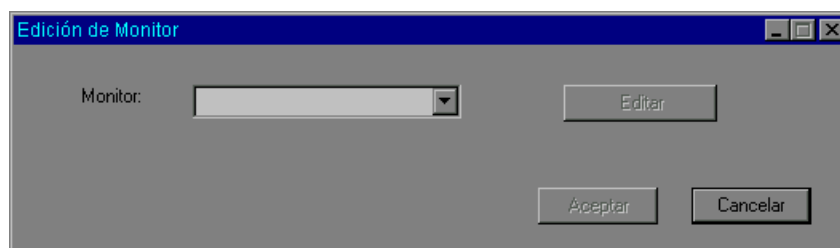
Submenú Monitor

Alta. En este menú se dan de alta los monitores que se encuentran dentro del Centro de Cómputo.



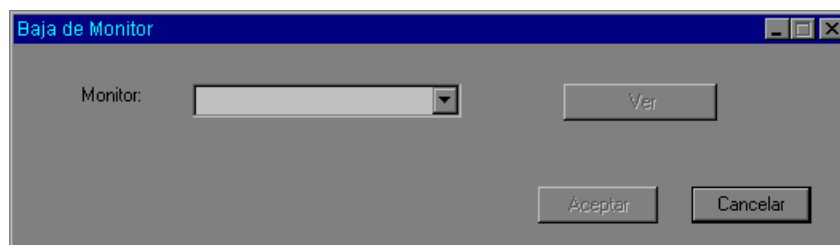
The screenshot shows a dialog box titled "Alta de Monitor". It contains several input fields: "Monitor:" with a small text box, "Marca:" with a text box, "Modelo:" with a text box, "Serie:" with a long text box, and "Inventario:" with a text box. To the right of the "Monitor:" field is a checkbox labeled "Funcional". At the bottom of the dialog are "Aceptar" and "Cancelar" buttons. A large text area for "Observaciones" is located below the "Inventario:" field.

Edición. Aquí se editan los datos de los monitores.



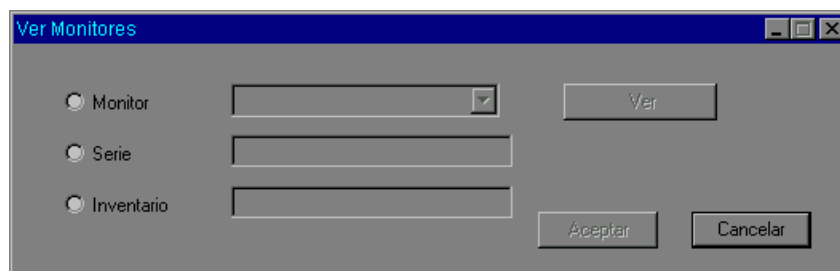
The 'Edición de Monitor' dialog box features a title bar with the text 'Edición de Monitor' and standard window controls. The main area contains a label 'Monitor:' followed by a dropdown menu. To the right of the dropdown is an 'Editar' button. Below these elements are two buttons: 'Aceptar' and 'Cancelar'.

Baja. En este menú se dan de baja los monitores, siempre y cuando no este siendo utilizados por un Equipo.



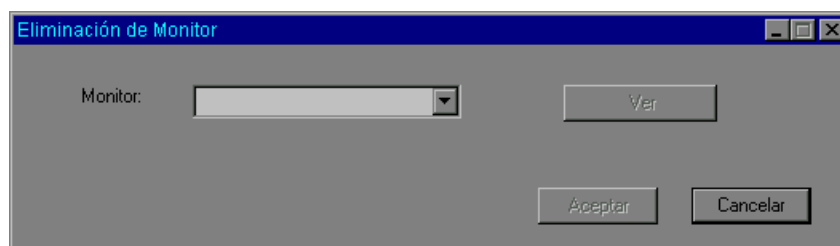
The 'Baja de Monitor' dialog box has a title bar with 'Baja de Monitor' and window controls. It includes a 'Monitor:' label and a dropdown menu. A 'Ver' button is positioned to the right of the dropdown. At the bottom, there are 'Aceptar' and 'Cancelar' buttons.

Ver todos. En este menú se pueden consultar los datos de los monitores.



The 'Ver Monitores' dialog box has a title bar with 'Ver Monitores' and window controls. It features three radio buttons labeled 'Monitor', 'Serie', and 'Inventario'. Each radio button is followed by a text input field. A 'Ver' button is located to the right of the first input field. At the bottom, there are 'Aceptar' and 'Cancelar' buttons.

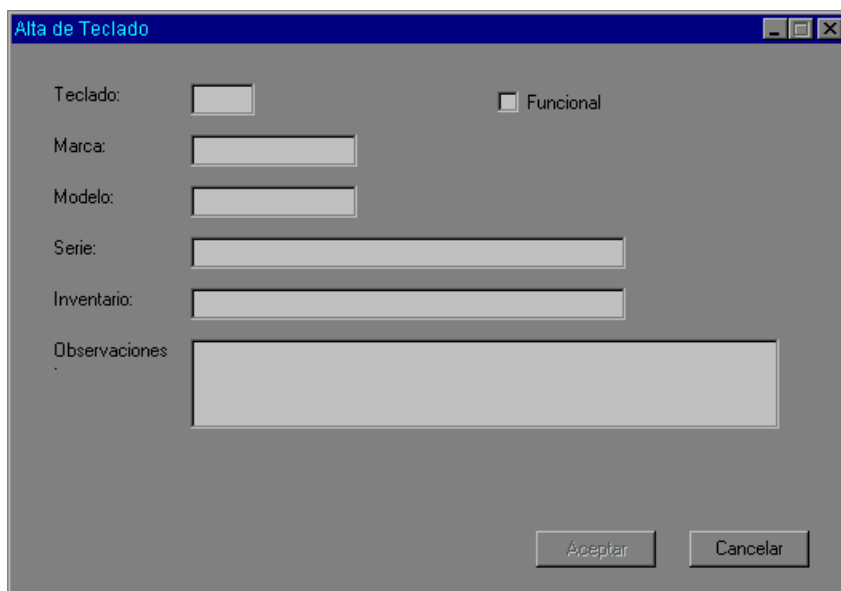
Eliminación. En este menú se pueden eliminar los datos de los monitores.



The 'Eliminación de Monitor' dialog box has a title bar with 'Eliminación de Monitor' and window controls. It contains a 'Monitor:' label and a dropdown menu. A 'Ver' button is to the right of the dropdown. At the bottom, there are 'Aceptar' and 'Cancelar' buttons.

Submenú Teclados

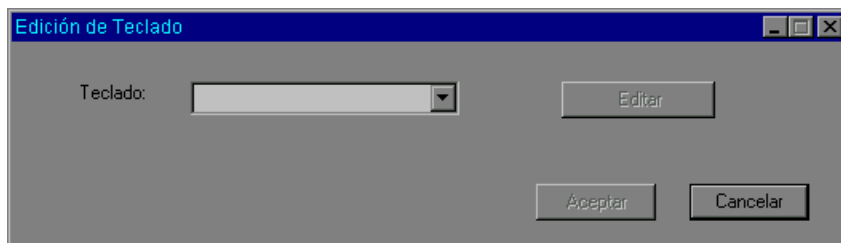
Alta. En este menú se dan de alta los teclados existentes dentro del Centro de Cómputo.



The 'Alta de Teclado' dialog box contains the following fields and controls:

- Teclado:
- Funcional:
- Marca:
- Modelo:
- Serie:
- Inventario:
- Observaciones:
- Buttons: Aceptar, Cancelar

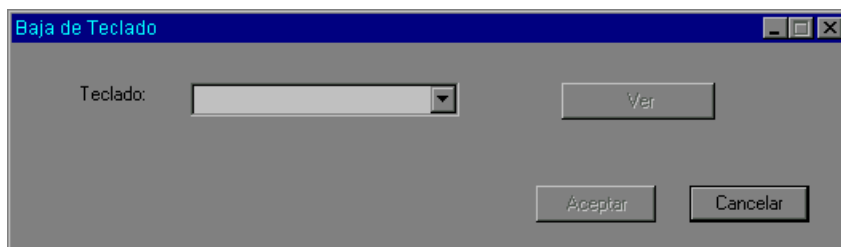
Edición. Aquí se editan los datos de los teclados.



The 'Edición de Teclado' dialog box contains the following fields and controls:

- Teclado:
- Editar:
- Aceptar:
- Cancelar:

Baja. En este menú se dan de baja los teclados, siempre y cuando no este asignado a un equipo.



The 'Baja de Teclado' dialog box contains the following fields and controls:

- Teclado:
- Ver:
- Aceptar:
- Cancelar:

Ver todos. En este menú se puede ver toda la información acerca de los teclados.

The 'Ver Teclados' dialog box features a blue title bar. It contains three radio buttons: 'Teclado', 'Serie', and 'Inventario'. Each radio button is followed by a text input field. To the right of the 'Teclado' field is a 'Ver' button. At the bottom right, there are 'Aceptar' and 'Cancelar' buttons.

Eliminación. En este menú se pueden eliminar los datos de los teclados.

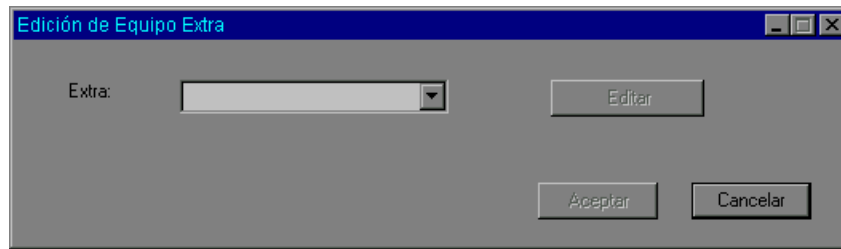
The 'Eliminación de Teclado' dialog box has a blue title bar. It includes a 'Teclado:' label followed by a dropdown menu. A 'Ver' button is positioned to the right of the dropdown. At the bottom right, there are 'Aceptar' and 'Cancelar' buttons.

Submenú Extras

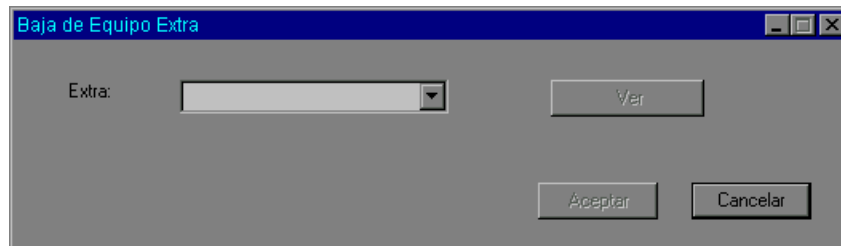
Alta. En este menú se dan de alta los extras que tiene asignado un equipo, como por ejemplo un scanner, una cámara, etc.

The 'Alta de Equipo Extra' dialog box has a blue title bar. It contains a 'Funcional' checkbox. Below it are several input fields: 'Tipo:', 'Equipo:' (with a dropdown arrow), 'Marca:', 'Modelo:', 'Serie:', and 'Inventario:'. At the bottom is a large text area for 'Observaciones :'. 'Aceptar' and 'Cancelar' buttons are located at the bottom right.

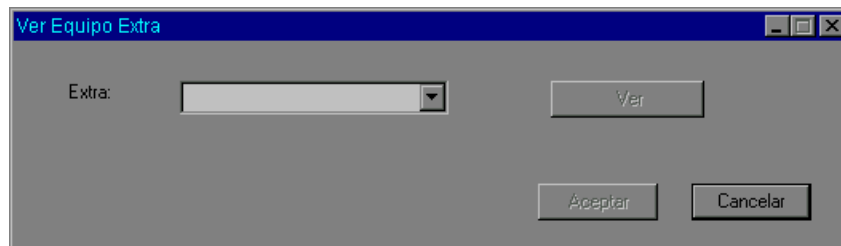
Edición. Aquí se editan los datos de los extras.



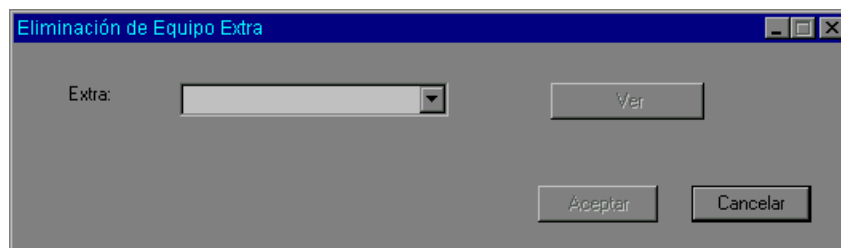
Baja. En este menú se dan de baja los extras.



Ver todos. En este menú se pueden consultar los datos de los extras que se tengan registrados hasta el momento de la consulta.

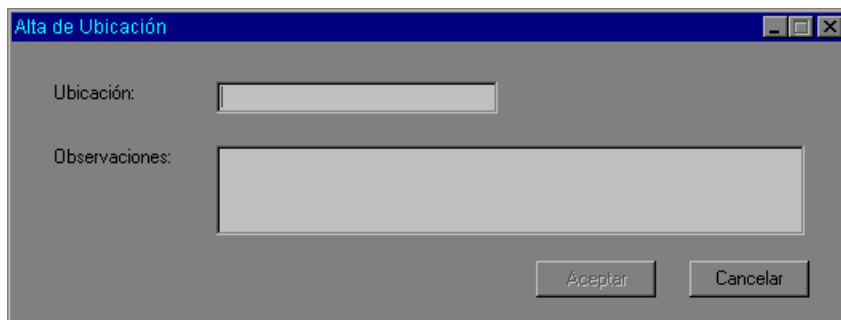


Eliminación. En este menú se eliminar los datos de los extras.



Submenú Ubicaciones

Alta. En este menú se dan de alta las ubicaciones donde se encontrarán dichos equipos.



The 'Alta de Ubicación' dialog box features a title bar with standard window controls. It contains two input fields: 'Ubicación:' with a single-line text box and 'Observaciones:' with a larger multi-line text area. At the bottom right, there are two buttons labeled 'Aceptar' and 'Cancelar'.

Edición. Aquí se editan los datos de tales ubicaciones.



The 'Edición de Ubicación' dialog box has a title bar with window controls. It includes a 'Ubicación:' label followed by a dropdown menu. To the right of the dropdown is an 'Editar' button. Below the dropdown are 'Aceptar' and 'Cancelar' buttons.

Baja. En este menú se dan de baja las ubicaciones.



The 'Baja de Ubicación' dialog box features a title bar with window controls. It contains a 'Ubicación:' label and a dropdown menu. To the right of the dropdown is a 'Ver' button. Below the dropdown are 'Aceptar' and 'Cancelar' buttons.

Ver todos. En este menú se pueden consultar todas las ubicaciones registradas de los equipos.



The 'Ver Ubicación' dialog box has a title bar with window controls. It includes a 'Ubicación:' label and a dropdown menu. To the right of the dropdown is a 'Ver' button. Below the dropdown are 'Aceptar' and 'Cancelar' buttons.

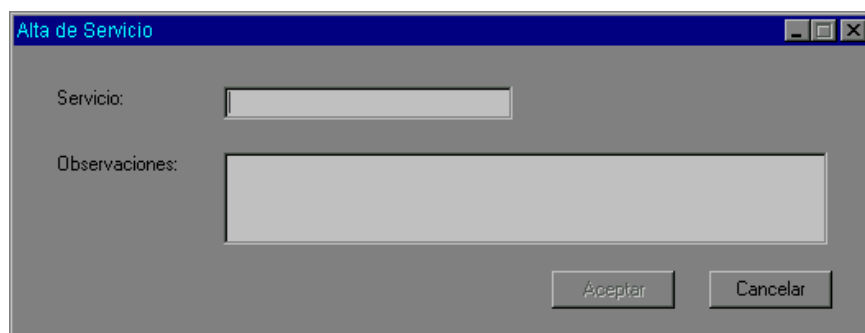
Menú Servicios

Este menú contiene todo lo relacionado con los servicios que brinda el Centro de Cómputo.

Contiene las siguientes opciones:

Submenú Servicios

Alta. En este menú se dan de alta los servicios que brinda el Centro de Cómputo.



The screenshot shows a dialog box titled "Alta de Servicio". It has a blue title bar with standard window controls. The main area is gray and contains two labels: "Servicio:" followed by a text input field, and "Observaciones:" followed by a larger text area. At the bottom right, there are two buttons: "Aceptar" and "Cancelar".

Edición. Aquí se editan los datos de los servicios que brinda el Centro.



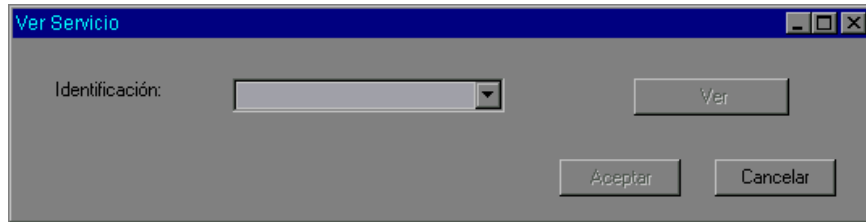
The screenshot shows a dialog box titled "Edición de Servicio". It has a blue title bar with standard window controls. The main area is gray and contains a label "Servicio:" followed by a dropdown menu. To the right of the dropdown is an "Editar" button. At the bottom right, there are two buttons: "Aceptar" and "Cancelar".

Baja. Aquí se eliminan los servicios que están registrados dentro de SACC.



The screenshot shows a dialog box titled "Baja de Servicio". It has a blue title bar with standard window controls. The main area is gray and contains a label "Servicio:" followed by a dropdown menu. To the right of the dropdown is a "Ver" button. At the bottom right, there are two buttons: "Aceptar" and "Cancelar".

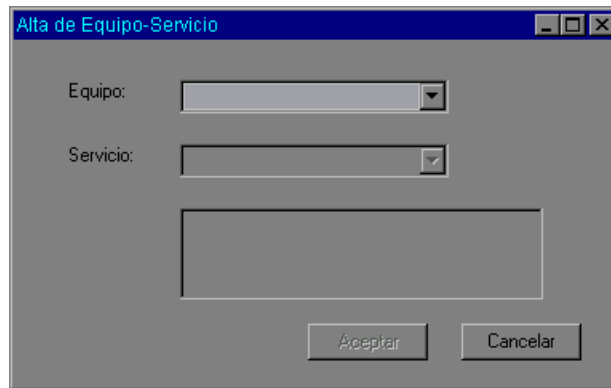
Ver todos. En este menú se pueden ver todos los servicios que brinda el Centro de Cómputo.



The screenshot shows a dialog box titled "Ver Servicio". It features a label "Identificación:" followed by a dropdown menu. To the right of the dropdown is a button labeled "Ver". At the bottom of the dialog are two buttons: "Aceptar" and "Cancelar".

Submenú Equipo-Servicios

Alta. En este menú se dan de alta las relaciones entre un equipo y los servicios que contiene.



The screenshot shows a dialog box titled "Alta de Equipo-Servicio". It contains two labels: "Equipo:" and "Servicio:", each followed by a dropdown menu. Below these dropdowns is a large empty rectangular text area. At the bottom of the dialog are two buttons: "Aceptar" and "Cancelar".

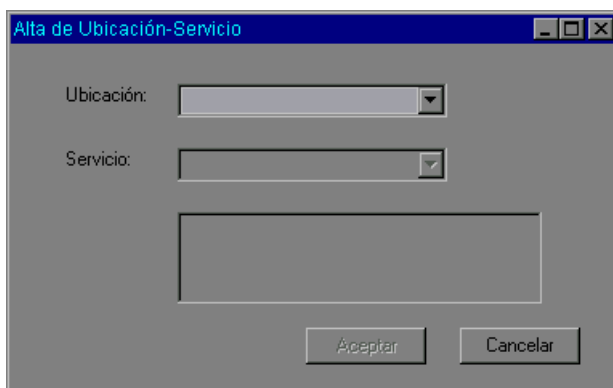
Eliminación. En este menú se eliminan las relaciones entre un equipo y los servicios que contiene.



The screenshot shows a dialog box titled "Eliminación de Equipo-Servicio". It contains two labels: "Equipo:" and "Servicio:", each followed by a dropdown menu. At the bottom of the dialog are two buttons: "Aceptar" and "Cancelar".

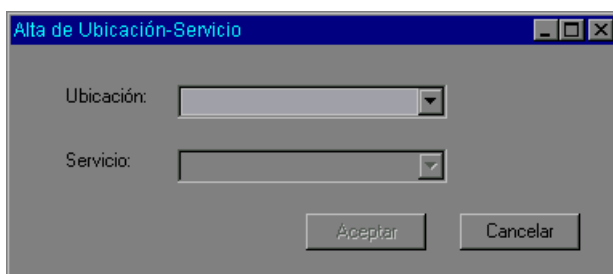
Submenú Ubicación-Servicio

Alta. En este menú se dan de alta las relaciones entre los servicios y su respectiva ubicación.



The screenshot shows a dialog box titled "Alta de Ubicación-Servicio". It contains two dropdown menus: "Ubicación:" and "Servicio:". Below these menus is a large empty rectangular box. At the bottom of the dialog are two buttons: "Aceptar" and "Cancelar".

Eliminación. En este menú se eliminan las relaciones entre los servicios y su respectiva ubicación.

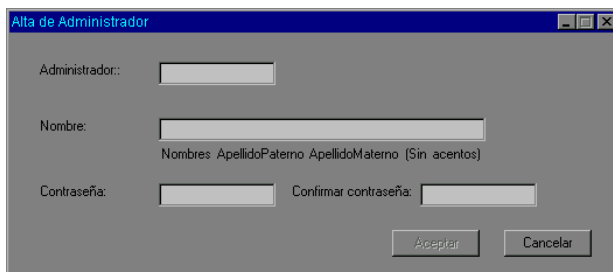


The screenshot shows a dialog box titled "Alta de Ubicación-Servicio". It contains two dropdown menus: "Ubicación:" and "Servicio:". Below these menus are two buttons: "Aceptar" and "Cancelar".

Menú Administradores

Submenú Administradores

Alta. En este menú se dan de alta los Administradores que pueden hacer uso de SACC.



The screenshot shows a dialog box titled "Alta de Administrador". It contains the following fields and controls:

- Administrador:
- Nombre:
Nombres ApellidoPaterno ApellidoMaterno (Sin acentos)
- Contraseña: Confirmar contraseña:
- Buttons: Aceptar, Cancelar

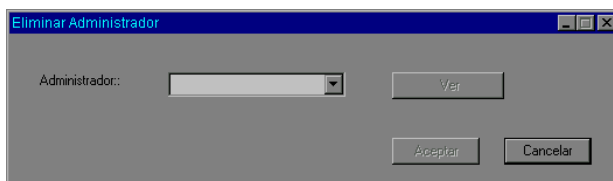
Edición. Aquí se editan los datos de los Administradores que pueden hacer uso de SACC.



The screenshot shows a dialog box titled "Edición de Administrador". It contains the following fields and controls:

- Administrador:
- Buttons: Aceptar, Cancelar

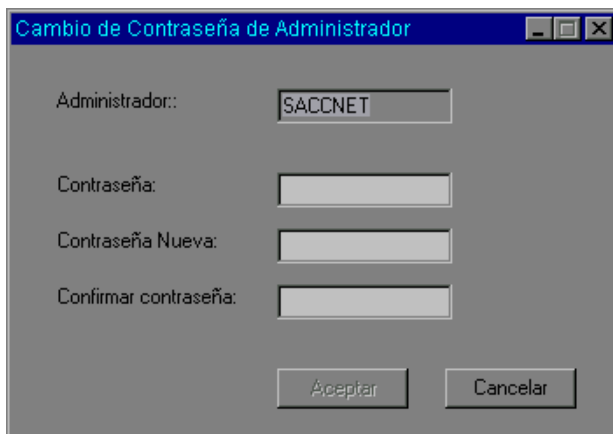
Eliminación. En este menú se eliminan los registros de los Administradores de SACC.



The screenshot shows a dialog box titled "Eliminar Administrador". It contains the following fields and controls:

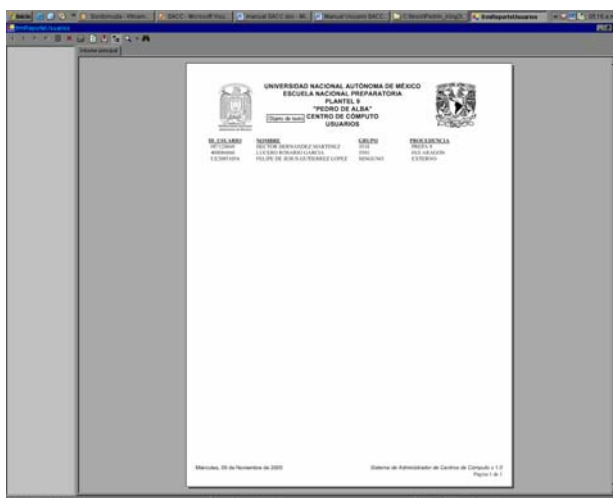
- Administrador:
- Buttons: Aceptar, Cancelar

Cambio de contraseña. En este menú se puede cambiar la contraseña del Administrador actual.



Menú Reportes

Submenú Reporte de usuarios. Este permite elegir los parámetros con los que se creará el reporte de usuarios.



El formato de los reportes es muy semejante al anterior.

Submenú Reporte de Equipos. Este permite elegir los parámetros con los que se creará el reporte de Equipos.

Submenú Reporte de Servicios. Este permite elegir los parámetros con los que se creará el reporte de Servicios.

Submenú Reporte de Identificaciones. Este permite elegir los parámetros con los que se creará el reporte de Identificaciones.

Submenú Reporte de Uso del Centro. Este permite elegir los parámetros con los que se creará el reporte de Uso del centro.

Submenú Reporte de Sanciones. Este permite elegir los parámetros con los que se creará el reporte de Sanciones.

Submenú Reporte de Ubicaciones. Este permite elegir los parámetros con los que se creará el reporte de Ubicaciones.

Submenú Reporte de Equipo-Servicio. Este permite elegir los parámetros con los que se creará el reporte de las relaciones Equipo-Servicio.

Menú ayuda

Submenú Ayuda. Este contiene una liga al manual de usuario que podrá utilizar para el buen manejo de SACC.

Bibliografía

ARCHER, Tom, WHITECHAPEL, Andrew, *Inside C#, Second Edition*. Editorial Microsoft Press, Estados Unidos de América, 2002.

BIRNIOS, Mariano, *Visual Basic .Net*. Editorial MP Ediciones, Buenos Aires Argentina, 2002.

CLARK, Dan, *Introducción orientada a objetos con Visual Basic .Net.*, Editorial ANAYA, Madrid España, 2002.

EZEQUIEL ROZIC, Sergio, *Bases de Datos y su aplicación con SQL*, Editorial MP Ediciones, Buenos Aires, 2004.

HALVORSON, Michael, *Aprenda ya Microsoft Visual Basic .Net*, McGraw Hill, Madrid España, 2002.

MATTHEWS, Martin S, *El libro de Microsoft Windows Server 2003*, Editorial ANAYA, Madrid España, 2003.

PORATTI, Gustavo Gabriel, *Windows Server 2003*, Editorial MP Ediciones, Buenos Aires, 2004.

SCHMULLER, Joseph, *Aprendiendo UML en 24 horas*, Editorial Prentice Hall, México, 2000.