13

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

PROPUESTA DE DESARROLLO DE UN SISTEMA DE INFORMACION PARA LA ADMINISTRACION Y AUTOMATIZACION DE ASIGNACIONES Y EL CONTROL DE INVENTARIO DE EQUIPO DE COMPUTO EN LA EMPRESA SOFTTEK MEXICO

T E S I S

PARA OBTENER EL TITULO DE:

INGENIERO EN COMPUTACION

P R E S E N T A N:

FIDEL CERVANTES RODRIGUEZ
EDUARDO JIMENEZ RUIZ
CECILIA MONTOYA GONZALEZ
CARLOS ROSETE TETETLA
CARLOS SANCHEZ VICARIO

Directora de tesis: Ing. Lucila Patricia Arellano Mendoza

MEXICO, D. F. 2001

UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DEDICATORIAS

Fidel Cervantes Rodríguez

Agradecimientos

Agradezco infinitamente el esfuerzo y dedicación de todos mis profesores de la Facultad de Ingeniería por forjar en mí el conocimiento, respeto, tenacidad y amor por mi Universidad, que me han guiado a lo largo de mis estudios profesionales que al día de hoy con la presente Tesis me ayudan a alcanzar una de las metas más importantes de mi vida.

Un especial agradecimiento a la Ing. Lucila Arellano por su apoyo en el desarrollo de la presente Tesis.

Un especial agradecimiento a mi esposa que me ha acompañado y apoyado desde el inicio de mis estudios profesionales. Marlene, gracias.

Un profundo agradecimiento a toda mi familia que me ha apoyado incondicionalmente y en todo momento a lo largo estos años.

Un especial agradecimiento:

Mis padres.
Antonio Cervantes Alonso.
Rosa Rodríguez de Cervantes.

Mi hermano y distinguida esposa.
Ing. Filemón Cervantes Rodríguez
Sra. Socorro Márquez Hernández

Doy gracias a Dios por permitirme contar con todos ellos a los cuales dedico mi titulación.

Con el presente trabajo termina un ciclo más en la vida de cada uno de nosotros, un ciclo donde tuvimos experiencias buenas y malas.

Un ciclo que todo estudiante sueña con terminar algún día. Pero ningún estudiante podría lograrlo sin la ayuda de todas las personas que estuvieron y están con ellos aún, como por ejemplo, los profesores, compañeros de clase amigos y sobre todo los familiares.

En este ciclo aprendimos muchas cosas, además de la cuestión académica, aprendimos, sobre todo de manera personal, a valorar la amistad entre muchas otras cosas.

Yo por mi parte quiero agradecer de manera muy especial a la Señora Engracia González Gómez, mi mamá que en todo momento me ha apoyado no sólo en mis estudios sino también en mi vida personal, yo sé que ella siempre estará conmigo, que siempre podré contar con ella, quien ha sido la persona más importante a lo largo de mi vida, junto con mi papá.

Al Señor Antolín Montoya, mi papá quien a su manera también ha estado conmigo tanto en las buenas como en las malas.

A mis hermanos, que en cierta manera fueron los responsables de que yo esté a estas alturas de mi formación profesional, ya que por ser la mayor, "tenía que ponerles el ejemplo".

A Ileana Cecilia que ha sido la razón para seguir adelante y superándome.

A Salvador Aguilar Mar, quien ha sido una persona muy importante y quién más me ha apoyado en todo no sólo en el desarrollo de este trabajo y él sabe perfectamente lo que todo esto significa para mí porque siempre ha estado muy cerca.

También quiero agradecer a mis amigos que hace mucho no veo pero que siempre me acuerdo de ellos; a Consuelo Huesca Rosas, Beatriz López Nájera, Jose Luis Yañez Rodríguez y a Guillermo Salvador Enríquez Galindo quien me ha hecho ver muchas cosas y me ha ayudado mucho en los momentos más difíciles, muchas gracias memo.

Quiero decirles a todos los anteriores que los quiero mucho y que siempre pueden contar conmigo para lo que deseen.

A la Ingeniera Lucila Patricia Arellano Mendoza quien con su guía pudimos sacar el trabajo adelante.

Y finalmente a todos los profesores que nos dieron su cátedra, bueno solamente con los que aprobamos, ya que gracias a ellos estamos aquí, terminando nuestra carrera.

CECILIA MONTOYA GONZALEZ.

A mis padres, por su amor y apoyo incondicional que me han brindado durante todos estos años, porque me han dado las bases necesarias para seguir creciendo y triunfando en la vida.

En recuerdo al Lic. Luis F. Carmona, quien me enseñó la importancia del conocimiento y el gran valor de un libro.

A todas las personas que han estado presentes en mi vida, que me han dado un lugar en su corazón, pero sobre todo, que han dejado en mi los más vivos recuerdos, experiencias y me han enseñado el valor de una amistad.

Carlos Rosete Tetetla.

Después de vivir algún tiempo, me he percatado de que las experiencias acumuladas tienen gran valor, ya que son las formadoras de la mentalidad en cada persona, y porque gracias a ellas me encuentro actualmente en la capacidad de desempeñarme como el profesional en que estoy convirtiéndome.

En muchas ocasiones he sentido caer en estados absorbentes, pero afortunadamente, con la guía y apoyo de personas valiosas, logré superar los retos que se han presentando, los cuales lejos de ser obstáculos, se convirtieron en zonas de oportunidad que sirvieron para obtener una formación integral.

En el momento en que realizo esta dedicatoria, considero el trabajo de tesis como mi máximo de liderazgo personal, ya que apliqué, al igual que los compañeros tesistas, la mayor parte de los conocimientos y experiencias adquiridos a lo largo de mi formación para su realización.

A continuación, quiero agradecer a las siguientes personas su valiosa colaboración en mi formación, ya que ello me permitió concluir exitosamente los estudios de licenciatura, así como mejorar mis criterios y perspectivas de vida.

Persona	Motivo de agradecimiento
María Dolores Vicario Fuster	Mujer amorosa de incansable lucha. Por su enorme paciencia, prudencia y confianza. Por creer en mí.
Lic. José Alberto Amezcua Manjarrez y Quevedo	Por su gran apoyo y cariño a lo largo de las distintas etapas de mi vida.
Lic. María Dolores Sánchez Vicario	Por su orientación y paradigma de dignidad y fuerza.
Lic. Enrique Fabián Cervantes	Por su gran apoyo especialmente como maestro y amigo en mi etapa de estudiante en la Facultad de Ingeniería.
Ing. Lucila Patricia Arellano Mendoza	Por su excelente guía a lo largo del proceso del desarrollo de la tesis y titulación.
Ing. Alfredo Rico Garza	Por su amistad y la confianza depositados en mi persona.
Ing. Marco Aurelio Torres Herrera	Por ser ejemplo de generosidad, fuerza, dignidad, conocimiento e inteligencia aplicados a lo largo de una larga y próspera vida.
Al personal docente de la Fac. de Ing.	Por su valioso apoyo en mi formación ético-académica.

Como estudiante adquirí el compromiso de prepararme para servir mejor a la sociedad. Ahora como un profesional deseo cumplir con el compromiso adquirido, así como seguir capacitándome para mejorar en el desempeño de mi vida y profesión.

Carlos Sánchez Vicario

Prólogo

La creciente demanda de servicios de cómputo, su manejo, diseño, operación, mantenimiento, servicio y control, ha creado la necesidad de implementar métodos y procedimientos adecuados a esta especialidad.

Conscientes de lo anterior y con el objeto de cubrir tal necesidad, se desarrolló la presente tesis, basada en conceptos, métodos y diagramas aplicados al desarrollo del software para manejar y controlar los recursos informáticos de una empresa.

Softtek México es la empresa para la cual se realizó el análisis y el diseño para su aplicación, considerando que es una empresa cuya actividad preponderante es la "Fabricación de Software", con el consabido empleo de equipo y material de cómputo.

En el primer capítulo de la presente tesis se hace una introducción al ámbito de la empresa, su historia, objetivos y actividades primordiales, así como las características específicas actuales del manejo y administración de equipo de cómputo del área de soporte técnico, tema fundamental y razón de ser de la presente.

En el segundo capítulo se definen los conceptos necesarios para el diseño del sistema a desarrollar, como son los elementos que componen las Bases de datos, su definición, construcción y manipulación de datos; Metodologías utilizadas (Estructurada y Orientada a objetos); Teoría de inventarios que señala los elementos a considerar para su manejo y control; Las normas y estándares que rigen la Calidad en el software y la Teoría de redes, ya que el sistema se manejará a ese nivel.

El Análisis Preliminar del problema nos permite ver, en el tercer capítulo, la situación actual del manejo y control de inventarios de equipo de cómputo, su asignación, responsabilidad y alcances; las herramientas comerciales para el manejo del inventario y la conveniencia de diseñar un sistema particular y específico para cubrir las necesidades de la empresa y sus ventajas sobre el empleo de sistemas de control comerciales.

Es en este capítulo donde se define la propuesta de solución al manejo, operación y control de los inventarios de equipo de cómputo de Softtek cubriendo las necesidades específicas de asignación, responsabilidades, tiempos y características de todos los elementos del equipo de cómputo asignado a cada persona que lo solicite y requiera.

El cuarto capítulo es el pilar del trabajo presentado, ya que en él se definen el análisis y diseño del sistema.

En el análisis se describen los escenarios según los casos de uso, definición de objetos, identificación de clases, objetos y atributos con todos los diagramas que interrelacionan las actividades y funciones que desarrollará el sistema.

El Diseño del sistema involucra los patrones del mismo, las clases y objetos necesarios para el manejo de las bases de datos y sus operaciones, los modelos entidad-relación y la especificación de todos los elementos de salida, como son las pantallas que manejará el usuario y los reportes que se podrán generar.

En el Planteamiento del Desarrollo del sistema, se establece el prototipo del sistema, con las correspondientes pruebas y resultados en función de lo definido en el capítulo anterior.

Cabe mencionar que el objetivo primordial del presente es la propuesta de diseño de un sistema que cubra las necesidades de Control de inventarios de equipo cómputo de la empresa, de forma directa, clara, transparente, eficiente y amigable, tanto para el usuario como para el área ejecutiva permitiéndoles tener las herramientas necesarias para tomar la mejor decisión y optimizar los recursos en pro de una mayor productividad.

Índice

Introducción

La empresa Softtek México es una empresa dedicada a la Tecnología de Información cuya base del negocio es el desarrollo software. Los desarrollos de software son organizados y elaborados a través de lo que se denomina "Proyectos" para los cuales se agrupa un determinado número de recursos humanos e infraestructura como bien puede ser equipo de cómputo, oficinas, etc.

Los recursos humanos son seleccionados minuciosamente de acuerdo al perfil profesional y conocimientos de tal manera que cumplan con los requerimientos de cada proyecto, lo cual da lugar a un desarrollo exitoso. De manera similar, en el caso de la planeación y organización de infraestructura a utilizar en el proyecto, depende de la naturaleza del mismo o requerimientos que la empresa (en lo sucesivo denominado cliente) necesite.

En muchos de los casos, una de las condiciones estipuladas por el cliente, es que el desarrollo se lleve a cabo dentro de sus instalaciones. Por mencionar un caso muy especial de este tipo, es el desarrollo de sistema de información para un banco como puede ser su Intranet o Extranet; por lo general se pide que el desarrollo se realice dentro de sus instalaciones por cuestiones de seguridad y confidencialidad de la información que se maneja. Cabe mencionar que la empresa Softtek desarrolla un 70% de sus proyectos en las instalaciones del cliente, las cuales pueden estar situadas en cualquier parte de la República Mexicana, motivo por el cual Softtek tiene que desplazar a su personal con equipo de cómputo a laborar fuera de sus instalaciones.

La actividad de desplazar el equipo de cómputo a otras instalaciones o localidades parecería sencilla, sin embargo, cuando esta se repite con frecuencia, suele ser muy difícil de controlar y/o administrar, tomando en cuenta que la "Base del Negocio" de la empresa Softtek México NO ES el estar desplazando equipo de cómputo — como bien pudiera ser el caso de una empresa cuyo negocio sea el arrendamiento de equipo de cómputo —.

El área de Soporte Técnico de Softtek es la que lleva a cabo la actividad de asignación de equipo, sin embargo, se ha dado cuenta de lo difícil que puede llegar a ser el administrar el inventario de equipo de cómputo en periodos de alta demanda y movimiento de equipo derivado de los requerimientos de cada uno de los proyectos y lo que esto puede repercutir en el desarrollo y tiempos de entrega de los mismos, lo cual puede indicar pérdidas económicas a la empresa. Este hecho motivó a esta área a plantearle a la parte administrativa de la empresa la necesidad de crear el diseño de un sistema de información que les facilite el desarrollo de esta actividad y evite los posibles problemas mencionados anteriormente. Este planteamiento fue aceptado por el área administrativa de la empresa, sin embargo, Softtek está muy orientado en la atención a sus clientes y al desarrollo de sus proyectos, sin embargo, en ese momento de decisión no se tenían recursos humanos disponibles para su realización. Tomando en cuenta lo anterior, se ofreció el diseño de este sistema de información a gente externa a la empresa, lo cual motivó al presente equipo la realización del presente diseño como parte de su tesis para titulación, siendo esto aceptado por la empresa.

Objetivo

Analizar las necesidades de administración de asignación e inventario de equipo de cómputo del área de sistemas en la empresa de Softtek de México y proponer alternativas de solución para el desarrollo de un sistema de información que facilite los procesos administrativos de esta área.

Capítulo 1

Marco General de la Empresa Softtek México

1. Marco General de la empresa Softtek México

1.1 Softtek una empresa de servicios

Softtek, es una empresa 100% mexicana enfocada al desarrollo y consultoría de sistemas, con presencia en 9 países del mundo, e innovando en México las fábricas de software.

1.1.1. Historia

Softtek fue creada el 1 de diciembre de 1982 por Gerardo López García. La compañía nace como proveedora de servicios de software, dirigida a servir a las empresas medianas y grandes en sus necesidades tanto de desarrollo como de soporte de software.

Con más de 18 años de trayecto, Softtek ha logrado colocarse, tanto en México como en los mercados internacionales, debido a que ofrece soluciones inteligentes, basadas en tecnologías de vanguardia mundial.

Softtek crea una nueva estrategia en servicios de desarrollo: Fábricas de Software. Estudios recientes realizados por la empresa Arthur D. Little señalan que en Estados Unidos existe un déficit de por lo menos 250 mil profesionales de sistemas de información que las universidades no han podido satisfacer.

Ante esta situación, muchas firmas norteamericanas buscan alternativas en otros países para cubrir sus necesidades. Por esta razón Softtek hace de México una excelente alternativa principalmente para el mercado estadounidense que permite hacer frente a las necesidades de desarrollo de sistemas ante el reto del nuevo milenio.

En junio de 1996 arranca el proyecto y se comienzan a acondicionar las Fábricas. En febrero de 1998 el Presidente Ernesto Zedillo inauguró oficialmente las Fábricas de Software de Softtek ubicadas en Monterrey N.L. en una superficie de 10 mil metros cuadrados.

Actualmente cuenta con un equipo de más de 2200 profesionales expertos distribuidos en Argentina, Brasil, Chile, Colombia, España, Estados Unidos, México, Perú y Venezuela. Las áreas que la conforman están especializadas en sectores específicos de tecnologías de información. Elabora soluciones que van desde la consultoría para establecer un sistema de nómina hasta el control automatizado de plantas industriales, manejo de bases de datos, construcción de redes internas y desarrollo de proyectos en e-Bussiness.

1.1.2. Funcionalidad

El modelo organizacional está constituido por ocho Megaprocesos, los cuales están diseñados en función de actividades específicas para hacer eficientes los compromisos con los clientes.

Desarrollo de Nuevos Negocios, Comercialización, Soluciones y Servicios.

Agregan valor a los servicios que se ofrecen. Para que éstos funcionen contamos con otros cuatro que les dan soporte.

Ejecutivo, Infraestructura, Capital Humano y Administración del Conocimiento.

Al trabajar bajo este esquema cada persona tiene una visión no sólo de su función en particular, sino del proceso global en el que participa. Además, estos modelos ayudan a entender a los integrantes de la empresa a que forman parte de un todo que debe caminar a la perfección.

1.1.3. Clientes, una lista de compromisos cumplidos.

Gracias a los clientes, Softtek tiene experiencia en la mayoría de las plataformas que existen en el mercado, tanto de software como de hardware. Ha realizado proyectos exitosos para importantes compañías tales como:

Serfin, Bancomer, Banamex, Citibank, Banco Latino, Banpais, Banco Interamericano de Finanzas, Interbank, Banco Wise, BBV, Alestra, Telmex, Cemex, Apasco, TMM, IMSS, PROCESAR, GNP, Dupont, Good-Year Oxo, ISSSTE, Colgate, HP, Probursa, Xerox, Iusacell, Afore Garante, IBM, por mencionar algunas.

1.1.4. Productos y Servicios

La evolución de la industria genera diversos cambios en todas y cada una de las empresas que las integran. Estos cambios obligan a las organizaciones a evolucionar al mismo ritmo, ya que se enfrentan a necesidades cambiantes y cada vez más demandantes, así como a un ambiente cada día más competido.

Softtek es una empresa en búsqueda continua de tecnologías de información de vanguardia, que vayan acorde con la evolución de los mercados. Los 19 años de experiencia les han permitido establecerse como una organización sólida y confiable, que sabe manejar los procesos de cambio propios, así como todos aquellos que los clientes atraviesan.

1.1.4.1. Desarrollos a la Medida

Actualmente existe una gran variedad de paquetes de software, de fácil y rápida implantación. Sin embargo, las empresas evolucionan y continuamente modifican sus procesos operativos para mantener y descubrir nuevas ventajas competitivas. Esto hace necesaria la existencia de desarrollos a la medida, que permitan contar con herramientas de Tecnología de Información diseñadas bajo requerimientos específicos, para servir como apoyo en la operación diaria y para obtener las metas establecidas en su estrategia de negocio.

La destacada labor y metodología de Softtek permite que un trabajo "artesanal" como el desarrollo de sistemas a la medida sea diseñado por un equipo y continuado o mantenido a lo largo del tiempo por otro equipo diferente, sin perder las características únicas bajo las cuales fue diseñado.

Esta metodología también permite establecer y cumplir un compromiso de tiempo, costo y alcance del proyecto, determinado desde el inicio del desarrollo. Esto da al cliente la posibilidad de planear y diseñar las estrategias a las que dará soporte el sistema con la confianza de que estará listo en el tiempo definido

1.1.4.2. Fábricas de Software

La evolución de la industria, y en especial la de tecnología de información, crea curvas de aprendizaje variables y genera costos de adopción tecnológica.

Con el concepto fábricas de software, Softtek utiliza estos factores convirtiéndolos en ventajas para el cliente. Esto es posible debido a que seleccionan el lugar que represente la mayor ventaja en conocimiento, experiencia y costo para llevar a cabo el desarrollo de un proyecto.

Las fábricas de software cuentan con áreas independientes por cliente y se tienen permanentemente en conexión en línea con ellos, para mantenerlos al tanto de lo que ocurre con sus proyectos. El 70% del trabajo es realizado en el lugar de residencia del cliente. Para ello, mientras se desarrolla el proyecto se seleccionan a los mejores profesionales para que se trasladen hasta las instalaciones de los clientes para entender a la perfección las necesidades que tienen.

El 30% del trabajo restante es fabricado en las instalaciones de Softtek, lo cual da principalmente dos ventajas: tarifas reducidas, gracias a que el trabajo se realiza en México y profesionales especializados, quienes desempeñan actividades de una planta de producción utilizando las mejores prácticas, herramientas avanzadas de software y productos de muy alta calidad y vanguardistas.

Las alianzas estratégicas de Softtek permiten ofrecer al cliente diversas líneas de producción con una amplia gama de opciones para utilizar el lenguaje más conveniente para el proyecto.

1.1.4.3. Delta

El servicio Delta otorga a los clientes personal con perfiles y formación especifica orientada a desarrollar tareas técnicas tales como: programación, mantenimiento, documentación, apoyo en pruebas y soporte a la operación de sistemas de información. Este servicio está especializado en herramientas de gran demanda en el mercado para desarrollar actividades técnicas con un alto nivel de productividad y oportunidad. Con el esfuerzo Delta se contribuye a mejorar la relación costo/beneficio al ofrecer servicios de profesionales orientados a tareas especificas de desarrollo de sistemas al mejor precio del mercado.

1.1.4.4. Integración de Sistemas

Los mercados se desenvuelven en entornos cada vez más competidos y con mayores retos. Esta condición obliga a las empresas a ser cada día más eficientes y optimizar sus recursos.

En Softtek se sabe que los clientes buscan soluciones integrales de tecnología de información que les permitan obtener el máximo desempeño en todas sus áreas operativas y hagan posible que los elementos de hardware, software, ERP's (Soluciones Empresariales) y redes de comunicación trabajen en conjunto hacia los resultados deseados.

Al integrar diversos servicios de tecnología de información se han desarrollado metodologías, prácticas y paquetes de soluciones de fácil y rápida implantación para las distintas áreas de cada empresa.

La flexibilidad de la metodología Softtek permite adaptar estas soluciones a las necesidades específicas de cada caso, o bien desarrollar soluciones a la medida para nuevas industrias o áreas de trabajo.

1.1.4.5. Capacitación

La capacitación del personal es una herramienta indispensable para anticipar los cambios tecnológicos e implementar estrategias adecuadas.

Softtek conoce la profunda transformación de la industria educativa en los procesos de Enseñanza-Aprendizaje; por eso ofrece nuevas alternativas para aprender y alcanzar los objetivos de certificación y aplicación del conocimiento, con sistemas apoyados por Internet y Capacitación Asistida por computadora.

Con conceptos como el de capacitación a distancia, Softtek integra en una sola herramienta las características pedagógicas que permiten transferir conocimientos de manera eficiente sin necesidad de movilizar personal o instructores, abatiendo los gastos de capacitación.

La flexibilidad y la amplitud de la oferta en el área de capacitación permite dar al cliente la solución que mejor se adapte a sus necesidades.

1.1.4.6. Capacidades Técnicas

Cliente / Servidor Orientada al desarrollo de aplicaciones de misión crítica, bajo el paradigma cliente/servidor. Estas aplicaciones se caracterizan por ser transaccionales, por ejemplo, sistemas que permiten recibir pedidos y facturar productos y servicios, recibir pagos y generar reportes financieros que, por lo general, requieren la capacidad de aceptar muchos usuarios simultáneos ubicados en puntos geográficamente dispersos a lo largo de uno o varios países.

Internet / Intranet Se ha enfocado al desarrollo de sistemas transaccionales, de colaboración y en general todas aquellas aplicaciones basadas en un ambiente de Web que interactúan con bases de datos. Convertimos las tecnologías relacionadas con Internet en soluciones de negocio, integrando expertos en tecnología de bases de datos y herramientas de colaboración junto a consultores en desarrollo de sistemas en el área aplicativa de la solución, con el apoyo de un equipo de especialistas en tecnologías Web.

Mainframe Ofrece consultoría y desarrollo de soluciones efectivas en tecnología IBM, con un fuerte compromiso de excelencia basado en la experiencia acumulada con la familia de productos DB2, CICS, MQ-SERIES y NET DATA.

Workflow e Imágenes Soluciones que soportan el manejo y control de imágenes y documentos, integrados mediante flujos de trabajo utilizando la tecnología de Eastman Software.

Groupware Ofrece infraestructura de comunicación, colaboración y coordinación a individuos y grupos en una forma integral que rompe barreras de jerarquía organizacional (acceso a información horizontal), distancia (acceso vía Internet) y de organizaciones (acceso a clientes, proveedores y distribuidores).

Definición de arquitectura La arquitectura de los sistemas de información es primordial para asegurar su calidad. La arquitectura debe garantizar la eficiencia de los procesos de negocio, utilizando al máximo los recursos tecnológicos involucrados.

Integración con ERP's Incluye todos los aspectos de la integración de tecnología ERP dentro del proceso de negocio de la empresa, incluyendo reingeniería, consultoría técnica, desarrollo de interfases, adaptación de funcionalidades del ERP y desarrollo de aplicaciones satélites.

Consultoría de Bases de Datos El objetivo es ofrecer profesionales especializados en Bases de Datos a las empresas que requieren un conocimiento profundo de su diseño y manejo.

Reingeniería de sistemas Análisis cuidadoso y detallado del entorno para optimizar los procesos de negocio a través de la solución más adecuada: reingeniería, definición de arquitecturas, integración de tecnologías, etcétera.

Ajuste en producción La experiencia acumulada en proyectos nos permite ofrecer un servicio comprometido y personalizado, orientado al óptimo desempeño en producción, desde la configuración de servidores hasta la administración de dominios y colas de trabajo.

Computer Telephony Integration (CTI) Tecnología que permite relacionar el entorno de red y las bases de datos con los sistemas telefónicos para ofrecer un mejor nivel de servicio.

1.1.4.7. Campos de Acción

A lo largo de su operación, Softtek ha enfocado sus esfuerzos y capacidades a entender la forma de operar y hacer negocio de diversas industrias, con el fin de desarrollar las soluciones de tecnología de información que mejor se adapten a las necesidades de cada segmento.

El trabajo en equipo desarrollado con los clientes ha permitido integrar soluciones específicas para los siguientes segmentos:

Servicios Financieros

Bancos y Aseguradoras

Los constantes cambios en el sector financiero de nuestro país obligan a las instituciones bancarias, tanto públicas como privadas, a buscar la actualización tecnológica continua.

Las herramientas de tecnología de información como el comercio electrónico y la Administración de la Relación con Clientes (CRM) permiten reducir el tiempo de respuesta al mercado para el lanzamiento de nuevos productos y servicios.

Softtek enfoca sus esfuerzos a diseñar soluciones que mediante el uso de herramientas de TI, influyan adecuadamente sobre variables como manejo y disminución del riesgo, tiempos de entrega, tecnología de vanguardia e integración de todas las áreas operativas, para convertirlas en ventajas competitivas con el fin de desarrollar clientes leales y aumentar la captación de nuevos nichos de mercado.

Pensiones

El nuevo esquema de pensiones en México representa uno de los retos más grandes en la historia social y económica de México por la cobertura que tiene y por su importante contribución al ahorro interno del país. La inversión de estos fondos en sociedades de inversión especializadas para fondos de retiro será un detonador en la economía nacional, que a través de inversiones tangibles en infraestructura, productos, servicios y sector público contribuirán para la generación de empleos, recuperación de salarios y crecimiento económico del país.

Softtek ha colaborado con los principales participantes en proyectos relacionados con el SAR y el nuevo esquema de pensiones, aportando conocimiento tecnológico y una visión especializada de la problemática.

Actualmente más del 72% del mercado potencial está afiliado a este nuevo esquema, lo que hace ver que está funcionando bien y se dirige hacia la consolidación; sin embargo, las empresas dedicadas a este ramo, en especial las Afores y las Aseguradoras de Pensiones de Rentas Vitalicias, necesitan vigilar su rentabilidad y la viabilidad de su labor para garantizar su permanencia.

Las iniciativas tecnológicas que pueden aplicarse para apoyar este reto son: Inteligencia de Negocios para la toma de decisiones respecto a productividad, conocimiento de los clientes y Sistemas de Administración de Relaciones con Clientes (CRM) que permitan acercar más al usuario los servicios de pensiones.

Esta es la oferta que Softtek presenta. Algunas de estas ofertas involucran la participación de nuestro aliado Sonda, empresa chilena de servicios informáticos líder en el segmento de Administradoras de Fondos de Pensiones (AFP) en Sudamérica.

Si bien las ofertas en su mayoría están dirigidas a las Afores, también son aplicables a las Aseguradoras de Pensiones de Rentas Vitalicias e Instituciones de Crédito Administradoras de SAR.

Telecomunicaciones

La necesidad de comunicarse no ha cambiado en esencia; lo que ha cambiado es la forma en que se realiza y más concretamente los medios que la hacen posible. Las Telecomunicaciones y la informática convergen para dar paso a la llamada "súper carretera de la información" incrementando el contenido multimedia y el tipo de servicios que los clientes exigen.

La convergencia de productos y servicios demanda a las empresas de telecomunicaciones aumentar al máximo su eficiencia y recursos mejorando el tiempo de respuesta a clientes cada vez más exigentes y con necesidad de servicios avanzados. Por eso muchas empresas han establecido alianzas para lograr cobertura mundial, incorporando dinámicos esquemas de tarifas y ofreciendo nuevas tecnologías de acceso.

Las alianzas complementan la oferta para dar al cliente productos y soluciones de fácil implantación bajo requerimientos específicos. La experiencia y especialización en diferentes áreas y servicios permiten responder a cada cliente con un excelente soporte local y con estrategias dirigidas a economías de escala para darle apoyo personalizado.

Gobierno

La administración pública tiene segmentos diferentes y especializados, tales como finanzas, justicia, seguridad pública, transporte, desarrollo social y contraloría.

La oferta de Softtek en el sector público está orientada a promover la incorporación de las nuevas tecnologías de información en las organizaciones, para coadyuvar en el mejoramiento y modernización de la gestión pública, a través de la asociación tecnológica que se crea durante el desarrollo de los proyectos.

Salud

El mercado de la salud en México se encuentra en una etapa de transición impulsada principalmente por una profunda reforma del sector, que busca promover la calidad y eficiencia en la prestación de servicios, ampliar la cobertura de la atención de las instituciones de seguridad social facilitando la afiliación de la población no asalariada y de la economía informal, y fortalecer el proceso de desconcentración de los servicios de salud. Al mismo tiempo, el IMSS, la principal organización pública de salud, ha iniciado una serie de actividades para fortalecer los esquemas de seguridad social en México.

Softtek tiene las herramientas y el conocimiento del mercado de la salud para habilitar el uso de la tecnología de información en las instituciones del ramo, desde los consultorios particulares hasta las grandes instituciones con múltiples sedes hospitalarias. La tecnología de información ha producido herramientas de cómputo que permiten a las instituciones de salud operar de manera más eficiente, tanto en los procesos de atención médica, a través del uso de expedientes electrónicos y telemedicina, como en sus procesos administrativos.

Manufactura

Las empresas de manufactura no pueden limitar sus esfuerzos a cumplir con estándares regulatorios y emplear procesos con niveles de productividad promedio; es necesario aprovechar todas las herramientas disponibles para sobresalir y obtener el máximo beneficio sobre los recursos invertidos.

Las marcadas diferencias en madurez, infraestructura, participación de mercado y organización de las empresas que componen el sector industrial, hacen necesario un portafolio de soluciones flexibles, de fácil y rápida implantación adaptables a las necesidades particulares de cada empresa.

Softtek ofrece a sus clientes en el sector industrial la garantía de contar con un socio experto en tecnología de información y con alianzas que aportan profundidad en la especialización tanto tecnológica como de los procesos básicos del sector.

La integración de especialistas, tecnologías y aplicaciones nos permiten estar a la vanguardia en la asistencia de principio a fin en los proyectos de transformación estratégica de las empresas.

1.2 Organización

Es la parte fundamental de la empresa, ya que a partir de ésta se sientan la bases para enfocar los esfuerzos corporativos hacia los diferentes mercados de trabajo y oportunidades comerciales.

1.2.1 Mundo Softtek

El mundo Softtek ha creado mercados en varias partes del globo geográfico, para brindar sus servicios en el desarrollo, integración y consultoría de sistemas.

- Argentina
- Brasil
- Colombia
- España
- Estados Unidos
- México
- Perú
- Puerto Rico
- Venezuela

La operatividad es similar en todas las sedes, ya que más de un área colaboran entre sí para obtener los mejores resultados en los planes y proyectos que la empresa pone en marcha, ya que en cada una se dedica a integrar las partes correspondientes para reunir los mejores elementos en la obtención de los objetivos propuestos para un proyecto.

El organigrama general del mundo Softtek en todas sus sedes se mestra en la figura 1.1.

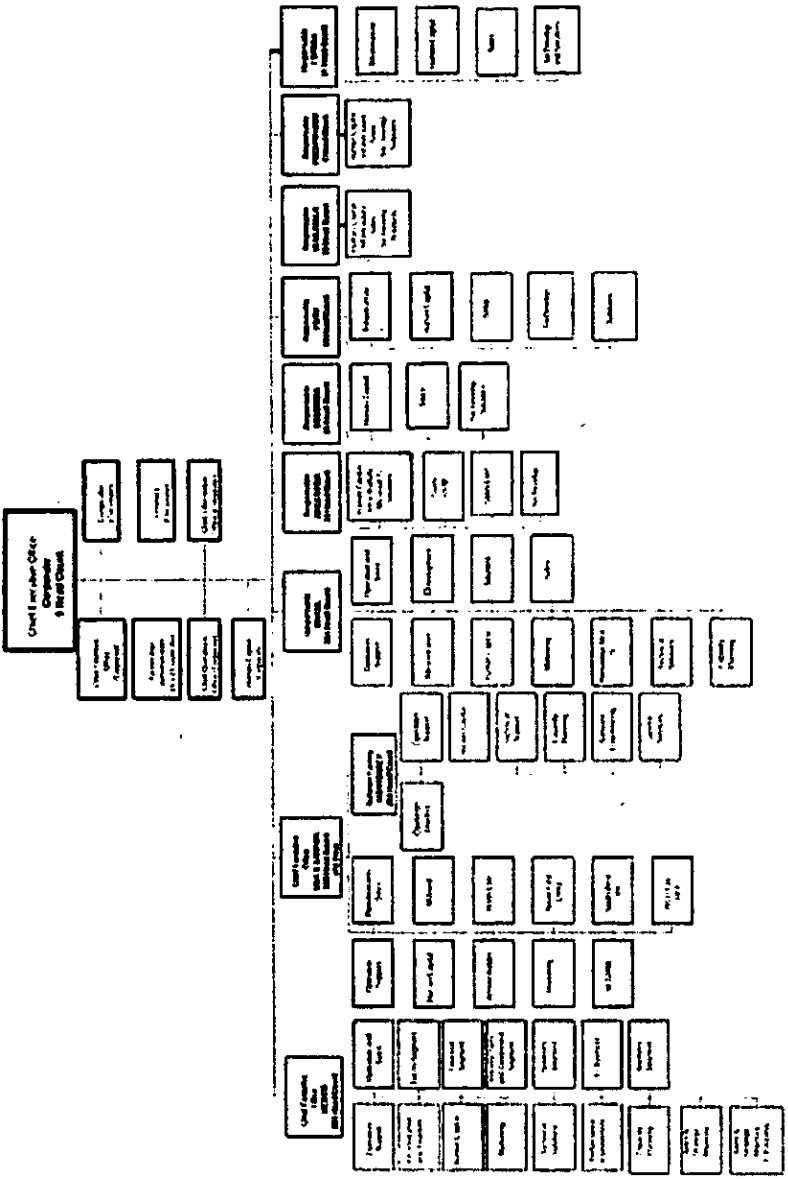


Fig. 1.1 Organigrama del Mundo Softtek

1.2.2. Softtek México

Dentro de la geografía México se tiene dos sedes, una en la ciudad de Monterrey donde se encuentran físicamente las fábricas de software y la sede de la ciudad de México.

La organización en general se divide en dos bloques:

Operación y Ventas

Son las áreas que se dedican a los procesos comerciales y ventas de sistemas, identifican los campos de acción y oportunidades de negocio para Softtek ante la demanda actual de sistemas para cubrir las necesidades y demandas de los clientes.

Soporte a la operación

Son áreas de apoyo funcional y estratégico a las áreas de Operación y Ventas, para ayudarles a consolidar sus acciones y facilitar los procesos de operación, para una una mejor comprensión se muetra la figura 1.2.

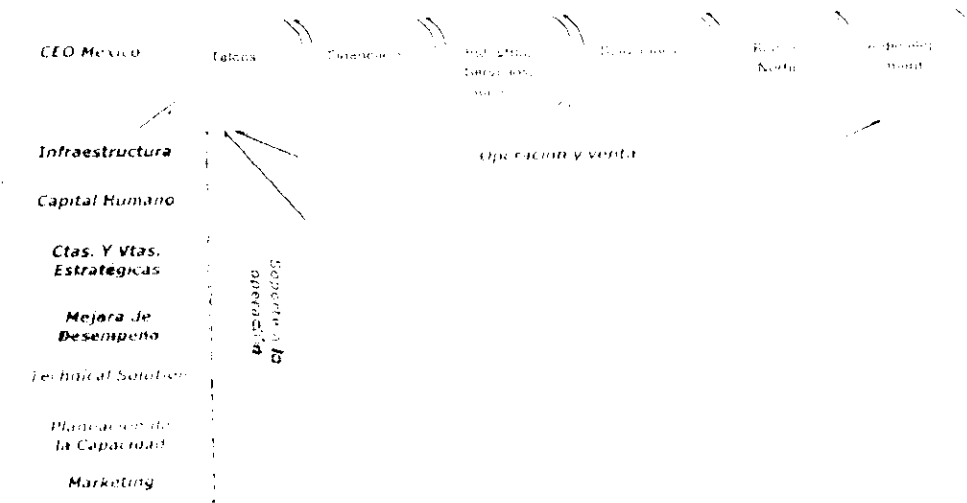


Fig 1.2 Organización Softtek México

1.2.3. Área de Soporte Técnico

El área de soporte técnico forma parte del área de Mejora de Desempeño, que se encarga de evaluar los procesos operativos y funcionales de los sistemas de información, ingeniería de software e ingeniería de información. Forma parte de está área debido a que el soporte técnico tiene una relación directamente proporcional a la operatividad y buen funcionamiento de los equipos de cómputo para el desarrollo de las tareas propias de la empresa.

1.2.3.1. Objetivo del área

Administrar los equipos de cómputo y periféricos, brindando soporte en hardware en las instalaciones de Softtek Nuevo León y proyectos.

1.2.3.2. Funciones y Servicios de área de soporte técnico

El área de soporte técnico comprende diversas funciones, así como varios servicios abarcado la geografía de ciudad de México, tanto en las instalaciones internas de la propia empresa, como a las personas y equipos de Softtek que se encuentran en las diversas instalaciones de los clientes donde Softtek tiene algún proyecto.

Administración de equipos de cómputo y periféricos

La administración de equipo de cómputo es una parte fundamental del área, ya que administra todos los equipos de cómputo y periféricos que son propiedad de Softtek en la ciudad de México, por lo que tiene a su cargo:

Asignaciones de equipo.

Consiste en controlar y asegurar la asignación y desasignación de equipos a los Proyectos y los usuarios de Softtek México, las actividades principales son:

- Definir la política de asignación y desasignación de equipo
- Controlar la asignación y el uso de los equipos asignados

Inventario

Controla y administra los equipos dentro de las instalaciones de Softtek México y proyectos, que consiste en:

- Realizar el inventario de equipos en Softtek México
- Definir la estrategia de actualización y sustitución del equipo existente en Softtek México
- Controlar la asignación y actualizar el uso y ubicación de los equipos asignados

Soporte técnico telefónico

Recibe, atiende y soluciona las peticiones de soporte telefónico recibidas por los usuarios de Softtek México, en resumen:

- Recepción de llamadas de usuarios de Softtek México
- Soporte telefónico para la solución de problemas
- Canalización al personal de soporte o al área correspondiente para la solución de problemas mayores en Softtek México
- Elaboración de estadísticas de atención a usuarios, tiempos de respuesta, análisis de fallas más comunes, etc.

Soporte técnico en sitio

Provee soporte técnico en sitio a los usuarios de Softtek México así como la definición de las políticas de soporte para usuarios en proyectos

- Soporte en sitio del usuario
- Corrección de errores o fallos en sitio
- Comunicar pasos a seguir en caso de requerir un soporte correctivo en la ubicación del área de soporte técnico.

Reinstalaciones, mantenimientos preventivos y correctivos a equipo de cómputo y periféricos

Consiste en mantener en óptimas condiciones los equipos de cómputo, tanto a nivel hardware como en software, con la finalidad de prolongar el tiempo de vida de los equipos.

Preventivos

Se programan mantenimientos preventivos a equipo de cómputo y dispositivos periféricos de acuerdo a su ubicación y a las fechas correspondientes, dentro de las instalaciones de Softtek ciudad de México y equipos propiedad de Softtek en las instalaciones de los clientes.

Correctivos

Corrige problemas en equipos y Software de usuarios en Softtek México, consiste en:

- Traslado de los equipos al área de soporte técnico para la corrección del problema
- Identificar garantías y soporte de equipos adquiridos para canalizar al área de compras en caso de ser requerido
- Determinar tiempo de corrección del fallo
- Registrar en bitácora las actividades realizadas

Reinstalaciones

Se reinstala y configura software dañado en equipo de usuarios de Softtek México, donde se realiza:

- Identifican daños o fallos en el software
- Define si la corrección se hace en el lugar del usuario o se lleva el equipo al departamento de soporte técnico
- Reinstala y configurar el software dañado
- Registrar en bitácora la actividad realizada

Administración red interna

Nodos de red

Instalación y monitoreo del funcionamiento de los nodos de trabajo dentro de las instalaciones de Softtek México, además de:

- Atender requerimientos de instalación de nodos nuevos
- Asegurar que los nodos existentes funcionen correctamente
- Registrar las incidencias de fallo en los nodos instalados

Cableado

Mantener en operación el cableado existente dentro de las instalaciones de Softtek México e instalar cableado adicional con base a requerimientos de los usuarios y:

- Proveer soporte en caso de fallo del cableado dentro de las instalaciones de Softtek México y proyectos.
- Instalar cableado dentro de las instalaciones de Softtek México y proyectos.

Administración de hubs

Programación y control del mantenimiento y operación de los Hubs de Softtek México, que consiste en:

- Mantenimiento y operación de los Hubs en Softtek México
- Monitoreo y administración

Impresoras de red

Administrar y asegurar el correcto funcionamiento de las impresoras de red instaladas en Softtek México:

- Administrar de forma centralizada las impresiones en base a los usuarios
- Controlar la ubicación de las impresoras de piso en Softtek México
- Programar el mantenimiento de las impresoras en base al volumen de operaciones de cada una de ellas
- Proveer soporte a impresoras en caso de fallos
- Definir las políticas de uso de las impresoras de Softtek México
- Registrar las incidencias en cada una de las impresoras de Softtek México

Administración de cursos internos

La educación y asesoría a los usuarios sobre las aplicaciones, manejo adecuado de los equipos y políticas de seguridad sirven para poder aprovechar mejor las ventajas de las aplicaciones así como alargar el tiempo de vida útil de los equipos.

Configuración de equipos

Configurar los equipos de trabajo en base a los requerimientos recibidos por el área que solicita la capacitación, llevando a cabo las actividades de:

- Definir configuraciones necesarias en base a requerimientos de los cursos
- Probar y asegurar que las configuraciones son adecuadas
- Capacitar a usuarios de acuerdo a la configuración establecida en los equipos
- Administrar las configuraciones de equipo
- Planear cambios y modificaciones en la configuración de equipos

Instalación de equipos

Instalar y desinstalar el equipo de computo con base en requerimientos del área usuaria en Softtek México, que consiste en:

- Definir las tareas necesarias para la instalación y desinstalación de equipos y componentes en Softtek México
- Realizar las pruebas de funcionamiento una vez instalado el equipo de trabajo

Investigación de nuevas tecnologías

Investigar y sugerir adquisición de nuevas tecnologías que apoyen la operación de Softtek México, que consiste en:

- Investigar y probar nuevas tecnologías de Hardware útiles para Softtek México
- Investigar y probar nuevas tecnologías de Software en el mercado
- Comunicar y difundir las tendencias del mercado en materia de Software y Hardware

Administración de las licencias de software y software

Controlar y monitorear el software autorizado para su uso en los equipos de Softtek México, además de administrar el préstamo de software a todas las áreas administrativas, operativas y de desarrollo. Realizando las siguientes actividades:

- Comunicar el software autorizado para su uso en los equipos de software
- Auditar equipos para eliminar software no autorizado
- Generar reporte de incidencias y amonestación a usuarios de software no autorizado
- Emisión y difusión de políticas relacionadas a software autorizado
- Préstamo de software a usuarios.

Evaluación y soporte en la compra de equipos

Se brinda asesoría al área de compras en la decisión de adquisiciones de equipos de cómputo y periféricos.

1.3. Infraestructura actual

La infraestructura obtiene un puesto fundamental dentro de la organización ya que en una parte es la herramienta principal para las actividades de su personal y por otra es la base de las comunicaciones tanto internas como externas con el resto del mundo.

1.3.1. Hardware y Software

Son las herramientas y los medios de trabajo básicos dentro de la organización, por lo que cumplen un papel muy importante en el desarrollo diario de las actividades de la empresa.

Hardware

Actualmente Softtek cuenta con una gama muy diversa de equipos de cómputo dado a la diversidad de aplicaciones, desarrollos, ventas, etc. Actualmente se tienen registrados 471 equipos de cómputo, incluyendo PC de escritorio y servidores.

La distribución de equipos se realiza de acuerdo a las necesidades y actividades de los usuarios, basándose en la figura 1.3.

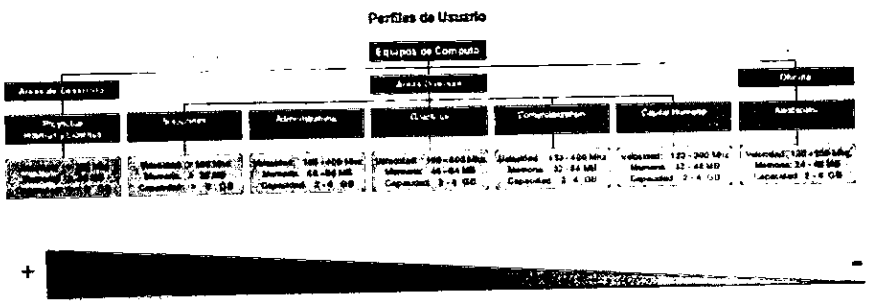


Fig. 1.3 Prioridades de asignación de equipo

El último inventario que se realizó, fue en el mes de enero del 2001, donde los resultados se muestran en la figura 1.4.

Fig. 1.4 Inventario de Softtek 2001

Donde las tecnologías utilizadas se distribuyen según la figura 1.5.

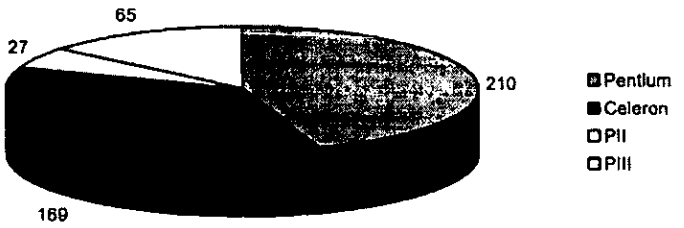


Fig. 1.5 Tecnologías de Hardware

Software

Debido a que Softtek en general se dedica al desarrollo y consultoría de sistemas, dentro de su inventario de software se encuentra una gran diversidad de programas de aplicaciones, desarrollo, diseño, bases de datos, herramientas de soporte etc.

Para no entrar en una larga lista de software, conviene sólo mencionar las alianzas y socios empresariales de Softtek, para dar un marco general de la variedad de software que tiene:

- Microsoft
- IBM
- OAO
- Epicor
- Eastman
- Onyx
- Kronos
- Compuware
- Lotus
- SAP
- Cognos

1.3.2. Análisis de conectividad

La conectividad hace aparición con la necesidad de intercambiar datos entre computadoras y de una mejor asignación de los recursos que se disponen, utilizando así los sistemas de cableado, los estándares para la transmisión de datos y sistemas de seguridad.

Conexión LAN – WAN de Softtek

En la figura 1.6 se muestra el esquema actual de conectividad entre la red LAN y la red WAN, la granja de servidores está integrada por los servidores de correo, intranet, servidores de aplicaciones y de desarrollo, mencionando sólo los más significativos.

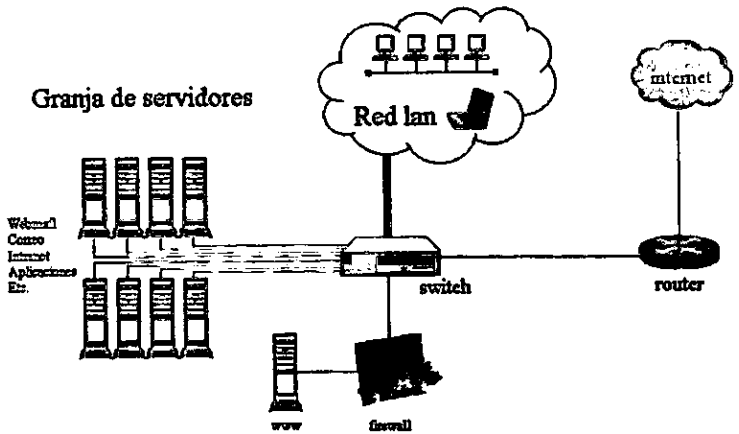


Fig. 1.6 Conectividad LAN-WAN

La figura 1.7 muestra la configuración actual de la red interna de Softtek México

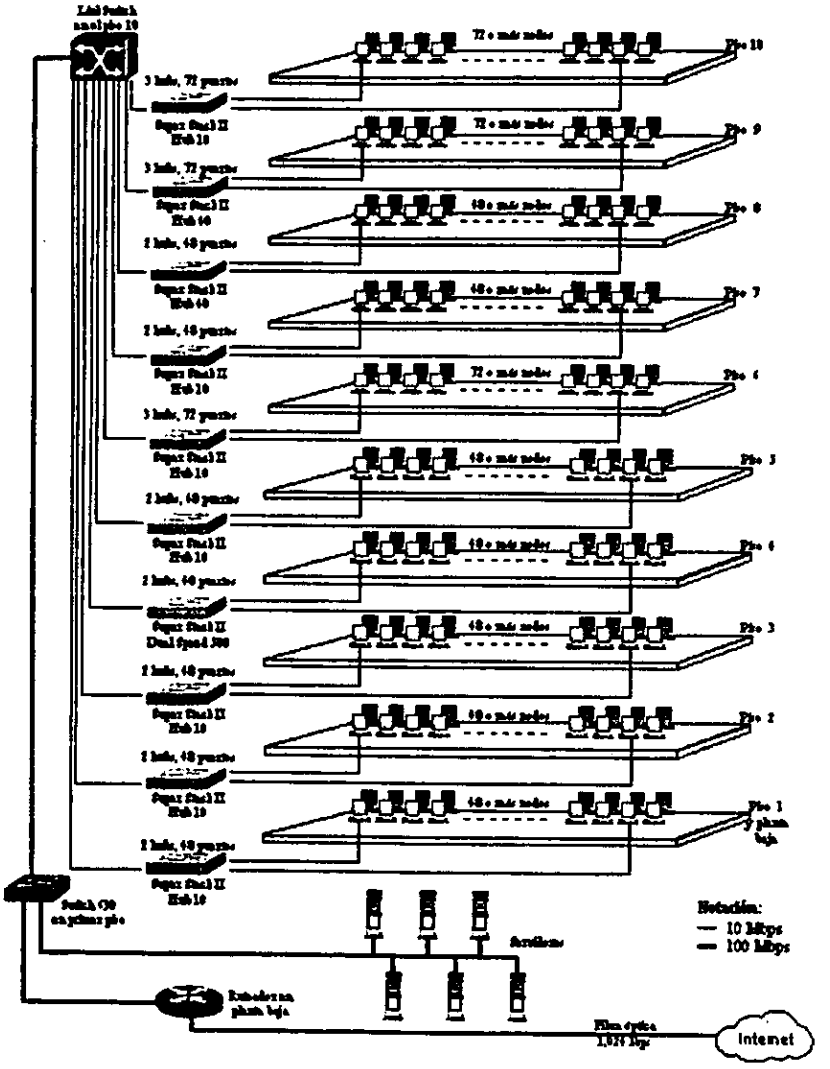


Fig. 1.7 Red LAN de Softtek

Actualmente se tiene una red LAN Ethernet distribuida en 10 pisos, aproximadamente con 60 nodos por piso, cada piso contiene 2 o 3 hub, los cuales se conectan un switch en el décimo piso, de ahí va un cable directo al switch del primer piso para de ahí salir al ruteador e Internet.

1.4. Proceso actual del manejo de inventario del área de soporte técnico

En forma esquemática en la figura 1.8, se describe el proceso actual de operación del área del manejo de inventario y asignaciones de equipos a los usuarios de Softtek México.

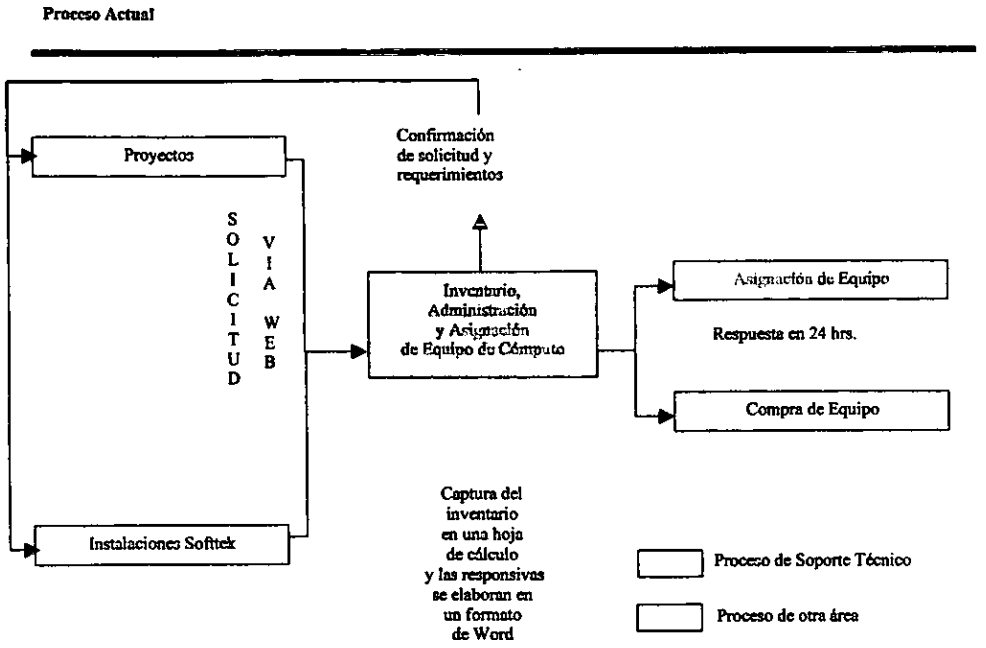


Fig. 1.8 Proceso de registro de Inventario

El usuario tiene que llenar una solicitud via web, que llena con los datos necesarios para su requerimiento de equipo, esta solicitud se tiene que imprimir. Posteriormente se tiene que firmar por el usuario y por el jefe de sector para su aprobación.

Una vez cumplido con esto, se entrega en el área de soporte técnico, donde se le asigna un número de folio, y se le da un tiempo de respuesta al usuario de 24 horas. Si el equipo se encuentra en almacen se prepara, se asigna y se registra en el inventario, en caso de que el equipo no se tenga disponible se manda una solicitud al área de compras para realizar la adquisición, una vez comprado se entrega a soporte técnico para etiquetarlo e inventariarlo.

Todas las compras de equipo de cómputo y asignaciones se registran en una hoja de Excel, donde se registran las características propias del equipo, como número de serie, marca, modelo y características específicas, número de factura proveedor y a que usuario es asignado, la ubicación donde se va a ubicar, proyecto o área y tiempo de asignación.

Una vez que el equipo lo deja de utilizar el usuario, se hacen los cambios en la hoja de cálculo y se pone el equipo como disponible.

El problema de este tipo de control de inventario es que se consume demasiado tiempo buscando los usuarios a los componentes en la hoja de calculo y hacer movimientos de los mismos, porque los equipos tienen demasiado movimiento en la empresa, por los proyectos o por descomposturas, cambios de equipos o de usarios, etc., por lo que el inventario no está actualizado y se tiene que hacer lavantamientos de inventario con una frecuencia de 5 o 6 meses, para hacer estas actualizaciones, ya que en ocaciones se hacen cambios de equipo y no se registran por la carga de trabajo en el área.

Capítulo 2

Marco Teórico

2. Marco Teórico

El actual capítulo tiene como propósito establecer los conceptos fundamentales requeridos en la elaboración del sistema de información, para lo cual su desglose se divide en cinco rubros, que son: Generalidades sobre Bases de Datos; Metodologías a utilizar; Teoría de Inventarios; Calidad en el Software y Teoría de Redes.

2.1. Generalidades sobre Bases de Datos

A lo largo de esta sección se van a desarrollar los siguientes conceptos: Sistemas de Bases de Datos; Funciones del DBMS; Lenguajes de Definición y Manipulación de Datos; Niveles de Representación; Construcción del Esquema Conceptual y Métodos de Acceso. Dichos conceptos forman el soporte de la teoría referente a bases de datos.

2.1.1. Conceptos Básicos

En esencia, un sistema de bases de datos no es más que un *sistema para archivar en computador*. La base de datos en sí puede considerarse como una especie de archivero electrónico; dicho de otra manera, es un lugar donde se almacena un conjunto de archivos de datos computarizados [Date, 1998]. Al usuario del sistema se le brindarán recursos para realizar diversas operaciones sobre estos archivos, incluidas entre otras las siguientes:

- agregar archivos nuevos (vacíos) a la base de datos;
- insertar datos nuevos en archivos ya existentes;
- obtener datos de archivos ya existentes;
- actualizar datos en archivos ya existentes;
- borrar datos en archivos ya existentes y
- eliminar archivos ya existentes (vacíos o no) de la base de datos.

2.1.1.1. Sistema de Bases de Datos

Un sistema de bases de datos es básicamente un sistema para archivar en computador; es decir, es un sistema computarizado cuyo propósito general es mantener información y hacer que esté disponible cuando se solicite. La información en cuestión puede ser cualquier cosa que se considere importante para el individuo o la organización a la cual debe servir el sistema; dicho de otro modo, cualquier cosa necesaria para apoyar el proceso general de atender los asuntos de ese individuo u organización. La figura 2.1 muestra una representación muy simplificada de un sistema de bases de datos.

El propósito de la figura 2.1 es ilustrar la forma como se integran los cuatro componentes principales de un sistema de bases de datos: la información, el equipo, los programas y los usuarios. Enseguida se analizan en forma concisa estos cuatro componentes.

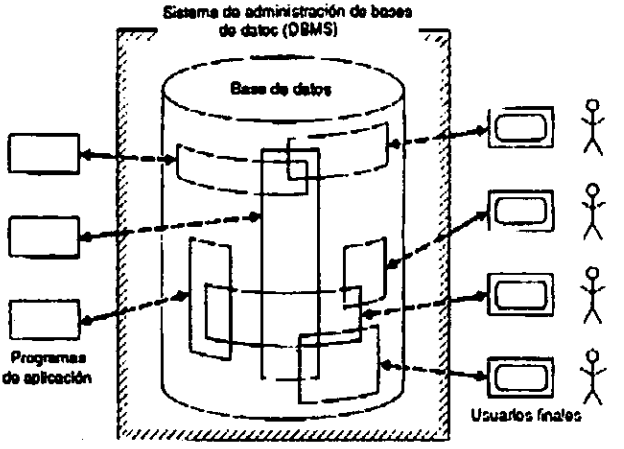


Fig. 2.1 Esquema simplificado de un sistema de bases de datos.

Información

En la actualidad existen sistemas de bases de datos para máquinas que van desde microcomputadores bastante pequeños (aun los portátiles) hasta los macrocomputadores más grandes. Cabe mencionar que las facilidades proporcionadas por un sistema dado dependen hasta cierto punto del tamaño y la capacidad de la máquina en la cual se trabaje. En particular, los sistemas en máquinas grandes ("sistemas grandes") casi siempre son *multiusuario*, mientras que los de máquinas más pequeñas ("sistemas pequeños") suelen ser de *un solo usuario*. En estos últimos, sólo un usuario puede tener acceso a la base de datos en un momento dado; en un sistema multiusuario, varios usuarios pueden tener acceso a la base de datos al mismo tiempo. Como lo sugiere la figura 2.1, se supondrá, por razones de generalidad, que el sistema es multiusuario, pero en realidad la distinción casi no tiene importancia en lo que toca a la mayoría de los usuarios: uno de los objetivos de casi todos los sistemas multiusuario es precisamente lograr que cada individuo pueda comportarse como si estuviera trabajando con un sistema de un solo usuario. Los problemas especiales de los sistemas multiusuario se refieren sobre todo a aspectos internos del sistema, no visibles para el usuario.

Nota: Casi siempre conviene, por sencillez, suponer que todos los datos almacenados en el sistema se mantienen en una sola base de datos, y aquí se hará en general esta suposición pues no afecta en forma apreciable el resto del análisis. No obstante, en la práctica pueden existir razones de peso, aun en sistemas pequeños, para repartir la información en varias bases de datos distintas.

En general, pues, la información en la base de datos -por lo menos en sistemas grandes- estará *integrada* y además será *compartida*. Estos dos aspectos, integración y compartimiento, constituyen una ventaja importante de los sistemas de bases de datos en ambientes "grandes"; y por lo menos la integración puede tener también relevancia en el ambiente "pequeño". Por supuesto, existen muchas otras ventajas, aun en los ambientes pequeños. Pero primero se explicará el significado de los términos "integrada" y "compartida".

- "Integrada" significa que la base de datos puede considerarse como una unificación de varios archivos de datos, por lo demás distintos, y que elimina del todo o en parte cualquier redundancia entre ellos [Date, 1998].

- "Compartida" significa que los elementos individuales de información en la base de datos pueden compartirse entre varios usuarios distintos, en el sentido de que todos ellos pueden tener acceso al mismo elemento de información (y diferentes usuarios pueden utilizarlo para propósitos diferentes). Como ya se indicó, es posible que distintos usuarios tengan acceso al mismo elemento de información al mismo tiempo ("acceso concurrente"). Esta capacidad de compartir (en forma simultánea o no) se desprende en parte de la integración de la base de datos [Date, 1998].

Otra consecuencia del mismo hecho (la integración de la base de datos) es que por lo regular un usuario determinado sólo se ocupará de un subconjunto de la base de datos total; es más, los subconjuntos de los distintos usuarios se superpondrán de diversas maneras. Dicho de otro modo, diferentes usuarios percibirán una base de datos determinada de varias maneras distintas. De hecho, aun cuando dos usuarios compartan el mismo subconjunto de la base de datos, la forma como vean ese subconjunto puede diferir de manera considerable en los detalles.

Equipo

Los componentes de equipo del sistema consisten en:

- Los volúmenes de almacenamiento secundario -por lo regular discos magnéticos de cabeza móvil- donde se conservan los datos almacenados, junto con los dispositivos de E/S asociados (unidades de disco, etcétera), controladores de dispositivos, canales de E/S, y demás.

- El procesador o procesadores y la memoria principal asociada que hacen posible la ejecución de los programas del sistema de bases de datos.

Programas

Entre la base de datos física misma (es decir, los datos tal y como están almacenados en realidad) y los usuarios del sistema existe un nivel de programas, el *manejador de base de datos* (manejador de BD) o, en la mayoría de los casos, el *sistema de administración de bases de datos* (DBMS, *database management system*). El DBMS maneja todas las solicitudes de acceso a la base de datos formuladas por los usuarios. Así, una de las funciones generales del DBMS es *distanciar a los usuarios de la base de datos de detalles al nivel del equipo* (de manera muy similar a la forma como los sistemas de lenguajes de programación evitan a los programadores de aplicaciones la necesidad de ocuparse de detalles al nivel de la máquina). En otras palabras, el DBMS presenta a los usuarios una vista de la base de datos en un nivel un tanto por encima del nivel del equipo, y hace posible sus operaciones expresadas en términos de esa vista de nivel más alto.

Nota: El DBMS es definitivamente el componente de software más importante de todo el sistema, pero no es el único. Entre los demás pueden mencionarse las utilerías, las herramientas para desarrollar aplicaciones, las ayudas para el diseño, los generadores de informes, etcétera.

Usuarios

Se toman en cuenta tres clases amplias de usuarios:

- En primer término, está el *programador de aplicaciones*, quien se encarga de escribir los programas de aplicación que utilizan la base de datos, casi siempre en lenguajes de alto nivel. Esos programas operan sobre los datos en todas las formas acostumbradas: recuperación de información ya existente, inserción de información nueva, eliminación o modificación de datos ya existentes. Por supuesto, todas estas funciones se llevan a cabo dirigiendo las solicitudes apropiadas al DBMS. Los programas en sí pueden ser aplicaciones por lote convencionales, o aplicaciones *en línea*, cuya función es servir a un usuario final que tiene acceso a la base de datos desde un terminal en línea. La mayor parte de las aplicaciones actuales son de este último tipo.

- La segunda clase de usuario es entonces el *usuario final*, quien interactúa con el sistema desde una terminal en línea. Un usuario final puede tener acceso a la base de datos a través de una de las aplicaciones en línea mencionadas anteriormente, o puede utilizar una interfaz incluida como parte integral de los programas del sistema de base de datos. Tales interfaces trabajan también mediante aplicaciones en línea, por supuesto, pero esas aplicaciones vienen integradas, y no las escriben los usuarios. Casi todos los sistemas incluyen por lo menos una aplicación integrada de este tipo, a saber, un *procesador de lenguaje de consulta* interactivo, mediante el cual el usuario puede formular mandatos o proposiciones de alto nivel (como SELECT, INSERT, etc.) al DBMS. El lenguaje SQL puede considerarse como un ejemplo representativo de los lenguajes de consulta de bases de datos.

Nota: La mayor parte de los sistemas incluyen también interfaces integradas adicionales con las que el usuario no necesita emitir mandatos explícitos como SELECT, sino que funcionan (por ejemplo) mediante la elección de opciones de un menú o el llenado de una forma. Estas interfaces *manejadas mediante* menús o *mediante formas* suelen ser más fáciles de usar en el caso de personas sin estudios formales de procesamiento de datos. En cambio, las interfaces *manejadas mediante mandatos* (como los lenguajes de consulta) tienden a requerir ciertos conocimientos sobre procesamiento de datos, aunque quizá no muy extensos (obviamente no tantos como se requieren para escribir un programa de aplicación en algún lenguaje de programación). Por otro lado, una interfaz manejada mediante mandatos suele ser más flexible que una manejada mediante formas o menús, debido a que los lenguajes de consulta casi siempre incluyen ciertas funciones que no se encuentran en esas otras interfaces.

- La tercera clase de usuario es el *administrador de la base de datos*, o DBA (*database administrator*).

2.1.1.2. Base de Datos

Datos persistentes

Conviene llamar "persistentes" a los datos de una base de datos (aunque quizá en realidad no persistan mucho tiempo). Esto tiene por objeto sugerir que la información de una base de datos difiere de otros tipos de datos, más efímeros, como son los datos de entrada y de salida, las proposiciones de control, las colas de trabajo, los bloques de control de programas, lo resultados intermedios y, en términos más generales, cualquier información cuya naturaleza sea hasta cierto punto transitoria. Enseguida se explican los términos "datos de entrada" y "datos de salida".

- "Datos de entrada" se refieren a la información que entra al sistema por primera vez (casi siempre desde el teclado de una terminal, desde un lector de tarjetas o desde un dispositivo similar). Esta información podría dar pie a una modificación de los datos persistentes (podría *convertirse* en parte de estos últimos), pero en principio no forma parte de la base de datos propiamente dicha.

- De manera similar, "datos de salida" se refieren a mensajes y resultados que emanan del sistema (casi siempre impresos o presentados en la pantalla de una terminal). Una vez más, esta información podría *derivarse* de los datos persistentes, pero no se le considera en sí como parte de la base de datos.

Por supuesto, la distinción entre datos persistentes y transitorios no es rígida y nítida, sino que depende hasta cierto punto del contexto (por ejemplo, de la utilización de los datos). No obstante, si suponemos que tal distinción tiene al menos cierto sentido intuitivo, es posible ahora presentar una definición un tanto más precisa del término "base de datos":

- Una *base de datos* está constituida por cierto conjunto de datos persistentes utilizado por los sistemas de aplicaciones de una empresa determinada.

El término "empresa" tal y como se emplea en esta definición no es más que un término genérico usado por comodidad, aplicable a cualquier organización comercial, científica, técnica o de otro tipo con un grado razonable de autosuficiencia. Una empresa podría ser una sola persona (con una pequeña base de datos privada) o una corporación o entidad similar de gran tamaño (con una enorme base de datos compartida), o cualquier cosa entre estos extremos.

Entidades y asociaciones

Consideremos con un poco más de detalle el caso de una compañía manufacturera. Una empresa de este tipo con toda seguridad deseará registrar información referente a los *proyectos* que está manejando; las partes utilizadas en esos proyectos; los *proveedores* que suministran esas partes; las bodegas donde se almacenan; los *empleados* asignados a los proyectos, etcétera. Los proyectos, partes, proveedores y demás constituyen así las *entidades* básicas acerca de las cuales la empresa necesita registrar información (en el campo de las bases de datos se utiliza ampliamente el término "entidad" para referirse a cualquier objeto distinguible que ha de representarse en la base de datos). Véase la figura 2.2.

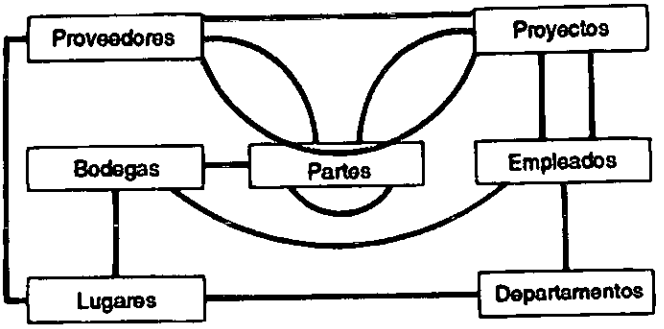


Figura 2.2 Un ejemplo de datos de operación

Es importante comprender que, además de las entidades básicas mismas, existirán también *asociaciones* que vinculen dichas entidades. Estas asociaciones se representan mediante líneas o arcos de conexión en la figura 2.2. Por ejemplo, existe una asociación entre los proveedores y las partes: Cada proveedor suministra ciertas partes, y a su vez cada parte es suministrada por ciertos proveedores (dicho en forma más precisa, cada proveedor suministra ciertas *clases* de partes, y cada *clase* de parte es suministrada por ciertos proveedores). De manera similar, las partes se utilizan en proyectos y, de modo recíproco, los proyectos emplean partes; las partes se almacenan en bodegas, y las bodegas almacenan partes; y así sucesivamente. Obsérvese que todas estas asociaciones son *bidireccionales*, es decir, se aplican en cualquiera de las dos direcciones. Por ejemplo, la asociación entre proveedores y partes puede servir para responder a cualquiera de las preguntas siguientes, o a ambas:

- Dado un proveedor, encontrar las partes correspondientes.
- Dada una parte, encontrar los proveedores correspondientes.

El aspecto importante de esta asociación, y de todas las demás asociaciones ilustradas en la figura 2.2, es que *forman parte de la información, tanto como las entidades básicas*. Por tanto, deben estar representadas en la base de datos.

1. Aunque la mayor parte de las asociaciones del diagrama implican *dos* tipos de entidad -es decir, son asociaciones *binarias*- no por fuerza todas las asociaciones deben ser binarias es este sentido. En el ejemplo hay una asociación que implica tres tipos de entidades (proveedores, partes y proyectos), es decir, una asociación *ternaria*. La interpretación manifiesta es que ciertos proveedores suministran ciertas partes a ciertos proyectos. Obsérvese con cuidado cómo esta asociación ternaria ("los proveedores suministran partes a los proyectos") *no* equivale en general a la combinación de las tres asociaciones binarias "los proveedores suministran partes", "se utilizan partes en los proyectos" y "los proyectos reciben suministros de los proveedores". Por ejemplo, la información

(a) Smith suministra llaves inglesas al proyecto Manhattan nos dice *más* que la combinación
(b) Smith suministra llaves inglesas,
(c) Se utilizan llaves inglesas en el proyecto Manhattan, y
(d) Smith es proveedor del proyecto Manhattan.

No podemos (válidamente) deducir (a) sabiendo sólo (b), (c) y (d). En forma más precisa, si sabemos (b), (c) y (d), entonces podríamos deducir que Smith suministra llaves inglesas a *algún* proyecto (digamos al proyecto Tz), que *algún* proveedor (digamos el proveedor Sx es Smith o que Py es llaves inglesas o que Tz es el proyecto Manhattan). Las inferencias falsas como éstas son ejemplos de lo que a veces se denomina *la trampa de conexión*.

2. El diagrama contiene también un arco que implica un solo tipo de entidad (partes). La asociación en este caso es que algunas partes incluyen a otras como componentes inmediatos (la llamada asociación de *desglose de materiales*). Por ejemplo, un tornillo es un componente de un juego de bisagras, el cual también se considera como una parte que podría a su vez ser componente de otra parte de un nivel superior, como una tapa. Cabe señalar que la asociación sigue siendo binaria, sólo que los dos tipos de entidad vinculados (es decir, partes y partes) resultan ser el mismo.

3. En general, un conjunto dado de tipos de entidad podría estar vinculado mediante cualquier cantidad de asociaciones distintas. En el diagrama, dos líneas conectan a proyectos y empleados. Una podría representar la asociación "asignado" (el empleado se asigna al proyecto) y la otra indicará la asociación "es gerente de" (el empleado es gerente del proyecto).

Obsérvese bien que una asociación puede considerarse como una entidad por sí misma. Si se acepta como definición de entidad "cualquier objeto acerca del cual deseamos registrar información", sin duda una asociación satisface la definición. Por ejemplo, "la parte P4 se almacena en la bodega B8" es una entidad acerca de la cual bien podríamos querer registrar información (por ejemplo, la cantidad correspondiente). Más aún, no hacer distinciones innecesarias entre entidades y asociaciones presenta claras ventajas. Por tanto, se tratarán las asociaciones como un tipo especial de entidad.

Propiedades

Como se acaba de señalar, se considera como entidad cualquier objeto acerca del cual deseamos registrar información. Dicho de otro modo, las entidades (y por ende también las asociaciones) tienen *propiedades*. Por ejemplo, los proveedores tienen *localidades;* las partes tienen *pesos;* los proyectos tienen *prioridades;* las tareas tienen *fechas de inicio;* y así sucesivamente. Por tanto, dichas propiedades deben estar representadas también en la base de datos. Por ejemplo, podría incluir un tipo de registro S, que representará al tipo de entidad "proveedor", y a su vez ese tipo de registro podría incluir un tipo de campo CIUDAD para representar la propiedad de "localidad".

2.1.2. Funciones del DBMS

El *sistema de administración de la base de datos* (DBMS) es por supuesto el conjunto de programas que maneja todo acceso a la base de datos. Conceptualmente, lo que sucede es lo siguiente:

1. Un usuario solicita acceso, empleando algún sublenguaje de datos determinado (por ejemplo, SQL).

2. El DBMS interpreta esa solicitud y la analiza.

3. El DBMS inspecciona en orden. El esquema externo de ese usuario, la correspondencia externa/conceptual asociada, el esquema conceptual, la correspondencia conceptual/interna y la definición de la estructura de almacenamiento.

4. El DBMS ejecuta las operaciones necesarias sobre la base de datos almacenada.

Como ejemplo, considérese lo que se necesita para extraer una cierta ocurrencia de registro externo. En general, se requerirán campos de varias ocurrencias de registro conceptual. Cada ocurrencia de registro conceptual, a su vez, puede requerir campos de varias ocurrencias de registro almacenado. Así, en teoría al menos, el DBMS debe obtener primero todas las ocurrencias de registro almacenado requeridas, construir enseguida las ocurrencias de registro conceptual necesarias, y después construir la ocurrencia de registro externo requerida. En cada etapa, quizá se requieran conversiones de tipos de datos u otras.

Nota: Desde luego, se ha simplificado la descripción precedente. En particular, hace pensar que todo el proceso es interpretativo, pues sugiere que los procesos de analizar la solicitud, inspeccionar los diversos esquemas, etcétera, se realizan todos en el momento de la ejecución. La interpretación, por su parte, casi siempre implica un desempeño pobre (debido al aumento en el tiempo de ejecución). En la práctica, puede ser posible *compilar* las solicitudes de acceso antes del momento de la ejecución.

Examinemos ahora las funciones del DBMS con un poco más de detalle. Dichas funciones incluirán por lo menos todas las siguientes.

- Definición de datos

El DBMS debe ser capaz de aceptar definiciones de datos (esquemas externos, el esquema conceptual, el esquema interno, y todas las correspondencias asociadas) en versión fuente y convertirlas en la versión objeto apropiada. Dicho de otro modo, el DBMS debe incluir componentes procesadores de lenguajes para cada uno de los diversos lenguajes de definición de datos (DDL). El DBMS también debe "entender" las definiciones en DDL, en el sentido en que, por ejemplo, "entiende" que los registros externos EMPLEADO contiene un campo SALARIO; y debe poder utilizar estos conocimientos para interpretar y responder las solicitudes de los usuarios (por ejemplo, una consulta de todos los empleados cuyo salario sea inferior a $50 000).

- Manipulación de datos

El DBMS debe ser capaz de atender las solicitudes del usuario para extraer, y quizá poner al día, datos que ya existen en la base de datos, o para agregar en ella datos nuevos. Dicho de otro modo, el DBMS debe incluir un componente procesador de lenguaje de manipulación de datos (DML).

En general, las solicitudes en DML pueden ser "planeadas" o "no planeadas":

- Una solicitud planeada es aquella cuya necesidad se previó mucho tiempo antes de que tuviera que ejecutarse por primera vez. El DBA habrá afinado con toda probabilidad el diseño físico de la base de datos a fin de garantizar un buen desempeño para estas solicitudes.

- Una solicitud no planeada, en cambio, es una consulta *ad hoc*, es decir, una solicitud cuya necesidad no se previó, sino que surgió de improviso. El diseño físico de la base de datos puede ser o no ideal para la solicitud específica de que se trate. En general, el logro del mejor desempeño posible con solicitudes no planeadas representa un reto considerable para el DBMS.

Las solicitudes planeadas son características de las aplicaciones "operacionales" o "de producción"; las no planeadas son representativas de las aplicaciones de "apoyo a decisiones". Es más, las solicitudes planeadas casi siempre se originan en programas de aplicación previamente escritos, en tanto que las solicitudes no planeadas, por definición, se emitirán de manera interactiva.

- Seguridad e integridad de los datos

El DBMS debe supervisar las solicitudes de los usuarios y rechazar los intentos de violar las medidas de seguridad e integridad definidas por el DBA.

- Recuperación y concurrencia de los datos

El DBMS -o en su defecto algún componente de software relacionado con él, al que por lo regular se denomina *administrador de transacciones*- debe cuidar del cumplimiento de ciertos controles de recuperación y concurrencia.

- Diccionario de datos

El DBMS debe incluir una función de *diccionario de datos*. Puede decirse que el diccionario de datos es una base de datos por derecho propio (pero una base de datos del sistema, no del usuario).

El contenido del diccionario puede considerarse como "datos acerca de los datos" (los cuales reciben en ocasiones el nombre de "metadatos"), es decir, *definiciones* de otros objetos en el sistema, y no sólo "datos en bruto". En particular, en el diccionario de datos se almacenarán físicamente todos los diversos esquemas y correspondencias (externos, conceptuales, etcétera) tanto en sus versiones fuente como en las versiones objeto.

Un diccionario completo incluirá también referencias cruzadas para indicar, por ejemplo, cuáles programas usan cuáles partes de la base de datos, cuáles usuarios requieren cuáles informes, qué terminales están conectadas al sistema, y cosas por el estilo. Es más, el diccionario podría (y quizá debería) estar integrado a la base de datos a la cual define, e incluir por tanto su propia definición. Deberá ser posible consultar el diccionario igual que cualquier otra base de datos de modo que se pueda saber, por ejemplo, cuáles programas o usuarios podrían verse afectados por alguna modificación propuesta para el sistema.

- Desempeño

Por último, huelga decir que el DBMS deberá ejecutar todas las funciones recién identificadas en la forma más eficiente posible

Como conclusión, una forma de resumir todo lo anterior -esto es, una forma de caracterizar la función general del DBMS- es decir que constituye la *interfaz* entre el *usuario* y el sistema de bases de datos.

La interfaz del usuario puede definirse como una frontera del sistema, más allá de la cual todo resulta invisible para el usuario. Por definición, entonces, la interfaz del usuario está en el nivel *externo*. Sin embargo, existen ciertas situaciones en las cuales la vista externa quizá no difiera en forma apreciable de la porción relevante de la vista conceptual subyacente.

2.1.3. Lenguajes de Definición y Manipulación de Datos

2.1.3.1. Definición de datos

Aquí nos ocuparemos de las proposiciones de SQL incluidas en el lenguaje de definición de datos (DDL). Limitaremos nuestra atención a las porciones "relacionales" del DDL; es decir, se analizarán sólo aquellos aspectos que interesan de manera directa al usuario, y no los que se relacionan exclusivamente con el nivel interno del sistema, aspectos que desde luego dependen en alto grado del sistema con el que se trabaje. Desde el punto de vista del usuario, las principales proposiciones de DDL son:

CREATE TABLE	CREATE INDEX
ALTER TABLE	
DROP TABLE	DROP INDEX

2.1.3.1.1. Tablas Base

Una tabla base es un caso especial (importante) del concepto más general de "tabla". Por tanto, se comenzará por hacer más preciso ese concepto general.

Definición

Una *tabla* en un sistema relacional se compone de una fila de *cabeceras de columna*, junto con cero o más filas de *valores de datos* (diferente número de filas de datos en diferentes momentos) [Date, 1998]. Para una tabla dada:

(a) La fila de cabecera de columna especifica una o más columnas (dando, entre otras cosas, un tipo de datos para cada una).

(b) Cada fila de datos tiene un solo valor escalar para cada una de las columnas especificadas en la fila de cabeceras de columna. Además, todos los valores de una columna dada, tienen el mismo tipo de datos, a saber, el tipo especificado en la fila de cabeceras de columna para esa columna.

Dos cuestiones surgen en relación con la definición anterior.

1.- Adviértase que no se menciona ningún *ordenamiento de filas*. En términos estrictos, se considera que las filas de una tabla relacional están desordenadas. (Las filas de una relación constituyen un *conjunto* matemático, y en matemáticas los conjuntos no están ordenados de manera alguna.) Es posible, *imponer* un orden en esas filas cuando se obtienen en respuesta a una consulta, pero tal ordenamiento se debe considerar sólo como algo conveniente para el usuario; no es un aspecto intrínseco del concepto de tabla en sí.

2.- En contraste con el primer punto, las columnas de una tabla *sí* se consideran ordenadas, de izquierda a derecha. (Al menos, esto se hace en SQL. Pero en realidad, éste es un punto en el cual SQL difiere de la teoría relacional.) Por ejemplo, en la tabla de proveedores S (véase la figura 2.3), la columna S# es la primera columna, la columna SNOMBRE es la segunda, etcétera. No obstante, en la práctica son pocas las situaciones en las cuales tiene importancia ese ordenamiento de izquierda a derecha, e incluso ésas pueden evitarse con un poco de disciplina. Es recomendable evitarlas.

Digresión: Por supuesto, las filas y columnas sí tienen un ordenamiento físico en la versión de la tabla almacenada en el disco y, es más, tal ordenamiento físico puede tener, y tiene, un efecto muy definido sobre el desempeño del sistema. Pero lo importante es que (en casi todas las situaciones, y en el caso ideal en todas las situaciones) esos ordenamientos físicos son *transparentes para el usuario*. Fin de la digresión.

S	SN	SNOMBRE	SITUACIÓN	CIUDAD
	S1	Salazar	20	Londres
	S2	Jaimes	10	París
	S3	Bernal	30	París
	S4	Corona	20	Londres
	S5	Aldana	30	Atenas

P	PN	PNOMBRE	COLOR	PESO	CIUDAD
	P1	Tuerca	Rojo	12	Londres
	P2	Perno	Verde	17	París
	P3	Birlo	Azul	17	Roma
	P4	Birlo	Rojo	14	Londres
	P5	Leva	Azul	12	París
	P6	Engrane	Rojo	19	Londres

SP	SN	PN	CANT
	S1	P1	300
	S1	P2	200
	S1	P3	400
	S1	P4	200
	S1	P5	100
	S1	P6	100
	S2	P1	300
	S2	P2	400
	S3	P2	200
	S4	P2	200
	S4	P4	300
	S4	P5	400

Fig. 2.3 La base de datos de proveedores y partes

Pasemos ahora al caso específico de las tablas base. Una tabla base es una tabla *autónoma con nombre*. Al decir "autónoma", queremos decir que la tabla existe por derecho propio, a diferencia (por ejemplo) de una vista, la cual no existe por sí misma sino que se deriva de una o más tablas base (es sólo otra forma de ver esas tablas base). Al decir "con nombre" queremos decir que a la tabla se le asigna de manera específica un nombre mediante una proposición CREATE (crear) apropiada, a diferencia (por ejemplo) de una tabla construida como resultado de una consulta, la cual carece de un nombre explícito propio y posee sólo existencia efímera.

CREATE TABLE

Ya se está en condiciones de analizar la proposición CREATE TABLE (crear tabla). El formato general de esta proposición es el siguiente:

CREATE TABLE tabla-base
(definición-de-columna [, definición-de-columna] ...
[, definición-de-clave-primaria]
[, definición-de-clave-ajena [, definición-de-clave-ajena] ...]) ;

donde una "definición-de-columna", a su vez, tiene la forma:

columna tipo-de-datos [NOT NULL]

Nota: A menos que se indique de manera explícita lo contrario, los corchetes se utilizarán en todas las definiciones sintácticas para indicar que lo encerrado por ellos es opcional (es decir, puede omitirse). Los puntos suspensivos (...) significan que la unidad sintáctica inmediata anterior puede repetirse de manera opcional una o más veces. Lo que ve en mayúsculas debe escribirse tal como se muestra; lo que va en minúsculas debe reemplazarse con valores específicos elegidos por el usuario.

El efecto de esta proposición es crear una tabla base nueva vacía.

Tipos de datos

En forma general se manejan los siguientes tipos de datos escalares.

- *Datos numéricos*

INTEGER	entero binario de palabra completa
SMALLINT	entero binario de media palabra
DECIMAL(*p,q*)	número decimal empacado, *p* dígitos y signo, con *q* dígitos a la derecha del punto decimal
FLOAT(*p*)	número de punto flotante, con precisión de *p* dígitos binarios

- *Datos de cadena*

CHARACTER(*n*)	cadena de longitud fija con exactamente *n* caracteres de 8 bits
VARCHAR(*n*)	cadena de longitud variable con hasta *n* caracteres de 8 bits
GRAPHIC(*n*)	cadena de longitud fija con exactamente *n* caracteres de 16 bits
VARGRAPHIC(*n*)	cadena de longitud variable con hasta *n* caracteres de 16 bits

- *Datos de fecha/hora*

DATE	fecha (aaaammdd)
TIME	hora (hhmmss)
TIMESTAMP	"marca de tiempo" (combinación de fecha y hora, con una precisión de microsegundos)

También se manejan varias abreviaturas o ortografías distintas (por ejemplo, CHAR en vez de CHARACTER, DEC en vez de DECIMAL, INT en vez de INTEGER).

Información faltante

En el mundo real surge con frecuencia el problema de la información faltante. Así pues, es conveniente contar con una forma de manejar situaciones de este tipo en los sistemas formales de base de datos.

La forma normal de representar información faltante en sistemas SQL es mediante indicadores especiales llamados *nulos (nulls)*.

Cualquier campo puede contener nulos a menos que la definición de ese campo especifique NOT NULL (no nulo) de manera explícita . Si un campo dado puede contener nulos, y se inserta un registro en esa tabla sin dar un valor para ese campo, se insertará en forma automática un nulo en esa posición. Si un campo dado no puede contener nulos, se rechazará cualquier intento de introducir un nulo en ese campo.

Digresión: Una columna capaz de aceptar nulos se representa físicamente en la base de datos almacenada mediante dos columnas. La columna de datos propiamente dicha y una columna indicadora oculta, con ancho de un carácter, la cual se almacena como prefijo de la columna de datos en sí. Si la columna indicadora contiene unos binarios, se hará caso omiso del valor correspondiente en la columna de datos (es decir, se tomará como nulo); si la columna indicadora contiene ceros binarios, el valor correspondiente en la columna de datos se tomará como válido. Pero desde luego la columna indicadora será siempre "transparente para el usuario". *Fin de la digresión.*

ALTER TABLE

Así como es posible crear una tabla base nueva en cualquier momento con CREATE TABLE, también se puede *alterar* una tabla base ya existente agregando una columna nueva a la derecha, mediante la proposición ALTER TABLE (alterar tabla):

```
ALTER  TABLE  tabla-base  ADD  columna  tipo-de-datos  ;
```

Pero desde el punto de vista del usuario es como si todos los registros se *hubieran* ampliado físicamente cuando se ejecutó ALTER TABLE. El usuario no tiene manera de detectar la diferencia.

Nota: En algunos sistemas son posibles otros tipos de alteraciones de las tablas. Por ejemplo, en algunos sistemas es posible desechar una columna de una tabla existente o cambiar el tipo de datos de una columna (digamos de SMALLINT a INTEGER).

DROP TABLE

También es posible eliminar en cualquier momento una tabla base existente:

```
DROP  TABLE  tabla-base ;
```

La tabla base especificada se elimina del sistema (mejor dicho, se elimina del catálogo la descripción de esa tabla). También se desechan en forma automática todos los índices y vistas definidos en términos de esa tabla base.

2.1.3.1.2. Índices

Los índices, como las tablas base, se crean y desechan mediante proposiciones de definición de datos de SQL. Sin embargo, CREATE INDEX (crear índice) y DROP INDEX (desechar índice) son las únicas proposiciones del lenguaje SQL que hacen referencia a los índices. La decisión de utilizar o no un índice determinado para responder a una cierta consulta en SQL no la toma el usuario, sino el DBMS.

El formato general de CREATE INDEX es:

```
CREATE  [UNIQUE]  INDEX  índice
      ON  tabla-base  ( columna  [orden] [ , columna  [ orden ] ]  ...)
                  [ CLUSTER ]  ;
```

Cada especificación de "orden" es ASC (ascendente) o bien DESC (descendente); si no se especifica una de las dos, se toma ASC por omisión. Además, la especificación opcional CLUSTER indica que se trata de un índice de agrupamiento; una tabla base puede tener como máximo un índice de agrupamiento.

Si se especifica la opción UNIQUE (único) en CREATE INDEX, no se permitirá que dos registros de la tabla base indexada tengan al mismo tiempo el mismo valor en el campo o combinación de campos de indización.

Los índices, al igual que las tablas base, se pueden crear y desechar en cualquier momento. Obsérvese, empero, que fracasará cualquier intento de crear un índice único para una tabla no vacía en la cual se viole ya la restricción de unicidad.

Se puede crear cualquier cantidad de índices para una misma tabla base.

La proposición para desechar un índice es:

```
DROP  INDEX  índice ;
```

El índice se destruye (es decir, su descripción se elimina del catálogo). Si un plan de aplicación ya existente depende de ese índice desechado, dicho plan será religado de manera automática la próxima vez que se invoque.

2.1.3.2. Manipulación de datos

Ahora se abordarán los aspectos de manipulación de datos de SQL. Este lenguaje ofrece cuatro proposiciones de DML -SELECT (seleccionar), UPDATE (actualizar), DELETE (eliminar) e INSERT (insertar)- y a continuación se describen las características principales de los cuatro. Como ya se dijo, las tablas manipuladas con esas proposiciones de DML pueden ser, en términos generales, tanto tablas base como vistas, pero nos ocuparemos de manera exclusiva de las tablas base.

2.1.3.2.1 Consultas

Con el siguiente ejemplo, se ejemplifica la consulta "obtener el número y la situación de todos los proveedores de París", la cual puede expresarse en SQL así:

```
SELECT  S#,  SITUACIÓN
FROM    S
WHERE  CIUDAD  =  "París" ;
```

Resultado:

S#	SITUACIÓN
S2	10
S3	30

Este ejemplo ilustra la forma más común de la proposición SELECT de SQL: "*SELECT* (seleccionar) los campos específicos *FROM* (de) la tabla especificada *WHERE* (donde) se cumpla la condición especificada".

2.1.3.2.2 Operaciones de Actualización

El lenguaje de manipulación de datos de SQL incluye tres operaciones de actualización: UPDATE (actualizar en el sentido de alterar o modificar), DELETE (eliminar) e INSERT (insertar).

UPDATE (ACTUALIZAR)

Formato general:

```
UPDATE   tabla
SET      campo = expresión-escalar
       [ ,  campo = expresión-escalar  ] ...
[ WHERE  condición ]  ;
```

Todos los registros de "tabla" que satisfagan "condición" serán modificados de acuerdo con las asignaciones ("campo = expresión-escalar") de la cláusula SET (establecer).

DELETE (ELIMINAR)

Formato general:

```
DELETE
FROM   tabla
[ WHERE  condición ]  ;
```

Se eliminarán todos los registros de "tabla" que satisfagan "condición".

INSERT (INSERTAR)

Formato general:

```
INSERT
INTO      tabla   [ ( campo [ , campo ] ... ) ]
VALUES   ( literal [ , literal ] ... ) ;
```

o bien

```
INSERT
INTO      tabla   [ ( campo [ , campo ] ... ) ]
subconsulta ;
```

En el primer formato se inserta en "tabla" una fila con los valores especificados en los campos especificados (el *i*ésimo literal en la lista de literales corresponde al *i*ésimo campo en la lista de campos). En el segundo formato, se evalúa la subconsulta y se inserta una copia del resultado (casi siempre varias filas) en "tabla"; la *i*ésima columna de ese resultado corresponde al *i*ésimo campo en la lista de campos. En ambos casos, omitir la lista de campos equivale a especificar una lista con todos los campos de la tabla, en el orden de izquierda a derecha.

2.1.4. Niveles de Representación

En la arquitectura ANSI/SPARC se divide en tres *niveles*, denominados niveles interno, conceptual y externo (véase la figura 2.4). En términos generales:

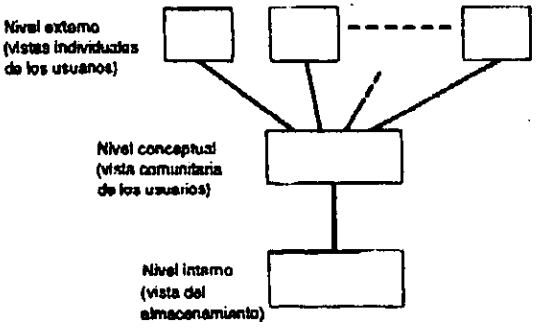


Fig. 2.4 Los tres niveles de la arquitectura

(a) El nivel *interno* es el más cercano al almacenamiento físico, es decir, es el que se ocupa de la forma como se almacenan físicamente los datos [Date, 1998].
(b) El nivel *externo* es el más cercano a los usuarios, es decir, es el que se ocupa de la forma como los usuarios individuales perciben los datos [Date, 1998].
(c) El nivel *conceptual* es un "nivel de mediación" entre los otros dos [Date, 1998].

Si el nivel externo se ocupa de las vistas *individuales* de los usuarios, puede considerarse que el nivel conceptual se ocupa de una vista *comunitaria* de los usuarios. Dicho de otro modo, existirán muchas "vistas externas" distintas, cada una formada por una representación más o menos abstracta de alguna parte de la base de datos total, y existirá sólo una "vista conceptual" formada por una representación igualmente abstracta de la base de datos en su totalidad. (Recuérdese que a la mayoría de los usuarios no les interesará toda la base de datos, sino sólo una porción limitada de ella.) De manera similar, habrá sólo una "vista interna", la cual representará a toda la base de datos tal como está almacenada físicamente.

2.1.4.1. El Nivel Externo

El nivel externo es el del usuario individual. El DBA es un caso especial importante. (Pero, a diferencia de los usuarios ordinarios, el DBA deberá interesarse también en los niveles conceptual e interno.)

Cada usuario dispone de un *lenguaje*:

- En el caso del programador de aplicaciones, dicho lenguaje será o bien uno de los lenguajes de programación convencionales, o bien un lenguaje propio ("de cuarta generación") específico para el sistema en cuestión.

- Para el usuario final será o bien un lenguaje de consulta, o algún lenguaje de aplicación especial, quizá manejado mediante formas o menús, adaptado a los requerimientos de ese usuario y apoyado por algún programa de aplicación en línea.

En lo que toca a este análisis, el aspecto importante de todos esos lenguajes es que deben incluir un *sublenguaje de datos*. Se dice que el sublenguaje de datos (abreviado DSL, *data sublanguage*, en la figura 2.5) está *embebido* (*o inmerso*) dentro del *lenguaje anfitrión* correspondiente.

En principio, cualquier sublenguaje de datos es en realidad una combinación de por lo menos dos lenguajes subordinados: un *lenguaje de definición de datos* (DDL, *data definition language*), con el cual es posible definir o declarar los objetos de la base de datos, y un *lenguaje de manipulación de datos* (DML, *data manipulation language*) con el que es posible manipular o procesar dichos objetos.

El término ANSI/SPARC para la vista individual de un usuario es *vista externa*. Así, una vista externa es el contenido de la base de datos tal como lo percibe algún usuario determinado.

En general, una vista externa se compone de varias ocurrencias de varios tipos de *registros externos*. Un registro externo no es por fuerza idéntico a un registro almacenado. El sublenguaje de datos del usuario se define en términos de registros externos.

Toda vista externa se define mediante un *esquema externo*, que consiste básicamente en definiciones de cada uno de los diversos tipos de registros externos en esa vista externa. El esquema externo se escribe con la porción DDL del sublenguaje de datos del usuario.

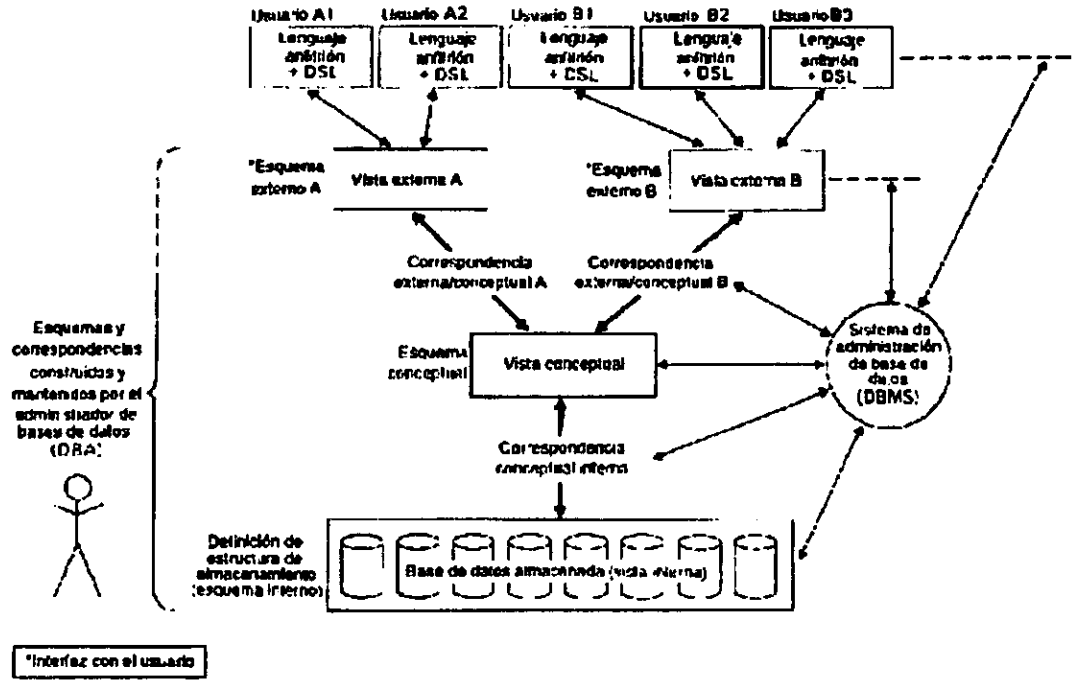


Fig. 2.5 Arquitectura detallada del sistema

2.1.4.2. El Nivel Conceptual

La vista conceptual es una representación de toda la información contenida en la base de datos, también (como en el caso de una vista externa) en una forma un tanto abstracta si se compara con el almacenamiento físico de los datos. Además, puede ser muy diferente de la forma como percibe los datos cualquier usuario individual. A grandes rasgos, la vista conceptual debe ser un panorama de los datos "tal como son", y no como por fuerza los perciben los usuarios debido a las limitantes del lenguaje o el equipo específicos utilizados.

La vista conceptual se compone de varias ocurrencias de varios tipos de *registro conceptual*. Un registro conceptual no es por necesidad idéntico a un registro externo, por un lado, ni a un registro almacenado, por el otro.

Digresión: Cabe señalar que bien podría haber otras formas de representar los datos en el nivel conceptual, es decir, sin utilizar en absoluto registros como tales; precisamente por esa razón, dichas formas podrían ser preferibles en cierto modo. Por ejemplo, en vez de manejar "registros conceptuales", podría ser preferible considerar entidades, y quizá también asociaciones, en alguna forma más directa.

La vista conceptual se define mediante un *esquema conceptual*, el cual incluye definiciones de cada uno de los tipos de registro conceptual. El esquema conceptual se escribe utilizando otro lenguaje de definición de datos, el *DDL conceptual*. Si ha de lograrse la independencia de los datos, esas definiciones en DDL conceptual no deberán implicar consideraciones de estructura de almacenamiento o de técnica de acceso. Deben ser *sólo* definiciones de contenido de información. Si el esquema conceptual se hace en verdad independiente de los datos de esta manera, entonces los esquemas externos, definidos en términos del esquema conceptual, serán por fuerza también independientes de los datos.

Así, la vista conceptual es una vista del contenido total de la base de datos, y el esquema conceptual es una definición de esa vista. Es de esperar que las definiciones en el esquema conceptual incluyan muchas características más, como son las verificaciones de seguridad y de integridad. En casi todos los sistemas existentes el "esquema conceptual" no es mucho más que una simple unión de todos los esquemas externos individuales, con la posible adición de algunas verificaciones sencillas de integridad y seguridad.

2.1.4.3. El Nivel Interno

El tercer nivel de la arquitectura es el nivel interno. La *vista interna* es una representación de bajo nivel de toda la base de datos; se compone de varias ocurrencias de varios tipos de *registro interno*. Este último término es el que utiliza ANSI/SPARC para referirse a la construcción que se ha estado llamando registro *almacenado*. La vista interna, por tanto, todavía está a un paso del nivel físico, ya que no maneja registros *físicos* (llamados también *páginas* o *bloques*), ni otras consideraciones específicas de los dispositivos como son los tamaños de cilindros o de pistas.

La vista interna se define mediante el *esquema interno*, el cual no sólo define los diversos tipo de registros almacenados sino también especifica cuáles índices hay, cómo se representan los campos almacenados, en qué secuencia física se encuentran los registros almacenados y más detalles. El esquema interno se escribe con otro lenguaje más de definición de datos, el DDL interno.

2.1.5. Construcción del Esquema Conceptual

Dentro del esquema conceptual, se desarrollan los siguientes temas que a continuación se mencionan: Modelado semántico; El modelo Entidad/Asociación; Dependencia funcional; Normalización. Dichos rubros comprenden la construcción del esquema conceptual en bases de datos.

2.1.5.1. Modelado Semántico

Los sistemas de bases de datos en general sólo tienen en realidad una comprensión limitada del *significado* de la información contenida en la base de datos. El concepto de dominio sirve para ilustrar el aspecto importante de que los modelos de datos existentes no carecen por completo de aspectos semánticos.

El modelado semántico se conoce también con otros nombres. Entre estos términos se mencionan el modelado de *datos*, el modelado de *entidades/asociaciones*, y el modelado de *entidades*.

Es posible caracterizar el problema del modelado semántico en términos de los siguientes cuatro pasos.

1.- Primero, se intenta identificar un conjunto de conceptos semánticos que parecen ser útiles al hablar informalmente del mundo real:

- Por ejemplo, podríamos convenir en que el mundo está formado de *entidades*, el concepto de entidad sí parece ser útil para hablar del mundo real, como una abstracción del objeto al que representan.
- Cada entidad posee una propiedad que sirve para *identificarla*.
- Cualquier entidad puede relacionarse con otras entidades mediante *asociaciones*.

2.- A continuación, se tratará de inventar un conjunto de objetos *simbólicos* que puedan emplearse para representar los conceptos semánticos anteriores.

3.- También se desarrollarán un conjunto de *reglas de integridad* correspondientes a esos objetos.

4.- Por último, se desarrollarán un conjunto de operadores para manipular esos objetos.

En la figura 2.6 se muestran unos cuantos de esos conceptos (entidad, propiedad, asociaciones, subtipo) con fines de ilustración.

Concepto	Definición informal	Ejemplos
ENTIDAD	Un objeto distinguible	Proveedor, parte, envió, empleado, departamento, persona Composición, concierto, orquesta, conductor, orden de compra, partida conductor, orden de compra, partida
PROPIEDAD	Un elemento de información que describe a una entidad	Número de proveedor, cantidad de envío, departamento de empleado, altura de persona, tipo de concierto, fecha de orden de compra
ASOCIACIÓN	Una entidad que sirve para conectar entre sí a otras dos o más entidades	Envío (proveedor-parte) Asignación (empleado-departamento) Grabación (composición-orquesta-conductor)
SUBTIPO	El tipo de entidad Y es un subtipo del tipo de entidad X si y sólo si todo Y es por fuerza un X	Empleado es un subtipo de persona Concierto es un subtipo de composición

Fig. 2.6 Algunos conceptos semánticos útiles

2.1.5.2. El Modelo Entidades/Asociaciones

Pasamos ahora a la aplicación de las ideas del modelado semántico al problema de diseñar bases de datos. Uno de los enfoques más conocidos -ciertamente uno de los más usados- es el llamado enfoque de *entidades /asociaciones* (E/R, *entity/relationship*), el cual se construye por medio de diagramas llamados "diagramas de entidad/relación".

Diagrama de Entidad - Relación

El diagrama de Entidad - Relación (también conocido como DER, o diagrama E - R) es un modelo de red que describe con un alto nivel de abstracción los datos almacenados en un sistema [Mylls R., 1994].

Los diagramas de Entidad - Relación se utilizan para representar el modelo conceptual de objetos del mundo real y las asociaciones entre ellos; definen la información que el sistema debe crear, mantener, procesar y eliminar, así como las asociaciones que deben ser soportadas por la base de datos; agrupa los elementos de información en entidades (que son abstracciones del mundo real) y refleja los hechos, eventos del mundo real que hemos representado en nuestro modelo de datos.

Componentes de un diagrama Entidad - Relación:

1. Entidades: se representan con un rectángulo, poseen un nombre que los identifica de los demás y tienen uno o más atributos característicos de él mismo que hacen que tenga un papel necesario en el sistema. Las entidades son como "cosas que se pueden identificar claramente. A continuación se clasifican como *entidades regulares* y *entidades débiles*. Una entidad débil es aquella cuya existencia depende de otra entidad, en el sentido de que no puede existir si no existe también esa otra entidad. Una entidad regular es una entidad que no es débil.

2. Propiedad: las entidades (y las asociaciones, como puede verse) tienen propiedades (también conocidas como *atributos*; evitamos este término porque ya tiene un significado específico en el modelo relacional). Todas las entidades de un tipo determinado tienen ciertas propiedades en común. Cada tipo de propiedad toma sus valores de un *conjunto de valores* correspondiente (o sea, dominio, en términos relacionales). Además, las propiedades pueden ser:

   -*Simples o compuestas* (por ejemplo, la propiedad compuesta "nombre del empleado" podría estar formada por las propiedades simples "nombre de pila", "apellido paterno" y "apellido materno")

   -*Clave* (es decir, únicas, quizá dentro de algún contexto; por ejemplo, el nombre de un dependiente podría ser único sólo dentro del contexto de un número de empleado dado)

   -*Univaluadas o multivaluadas* (es decir, se permiten grupos repetitivos)

   -*Faltantes* (o sea, "desconocidas" o "no aplicables")

   -*Base o derivadas* (por ejemplo, "cantidad total" de una parte dada podría derivarse sumando las cantidades base de los envíos individuales de esa parte)

3. Asociaciones: Las entidades se conectan entre sí mediante asociaciones y se representan por medio de un rombo y un nombre que define la forma de la asociación. La asociación es una vinculación entre entidades. Por ejemplo, existe una asociación (DEPTO_EMP) entre departamentos y empleados, la cual representa el hecho de que un cierto departamento ocupa a un conjunto dado de empleados. Como en el caso de las entidades, es necesario distinguir entre los *tipos* de asociaciones y los *casos* de las asociaciones.

Se dice que las entidades implicadas en una asociación dada son los *participantes* de esa asociación. El número de participantes en una asociación se conoce como el *grado* de esa asociación. (Adviértase, pues, que este término *no* significa lo mismo que "grado" en el modelo relacional.)

Sea *R* un tipo de asociación en la cual participa el tipo de entidad *E*. Si cada caso de *E* participa en por lo menos un caso de *R*, se dice que la participación de *E* en *R* es total; en caso contrario se dice que es *parcial*. Por ejemplo, si todo empleado debe pertenecer a un departamento, la participación de empleados en DEPTO_EMP es total; pero si un empleado puede no pertenecer a departamento alguno, la participación de empleados en DEPTO_EMP será parcial.

Adviértase que una asociación E/R puede ser de *uno a uno*, *de uno a muchos* (conocida también como *de muchos a uno*) o de *muchos a muchos*. (Suponemos por sencillez que todas las asociaciones son binarias, es decir, de grado dos; desde luego, la extensión de los conceptos y la terminología a asociaciones de grado mayor que dos es en esencia directa.

4. Subtipo: toda entidad pertenece por lo menos a un tipo entidad; pero una entidad puede ser de varios tipos al mismo tiempo. Por ejemplo, si algunos empleados son programadores (y todos los programadores son empleados), podríamos decir que PROGRAMADOR es un *subtipo del supertipo* EMPLEADO. Todas las propiedades de los empleados se aplican de manera automática a los programadores, pero lo contrario no cumple (por ejemplo, los programadores podrían tener una propiedad "lenguaje principal de programación" que no se aplica a los empleados en general). De manera similar, los programadores participan de manera automática en todas las asociaciones en las cuales participan los empleados, pero lo contrario no es cierto (por ejemplo, los programadores podrían pertenecer a alguna sociedad profesional de programación, sin que esto suceda con los empleados en general).

   Adviértase además que algunos programadores podrían ser programadores de aplicaciones, y otros podrían ser programadores de sistemas; por lo tanto, podría decirse que PROGRAMADOR DE APLICACIONES y PROGRAMADOR DE SISTEMAS son subtipos del supertipo PROGRAMADOR (y así sucesivamente). En otras palabras, un subtipo de entidad sigue siendo un *tipo* de entidad, y por tanto puede tener sus propios subtipos. Un tipo de entidad y sus subtipos inmediatos, los subtipos inmediatos de estos últimos, etcétera, constituyen todos juntos la *jerarquía de tipos* del tipo de entidad en cuestión.

   Nota: Las jerarquías de tipos se conocen con varios nombres distintos, entre ellos:

   - *Jerarquías de generalización* (por la razón de que, por ejemplo, un empleado es una generalización de un programador)
   - *Jerarquías de especialización* (por la razón de que, por ejemplo, un programador es una especialización de un empleado)
   - *Jerarquías ESUN* (ISA) (por la razón de que, por ejemplo, todo programador "es un" empleado (en inglés, *"is a"*))

2.1.5.3. Dependencia Funcional

Concepto:

- Dada una relación *R*, el atributo *Y* de *R* *depende funcionalmente* del atributo *X* de *R* -en símbolos,

  $R.X \longrightarrow R.Y$

  (léase "*R.X determina funcionalmente a R.Y*")- si y sólo si un solo valor *Y* en *R* está asociado a cada valor *X* en *R* (en cualquier momento dado). Los atributos *X* y *Y* pueden ser compuestos [Date, 1998].

En la base de datos de proveedores y partes, por ejemplo, los atributos SNOMBRE, SITUACIÓN y CIUDAD de la asociación S son todos funcionalmente dependientes del atributo S# de la asociación S, porque, dado un valor específico de S.S#, existe sólo un valor correspondiente de S.SNOMBRE, de S.SITUACIÓN y de S.CIUDAD.

En símbolos,

S.S# ⟶ S.SNOMBRE
S.S# ⟶ S.SITUACIÓN
S.S# ⟶ S.CIUDAD

o, en forma más concisa,

S.S# ⟶ S.(SNOMBRE, SITUACIÓN, CIUDAD)

He aquí un ejemplo opuesto:

S.COLOR ↛ P.PESO

(en la asociación P no sucede que para cada color haya un solo peso; por ejemplo, P1 es roja y tiene un peso de 12, P6 también es roja pero tiene un peso de 19).

Adviértase que si el atributo *X* es una clave candidata de la asociación *R* -en particular, si es la clave *primaria*- entonces todos los atributos *Y* de la asociación *R* deben por fuerza depender funcionalmente de *X* (esto se desprende de la definición de clave candidata). Los ejemplos anteriores con la relación S ilustran este punto.

De manera similar, en las relaciones P y SP tenemos:

P.P# ⟶ P.PNOMBRE
P.P# ⟶ P.COLOR
P.P# ⟶ P.PESO
P.P# ⟶ P.CIUDAD
SP. (S#, P#) ⟶ SP.CANT

Adviértase, empero, que la definición de dependencia funcional (en adelante abreviada DF) no requiere que *X* sea una clave candidata de *R*; en otras palabras, no es obligatorio que un valor de *X* dado aparezca en una sola tupla de *R*.

Se presenta otra definición de DF en la cual se expresa esto de manera más explícita:

- Dada una asociación *R*, el atributo *Y* de *R* depende funcionalmente del atributo *X* de *R* si y sólo si, siempre que dos tuplas de *R* concuerden en su valor de *X*, deben por fuerza concordar en su valor de *Y*.

Por ejemplo, la asociación SP' de la figura 2.7 satisface la DF

SP'.S# ⟶ SP'.CIUDAD

SP'	S#	CIUDAD	P#	CANT
	S1	Londres	P1	300
	S1	Londres	P2	200
	S1	Londres	P3	400
	S1	Londres	P4	200
	S1	Londres	P5	100
	S1	Londres	P6	100
	S2	París	P1	300
	S2	París	P2	400
	.	.	.	.
	.	.	.	.

Fig. 2.7 Tabulación parcial de la relación SP'

(todas las tuplas de SP' con un valor de S# dado deben tener el mismo valor de CIUDAD), pero el atributo S# no es una clave candidata de la relación SP'.

También se definió el concepto de dependencia funcional *completa*. Se dice que el atributo Y de la asociación R es por completo dependiente funcionalmente del atributo X de la asociación R si depende funcionalmente de X y no depende funcionalmente de ningún subconjunto propio de X (es decir, no existe un subconjunto propio Z de los atributos componentes de X tales que Y sea funcionalmente dependiente de Z. Por ejemplo, en la asociación SP', no hay duda de que el atributo CIUDAD depende funcionalmente del atributo compuesto (S#, P#)):

SP'.(S#, P#) ⟶ SP'.CIUDAD

Sin embargo, no es una DF *completa*, porque desde luego se tiene la DF:

SP'.S# ⟶ SP'.CIUDAD

(CIUDAD también depende funcionalmente de S# solo). Obsérvese que si Y depende funcionalmente de X, pero o por completo, X debe ser compuesto. De aquí en adelante se supondrá que la "dependencia funcional" se refiere a una dependencia funcional completa, a menos que se diga de manera explícita lo contrario. También se abreviará "dependencia funcional completa" a "dependencia funcional" cuando no exista riesgo de ambigüedad.

Conviene representar las dependencias funcionales en una asociación dada mediante un *diagrama de dependencias funcionales* (diagrama DF). Los diagramas DF para las asociaciones S, P y SP se representan en la figura 2.8.

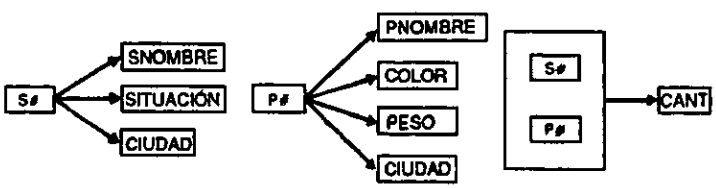


Fig. 2.8 Dependencia funcional en las asociaciones S, P, SP

Digresión: Como puede verse, cada flecha que se ve en los diagramas de la figura 2.8 es una flecha que sale de una clave candidata (de hecho la clave primaria) de la asociación pertinente. Por definición, siempre saldrán flechas de una clave candidata (porque, para cada valor de esa clave candidata, siempre hay un valor de todo lo demás); esas flechas jamás se pueden eliminar.

Cuando surgen problemas es cuando existen otras flechas. Así, el procedimiento de normalización puede caracterizarse, en términos muy informales, como un procedimiento para eliminar flechas que no salgan de claves candidatas.

1.- En primer lugar, debe entenderse que la dependencia funcional es un concepto *semántico*. Reconocer las dependencias funcionales es parte del proceso de entender que *significan* los datos. El hecho de que CIUDAD dependa funcionalmente de S#, por ejemplo, significa que cada proveedor está situado en una sola ciudad. Viéndolo de otra manera:

- Existe una restricción en el mundo real representado por la base de datos; a saber, que cada proveedor está situado en una sola ciudad.
- Como es parte de la semántica de la situación, esa restricción debe cumplirse de alguna manera en la base de datos.
- La forma de garantizar su cumplimiento es especificándolo en el esquema, para que el DBMS pueda hacer que se cumpla.
- La forma de especificarlo en el esquema es declarar la DF.

Los conceptos de normalización conducen a una forma muy sencilla de declarar tales dependencias (no por fuerza todas las dependencias funcionales, pero quizá las más importantes en la práctica).

2.- Como consecuencia del punto anterior: puesto que una dependencia funcional es de hecho un tipo especial de restricción de integridad, deberá ser posible expresarla en la sintaxis del lenguaje general de integridad. He aquí, por ejemplo, una formulación en "lenguaje general de integridad" de la restricción según la cual hay una dependencia funcional de S.S# a S.CIUDAD:

```
CREATE INTEGRITY RULE SCDF
       CHECK FORALL SX FORALL SY
             ( IF SX.S# = SY.S# THEN SX.CIUDAD = SY.CIUDAD ) ;
```

La formulación

S.S# ⟶ S.CIUDAD

puede considerarse como una abreviatura de esa proposición más compleja.

3.- Adviértase que las dependencias funcionales representan *asociaciones de muchos a uno*. La DF anterior, por ejemplo, puede leerse como "muchos proveedores están situados en la misma ciudad" (pero un proveedor está situado en una sola ciudad, por supuesto).

2.1.5.4. Normalización

La teoría de la normalización tiene como fundamento el concepto de *formas normales*. Se dice que una relación está en una determinada forma normal si satisface un cierto conjunto de restricciones. Por ejemplo, se dice que una relación está en primera forma normal (abreviada 1NF) si y sólo si satisface la restricción de que sus dominios simples subyacentes contengan sólo valores atómicos.

Se ha definido un gran número de formas normales. Originalmente, Codd definió la primera, segunda y tercera formas normales (1NF, 2NF, 3NF). En pocas palabras, todas las relaciones normalizadas están en 1NF; algunas relaciones 1NF están también en 2NF; y algunas relaciones 2NF están también en 3NF. El motivo de las definiciones de Codd era que 2NF era "más deseable" que 1NF, y 3NF a su vez era más deseable que 2NF. Es decir, el diseñador de una base de datos en general debe tratar de lograr un diseño con relaciones en 3NF, no relaciones que estén sólo en 2NF o 1NF.

También se introdujo la idea de un procedimiento, el llamado *procedimiento de normalización* (en términos más precisos, procedimientos de normalización *adicional*), con el cual una relación en cierta forma normal, digamos 2NF, se puede convertir en un conjunto de relaciones en una forma más deseable, digamos 3NF. Es posible caracterizar ese procedimiento como *la reducción sucesiva de un conjunto dado de relaciones a una forma más deseable*. Cabe señalar que el procedimiento es reversible; es decir, siempre es posible tomar la salida del procedimiento (el conjunto de relaciones 3NF, digamos) y convertirlas otra vez en la entrada (la relación 2NF original, digamos). Desde luego, esa reversibilidad es importante, porque significa que *no se pierde información* durante el proceso de normalización adicional.

Hoy día la nueva 3NF se conoce por lo regular como "forma normal Boyce/Codd" (BCNF).

También se mencionará lo que es la 4NF, la cual surge con el propósito de manejar las dependencias multivaluadas, y además se puede hacer extensivo el concepto de que una relación en 4NF cumple con las restricciones de la 3NF, y también es reversible.

2.1.5.4.1. Primera, Segunda y Tercera Formas Normales

- Una relación está en 3NF si y sólo si los atributos no clave (si los hay) son:

(a) mutuamente independientes, y
(b) dependientes por completo de la clave primaria.

Se explicarán los términos "atributos no clave" y "mutuamente independientes" de la siguiente manera.

- Un *atributo no clave* es cualquier atributo que no participa en la clave primaria de la relación en cuestión. Nota: Por sencillez se supondrá que cada relación tiene sólo una clave candidata (es decir, una clave primaria y ninguna clave alternativa). Esta suposición se refleja en las definiciones, las cuales no son muy rigurosas.

- Dos o más atributos son *mutuamente independientes* si ninguno de ellos depende funcionalmente de cualquier combinación de los otros. Tal independencia implica que cualquiera de esos atributos puede actualizarse sin tomar en cuenta a los demás.

La definición anterior de 3NF se puede interpretar, de manera aún más intuitiva, así:

- Una relación está en tercera forma normal (3NF) si y sólo si, en todo momento, cada tupla está formada por un valor de clave primaria que identifica a alguna entidad, junto con un conjunto de cero o más valores de atributos independientes entre sí, los cuales describen de alguna manera a esa entidad.

A continuación se pasará al proceso de reducción. Primero se dará una definición de la primera forma normal.

- Una relación está en *primera forma normal* (1NF) si y sólo si todos los dominios simples subyacentes contienen sólo valores atómicos.

Esta definición sólo declara que cualquier relación normalizada está en 1NF, lo cual desde luego es correcto. Una relación que sólo está en primera forma normal (o sea, una relación en 1NF que no está también en 2NF y por tanto tampoco en 3NF) tiene una estructura indeseable.

Las redundancias en una relación sólo en primera forma normal provocan diversos casos de lo que suele llamarse "anomalías de actualización"; es decir, problemas con las tres operaciones de actualización INSERT (insertar), DELETE (eliminar) y UPDATE (actualizar).

Ahora se presentará una definición de la segunda forma normal.

- Una relación está en *segunda forma normal* (2NF) si y sólo si está en 1NF y todos los atributos no clave dependen por completo de la clave primaria.

El proceso de reducción consiste en sustituir la relación en 1NF por *proyecciones* apropiadas; el conjunto de proyecciones así obtenido es equivalente a la relación original, en el sentido de que la relación original siempre se podrá recuperar efectuando la *reunión* (natural) de esas proyecciones, de manera que no se pierde información con la reducción (cosa desde luego muy importante). En otras palabras, el proceso es reversible.

1.- El proceso de reducción es, precisamente, un proceso de sacar proyecciones; es decir, el *operador de descomposición* es la proyección. De manera similar, el operador de recomposición es la reunión natural.

2.- El primer paso en el procedimiento de normalización es sacar proyecciones para eliminar las dependencias funcionales no completas. Dicho de otro modo, dada la siguiente relación *R*

```
R ( A, B, C, D )
  PRIMARY KEY ( A )
  R.A ⟶ R.D
```

la disciplina de normalización recomienda sustituir *R* por sus dos proyecciones *R1* y *R2*, así:

```
R1 ( A, D )
  PRIMARY KEY ( A )
R2 ( A, B, C )
  PRIMARY KEY ( A, B )
  FOREIGN KEY ( A ) REFERENCES R1
```

La relación *R* puede recuperarse si se efectúa la reunión de clave ajena a primaria de *R2* y *R1*.

Esta sustitución de R por R1 y R2 es una muestra de lo que se conoce como *descomposición sin pérdidas*. Como no se pierde información en esa descomposición, cualquier información derivable de la estructura original podrá derivarse también de la nueva estructura. Pero lo contrario no se cumple: la nueva estructura puede contener información que no podría representarse en la original.

En este sentido, la nueva estructura puede considerarse como una representación un poco más fiel del mundo real.

Ahora se da una definición de la tercera forma normal.

- Una relación está en *tercera forma normal* (3NF) si y sólo si está en 2NF y todos los atributos no clave dependen de manera no transitiva de la clave primaria.

Digresión: De esta definición se desprende un punto: "falta de dependencias transitivas" implica falta de dependencias *mutuas*.

El segundo paso en el procedimiento de normalización es sacar proyecciones para eliminar dependencias transitivas. Dicho de otro modo, dada una relación *R* del tipo

```
R ( A, B, C )
  PRIMARY KEY ( A )
  R.B ⟶ R.C
```

la disciplina de normalización recomienda sustituir *R* por sus dos proyecciones *R1* y *R2*, así;

```
R1 ( B, C )
    PRIMARY KEY ( B )
R2 ( A, B )
    PRIMARY KEY ( A )
    FOREIGN KEY ( B ) REFERENCES R1
```

La relación *R* puede recuperarse si se efectúa la reunión de clave ajena a primaria de *R2* y *R1*.

El nivel de normalización de una relación dada consiste en la semántica, no sólo es cuestión de los valores de los datos que aparecen en una relación en un momento específico. No es posible observar la tabulación de una relación dada en un instante dado y decir, sin más información, si esa relación está en 3NF (digamos); es necesario además conocer el significado de los datos, es decir, las dependencias, antes de emitir un juicio semejante.

En particular, el DBMS no puede garantizar el mantenimiento de una relación en 3NF (o en cualquier otra forma normal aparte de la primera) si no se le indican todas las dependencias pertinentes. Pero en el caso de una relación en 3NF, todo lo que se necesita para informar al DBMS de esas dependencias es indicarle cuál o cuáles atributos constituyen la clave primaria (porque las únicas dependencias serán aquellas que son consecuencia de la clave primaria, "sólo salen flechas de la clave primaria"). Si una relación no está en 3NF, serán necesarias especificaciones adicionales.

Nota: Aun dadas las dependencias, nunca es posible demostrar, a partir de una cierta tabulación, que una relación está en 3NF (aunque podría ser posible demostrar que no lo está). Lo mejor que puede hacerse es demostrar que la tabulación dada no viola ninguna de las dependencias; suponiendo que no lo haga, la tabulación será *consistente con la hipótesis* de que la relación está en 3NF, pero desde luego ese hecho no garantiza la validez de la hipótesis. Y para demostrar que una tabulación dada no viola ninguna de las dependencias requeridas por 3NF basta demostrar que no contiene valores repetidos de la clave primaria, lo cual no es un requerimiento muy exigente. De hecho, en un sistema que maneja claves primarias y en el cual es por tanto obligatorio que los valores de clave primaria sean únicos, *toda tabulación posible* será "consistente con la hipótesis" de que la relación en cuestión está en 3NF.

2.1.5.4.2. Cuarta Forma Normal

Una relación *R* está en cuarta forma normal (4NF) si y sólo si, siempre que existe una dependencia multivaluada en *R*, digamos $A \rightarrow\!\!> B$, todos los atributos de R dependen también *funcionalmente* de *A*. En otras palabras, las únicas dependencias (funcionales o multivaluadas) en *R* son de la forma $K \rightarrow X$ (es decir, una dependencia funcional con respecto a una clave candidata *K* de algún otro atributo *X*). O lo que es equivalente: *R* está en 4NF si está en *BCNF* y todas las dependencias multivaluadas en *R* son de hecho dependencias funcionales.

2.1.6. Métodos de Acceso

Localizar un elemento de información específico en la base de datos y presentarlo al usuario requiere varios niveles de programas para acceso de datos. Los detalles de esos niveles varían en forma apreciable en los distintos sistemas, y lo mismo sucede con la terminología, pero los principios son bastante generales y pueden explicarse a grandes rasgos de la siguiente manera (consúltese la figura 2.9).

1. En primer término, el DBMS decide cuál registro almacenado se necesita, y pide al manejador de archivos que extraiga ese registro. (Se supondrá para los propósitos de esta sencilla explicación que el DBMS es capaz de localizar por adelantado el registro exacto deseado. En la práctica puede ser necesario extraer un conjunto de varios registros y examinarlos dentro de la memoria principal para encontrar el que se busca. En principio, empero, esto sólo significa que la secuencia de pasos 1 a 3 deberá repetirse para cada uno de los registros almacenados de ese conjunto.)

2. A su vez, el manejador de archivos decide cuál página contiene el registro deseado, y pide al manejador de disco que lea esa página. La página es la unidad de E/S, es decir, la cantidad de datos transferidos entre el disco y la memoria principal en un solo acceso a disco.

3. Por último, el manejador de disco determina la localización física de la página deseada en el disco, y realiza la operación de E/S necesaria. *Nota*: Desde luego, habrá ocasiones en que la página requerida esté ya en un área de almacenamiento temporal en la memoria principal como resultado de una lectura anterior, en cuyo caso resulta obvio que no será necesario leerla otra vez.

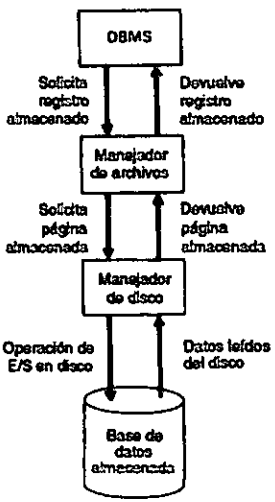


Fig. 2.9 El DBMS, el manejador de archivos y el manejador de disco

Manejador de disco

El manejador de disco es un componente del sistema operativo subyacente. Es el encargado de todas las operaciones físicas de E/S (en algunos sistemas se le llama componente de "servicios básicos de E/S"). Como tal, es evidente que necesita conocer las *direcciones físicas en el disco*. Por ejemplo, cuando el manejador de archivos solicita la lectura de una página específica *p*, el manejador de disco necesita saber con exactitud dónde está situada esa página en el disco físico. Por otro lado, el usuario del manejador de disco -es decir, el manejador de archivos- *no* necesita conocer esa información. Para el manejador de archivos, el disco es tan sólo una colección lógica de *conjuntos de páginas*, cada uno de los cuales se compone de un grupo de páginas de tamaño fijo.

Cada conjunto de páginas se identifica mediante un *número de página* que es único dentro del disco; los diferentes conjuntos de páginas no se traslapan (es decir, no tienen páginas en común). El manejador de disco entiende y mantiene la correspondencia entre números de página y direcciones físicas en el disco. La ventaja principal de este arreglo (que no es el único) es que todas las instrucciones codificadas específicas para un dispositivo pueden aislarse dentro de un solo componente del sistema, el manejador de disco, y todos los componentes de nivel más alto (sobre todo el manejador de archivos) pueden ser así independientes del dispositivo.

Entre las operaciones que puede realizar el manejador de disco con los conjuntos de páginas (es decir, las operaciones que puede solicitar el manejador de archivos) están las siguientes:

- Leer la página *p* del conjunto de páginas *c*;
- Reemplazar la página *p* dentro del conjunto de páginas *c*;
- Añadir una página nueva al conjunto de páginas *c*;
- Eliminar la página *p* del conjunto de páginas *c*;

Las primeras dos de estas operaciones son las operaciones básicas de E/S a nivel de páginas requeridas por el manejador de archivos. Las otras dos hacen posible el crecimiento y reducción de los conjuntos de páginas cuando se necesita.

Manejador de archivos

El manejador de archivos utiliza los recursos recién descritos del manejador de disco de manera tal que su usuario (el DBMS) puede percibir el disco como un conjunto de *archivos almacenados*. Cada conjunto de páginas contendrá uno o más archivos almacenados. *Nota*: El DBMS necesita saber cuándo dos archivos almacenados comparten el mismo conjunto de páginas o cuándo dos registros almacenados comparten la misma página.

Entre las operaciones que puede realizar el manejador de archivos con los archivos almacenados (es decir, las operaciones que puede solicitar el DBMS) están las siguientes:

- Leer el registro almacenado *r* del archivo almacenado *a*;
- Reemplazar el registro almacenado *r* dentro del archivo almacenado *a*;
- Añadir al archivo almacenado *a* un nuevo registro y devolver el nuevo identificador de registro *r*;
- Eliminar el registro almacenado *r* del archivo almacenado *a*;
- Crear un nuevo archivo almacenado *a*;
- Destruir el archivo almacenado *a*.

Con estas operaciones primitivas de manejo de archivos, el DBMS es capaz de construir y manipular las estructuras de almacenamiento.

Agrupamiento

El concepto básico del agrupamiento es procurar almacenar juntos físicamente los registros que tienen una relación lógica entre sí.

2.1.6.1. Llave Primaria

Una llave primaria o clave primaria es un identificador único para una tabla, y se forma de una columna o la combinación de varias columnas de dicha tabla (la tabla representa a una entidad y sus campos o columnas son los atributos de dicha entidad) con la siguiente propiedad: nunca existen dos filas o tuplas de la tabla con el mismo valor en esa columna o combinación de columnas [Ullman, 1982].

Esta clave primaria se construye a partir de la técnica llamada indización, por medio de la cual se obtienen los datos con mayor agilidad.

En esencia, los índices se pueden utilizar de dos maneras distintas. En primer término, pueden servir para tener acceso *secuencial* al archivo indexado, donde "secuencial" significa "en el orden definido por los valores del campo indexado". En segundo término, los índices pueden servir también para tener acceso *directo* a los registros individuales del archivo indexado con base en un valor dado del campo indexado.

De hecho, las dos formas básicas recién delineadas de utilizar índices se pueden generalizar aún más:

- *Acceso secuencial*: El índice puede ser útil también en consultas de intervalos.
- *Directo*: El índice puede ser útil también en consultas por *listas*.

2.1.6.1.1. Árboles B

Una clase de índices muy importante y de uso frecuente es el *árbol B*. Si bien no hay una estructura de almacenamiento óptima para todas las aplicaciones, también es casi seguro que, si ha de elegirse una sola estructura, el árbol B (de un tipo u otro) es la más conveniente. Los árboles B ofrecen, al parecer, el mejor desempeño general. (Por esta razón, la mayor parte de los sistemas relacionales incluyen árboles B como su forma principal de estructura de almacenamiento, y en varios casos es la única.)

En la variación de Knuth del árbol B, el índice tiene dos partes, el *conjunto secuencia* y el *conjunto índice* .

- El *conjunto secuencia* consta de un índice de un solo nivel de los datos reales; en circunstancias normales este índice es denso, pero podría ser no denso si el archivo indexado está agrupado según el campo indexado.

- El *conjunto índice*, a su vez, hace posible un acceso rápido al conjunto secuencia (y por tanto también a los datos). El conjunto índice es en realidad un índice con estructura de árbol del conjunto secuencia; de hecho, el conjunto índice es, en términos estrictos, el árbol B. La combinación del conjunto índice y el conjunto secuencia se denomina en ocasiones árbol "B más" (árbol B +).

  En la figura 2.10 se muestra un ejemplo de lo que es un árbol B +.

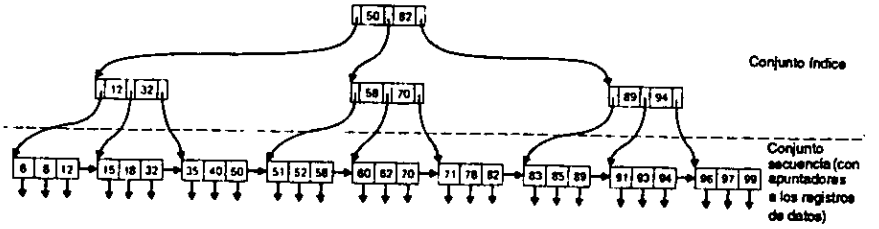


Fig. 2.10 Parte de un árbol B sencillo (variación de Knuth)

2.1.6.1.2. Hashing

El concepto fundamental en la organización de archivos por medio de un *acceso mediante hashing* consiste en poder dividir los registros de un archivo entre varios *buckets*, los cuales consisten de uno o varios bloques de almacenamiento [Ullman, 1982]. Por cada archivo de almacenamiento en esta forma se tiene una *función hash h* que se toma como un argumento de valor para la llave del archivo y producir un entero que va desde 0 a algún valor máximo. Si *v* es un valor clave, *h(v)* indica el número de bucket en el cual el registro con el valor clave *v* va a ser encontrado.

En la figura 2.11 se puede apreciar una organización de archivos en "hash" con *B* buckets. Este es un directorio de bucket el cual consiste de B puntos, uno para cada bucket. Cada punto es la dirección del primer bloque para cada bucket.

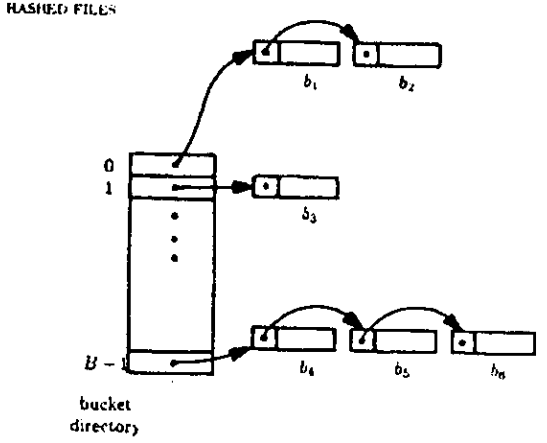


Fig. 2.11 Organización de Archivos en Hash

2.1.6.2. Llave Secundaria

La llave secundaria se forma por medio del método de indización, y su propósito es proporcionar criterios de búsqueda para comprobar -entre otros requerimientos- la existencia de grupos de registros que cumplan con los criterios de búsqueda. Aunque su técnica es igual al de una llave primaria, la diferencia consiste en el momento de plantear las búsquedas, porque no se busca algún registro específico, sino que se obtienen grupos de registros que cumplen los criterios antes establecidos. Un archivo almacenado puede tener cualquier cantidad de índices.

También es posible construir un índice con base en los valores combinados de dos o más campos. En general, un índice según la combinación de campos *C1*, *C2*, *C3*, ..., *Cn* (en ese orden) servirá también como índice según *C1* solo, según la combinación *C1C2* (o *C2C1*), según la combinación C1C2C3 (en cualquier orden). Así, el número total de índices necesarios para una indización completa de esta manera no es tan grande como pudiera parecer a primera vista.

El objetivo fundamental de un índice es acelerar la obtención de datos o, dicho de manera más específica, reducir el número de operaciones de E/S de disco necesarias para obtener algún registro necesario. En lo fundamental, este objetivo se logra mediante apuntadores, y hasta aquí se ha supuesto que todos estos apuntadores apuntan a *registros* (es decir, son identificadores de registro, o RID). Pero en realidad podría alcanzarse el objetivo postulado si esos apuntadores no fueran más que apuntadores de *página* (es decir, números de página).

Este concepto podría ir más lejos. Se recordará que un archivo almacenado tiene una sola secuencia "física", representada por la combinación de a) la secuencia de registros almacenados dentro de cada página y b) la secuencia de páginas dentro del conjunto de páginas al cual pertenecen. El índice no deberá por fuerza incluir una entrada para cada registro almacenado en el archivo indexado; sólo se necesita una entrada para cada *página*, con el número de página correspondiente.

2.1.6.2.1. Archivo Invertido

El concepto de archivo invertido ó listas invertidas se aplica a los índices por la siguiente razón: un archivo "normal" puede considerarse como todo aquel que almacena los registros de información, el cual incluye para cada registro, los valores de los campos de ese registro. En cambio, un índice incluye, para cada valor del campo indexado, los registros que contienen ese valor. Y un término más: cuando un archivo tiene un índice para cada campo se dice en ocasiones que está *totalmente invertido*.

2.2. Metodologías a Utilizar

Para iniciar esta sección, se empezará por definir los conceptos de *metodología* y *método*, los cuales son fundamentales para proseguir con el estudio de la metodología estructurada y la orientada a objetos.

Método:

Un método es un proceso disciplinado para generar un conjunto de *modelos* que describen varios aspectos de un sistema de software en desarrollo, usando alguna notación establecida [Booch, 1994].

Metodología:

Una metodología es una *colección de métodos* aplicados a través del ciclo de vida del desarrollo y unificado por algo en general, un acercamiento filosófico [Booch, 1994].

Una vez establecidos los conceptos de metodología y método, se procederá con el estudio de las metodologías *estructurada* y *orientada a objetos*, así como la explicación de sus distintos métodos.

2.2.1. Metodología Estructurada

La gran mayoría de los proyectos de desarrollo de sistema empezaban con la creación de una "novela victoriana" de requerimientos del usuario. Es decir, el analista escribía lo que entendía de los requerimientos del usuario en un enorme documento que consistía primariamente en una narrativa.

- Eran *monolíticos*: había que leer completamente la especificación, de principio a fin, para poder entenderla
- Eran *redundantes*: a menudo se requería la misma información en diversas partes del documento.
- Eran *ambiguas*: el reporte detallado de los requerimientos podía ser (y a menudo era) interpretado de diferente manera por el usuario, el analista, el diseñador y el programador.
- Eran *imposibles* de mantener: por todas las razones descritas anteriormente, la especificación funcional era casi obsoleta para cuando llegaba el final del proceso de desarrollo del sistema, y a menudo era obsoleto para el final de la fase de análisis.

Mientras se debatían todos estos problemas, ya se estaba adoptando un conjunto complementario de ideas en el área de programación y diseño. Estas ideas, normalmente conocidas como *diseño y programación estructurados*, prometían grandes mejoras en la organización, codificación, prueba y mantenimiento de los programas de computadora.

Como resultado, ha habido un movimiento tendiente a hacer especificaciones funcionales que sean:

- *Gráficas*: compuestas de una variedad de diagramas, apoyadas con material textual.
- *Particionadas*: de tal manera que se puedan leer independientemente porciones individuales de la especificación.
- *Mínimamente redundantes*: de tal manera que los cambios en los requerimientos del usuario puedan incorporarse normalmente en sólo una parte de la especificación.

Este enfoque, al que por lo general se conoce como análisis estructurado, se utiliza en la mayoría de las organizaciones de desarrollo de sistemas orientados a los negocios, al igual que en gran número de las orientadas hacia la ingeniería.

2.2.1.1. Cambios en el Análisis Estructurado Clásico

La mayoría de las organizaciones que usan el análisis estructurado basan su enfoque en los primeros textos de DeMarco, Gane y Sarson, y Weinberg y otros, al igual que en seminarios, videos y otros materiales de capacitación basados en dichos libros.

Los principales cambios son:

- El énfasis en la construcción de modelos "físicos actuales" y "lógicos actuales" del sistema del usuario se han demostrado que es políticamente peligroso por el tiempo y esfuerzo potencialmente desperdiciables.
- El análisis estructurado clásico hacía una distinción difusa y poco definida entre los modelos físicos y los modelos lógicos. Ahora se hará referencia a modelos esenciales (modelos de la "esencia" del sistema) en lugar de modelos lógicos, y a modelos de implantación en lugar de modelos físicos.
- Cada vez son más las organizaciones que están usando las técnicas del análisis estructurado para construir sistemas en *tiempo real*.
- Los *diagramas de transición de estados* se han añadido al análisis estructurado para permitir el modelado de sistemas en tiempo real. Y para permitir el modelado de sistemas con relaciones complejas entre datos se introdujeron los *diagramas de entidad-relación*.

- El proceso del análisis estructurado ha cambiado enormemente. El análisis estructurado clásico suponía que el analista empezaría por dibujar un diagrama de contexto, es decir, un diagrama de flujo de datos con una sola burbuja que representa a todo el sistema, y luego lo dividiría en varias funciones y almacenes de datos, en una forma estrictamente descendente, pero este proceso no funciona bien. En consecuencia, se ha añadido un nuevo enfoque, conocido como *división de eventos*.

2.2.1.2. Surgimiento de Herramientas Automatizadas

El trabajo necesario para crear diagramas de flujo de datos, diagramas de entidad-relación, diagramas de estructura, diagramas de transición de estados y otros modelos gráficos a menudo era abrumador.

Muchos de estos problemas se pueden resolver con apoyo automatizado adecuado. El desarrollo de poderosas estaciones de trabajo gráficas llevó a una industria llamada CASE (siglas que significan *Computer-Aided Software Engineering*: ingeniería de software auxiliada por computadora). Varios proveedores ofrecen productos (normalmente basados en PC) que dibujan diagramas de flujo de datos y otros, además de llevar a cabo una variedad de labores de revisión de errores.

2.2.1.3. Uso de Prototipos

En muchas organizaciones se considera que el análisis estructurado clásico consume demasiado tiempo; para cuando concluye la fase de análisis, el usuario habrá olvidado para qué quería en un principio el sistema.

Las herramientas de generación de prototipos (herramientas de software que permiten al analista construir un simulacro del sistema) se ven, por ello, como una solución efectiva a este problema. Nótese también que el uso de prototipos tiende a concentrarse en el aspecto de la interfaz humana en los proyectos de desarrollo de sistemas.

2.2.1.4. Unión del Análisis y el Diseño de Sistemas

El análisis estructurado trataba con la especificación de sistemas grandes y complejos, mientras que el diseño estructurado parecía ser más apropiado para el diseño de programas individuales que se ejecutaban en una misma computadora. El puente entre el análisis del sistema y el diseño de los programas es el *diseño de sistemas*.

2.2.1.5. Modelado

La mayoría de los sistemas requieren de *múltiples* modelos: cada modelo se enfoca a un número limitado de aspectos del sistema, a la vez que minimiza (o ignora totalmente) otros de sus aspectos [Yourdon, 1996].

Cualquier herramienta que se use para modelar debe tener las siguientes características:

- Debe ser gráfica, con detalles textuales de apoyo apropiados.
- Debe permitir que el sistema sea visto en segmentos, en forma descendente.
- Debe tener redundancia mínima.
- Debe ayudar al lector a predecir el comportamiento del sistema.
- Debe ser transparente para el lector.

2.2.1.5.1 Modelos Gráficos

La mayoría de los modelos populares de sistemas se apoyan en gráficas. Una imagen bien escogida puede transmitir de manera concisa y compacta una gran cantidad de información.

En general, se utilizan los gráficos para identificar los *componentes* de un sistema y su *interfaz*.

2.2.1.5.2 Modelos Segmentables en Forma Descendente

Un segundo aspecto importante de una buena herramienta de modelado es su capacidad de mostrar un sistema por partes en forma descendente.

Un buen modelo de un sistema complejo de información debe proceder de manera descendente, al igual que proporcionar un mecanismo conveniente para pasar tranquilamente de un nivel alto a uno bajo.

2.2.1.5.3. Modelos Mínimamente Redundantes

Los modelos son representación de algún sistema del mundo real y el sistema mismo pudiera ser estático (no cambiante) o dinámico. Si el sistema cambia, entonces debe cambiar el modelo, si ha de tenerse actualizado.

Para evitar la necesidad de tener que corregir en cada sitio donde se localice alguna instancia perteneciente a un elemento modificado del modelo, se buscará que dicho modelo tenga la menor cantidad de información redundante a lo largo de este, y así las modificaciones de aspecto local no repercutirían ó repercutirían en menor grado en dicho modelo, con lo cual el modelo no pierde precisión.

2.2.1.5.4. Modelos Transparentes

Finalmente, un buen modelo debe ser tan fácil de leer que el lector no tenga que detenerse a pensar siquiera que se trata de la *representación* de un sistema y no del sistema mismo.

2.2.2. Métodos Estructurados

Después de haber explicado las características fundamentales y la metodología estructurada para la creación de software, se procederá a dar un breve desglose de los principales métodos estructurados. Y nos concentraremos en el método "Ingeniería de la Información" [Martin], ya que éste resulta el más completo y representativo de todos los métodos estructurados, con lo cual se puede considerar como un método maduro que reúne todas las características de la metodología.

2.2.2.1. Modelo en Cascada

- Royce lo implanto por primera vez.
- Introdujo el concepto de ciclo de vida y se aplica en sistemas de información.

Consta de las siguientes etapas:

1.- Análisis de requerimientos.
2.- Especificaciones.
3.- Diseño.
4.- Implantación.
5.- Pruebas.
6.- Mantenimiento.

2.2.2.2. Modelo en Cascada Ampliado

- Lo usó Bohem.
- Su metodología tiene 7 etapas.
- Da a conocer más esta aplicación.

Consta de las siguientes etapas:

1.- Especificación de requerimientos en sistemas.
2.- Identificación de requerimientos en software.
3.- Diseño conceptual.
4.- Diseño detallado.
5.- Codificación y depuración.
6.- Integración, pruebas y preoperación.
7.- Operación, evaluación y mantenimiento.

2.2.2.3. Ciclo de Vida en Desarrollo Espiral

- Lo implantó Bohem.
- En cada etapa se abarca lo siguiente:
  - Formulación.- Identificar objetivos y tipos de software.
  - Análisis.- Identificación de alternativas.
  - Interpretación.- Se desarrolla alguna alternativa y se planea la siguiente etapa (se evalúa alguna alternativa).

Consta de las siguientes etapas:

1.- Definición.
2.- Desarrollo.
3.- Operación.

2.2.2.4. Método Estructurado [Yourdon]

- Lo implantó Yourdon.
- Inició con la Programación Estructurada.
- Consiste en la representación gráfica con elementos que faciliten la comunicación y representación de ideas.
- Su principal herramienta son los diagramas de flujo de datos.

Consta de las siguientes etapas:

1.- Análisis.
2.- Diseño.
3.- Programación.

Etapas:
1.- Análisis.
2.- Diseño.
3.- Programación.

2.2.2.5. Ingeniería de la Información [Martin]

- Autores: J. Martin; C. Finklestein.

Consta de las siguientes etapas:

1.- Planeación.- Arquitectura de la Información.
Arquitectura de Aplicaciones.
Arquitectura Técnica.
2.- Análisis.- Modelo de datos y actividades.
3.- Diseño.- Estructura sistemas para su desarrollo.
4.- Construcción.- Elaboración detallada del sistema.

2.2.2.5.1. Descripción de la Ingeniería en la Información

La Ingeniería en Información es una metodología para el desarrollo del ciclo de vida de sistemas de información que relaciona sistemas de negocios a los objetivos de negocios basado sobre el principio de reducir un sistema complejo en subsistemas simples [Mylls, 1994].

La Ingeniería de la Información esta compuesta por cuatro fases: Planeación, Análisis, Diseño, y Construcción. Las fases de Planeación, Análisis, Diseño, y Construcción están interligadas y dependen cada una de las otras porque estas determinan el cambio en el desarrollo y requerimientos de los negocios.

La Planeación establece las prioridades para el subsecuente Análisis. El Análisis determina los sistemas para el Diseño. Los Sistemas Diseñados son entonces Construidos, probados, y operados. La información producida por el sistema de información deberá satisfacer las necesidades de información identificadas en la Planeación. Cada fase esta sujeta a ciclos de iteraciones en un ambiente de cambio en requerimientos. Los cambios causados en cualquier fase pueden ocasionar modificaciones en otras fases. Las cuatro fases actuando conjuntamente definen la información enviada a la dirección para la empresa.

Ingeniería de la Información desarrolla arquitecturas que proveen la estructura para construir sistemas de información integrada que comparten datos entre funciones. Las arquitecturas son inicialmente desarrolladas en un nivel abstracto y entonces mediante las fases son entregadas en forma refinada. Cada fase del ciclo de implementación da una visión distinta de las arquitecturas.

Ingeniería en Información incorpora conceptos y principios que proporcionan el rigor para la transformación precisa entre fases de las arquitecturas.

2.2.2.5.1.1. Planeación

La misión de la Fase de Planeación es el asegurar que los sistemas de información son consistentes con los objetivos de negocio de la empresa. La Fase de Planeación produce el plan estratégico de información que refleja para donde va la empresa, no donde está actualmente.

Mientras la misión es fácilmente constatable, no siempre se realiza con facilidad.

La fase de Planeación crea el plan estratégico de información para toda o una parte de la empresa. Las actividades de Planear incluyen el análisis de los objetivos de la empresa de negocios, los cuales incluyen la declaración de la Misión, Objetivos, Metas, Estrategias, Factores de Sucesos Críticos, Planes, Funciones, Organización, Necesidades de Información, y Tipos de Entidades. Modelos de Datos y actividad (tipo de entidad y función) son construidos para mostrar la interacción entre los objetos de la empresa. Los tipos de entidad almacenan la información usada por la empresa, la cual es requerida para cumplir con las necesidades de información para medir el desempeño de como las metas van siendo logradas sobre los objetivos de negocios. Las prácticas y técnicas descritas son una guía para construir correctamente modelos de actividad con datos.

Modelos, diagramas, y mapas son usados para desarrollar tres arquitecturas de alto nivel las cuales forman la base para la construcción de sistemas de información, las cuales son:

- Arquitectura de la Información
- Arquitectura de Aplicaciones
- Arquitectura Técnica

De las tres arquitecturas creadas durante la Fase de Planeación, la Arquitectura de la Información es la más importante porque es la base sobre la que están los requerimientos del negocio. La Arquitectura de la Información es el modelo para encaminar las operaciones empresariales. La arquitectura de aplicaciones se deriva de la Arquitectura de la Información examinando el agrupamiento de los tipos de entidades y funciones para definir Áreas de Aplicación, Sistemas Potenciales de Negocios, y Áreas Subalternas.

2.2.2.5.1.2. Análisis

En teoría, la Fase de Análisis desarrolla los detalles de la lógica en los datos y la actividad en los modelos para áreas de aplicación definidas en la Fase de Planeación. En la práctica, la Fase de Análisis muchas veces se enfoca en determinar la aplicación del negocio.

Hay dos razones primarias para esbozar la arquitectura de la información de la Fase de Análisis. Una razón es la de tener mejor posición los datos para ser compartidos con funciones fuera del alcance de la Fase de Análisis proyectada. La probabilidad de compartir datos aumenta cuando los puntos de interfase están bien definidos. La otra razón es la de mejorar el alcance de la Fase de Análisis proyectada. No solamente son las funciones y procesos con el alcance del proyecto identificadas, pero, son tan importantes, como aquellas fuera de alcance.

La técnica básica entregada de la Fase de Análisis es

- Modelo Explícito de Datos
- Diagrama de Actividad Jerárquica para Procesos de Nodo Terminal
- Diagramas de Procesos de Acción
- Prototipos

Los resultados de la Fase de Análisis incluyen modelos completos de datos y actividad y recomendaciones proyectadas para la Fase de Diseño.

La dependencia entre datos proyectada en la Fase de Diseño usualmente determina la secuencia de implementación; esto es, los datos deben crearse antes de poderse referenciar o cambiar. La revisión al final de la Fase de Análisis considera los requerimientos de negocios del Área de Aplicación y los requerimientos técnicos de los proyectado y planeado en la Fase de Diseño.

2.2.2.5.1.3. Diseño

La Fase de Diseño usa la lógica de los datos y modelos de procesos para diseñar las representaciones externas de los sistemas ó aplicaciones. En esencia, los procesos de nodo final son empaquetados o sintetizados dentro de línea en diálogos de pantalla ó programas en lote.

No es inusual en la Fase de Diseño el agregar atributos, campos, o tipos de entidades diseñadas o físicas para el modelo de datos. Estas adiciones son el resultado de agregar detalles y usualmente son requeridas para controlar la ejecución de los sistemas de negocios.

La Fase de Diseño empieza la transición para la orientación técnica desde la orientación de negocios de las Fases de Planeación y Análisis. Esta fase empieza la transformación desde la lógica al diseño físico. La transformación es completada en la Fase de Construcción.

Los resultados de una Fase de Diseño proyectada incluyen pantallas completas definidas, reportes, y diálogos, ambos en línea y por lotes. También en los resultados, están varios cambios incluidos en modelos previos.

2.2.2.5.1.4. Construcción

La Fase de Construcción incluye no sólo la generación del código para sistemas de información sino también pruebas y modificaciones apropiadas para los modelos. Técnicas de construcción y prácticas son herramientas del kit CASE y arquitectura técnica específica. La Fase de Construcción crea los sistemas de información desde los datos y módulos de programa.

Todo el ciclo completo del kit de herramientas CASE automáticamente crea el código y las estructuras de las bases de datos desde los objetos definidos en los datos y los modelos de actividad.

La Fase de Diseño entrega como entrada para la Fase de construcción lo siguiente incluido:

- Modelo de Datos
- Diagrama de Jerarquización de Actividades
- Diagrama de Dependencia de Actividades
- Diagramas Detallados de Acción
- Desplegados de Pantalla
- Esquemas de Reportes
- Procesos en Línea
- Procesos por Lotes

Los primeros tres resultados son evaluados como referencia para los Constructores. Los últimos cinco resultados son directamente manipulados por los Constructores para producir un sistema de información trabajando. Los resultados de la Fase de Construcción son un sistema de información laborando y cambios apropiados para los resultados de entrada.

2.2.3. Modelo Relacional de Bases de Datos

El *modelo relacional* de datos fue inventado por E. F. Codd, y está basado en un solo concepto: las tablas. Un *DBMS relacional* (RDBMS) es un programa de computadora para administrar estas tablas. Un RDBMS, según la definición de Codd, tiene tres partes fundamentales:

- Datos que se presentan en forma de tablas
- Operadores para manipular las tablas
- Reglas de integridad aplicables a las tablas

A continuación se considerarán los aspectos antes mencionados.

2.2.3.1. Estructura lógica de datos de un RDBMS

Una base de datos relacional tiene el aspecto lógico de una sencilla colección de tablas. Las tablas tienen un número concreto de columnas, y un número arbitrario de filas. Las columnas de tablas se denominan *atributos* y se corresponden directamente con los atributos de los modelos de objetos. Las filas se denominan *tuplas* y se corresponden con instancias de objetos y con enlaces. En la intersección de cada fila y columna de una tabla se almacena un *valor simple*.

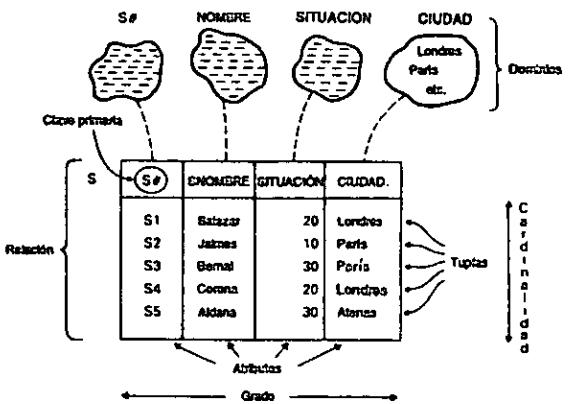


Figura 2.12 Descripción de una base de datos

Término relacional formal	Equivalentes informales
relación	Tabla
Tupla	fila o registro
cardinalidad	Número de filas
atributo	columna o campo
grado	Número de columnas
clave primaria	identificador único
dominio	Fondos de valores legales

Tabla referente a la terminología de la estructura de los datos

La teoría de las bases de datos relacionales impone que cada atributo debe de serle asignado a un dominio. Un *dominio* es un conjunto de valores válidos. Los dominios comportan más información que un simple formato de datos, y hacen posible una mayor comprobación semántica

Todo valor de una tabla debe de pertenecer al dominio de su atributo, o bien puede ser nulo. *Nulo* quiere decir que el valor de un atributo es desconocido, o que no tiene sentido para una fila dada. Existen complejos problemas teóricos acerca de los valores nulos, y suelen dar lugar a problemas para las aplicaciones reales.

2.2.3.2. Operadores de los RDBMS

El SQL se ha convertido en el lenguaje más popular para los RDBMS, y además es un estándar ISO y ANSI. Desafortunadamente, SQL está muy lejos de ser un lenguaje ideal, y tiene muchas deficiencias técnicas. Por ejemplo, SQL viola los principios de la moderna teoría de lenguajes.

El ámbito del estándar SQL es pequeño e incompleto; no trata temas importantes tales como el ajuste de rendimiento y las herramientas de productividad para la programación. Sin embargo, se utilizará por ser un estándar.

SQL proporciona operadores para manipular tablas. La instrucción *select* de SQL sirve para consultar tablas. La sintaxis de la orden *select* tiene un aspecto parecido a este:

SELECT lista-de-atributos
FROM tabla-1, tabla-2, ...
WHERE predicado-es-cierto

Lógicamente, la tabla-1, la tabla-2 y cualquier otra se combinan en una sola tabla temporal. La lista de atributos especifica las columnas que deben de retenerse en la tabla temporal. La expresión del predicado indica las filas que hay que retener. El contenido de la tabla temporal se proporciona como respuesta de la consulta. Hay otras órdenes de SQL que crean tablas, insertan filas en tablas, borran filas de tablas y efectúan otras operaciones.

Las órdenes SQL interactivas están orientadas a conjuntos; operan sobre tablas completas en lugar de actuar sobre filas o valores individuales. SQL ofrece un lenguaje similar para su uso en programas de aplicación, mediante una interfaz farragosa que se aplica a una fila de cada vez. Los RDBMS pueden tratar tablas completas, pero la mayoría de los lenguajes de programación no pueden.

2.2.3.2.1. SQL

SQL es al mismo tiempo un lenguaje de consulta interactiva y un lenguaje de programación para bases de datos. En su enfoque como lenguaje de programación suele llamársele "SQL embebido".

El principio fundamental del SQL embebido, al cual llamaremos *principio del modo dual* es el siguiente: *cualquier proposición de SQL que se pueda utilizar en una terminal también se podrá emplear en un programa de aplicación* [Date, 1998].

Debe indicarse claramente que el principio del modo dual se aplica a todo el lenguaje SQL, no sólo a las operaciones de manipulación de datos.

La mayor parte de esas operaciones pueden manejarse en forma bastante directa (o sea, con un mínimo de modificaciones de su sintaxis).

2.2.3.2.1.1. SQL Dinámico

El "SQL dinámico" está formado por una serie de recursos de SQL embebido, que se ofrecen de manera específica para hacer posible la construcción de aplicaciones generalizadas, en línea (y posiblemente interactivas). (Una aplicación en línea permite el acceso a la base de datos desde una terminal en línea).

A grandes rasgos, debe realizar las siguientes acciones:

1. Aceptar un mandato de la terminal.
2. Analizar ese mandato.
3. Emitir proposiciones SQL apropiadas a la base de datos.
4. Devolver un mensaje o resultado, o las dos cosas, a la terminal.

2.2.3.2.1.2. Observaciones

Se concluirá el análisis de SQL embebido exponiendo varias observaciones sobre el enfoque SQL en la programación de aplicaciones de bases de datos.

1.- El empleo de lo que es en esencia el mismo lenguaje para el acceso tanto interactivo como programado a la base de datos tiene una consecuencia muy importante: las porciones de base de datos de un programa de aplicación se pueden probar y depurar de manera interactiva.

2.- En segundo lugar, ese mismo hecho (a saber, que se emplee en esencia el mismo lenguaje en ambas interfaces) facilita en gran medida la comunicación entre los usuarios y los programadores de aplicaciones (y también la comunicación con el DBA).

3.- Como las operaciones "manipulativas" en SQL se compilan (es decir, se ligan antes del momento de la ejecución), estos sistemas ofrecen una ventaja importante en cuanto al desempeño con respecto a sistemas en los cuales se adopta un enfoque interpretativo más convencional. Todas estas operaciones:

- Análisis sintáctico de la solicitud original;
- Detección e informe de los errores de sintaxis;
- Elección de una estrategia de acceso;
- Verificación de la autorización;
- Generación de código ejecutable,

No necesitan realizarse en el momento de la ejecución. (De estas operaciones, la más importante desde el punto de vista del desempeño es la elección de una estrategia de acceso; es decir, la optimización.) Así, la trayectoria de ejecución es mucho más corta; y como las instrucciones de la trayectoria de ejecución deben ejecutarse en cada solicitud, y como muchas veces esas solicitudes se presentan dentro de los ciclos del programa, el ahorro puede ser considerable.

4.- Aun en el caso de proposiciones preparadas y ejecutadas en forma dinámica, donde todas las operaciones mencionadas están por fuerza dentro de la trayectoria de ejecución, el enfoque de compilación puede resultar provechoso.

5.- Una desventaja de SQL como lenguaje de programación de aplicaciones es su difícil acoplamiento con el lenguaje anfitrión.

2.2.3.3. Integridad de un RDBMS

Un aspecto importante de los RDBMS es su apoyo para la integridad. Los dos aspectos de la integridad en el modelo de Codd son la integridad de entidades y la integridad de referencias. La integridad de entidades impone que toda tabla tenga exactamente una clave primaria. Una *clave primaria* es una combinación de uno o más atributos cuyo valor ubica sin ambigüedad a cualquier fila de la tabla. En la figura 2.13, *ID-persona* es la clave primaria de la tabla *Persona*, *ID-compañía* es la clave primaria de la tabla *Compañía*.

tabla Persona

ID-persona	nombre-persona	dirección	ID-compañía
1	Juan García	Olivas, 314	1001
5	Mario Bravo	Ancha, 722	1002
999	Juan García	Mayor, 1561	1001
14	Juana Bravo	Ancha, 722	nulo

tabla Compañía

ID-compañía	nombre-compañía	dirección
1001	Ajustes Pérez	Adobe, 33
1002	AAA Licores	Ancha, 724
1003	Velas López	Industria, 1877

ID-persona es la clave primaria de la tabla Persona
ID-compañía es la clave primaria de la tabla Compañía
ID-compañía es una clave externa dentro de la tabla Persona

Figura 2.13 Clave primaria y clave externa

La integridad de referencias exige que el RDBMS mantenga la congruencia de todas las claves externas con su correspondiente clave primaria. Una *clave externa* es una clave primaria de una tabla que está incluida dentro de otra tabla (o bien dentro de la misma). En la figura 2.13, *ID-compañía* es una clave externa de la tabla *Persona*. No sería admisible modificar la *ID-compañía* de Mario Bravo para que fuera 1004, porque 1004 no está definida en la tabla *Compañía*. Si se borrase la fila correspondiente a Ajustes Pérez dentro de la tabla *Compañía*, entonces habría que dar el valor nulo a las dos filas de Juan García presentes en la tabla *Persona*. El enlace entre la clave externa y la clave primaria es con frecuencia una vía de navegación entre tablas.

El estándar SQL y las implementaciones comerciales de RDBMS van convergiendo lentamente hacia un apoyo para la integridad referencial.

2.2.3.4. DBMS Relacionales Avanzados

Un objetivo explícito de los RDBMS avanzados es hacer el menor número de cambios posible en el modelo relacional. La meta principal es extender el modelo relacional con nuevos tipos de datos, operadores y métodos de acceso. Por ejemplo, una aplicación matemática puede requerir los números complejos como tipo de datos. Una aplicación de tablas de decisión puede requerir nuevas operaciones que comprueben la compleción de la tabla de decisiones y que consoliden las filas correspondientes a casos en los cuales sea irrelevante un criterio de decisión. Los métodos de acceso especiales, como la búsqueda en retícula, servirían de ayuda para las aplicaciones espaciales.

No hay ningún DBMS que pueda implementar las características necesarias para satisfacer a todas las aplicaciones. De esta forma, un RDBMS avanzado debe de proporcionar un sistema de apoyo que capacite a los desarrolladores de aplicaciones para añadir posibilidades personalizadas. La arquitectura es abierta, por contraste con respecto a la arquitectura cerrada de los RDBMS convencionales.

Algunas de las ventajas de los RDBMS avanzados son:

- *Una mejora indiscutible con respecto a la funcionalidad de los RDBMS convencionales.* Se mantienen las ventajas tradicionales de los RDBMS: muchos usuarios simultáneos, grandes cantidades de datos, fiabilidad, manejo de datos distribuidos, y herramientas de apoyo para la programación.
- *Se integra bien con bases de datos relacionales ya existentes.* Proporciona un flujo de datos suave entre las aplicaciones de ingeniería y las de negocios.
- *Datos compartidos.* La base de datos es realmente un depósito central y no está ligada a ningún lenguaje de programación o aplicación particular.

Algunas de las posibles desventajas con:

- *Rendimiento.* ¿Puede un RDBMS (aunque esté mejorado) realizar operaciones eficientes en objetos individuales?
- *Funcionalidad.* ¿Interfiere el paradigma del RDBMS con la posibilidad de proporcionar las capacidades necesarias? De ser así, ¿causarán estas limitaciones posibles problemas para las aplicaciones reales?
- *Seguridad.* Una arquitectura abierta puede hacer más difícil proteger los datos contra lectura o escritura no autorizadas.

2.2.4. Metodología Orientada a Objetos

El desarrollo de las metodologías de ingeniería de software fue propiciado por las necesidades de atender problemas específicos de la producción de sistemas cada vez más complejos. Vistas en perspectivas puede notarse una evolución es la especialización y el enfoque de los nuevos métodos. Estos nuevos métodos tienen la ventaja de fundamentar sus propuestas en un estudio crítico de los métodos anteriores, a fin de ofrecer una metodología más próxima a un estándar.

Una de las diferencias principales entre la metodologías tradicionales y las metodologías orientadas a objetos es que los procedimientos tradicionales estaban limitados al desarrollo de sistemas convencionales de procesamiento de datos. Por otra parte, las metodologías orientadas a objetos pueden utilizarse para desarrollar cualquier tipo de sistema se instrumente o no con tecnología orientada a objetos.

Durante muchos años, la orientación a objetos se ha asociado de manera exclusiva con un tipo especial de lenguaje de programación , Hoy en día los conceptos empleados por los lenguajes de programación orientados a objetos son aplicados como una filosofía de tipo general para el desarrollo de sistemas. Esto no significa que el desarrollo de sistemas orientado a objetos quede especificado en términos de clases que físicamente abarquen definiciones de variables de objetos y métodos codificados. Tampoco significa que un sistema quede especificado en función de códigos heredados. Aunque las nociones de estructura de clase y de herencia son utilizados para la definición de la orientación a objetos, éstas de hecho son simplemente instrumentaciones de la orientación a objetos . De manera conceptual la orientación a objetos a llegado a interpretar de forma más general.

Los métodos orientados a objetos abarcan métodos de diseño y métodos de análisis , Hay ocasiones en que se produce un cierto grado superposición, y decir que se trata de actividades completamente disjuntas no pasa de ser un idealización. Hodgson (1990) Afirma que el proceso de desarrollo de un sistema consta de comprensión, invención y realización, por lo cual el dominio del problema se capta o comprende, en primera aproximación, como fenómenos , conceptos, actividades, roles y afirmaciones. Esto es una forma de compresión , y corresponde por completo al análisis . Sin embargo la compresión del dominio del problema implica , también el entendimiento de marcos de trabajo, componentes, modelos computacionales y otras estructuras mentales que tienen en cuenta los posibles dominios de soluciones. Esta actividad inventiva corresponde al proceso de diseño.

2.2.5. Métodos Orientados a Objetos

Análisis de Diseño Orientado a Objetos.

El Análisis Orientado a Objetos (OOA, object – oriented analysis) es el nombre para una clase de métodos que analizan un problema mediante el estudio de los objetos dentro del ámbito en el que se presenta.

Un objeto es una abstracción de algo dentro del terreno de un problema o su producción , que refleja las capacidades de un sistema para mantener la información sobre él, interactúa con él o con ambos.; un encapsulado de los valores de atributo y de sus servicios exclusivos.
Una clase describe un conjunto de objetos con atributos y servicios comunes. Un objeto es una instancia de una clase y el acto de crear un objeto se denomina "instanciar".

Las clases pueden ser descompuestas en subclases. Por ejemplo, pueden existir diversos tipos de Paquetes de Telemetría, y pueden ser creadas subclases del mismo tales como Paquetes de Fotometría y Paquete de Espectrometría, Las subclases comparten características familiares, y el OOA asegura lo anterior permitiendo que las subclases hereden las operaciones y los atributos de sus parientes.

El análisis orientado a objetos puede ser visto como una síntesis de los conceptos de objetos explorados por primera vez en los lenguajes de programación Simula67 y Smalltalk, y en las técnicas de análisis estructurado, particularmente el modelado de datos. El Análisis orientado a objetos difiere del análisis estructurado en dos aspectos:

Construir un modelo de objeto, en vez de un modelo funcional (p.ej. la jerarquía de los diagramas de flujo de datos)

Integrar los objetos, atributos y operaciones en vez de separarlos entre el modelo de datos y el modelo funcional.

El OOA ha sido muy exitoso en derribar problemas que se resisten al análisis estructurado, como las interfaces de usuario. Además, los partidarios de OOA argumentan que los objetos dentro de un sistema son más fundamentales (importantes, necesarias) para su naturalcza que las funciones que proporciona. Las especificaciones basadas en los objetos serán más adaptables que las especificaciones basadas en las funciones.

Los métodos que marcan las directrices del OOA son:

Coad – Yourdon
Rumbaugh et al. Object Modelling Technique (OMT)
Shlaer – Mellor
Booch.

Los métodos OOA están evolucionando, y los analistas a menudo combinan las técnicas de los diferentes métodos cuando analizan los problemas. Se recomienda a los usuarios de los métodos OOA que adopten un enfoque pragmático semejante.

Coad – Yourdon

Describe un Método de Análisis Orientado a Objetos Basado en cinco actividades principales.

- Definición de clases y objetos
- Identificación de estructuras
- Identificación de temas.
- Definición de atributos
- Definición de servicios.

Estas actividades son usadas para construir cada capa de un modelo de objeto de cinco niveles.

Los objetos existen en el ámbito del problema. Las clases son abstracciones de objetos. Los objetos son instancias de clases. La primera tarea del método es identificar las clases y los objetos.

La segunda tarea del método es identificar las estructuras. Se reconocen dos tipos de estructuras: estructuras de generalización – especialización y estructuras globales. El primer tipo de estructura es como un árbol genealógico: es posible la herencia entre los miembros de una estructura. El segundo tipo de estructura es utilizado para modelar relaciones de entidades (p.ej. cada motor contiene una armadura)

Los modelos grandes y complejos pueden necesitar una organización temática, con cada tema dedicado a un aspecto particular del problema. Por ejemplo el modelo de objetos de un vehículo de motor puede tener una perspectiva mecánica o una perspectiva eléctrica.

Los atributos caracterizan a cada clase. Por ejemplo, un atributo de una máquina puede ser el número de cilindros. Cada objeto tendrá un valor para ese atributo.

Los servicios definen lo que los objetos hacen. Definir los servicios es equivalente a definir las funciones del sistema.

La importancia fundamental del Método Coad – Yourdon es su descripción breve y concisa, así como el uso de textos generales como fuentes para las definiciones; de modo que las definiciones se enmarcan dentro del sentido común y reduce el empleo de modismos. La debilidad principal del método es su notación compleja, la cual es difícil de utilizar sin el apoyo de una herramienta. Alguno usuario s del método Coad – Yourdon han utilizado la notación de diagramación de OMT en su lugar.

OMT

La Técnica de Modelado de Objetos (OMT Object modelling technique) transforma la definición del problema del usuario en tres modelos:

- Modelo de objetos
- Modelo dinámico
- Modelo funcional.

Los tres modelos en conjunto crean el modelo lógico requerido por los Estándares de Ingeniería de Software.

El modelo de objetos muestra la estructura estática en el mundo real. El procedimiento para construirlo es el siguiente.

- Identificación de los objetos
- Identificación de las clases de objetos
- Identificación de las asociaciones entre los objetos
- Identificación de los atributos de los objetos.
- Uso de herencia para organizar y simplificar la estructura de clases.
- Organización de las clases y asociaciones estrechamente acopladas dentro de los módulos.
- Acompañar a cada objeto con breves descripciones textuales.

Los tipos más importantes de asociación son la adición y la generalización.

El modelo dinámico muestra el comportamiento del sistema, especialmente la secuencia de interacciones. El procedimiento para construirlo es el siguiente.

Identificar las secuencias de eventos dentro del ámbito del problema y documentarlo en el seguimiento de eventos.

Construir un diagrama de transición de estados para cada objeto que sea afectado por los eventos, mostrando los mensajes que fluyen, las acciones que son realizadas y los cambios en el estado de los objetos que tienen lugar cuando ocurren los eventos.

El modelo funcional muestra la forma en que se obtienen los valores, sin considerar en el momento en que sean computadas. El procedimiento para construirlo no requiere el uso de la descomposición funcional, sino de.

- Identificar la entrada y salida de valores que el sistema recibe y produce
- Construir diagramas de flujo de datos que muestren la forma en que los valores de salida son computados a partir de los valores de entrada.
- Identificar los objetos que son utilizados como almacenes de datos.
- Identificar las operaciones de los objetos que comprenden cada proceso.

El modelo funcional es sintetizado a partir de las operaciones de objetos, en vez de descomponerlo desde una función de alto nivel. Las operaciones de los objetos pueden ser definidos en cualquier etapa durante el modelado.

Los aspectos más importantes del OMT son sus capacidades simples aunque poderosas de notación así como su madurez. Ha sido aplicado en varios proyectos por sus actores antes de ser publicado. La principal debilidad es la falta de técnicas para integrar los modelos de objetos, los modelos dinámicos y los modelos funcionales.

Schlaer – Mellor

Schlaer y Mellor comienzan el análisis identificando el ámbito del problema del sistema. Cada ámbito es un mundo separado habitado por sus propias entidades conceptuales u objetos. Los ámbitos más grandes son divididos en subsistemas. Después, cada ámbito o subsistema es analizado de forma separada en tres etapas.

- Modelo de la información.
- Modelado de estado.
- Modelado de procesos.

Las tres actividades de modelado crean en conjunto el modelo lógico requerido por los Estándares de Ingeniería de Software.

El objetivo del modelado de información es identificar:

- Los objetos dentro del sistema.
- Los atributos de cada objeto.
- Las relaciones entre cada objeto.

El modelo de información es documentado mediante diagramas de definiciones de objetos y definiciones de objetos, atributos y relaciones.

El objetivo del modelo de estado es identificar.

- Los estados de cada objeto, y las acciones que se realizan sobre ellos.
- Los eventos que causan que los objetos pasen de un estado a otro.
- La secuencia de estados que forman el ciclo de vida de cada objeto.
- Las secuencias de mensajes que comunican los eventos que fluyen entre los objetos y los subsistemas.

Los modelos de estado son documentados mediante diagramas de modelo de estados (mostrando las secuencias de estados), diagramas de modelos de comunicación de objetos (mostrando los objetos que fluyen entre los estados), y listas de eventos.

El objetivo del modelo de proceso es identificar.

- Las operaciones de cada objeto requeridas durante cada acción.
- Los atributos de cada objeto que son almacenados en cada acción.

Los modelos de proceso son documentados mediante diagramas de acción de flujo de datos, mostrando las operaciones y el flujo de datos que ocurren en cada acción, y los diagramas de modelo de acceso de objeto., mostrando el acceso de datos entre objetos, Los procesos complejos también deben ser descritos.

La fuerza del método Schlaer – Mellor. Es su madurez (sus autores declaran haber estado desarrollándolo desde 1979) y la existencia de técnicas para integrar los modelos de información, los modelos de estado y los modelos de proceso. La principal debilidad del método es su complejidad.

Booch

Booch modela su diseño orientado a objetos desde un punto de vista lógico, el cual define las clases, los objetos y sus relaciones; y desde un punto de vista físico, que define la arquitectura de módulos y procesos. La perspectiva lógica corresponde al modelo lógico que deben construir los ingenieros de software de acuerdo a los requerimientos de los Estándares de Ingeniería de Software. El método orientado da objetos Booch tiene cuatro pasos.

- Identificación de clases y objetos a un nivel dado de abstracción.
- Identificación de la semánticas de estas clases y objetos.
- Identificación de las relaciones entre esas clases y objetos.
- Implementación de las clases y objetos.

Las primeras tres etapas deben ser completadas dentro de la etapa de requerimientos del sistema. La última etapa es realizada dentro de las fases de AD y DD, Booch sostiene que el proceso de diseño orientado a objetos no es deductivo, ni inductivo, sino algo que él denomina round – trip gestalt design (diseño gestalt (conocimiento) de viaje redondo) . El proceso desarrolla un sistema a manera de incrementos e interacciones. A los usuarios del método de Booch se les advierte que deben integrar las SR y AD en una sola fase de modelado.

Booch ofrece cuatro técnicas para la documentación de la perspectiva lógica.

- Diagramas de clase, los cuales son utilizados para mostrar la existencia de clases y sus relaciones.
- Diagramas de objetos, los cuales son usados para mostrar la existencia de objetos y su comportamiento, especialmente relacionados con la comunicación de mensajes.
- Diagramas de estado transición , los cuales muestran los estados posibles de cada clase, así como los eventos que ocasionan las transiciones de un estado a otro.
- Diagramas de tiempo, los cuales muestran la secuencia de operaciones de los objetos.

Los libros de Booch sobre métodos orientados a objetos han sido descritos por Stroustrup, el inventor de C++, como únicos y mejores libros sobre el tema. Este cumplido revela los diversos alcances y la profundidad de un buen análisis y práctica de diseño en sus escritos. Sin embargo, la notación de Booch es molesta y hay pocas herramientas disponible.

UML (unified modeling language)

El Unified Modeling Language (UML) es una notación que evolucionó a partir del diseño en Booch, OMT (Rumbaugh), OOSE (Jacobson) y de los métodos orientados a objetos. Las metas claves para el desarrollo del UML están en:

- Integrar la mejor práctica en la industria del software en una notación y terminología aceptada comúnmente.
- Proporcionar la habilidad para representar todo de los conceptos generalmente relevantes para los sistemas de software , y;
- Preparar Software Avanzado que proporcione los siguientes diagramas gráficos para el diseño del software.
- Diagrama de Uso de Casos (Use Case Diagram).
- Diagrama de la Estructura de Clases (Class Structure Diagram).
- Diagrama de Comportamiento (Behavior Diagram).
- Diagrama de Estado (State Diagram).
- Diagramas de Secuencias (Séquense Diagram)
- Diagramas de Colaboración (Collaboration Diagram)
- Diagrama de Actividad (Activity Diagram).
- Diagrama de Implementación (Implementation Diagram)
- Diagrama de Componentes (Component Diagrama)
- Diagrama de Despliegue (Deployment Diagram).

Modelado de Casos de Uso con UML.

Es una técnica orientada a los Casos de Uso y al manejo de los requerimientos necesarios durante el progreso de un proyecto. El modelo de Casos de Uso fue introducido por primera vez por Ivar Jacobson en el Object Oriented Software Engineering (OOSE)

Requerimientos
Análisis
    Casos de Uso
    Diagramas de Clases
    Diagramas de Estado.
Diseño.
Diagramas de Interacción.
    Diagramas de Colaboración
    Diagramas de Secuencia
Diagramas de Actividad
Diagramas del Detalle de Clases
Implementación.
    Diagramas de Componentes.
    Diagramas de Despliegue.

ROOM (Real Time Object – Oriented Modeling)

ROOM es una metodología del Modelo Orientado a Objetos de Tiempo Real desarrollado por la compañía Object Time Developer. Esta metodología ofrece varios puntos importantes.

La complejidad esencial de reactivar sistemas es suficientemente alta para requerir conceptos de modelado especializado.

ROOM toma ventajas de muchos desarrollos recientes de la tecnología de computadoras (métodos formales, el paradigma de objetos, interfaces gráficas al usuario).

También ROOM fue diseñado explícitamente para tomar ventaja de la automatización basada en computadoras de las de mas actividades mecánicas de desarrollo.

Esto proporciona un potencial único para beneficios significativos en productividad y calidad.

Estructura del modelado.

- Utiliza sintaxis gráfica para una fácil comprensión
- Abstracciones para tratar con fenómenos arquitectónicos de alto nivel de grandes sistemas de tiempo real.
- Protocolo basado en múltiples interfaces.
- Objetos concurrentes (Actores)
- Estructuras Dinámicas
- Estructura reproducidas.

Modelado de Comportamiento.

- Alto nivel basado en sintaxis gráfica.
- Utiliza máquinas de estado jerárquicas
    - Permite elegir el modelado poderoso de capacidades, al tiempo que permite implementaciones automatizadas avanzadas y eficientes.
- Prioridad basada en la manipulación de eventos para tratar con requerimientos de tiempo real.
- Incorporar la industria de lenguajes de programación estándar (por ejemplo C++) para detalles específicos.

Diseño Orientado a Objetos

El diseño orientado a objetos (OOD) es un enfoque del diseño de software basado en objetos y clases. Un objeto es una abstracción de algo en el dominio de un problema o su implementación, reflejando las capacidades de un sistema para proporcionar información acerca de él mismo, interactuar con él o con ambos; es un encapsulamiento de valores de atributo y sus servicios exclusivos. Una clase describe un conjunto de objetos con atributos y servicios comunes.

Las clases pueden ser entidades con tipos de datos abstractos como el Paquete de Telemetría y Espectro, así como entidades más simples con tipos de datos primitivos como números reales, enteros y cadenas de caracteres. Una clase es definida por sus atributos y servicios.

Las clases pueden ser divididas en subclases. Por ejemplo, pueden existir varios tipos de paquetes de telemetría, y pueden ser creadas subclases de Paquetes de Telemetría tales como Paquete de Fotómetro y Paquete de Espectrómetro. Las subclases comparten características familiares, y el ODD permite para ello, que las subclases hereden operaciones y atributos de sus padres. Esto conduce hacia sistemas modulares y estructurados. Que requieren menos código para ser implementados.

Los métodos de diseño orientado a objetos, a diferencia de otros, ofrecen un mejor soporte para la reutilización. El mecanismo tradicional de reuso de la programación de abajo hacia arriba donde es perfectamente posible que un modulo de aplicación llame a un modulo de librería. Además, la característica de herencia permite el reuso en la programación de arriba hacia abajo de los atributos y las operaciones de las superclases.

El OOD combina servicios e información e incrementa la modularidad, Las estructuras de control y datos pueden ser definidas en una manera integrada.

Otras características del enfoque orientado a objetos, además de las clases, los objetos y la herencia son la transmisión de mensajes y el polimorfismo. Los objetos envían mensajes a otros objetos para dirigir sus servicios. Los mensajes también son utilizados para transmitir información. El polimorfismo es la capacidad, al momento de la ejecución del programa, para referirse a las instancias de varias clases. El polimorfismo es a menudo implementado para permitir enlaces dinámicos.

Al igual que el diseño estructurado, el diseño orientado a objetos no es un método único, sino un nombre para designar una clase de métodos. Los miembros de esta clase incluyen.

- Booch;
- Diseño Jerárquico Orientado a Objetos (HOOD);
- Coad – Yourdon;
- Técnicas del Modelado de Objetos (OMT) de Rumbaugh et. Al;
- Shlaer – Mellor.

Booch

Booch creó el diseño orientado a objetos, y continua jugando un papel principal en el desarrollo del método. Booch modela un diseño orientado a objetos en términos de un enfoque lógico, el cual define las clases, los objetos y sus relaciones, y un enfoque físico.

El enfoque lógico corresponde al modelo lógico que requieren los estándares de la Ingeniería de Software para construir en la fase de SR. El enfoque físico corresponde al modelo físico que estos mismos estándares requieren para construir en la fase de AD.

Booch proporciona dos técnicas de diagramación para documentar el enfoque físico.

Diagramas de módulo, los cuales son utilizados para mostrar la asignación de clases y objetos hacia los módulos, como los programas, paquetes y tareas en el diseño físico (El termino modulo) en el método de Booch es utilizado para describir cualquier componente del diseño).

Diagramas de procesos, los cuales muestran la asignación de módulos hacia los procesadores de hardware.

HOOD (Hierarchical Object Oriented Design)

El diseño jerárquico orientado a objetos (HOOD) es miembro de una familia de métodos orientados a objetos que tratan de integrar la orientación a objetos con los métodos de diseño estructurado. La jerarquía sigue naturalmente a la división del objeto raíz del nivel más alto. Al igual que en el diseño estructurado, las parejas de datos fluyen entre los componentes del software. La principal diferencia entre HOOD y los métodos estructurados es que los componentes del software obtienen su identidad de su correspondencia con cosas en el mundo real, en vez de las funciones que el sistema tiene que realizar.

HOOD fue originalmente diseñado para utilizarse con Ada, aunque Ada no soporta la herencia, y no es un lenguaje de programación orientado a objetos, Este no es un problema serio para el diseño HOOD. Porque el método no utiliza clases para estructurar los sistemas.

HOOD no tiene un método de análisis complementario. El modelo lógico normalmente es construido utilizando el análisis estructurado. La transformación del modelo lógico al modelo físico es difícil. Haciendo difícil la construcción de un diseño coherente.

Coad y Yourdon

Coad y Yourdon han publicado un enfoque integral para el análisis y diseño orientado a objetos. Un diseño orientado a objetos es construido a partir de 4 componentes.

- Componente del ámbito del problema;
- Componente de interacción humana;
- Componente de administración de tareas;
- Componente de administrador de datos.

Cada elemento está compuesto de clases y objetos. El componente del ámbito del problema está basado en el modelo (lógico) construido con el OOA en la fase de análisis. Define el tema de estudio del sistema y sus responsabilidades. Si el sistema va a ser implementado en un lenguaje orientado a objetos, la correspondencia entre las clases y los objetos del ámbito del problema serán uno a uno, y el componente del ámbito del problema podrá ser programado directamente. Sin embargo, el refinamiento substancial del modelo lógico es normalmente requerido, resultando en la incorporación de más atributos y servicios.

Los componentes poco amigables en las interacción humana envían y reciben mensajes a y desde el usuario. Las clases y objetos en el componente de interacción humana tienen nombres que son tomados desde el lenguaje de interfaz del usuario. Por ejemplo: una ventana y un menú.

Muchos sistemas tendrán hijos múltiples de ejecución, y el diseñador debe construir un componente de manejo de tareas para organizar el procesamiento. El diseñador necesita definir tareas como manejo de eventos (event driven) o manejo del tiempo (clock – driven), así como sus prioridades. De manera crítica.

El componente de la administración de datos proporciona la infraestructura para guardar y recuperar objetos. Puede ser un simple sistema de archivos, un sistema de administración de base de datos relacional, o igualmente un sistema de administración de bases de datos orientado a objetos.

Los cuatro niveles juntos hacen el modelo físico del sistema. En el nivel más alto, todos los diseños de Coad y Yourdon Orientado a Objetos tienen a misma estructura.

La fuerza de los métodos Coad y Yourdon son sus instrucciones, descripciones breves y su uso de texto general como fuentes de definiciones, así como las definiciones se ajustan al sentido común y el argot es minimizado. El significado de debilidad del método és su notación compleja, la cual es difícil de utilizarla sin herramientas de soporte.

OMT

La técnica de modelación de objetos (Object Modelling Technique OMT) de Rumbaugh contiene dos actividades de diseño.

- Diseño de sistemas;
- Diseño del objeto.

El diseño del sistema debe ser ejecutado en la fase de AD. El diseño del objeto debe ser ejecutado en la fase de DD.

Los pasos convencionales del diseño del sistema son:

- Organizar el sistema en subsistemas y ordenarlos en capas y divisiones
- Identificar la concurrencia inherente en el problema
- Asignar subsistemas a los procesadores
- Definir la estrategia de implementación del administrador de datos

- Identificar las fuentes globales y definir el mecanismo para controlar el acceso a ellos
- Elegir un enfoque para la implementación de control, de software
- Considerar las condiciones de los límites
- Establecer cambios fuera de las prioridades.

Muchos sistemas son pocos similares, y Rumbaugh sugiere que el diseño del sistema esté basado en una de varias áreas de trabajo o arquitectura canónica. Lo único que ellos proponen es Transformación de lotes una transformación de datos ejecutada una vez en conjunto completo de entradas.

- Transformación continua, una transformación de datos ejecutada continuamente como cambios en las entradas.
- Interfase Interactiva , un sistema dominado por interacciones externas;
- Simulación dinámica, un sistema que simula objetos del mundo real.
- Sistemas de Tiempo real, un sistema dominado por restricciones de tiempo rígido.
- Administrador de Transacciones, un sistema concerniente con almacenamientos y actualización de datos.

El enfoque OMT para el diseño de sistemas contiene varias ideas de diseño que son aplicadas de manera genérica.

Shlaer y Mellor

Shlaer y Mellor describen un lenguaje de Diseño Orientado a Objetos (OODLE), derivado de la notación de Booch y Buhr.

Existen cuatro tipos de diagramas.

- Diagrama de Clases;
- Gráfica de la estructura de Clase (class);
- Diagrama de Dependencias;
- Diagrama de Herencia.

Existe un diagrama de clase para casa clase. El diagrama de clase define las operaciones y atributos de la clase.

Las gráficas de la estructura de clase definen la estructura del modulo de la clase (class), el control y el flujo de datos entre los módulos de las clases.

Los diagramas de Dependencia muestran las dependencias entre clases, las cuales pueden ser.

- Cliente – servidor.
- Amigables.

Una dependencia cliente – servidor existe cuando una clase (el cliente) llama a las operaciones a otras clases (el servidor).

Una dependencia de amistad existe cuando una clase accesa el dato interno de otra clase. Esto es una violación de información – ocultación.

Los diagramas de herencia muestran la herencia de relaciones entre clases.

Shlaer y Mellor definen un método de diseño recursivo que utiliza la notación OODLE como sigue.

- Define como el proceso genérico de computación será implementado;
- Implementar los objetos del modelo de clases utilizando los procesos genéricos de computación

El enfoque del diseño de Shlaer – Mellor es más complejo que los otros métodos orientados a objetos.

UML (Unified Modeling Language)

Diseño

Diagrama de secuencia. En este diagrama se muestra el tiempo de secuencia de la interacción, así como de la participación de los objetos. Además contiene dos dimensiones de secuencia que consisten en una dimensión vertical (tiempo), y horizontal (objetos diferentes).

Diagrama de colaboración. Este diagrama muestra una interacción organizada alrededor de los objetos en la interacción y sus ligas entre otros. Además de las relaciones entre los objetos, no son en tiempo, pero la secuencia de números son utilizados para mostrar la secuencia de mensajes.

Diagrama de Actividad. Este es un diagrama especial de un diagrama de estados, donde todos los estados o al menos la mayoría de ellos, son estados de acción y gran parte de las transiciones son activadas por la realización de acciones en los estados fuente.

Todo el diagrama de actividad es ligado (a través del modelo) a las clases o a la implementación de una operación o un uso de casos.

La propuesta de este diagrama es enfocarse en el manejo de flujos por procesamiento interno. Se utiliza los diagramas de actividad en situaciones donde todo o la mayoría de los eventos representan la realización de acciones generadas internamente (procedimientos de flujo de control).

Implementación.

Diagramas de implementación. Las dos formas son los diagramas de componentes, los cuales muestran la estructura del tiempo de ejecución del sistema.

Diagrama de componentes. Dependencias entre los componentes del software muestran componentes del código fuente, componentes del código binario, y componentes ejecutables. Algunos componentes existen en tiempos de ejecución, y otros existen en más de un tiempo.

Diagramas de Despliegue. Los diagramas de despliegue muestran la configuración del tiempo de ejecución del procesamiento de elementos y de componentes de software, procesos, y objetos que viven en ellos. Instancias de los componentes del software representen la manifestación del tiempo de ejecución de las unidades de código.

ROOM (Real Time Object Oriented Modeling)

Esta metodología integra tanto al análisis como al diseño del sistema dentro de un mismo proceso en el desarrollo del sistema. A diferencia del enfoque tradicional, caracterizado por una compleja división en las etapas del proceso ROOM busca simplificar una relación más estrecha entre las actividades.

Enfoque ROOM

El proyecto es contemplado en dos niveles con relaciones formalmente definidas entre niveles.

Alto nivel (Modelado Arquitectónico)	ROOM Lenguaje de Modelado esquemáticos
Modelado detallado	Lenguajes de Programación Análisis y Diseño Implementación.

Incorpora lenguajes de programación estándar (por ejemplo: C++ para el modelado detallado.

El objetivo de ROOM es proporcionar un método y un lenguaje de modelado para el desarrollo de software para sistemas de tiempo real distribuidos

Es conveniente para el manejo de eventos del sistema.

- Ofrece un método del ciclo completo (Análisis – Diseño – Implementación)
- Utiliza un lenguaje formal de modelado gráfico.
- Permite la automatización basada en modelos ejecutables y generación automática de código).

2.2.6. Modelo Orientado a Objetos de Bases de Datos

Como cualquier base de datos programable, una base de datos orientada a objetos (BDOO) da un ambiente para el desarrollo de aplicaciones con un depósito persistente listo para su explotación. Una bases de datos orientadas a objetos almacena y manipula información que puede ser digitalizada (representada) como objetos, proporciona una estructura flexible con acceso ágil, rápido, con gran capacidad de modificación. Además combina las mejores cualidades de los archivos planos, las bases jerárquicas y relacionales.

Las bases de datos orientadas a objetos las extensiones orientadas a objetos para las tecnologías de bases de datos existentes ofrecen no sólo formas de almacenar y recuperar datos sino también los mecanismos para definir y gestionar las complejas relaciones entre los datos. Las bases de datos orientadas a objetos no almacenan datos aisladamente sino más bien siguen el paradigma orientado a objetos de vincular los datos junto con el comportamiento asociado dentro de los objetos.

Las capacidades funcionales de las bases de datos orientadas a objetos incluyen soporte de construcciones ricas en modelado de datos orientadas a objetos, soporte directo a inferencia o deducción, y la posibilidad de almacenar tipos de datos, como por ejemplo, imágenes, audio o voz y video. Las bases de datos orientadas a objetos van más allá de la simple representación de datos con números y textos pasivos; en su lugar modelan con mayor exactitud el mundo con su riqueza y textura.

Funcionalidad de las bases de datos orientadas a objetos.

Las bases de datos orientadas a objetos ofrecen partes de la misma funcionalidad que los lenguajes orientados a objetos. Permiten la encapsulación dentro de los objetos de los datos y métodos que actúan sobre ellos. Activan métodos mediante mensajes a los objetos. Permiten la declaración de relaciones jerárquicas entre los objetos a través del uso de la herencia. Además, las bases de datos orientadas a objetos ofrecen a las bases de datos tradicionales la funcionalidad de la que carecen los lenguajes orientados a objetos, como persistencia y participación.

La tecnología de base de datos depende de varias características importantes que van más allá de las que ofrecen los lenguajes orientados a objetos. Estas características incluyen:

Persistencia: Creando objetos que sobreviven al proceso que las creó.

Integridad: Garantizando que los cambios en la base de datos no destruyen su consistencia.

Compartición: Permitiendo procesos múltiples para coordinar el acceso simultáneo a los objetos de datos.

Consulta: Utilizando expresiones lógicas para definir un subconjunto de la base de datos para su acceso.

Características orientadas a objetos	Características de bases de datos
Objetos	Persistencia
Métodos	Integridad
Herencia	Compartición
	Consulta

Características orientadas a objetos y bases de datos tradicional de las bases de datos orientadas a objetos.

Las bases de datos orientadas a objetos extienden aún más la semántica de los datos. Permitiendo arbitrariamente la definición de tipos de datos complejos y proporcionando un medio para asociar el comportamiento con los datos, la semántica de las bases de datos orientadas a objetos están más cerca de sus equivalentes del lenguaje de programación.

La orientación a objetos desvía la atención del programador para centrarse más de cerca en las preocupaciones de los diseñadores de bases de datos enfatizando la organización del software alrededor de los datos más que en el flujo del control.

Las bases de datos se centran en un método más declarativo, datos compartidos fuera del dominio de las aplicaciones y soporte para grandes cantidades de datos.

Las bases de datos orientadas a objetos difieren significativamente en funcionalidad respecto a las bases de datos relacionales. Las bases de datos relacionales se basan en derivar una estructura virtual en la ejecución basada en valores de los conjuntos de datos almacenados en tablas. Las bases de datos orientadas a objetos contienen objetos predefinidos que no necesitan derivarse en el momento de la ejecución . En una base de datos relacional, las vistas se construyen seleccionando datos de múltiples datos y cargándolos en una única tabla. En una base de datos orientada a objetos una vista se obtiene pasando punteros de objeto en objeto.

Una base de datos orientada a objetos juega un papel muy activo, mientras que una base relacional es pasiva. Mientras una base de datos relacional ofrece principalmente la capacidad de añadir o borrar registros, la base de datos orientada a objetos ofrece la capacidad de incorporar métodos dentro de los objetos, permitiendo así a la base de datos incorporar muchas de las operaciones que deben dejarse a las aplicaciones con una base de datos relacional. Las funciones fundamentales de una base de datos orientada a objetos son:

Objetos

Los objetos son los elementos que una base de datos orientada a objetos almacena, modifica y recupera por orden del programa de la aplicación. Estos objetos pueden ser tan simples como números y cadenas o tan complejos como las especificaciones completas de un circuito electrónico. Tienen cabida también los tipos múltiples de datos, incluyendo gráficos, sonido y video. Una de las principales ventajas que las bases orientadas a objetos tienen sobre las bases de datos tradicionales es su capacidad para representar como objeto cualquier entidad del mundo real.

Los objetos hacen que sea más fácil la escritura de las aplicaciones, porque todos los datos de una entidad determinada están localizados en un lugar, el programador no necesita buscar a través de múltiples archivos por medio de punteros. Además de los datos, los objetos pueden almacenar relaciones, representadas internamente como enlaces a otros objetos, y pueden almacenar el comportamiento, representado internamente como métodos.

Los objetos pueden hacer también que las aplicaciones se ejecuten más rápido. Como los datos de una entidad están relacionados lógicamente, la base de datos orientada a objetos tiene los medios para optimizar su localización física. Las aplicaciones son capaces de leer menos archivos para recuperar todos los datos relevantes.

El almacenamiento de objetos como elementos de datos cambia la realización de otras características que todas las bases de datos deben proporcionar. Como los objetos pueden transferirse eficazmente en una sola operación, es más fácil desarrollar algoritmos de caché de disco para transferir grupos lógicos de objetos. El acceso concurrente multiusuario, el procesamiento de transacciones, el control de versión, la seguridad y los esquemas de recuperación, son más fáciles de realizar debido a que puede ponerse un único bloqueo en todos los datos relevantes a nivel de objetos, en vez de requerir del programa de aplicaciones que intenten establecer bloqueos múltiples a través de los datos relacionados, aunque dispersos, en una base de datos relacional.

A diferencia de las bases de datos relacionales, que solo permiten conjuntos del mismo tipo, las bases de datos orientadas a objetos permiten conjuntos arbitrarios de objetos. Estos conjuntos pueden manipularse por el usuario o por la aplicación, bloqueados para gestión de transacciones, o grupos para optimización del rendimiento y facilidad de acceso.

Métodos

Cada objeto en una base de datos orientada a objetos puede tener encapsulados en su interior un número de métodos para actuar sobre sus datos. Estos métodos son activados cuando el programa de la aplicación envía mensajes al objeto. La encapsulación de métodos y datos elimina buena parte del trabajo de programación para el programador, desviando la responsabilidad de recuperación y actualización de la base de datos hacia la propia base de datos.

Herencia

La herencia proporciona un medio de reducir el esfuerzo necesario para desarrollar y mantener una base de datos. Pero esta capacidad de permitir que una base de datos envejezca graciosamente es bastante más importante. Cuando se utiliza adecuadamente, la herencia permite que se añadan nuevas características y tipos de datos a una base de datos exigiendo solamente unos cambios muy concretos.

Una base de datos orientada a objetos proporciona un conjunto de tipos y operaciones incorporados. Más importante aún: cada objeto de una base de datos orientada a objetos es un miembro de una clase, que determina la estructura del objeto y define que operaciones pueden realizarse sobre él. Las nuevas clases se crean a partir de las clases existentes por medio de la técnicas de la subclasificación y obviando los métodos heredados. Esto da como resultado operaciones y tipos de datos complejos arbitrariamente, que pueden ser tratados entonces como si fueran tipos incorporados.

La herencia no solamente ayuda en este proceso de definir el tipo y las relaciones de los datos a almacenar en una base de datos, sino que también reduce el esfuerzo necesario para acomodar los inevitables cambios en la estructura de la base de datos. La adición de un nuevo tipo de datos es llevada a cabo fácilmente por medio de la subclasificación, con la garantía de que los tipos de datos y métodos existentes pueden estar localizados en aquella clase de la jerarquía de clases en al que se define el tipo de datos o métodos. Finalmente, debido a que los cambios en la estructura jerárquica de la base de datos continúan para asociar los datos que están relacionados lógicamente, el mecanismo de la herencia soporta funciones de base de datos como bloqueo, autorización y consulta.

Bibliotecas de clases

Las bibliotecas de clases predefinidas son suministradas por vendedores de bases de datos orientadas a objetos. Las bibliotecas de clases permiten a los programadores reutilizar, en lugar de tener que volverlas a crear, las estructuras de datos más comúnmente necesarias como por ejemplo matrices, diccionarios tablas y fechas. Las bibliotecas de clases incluyen también métodos, como recorrido lineal o dispersión.

Las bibliotecas de clases para las bases de datos orientadas a objetos difieren principalmente de las bibliotecas de clases para los lenguajes orientados a objetos en el tipo de clases que definen, Ambos incluyen clases para los tipos básicos como enteros, reales y cadenas, así como tipos de agregados o totales como listas, conjuntos y matrices.

Las bibliotecas de clases para las bases de datos orientadas a objetos también incluyen clases para objetos persistentes, excepciones, directorios, bloqueos, esquemas y demás funcionalidad propia de una base de datos.

Persistencia

Se dice que un objeto posee persistencia si continúa existiendo después que ha terminado el programa de la aplicación. Normalmente, los objetos que existen durante la ejecución de un programa orientado a objetos surgen por medio de una de estas tres formas.

1. El objeto es creado estáticamente o dinámicamente por el programa.
2. El objeto se recupera a partir de un archivo o base de datos relacional y se convierte en forma objeto.
3. El objeto se recupera a partir de una base de datos orientada a objetos y ya está listo para su uso.

Los objetos persistentes contienen datos que permanecen durante más tiempo que el que emplea el programa de aplicación. A diferencia de las aplicaciones construidas a partir de lenguajes orientados a objetos, en las que los objetos se crean durante la ejecución y finalizan con la sesión de la aplicación, los objetos de una base de datos sobreviven múltiples sesiones.

La persistencia de un objeto está íntimamente ligado al concepto de identidad de objeto, que requiere que todo objeto tenga algo que sea único e invariable, con independencia de la forma en que el objeto cambie. La identidad del objeto permite una gestión eficaz de relaciones complejas. Si los objetos se refieren entre sí por medio de sus únicos identificadores de objeto, sus relaciones continuarán incluso cuando los objetos cambien su estado o situación. La identidad del objeto contrasta con las bases de datos basadas en valores, como las bases de datos relacionales, en las que las entidades se identifiquen por sus atributos y por ello pueden cambiar a lo largo del tiempo.

Consulta

Una diferencia fundamental entre sistemas de bases de datos relacionales y orientadas a objetos radica en la cantidad de información que se mueve entre la aplicación y el sistema de gestión de bases de datos. Cuando las aplicaciones envían mensajes a las bases de datos orientadas a objetos, la base de datos manipula los datos con los métodos, recupera o calcula un valor, y devuelve el valor a la aplicación.

En una base de datos relacional las consultas se aplican secuencialmente, de forma que una consulta depende del resultado de la consulta anterior. La base de datos no conoce la petición global; conoce solamente las consultas individuales y por consiguiente no puede reordenar las consultas para optimizarlas . En una base de datos totalmente orientada a objetos, un solo mensaje toma el lugar de muchas consultas de base de datos relacional. Un mensaje puede solicitar cálculos y hacer que se envíen mensajes a otros objetos antes de entregar un resultado.

Integridad

Cada programa que accede a una base de datos es una amenaza potencial para la integridad de la base de datos. Los sistemas de gestión de base de datos están en guardia contra estas amenazas proporcionado restricciones de la integridad, o condiciones que deben obedecer siempre los elementos de datos.

Un medio utilizado por las bases de datos orientadas a objetos para bloquear violaciones de las restricciones es un mecanismo de excepciones como el que se emplea en Persist (Juniper 1989). Cuando se encuentra una situación excepcional en una base de datos Persist, el programa genera una excepción creando un objeto excepción y transfiriendo el control al gestor de excepciones correspondientes. El objeto excepción puede entonces ser interrogado por el gestor de excepciones para comunicar información sobre la situación excepcional. Como una excepción es un objeto, éste tiene un tipo permitiendo por tanto el agrupamiento por tipos de condiciones similares de excepciones.

Otro medio de proporcionar restricciones de la integridad es por medio de disparadores. Los disparadores son mecanismos conectados o agregados a determinados elementos datos dentro de una base de datos que se activan siempre que se produce un intento de acceso o modificación de los elementos datos.

Las bases de datos orientadas a objetos proporcionan un método único para realizar los disparadores efectuando el seguimiento de las llamadas o invocaciones a los métodos. Como los datos pueden accederse solamente a través de los métodos, la base de datos puede identificar los intentos de modificación de los datos efectuando el seguimiento de aquellos métodos que tienen acceso a los datos e invocando un disparador adecuado siempre que se invoque un método. En concreto, en una base de datos orientada a objetos, en la que cualquier operación puede ser definida por el usuario, el sistema no pueden determinar fácilmente por adelantado cómo podrían ser afectadas otras operaciones. El problema está en localizar aquellos puntos críticos en los que deben activarse las defensas de la integridad. Es un problema general que está pendiente todavía de resolver

Herramientas para el desarrollo de aplicaciones

Es Fundamental que el SGBDO posea distintos tipos de herramientas para el desarrollo de aplicaciones, que pueden ir desde lenguajes de desarrollo hasta visualizadores (browser) que permiten una mejor manipulación de las bibliotecas de clase. Sin embargo los lenguajes de cuarta generación (L4G) de los SGBO dejan en algunos casos bastante que desear, sobre todo si los comparamos con los actuales de los SGBDR, lo que tampoco es de extrañar dado que estos últimos llevan años desarrollándose y han alcanzado un mayor grado de madurez.

Herramientas para la transferencia de datos

Los SGBDO deben proporcionar herramientas que permitan transferir datos entre distintos tipos de SGBD, pudiéndose volcar datos de un SGBO a un sistema relacional o viceversa. Este temas, conocido también como importación / exportación de datos, es uno de los más activos tanto en el campo de los SGBD. Actualmente se está trabajando en un Marco de Referencia para la estandarización de transferencia entre gestores de datos. Este trabajo lleva consigo la definición del formato de los ficheros de importación / exportación, así como el acuerdo sobre la definición de los datos (con independencia de la facilidad de importación / exportación).

Herramientas y facilidades de usuario

El SGBO debe proporcionar distintas interfaces a los diferentes usuarios del sistema según la función que desempeñan en cada momento; por ejemplo administrador de la base de datos, programador de servicios, diseñador de clases, usuario final, etc.

Prototipos y Productos

En los últimos años se han desarrollado numerosos prototipos de SGBO tanto en departamentos universitarios como en centros de investigación de empresas, algunos de los cuales se han convertido o han servido de base para productos comerciales.

Entre los prototipos destacan: IRIS (desarrollado por los laboratorios de Hewlett Packard, en Palo Alto California , con las contribuciones entre otros, de P.H. Fishman. D. Beech, W. Kent y P.L.ynbaek, que han servido de base al producto de OpenODB de HP; el MCC, bajo la dirección de Won Kim cuyas conclusiones se han aprovechado en el desarrollo del producto ITASCA de la empresa del mismo nombre; POSTGRES (Desarrollado en la Universidad de California, bajo la dirección de Mike Stonebraker) como sucesor de INGRES; y O2 (desarrollado en Altair, por O.Deux, G. Arango, F. Bancilhom, C. Delovel, F. Vélez, etc.) del que ya existe una versión comercial.

Otros prototipos de interés son: ZEITGEIST (Texas Instrument), OZ+ (Universidad de Toronto), PROBE (Xerox), OBSERVER / ENCORE (Universidad de Brown), STARBURST (IBM), ODE (ATT), AVANCE (Universidad de Estocolmo), POSE (Rensselaer Polytechnic Institute), Mneme (Universidad de Massachussets), MANDRIL (Hitachi) y ODIN (NEC).

En la actualidad se está desarrollando un conjunto de estándares, que afectan de manera muy directa a los sistemas de bases de datos orientados a objeto.

En cuanto a modelos de objetos, entre los trabajos más importantes destacan los llevados a cabo por el OMG (Object Management Group), consorcio formado por más de cuatrocientas empresas interesadas en la tecnología de objetos, que además de definir un modelo de objetos básicos propugna la arquitectura CORBA (Common Object Resquest Broker Architecture) con el fin de lograr portabilidad e interoperabilidad entre los sistemas de objetos.

A pesar de todas las propuestas, uno de los estándares más representativos en la actualidad es el ODMG-93

ODMG –93

El principal objetivo del ODMG (Object Data Management Group), es definir un conjunto de estándares que permita a los clientes de un SGBDO escribir aplicaciones "portables" también persiguen conseguir la interoperabilidad entre SGBO.

El estándar ODMG-93 de divide en las siguientes partes

Modelo de Objetos, basado en el propuesto por el OMG (Object Management Group) y extendido con los conceptos propios de base de datos constituyendo un perfil para los SGBO, de forma análoga al que define el ORB (Object Request Broker).

Lenguaje de Definición de Objetos, en siglas inglesas ODL, (Object Definition Languaje), que se utiliza para describir la interfaz de los objetos: está basado en el IDL (Interface Definition Languaje) de CORBA, por lo que tiene un estilo muy parecido al de C++

Lenguaje de Consulta de Objetos, cuyas siglas originales son OQL (Object Query Languaje) y que se basa en el lenguaje SQL, aunque no se persigue una compatibilidad con éste al 100%, ya que se limitaría enormemente la claridad y poder de un lenguaje de consulta de objetos.

Vinculación con el lenguaje C++

Vinculación con el lenguaje Smalltalk.

En estas dos últimas partes se ha procurado que los programadores piensen que están utilizando un solo lenguaje, esto es que se integren el lenguaje de manipulación de datos (OML, Object Manipulación Languaje) y el lenguaje anfitrión de forma natural y consistente.

Ventajas de las Bases de Datos Orientadas a Objetos

En sistemas diseñados con lenguajes orientados a objetos, los objetos se crean durante la ejecución de un programa y se destruye cuando finaliza el programa. Proporcionar una base de datos que pueda almacenar los objetos entre ejecuciones de un programa ofrece flexibilidad y seguridad crecientes. La capacidad de almacenar objetos permite también que los objetos sean compartidos en un entorno distribuido. Una base de datos orientada a objetos puede permitir solamente que los objetos utilizados activamente sean cargados en memoria, y así minimizar o asegurar con antelación la necesidad de un paginación de memoria virtual. Esto es especialmente útil en sistemas de gran escala. Los objetos persistente permiten también que los objetos sean almacenados para cada versión. Este control de versión es útil no solamente para aplicaciones de comprobación o prueba, sino también para muchas aplicaciones de diseño orientadas a objetos en las que el control de versión es una exigencia funcional de la aplicación en sí. El acceso a otra fuentes de datos puede ser también facilitado con las bases de datos orientadas a objetos.

Las bases de datos orientadas a objetos ofrecen así mismo muchas de las ventajas que antiguamente se encontraban solamente en los sistemas expertos. Con una base de datos orientada a objetos, las relaciones entre objetos y las limitaciones en los objetos se mantienen por el sistema de gestión de la base de datos, es decir los mismos objetos. Las reglas asociadas con el sistema experto son sustituidas fundamentalmente por el esquema objeto y los métodos. Como muchos sistemas expertos no poseen actualmente un soporte de base de datos adecuado. Las bases de datos orientadas a objetos permiten la posibilidad de ofrecer funcionalidad de sistema experto con mucho mejor rendimiento.

Las bases de datos orientadas a objetos proporcionan ventajas en comparación con los actuales modelos de bases de datos jerárquicas y relacionales. Posibilitan soporte de aplicaciones complejas que no son soportadas perfectamente por el resto de modelos. Amplían la "programabilidad" y el rendimiento, mejoran el acceso a la navegación y simplifican el control de concurrencia . Disminuyen los riesgos asociados con la integridad referencial.

Las bases de datos orientadas a objetos permiten por definición, la inclusión de una mayor cantidad de código en la propia base de datos. Este mayor conocimiento de la aplicación tiene bastantes ventajas potenciales para el propio sistema de base de datos, incluyendo la capacidad de optimizar el proceso de consultas y controlar la ejecución concurrente de transacciones.

El rendimiento, concepto siempre importante en la realización del sistema, puede mejorarse significativamente utilizando un modelo orientado a objetos en lugar de un modelo relacional. La mejora más grande puede esperarse en aplicaciones con alta complejidad de datos y un gran número de interrelaciones. El agrupamiento (clustering), o localización de objetos relacionados en proximidad cercana, puede llevarse a cabo a través de la jerarquía de clases o por medio de otras interrelaciones. El caching o retención de determinados objetos en memoria o almacenamiento, puede ser optimizado anticipado que el usuario o la aplicación puede recuperar una determinada variable instancia (modelo) de la clase. Cuando existe una gran complejidad de datos, las técnicas de agrupamiento y caching en las bases de datos orientadas a objetos obtienen enormes ventajas en rendimiento, que las bases de datos relaciónales. debido a su arquitectura fundamental, nunca podrán seguir.

Las bases de datos orientadas a objetos pueden almacenar no sólo componentes complejos de aplicación sino también estructuras de datos mayores. Aunque los sistemas relacionales pueden soportar un gran número de tuplas (p. ej. filas en una tabla). éstas, consideradas individualmente, están limitadas en tamaño. Las bases de datos orientadas a objetos que disponen de grandes objetos no sufren una degradación en rendimiento porque los objetos no necesitan ser descompuestos y vueltos a ensamblar por las aplicaciones, sin importar la complejidad de las propiedades de los objetos de la aplicación.

Como los objetos contienen referencias directas a otros objetos, los conjuntos de datos complejos pueden ser ensamblados eficazmente utilizando estas referencias directas. La capacidad de buscar por referencias directas mejora significativamente el acceso por navegación. En contraste, los conjuntos de datos complejos en las bases de datos relacionales deben ensamblarse por el programa de la aplicación utilizando el lento proceso de unir tablas.

Para el programador, uno de los retos en la construcción de una base de datos es el lenguaje de manipulación de los datos (DML, Data Manipulation Languaje) de la base de datos. Los DML para las bases de datos relacionales difieren normalmente del lenguaje de programación utilizado para construir el resto de la aplicación. Este contraste es debido a diferencias en los paradigmas de programación y falta de coincidencias de los sistemas de tipo. El programador debe aprender dos lenguajes, dos conjuntos de herramientas y dos paradigmas, porque ninguno de ellos tiene la funcionalidad de construir aplicaciones completas. Ciertos tipos de herramientas de programación, como por ejemplo los generadores de aplicación y lenguajes de cuarta generación (4GL), han surgido para producir el código de toda la aplicación, cerrando así el intervalo de falta de coincidencia entre el lenguaje de programación y el DML; pero la mayoría de las herramientas comprometen el proceso de programación de la aplicación.

Con las bases de datos orientadas a objetos se elimina gran parte de este problema. El DML puede ampliarse para que una mayor parte de la aplicación pueda escribirse en el DML. O también, puede emplearse un lenguaje de aplicación orientado a objetos, por ejemplo C++ para que sea el DML.

Una mayor parte de la aplicación puede construirse en la propia base de datos. El movimiento a lo largo del interfaz de programación entre la base de datos y la aplicación ocurre entonces en un único paradigma con un conjunto común de herramientas. Las bibliotecas de clases pueden ayudar también al programador a acelerar la creación de las bases de datos. Las bibliotecas de clases fomentan la reutilización del código existente y ayudan a minimizar el costo de posteriores modificaciones.

La programación es más fácil porque las estructuras de datos modelan el problema de un amanera más fiel. La disposición de datos y procedimientos encapsulados en un único objeto hace que sea menos probable que un cambio en un objeto afecte a la integridad de otros objetos en la base de datos. también se simplifica el control de la concurrencia con una base de datos orientada a objetos. En una base de datos relacional, la aplicación necesita bloquear explícitamente cada registro en cada tabla , ya que los datos relacionados se representan a lo largo de un buen número de tablas. La integridad, una exigencia clave de las bases de datos, puede soportarse mejor con una base de datos orientada a objetos, porque la aplicación puede bloquear todos los datos relevantes de una operación.

La integridad referencial no está garantizada en una base de datos relacional, en la que el borrado de un registro de un tabla puede dejar pendiente un puntero hacia el otro registro, a menos que la aplicación compruebe específicamente esta situación. La integridad referencial está mejor soportada en una base de datos orientada a objetos, porque los punteros se mantienen y actualizan por la propia base de datos. Finalmente, las bases de datos orientadas a objetos ofrecen una mejor metáfora de usuario que las bases de datos relacionales. La tupla o tabla, aunque posibilita una estrategia de realización bien definida, no es una estructura de modelación intuitiva, especialmente fuera del dominio de los números. Los objetos ofrecen una metáfora de modelación natural y completa.

2.2.7. Diferencias entre la Metodología Estructurada y la Orientada a Objetos

Una de las diferencias entre las metodologías tradicionales y las orientadas a objetos es que los procedimientos tradicionales estaban limitados al desarrollo de sistemas convencionales de procesamiento de datos. Por otra parte, las metodologías orientadas a objetos pueden utilizarse para desarrollar cualquier tipo de sistema.

La programación orientada a objetos busca minimizar el impacto del cambio a través de la técnica de encapsulación. También apoya la reutilización de código ya existente y hace que está sea práctica

Aunque los lenguajes estructurados reconocen como meta la reutilización de código, no han podido producir códigos susceptibles de modificarse con facilidad. Aun cuando se citan las bibliotecas en lenguajes estructurados como medios para reutilizar código, la estabilidad necesaria para las bibliotecas es un impedimento para adaptarlas de manera fácil para problemas especiales de caso. La programación orientada a objetos supera esta restricción a través de los nexos dinámicos.

La diferencia con la metodología orientada a objetos estiba en que se subraya por igual la importancia de datos y acciones. La metodología estructurada tiende a acentuar las acciones. Las funciones y procedimientos se consideran como las unidades básicas de construcción. Esta perspectiva da origen a la técnica de refinamiento descendente por pasos. Las acciones abstractas se descomponen en pasos cada vez más concretos hasta que cada uno de ellos puedan especificarse como una proposición en lenguaje de programación. Sin embargo, muchos problemas se resuelven mejor trabajando de abajo hacia arriba o en ambas direcciones al mismo tiempo, o incluso utilizando un tercer sistema en el cual la estructura ni siquiera una parte superior (sistemas manejados por eventos). La construcción de componentes reutilizables es igual de importante que el diseño de la estructura general. Las bibliotecas de clase orientadas a objetos proporcionan componentes reutilizables, sin atrapar a los usuarios en un conjunto limitado de tipos de datos.

2.3. Teoría del Inventario

2.3.1. Antecedentes

El mantenimiento del inventario es un problema común para todas las empresas en cualquier sector de la economía. Es necesario mantener inventarios en la industria, en la agricultura, en el comercio, etc. debido a que física y económicamente es imposible conseguir los bienes en el momento preciso en que se demandan. El no tener inventarios implica que los consumidores tengan que esperar hasta que las órdenes sean surtidas por algún proveedor, o bien, que sean fabricadas, y, por lo general, los demandantes no están dispuestos a esperar por mucho tiempo. Esta razón hace que el mantenimiento del inventario sea necesario para todas las empresas, tanto manufactureras como distribuidoras de artículos. Para las empresas manufactureras, el inventario tiene una importancia vital, ya que apoya las operaciones de producción y ventas, influyendo directamente en el monto de inversiones en equipo de transporte, producción y nivel de empleo. De esta manera, es posible analizar el inventario en términos de la relación, producción-inventario.

Manteniendo un nivel de inventario suficiente, se podrá sostener un ritmo de producción estable, y lograr, por lo tanto, una alta utilización de la capacidad de la planta, quedando así en aptitud de satisfacer la demanda y proporcionar un servicio adecuado al consumidor.

2.3.1.1. El Inventario

Por inventario entendemos al conjunto de operaciones que se llevan a cabo para conocer las cantidades que hay de cada producto en el almacén en un momento determinado.

Esas operaciones están íntimamente ligadas al modelo, organización y al grado de información de los Almacenes. Un sistema moderno dispondrá de la información sobre existencias en tiempo real. Por el contrario un sistema rudimentario de control de Almacenes no permite garantizar que la información de existencias sea confiable en el momento necesario a causa de retrasos en el registro de las fichas.

Las empresas han desarrollado sus sistemas de Inventario en función de las exigencias del servicio, aprovechando sus propias experiencias y con el doble objetivo de garantizar la exactitud del recuento y de reducir al mínimo los inconvenientes de su realización.

En cualquier caso es necesario chequear la información que contiene el ordenador o las fichas, a fin de detectar posibles errores en el tratamiento informático o bien equivocaciones en las operaciones de manipulación en las entradas y salidas de los productos. Otro objeto de importancia es detectar la comisión de robos, pérdidas y roturas o mermas.

Los problemas de inventario se han estudiado desde hace mucho tiempo, pero fue hasta principios del presente siglo, en que, estimulados por el crecimiento de la industria manufacturera y de las diversas ramas de la ingeniería, los investigadores empezaron a utilizar técnicas analíticas para estudiar estos problemas.

2.3.2. El Inventario en la Empresa

Almacenar cosas implica incurrir en costos, ya que mientras éstas duren almacenadas, su valor será improductivo. Si nos encontrásemos en una economía en la cual la demanda y los costos futuros permanecieran constantes, la utilidad que reportaría el inventario sería mínima, ya que su función solamente sería la de compensar las fluctuaciones en la producción y el consumo. Lo anterior permite entender el porqué de la existencia del inventario, pero no es lo único; se pueden citar otros motivos que también son determinantes de la existencia de estas inversiones de capital inmovilizado.

a) La probabilidad de obtener utilidades futuras motivadas por un aumento en el precio del artículo, de la demanda o de los costos.
b) Como medida de seguridad para hacer frente a un incremento excepcional de la demanda.
c) Para disminuir los costos de transporte, ya que las compras realizadas en volúmenes importantes es posible efectuarlas a precios inferiores a los prevalecientes en pequeñas adquisiciones, lo que se refleja en la misma forma con los servicios de transporte.
d) Para asegurar la continuidad de las operaciones de producción ante la probabilidad de que sean interrumpidas por elementos externos al negocio.
e) La conveniencia de mantener artículos a la vista de los clientes con el propósito de incrementar las ventas, mediante la motivación de los compradores potenciales.
f) La acumulación de artículos originada por una disminución repentina en las ventas.

El inventario constituye un tipo de activo industrial que significa disminución de la inversión en otros, como pudiera ser en una empresa, la efectuada en la adquisición de maquinaria y equipo. El inventario es tan importante en la mayoría de los casos para la operación de un sistema de producción- distribución, como son la maquinaria, la planta y el equipo de transporte. De aquí que la planeación y el control del inventario sean factores tan importantes como lo son los planes para otros tipos de activos, cuando están orientados a incrementar la productividad de la empresa.

La importancia del inventario ha aumentado, fundamentalmente, por los avances de las técnicas de producción, que han permitido obtener nuevos productos y acelerar el ritmo de producción, lo cual ha traído como consecuencia, la necesidad de efectuar mayores inversiones en el inventario y en los diverso canales de distribución. De igual manera, la competencia prevaleciente entre los productores, ha conducido a la fabricación de una gran variedad de artículos, significando esto, la necesidad de enfrentarse a problemas de programación de producción e inventario cuya complejidad aumenta proporcionalmente al incremento en los volúmenes y variedades de los artículos. Son estas las razones que mayor influencia han tenido para hacer del inventario un problema que ha merecido la atención de especialistas e investigadores de alto prestigio en todo el mundo.

Se puede concluir diciendo que el inventario desempeña un papel crucial en la organización de una empresa, ya que hace las veces de regulador de la producción ante los efectos de las fluctuaciones de la actividad económica, contrarresta los errores en las estimulaciones de las ventas futuras, permite un mejor aprovechamiento de la fuerza de trabajo y del equipo de producción, etc. por consiguiente, el inventario constituye una parte indispensable en una empresa, que cuando es controlado eficientemente, se convierte en uno más de los factores productivos de la misma.

2.3.3. Tipos y costos del Inventario

En una empresa, los inventarios pueden existir en diversas formas: como existencia de materia prima en espera de entrar a la producción; como artículos parcialmente acabados o componentes; como inventarios de productos terminados; como bienes de tránsito, o como bienes en las tiendas de los consumidores.

En cada una de estas etapas, hay razones económicas que justifican su existencia. Estos diferentes tipos de inventarios pueden clasificarse en:

a) Inventarios de partida.- Este tipo de inventarios es el que se forma para fabricaciones o ventas de mercancía en volúmenes superiores a las necesidades normales, con el propósito de obtener rebajas en los precios de compra, y transferir o reducir los costos ajenos de demanda, transporte, pago y control.

b) Inventarios de seguridad.- Estos inventarios se mantienen para hacer frente a las variaciones inesperadas en la demanda, permitiendo suavizar sus efectos. Cubren las variaciones entre la demanda real con relación a la pronosticada, de la producción real y la planeada, y de las modificaciones en los tiempos de espera de abastecimiento.

c) Inventarios de anticipación.- Son aquéllas existencias que se obtienen debido a que es la única ocasión en que se encuentran disponibles. Es el caso de las industrias de conservas o las que producen artículos relacionados con la moda.

2.3.3.1. Elección y Clasificación del inventario

El esfuerzo requerido para controlar el inventario, debe variar en proporción a la utilidad potencial que representa para la empresa, cada uno de los artículos que se encuentran en él. Teóricamente podríamos analizar en detalle cada unidad en inventario y estructurar una técnica de control adecuada a las características particulares de la mercancía.

En la mayoría de las aplicaciones, el esfuerzo requerido para realizar este trabajo, posiblemente no sea compensado por los beneficios obtenidos. Por otro lado, una técnica de control de inventario que comprenda todas las unidades existentes en el inventario, resulta muy difícil, ya que significa un trabajo enorme el determinar la importancia de cada artículo.

2.3.3.2. El Inventario Tradicional

El inventario "tradicional" consiste en el "cierre del almacén" durante el tiempo necesario para efectuar un recuento total de las existencias. Exige una interrupción de las operaciones de entrada y salida así como un control de los últimos movimientos anteriores al inventario para garantizar el "corte" de operaciones administrativas que se incluyen en las Cuentas de Compras y Ventas. Hay que tener cuidado de que la mercancía que se recuenta en almacén sea debidamente contabilizada en compras y de que las últimas salidas anteriores al inventario sean contabilizadas en Ventas o en Devoluciones de Compras.

El resultado del Inventario se tiene que reflejar en un soporte material, bien un listado o en las fichas para posteriormente poder efectuar la valoración individualizada y la suma de esos importes. Es conveniente un desglose del inventario por Grupos-Familias, etc.

2.3.3.3. El Inventario Rotativo

El inventario tradicional es de difícil ejecución en Empresas con gran número de artículos o con fuerte exigencia de continuidad en el servicio de preparación de pedidos. Además las necesidades de personal para llevarlo a cabo se concentran en un periodo de tiempo muy corto.

Por esas razones se ha desarrollado un procedimiento alternativo llamado de inventario "Rotativo". Consiste en el recuento sistemático de las existencias durante todo el ejercicio de acuerdo con un Plan que permite distribuir la carga de trabajo a lo largo del año.

La representación gráfica de la lista de artículos vendidos, en relación a los artículos en el inventario, expresados ambos en porcentajes, nos proporciona la curva de la figura 2.3.3.3.1.

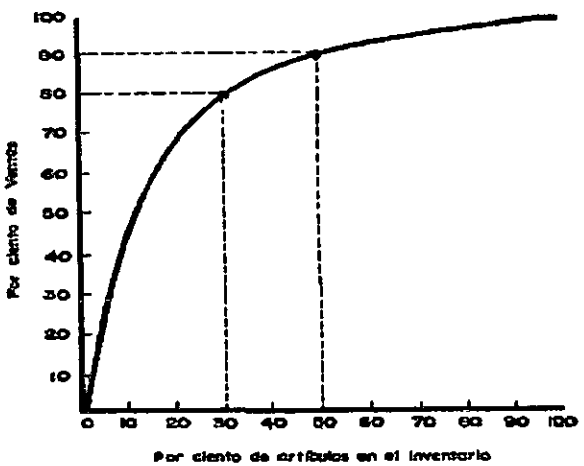


Figura 2.3.3.3.1

Esta curva nos indica que el 30% de los artículos en el inventario constituyen el 80% de las ventas; ello nos hace pensar que en muchas empresas no debemos extender el control más allá del 80%, ya que, posiblemente, el intento de controlar un 20% más de los artículos en el inventario, signifique efectuar demasiados refinamientos al sistema que no vayan a compensar los gastos necesarios en trabajo y equipo en que haya de incurrirse para alcanzar tal propósito.

Basados en esta propiedad que tienen los artículos en el inventario, han surgido varios sistemas de clasificación. Una de esas técnicas consiste en clasificar los artículos en tres categorías, designadas A, B, C. Este método tiene como objetivo clasificar las partidas del inventario en función de los siguientes parámetros:

1. Importancia relativa de las ventas totales.
2. Costo unitario.
3. Problemas de almacenaje.
4. Naturaleza crítica (costo de carecer de existencias y frecuencias del mismo).

Con base en los criterios anteriores, podemos formar categorías de artículos, como podrían ser:

A. Los artículos que constituyen del 10 al 25% de la contribución a las ventas.
B. Los siguientes 20 a 30 % de los artículos que contribuyen a menor proporción, y
C. El grupo más grande restante de artículos que contribuyen en un porcentaje relativamente pequeño.

Posteriormente se determinan rangos en función de las ventas, a los cuales se les designa con las letras A; B, C. Como ejemplo de esta clasificación se presenta el siguiente cuadro:

Clase del artículo	Número de unidades	Porcentaje del total de artículos	Porcentaje de las ventas	Porcentaje del promedio del inventario	Porcentaje de la contribución a la utilidad bruta total
A	229	7	53	50	52
B	1,225	35	37	38	38
C	2,062	58	10	12	10
	3,516	100%	100%	100%	100%

Las partidas de la categoría B necesitan un control general, por lo que será indispensable llevar:

1. Los registros perpetuos, actualizados con frecuencia.
2. Control por grupo de partidas.
3. Propagación y manejo rápido.
4. Revisión periódica de sus factores.
5. Aplicación de técnicas generales de control para equilibrar sus costos.

Los artículos pertenecientes al grupo C requieren de un control mínimo y por consiguiente

Solo es necesario mantener:

1. Registros perpetuos, al mínimo necesario.
2. Minimizar la frecuencia de carecer de inventario y de las órdenes de compra.
3. Revisión anual de los factores.

2.3.3.4. Costo del Inventario

El inventario representa, para el contador y una gran mayoría de hombres de empresa, La suma total de los gastos indirectos o bien, los artículos que van a la línea de producción a ser consumidos. Debido a que el inventario está constituido por muchos productos, frecuentemente se piensa que la suma de los costos de éstos, representan los costos totales del inventario. Sin embargo, esto no es así.

Reflexionando un momento, nos podemos dar cuenta de la existencia del inventario como una entidad y no como una acumulación de los costos de todos los artículos que contiene, ya que el inventario, por sí solo, tiene sus costos. Estos costos no pueden obtenerse de los sistemas tradicionales de contabilidad, sino que normalmente es necesario efectuar investigaciones especiales.

Los costos anteriores los podemos resumir como sigue:

a) Costos del pedido o costo de montaje.
b) Costo de mantenimiento.
c) Costo de carecer de existencias.
d) Costo del sistema.

La definición, determinación, control y equilibrio de estos costos, constituye la base para una administración efectiva del inventario.

a) Costo del pedido o costo de montaje.

Cada vez que se hace una requisición de compra para aumentar el inventario, cuando se describe la orden para un producto, o para una nueva tanda de producción, se incurre en costos.

El costo del pedido puede definirse como aquellos gastos necesarios para comprar un nuevo lote de material. El costo de montaje se refiere al gasto incurrido cuando se coloca una orden de producción o de instalación de operaciones en una máquina.

En la misma clasificación se deben considerar los descuentos ofrecidos por los vendedores, por adquisición de pedidos en grandes cantidades (solamente que se tomarán como costos negativos).

Los costos anteriores se pueden obtener de los registros contables de la empresa. Estas cifras pueden ser divididas por el número de órdenes de compra formuladas durante el año, para determinar el costo por orden de compra.

Los costos anteriores varían en relación directa al lote de producción, ya que el tamaño de lote determina la cantidad de equipo que se utilizará.

b) Costo de mantenimiento.

Antes de que se realicen las ventas de los artículos, se deben crear inventarios, lo que significa efectuar ciertos gastos, siendo algunos de ellos los siguientes:

i) Capital invertido en el inventario.
ii) Seguros.
iii) Impuestos.
iv) Obsolescencia y deterioro.
v) Almacenamiento y manejo.
vi) Mantenimiento de registros contables.

i) Capital invertido en el inventario.

Este factor constituye uno de los costos más importantes, el cual no implica gastos directos. Es decir, es un costo incurrido mediante la inversión en el inventario, en lugar de tener ese mismo monto de capital invertido en cualquier otra cosa, y es igual a la tasa de utilidades más grande que el sistema obtendría de otras inversiones.

Para determinar el costo de oportunidad del capital invertido, se debe tomar en cuenta la liquidez de la inversión y los riesgos de la misma, así como la redituabilidad que se obtendría si se realizara en cualquier otra cosa. Los intereses que se determinen, serán los que se agregarán a los demás factores que integran el costo de mantenimiento del inventario. Otra alternativa es la de aplicar la tasa de interés bancaria vigente.

Al aplicar las ideas anteriores, debe tenerse cuidado de analizar las siguientes características de la empresa:

- Fuentes de aprovisionamiento financieras (internas o externas).
- Otras alternativas de empleo del capital y sus utilidades respectivas.
- Tiempo necesario para recuperar la inversión en el inventario o grado de liquidez del mismo.
- Errores en las estimaciones de ventas.
- Grado de obsolescencia de los artículos en el inventario.

Todos estos elementos, valuados en forma realista, permitirán determinar el costo del capital inmovilizado en el inventario.

ii) Seguros.

Este elemento se refiere a la cantidad de dinero pagado para mantener protegidas las existencias de materiales y artículos terminados, en los almacenes. Estos seguros son contra imprevistos como incendios, tormentas, robo, etc. Los gastos originados por este concepto no varían estrictamente en forma proporcional a la inversión en el inventario. Se puede mantener una política constante de gastos en seguros, independientemente de las fluctuaciones del nivel de inventario, revisándola periódicamente para modificar dichos gastos en función de la magnitud de los cambios de un periodo a otro. Sin embargo, la forma en los costos de seguros cambiará con el nivel de inventario y varía de una empresa a otra.

iii) Impuestos.

Este factor se fija considerando los impuestos pagados sobre el capital, los bienes raíces y los impuestos del negocio, variando en función de los niveles de inventario.

iv) Obsolescencia y deterioro.

Es el costo de aquellas unidades que pueden estar sujetas a una disminución en su valor real, porque se hace obsoleta en un periodo determinado. Este "costo de obsolescencia" es la diferencia entre el costo original de la unidad (más alguna utilidad que se hubiera obtenido –desde la fecha en que se compró, hasta la fecha en que se convirtió en obsoleto- si ese capital, dedicado a procurarse la unidad o artículo, se hubiese invertido en cualquier otra cosa), y su costo de recuperación.

Se incurre en estos costos en una fecha fijada en el tiempo, que no se puede predecir con certeza.

Podemos fácilmente distinguir el costo de obsolescencia en aquellas empresas que operan dentro de ramas industriales, en los cuales se experimentan rápidos cambios, tanto en el material que utilizan como en la maquinaria, debido fundamentalmente a la competencia existente entre los fabricantes que los obliga a lanzar constantemente productos nuevos al mercado. Esto origina que los materiales, moldes y maquinaria, queden de un momento a otro obsoletos.

El riesgo de deterioro se presenta generalmente en las empresas dedicadas a la elaboración de artículos alimenticios y de drogas; por lo tanto, para calcular el costo correspondiente se debe determinar el tiempo de duración de los productos.

v) Almacenamiento y manejo.

El "costo de almacenamiento" se refiere al costo del espacio necesario para almacenar artículos a diferentes niveles. El ingreso derivado de los usos alternativos del espacio empleando para almacenar el inventario, se puede considerar como un "costo de oportunidad". Los costos de almacenamiento normalmente están relacionados en forma más directa al tamaño que al valor de los artículos comprados, incluyendo la renta, calefacción y luz, los cuales son frecuentemente fijos. Siempre hay un gasto relacionado con el espacio de almacenamiento, ya sea el pago de la renta cuando no es propio, o el costo de oportunidad en el caso de que lo sea.

vi) Mantenimiento de registros contables.

Este factor considera la cantidad de dinero que es necesario gastar en papelería, registro de las existencias, envío de cartas, revisiones periódicas del inventario, etc., para cada uno de los diferentes niveles de existencias.

Cuando se hace un estudio de los costos de mantenimiento del inventario, generalmente se determina un porcentaje anual que se aplica a cada unidad, o el porcentaje del promedio anual de la inversión en el inventario.

c) Costo de carecer de existencias

Uno de los objetivos más importantes en la mayoría de los sistemas de control de inventario, es el de mantener un servicio ajustado a la demanda del consumidor. La determinación del valor del servicio al consumidor o la pérdida de utilidad sufrida por no haber satisfecho las necesidades del consumidor, constituye un aspecto esencial de la derivación de un programa de minimización de costos o un sistema de control de inventario.

Los costos resultantes de tener un inventario inadecuado, son quizá los más difíciles de definir, teniendo una gran cantidad de interpretaciones. Algunas de ellas son las siguientes:

i. En una tienda, se incurriría en un costo por carecer de existencia, si un consumidor no encontrase el producto que busca, porque no está disponible en el momento. El costo de carecer de existencias será el de la pérdida de utilidad que se hubiera obtenido si ese artículo hubiera sido vendido.
ii. De acuerdo con la situación anterior, el costo de carecer de existencias también sería la pérdida futura de los márgenes de utilidad, si ese consumidor ya no regresara a comprar.
iii. Cuando un proceso de producción se detenga o interrumpa por la falta de una materia prima básica, el costo de carecer de existencias serían las horas de trabajo improductivas, resultantes de tal anomalía.
iv. El costo de carecer de existencias se reduciría en el caso anterior, si el programador de producción hiciese una nueva programación para fabricar otro artículo.

El costo de carecer de existencias puede determinarse, en el caso de un inventario de materias primas, calculando el costo de las demoras en la producción, cuando no haya existencias, y el costo de los trámites urgentes para surtir el inventario. Cuando se trate de un almacén de productos terminados, puede obtenerse el costo de carecer de existencias mediante el cálculo del costo por la pérdida de ventas y de clientes, originada por la mala reputación, generada por la imposibilidad de satisfacer la demanda de los clientes.

d) Costo del sistema

Cada uno de los sistemas de control de inventario implica gastos causados en función de la naturaleza del sistema de control seleccionado, los cuales se relacionan especialmente con el nivel del inventario.

El costo del sistema también está en función de la exactitud del control necesario, ya que si se desea que los resultados sean muy refinados, se pueden utilizar computadores electrónicos de alta capacidad. En caso de requerirse un sistema menos óptimo, se pueden utilizar técnicas periódicas de conteo o un sistema de kardex.

El costo de operación de un sistema de control de inventario, deberá confrontarse con los beneficios resultantes de aplicar métodos más refinados y con las necesidades de la organización de la empresa, así como con las consideraciones relativas a la disponibilidad de mejores y más rápidos informes de la estructura del sistema.

2.3.4 Modelos

Existe una gran variedad de modelos de inventario, los cuales van desde los más simples hasta los más complejos, en los que se requiere un nivel matemático elevado. En este caso se hará el desarrollo de los primeros, debido a que la teoría se puede aplicar a problemas reales, y el grado de complejidad de los segundos cae meramente dentro del campo de la investigación, siendo en la actualidad de más difícil aplicación a los problemas de nuestro país.

Los modelos de inventario han sido divididos en:

1) Estáticos.
2) Dinámicos.

1) Modelos de Inventario Estático.

Este tipo de modelo se refiere a aquellos casos en los cuales solamente se puede hacer un pedido u orden de fabricación para satisfacer la demanda, sin restricciones de tiempo, como podría ser, por ejemplo, el problema del vendedor de periódicos: esta persona tiene que decidir la cantidad de periódicos que debe tener en existencia para satisfacer la demanda de sus clientes potenciales, minimizando al mismo tiempo, el riesgo de mantener un inventario de periódicos mayor que el que pueda vender. En esta situación, el vendedor de periódicos no tiene ninguna oportunidad de equivocarse respecto al tamaño de la orden, porque no puede reabastecerse en caso de que las ventas excedan sus expectativas iniciales. De modo que su decisión es definitiva.

Las decisiones como la anterior se conocen como "estáticas", porque deben hacerse en un momento determinado y son irrevocables, ya que no es posible tomar alguna acción futura para corregir una orden errónea.

Los modelos de inventario estático, aunque prácticos y de interés, no son tan promisorios como los modelos dinámicos; no obstante, permiten introducirse al problema de determinar el inventario inicial de nuevos productos, así como a lograr una mejor comprensión de muchos de los problemas del inventario dinámico.

2) Modelos de Inventario Dinámico.

Estos modelos están relacionados con los artículos que tienen una demanda más o menos continua en el tiempo. En ocasiones, la constancia de la demanda llega a ser tal que los pedidos u órdenes de fabricación para reaprovisionar el inventario son una repetición de las cantidades anteriores.

Los artículos con esa característica se controlan más efectivamente en la práctica, ya que los modelos de demanda se hacen más estables, lo cual permite formular determinadas "reglas de decisión", con el objeto de mantener un control económico del inventario que sea idóneo con los objetivos de la empresa.

La forma de resolverlo es a través del análisis de los factores de costo y el número de órdenes de compra, de manera que los costos totales sean minimizados. El problema se puede reducir a expresar la relación entre cada costo y la magnitud del pedido; analizando las variaciones de estos costos a través de un rango de magnitudes de pedidos, podemos obtener la política óptima a seguir.

Una vez establecidas las relaciones entre costos y magnitudes de los pedidos, tendremos los elementos suficientes para construir un modelo que represente los principales aspectos que se presentan en la práctica y así poder formular las reglas de decisión.

La estructuración de un modelo que represente las condiciones de operación reales, no es una tarea fácil; no obstante, es posible obtener modelos útiles a través de métodos de aproximación que permiten considerar las variables críticas.

Para este tipo de modelos, las técnicas matemáticas no son muy complejas.

2.4. Calidad de Software

2.4.1. Introducción

En la actualidad el desarrollo de programas para computadora (software) se ha incrementado grandemente en México y en el mundo, debido al gran auge que ha tenido la computación. La mayoría de las empresas quieren utilizar a la computadora como una herramienta para realizar los procedimientos que efectúan actualmente en forma manual. Dichos procedimientos están siendo automatizados para ser ejecutados por una computadora, por lo que se requiere elaborar programas de computadoras para ellos, los cuales serán integrados en sistemas de información, con la intención de contar con información verídica y proporcionarla en el momento adecuado para la toma de decisiones y de esta manera ofrecer una mejor atención y servicio a sus clientes, aunque en la realidad los programas para computadora desarrollados presentan diferentes problemas.

A menudo los desarrolladores se enfrentan al problema de fabricar un software y antes de liberarlo presenta diferentes problemas, como pueden ser: que no cumple con la totalidad de los requerimientos del usuario; que no cumple con los requerimientos técnicos estipulados; que el software no resuelve todo lo que se quería originalmente; que el software no presenta la adecuada documentación e información para futuros mantenimientos; que el mantenimiento se vuelve muy difícil de realizarse, debido a que no se cuenta con los manuales técnicos respectivos; que no siguieron estándares de programación; que el software no se concluye en el tiempo establecido; que el software presenta tiempos de respuesta no adecuados; etc.

Ahora bien, por otro lado, puede suceder que el software desarrollado cumple con todos los requerimientos del usuario, tanto funcionales como técnicos, pero la forma como se liberó el sistema no fue la más adecuada, por ejemplo no se capacitó a las personas que lo utilizarían, o no se proporcionó toda la información necesaria para utilizarlo, etc., por lo que el sistema no es bien aceptado.

Todo lo anterior se detecta, surge o sale a la vista principalmente cuando se quiere liberar el sistema a producción, es entonces, cuando los desarrolladores se dan cuenta de la necesidad de contar con un mecanismo por medio del cual, se garantice o se asegure la calidad del software que se desarrolla, cumpliendo entre otras cosas con todos los requerimientos del usuario, tanto explícitos como implícitos; así como ir complementando todos aquellos procedimientos con que cuentan para desarrollar un software adecuado a las necesidades; con los estándares establecidos; con los procedimientos definidos; en los tiempos fijados, etc.

Hoy en día, existe una gran cantidad de enfoques para desarrollar software, los cuales son muy diversos, heterogéneos y están englobados todos dentro de la ingeniería de software. Así mismo el enfoque que se le da al aseguramiento de calidad en el desarrollo de software actualmente no es suficiente, por lo que se requiere que las empresas desarrolladoras de software tengan la confianza y seguridad de que los productos que elaboran satisfacen en primer lugar con todos los requisitos que desea el cliente y además cumplan con los requerimientos de calidad esperados.

Para las empresas cada vez es más necesario contar con sistemas de información, que de alguna manera les ayuden a dar un mejor servicio a sus clientes y sobre todo tener la confianza de que la información que están proporcionando los programas de cómputo sea la correcta, así como la integridad de la misma, es decir, que todas las actualizaciones y operaciones que hagan dichos programas sobre la información, queden hechas debidamente.

Las compañías esperan que los sistemas de información les proporcionen información verídica en el instante adecuado, por lo tanto se requiere que los programas desarrollados para sus sistemas de información tengan una alta calidad. Pero, ¿ qué se entiende que un programa de computadora desarrollado tenga calidad ?, por ejemplo en el caso de los bancos sería: que el software no tuviera problemas de fallas o interrupciones debidas a variables no tomadas en cuenta; que se cuente siempre con información verídica; que el programa de cómputo realice todas las operaciones correctamente y actualice todas sus bases de datos involucradas; que todos los programas de cómputo cuenten con el soporte de documentación necesario para el caso de falla, modificaciones o actualizaciones, se puedan hacer de una manera adecuada y correcta; así mismo se deberá actualizar la información que se modificó o aumentó, así como la adecuada publicación y distribución necesaria entre todo el personal involucrado para su correcto manejo, etc.

Como se sabe, para lograr calidad en un producto se necesitan controlar muchos aspectos como pueden ser: contar con la gente más capaz para desarrollar el software, se requiere que esta gente esté actualizada, capacitada, que sigan metodologías de desarrollo, planes de aseguramiento de calidad y sobre todo que sea profesional y ética.

2.4.2. Conceptos Generales

En la norma ISO 8402 (administración y aseguramiento de la calidad), los productos se clasifican en cuatro categorías genéricas:

1. Hardware. Un producto tangible con características distintas, por ejemplo, piezas, componentes, ensambles, etc.

2. Software. Una creación intelectual que consiste en información, expresada a través de medios de soporte, por ejemplo, programas de cómputo, procedimientos, información, datos, registros, etc.

3. Materiales procesados. Un producto tangible generado por la transformación de materias primas en un estado deseado, por ejemplo, materias primas, líquidos, sólidos, gases, alambres, etc.

4. Servicios. Es el resultado generado por actividades en la interrelación entre el proveedor y el cliente y por las actividades internas del proveedor para satisfacer las necesidades del cliente, por ejemplo, mantenimiento, garantía, etc.

Por otro lado, los sistemas de información, están compuestos de cinco componentes principales:

1. Un equipo físico o hardware, el cual a su vez está integrado por una unidad central de procesamiento y periféricos de entrada y salida

2. Un soporte lógico o software el cual está integrado por un conjunto de programas, documentación, lenguajes, etc. El software debe realizar una gestión de los datos, (es decir, creación, recuperación y actualización), un manejo de las comunicaciones y tratamientos específicos.

3. Un contenido de datos que puede ser factual o referencial.

4. Un administrador, su misión es asegurar la calidad y permitir el uso correcto y permanente de los datos memorizados.

5. Los usuarios, es decir, el grupo de personas que han de acceder al sistema de información, los cuales pueden ser informáticos o no informáticos.

Durante los primeros años de la informática, la calidad era responsabilidad únicamente del programador.

Actualmente, la responsabilidad del aseguramiento de calidad en el desarrollo de software, corresponde a todos los constituyentes de una organización que de una u otra manera participan en el desarrollo del mismo, entre los que podemos mencionar a: ingenieros de computación o de software, gestores del proyecto, clientes, desarrolladores, personas que trabajan dentro del grupo de aseguramiento de la calidad, etc.

2.4.3. Enfoques para el desarrollo del software

Existen varios enfoques para el desarrollo del software:

- Enfoque del ciclo de vida clásico para el desarrollo de software, según Pressman, Sommerville y Fairley.
- Enfoque del ciclo de vida estructurado para el desarrollo de software, según Yourdon.
- Enfoque de la construcción de prototipos para el desarrollo de software, según Pressman, Kendall & Kendall y Fairley.
- Enfoque de las técnicas de cuarta generación para el desarrollo de software, según Pressman y Mcclure.
- Enfoque de un proceso de desarrollo de software, según Pressman.
- Enfoque del ciclo de vida del software, según la Nasa.
- Enfoque del modelo en espiral para el desarrollo de software, según Sybase.

2.4.4. Normas y estándares de calidad para software

En esta sección se atenderán siete de las principales Normas, Estándares y/o Planes a considerar para el aseguramiento de la calidad del software:

IEEE estándar 730 (1984) y 983 (1986), Pressman (1995), Acis (1994), Butler (1995), Dobbins (1990), Stamm (1981) e ISO 9000-3 (1991).

A continuación se enumeran los puntos básicos de los que consta cada plan y las especificaciones que señalan las Normas o Estándares referidos

2.4.4.1. IEEE Estándar 730 Y 983

- Propósito del plan.
- Documentos de referencias.
- Gestión. Referida a organización y asignación de tareas y responsabilidades.
- Documentación. De requerimiento, diseño, verificación, gestión, desarrollo y procedimientos y estándares.
- Estándares, prácticas y convenciones.
- Revisiones y auditorias. Tipo de revisiones y auditorias y cómo, cuándo y quién las aplicará.
- Gestión de la configuración del software.
- Reporte de problemas y acciones correctivas.
- Herramientas, técnicas y metodologías.
- Control del código.
- Control de los medios físicos de almacenamiento del software.
- Control del distribuidor.
- Agrupación, mantenimiento y retención de registros.

Ventajas:

Es un formato estándar normalizado.

Desventajas:

Los puntos de documentación, gestión de la configuración del software, control del código, control de los medios físicos del almacenamiento, podían ser integrados en uno sólo que sería, por ejemplo, configuración del software; Además le hace falta algunos otros, como por ejemplo, pruebas. Asimismo, menciona de manera muy general los puntos que conviene abarcar, pero omite el decir cómo se pueden implementar y llevarlos a la práctica en alguna organización.

2.4.4.2. Pressman

El plan se basa en siete actividades básicas, que son:

- Aplicación de métodos técnicos.
- Realización de revisiones técnicas formales.
- Prueba del software.
- Ajuste a los estándares.
- Control de cambios.
- Mediciones.
- Registro y realización de informes.

Ventajas:

Presenta información formal y exhaustiva sobre la Ingeniería de Software.

Desventajas:

La información se presenta desde el punto de vista de la ingeniería de software y no como parte de un aseguramiento de la calidad.

2.4.4.3. ACIS

- Introducción.
- Aplicabilidad.
- Documentos aplicables.
- Gestión y planeación del programa.
- Programa de monitoreo de recursos.
- Programa de auditorias SQA.
- Registros SQA.
- Reportes de estado.
- Documentación de software.
- Definición de requerimientos.
- Procesos de desarrollo del software.
- Revisiones del proyecto.
- Herramientas y técnicas.
- Gestión de la configuración del software.
- Liberación de procesos.

- Control de cambios.
- Reporte de problemas.
- Pruebas del Software.

Ventajas:

Hace énfasis en las técnicas, procedimientos y metodologías que se deberán usar para asegurar la entrega a tiempo del software y que cumplen con los requerimientos especificados.

Desventajas:

Los puntos de documentación y control de cambios podían estar dentro de la gestión de configuración del software. Además menciona los puntos que se deben tomar en consideración en forma general, pero no dice cómo implementarlos.

2.4.4.4. Bntler

- Historia de cambios.
- Introducción
- Gestión.
- Documentación.
- Estándares y convenciones.
- Revisiones y auditorias.
- Reportes de problemas y acciones de corrección.
- Herramientas técnicas y metodologías.
- Colección de registros.

Ventajas:

Toma en consideración el punto de la versión actual y de la historia de cambios de las anteriores versiones que se tienen del plan, ninguno de los otros enfoques lo consideran.

Desventajas:

No toma en consideración varios puntos importantes como por ejemplo: la gestión de configuración del software y la planeación. Además son tratados de manera muy general.

2.4.4.5. Dobbins

- Organización del aseguramiento de la calidad del software (SQA).
- Iniciación de las actividades del SQA.
- Planeación del SQA.
- Proceso de inspección del software.
- Inspección de documentos.
- Control de la configuración del software.
- Actividades de pruebas SQA.
- Procuramiento del aseguramiento de la calidad de software.
- Auditorias de calidad.
- Estándares y especificaciones.

Ventajas:

Hace mucho énfasis en que la calidad del software se obtiene planeando y construyendo con calidad y además planeando y realizando evaluaciones de la misma.

Desventajas:

No le da importancia a los métodos, herramientas y procedimientos, los cuales son vitales para el desarrollo del software, además los puntos son tratados de manera muy general.

2.4.4.6. STAMM

- Organización.
- Definición de requerimientos.
- Documentación.
- Metodología de ingeniería de software.
- Entrenamiento.
- Revisiones formales.
- Programa de pruebas.
- Gestión de configuración.

Ventajas:

Hace mucho énfasis en que el aseguramiento de la calidad del software es parte del trabajo de todos los que participan en la organización y además de que se debe aplicar el aseguramiento de la calidad en cada una de las etapas que integran el ciclo de vida de desarrollo del mismo. Así mismo le da mucha importancia al entrenamiento y capacitación del personal desarrollador para la certificación del mismo, el cual no es considerado en los otros.

Desventajas:

No le da importancia a las auditorias. Los puntos son tratados de manera muy general.

2.4.4.7. ISO 9000-3

- Estructura.
  - Responsabilidad directiva.
  - Sistema de calidad.
  - Auditorias internas.
  - Acciones correctivas.

- Actividades del ciclo de vida.
  - Revisión del contrato.
  - Especificación de requerimientos del cliente.
  - Planeación del desarrollo.
  - Planeación de la calidad.
  - Diseño e implementación.
  - Pruebas y validaciones.
  - Aceptación.
  - Replicación, liberación e instalación.
  - Mantenimiento.

- Actividades de soporte.
  - Gestión de la configuración.
  - Control de documentos.
  - Registros de calidad.
  - Medición.
  - Reglas, prácticas y convenciones.
  - Herramientas y técnicas.
  - Adquisición.
  - Inclusión de productos de software.
  - Entrenamiento.

Ventajas:

Proporciona una serie de lineamientos o estándares muy importantes para el aseguramiento de la calidad y gestión de la misma en el desarrollo, suministro y mantenimiento del software.

Desventajas:

No es en sí un plan, solamente estándares.

2.4.5. Consideraciones para aseguramiento de Calidad en el Software

Se pueden considerar cinco puntos substanciales para el aseguramiento de calidad del software, los cuales de una u otra manera, se encuentran contemplados en las Metodologías, Estándares, Normas y Procedimientos, tratados en el punto anterior; los cuales son:

Gestión

Abarcará todo lo relacionado con la planeación, organización, preparación, ejecución, evaluación y control de cada uno de los demás puntos importantes del plan propuesto.

Métodos, técnicas y herramientas

Contemplará algunos de los paradigmas de la Ingeniería de Software, es decir, los métodos, las técnicas y las herramientas que existen para desarrollar sistemas de información.

Estándares y Procedimientos

Abarcará los estándares y manuales de procedimientos que se deberán tener y seguir internamente en la organización para desarrollar el software.

Revisiones y Auditorías

Se refiere a cada etapa del desarrollo del software, las revisiones formales y auditorias que se deberán aplicar, conteniendo la planeación de cuándo se deben de llevar a cabo, quienes deberán participar, qué se revisará, formatos de revisiones y auditorías, formatos de reporte, firmas de legalización de documentación, etc.

Configuración del Software

La configuración del software consiste en identificar, organizar, mantener y controlar las modificaciones que sufre el software que construye un equipo de desarrollo y se aplica a lo largo de todo el proceso de ingeniería del software, por lo tanto, la gestión de la configuración del software abarcará varios puntos como: La identificación de las líneas bases, el control de la documentación (programas fuentes, programas de código, manuales técnicos, de usuarios, información de procedimientos, formatos de reportes, errores, fólder de desarrollo, etc.), el control de cambios, la actualización y control de las versiones, la publicación y repartición de las nuevas versiones, auditorias de la configuración, etc.

Finalmente cabe señalar que el aseguramiento de calidad del software, depende en gran medida de:

- La correcta definición de los requerimientos del software.
- Estándares especificados que definen un conjunto de criterios o procedimientos de desarrollo que guíen la forma correcta para la elaboración del software.
- La correcta interpretación de los requerimientos implícitos.
- Del trabajo de todos los que participan en la elaboración del software, incluyendo al usuario.
- Del cuidado que se le dé en cada una de las etapas del proceso de desarrollo del software.

2.5 Teoría de Redes

En este apartado se dará una breve introducción al tema de Redes abarcando los puntos más importantes en este rubro como son: Estándares, Protocolos y tecnologías; lo cual será una base para poder definir si el tipo de tecnología en red que tiene la empresa Softtek México es suficiente para la implantación del Sistema de Información, tomando en cuenta que éste operará a través de su red.

2.5.1 Introducción a Redes

Una red de computadoras consiste en una o más computadoras conectadas por un medio físico y que ejecutan un software que permite a las computadoras comunicarse entre sí.

En los primeros años de las redes las grandes compañías, incluyendo IBM, Honeywell y Digital Equipment Corporation, crearon su propio estándar de cómo las computadoras debían conectarse. Estos estándares describían los mecanismos necesarios para mover datos de una computadora a otra. Estos primeros estándares, sin embargo, no eran eternamente compatibles. Por ejemplo, las redes que se adherían al SNA(Systems Network Architecture) de IBM no podían comunicarse directamente con las redes usando el DNA(Digital Network Architecture) de DEC. En años posteriores, organizaciones de estándares, incluyendo la Organización Internacional de Estandarización(ISO) y el instituto de Ingenieros Eléctricos y Electrónica(IEEE), desarrollaron modelos que llegaron a ser globalmente reconocidos y aceptados como estándares para el diseño de cualquier red de computadoras.

2.5.2 Estándares de Redes

Modelo OSI

En 1984, la Organización Internacional de Estandarización (ISO) desarrolló un modelo llamado Modelo OSI (Open Systems Interconection, Interconexión de sistemas abiertos). El cual es usado para describir el uso de datos entre la conexión física de la red y la aplicación del usuario final. Este modelo es el mejor conocido y el más usado para describir los entornos de red (figura 2.5.1)

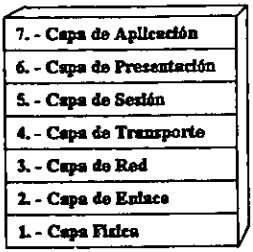


Fig. 2.5.1 Modelo OSI

Como se muestra las capas OSI están numeradas de abajo hacia arriba. Las funciones más básicas, como el poner los bits de datos en el cable de la red están en la parte de abajo, mientras las funciones que atienden los detalles de las aplicaciones del usuario están arriba.

Descripción de las capas del modelo OSI.

Capa Física. Este nivel es un conjunto de reglas sobre el hardware que se emplea para transmitir los datos. Comprende el medio físico de interconexión entre los equipos (conectores y cableado) ocupándose entre otras cosas, de la transmisión de bits a lo largo de un canal de comunicación y realizar transmisiones bidireccionales en forma simultánea.

Capa de Enlace. A partir de un medio físico define la estrategia de acceso para compartir el medio físico, encargándose de proveer el secuenciamiento, direccionamiento, recuperación de errores, etc.

Capa de Red. Especifica las operaciones de encaminamiento por la red, comunicación entre distintas redes, rutéo de los mensajes y notificaciones de errores.

Capa de Transporte. Esta capa es la encargada de proporcionar los últimos elementos para una entrega confiable de información entre dos puntos de la red, acepta los datos de la capa de sesión, los divide en unidades más pequeñas, los pasa al nivel de red y asegura que todos ellos lleguen correctamente al otro extremo.

Capa de Sesión. Esta capa Controla la conexión de dos nodos de la red encargándose principalmente de: la creación, mantenimiento y terminación de la sesión de red.

Capa de Presentación. Esta capa se encarga de la seguridad de la red, formato de los datos, traducción de la información así como de los servicios de conversión de protocolo, descompresión de datos, traducción, codificación.

Capa de Aplicación. Esa capa es la encargada de manejar mensajes y solicitudes de acceso remotas, es responsable de las estadísticas de la administración de la red y además, se encarga de atender al proceso de aplicación del usuario final. En esta capa se encuentran los programas como Administración de bases de datos y Correo electrónico.

Estándar IEEE (802)

Este otro modelo de red fue desarrollado por el mismo instituto de Ingenieros Eléctricos y Electrónica (IEEE, grupo de trabajo que desarrolla estándares para intercambio de información). Debido a la proliferación de Redes de Área Local(LAN) muchos productos aparecieron, y con ello la necesidad de una consistencia, entonces la IEEE empezó a definir estándares de red. El proyecto fue llamado 802, por el año y el mes en que empezó: Febrero de 1980.

El estándar 802 está integrado por 14 comités:

802.1 Interfaces de alto nivel HLI (High Level Interface). Incluye el sistema de direcciones, administración de las redes y puentes. Las funciones principales que realiza son: Sincronización lógica del transmisor y receptor, control de flujo de datos, control y detección de errores así como el secuenciamiento de los paquetes de datos para garantizar una entrega ordenada.

802.2 Control de Enlace Lógico LLC (Logical Link Control). El LLC genera paquetes o cuadros de comandos llamados PDU (unidades de datos de protocolo) o interpreta dichas PDU. Las responsabilidades del LLC en general son: Iniciación del intercambio de señales de control, Organización del flujo de datos y Funciones de control y recuperación de errores en el LLC.

802.3 Acceso Múltiple con Detección de Portadora y Detección de Colisiones CSMA/CD (Carrier Sense Multiple Acces/Collision Detection). La versión más extendida de éste método es la de la especificación Ethernet. Ethernet es el estándar más popular para las LAN que se usan actualmente. Ethernet emplea una topología lógica de bus (backbone) y una topología física de estrella o de bus. Antes de que un nodo envíe algún dato a través de una red Ethernet, primero se escucha y se da cuenta si algún otro nodo está transfiriendo información, de no ser así, el nodo transferirá la información a través de la red. Todos los nodos escucharán y el nodo seleccionado (destino) recibirá la información.

Thicknet (10BASE5). Thicknet también llamado Ethernet estándar, Thick Ethernet ó 10BASE5, fue el primer tipo de Ethernet que se diseñó. Tiene un estándar de topología física de bus que consiste en un segmento de cable de red con terminadores en los extremos. El cable empleado por Thicknet es un cable coaxial grueso (Baseband) a una velocidad de transmisión de 10 Mbps a una longitud máxima de segmento de red es de 500m y una cantidad máxima de nodos en una red de 1024.

Thinnet (10BASE2). Thinnet conocido también como 10BASE2, Thin Coax y Thin Ethernet. Thinnet se instala por medio de una topología física de bus, que consiste en segmentos de cables de red con terminaciones en cada extremo. Utiliza un cable coaxial delgado a una velocidad de transmisión de 10 Mbps a una longitud máxima del segmento debe ser de 185m y una cantidad máxima de nodos en una red de 1024

Par Trenzado (10BASE-T). El estándar 10BASE-T también se le conoce como UTP (par trenzado sin blindaje) o par trenzado. El 10BASE-T se instala por medio de una topología física de estrella, operando a 10 Mbps, utiliza conectores RJ45 y cable UTP, sin el uso de puentes el cable UTP puede acomodar un máximo de 1024 estaciones de trabajo.

Fibra Óptica (10BASE-F). El sistema 10BASE-F es el estándar de medio de fibra óptica para redes Ethernet, es el más frecuente usado actualmente para cubrir largas distancias. La mejor ventaja que da un segmento de enlace de fibra óptica 10BASE-F es que puede soportar velocidades de transmisión mucho más altas que 10 Mbps.

802.4 Token Bus. Se le conoce por lo general como paso de testigo en bus, utiliza una topología de bus físico o tipo árbol, al cual se conectan las estaciones. Estas, lógicamente están organizadas en un anillo, en la que cada una de las estaciones conoce la dirección de la estación ubicada a su izquierda y a su derecha. A cada nodo se le asignan posiciones lógicas en secuencia ordenada. Cuando el anillo se inicia, la estación que tiene el número mayor es la que puede enviar la primera trama. Después de que ésta lo hizo, pasa la autorización a su vecino inmediato, mediante una trama de control especial llamada token (o señal) para que éste a su vez pueda transmitir información. El token se propaga alrededor del anillo lógico, de tal forma que su poseedor esté autorizado para transmitir tramas. Como solamente una estación puede tener el token a la vez, no hay posibilidad de colisiones.

802.5 Token Ring. Token Ring emplea una topología lógica de anillo y una topología física de estrella. Los nodos se cablean a una unidad de acceso múltiple central (concentrador MAU Meda Access Unit) que repite las señales se una estación a la siguiente. Este tipo de red es una concatenación de enlaces punto a punto, donde cada estación actúa como un repetidor. Token Ring se basa en un esquema de paso de señales (token passing), es decir que pasa un token (o señal) a todas las computadoras de la red. Se puede pensar en un token como en una forma de obtener acceso a la red. La computadora que esté es posesión del token tiene autorización para transmitir su información a otra computadora de la red. Cuando termina, el token pasa a la siguiente computadora del anillo, si la siguiente computadora tiene que enviar información, acepta el token y procede a enviarla, en caso contrario, el token pasa a la siguiente computadora del anillo y el proceso continúa.

802.6 Redes de Área Metropolitana MAN (Metropolitan Area Networks). La norma 802.6 para redes MAN define un protocolo de alta velocidad en la cual las estaciones enlazadas comparten un bus doble de fibra óptica que utiliza un método de acceso llamado Bus Dual de Cola Distribuida (DQDB: Distributed Queue Dual Bus). La norma MAN se designa para proporcionar servicios de voz, datos y vídeo en una área metropolitana de aproximadamente 50 Km. con una velocidad de transmisión de datos hasta 155 Mbps. Debido al DQDB es una red de transmisión de celdas conmutadas con una longitud fija de 53 bytes, es compatible con la ISDN y el Modo de Transferencia Asíncrono (ATM: Asynchronous Transfer Mode).

802.7 Broadband Technical Advisory Group. Esta norma proporciona consejos técnicos a otros subcomités en técnicas de conexión de redes de banda ancha.

802.8 Fiber Optic Technical Advisory Group. Grupo que ofrece consejo a otros subcomités de redes que utilizan fibra óptica como una alternativa a las redes actuales basadas en cables de cobre.

802.9 Integrated Data and Voice Networks (Redes integradas por voz/vídeo).
Este grupo trabaja en la integración de tráfico de voz, datos y vídeo en LAN 802 y en redes digitales de servicios integrados (ISDN: Integrated Services Digital Networks). Los nodos definidos en las especificaciones incluyen teléfonos, computadoras, además de codificadores/decodificadores (codecs) de vídeo. Las especificaciones se han llamado Integración de Datos y Voz, o IVD (Integrated Voice and Data). El servicio proporciona un flujo multiplexado que puede llevar información de datos y voz por los canales que conectan las dos estaciones sobre cables de par trenzado de cobre.

802.10 LAN Security. Este grupo trabaja en la definición de un modelo normalizado de seguridad que inter opera sobre distintas redes e incorpora métodos de autentificación y cifrado.

802.11 Redes Inalámbricas (Wireless LAN's). Este comité define las normas para redes inalámbricas, trabaja en la normalización de interfaces inalámbricas para redes informáticas, donde los usuarios se conectan a sistemas de computadoras que usan computadoras basadas en lápices, asistentes digitales personales (PDA: Personal Digital Assistants) y otros dispositivos portátiles.

802.12 100VG AnyLAN (LAN de Acceso de Prioridad por Demanda).
Este comité define una de las normas de Ethernet a 100 Mbps con el método de acceso de prioridad por demanda (DPAM: Demand Priority Access Method) propuesto por Hewlett-Packard (HP) y otros fabricantes. El cable especificado es de par trenzado de cuatro hilos de cobre. El DPAM utiliza un concentrador central para controlar el acceso al canal de comunicación compartido. Las prioridades están disponibles para soportar la distribución de la información multimedia en tiempo real.

802.14 Multimedia Working Group. Existe la comunidad de redes para cable de televisión que se ha desarrollado en los híbridos bidireccionales de fibra/coaxial que pueden soportar aplicaciones iguales al de vídeo, teleconferencia, telefonía y el acceso a Internet.

2.5.3. Arquitecturas

Ethernet

A finales de 1960, la universidad de Hawai desarrolló una red de área amplia(WAN, Red que se extiende a través de un área geográfica mayor a una LAN). La universidad necesitaba conectar varias computadoras que estaban esparcidas a través de su campus. La pieza principal en el diseño de la red fue llamado Carrier-Sense Multiple Access with Collision Detection (CSMA/CD). Carrier-Sense significa que la computadora escucha el cable de la red y espera hasta un periodo de silencio para poder mandar su mensaje. Multiple Access se refiere a que múltiples computadoras pueden estar conectadas en el mismo cable de red. Collision Detection es una protección contra mensajes chocando en el transito.

En 1972, Xerox Corporation creó el experimental Ethernet, y en 1975 introdujo el primer producto Ethernet. El Ethernet de Xerox fue tan exitoso que Xerox, Intel y Digital crearon un estándar para Ethernet de 10mbps. Este diseño fue la base de la especificación IEEE 802.3.Los tres cableados más comunes son Thinnet, Thicknet, y Twisted Pair (Par trenzado).

Conforme se han expandido las redes, tanto en el área física como en la cantidad de nodos que las conforman, los fabricantes deben producir nuevas tecnologías de red que resuelvan los problemas producidos por redes más grandes y por el tráfico aumentado por la red.

Entre las nuevas tecnologías existe Fast Ethernet y Gigabit Ethernet. Fast Ethernet, llamado también 100BASE-X, es una extensión del estándar Ethernet que opera a velocidades de 100 Mbps, un incremento diez veces mayor que el Ethernet estándar de 10 Mbps, cifra lo suficientemente importante como para asegurar su competitividad en el futuro. Opera, al igual que 10BASE-T, sobre cable de par trenzado y fibra óptica. Gigabit Ethernet opera a una velocidad de 1000 Mbps.

Token –Ring

La red Token-Ring es una implementación del estándar IEEE 802.5, en el cual se distingue más por su método de transmitir la información que por la forma en que se conectan las computadoras. El primer diseño de una red de Token-Ring es atribuido a E. E. Newhall en 1969. IBM publicó por primera vez su topología de Token-Ring en marzo de 1982 , cuando esta compañía presentó los papeles para el proyecto 802 del IEEE. IBM anunció un producto Token-Ring en 1984, y en 1985 éste llegó a ser un estándar de ANSI/IEEE a velocidades de 4 o 16 Mbps.

2.5.4. Topologías

La topología se refiere a la forma en que están interconectados los distintos equipos (nodos) de una red. Un nodo es un dispositivo activo conectado a la red, como una computadora o una impresora. Un nodo también puede ser dispositivo o equipo de la red como un concentrador, conmutador o un router.

Topologías de redes de computadoras:

- **Bus**

Cada nodo o en lace en la red está conectado a un medio único de comunicación central llamado bus. El bus tiene dos extremos, en los que se sitúan dos terminadores que hacen que la señal rebote continuamente a lo largo del canal. Los dispositivos se sitúan secuencialmente uno detras de otro a lo largo de todo el cable a través de dispositivos de enlace, de tal manera que mientras estos dispositivos se encuentran conectados al cable central la comunicación puede continuar abierta, independientemente de que tras éstos se halla una computadora o no. En el bus no existe ninguna unidad central de control de la transmisión, sino que cada uno de los dispositivos se encarga individualmente de

realizar este control. Es lo que se conoce como sistema de control distribuido, que consiste en: Cada uno de los dispositivos que existe en la red ha de conocer su identidad y ésta debe de ser única en todo el bus. Una vez que se encuentre el canal libre, depositará la señal en el bus y ésta comenzará a circular por él. El mensaje lleva necesariamente aparejado la identidad del destinatario, de tal forma que todos los dispositivos comprobarán si va dirigido a ellos, tomándolo o dejándolo circular otra vez por el bus. El equipo al que vaya dirigido finalmente comprobará que es el receptor del mensaje y lo eliminará del bus. Este sistema, basado en la técnica de "escuchar" el canal antes de transmitir, se conoce como *CSMA* (Carrier Sense Múltiple Access).

- **Jerárquica o de Árbol**

La topología de árbol no es más que un *plus ultra* de la tecnología de bus. Técnicamente consiste en un bus central al que se conectan otros buses secundarios en los que se sitúan los distintos dispositivos que forman parte de la red. Es una estructura determinada por la unión en un único bus central de varias subredes en forma de buses secundarios. Una red jerárquica representa una red completamente distribuida en la que computadoras alimentan de información a otras computadoras, que a su vez alimentan a otras. Las computadoras que se utilizan como dispositivos remotos pueden tener recursos de procesamiento independientes y recurren a los recursos en niveles superiores o inferiores conforme se requiere información u otros recursos.

- **Estrella**

El diseño es relativamente simple para una red de computadoras. Consta de una Unidad Central de Procesamiento (UCP) que controla el flujo de información a través de la red hacia y desde cada dispositivo del sistema (nodos). El tamaño de la red se controla por intermedio del poder de la UCP central. El núcleo central de la red se denomina *hub* o *concentrador* y adquiere una importancia determinante dado que todo el funcionamiento de la red depende de él. Los hubs disponen de una serie de conexiones que permiten unir a ellos un número determinado de dispositivos, lo cual limita su crecimiento hasta la adquisición de nuevos concentradores capaces de unirse entre sí que amplíen la capacidad de la red. El auge de esta topología viene dado por la estandarización del sistema que se ha producido en todo el mundo y por la variada oferta de productos destinados a este mercado, así como por la comodidad de mantenimiento y detección de problemas. Así, puede ocurrir que una trama tenga una falla cualquiera y que ésta no afecte al resto de los dispositivos que forman la red, dado que la conexión entre éstos y el hub permanecen en funcionamiento. Esta es la estructura más simple de diseño de una red. La desventaja principal radica en las limitaciones en cuanto a rendimiento y confiabilidad generales.

- **Anillo**

La topología de anillo se llama así por el aspecto circular del flujo de datos. En la mayoría de los casos, los datos fluyen en una sola dirección y cada estación recibe la señal y la transmite a la siguiente del anillo. La organización en anillo resulta atractiva porque con ella son bastante raros los embotellamientos, tan frecuentes en los sistemas de estrella o en árbol. Además, la lógica necesaria para poner en marcha una red de este tipo es relativamente simple. Cada componente sólo ha de llevar a cabo una serie de tareas muy sencillas: aceptar los datos, ponerlos en el canal de comunicación o retransmitirlos al próximo componente del sistema. Sin embargo, como todas las redes, esta topología tiene algunos defectos. El problema más importante es que todos los componentes del anillo están unidos por un mismo canal, así si el canal falla entre nodos, toda la red se interrumpe, por eso algunos fabricantes han ideado diseños especiales que incluyen canales de seguridad, por si se produce la pérdida de algún canal, otros fabricantes construyen conmutadores que redirigen los datos automáticamente, saltándose el nodo averiado, hasta el siguiente nodo del anillo, con el fin de evitar que el fallo afecte a toda la red.

- **Malla**

Esta topología se ha venido empleando en los últimos años. Lo que la hace atractiva es su relativa inmunidad a los problemas de embotellamiento y averías. Gracias a la multiplicidad de caminos que ofrece a través de los distintos nodos de la red, es posible orientar el tráfico para trayectorias alternativas en caso de que algún nodo esté averiado u

ocupado. A pesar de que la realización de este método es compleja y cara (para proporcionar estas funciones especiales, la lógica de control de los protocolos de una red en malla puede llegar a ser sumamente complicada), muchos usuarios prefieren la fiabilidad de una red en malla a otras alternativas.

2.5.5 Protocolos

Se podría definir protocolo como el conjunto de normas que regulan la comunicación entre los distintos dispositivos de una red. Los protocolos están presentes en todas las etapas necesarias para establecer una comunicación entre equipos de cómputo, desde aquellas de más bajo nivel (p. ej. la transmisión de flujos de bits a un medio físico) hasta aquellas de más alto nivel (p. ej. el compartir o transferir información desde una computadora a otra en la red).

Los protocolos de se clasifican básicamente en dos categorías:

Protocolos de bajo nivel

Protocolos de alto nivel o protocolos de red.

2.5.5.1 Protocolos de bajo nivel

Los protocolos de bajo nivel controlan el acceso al medio físico, lo que se conoce como *MAC(Media Access Control)*, y además, parte del nivel de Enlace del modelo OSI de transmisión de datos, ya que se encargan también de las señales de temporización de la transmisión.

Sobre todos los protocolos de bajo nivel que corresponden a la capa *MAC*, se asientan los protocolos de control lógico del enlace o *LLC (Logical Link Control)*, definidos en el estándar IEEE 802.2.

A continuación se enumeran los protocolos de bajo nivel pertenecientes al nivel MAC y nivel LLC:

Nivel MAC

- **Ethernet.** El protocolo de red Ethernet es uno de los protocolos mas utilizadas actualmente. (estándar 802.3).
- **Token ring.** Las redes de tipo token ring tienen una topología en anillo y están definidas en la especificación IEEE 802.5. Las redes basadas en protocolos de paso de testigo (*token passing*) basan el control de acceso al medio en la posesión de un testigo.
- **Token bus.** Es una especificación de red basada en control de acceso al medio por paso de testigo con topología de bus.
- **FDDI. Fiber Distributed Data Interface.** Es una especificación de red sobre fibra óptica con topología de doble anillo, control de acceso al medio por paso de testigo.
- **CDDI.** Es una modificación de la especificación FDDI para permitir el uso de cables de cobre de la llamada categoría cinco, cables de alta calidad específicos para transmisión de datos, en lugar de fibra óptica.

Nivel LLC

- **HDLC.** Es la especificación de red utilizada principalmente en las transmisiones por líneas telefónicas para comunicaciones de datos, como pueden ser las líneas punto a punto y las redes públicas de conmutación de paquetes.

Frame Relay. *Frame Relay* (Paso de tramas) puede ser tanto un servicio prestado por una compañía telefónica como una especificación de red privada. Este sistema de transmisión permite velocidades de 56 kbps, 64 kbps, 2 Mbps, y superiores. El servicio se puede establecer con líneas punto a punto entre ruteadores o por medio de una conexión con una red pública.

Un parámetro básico del servicio Frame Relay es el *CIR (Commited Information Rate, Tasa de información asegurada)*, el cual se utiliza para facturar las conexiones a redes públicas. Este valor se basa en la naturaleza aleatoria de la transmisión de datos, ya que no todas las estaciones transmiten al mismo tiempo, con lo cual, la suma de la capacidad, en bits/s, de los canales de cada una de ellas, puede ser superior a la capacidad de los canales de interconexión. Cada estación puede transmitir toda la información que permita el canal, pero, en caso de que la red se congestione, sólo podrá transmitir, en principio, la cantidad permitida por el CIR.

ATM. *Asynchronous Transfer Mode*(Modo de transferencia asíncrono). Es la especificación más reciente y con mayor futuro. Permite velocidades de a partir de 156 Mbps llegando a superar los 560 Mbps. Se basa en la transmisión de pequeños paquetes de datos de 56 bytes, con una mínima cabecera de dirección que son conmutados por equipos de muy alta velocidad. La gran ventaja de esta especificación es la capacidad que tiene para transmitir información sensible a los retardos como pueden ser voz o imágenes digitalizadas combinada con datos, gracias a la capacidad de marcar los paquetes como "eliminadles", para que los equipos de conmutación puedan decidir que paquetes transmitir en caso de congestión de la red.

2.5.5.2 Protocolos de Red

El protocolo de red determina el modo y organización de la información (tanto datos como controles) para su transmisión por el medio físico con el protocolo de bajo nivel. Los protocolos de red más comunes son: IPX/SPX, DECnet, X.25. AppleTalk, NetBEUI y TCP/IP

IPX/SPX. *Internet Packet eXchange/Sequenced Packet eXchange.* Es el conjunto de protocolos de bajo nivel utilizados por el sistema operativo de red Netware de Novell. SPX actúa sobre IPX para asegurar la entrega de los datos.

DECnet. Es un protocolo de red propio de Digital Equipment Corporation (DEC), que se utiliza para las conexiones en red de los ordenadores y equipos de esta marca y sus compatibles. Está muy extendido en el mundo académico. Uno de sus componentes, LAT (*Local Area Transport*, transporte de área local), se utiliza para conectar periféricos por medio de la red y tiene una serie de características de gran utilidad como la asignación de nombres de servicio a periféricos o los servicios dedicados.

X.25. Es un protocolo utilizado principalmente en WAN y, sobre todo, en las redes públicas de transmisión de datos. Funciona por conmutación de paquetes, esto es, que los bloques de datos contienen información del origen y destino de los mismos para que la red los pueda entregar correctamente aunque cada uno circule por un camino diferente.

AppleTalk. Este protocolo está incluido en el sistema operativo del ordenador Apple Macintosh desde su aparición y permite interconectar ordenadores y periféricos con gran sencillez para el usuario, ya que no requiere ningún tipo de configuración por su parte, el sistema operativo se encarga de todo. Existen tres formas básicas de este protocolo: LocalTalk, Ethertalk y Tokentalk

NetBEUI. NetBIOS Extended User Interface (Interfaz de usuario extendido para NetBIOS). Es la versión de Microsoft del NetBIOS (Network Basic Input/Output System, sistema básico de entrada/salida de red), que es el sistema de enlazar el *software* y el *hardware* de red en los PC's. Este protocolo es la base de la red de Microsoft Windows para Trabajo en Grupo.

TCP/IP. Este no es un protocolo, si no un conjunto de protocolos, que toma su nombre de los dos más conocidos: TCP (*Transmission Control Protocol*, protocolo de control de transmisión) e IP (*Internet Protocol*). Esta familia de protocolos es la base de la red Internet, la mayor red de ordenadores del mundo. Por lo cual, se ha convertido en el más extendido.

Resumen

La empresa Softtek México cuenta con una red LAN Ethernet en una topología en estrella operando con el protocolo TCP/IP y conectividad WAN (Internet) a una velocidad de 1024 Kbps.

Considerando el tipo de red y de acuerdo al análisis de infraestructura en equipo de cómputo y comunicaciones con el que cuenta la empresa Softtek México (descrito en el capítulo 1.3), se llega a la conclusión de que el sistema puede ser implementado y operado por la empresa aprovechando las ventajas y características de la herramienta que se utilizará en el desarrollo (Java), permitiendo su operabilidad desde cualquier computadora y desde cualquier parte.

Capítulo 3

Análisis Preliminar

3. Análisis Preliminar

Para hacer el análisis y diseño del sistema, es necesario un análisis preliminar. En el cual se toman en cuenta todas las herramientas con que podemos contar para el diseño. Así como una comparación entre ellas, para aplicar el método y plataforma que más se adecue a las necesidades del sistema, y así obtener una propuesta de solución.

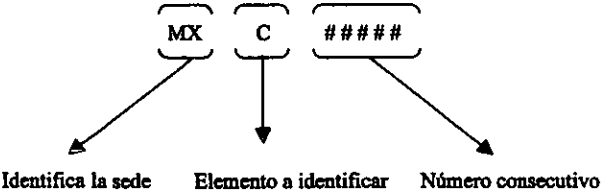
3.1. Planteamiento del Problema

El área de sistemas a partir del mes de junio del 2000 comenzó a encargarse de la administración y control de inventarios del equipo de cómputo, llevando a cabo el proceso de registro, asignación e inventarios, durante está administración se ha detectado el extravío tanto de equipos como algunas de sus partes y al no tener un control estricto de asignaciones del equipo así como el resguardo del mismo, no se puede saber con exactitud su localización, tampoco aquellos que se encuentran disponibles para asignaciones al personal, equipos en mantenimiento correctivo o preventivo, ni los equipos que se han dado de baja por desperfectos o aquellos que presenten garantía vigente y que se puedan mandar con los distribuidores.

Es necesario conocer con exactitud el activo con el que cuenta la empresa a nivel de tecnología, esto es todo el equipo de cómputo con el que cuenta, conociendo a detalle todos sus elementos como marcas, modelos, números de serie, etc. además de los componentes que los forman, como por ejemplo:

- CPU: Procesador, Memoria, Discos Duros, Tarjeta de Red, CD-ROM, tarjeta de audio, etc.
- Monitor: Tamaño y tipo de conexión de video
- Teclado: Conector
- Mouse: Conector
- Discos Duros: Capacidad
- Memorias: Tipo de tecnología (DIMM o SIMM)
- Impresoras: Tipo de impresión (inyección o láser)
- Tarjetas de red: Velocidad y tecnología (ISA o PCI)
- Otros: Periféricos adicionales (webcam, unidades zip, scanner, etc.)

Otro de los problemas actuales es que el inventario se registra en una hoja de Excel, sólo registrando en ella los cambio de equipo, partes, periféricos, ubicación y usuarios, complicando su administración por la gran cantidad de cambios que se tienen que registrar ya que es frecuente el traslado de equipo entre las instalaciones de Softtek y los proyectos, ubicados en las instalaciones de los clientes. No se tiene ninguna estrategia para el manejo del inventario, por lo que en ocasiones es necesario hacer modificaciones a las tablas de captura debido a la mal planeación del inventario. Actualmente el equipo ya está inventariado por medio de etiquetas con código de barras, pero sólo reflejado en hojas de Excel. La nomenclatura de los códigos es el siguiente:



Donde:

Sede		Elemento	
Argentina	AR	CPU	C
Brasil	BR	MONITOR	V
COLOMBIA	CO	TECLADO	K
ESPAÑA	ES	MOUSE	M
ESTADOS UNIDOS	EU	IMPRESORA	I
MÉXICO	MX	OTRO	O
PERU	PE		
PUERTO RICO	PR		
VENEZUELA	VE		

Otro punto importante es que no existe una responsabilidad directa de los usuarios por los equipos asignados, por lo que se tiene que hacer una responsiva por equipo para hacer responsables a los usuarios por los equipos que usan y tienen a su cargo, registrando:

- Nombre del usuario
- Sector o área
- Ubicación
- Proyecto
- Teléfono
- Equipos asignados

La mayor parte de los proyectos se desarrollan en las instalaciones del cliente, en los proyectos existe gran movimiento de personas durante todo el ciclo del proyecto por lo que intervienen diversos perfiles usuarios y es un problema mantener este control al día, por lo cual una forma de solucionarlo es acceder desde cualquier ubicación al inventario para hacer los cambios en línea y mantener actualizado el inventario.

Es así por lo que es necesario la implantación de un sistema de información para la empresa, con el propósito de facilitar el control en los procesos mencionados anteriormente y mantener actualizado el inventario de equipo de cómputo, por lo que se llevará a cabo en forma general el análisis y diseño del sistema, que comprenderá tanto el software y las bases de datos ha utilizar; el fin es automatizar el proceso y con esto se establecerían las bases y opciones necesarias para que posteriormente sea desarrollado el sistema, y así el personal de la empresa asignado a dicha labor pueda administrar y ejecutar sus funciones con mayor eficacia.

Por lo que se requiere de un control total sobre los componentes, equipo, facturas, mantenimiento, etc. Es decir, que se encuentren registrados sus características, velocidad, capacidad, descripción, marcas, etc. También con el propósito de lograr este control se debe tener acceso a los diferentes catálogos y con esto poder obtener, acceder y modificar la información obtenida en ellos, así como obtener los reportes de la información necesaria.

Actualmente ya se tiene almacenada la mayor parte de información en hojas de cálculo como base de datos y los campos necesarios de captura, los cuales se utilizarán en el diseño del sistema por la información que en ellos se encuentra, de donde se podrán obtener los catálogos de equipo y componente.

3.2. Análisis Comparativo de las Herramientas Comerciales de Desarrollo

Resulta fundamental que el SGBO posea distintos tipos de herramientas para desarrollo de aplicaciones, que pueden ir desde lenguajes de desarrollo hasta visualizadores (browser) que permiten una mejor manipulación de las bibliotecas de clase. Los lenguajes de cuarta generación (L4G) de los SGBO dejan en algunos casos bastante que desear, sobre todo si los comparamos con los actuales de los SGBDR, lo que tampoco es de extrañar dado que estos últimos llevan años desarrollándose y han alcanzado un mayor grado de madurez.

Los SGBO deben proporcionar herramientas que permitan transferir datos entre distintos tipos de SGBD pudiéndose volcar datos de un sistema SGBO a un sistema relacional o viceversa.

Al igual que para los SGBDR, en los SGBO deben existir utilidades que permitan monitorizar la utilización de recursos con el fin de aumentar el rendimiento de los sistemas y proceder a su ajuste.

El SGBO debe proporcionar distintas interfaces a los diferentes usuarios del sistema según sea la función que desempeñan en cada momento; por ejemplo. Administrador de bases de datos, programador de servicios, diseñador de clases, usuarios finales, etc.

En cuanto a las IGU (interfaces gráficas de usuario) que presentan los SGBO, es lógico que también sigan el paradigma de la orientación al objeto facilitando así la interacción del usuario con el sistema.

Como señala FRIEDMAN (1993) " Hasta hace poco, los desarrolladores de software tenían que elegir entre las capacidades singulares de los SGBDO y la madurez y estandarización de las ofertas de los SGBDR. Ahora, sin embargo están emergiendo más oportunidades para mezclar los puntos fuertes de las diferentes tecnologías".

Este autor considera tanto para los SGBO como para los SGBDR, dos aspectos: por un lado, el modelo lógico de datos soportado (relacional u orientado al objeto) y por otro, el nivel interno (la "maquinaria" que se ofrece, esto es lo que se concierne a la gestión de E/S, concurrencia, índices, transacciones, recuperación, caching, etc. De acuerdo a estos criterios se pueden clasificar los SGBD, de la siguiente manera.

SGBD relacional puros, en los que se encuentran la mayoría de los existentes en la actualidad: ORACE, INGRES, DB2, RDB, SYBASE, INFORMIX, INTERBASE, ETC.

SGBD relacionales con frontal OO, en los añade un nivel o capa de orientación al objeto sobre un SGBDR, que es el enfoque seguido por Hewlett Packard en su producto OpenOBD, en el que el nivel OO lleva a cabo todas las translaciones de los aspectos orientados al objeto del lenguaje a sus equivalentes relacionales (en el caso de OpenODB, los soportados por el SGBDR Allbase). En este caso la capa OO puede actuar con el SGBDR a dos niveles:

Gestor de datos, mediante llamadas SQL.

Gestor de almacenamiento, mediante llamadas a procedimientos de bajo nivel.

E problema de este enfoque es que al, no poder modificar normalmente el SGBDR para atender a las necesidades del nivel OO se puede dar problemas de rendimiento y operacionales (por ejemplo, bloqueos), aunque se tenga la ventaja de la posible portabilidad del nivel OO sobre varios de los SGBDR existentes en el mercado.

SGBO puros, como O2, Objectivity, ObjectStore, Ontos, Itasca, Versant, Jasmine, etc.

SGBO con SQL. Que siguen un enfoque unificador en el que se combinan la OO y el modelo relacional, realizándose la necesaria adaptación tanto a nivel de gestor de base de almacenamiento como de datos del SGBDR. Un sistema que sigue este enfoque es UniSQL/X de la empresa UniSQL.

Existen otros tipos de sistemas que poseen algunas características parecidas a los SGBO y que se catalogan a veces como tales, por ejemplo.

Lenguajes de programación orientados a objeto persistentes; que se diseñan para un usuario (o un número muy reducido) y pequeñas bases de datos, mientras que los SGBDO se diseñan pensando en un número elevado de usuarios que acceden concurrentemente a grandes bases de datos.

Gestores de objetos, que son extensiones de sistemas de ficheros o de gestión de memoria virtual y que poseen un modelo de datos limitado, como Mneme. Camelot, ObServer, LOOM. Etc.

"metasSGBD", que son generadores de sistemas de datos que permiten construir un SGBD adaptado a las características de un determinado grupo de aplicaciones. Ejemplos de este tipo son Genesis o EXODUS.

Hay que destacar que a partir de 1995 han aparecido nuevos tipos de SGBD en los que el modelo relacional se integra con un modelo de objetos (no es una capa superpuesta) y que se denominan "Object-Relational" (Relational-Object) como por ejemplo. Illustra.

A lo largo de esta sección se van a exponer los siguientes rubros: Métodos a utilizar, Bases de datos, y también Lenguajes y herramientas de desarrollo. El desglose de estos temas conforman el soporte del análisis comparativo de las herramientas comerciales de desarrollo para el sistema de información.

3.2.1. Método a utilizar

La metodología que se utilizará será formada por varios métodos de acuerdo a las características y facilidades de uso que nos sean más óptimos en el desarrollo de la propuesta.

Los métodos que se van a utilizar para el desarrollo de la propuesta son los siguientes:

Coad y Yourdon, OMT y UML

Coad y Yourdon.

La importancia fundamental del método Coad y Yourdon es que utiliza una descripción breve y concisa, así como el uso de textos generales como fuentes para las definiciones; de modo que las definiciones se enmarcan dentro del sentido común y reduce el empleo de modismos, la desventaja del método es su notación compleja, la cual es difícil de utilizar sin una herramienta, por lo que algunos usuarios del método han utilizado la notación de diagramación de OMT en su lugar.

OMT. La técnica de Modelado de Objetos (Object modelling technique). Los aspectos más importantes son sus capacidades simples aunque poderosas de notación, así como su madurez. Este ha sido aplicado en varios proyectos por sus autores antes de ser publicado.

Método UML (Unified Modeling Language)

El lenguaje unificado del modelo (UML) es un lenguaje estándar para desarrollar proyectos de software. El UML pude ser usado para visualizar, especificar, construir y documentar los componentes de un sistema de software.

El UML es apropiado para el modelado de sistemas de empresas con sistemas de información con aplicaciones distribuidas basadas en Web aún en desarrollos de sistemas en tiempo real.

3.2.2. Bases de datos

Las bases de datos orientadas a objetos están evolucionando como resultado de las actuales limitaciones del modelo relacional en su capacidad para soportar tipos de datos numerosos y complejos. Las extensiones orientadas a objetos para bases de datos relacionales existentes van a establecer el rumbo de las compañías de bases de datos. Las bases de datos orientadas a objetos no solamente complementan a los lenguajes orientados a objetos; también ofrecen ventajas sobre el modelo relacional soportando aplicaciones complejas, perfeccionando la programabilidad y el rendimiento, mejorando al acceso por navegación, simplificando el control de la concurrencia y reduciendo los riesgos de integridad referencial.

Los DBMS relacionales dominan el mercado actual. Los DBMS relacionales presentan la base de datos desde un nivel de abstracción fácil de utilizar. Las implementaciones de los RDBMS relacionales van mejorando de rendimiento a medida que maduran, y también con el uso de técnicas de optimización más inteligentes. Los DBMS orientados a objetos y/o los DBMS relacionales avanzados pueden ser el futuro.

Los conceptos orientados a objetos proporcionan una base excelente para modelar DBMS relacionales, y considerando las siguientes reglas que permiten transformar un modelo de objetos en tablas de un DBMS relacional:

- Las clases de objetos se corresponden con tablas.
- Las asociaciones se corresponden con tablas.
- Las generalizaciones se corresponden con una tabla de superclase más una serie de tablas de subclase.

Además de que en el mercado dominan los RDBMS, se procederá a realizar un estudio comparativo entre los principales DBMS relacionales comerciales.

Existen diferentes opciones de solución para poder implantar el sistema con distintos manejadores de bases de datos comerciales. A continuación se describen los siguientes cinco manejadores de bases de datos, los cuales son los más utilizados en las distintas plataformas de implantación.

Las opciones de manejadores de bases de datos presentadas son:

- SQL Server 6.5
- IBM DB/2 6.1
- Informix Online 7.2
- Oracle7 7.3
- Sybase SQL Server 11

3.2.2.1. SQL Server 6.5

- Miles de Soluciones Disponibles: Se tiene libertad de elección, ya que todas las aplicaciones de gestión del mercado corren sobre Microsoft SQL Server.
- Escalabilidad: Se adapta a las necesidades de la empresa, soportado desde unos pocos usuarios a varios miles. Empresas centralizadas u oficinas distribuidas, replicando cientos de sitios.
- Gestión: Con un completo interfaz gráfico que reduce la complejidad innecesaria de las tareas de administración y gestión de la base de datos.
- Orientada al desarrollo: Visual Basic, Visual C++, Visual J++, Visual Interdev, Microfocus Cobol y muchas otras herramientas son compatibles con Microsoft SQL Server.
- Plataforma de desarrollo fácil y abierta: integrada con las mejores tecnologías de Internet como ActiveX, ADC y Microsoft Transaction Server y con las mejores herramientas de gestión y desarrollo para Internet como FrontPage97, Microsoft Office97 y Visual Interdev.

- Diseñada para INTERNET: Es el único gestor de base de datos que contiene de forma integrada la posibilidad de generar contenido HTML de forma automática.
- En cuanto arquitectura RDBMS, tiene un completo proceso transaccional interactivo con rollback automático y recuperación de roll-forward.
- Maneja datos distribuidos, puede hacer llamadas a procedimientos remotos servidor-a-servidor (procedimientos almacenados remotos).
- Cuenta con la replicación de subscriptores ODBC, incluyendo IBM DB2, ORACLE, SYBASE y Microsoft Access.
- Maneja un sistema de dispositivos espejo con recuperación automática para tolerancia a fallos de dispositivos.
- En cuanto a programación, cuenta con triggers, procedimientos almacenados, disparador de eventos antes y después de conexiones.
- Para mejorar la seguridad y facilitar la administración de la base de datos, sólo maneja un usuario tanto para la red como para la base de datos.

3.2.2.2. IBM DB/2 6.1

- Incluye uno de los mejores conjuntos de herramientas de administración y de ajuste en el mercado, con un mecanismo de base de datos que se puede utilizar tanto en una pequeña laptop que trabaje con Windows 95 como en un grupo de mainframes S/390 que trabajen con OS/390.

Existen dos versiones de DB2: DB/2 Workgroup y DB/2 Enterprise Edition.

- DB/2 Workgroup: incluye la tecnología más avanzada para mecanismos de bases de datos, como el paralelismo dentro de una sola consulta, soporte completo para replicaciones, tablas de consulta de resúmenes para acelerar el desempeño de la base de datos, diseño relacional de objetos para la base de datos, y soporte para la programación Java. Sus únicas limitaciones en la licencia implican que no se puede emplear con máquinas que tengan más de cuatro CPU y no incluye DB/2 Connect, la compuerta de IBM para DB/2 en el mainframe.
- DB/2 tiene un conjunto completo de extensiones multimedia para almacenar y manipular información de texto, de sonido, de video, de imágenes y geográfica. Con estas extensiones es mucho más sencillo diseñar aplicaciones basadas en el Web y aplicaciones que incluyan fotografías o extensos reportes de texto. Son opciones que tiene un costo adicional, pero ofrecen características y precios que son muy competitivos frente a los de Informix y de Oracle.
- DB/2 es un producto competitivo como una plataforma de desarrollo de aplicaciones, ya que se puede utilizar Java para codificar la lógica de la base de datos, y una nueva herramienta de DB/2 6.1, Stored Procedure Builder, que convierte en forma automática una instrucción SQL en su equivalente de clase Java, y después se puede instalar en la base de datos apropiada.
- DB/2 6.1 incluye soporte integrado para OLE DB, el nuevo estándar de acceso a las bases de datos de Microsoft. Con este soporte, DB/2 puede conectarse y acceder al contenido (aunque no puede actualizar la información) de casi cualquier otra base de datos. Por ejemplo con un vínculo OLE DB podemos lograr que una tabla de Microsoft SQL Server tenga la apariencia de una tabla de DB/2.
- Las herramientas de administración de DB2, reescritas para Java y que permiten acceso a través del Web, son de primer nivel ya que posee una detallada utilería para la presentación gráfica de los planes de consulta, excelentes herramientas de monitoreo del desempeño, y un planificador de tareas que permite ejecutar trabajos a intervalos regulares.

3.2.2.3. Informix Online 7.2

- Opera bajo el sistema operativo UNIX, proporciona alto performance y alta disponibilidad de información. Incluye características como discos espejo y mecanismos ágiles de recuperación de información. Es por ello que con frecuencia se utiliza en aplicaciones de misión crítica.
- Puede trabajar bajo la plataforma cliente/servidor mediante el producto de conectividad Informix Net. Este soporta los protocolos TCP/IP y StarGroup, y una variedad de computadoras y hardware de comunicación de datos como Ethernet y Token Ring.
- Soporta bases de datos distribuidas mediante el producto de conectividad Informix Star y el manejador de base de datos OnLine. Mediante estos productos se puede consultar y modificar varias bases de datos Informix, en diferentes máquinas, como si fuera una sola base de datos centralizada.
- Cuenta con un conjunto de tablas para registrar el diccionario de datos. En el caso particular del manejador de bases de datos OnLine se tiene la posibilidad de guardar en forma histórica las modificaciones hechas al diccionario de datos.

3.2.2.4. Oracle 7 7.3

- Corre sobre más hardware, software y redes que cualquier otra base de datos relacional, lo que lo hace que sea un producto muy portable y que sea soportado por más del 70% de los back-ends y front-ends que hay en el mercado.
- Soporta el manejador de triggers y store procedures.
- Cuenta con opciones de seguridad para el manejo de la base de datos. Maneja los esquemas de discos espejos para poder continuar con los procesos en caso de que un disco se dañe.
- Cuenta con un log (registro) de transacciones, en donde se almacenan las modificaciones que se hagan a la base de datos.
- Oracle7 7.3 está diseñado para proveer todas las facilidades de almacenamiento y recuperación de información de diferentes formatos.
- Los datos almacenados son almacenados en una serie de archivos en los discos de la computadora en donde se encuentra corriendo Oracle.
- Para aumentar la velocidad de procesamiento y la consistencia de los datos, implementa el uso de grandes áreas de memoria para almacenar datos, además de que utiliza transacciones lo que hace que garantice la integridad de los datos, llevando un control de los registros que han sido modificados dentro de una transacción a fin de poder deshacer todos los cambios en caso de que por algún motivo el proceso no se termine completamente.
- Oracle7 7.3 es soportado por un gran número de lenguajes de cuarta generación, con lo cual se pueden desarrollar de forma rápida aplicaciones que impliquen un gran volumen de información.
- Existen en el mercado diversas herramientas para el manejo de Oracle, que facilitan las tareas de desarrollo de aplicaciones, por ejemplo existen herramientas de modelado de base de datos que generan completamente el código para la construcción de la base de datos.

3.2.2.5. Sybase SQL Server 11

- El SQL Server de Sybase es una arquitectura multihilos, con su propio kernel que tiene integrado un monitor de control de SQL.
- Cuenta con Transact SQL que es un conjunto de extensiones para ANSI estándar SQL que es usado para escribir stored procedures.
- Es posible construir triggers que se utilizan, entre otras cosas, para garantizar la importante propiedad de las bases de datos, denominada integridad referencial.
- El producto integra funciones de seguridad que permiten a los administradores de sistemas, controlar el personal que tendrá acceso a la base de datos y el nivel de autoridad con que lo hará. Como parte de su seguridad. Sybase proporciona la opción de generar discos espejos.

- Cuenta con un log de transacciones en el cual escribe cualquier modificación de la que sea objeto la base de datos.
- El API de SQL Server permiten a los desarrolladores de software construir aplicaciones cliente, que trabajen con SQL Server y con una interfaz transparente. DB-Library es un conjunto de funciones creadas en C, que permiten realizar comandos de SQL para recuperar y actualizar datos. Las rutinas de DB Library pueden ser usadas con C o Cobol. DB Library es una biblioteca con ligas estáticas bajo DOS y ligas dinámicas bajo Windows y OS/2.
- Cuenta con un optimizador de consultas, el cual analiza las sentencias SQL que el usuario introduce y si la consulta que el usuario está realizando no es la óptima, genera una nueva consulta que de manera más eficiente obtenga los datos requeridos.
- Sybase cuenta con un servidor de respaldos, el cual corre junto con el servidor SQL, a fin de brindar un alto desempeño en respaldos de bases de datos, cargas y recuperaciones.
- Cuenta con herramientas tales como editores de archivos de configuración, probadores de comunicaciones para testificar la comunicación entre el servidor y los clientes, editores para la creación de procedimientos con el lenguaje Transac-SQL, herramientas para monitoreo de procesos en el servidor y herramientas de reaplicación de datos.

Tabla de comparación entre los diferentes manejadores de bases de datos:

Modelo de Datos Relacional

	Microsoft SQL Server 6.5	**IBM DB/2 6.1**	**Informix Online7.2**	**Oracle7 7.3**	**Sybase SQL Server 11**
Opciones violación de Integridad referencia	Solo restrictiva	Restrictiva y en cascada	Restrictiva excepto el borrado en cascada	Restrictiva excepto el borrado en cascada	Solo restrictiva
Vistas actualizable (con opción de verificación)	Si	Si incluyendo vistas con unión	Si	Si	Si
Tipos de datos definidos por el usuario	Si	Si	No	Si	Si
Estructuras de índices	Clustered	Clustered	B + tree y clustered	B-tree, bitmap y hash	B-tree
Estructuras de tablas	Sin selección	Sin selección	Sin selección	Heap y clustered	Heap y clustered
Facilidades de optimización "tunning"	Fill factors	Distribución de índices y tablas cluster ratio y cluster factor	Extents fragmentación de tablas o round robin	Distribución de índices de tablas	Index prefetch i/o buffer cache, block size, table partitioning
Nivel de triggers	Basado en conjuntos de registros	Basado en registro y basado en conjuntos de registros	Basado en registro y basado en conjuntos de registros	Basado en registro y basado en conjuntos de registros	Basado en conjuntos de registros
Ejecución del trigger	Después de la operación que disparo el trigger	Antes y después de la operación que disparo el trigger	Antes y después de la operación que disparo el trigger	Antes y después de la operación que disparo el trigger	Después de la operación que disparo el trigger
Anidamiento del trigger	Si	Si	Si	Si	Si
Lenguaje empleado en Stored procedure. Transact-sql	Transact-sql	Sql&3gl	Sql	Pl/sql	Transact-sql

3.2.2.6. Sistemas Operativos para el DBMS

Algunos de los sistemas operativos en que son ejecutados estos DBMS son:

- Windows NT
- UNIX
- OS/2

3.2.2.6.1. Windows NT

Existe el Windows NT Server y el Windows NT Workstation.

Windows NT Server

- Desempeño del servidor. Ajuste del desempeño para aplicaciones, archivos e impresión. La versión liberada de Windows NT Server soporta hasta cuatro multiprocesadores en un ambiente de multiprocesamiento simétrico.
- 256 sesiones RAS (Remote Access Service).
- Tolerancia a fallos. Soporta tecnología RAID (Redundant Array of Inexpensive Disks) para protección de datos.
- Microsoft IIS. Integración de IIS con Windows NT Server 4.0 significa que la instalación y administración de un web server es simplemente otra parte del sistema operativo. Con IIS 2.0 o superior, es posible administrar en forma remota.
- Ayudantes administrativos. Incluye ayudantes que favorecen el desempeño de tareas comunes.
- Servicios adicionales de red. Proporciona servicios adicionales de red, incluye ruteo multiprotocolo, servidor DNS y WINS.

Windows NT Workstation

- Desempeño de escritorio. Soporta multitarea preemptive para todas las aplicaciones. Soporta múltiples microprocesadores para un desempeño real en multitareas.
- Perfiles de hardware. Crea y mantiene una lista de configuraciones de hardware para conocer las necesidades de computadoras específicas.
- Microsoft Internet Explorer. Provee un navegador que es rápido y simple de usar y compatible con los estándares existentes.
- Windows Messaging. Recibe y mantiene correo electrónico, incluyendo archivos y objetos creados en otras aplicaciones.
- Servicios Peer Web. Provee un servidor personal de Web, optimizado para correr sobre Windows NT Workstation 4.0
- Seguridad. Provee seguridad local por archivos, carpetas, impresoras y otros recursos. Los usuarios pueden ser autentificados tanto por la computadora local o un controlador de dominio en lugar de acceder cualquier recurso en la computadora o red.
- Estabilidad del sistema operativo. Soporta cada aplicación en su propio espacio de direcciones de memoria. Las aplicaciones con fallas de funcionamiento no afectarán otras aplicaciones o al sistema operativo.

3.2.2.6.2. UNIX

- Su componente principal es el kernel que se encarga de asignar tareas y manejar el almacenamiento de datos. El usuario rara vez opera directamente con el kernel, que es la parte residente en memoria del sistema operativo.

- Otro componente importante es el shell, esta es la utilidad que procesa las peticiones de los usuarios. Cuando alguien teclea un comando en la terminal, el shell interpreta el comando y llama el programa deseado. También es un lenguaje de programación de alto nivel que puede utilizarse en la combinación de programas de utilidad para crear aplicaciones completas.
- El shell puede soportar múltiples usuarios, múltiples tareas, y múltiples interfaces para sí mismo. Los dos shells más populares son el BourneShell (System V) y el Cshell (BSD Unix), debido a que usuarios diferentes pueden usar diferentes shells al mismo tiempo, entonces el sistema puede aparecer diferente para usuarios diferentes. Existe otro shell conocido como KornShell (así llamado en honor de su diseñador), que es muy popular entre los programadores.
- UNIX incluye una gran variedad de programas de utilidad que pueden ser fácilmente adaptadas para realizar tareas específicas. Estas utilerías son flexibles, adaptables, portables y modulares, y pueden ser usadas junto con filtros y redireccionamientos para hacerlos más poderosos.
- Es un sistema multiusuario, dependiendo del equipo disponible, un UNIX puede soportar desde uno hasta más de 100 usuarios, ejecutando cada uno de ellos un conjunto diferente de programas.
- Es un sistema multitareas, permite la realización de más de una tarea a la vez. Pueden ejecutarse varias tareas en su interior, mientras se presta toda la atención al programa desplegado en la terminal.
- Además, UNIX permite proteger los archivos del usuario contra el acceso por parte de otros usuarios.
- Los dispositivos (como una impresora o una terminal) y los archivos en disco son considerados como archivos por UNIX. Cuando se da una instrucción al UNIX puede indicársele que envíe el resultado a cualquiera de los diversos dispositivos o archivos. Esta desviación recibe el nombre de redireccionamiento de la salida.
- En forma similar, la entrada de un programa puede redireccionarse para que venga de un archivo en disco. En el UNIX, la entrada y la salida son independientes del dispositivo, pueden redireccionarse hacia o desde cualquier dispositivo apropiado.
- Comunicación entre procesos: UNIX permite el uso de conductos y filtros en la línea de comandos. Un conducto (pipe) redirige la salida de un programa para que se convierta en entrada de otro. Un filtro es un programa elaborado para procesar un flujo de datos de entrada y producir otro de datos de salida. Los conductos y filtros suelen usarse para unir utilerías y realizar alguna tarea específica.

3.2.2.6.3. OS/2

- OS/2 fue diseñado para ejecutar programas DOS al tiempo que proporcionaba capacidades extendidas, incluyendo multitarea y redes.
- Cuando en OS/2 se ejecuta el software de gestión LAN, se pueden aprovechar las características de seguridad similares a las proporcionadas por UNIX.
- Cuenta con un kit completo de multimedia y el Bonus Pack, un kit de aplicaciones que permite ponerse a trabajar con el ordenador, contiene elementos como un Kit de conexión a Internet completo, el paquete integrado IBM Works basado en OpenDoc, (formado por un procesador de textos, hoja de cálculo, base de datos y gráficos de empresa, junto con el PIM, que añade más funcionalidades aprovechando las capacidades drag&drop del WPShell), soft de terminal, soft de captura y tratamiento de video, etc.
- Soporta casi cualquier dispositivo existente en el mercado: CD-Roms, impresoras, tarjetas de sonido, soporte PCMCIA, tarjetas de video, tarjetas de captura de video, tarjetas SCSI, etc.
- Puede ejecutar programas DOS, OS/2 16bits, OS/2 32bits, Windows 3.x (incluía además el API Win32s, con lo que se podían ejecutar incluso programas Windows de 32bits).
- Cuenta con librerías OpenDoc (compatibles con OLE 2.0, pero más potentes).
- Librerías OpenGL, que permiten aprovechar las capacidades 3D de las tarjetas que soportan este estándar.
- API de desarrollo Open32, que permiten recompilar con suma facilidad las aplicaciones escritas para Windows'95 y Windows NT, de forma que aprovechen al máximo los recursos de OS/2.
- Cuenta con un extenso soporte de conectividad, lo que le convierte en el cliente de red universal, pudiendo conectarse a casi cualquier servidor (no solo Warp Server, sino Windows NT Server, Novell, etc.).
- Sesiones DOS reales (el micro se conmuta realmente a modo real, y todo el contenido de la RAM se guarda en disco, quedando el Sistema Operativo y el resto de las utilidades congelados, pudiendo rearrancar en cualquier

momento. Es útil para juegos o programas de DOS muy exigentes, que se niegan a funcionar en una sesión DOS virtual).

- La característica estrella de cara al marketing: el VoiceType. Se trata de un software reconocedor de voz, capaz de funcionar con cualquier tarjeta de sonido, y que permite al usuario trabajar exclusivamente mediante el dictado de comandos. Este sistema, al contrario de otros disponibles hasta el momento, realmente reconoce el habla de forma continua, de modo que no solo se puede usar para navegar por el escritorio y controlar programas, sino que sirve perfectamente para dictar cualquier tipo de texto, como artículos, cartas, etc. sin tocar una sola tecla. Se trata, por tanto, de un avance de los que serán, sin duda, los sistemas operativos del futuro.

Sistemas Operativos en los diferentes manejadores de bases de datos.

	Microsoft SQL Server 6.5	**IBM DB/2 6.1**	**Informix Online 7.2**	**Oracle7 7.3**	**Sybase SQL Server 11**
Windows NT	Si	Si		Si	Si
UNIX		Si	Si	Si	Si
OS/2	Si	Si		Si	Si

3.2.3 Lenguajes y herramientas de desarrollo

En este apartado se exponen algunas opciones sobre el leguaje y herramientas para el desarrollo del front-end para la implantación del sistema de información.

Para la definición del lenguaje y la herramienta de desarrollo, se tomarán en cuenta a los más importantes y convenientes para el sistema de información, resultando los siguientes:

- **Visual C++**
- **Power Builder**
- **Visual Basic**
- **Java**
- **Borland JBuilder**
- **Html**

Visual C++

Es una herramienta poderosa y flexible, para generar soluciones de negocios, además de ser una herramienta rápida para los programadores que utilizan C y C++.

Ventajas:

- Incluye gran variedad de librerías construidas.
- Controles visuales.
- Permite la reutilización total de código.
- Manejo completo del esquema de la programación orientada a objetos (herencia, polimorfismo, etc)
- Maneja controles VBX (Visual Basic eXtension), los cuales son componentes preconstruidos que aceleran el desarrollo del software, compatibles con la versión 1.0 de Visual Basic.

- Todo el manejo de dispositivos de entrada/salida (I/O), se hace a través de librerías.
- Cuenta con compilador, no con interprete.
- Se pueden crear clientes OLE a través de él.
- Maneja multitarea.
- Rapidez en ejecución de programas.
- Manejo de variables locales y globales.
- Permite obtener funcionalidades de otras aplicaciones para utilizarlas como componentes en aplicaciones propias (por ejemplo: Word, Excel, Project, etc.).
- Manejo de controles OLE.
- Libertad total para el manejo de memoria y accesos a sistema operativo.
- Permite creación de servidores OLE.

Desventajas:

- Curva de aprendizaje alta.
- Tiempo de compilación largos (superable en nuevas versiones).

Power Builder

Es un ambiente para desarrollar aplicaciones gráficas. Es un entorno de desarrollo comprensivo para construir aplicaciones cliente/servidor de alto desempeño para la familia de Windows que combina una interfase gráfica intuitiva con un poderoso lenguaje de programación orientada a objetos.

Power Builder soporta multiplataformas desarrolladas y desplegadas. Por ejemplo, se puede desarrollar una aplicación usando Power Builder bajo Windows (Windows 95 o Windows NT) y desplegar la misma aplicación –sin hacer cambios- sobre máquinas Windows 3.11, Macintosh, o Unix.

Power Builder incluye herramientas que le permiten construir aplicaciones basadas en Web y extender la existencia de su aplicación al Internet. Es un front-end que puede interactuar con la mayoría de DBMS basados en ODBC.

Ventajas:

- Permite el desarrollo de aplicaciones robustas, bajo ambientes multiplataforma, cuenta con un optimizador de código, posibilidad de distribución de objetos en un ambiente de red y manejadores nativos para diferentes bases de datos.
- Posee un soporte completo para ambiente Windows de 16 y 32 bits en plataformas Intel, incluyendo Microsoft Windows 3.x, Windows NT, Windows 95, OS/2, Macintosh y Unix.
- Posee una familia de herramientas de desarrollo escalable que incrementan la productividad de las aplicaciones. La serie de power Builder Enterprise, power Builder Team/PDBS, Power Builder Desktop, Infomaker y Power Builder Library for Lotus Notes.
- Puede definir, compilar y corregir clases integradas de C++ basadas en el compilador Watcom C/C++ de alto rendimiento.
- Ofrece un extenso lenguaje orientado a objetos con miles de funciones. Los programadores pueden escribir sus propias funciones o utilizar las ya existentes escritas en C o en otros lenguajes. Cuenta con un compilador manejador y un debugger con muchas particularidades positivas.
- Incluye toda una gama de programación orientada a objetos, pero en Watcom C++ y en 3GL. Power Builder soporta la definición de clases para modelados visuales y objetos no visuales. Incluye herencia, encapsulamiento de datos y procesos lógicos, y polimorfismo.
- Con respecto a la programación orientada a eventos, se basa en los eventos de Windows en si totalidad.
- Se incluye una biblioteca de objetos centralizada y un administrador de código fuente.
- Soporta desarrollo en grupo.

- El código se separa del lenguaje de transacción, optimizando la programación y el aprendizaje de la herramienta.
- Tiene un amplio manejo de gráficos de dos o tres dimensiones.

Desventajas

- Tiene un ambiente de programación diferente al normal.

Visual Basic

Este lenguaje está disponible para programadores de MS-DOS y Windows. Es una herramienta rápida para crear aplicaciones generales de negocios. Requiere pocos conocimientos de programación y tiene una rápida integración y reutilización de componentes.

Proporciona gran flexibilidad para llevar a cabo llamadas a archivos con extensión DLL y agregar una gran cantidad de controles, al gusto del cliente.

Ventajas:

- Es un sistema productivo para crear soluciones en Windows.
- Controles visuales preconstruidos por terceros.
- Acceso a bases de datos.
- Manejo de multimedia.
- Permite un ensamblaje fácil y rápido entre la interfase de un usuario y los componentes prefabricados.
- Ofrece una gran capacidad y velocidad en su debugger sofisticado.
- Permite manipular otras aplicaciones para utilizarlas como componentes en aplicaciones propias (como Word, Excel, Project, etc), siempre y cuando dichas aplicaciones soporten OLE automation.
- Soporta variedad de manejadores de bases de datos.
- Incluye como base de datos nativa a Microsoft Access, la cual provee acceso simultaneo con FoxPro, Dbase, Paradox, SQL, etc.
- Maneja aplicaciones cliente-servidor.

Desventajas

- Visual Basic se tiene que auxiliar de lenguajes mas robustos como "C" o Pascal, para hacer manipulación de memoria, accesos directos al sistema operativo, etc.
- No cuenta con la característica de herencia, que es muy importante dentro de la programación orientada a objetos.
- Su interprete es lento en la creación de ejecutables.
- Su código no es compatible, en plataformas diferentes a Windows.

Java

Las características principales que Java respecto a cualquier otro lenguaje de programación, son:

Es *SIMPLE*: Java ofrece toda la funcionalidad de un lenguaje potente, pero sin las características menos usadas y más confusas de éstos. C++ es un lenguaje que adolece de falta de seguridad, pero C y C++ son lenguajes más difundidos, por ello Java se diseñó para ser parecido a C++ y así facilitar un rápido y fácil aprendizaje. Java elimina muchas de las características de otros lenguajes como C++, para mantener reducidas las especificaciones del lenguaje y añadir características muy útiles como el *garbage collector* (reciclador de memoria dinámica). No es necesario preocuparse

de liberar memoria, el reciclador se encarga de ello y como es un *thread* de baja prioridad, cuando entra en acción, permite liberar bloques de memoria muy grandes, lo que reduce la fragmentación de la memoria. Java reduce en un 50% los errores más comunes de programación con lenguajes como C y C++ al eliminar muchas de las características de éstos.

Es ***ORIENTADO A OBJETOS***: Java implementa la tecnología básica de C++ con algunas mejoras y elimina algunas cosas para mantener el objetivo de la simplicidad del lenguaje. Java trabaja con sus datos como objetos y con interfaces a esos objetos. Soporta las tres características propias del paradigma de la orientación a objetos: encapsulación, herencia y polimorfismo. Las plantillas de objetos son llamadas, como en C++, *clases* y sus copias, *instancias*. Estas instancias, como en C++, necesitan ser construidas y destruidas en espacios de memoria. Java incorpora funcionalidades inexistentes en C++ como por ejemplo, la resolución dinámica de métodos. Esta característica deriva del lenguaje Objective C, propietario del sistema operativo Next. En C++ se suele trabajar con librerías dinámicas (DLLs) que obligan a recompilar la aplicación cuando se retocan las funciones que se encuentran en su interior. Este inconveniente es resuelto por Java mediante una interfaz específica llamada RTTI (*RunTime Type Identification*) que define la interacción entre objetos excluyendo variables de instancias o implementación de métodos. Las clases en Java tienen una representación en el runtime que permite a los programadores interrogar por el tipo de clase y enlazar dinámicamente la clase con el resultado de la búsqueda.

Es ***DISTRIBUIDO***: Java en sí no es distribuido, sino que proporciona las librerías y herramientas para que los programas puedan ser distribuidos, es decir, que se corran en varias máquinas, interactuando. Java se ha construido con extensas capacidades de interconexión TCP/IP. Existen librerías de rutinas para acceder e interactuar con protocolos como *http* y *ftp*. Esto permite a los programadores acceder a la información a través de la red con facilidad.

Es ***ROBUSTO***: Java realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución. La comprobación de tipos en Java ayuda a detectar errores, lo antes posible, en el ciclo de desarrollo. Java obliga a la declaración explícita de métodos, reduciendo así las posibilidades de error. Maneja la memoria para eliminar las preocupaciones por parte del programador de la liberación o corrupción de memoria. También implementa los *arrays auténticos*, en vez de listas enlazadas de punteros, con comprobación de límites, para evitar la posibilidad de sobrescribir o corromper memoria resultado de punteros que señalan a zonas equivocadas. Estas características reducen drásticamente el tiempo de desarrollo de aplicaciones en Java. Además, para asegurar el funcionamiento de la aplicación, realiza una verificación de los *byte-codes*, que son el resultado de la compilación de un programa Java. Es un código de máquina virtual que es interpretado por el intérprete Java. Java proporciona principalmente: comprobación de punteros, comprobación de límites de arrays , excepciones y verificación de byte-codes.

Es de ***ARQUITECTURA NEUTRAL***: Para establecer Java como parte integral de la red, el compilador Java compila su código a un fichero objeto de formato independiente de la arquitectura de la máquina en que se ejecutará. Cualquier máquina que tenga el sistema de ejecución (*run-time*) puede ejecutar ese código objeto, sin importar en modo alguno la máquina en que ha sido generado. Actualmente existen sistemas run-time para Solaris 2.x, Sun Os 4.1.x, Windows 95, Windows NT, Linux, Irix, Aix, y Mac.

Es ***SEGURO***: La seguridad en Java tiene dos facetas. En el lenguaje, características como los punteros o el *casting* implícito que hacen los compiladores de C y C++ se eliminan para prevenir el acceso ilegal a la memoria. Cuando se usa Java para crear un navegador, se combinan las características del lenguaje con protecciones de sentido común aplicadas al propio navegador. El lenguaje C, por ejemplo, tiene lagunas de seguridad importantes, como son los *errores de alineación*. Los programadores de C utilizan punteros en conjunción con operaciones aritméticas. Esto le permite al programador que un puntero referencie a un lugar conocido de la memoria y pueda sumar (o restar) algún valor, para referirse a otro lugar de la memoria. Si otros programadores conocen nuestras estructuras de datos pueden extraer información confidencial de nuestro sistema.

Es ***PORTABLE***: Más allá de la portabilidad básica por ser de arquitectura independiente, Java implementa otros estándares de portabilidad para facilitar el desarrollo. Los enteros son siempre *enteros* y además, enteros de 32 bits en complemento a 2. Además, Java construye sus interfaces de usuario a través de un sistema abstracto de ventanas de forma que las ventanas puedan ser implantadas en entornos Unix, PC o Mac.

Es ***INTERPRETADO***: El intérprete Java (sistema run-time) puede ejecutar directamente el código objeto. Enlazar (linkar) un programa, normalmente, consume menos recursos que compilarlo, por lo que los desarrolladores con Java pasarán más tiempo desarrollando y menos esperando por el ordenador. No obstante, el compilador actual del JDK es bastante lento. Por ahora, que todavía no hay compiladores específicos de Java para las diversas plataformas, Java es más lento que otros lenguajes de programación, como C++, ya que debe ser interpretado y no ejecutado como sucede en cualquier programa tradicional.

Es ***MULTITHREADED***: Al ser multithreaded (multihilos), Java permite muchas actividades simultáneas en un programa. Los threads (a veces llamados, procesos ligeros), son básicamente pequeños procesos o piezas independientes de un gran proceso. Al estar los threads construidos en el lenguaje, son más fáciles de usar y más robustos que sus homólogos en C o C++. El beneficio de ser miltithreaded consiste en un mejor rendimiento interactivo y mejor comportamiento en tiempo real. Aunque el comportamiento en tiempo real está limitado a las capacidades del sistema operativo subyacente (Unix, Windows, etc.), aún supera a los entornos de flujo único de programa (single-threaded) tanto en facilidad de desarrollo como en rendimiento. Cualquiera que haya utilizado la tecnología de navegación concurrente, sabe lo frustrante que puede ser esperar por una gran imagen que se está trayendo. En Java, las imágenes se pueden ir trayendo en un thread independiente, permitiendo que el usuario pueda acceder a la información en la página sin tener que esperar por el navegador.

Es ***DINAMICO***: Java se beneficia todo lo posible de la tecnología orientada a objetos. Java no intenta conectar todos los módulos que comprenden una aplicación hasta el tiempo de ejecución. Las librería nuevas o actualizadas no paralizarán las aplicaciones actuales (siempre que mantengan el API anterior).

Borland JBuilder

JBuilder es un entorno de desarrollo rápido de aplicaciones que facilita la creación y edición de código Java, permitiendo incluso el diseño de JavaBeans, los componentes basados en Java que podrán ser posteriormente incluidos en cualquier programa escrito con este lenguaje.

Los programadores que conozcan Delphi o C++ Builder encontrarán el entorno de JBuilder ligeramente familiar, ya que en él existen elementos como la Paleta de componentes o el área de botones de acciones rápidas, habituales en otras herramientas de desarrollo de Borland.

JBuilder es una herramienta de desarrollo Java de segunda generación, basada en el JDK 1.1 hecho público en febrero de 1997. Con JBuilder se pueden escribir sencillos *applets* para páginas Web y también desarrollar completas aplicaciones distribuidas y cliente/servidor para Internet o intranets corporativas. El entorno de JBuilder consta de una ventana principal, en la que se encuentra el menú, la Paleta de componentes y los botones de acciones rápidas; y uno o más AppBrowser, nombre asignado al visor de proyectos, clases, interfaz y código. La portabilidad de los programas escritos con JBuilder está asegurada, ya que es una herramienta que genera puro código Java cien por ciento. Lo único preciso es disponer de la VM (Máquina virtual Java) basada en la versión 1.1.

El entorno de JBuilder está basado en las conocidas *two-way-tools* de Borland, que consiguen que cualquier modificación de la interfaz tenga un reflejo inmediato en el código y viceversa. De esta forma el programador puede optar en cada momento por realizar una determinada tarea como le sea más cómodo, arrastrando y soltando componentes y editando propiedades, o bien escribiendo directamente código Java. El *AppBrowser* de JBuilder aúna en una ventana, dividida en tres paneles y con múltiples páginas, el gestor de proyectos, visor de jerarquía de clases, editores de código, interfaz y documentación, etc. Podemos tener múltiples aplicaciones abiertas de forma simultánea, cada una de ellas en su propio *AppBrowser*. Cuando nuestro código esté dispuesto para ser usado, un asistente nos

permitirá incluso su publicación, incluyendo los archivos necesarios en un paquete cuyo formato podemos seleccionar, entre ZIP y JAR con o sin comprimir. JBuilder incorpora soporte total para JDBC, lo que le facilita el acceso a bases de datos Sybase, DB2, InterBase, Informix, Oracle, etc.

Si los datos se alojan en bases accesibles mediante ODBC tampoco se encontrará problema alguno para utilizarlas desde JBuilder, gracias a que incorpora un puente JDBC-ODBC.

Html

En 1989, Tim Berners Lee propuso diseñar un sistema de unificación del acceso a todos los datos que poseía el Centro Europeo para la Investigación Nuclear. Se comenzó así a desarrollar una plataforma de tipo hipertexto y un protocolo de comunicaciones que se denominó HTTP (Hyper Text Transfer Protocol), que permitiría a todos los científicos del CERN, consultar cualquier información de cualquier tema, aunque se encontrase diseminada en los diferentes ordenadores, tanto del propio centro, como en los ordenadores de las diferentes instituciones que colaboraban con el CERN. El sistema alcanzó un éxito enorme, tanto es así que se comenzó a definir un lenguaje de creación de documentos estructurados que vino a llamarse HTML (Hyper Text Markup Language).

Debido a que existen varias versiones de HTML muchas páginas Web actualmente especifican con que navegador pueden ser vista de mejor forma, sin embargo, si se cuenta con alguno de última generación (Internet Explorer 4 0 superior o Netscape 4 o superior) por norma se deberá de poder visualizar cualquier página sin problemas.

La creación de código HTML es muy sencillo de realizar utilizando casi cualquier editor de texto como NotePad y WordPad de Microsoft, Editor "Vi" de sistemas Unix, y aplicaciones dedicadas específicamente a esta tarea como DreamWeaver, HomeSite y FrontPage, entre otros.

De acuerdo al análisis realizado en este apartado, se ha definido utilizar lo siguiente para el desarrollo del sistema de información:

Lenguaje/herramienta de desarrollo	Justificación
Java (Lenguaje)	Es el lenguaje que cumple satisfactoriamente los siguiente rubros: Orientado a Objetos Seguridad (evita accesos a memoria y ataques como "caballo de troya"). Robustez (verificaciones tiempo de compilación como en tiempo de ejecución) Sencillez (fácil aprendizaje) Independiente del sistema operativo (Arquitectura neutral) Distribuido (opera con TCP/IP).
Borland Jbuilder (Herramienta)	Es la herramienta de desarrollo que cumple con el lenguaje seleccionado y se aprovechan sus características: Facilita la creación y edición de código Java. Interfaz gráfica de fácil uso. Está asegurada la portabilidad de los programas escritos con JBuilder. Amplio soporte en conectividad a Bases de Datos a través de JDBC y puentes a ODBC.
HTML (Lenguaje)	Es el lenguaje esencial para el desarrollo de aplicaciones para Intranet/Internet.

3.3. Propuesta de solución

A lo largo de este capítulo se desarrollaron tanto el planteamiento del problema como el análisis comparativo de las herramientas comerciales de desarrollo, con lo cual ya se tienen los elementos necesarios para poder elegir los métodos más adecuados al igual que la solución óptima al problema referente al manejador de bases de datos, sistema operativo y lenguaje de programación a utilizar.

3.3.1. Método

Se van a utilizar varios métodos de acuerdo a las características y facilidades de uso que sean óptimos en el desarrollo de la propuesta.

Los métodos que principalmente se utilizarán para el desarrollo de la propuesta son los siguientes:

- OOA/OOD por Coad-Yourdon
- OMT

3.3.1.1. OOA/OOD por Coad-Yourdon

La importancia fundamental del método Coad-Yourdon es que utiliza una descripción breve y concisa, así como el uso de textos generales como fuentes para las definiciones; de modo que las definiciones se enmarcan dentro del sentido común y reduce el empleo de modismos, la desventaja del método es su notación compleja, la cual es difícil de utilizar sin una herramienta, por lo que algunos usuarios del método han utilizado la notación de diagramación de OMT en su lugar.

3.3.1.2. OMT

Técnica de Modelado de Objetos (Object modeling technique). Los aspectos más importantes son sus capacidades simples aunque poderosas de notación, así como su madurez. Este ha sido aplicado en varios proyectos por sus autores aún antes de ser publicado.

3.3.2. Selección de la solución óptima

Dentro de la solución óptima se va a incluir el lenguaje UML, ya que es una herramienta útil para el modelado de la información.

UML es apropiado para el modelado de sistemas de empresas con sistemas de información para aplicaciones distribuidas basadas en Web, aún en desarrollos de sistemas en tiempo real.

Un factor también importante en la selección de la solución del problema, es el manejador de la base de datos; en la siguiente tabla se presentan características generales de los diferentes manejadores de bases de datos, a cada uno se le otorga una calificación, de 1 a 5, donde 5 es considerada la calificación más alta.

	Microsoft SQL Server6.5	IBM DB/2 6.1	Informix Online 7.2	Oracle7 7.3	Sybase SQL Server 11
A) Eficiencia y recuperación de caídas	5	4	4	5	4
B) Capacidad de transacciones On-Line	5	3	4	5	5
C) Integridad	4	3	4	4	5
D) Herramientas de programación	2	2	4	4	5
E) Administración del sistema	4	2	4	4	4
F) Herramientas del usuario final	5	2	3	5	4
G) Seguridad del sistema operativo	5	3	4	4	4
H) Seguridad multinivel	5	4	5	5	4
I) Soporte SQL estándar	5	4	4	5	5
J) Aplicaciones de soporte en toma de decisiones	4	3	4	4	3
K) Funciones de Auditoria	4	2	3	4	3
L) Facilidades de migración	4	2	4	4	4
M) Extensiones SQL	4	3	4	5	4
N) Actualización, recuperación y administración remota	3	3	3	3	4
O) Integración de herramientas CASE	4	2	4	5	4

Esta tabla es el resultado de un estudio realizado por la revista Computer World de México.

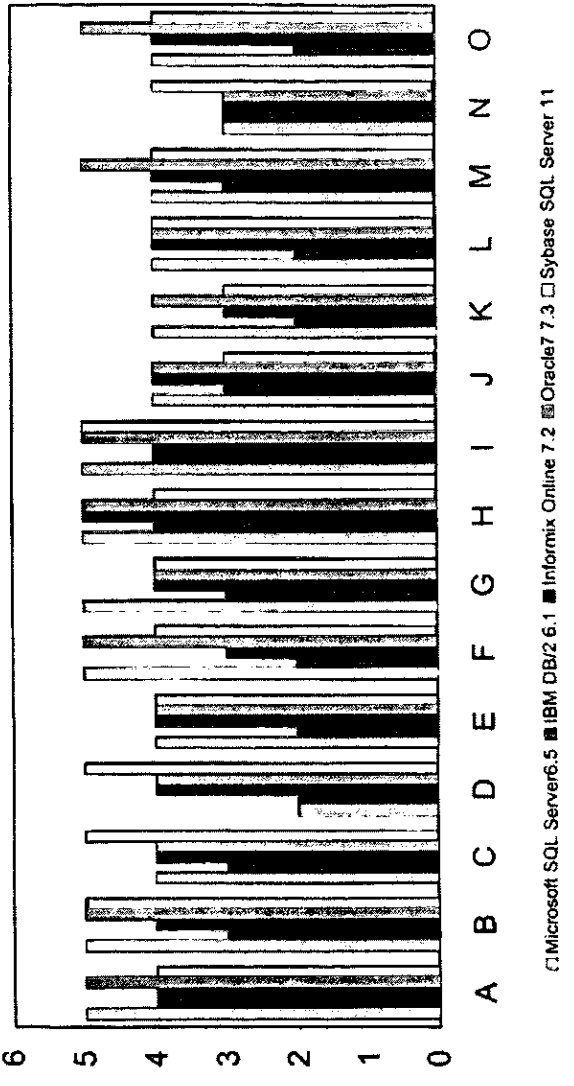


Figura 3.1 Selección de la solución óptima

De acuerdo a la figura 3.1 que es la gráfica de la tabla anterior, es posible observar que los que tienen calificación más alta son Oracle y SQL Server.

Como resumen y enfatizando los puntos anteriores la conclusión de nuestra solución es:

Manejador de base de datos óptimo:

Oracle 7 7.3

Ya que tiene mayor cantidad de puntos y es compatible para una gran diversidad de hardware, diversas plataformas de software y topologías de redes.

También es soportado por un gran número de lenguajes de cuarta generación manejando volúmenes de información muy grandes, así como contar con opciones de seguridad para el manejo de la base de datos, por estas razones se decidió que el DBMS óptimo para usar en la creación del sistema sería Oracle7 7.3

Sistema operativo óptimo:

Solaris (Unix)

El ambiente del Sistema Operativo Solaris brinda algunas caracteristicas, las cuales ayudarán a dar una buena portabilidad, escalabilidad, compatibilidad y seguridad en las aplicaciones para así operar en un ambiente creciente. También presenta características para los usuarios la cual cuenta con ventanas para manejo de servicios rápidos, al igual que integra servicios desktop, bibliotecas gráficas, administración de calendario y herramientas de imagen. Este sistema tiene características para el administrador de sistemas en donde se puede obtener información sobre dispositivos.

Nativamente Oracle 7 7.3 opera sobre la plataforma UNIX, y ya que este sistema operativo cuenta con un sistema multitareas y multiusuarios, además de existir la comunicación entre procesos y la fácil adaptación de programas de utilidad para realizar tareas específicas.

Por estas razones Solaris es la mejor opción como plataforma operativa para la implantación del sistema.

Se utilizará Solaris 7 ya que es el mejor entorno de network computing para dar servicio a las necesidades del sistema, ya que Solaris 7 incluye un entorno de 64-bit, interoperabilidad mejorada con PCs y una administración más sencilla que versiones anteriores.

Herramientas de desarrollo:

JBuilder 4 (Java)

Para el software de desarrollo del front-end, se tomó la decisión de usar Java de Sun debido a que es el lenguaje de programación utilizado para generar aplicaciones para uso en Internet o en intranet, además de que es compatible con Oracle 7 7.3 y con el sistema operativo UNIX (Solaris), con lo cual se añade mayor funcionalidad al sistema por medio de este lenguaje que también se puede considerar una herramienta para creación y uso en ambiente web.

Arquitectura de red:

La solución propuesta para el desarrollo del sistema se puede implementar fácilmente en la arquitectura de red que se tienen dentro de la empresa, utilizando la tecnología Ethernet que actualmente se usa.

En función a los requerimientos del sistema, la plataforma del sistema operativo y el manejador de la base de datos que se escogieron la arquitectura Cliente/Servidor es la base para el funcionamiento del sistema.

Para la selección de la solución óptima, además de los aspectos anteriores se consideraron estos puntos:

- Facilidad de uso y administración en equipos Cliente/Servidor. Solaris facilita el usar y manipular sus operaciones, porque todo se maneja como si fueran archivos los que se usan, y de hecho así ocurre.
- Seguridad de parte del sistema operativo como del manejador de la base de datos. Solaris cuenta con un sistema de seguridad el cual consiste en proteger los archivos del usuario contra el acceso por parte de otros usuarios. Oracle 7 7.3 cuenta con opciones de seguridad para el manejo de la base de datos y maneja los esquemas de discos espejos para poder continuar con los procesos en caso de que un disco se dañe.
- Compatibilidad. Solaris incluye una gran variedad de programas de utilidad que pueden ser fácilmente adaptados para realizar tareas específicas. También la mayoría de las aplicaciones de gestión del mercado corren sobre Oracle 7 7.3.
- Integración. Java como lenguaje y herramienta Web, habilita tecnologías tales como HTML, CGI, NSAPI y ActiveX, con lo que se obtiene mayor funcionalidad e integración de elementos con este lenguaje.
- Desarrollo de interfaces gráficas. Java, y especialmente J-Builder, es un sistema que permite crear interfaces gráficas y hacer que el sistema sea más amigable, que tenga mayor facilidad para ser manejado por el usuario. J-Builder permite utilizar los controles existentes y además permite crear controles con funciones particulares que serán utilizados por sistema.
- Soporta un gran volumen de datos. Oracle 7 7.3 puede manejar un gran volumen de datos sin ningún problema.
- Arquitectura Ethernet. En las instalaciones se cuenta con una red LAN, que opera a 10/100 Mbps utilizando como protocolo predeterminado TCP/IP.

A partir de la información seleccionada en este capítulo se sientan las bases para hacer el análisis y diseño del sistema de inventario para la empresa Softtek.

Capítulo 4

Análisis, Diseño y Desarrollo del Sistema

4. Análisis, Diseño y Desarrollo del Sistema

A continuación, se procederá a elaborar el sistema de información, aplicando la metodología orientada a objetos en el "ciclo de vida del software", con lo que su elaboración se divide en tres rubros, los cuales son: Análisis, Diseño y Desarrollo del Sistema.

4.1. Análisis

El primero de los puntos esenciales para el planteamiento a la solución, este tema es vital ya que es necesario estar en contacto frecuente con el usuario, para conocer a fondo sus necesidades y reflejarlas en el diseño del sistema.

4.1.1. Descripción de escenarios según los casos de uso

La recopilación de requisitos es siempre el primer paso en cualquier actividad de análisis de software. Dicha recopilación puede tomar forma de un rápido encuentro con el cliente y el desarrollador en el cual acuerdan definir los aspectos básicos del sistema.

En muchos casos la dinámica del comportamiento tiene efecto en respuesta a un estímulo inicial del usuario. Así que una primera aproximación es el considerar las cosas que el usuario puede hacer con el sistema y luego diseñar marcos de objetos e interacciones que soporten dicho comportamiento. Esta técnica se llama casos de uso.

Los casos de uso describen bajo la forma de acciones y reacciones el comportamiento de un sistema desde el punto de vista del usuario; permiten definir los límites del sistema y las relaciones entre el sistema y el entorno.

Un caso de uso es una manera específica de utilizar un sistema. Es la funcionalidad del sistema, que se desencadena en respuesta a la estimulación de un actor externo.

Un caso de uso representa un requisito funcional del sistema y describe al mismo tiempo un conjunto de secuencias, donde cada secuencia representa la interacción de los elementos externos al sistema (actores) con el propio sistema.

A continuación se hace la descripción de escenarios a partir de los casos de uso obtenidos para el diseño del sistema.

- Entrar al Sistema

Permitir a los actores acceder al sistema, validándose mediante un nombre de usuario y una clave o password.

- Alta de Registro Facturas

El actor podrá registrar y dar de alta facturas, capturando todos los datos del equipo adquirido y registrarlo en una base de datos.

- Baja Registro Facturas

Se podrán dar de baja facturas ya registradas por razones de cancelación.

- Alta registro de Equipo

Una vez que se ha dado de alta una factura es necesario registrar el equipo adquirido agregándolo a la base de datos.

- Baja Registro de equipo

Se harán los movimientos de baja de equipo, ya sea componentes, periféricos o equipos completos por motivos de obsolescencia o descompostura.

- Consulta Equipo

En este caso será posible hacer un desplegado del equipo que se tiene registrado y ver a detalle las características de cada uno.

- Alta registro de Equipo Procesador
- Alta registro de Equipo Memoria
- Alta registro de Equipo Disco Duro
- Alta registro de Equipo Tarjeta Red
- Alta registro de Equipo Conector
- Alta registro de Equipo Otros
- Alta registro de Equipo Monitor
- Alta registro de Equipo Teclado y Mouse

En los casos de uso anteriores, una vez que el equipo ha sido registrado (No. de Serie, Marca, Modelo, etc.) se capturan las características especiales del CPU, como bahías PCI, slots de Memoria, tipo de conectores para Mouse y Teclado, Tipo y Velocidad del procesador o componentes internos adicionales que necesite como módems y tarjetas de red, con sus características específicas cada uno.

- Modificar Equipo CPU

Una vez que el equipo ya se ha sido registrado, se podrán hacer modificaciones a las características o descripciones de los mismos, para los casos donde se incrementes partes o componentes a los equipos.

- Alta registro de Seguro de Equipo

Para cada registro de alta de los equipos, para CPU y Laptop se les asigna un seguro donde se especifica la fecha de inicio y término del mismo.

- Modificación de Seguro de Equipo

En este caso de dará la opción de modificar la fecha de vencimiento del seguro por renovación del mismo.

- Alta de Asignación de Equipo

Una vez que el equipo ha sido capturado y se tiene en la base de datos, el actor podrá asignarlo a alguna persona, agregando datos como fecha de asignación, fecha de devolución, ubicación, área, proyecto, etc. y datos necesarios para identificar el equipo y a la persona responsable.

- Baja de Asignación de Equipo

En este escenario se registrará la devolución de equipo, cambiando el estatus del mismo de asignado a disponible y mantener el equipo en stock para futuras asignaciones.

- Consulta de Asignación de Equipo

Para llevar un mejor control sobre los equipos es necesario poder hacer consultas de las asignaciones de equipo, ya sea para saber quien tiene un equipo en particular o para saber cuantos equipos tiene asignados un usuario en particular.

- Modificación de Asignación de Equipo

Una vez que el equipo ha sido asignado frecuentemente se tiene que realizar modificaciones a la asignación, como lugar de trabajo, fechas de asignación, cambio de sector ó responsable del equipo, esto debido a la rotación de la gente en proyectos.

- Consulta Status
- Consulta Sector
- Consulta Procesador
- Consulta Pertenencia
- Consulta Bitácora
- Consulta Clave de Proyecto

Para el manejo de información del inventario de la empresa es necesario tener algunas estadísticas para conocer las tecnologías que se tienen como los volúmenes de equipo, así como el stock disponible, equipos arrendados, la demanda de equipos por sectores y proyectos, el actor podrá visualizar de manera gráfica todas estas especificaciones.

- Alta Mantenimiento
- Modificación Mantenimiento
- Baja Mantenimiento
- Consulta Mantenimiento

Es necesario llevar un registro de los mantenimientos preventivos y correctivos a los equipos de cómputo y periféricos, por lo que se tienen que dar de alta el servicio, hacer modificaciones de fechas y/o componentes que se hayan cambiado, consultar los estatus de los equipos en mantenimiento o garantía y por último dar de baja el servicio de mantenimiento.

- Alta Catálogo Equipo Componente
- Modificación Catálogo Equipo Componente
- Baja Catálogo Equipo Componente
- Consulta Catálogo Equipo Componente

Para hacer le registro o alta de equipo previamente se tienen que dar de alta los catálogos correspondientes a los componentes del CPU, como pueden ser módems, tarjetas de red, tarjetas SCSI, etc., especificando sus características especiales de cada uno, así mismo será necesario poder hacer modificaciones a los registros ya capturados, hacer consultas del catálogo y poder dar de baja elementos previamente registrados.

- Alta Catálogo Marca
- Modificación Catálogo Marca
- Baja Catálogo Equipo Marca
- Consulta Catálogo Marca

Para cada registro de equipo en la base de datos es necesario que previamente se capture un catálogo de marcas, ya que al dar de alta un equipo o componente es necesario especificar la marca del fabricante, una vez que se ha capturado la marca de los componentes va a ser necesario hacer modificaciones de las mismas, hacer consultas y poder dar de baja registros de la base de datos.

- Alta Catálogo Sector
- Modificación Catálogo Sector
- Baja Catálogo Equipo Sector
- Consulta Catálogo Sector

Un parámetro necesario en los escenarios de asignación de equipo es el sector que se va a signar el equipo, estos sectores están definidos en el primer capítulo, pero es necesario poder modificar dichos sectores, consultarlos y darlos de baja cuando sea necesario.

- Alta Catálogo Ubicación
- Modificación Catálogo Ubicación
- Baja Catálogo Equipo Ubicación
- Consulta Catálogo Ubicación

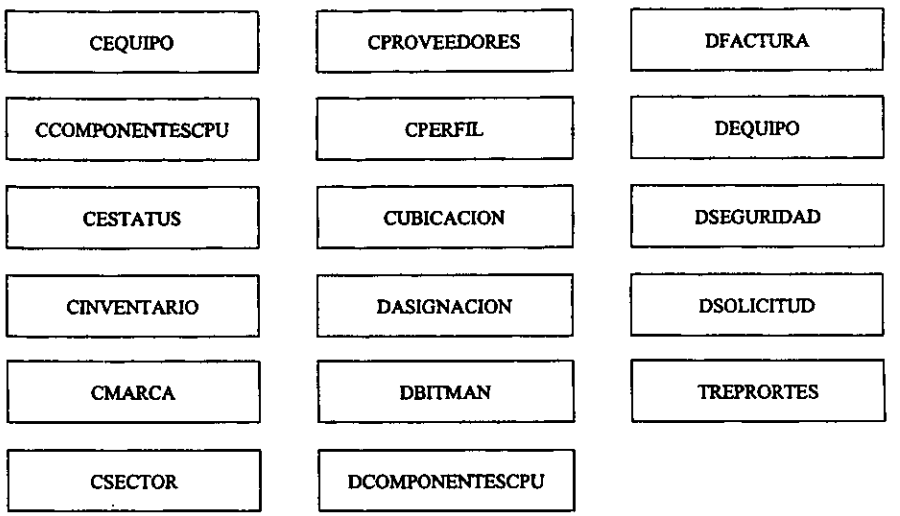
En los escenarios de asignación de equipo es necesario especificar la ubicación física del equipo, tanto dentro de las instalaciones de la empresa especificando la oficina como su ubicación en los proyectos en las instalaciones de los clientes, sobre este catálogo es necesario poder hacer consultas, modificaciones y bajas de aquellas direcciones que no se vayan a usar más.

- Alta de Usuario
- Modificación de Usuario
- Baja de Usuario

En estos escenarios se dan de alta los usuarios que podrán tener acceso al sistema de inventarios, modificar su usuario o dar de baja a los usuarios que ya no operaran el sistema.

4.1.2. Filtrado, identificación y definición de objetos

En el diagrama entidad-relación se hizo la identificación de las entidades para la creación del sistema las cuales fueron:



4.1.3. Identificación de clases y lista de operaciones

Una vez identificadas las entidades se procederá a identificar las clases y listas de operaciones.

CLASES	OBJETOS	DESCRIPCIÓN
CCAPTIPOCOMPCPU	nCAT_INV_CVE	Clave de Inventario según catálogo
	nCVE_CARACT	Clave de características
	sCAT_COMP_TIPO	Tipo de componente según catálogo
	sCAT_COMP_CAP	Capacidad de componente según catálogo
	sCAT_COMP_DESC	Descripción de componente según catálogo

	nCAT_INV_CVE	Clave de inventario según catálogo
	nCVE_CARACT	Clave de características
CVELMARCCOMPCPU	sCAT_COMP_VEL	Velocidad de componente según catálogo
	sCAT_COMP_MCADDRESS	Marca y dirección de componente según catálogo
	nCAT_INV_CVE	Clave de inventario según catálogo
	nCVE_CARACT	Clave de características
CEQUIPO	nCAT_MARC_CVE	Clave de marca según catálogo
	sCAT_EQP_TIPO	Tipo de equipo según catálogo
	sCAT_EQP_DESC	Descripción de equipo según catálogo
	nCAT_STA_CVE	Clave del estatus según catálogo
CESTATUS	sCAT_STA_DESC	Descripción del estatus según catálogo
	nCAT_INV_CVE	Clave de inventario según catálogo
CINVENTARIO	sCAT_INV_DESC	Descripción de inventario según catálogo
	nCAT_MARC_CVE	Clave de marca según catálogo
CMARCA	sCAT_MARC_DESC	Descripción de marca según catálogo
	nPERF_CVE	Clave del perfil
CPERFIL	sPERF_DESC	Descripción del perfil
	nPROV_CVE	Clave de proveedores
CPROVEEDORES	sPROV_DESC	Descripción de proveedores
	nCAT_SEC_CVE	Clave de sector según catálogo
CSECTOR	sCAT_SEC_DESC	Descripción de sector según catálogo
	nCAT_UBI_CVE	Clave de la ubicación según catálogo
CUBICACIÓN	sCAT_UBI_DESC	Descripción de la ubicación según catálogo
	nASI_FOL	Número de folio de asignación
	sCOM_COD_SOFT	Código de serie Softtek
DASIGNACION	dASI_FECH	Fecha de asignación
	dASI_FECH_DEV	Fecha de devolución

DBITMAN	sCOM_COD_SOFT	Código de serie Softtek
	sNUM_MTTO	Número de mantenimiento
	nCAT_INV_CVE	Clave de inventario según catálogo
	dBIT_FECH_ING	Fecha de ingreso en la bitácora
	dBIT_FECH_FIN	Fecha de realización del mantenimiento.
	sBIT_COMENT	Comentarios en la bitácora
	nCAT_STA_CVE	Clave del estatus según catálogo
	sEQU_NPAR	Código de partes de un equipo
DCLASCARACTCOMPCPU	sCMP_CPU_NPAR	Código de partes de una computadora
	nCAT_INV_CVE	Clave de inventario según catálogo
	nCVE_CARACT	Clave de características
	nCAT_STA_CVE	Clave de catálogo de status de equipo
	sCOM_COD_SOFT	Código de serie Softtek
	sEQU_NPAR	Código de partes de un equipo
DFACPROVCOMPCPU	sCMP_CPU_NPAR	Código de partes de una computadora
	nCAT_INV_CVE	Clave de inventario según catálogo
	nCVE_CARACT	Clave de características
	sFAC_FOLIO	Número de folio de la factura
	nPROV_CVE	Clave del proveedor
DLOCCARACTEQU	sCOM_COD_SOFT	Código de serie Softtek
	nCAT_INV_CVE	Clave de inventario según catálogo
	nCVE_CARACT	Clave de características
	nASI_FOL	Número de folio de asignación
	nCAT_UBI_CVE	Clave de la ubicación según catálogo
	nCAT_MARC_CVE	Clave de marca según catálogo
	sEQU_PER	Persona
	sEQU_NPAR	Número de partes del equipo
	nCAT_STA_CVE	Clave del estatus según catálogo
	sFAC_FOLIO	Número de folio de la factura
	sEQU_COMENT	Comentarios referentes al equipo

	sCOM_COD_SOFT	Código de serie Softtek
DMODPROVEQU	sEQU_MOD	Modelo del equipo
	nPROV_CVE	Clave de proveedores
	sFAC_FOLIO	Número de folio de la factura
	nPROV_CVE	Clave del proveedor
	nEQU_PREC	Precio del equipo
DFACTURA	dFAC_FECH	Fecha de la factura
	dFAC_FECH_GARAN	Fecha de garantía
	dFAC_SEG_FECH_INI	Fecha de inicio de seguro
	dFAC_SEG_FECH_FIN	Fecha de fin de seguro
	sSEG_NOM	Nombre
DSEGURIDAD	sSEG_PASS	Pasaporte
	nPERF_CVE	Clave del perfil
	nASI_FOL	Número de folio de asignación
	sSOL_PROY_ID	Identificador del proyecto solicitante
	sSOL_PROY_DESC	Descripción del proyecto solicitante
	nCAT_SEC_CVE	Clave del sector según catálogo
	nCAT_STA_CVE	Clave del estatus según catálogo
DFOLSOL	sSOL_ASI_IS	Asignación de la solicitud
	sSOL_ASI_NOM	Nombre de la asignación solicitada
	sSOL_TEL	Teléfono del solicitante
	nSOL_CANTIDAD	Cantidad solicitada
	nSOL_COMENT	Comentarios de la solicitud
	nASI_FOL	Número de folio de asignación
DPERSOL	nSOL_PERSONA	Persona que solicita

	nUSER	Número de usuario
	sCOM_COD_SOFT	Código de serie Softtek
	sFAC_FOLIO	Número de folio de la factura
	nPROV_CVE	Clave del proveedor
	sPROV_DESC	Descripción del proveedor
TREPINVGAREQU	sCAT_INV_DESC	Descripción de inventario según catálogo
	sCAT_STA_DESC	Descripción del estatus según catálogo
	dFAC_FECH	Fecha de la factura
	dFAC_FECH_GARAN	Fecha de garantía
	dFAC_SEG_FECH_INI	Fecha de inicio de seguro
	dFAC_SEG_FECH_FIN	Fecha de fin de seguro
	nUSER	Número de usuario
	nASI_FOL	Número de folio de asignación
	nCAT_INV_CVE	Clave de inventario según catálogo
	nCAT_STA_CVE	Clave del estatus según catálogo
	sEQU_PER	Persona
	dASI_FECH	Fecha de asignación
TREPASIGDEVEQU	dASI_FECH_DEV	Fecha de devolución
	sSOL_PROY_ID	Identificador del proyecto solicitante
	sSOL_PROY_DESC	Descripción del proyecto solicitante
	nCAT_SEC_CVE	Clave del sector según catálogo
	sCAT_SEC_DESC	Descripción de sector según catálogo
	sSOL_ASI_IS	Asignación de la solicitud
	sSOL_ASI_NOM	Nombre de la asignación solicitada

4.1.4. Descripción de las operaciones y atributos

A continuación se describirán las características referentes a las operaciones (métodos) y atributos (objetos) pertenecientes a las clases del dominio del problema que se usarán en el sistema de información, con lo cuál se tendrá una exposición de lo que significan dichas operaciones y atributos dentro del contexto del modelado del problema para el sistema. Para tal descripción, se pondrá el nombre de la clase y enseguida sus operaciones argumentando por que objetos son llamadas y cual es su funcionalidad de dichas operaciones. (Establecimiento, de convenciones y reglas. Según IEEE Estándar 730 y 983. Documentación. Estándares, prácticas y convenciones, Pressman 1995 Aplicación de métodos técnicos. Stamm 1981 Metodología de ingeniería del software. ISO 9000 –3 Actividades del ciclo de vida (Diseño e implantación)).

Clase	Método	Llamado por:	Acción
CCAPTIPOCOMPCPU	Coloca_nCAT_INV_CVE()	nCAT_INV_CVE	Coloca el número de clave en el catálogo Inventario
	Coloca_nCVE_CARACT()	nCVE_CARACT	Coloca el número de clave de característica en el catálogo Características
	Coloca_sCAT_COMP_TIPO()	sCAT_COMP_TIPO	Coloca el tipo en el catálogo de Componentes
	Coloca_sCAT_COMP_CAP()	sCAT_COMP_CAP	Coloca la capacidad en el catálogo de Componentes
	Coloca_sCAT_COMP_DESC()	sCAT_COMP_DESC	Coloca la descripción en el catálogo de Componentes
CVELMARCCOMPCPU	Coloca_nCAT_INV_CVE()	nCAT_INV_CVE	Coloca el número de clave en el catálogo Inventario
	Coloca_nCVE_CARACT()	nCVE_CARACT	Coloca el número de clave de característica en el catálogo Características
	Coloca_sCAT_COMP_VEL()	sCAT_COMP_VEL	Coloca la velocidad en el catálogo de Componentes
	Coloca_sCAT_COMP_MCADDRESS()	sCAT_COMP_MCADDRESS	Coloca marca y dirección en el catálogo de Componentes
CEQUIPO	Coloca_nCAT_INV_CVE()	nCAT_INV_CVE	Coloca el número de clave en el catálogo Inventario
	Coloca_nCVE_CARACT()	nCVE_CARACT	Coloca el número de clave de característica en el catálogo Características
	Coloca_nCAT_MARC_CVE()	nCAT_MARC_CVE	Coloca la clave en el catálogo de Marcas
	Coloca_sCAT_EQP_TIPO()	sCAT_EQP_TIPO	Coloca el tipo en el catálogo de Equipo
	Coloca_sCAT_EQP_DESC()	sCAT_EQP_DESC	Coloca la descripción en el catálogo de Equipo
CESTATUS	Coloca_nCAT_STA_CVE()	nCAT_STA_CVE	Coloca el número de clave en el catálogo de Estatus
	Coloca_sCAT_STA_DESC()	sCAT_STA_DESC	Coloca la descripción en el catálogo de Estatus
CINVENTARIO	Coloca_nCAT_INV_CVE()	nCAT_INV_CVE	Coloca el número de clave en el catálogo Inventario
	Coloca_sCAT_INV_DESC()	sCAT_INV_DESC	Coloca la descripción en el catálogo Inventario
CMARCA	Coloca_nCAT_MARC_CVE( ) Coloca_sCAT_MARC_DESC( )	nCAT_MARC_CVE	Coloca un campo de información para la clave de marca según catálogo.
		sCAT_MARC_DESC	Coloca un campo de información para la descripción de marca según catálogo.
CPERFIL	Coloca_nPERF_CVE( )	nPERF_CVE	Coloca un campo de información para la clave del perfil.
	Coloca_sPERF_DESC( )	sPERF_DESC	Coloca un campo de información para la descripción del perfil.
CPROVEEDORES	Coloca_nPROV_CVE( )	nPROV_CVE	Coloca un campo de información para la clave de proveedores.
	Coloca_sPROV_DESC( )	sPROV_DESC	Coloca un campo de información para la descripción de proveedores.
CSECTOR	Coloca_nCAT_SEC_CVE( )	nCAT_SEC_CVE	Coloca un campo de información para la clave de sector según catálogo.
	Coloca_sCAT_SEC_DESC( )	sCAT_SEC_DESC	Coloca un campo de información para la descripción de sector según catálogo.

CUBICACION	Coloca_nCAT_UBI_CVE( )	nCAT_UBI_CVE	Coloca un campo de información para la clave de la ubicación según catálogo.
	Coloca_sCAT_UBI_DESC( )	sCAT_UBI_DESC	Coloca un campo de información para la descripción de la ubicación según catálogo.
DASIGNACION	Coloca nASI_FOL()	nASI_FOL	Coloca el folio de asignación en asignación
	Coloca sCOM_COD_SOFT()	sCOM_COD_SOFT	Coloca el código de serie softtek en asignación
	Coloca sASI_FECH()	sASI_FECH	Coloca la fecha de asignación en asignación
	Coloca sASI_FECH_DEV()	sASI_FECH_DEV	Coloca la fecha de devolución en asignación
DBITMAN	Coloca sCOM_COD_SOFT()	sCOM_COD_SOFT	Coloca código serie softtek en la bitacora
	Coloca sNUM_MTTO()	sNUM_MTTO	Coloca el número de mantenimiento en la bitacora
	Coloca nCAT_ INV_CVE()	nCAT_ INV_CVE	Coloca la clave del inventario según catalogo en la bitacora
	Coloca dBIT_ FECH_ING()	dBIT_ FECH_ING	Coloca la fecha de ingreso en la bitacora
	Coloca dBIT_ FECH_FIN()	dBIT_ FECH_FIN	Coloca la fecha de realización del mantenimiento en la bitacora
	Coloca sBIT_ COMENT()	sBIT_ COMENT	Coloca comentario en la bitacora
	Coloca nCAT_ STA_CVE()	nCAT_ STA_CVE	Coloca la clave del estatus en la bitacora
	Coloca sEQU_ NPAR()	sEQU_ NPAR	Coloca codigo de partes del equipo en bitacora
DCLASCARACTCOMPCPU	Coloca sCOM_CPU_NPAR()	sCOM_CPU_NPAR	Coloca el código de partes de una computadora
	Coloca nCAT_ INV_CVE()	nCAT_ INV_CVE	Coloca la clave del inventario según catalogo
	Coloca nCVE_ CARACT()	nCVE_ CARACT	Coloca la clave de características del componente
	Coloca nCAT_ STA_CVE()	nCAT_ STA_CVE	Coloca la clave del estatus del equipo
	Coloca sCOM_COD_SOFT()	sCOM_COD_SOFT	Coloca el codigo de serie softtek
	Coloca sEQU_NPAR()	sEQU_NPAR()	Coloca el codigo de partes de un equipo
DFACPROVCOMPCPU	Coloca sCOM_CPU_NPAR()	sCOM_CPU_NPAR	Coloca el código de partes de una computadora
	Coloca nCAT_ INV_CVE()	nCAT_ INV_CVE	Coloca la clave del inventario según catalogo
	Coloca nCVE_ CARACT()	nCVE_ CARACT	Coloca la clave de características del componente
	Coloca nFAC_ FOLIO()	nFAC_ FOLIO	Coloca el número de folio de la factura
	Coloca nPROV_ CVE()	nPROV_ CVE	Coloca la clave del proveedor
DLOCCARACTEQU	Coloca_sCOM_COD_SOFT()	sCOM_COD_SOFT	Coloca el código de serie de Softtek
	Coloca_nCAT_INV_CVE()	nCAT_INV_CVE	Coloca la clave del inventario según catalogo
	Coloca_nCVE_CARACT()	nCVE_CARACT	Coloca la clave de características
	Coloca_nASI_FOL()	nASI_FOL	Coloca el número de folio de asignación
	Coloca_nCAT_UBI_CVE()	nCAT_UBI_CVE	Coloca la clave de ubicación según catálogo
	Coloca_nCAT_MARC_CVE()	nCAT_MARC_CVE	Coloca la clave de marca según catálogo
	Coloca_sEQU_PER()	sEQU_PER	Coloca el nombre del usuario
	Coloca_sEQU_MOD()	sEQU_MOD	Coloca el modelo de equipo
	Coloca_sEQU_NPAR()	sEQU_NPAR	Coloca el número de parte del equipo
	Coloca_nCAT_STA_CVE()	nCAT_STA_CVE	Coloca la clave del estatus según catálogo
	Coloca_sFAC_FOLIO()	sFAC_FOLIO	Coloca el número de folio de factura
	Coloca_sEQU_COMENT()	sEQU_COMENT	Coloca comentario sobre el equipo

DMODPROVEQU	Coloca_sCOM_COD_SOFT() Coloca_sEQU_MOD() Coloca_nPROV_CVE()	sCOM_COD_SOFT sEQU_MOD nPROV_CVE	Coloca el código de serie de Softtek Coloca el modelo del equipo Coloca la clave del proveedor
DFACTURA	Coloca_sFAC_FOLIO() Coloca_nPROV_CVE() Coloca_nEQU_PREC() Coloca_dFAC_FECH() Coloca_dFAC_FECH_GARAN() Coloca_dFAC_SEG_FECH_INI() Coloca_dFAC_SEG_FECH_FIN()	sFAC_FOLIO nPROV_CVE nEQU_PREC dFAC_FECH dFAC_FECH_GARAN dFAC_SEG_FECH_INI dFAC_SEG_FECH_FIN	Coloca el número de folio de factura Coloca la clave del proveedor Coloca el precio de equipo Coloca la fecha de factura Coloca la fecha de garantía Coloca la fecha de inicio del seguro Coloca la fecha de fin del seguro
DSEGURIDAD	Coloca_sSEG_NOM() Coloca_sSEG_PASS() Coloca_nPERF_CVE()	sSEG_NOM sSEG_PASS nPERF_CVE	Coloca el nombre de usuario Coloca password Coloca la clave de perfil de usuario
DFOLSOL	Coloca_nASI_FOL( ) Coloca_sSOL_PROY_ID ( ) Coloca_sSOL_PROY_DESC( ) Coloca_nCAT_SEC_CVE ( ) Coloca_nCAT_STA_CVE ( ) Coloca_sSOL_ASI_IS( ) Coloca_sSOL_ASI_NOM( ) Coloca_sSOL_TEL( ) Coloca_nSOL_CANTIDAD( ) Coloca_nSOL_COMENT( )	nASI_FOL sSOL_PROY_ID sSOL_PROY_DESC nCAT_SEC_CVE nCAT_STA_CVE sSOL_ASI_IS sSOL_ASI_NOM sSOL_TEL nSOL_CANTIDAD nSOL_COMENT	Coloca el numero de Folio de asignación Coloca el identificador del proyecto solicitante Coloca la descripción del proyecto solicitante Coloca la clave del sector según catálogo Coloca la clave del sector según catálogo Coloca la asignación de la solicitud Coloca el Nombre de la asignación solicitada Coloca el teléfono del solicitante Coloca la cantidad solicitada Coloca los Comentarios de la solicitud
DPERSOL	Coloca_nASI_FOL( ) Coloca_sSOL_PERSONA( )	nASI_FOL sSOL_PERSONA	Coloca el numero de folio de asignación Coloca el nombre de la persona que solicita
TREPINVGAREQU	Coloca_nUSER( ) Coloca_sCOM_COD_SOFT( ) Coloca_sFAC_FOLIO( ) Coloca_nPROV_CVE( ) Coloca_sPROV_DESC( ) Coloca_sCAT_INV_DESC( ) Coloca_sCAT_STA_DESC( ) Coloca_dFAC_FECH( ) Coloca_dFAC_FECH_GARAN( ) Coloca_dFAC_SEG_FECH_INI( ) Coloca_dFAC_SEG_FECH_FIN( )	nUSER sCOM_COD_SOFT sFAC_FOLIO nPROV_CVE sPROV_DESC sCAT_INV_DESC sCAT_STA_DESC dFAC_FECH dFAC_FECH_GARAN dFAC_SEG_FECH_INI dFAC_SEG_FECH_FIN	Coloca el numero de usuario Coloca el código de serie Softtek Coloca el numero de folio de la factura Coloca la clave del proveedor Coloca la descripción del proveedor Coloca la descripción de inventario según catálogo Coloca la descripción del estatus según catálogo Coloca la fecha de la factura Coloca la fecha de la garantía Coloca la fecha de inicio de seguro Coloca la fecha de fin de seguro

	Coloca_nUSER( )	nUSER	Coloca el numero de usuario
	Coloca_nASI_FOL ( )	nASI_FOL	Coloca el numero de folio de asignación
	Coloca_nCAT_INV_CVE ( )	nCAT_INV_CVE	Coloca la clave de inventario según catálogo
	Coloca_nCAT_STA_CVE ( )	nCAT_STA_CVE	Coloca la clave del estatus según catálogo
	Coloca_sEQU_PER ( )	sEQU_PER	Coloca el nombre de la persona
TREPASIGDEVEQU	Coloca_dASI_FECH ( )	dASI_FECH	Coloca la fecha de asignación
	Coloca_dASI_FECH_DEV ( )	dASI_FECH_DEV	Coloca la fecha de devolución
	Coloca_nSOL_PROY_ID ( )	nSOL_PROY_ID	Coloca el identificador del proyecto solicitante
	Coloca_sSOL_PROY_DESC ( )	sSOL_PROY_DESC	Coloca la descripción del proyecto solicitante
	Coloca_nCAT_SEC_CVE ( )	nCAT_SEC_CVE	Coloca la clave del sector según catálogo
	Coloca_sCAT_SEC_DESC ( )	sCAT_SEC_DESC	Coloca la descripción de sector según catálogo
	Coloca_sSOL_ASI_IS ( )	sSOL_ASI_IS	Coloca la asignación de la solicitud
	Coloca_sSOL_ASI_NOM ( )	sSOL_ASI_NOM	Coloca el nombre de la asignación solicitada

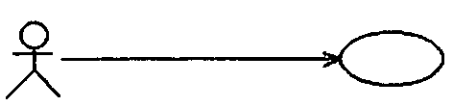
4.1.5. Diagrama de interacción

Un diagrama de interacción es usado para trazar la ruta de ejecución que indica lo que el sistema debe hacer. Un diagrama de interacción es simplemente otra forma de representar un diagrama de objetos. Un diagrama de objetos es usado para mostrar la existencia de objetos y sus relaciones en el diseño lógico de un sistema. (Utilización de métodos, técnicas, herramientas y estándares para el desarrollo del software. Según IEEE Estándar 730 y 983. Herramientas, técnicas, metodologías, Pressman 1995 Aplicación de métodos técnicos. Stamm 1981 Métodología de ingeniería del software. ISO 9000 –3 Actividades del ciclo de vida (Diseño e implantación))

Utilización de métodos y técnicas para el desarrollo del software. Según IEEE Estándar 730 y 983. Herramientas, técnicas y metodologías. Butler Estándares y convenciones.

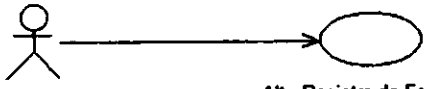
A continuación se procederá a mostrar los diagramas de interacción del sistema de información.

Entrar al sistema



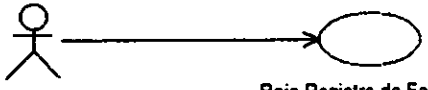
Acceso al sistema para los usuarios autorizados.

Alta Registro de Facturas



Añade nuevas facturas de equipo en el sistema.

Baja Registro de Facturas

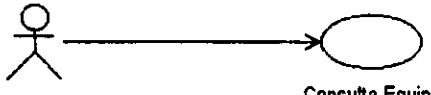


Elimina facturas de equipo en el sistema.

Baja Registro de Equipo

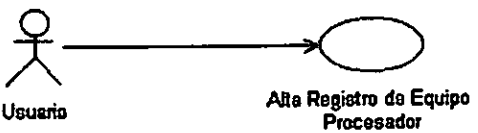
Elimina el registro de equipo del sistema.

Consulta Equipo



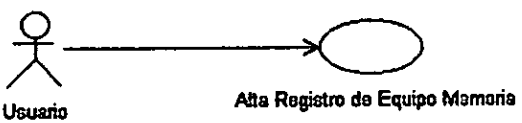
Revisa la existencia de un equipo.

Alta Registro de Equipo Procesador



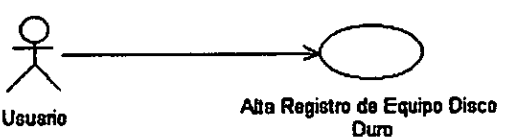
Añade componentes de cpu al sistema.

Alta Registro de Equipo Memoria



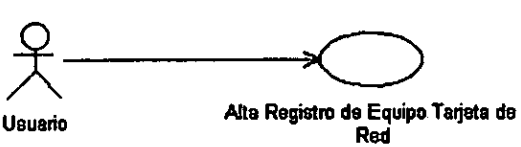
Añade componentes de cpu-memoria al sistema.

Alta Registro de Equipo Disco Duro



Añade componentes de cpu-disco duro al sistema.

Alta Registro de Equipo Tarjeta de Red

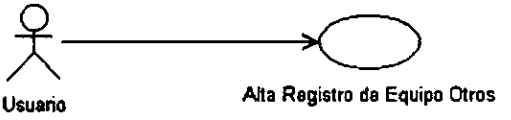


Añade componentes de cpu-tarjeta de red al sistema.

Alta Registro de Equipo Conector

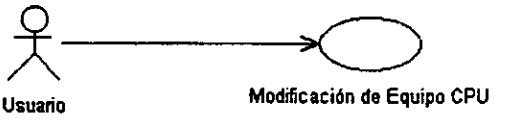
Añade tipos de conectores al sistema.

Alta Registro de Equipo Otros



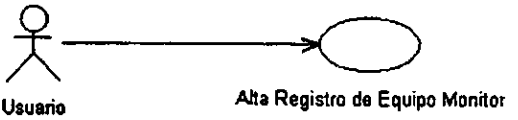
Añade diversos componentes de cpu al sistema.

Modificación de Equipo CPU



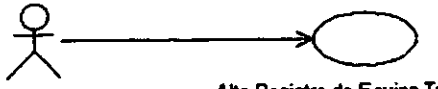
Cambia características de componentes de cpu en el sistema.

Alta Registro de Equipo Monitor



Añade periféricos referentes a monitores al sistema.

Alta Registro de Equipo Teclado y Mouse

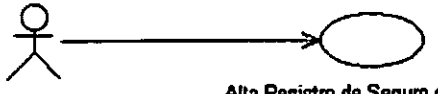


Añade periféricos referentes a teclado y mouse al sistema.

Modificación de Equipo Periférico

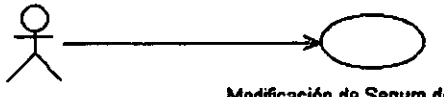
Cambia características de componentes periféricos en el sistema.

Alta Registro de Seguro de Equipo



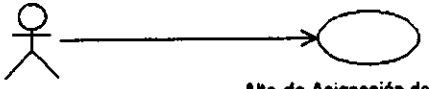
Añade datos del seguro de componentes al sistema.

Modificación de Seguro de Equipo



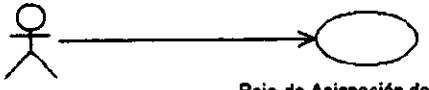
Cambia características de datos del seguro de componentes al sistema.

Alta de Asignación de Equipo



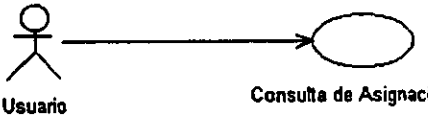
Proporciona un equipo a un usuario por una fecha determinada.

Baja de Asignación de Equipo



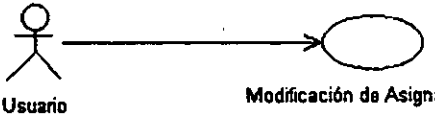
Devolución de equipo en la fecha establecida.

Consulta de Asignaciones



Verifica las asignaciones de los equipos.

Modificación de Asignaciones



Actualiza los cambios en la asignación.

Consulta Status

Genera gráficas por equipo disponible o asignado.

Consulta Sector

Genera gráficas por Sectores.

Consulta Procesador

Genera gráficas por tipo de procesador.

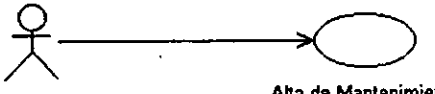
Consulta Bitácora

Genera gráficas por mantenimientos y/o garantías.

Consulta Clave Proyecto

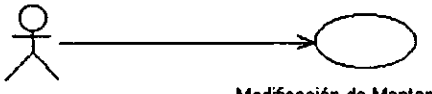
Genera gráficas de equipos asignados por proyectos.

Alta de Mantenimiento



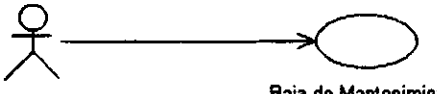
Introducción de un equipo a mantenimiento.

Modificación de Mantenimiento



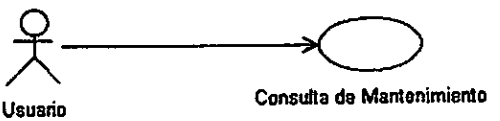
Cambio en los datos de mantenimiento.

Baja de Mantenimiento



Cambio de estatus del equipo para asignación.

Consulta de Mantenimiento

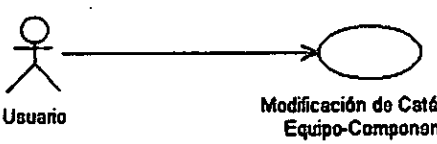


Visualización de equipos en mantenimiento.

Altas de Catálogo Equipo-Componente

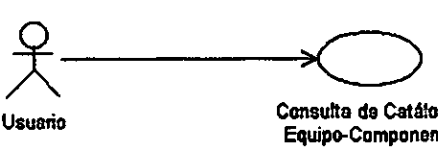
Introduce nuevos artículos en el catálogo equipo-componente.

Modificación de Catálogo Equipo-Componente



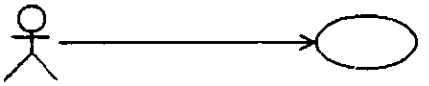
Cambia características de datos del catálogo equipo-componente del sistema.

Consulta de Catálogo Equipo-Componente



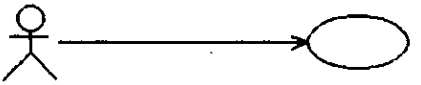
Visualización de equipos en el catálogo equipo-componente.

Baja de Catálogo Equipo-Componente



Elimina artículos del catálogo equipo-componente en el sistema.

Altas de Catálogo Marca



Introduce nuevos registros en el catálogo marca.

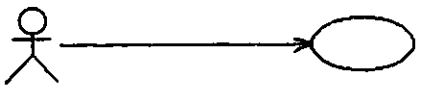
Modificación de Catálogo Marca

Cambia características de datos del catálogo marca del sistema.

Consulta de Catálogo Marca

Visualización de registros de marcas en el catálogo marca.

Baja de Catálogo Marca



Elimina registros del catálogo marca en el sistema.

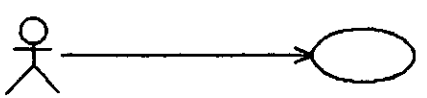
Altas de Catálogo Sector

Introduce nuevos registros en el catálogo sector.

Modificación de Catálogo Sector

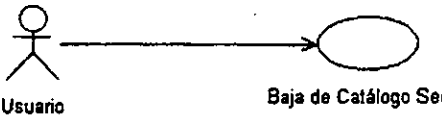
Cambia características de datos del catálogo sector del sistema.

Consulta de Catálogo Sector



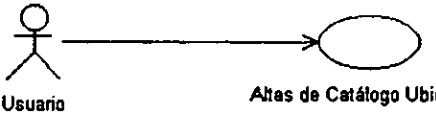
Visualización de registros de sector en el catálogo sector.

Baja de Catálogo Sector



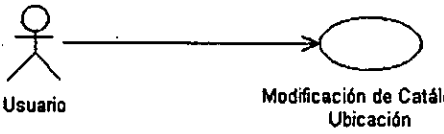
Elimina registros del catálogo sector en el sistema.

Altas de Catálogo Ubicación



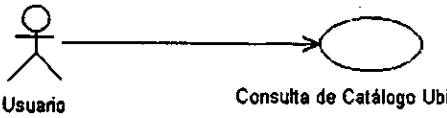
Introduce nuevos elementos en el catálogo ubicación.

Modificación de Catálogo Ubicación



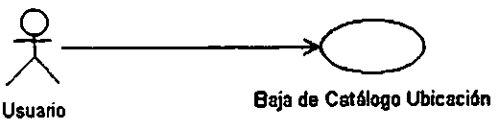
Cambia características de datos del catálogo ubicación del sistema.

Consulta de Catálogo Ubicación



Visualización de registros de ubicación en el catálogo ubicación.

Baja de Catálogo Ubicación



Elimina registros del catálogo ubicación en el sistema.

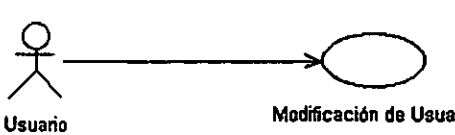
Altas de Usuario

Añade usuarios en el sistema.

Baja de Usuario

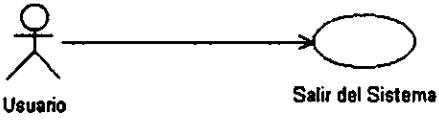
Elimina registros de usuario del sistema.

Modificación de Usuario



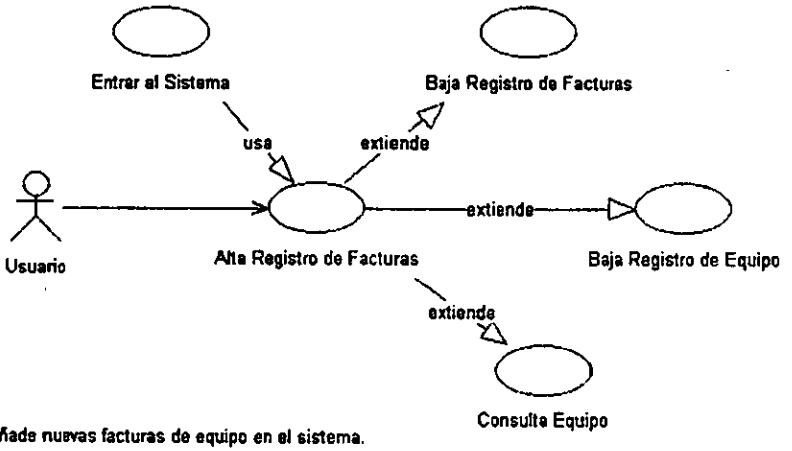
Cambia características de datos del usuario en el sistema.

Salir del Sistema



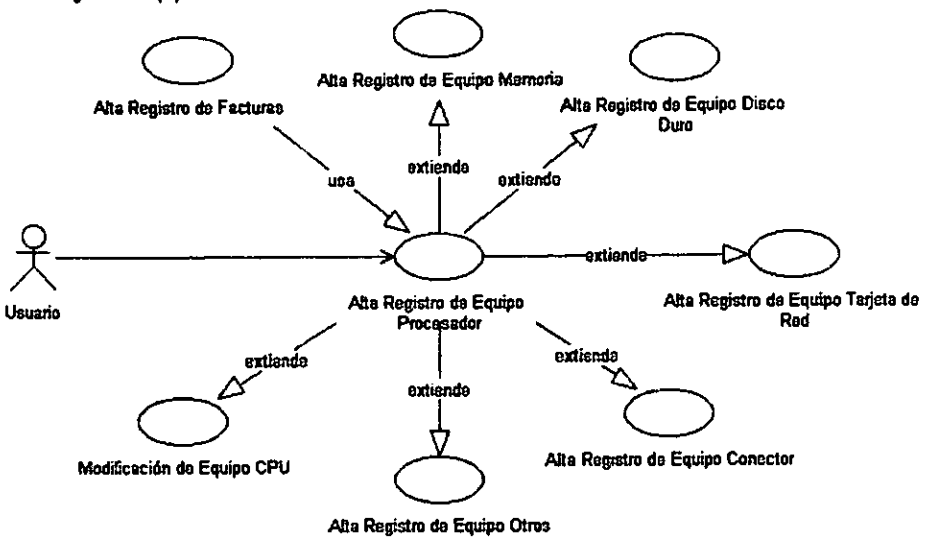
Se desconecta del sistema y se cierra la aplicación.

Alta Registro de Facturas



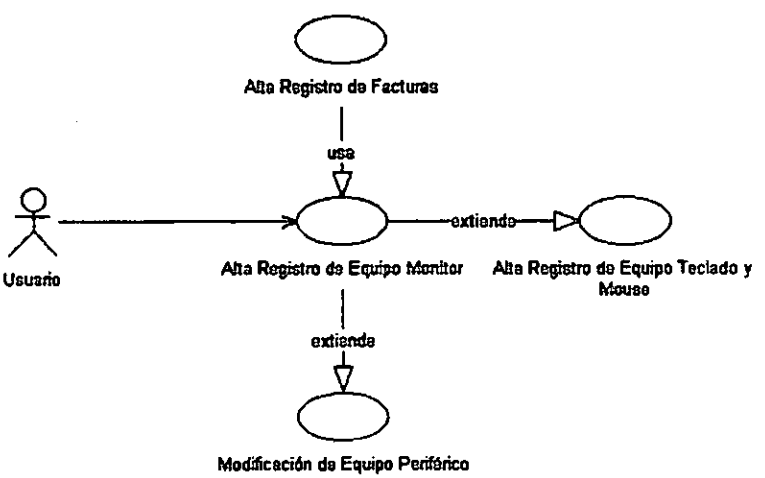
Añade nuevas facturas de equipo en el sistema.

Alta Registro de Equipo Procesador



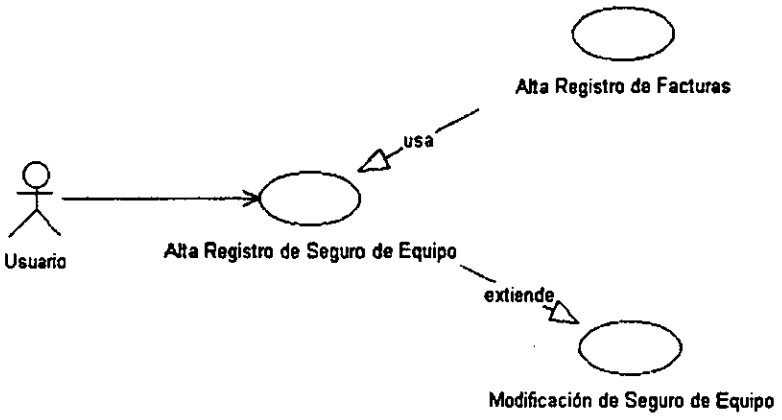
Añade componentes de cpu al sistema.

Alta Registro de Equipo Monitor



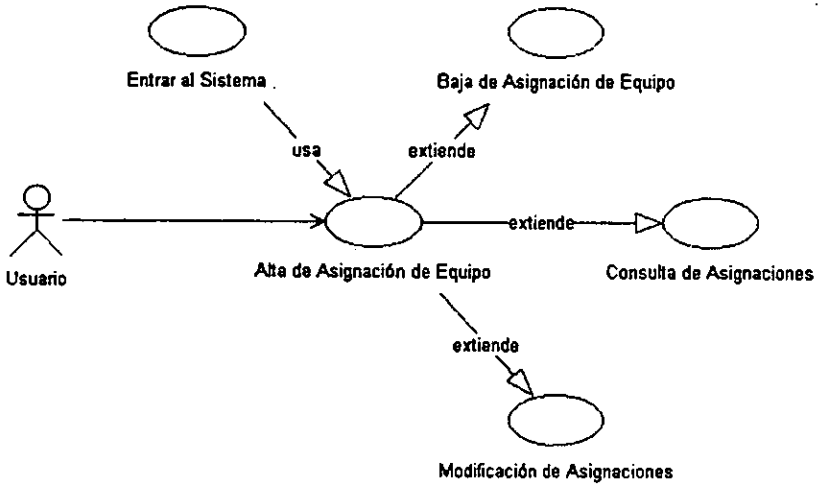
Añade periféricos monitores al sistema.

Alta Registro de Seguro de Equipo



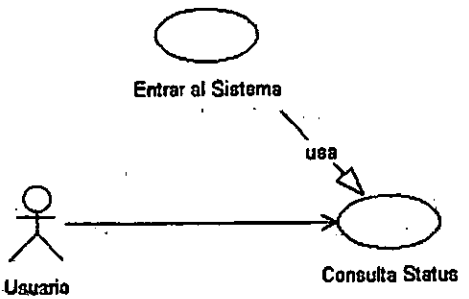
Añade datos del seguro de componentes al sistema.

Alta de Asignación de Equipo



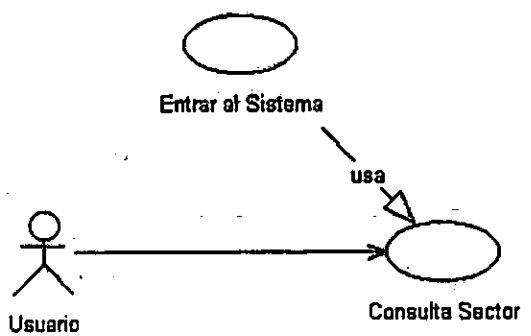
Proporciona un equipo a un usuario por una fecha determinada.

Consulta Status



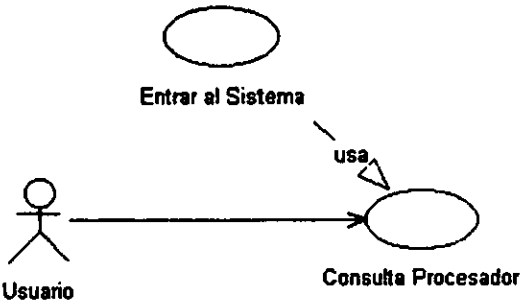
Genera gráficas por equipo disponible o asignado.

Consulta Sector



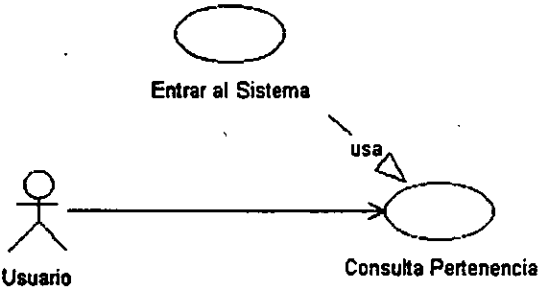
Genera gráficas por Sectores.

Consulta Procesador



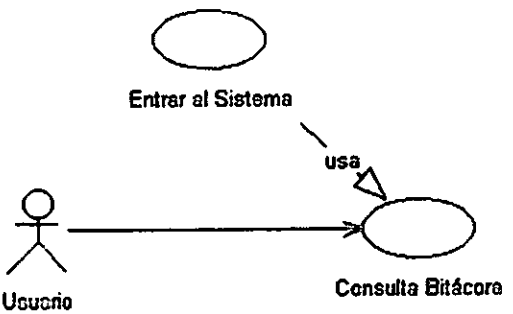
Genera gráficas por tipo de procesador.

Consulta Pertenencia



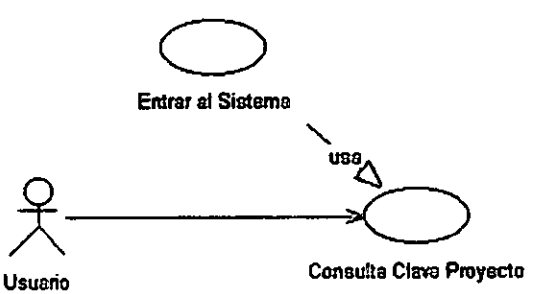
Genera gráficas de asignaciones por usuario.

Consulta Bitácora



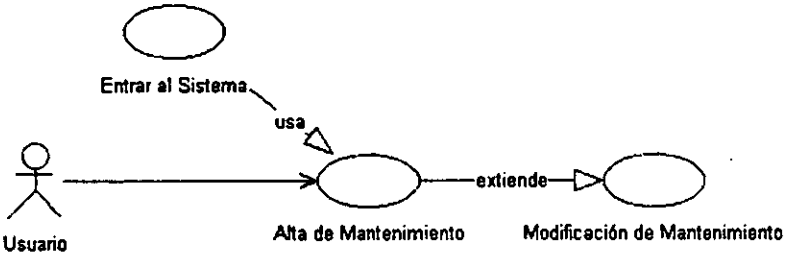
Genera gráficas por mantenimientos y/o garantías.

Consulta Clave Proyecto



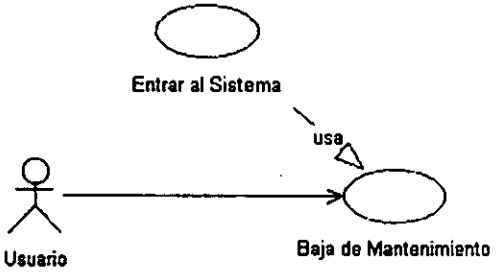
Genera gráficos de equipos asignados por proyectos.

Alta de Mantenimiento



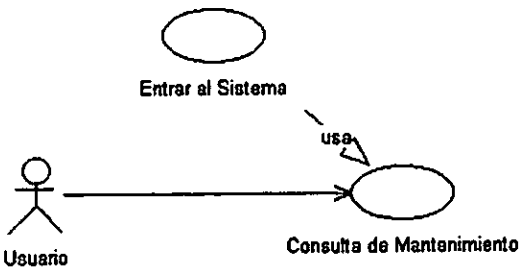
Introducción de un equipo a mantenimiento.

Baja de Mantenimiento



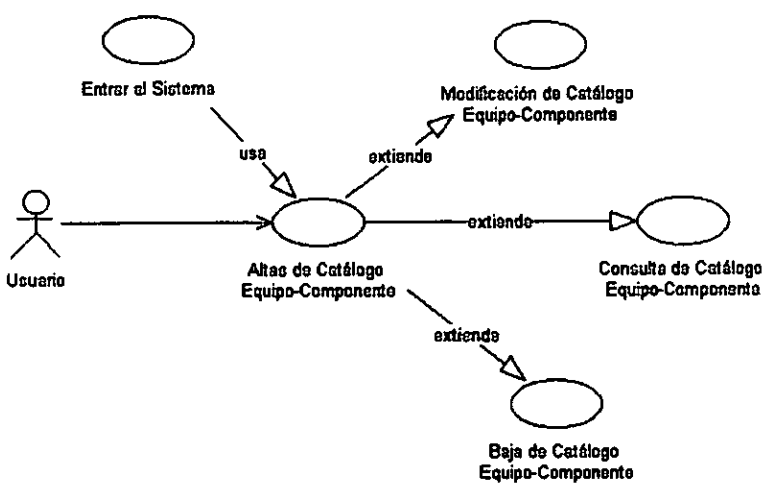
Cambio de estatus del equipo para asignación.

Consulta de Mantenimiento



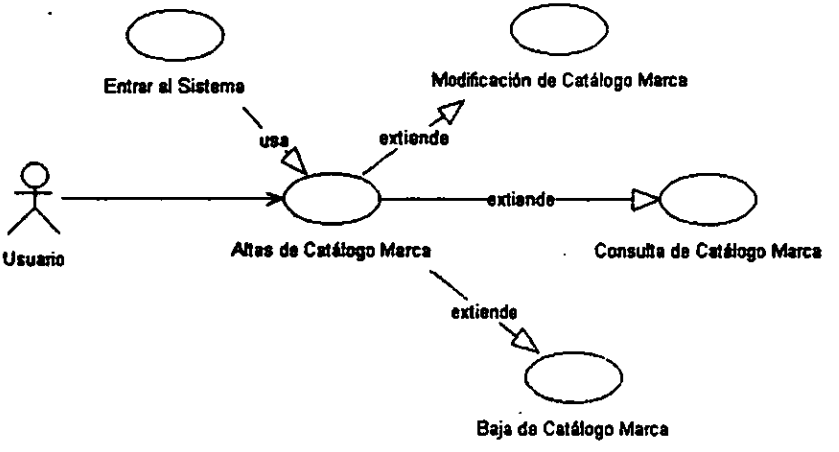
Visualización de equipos en mantenimiento.

Altas de Catálogo Equipo-Componente



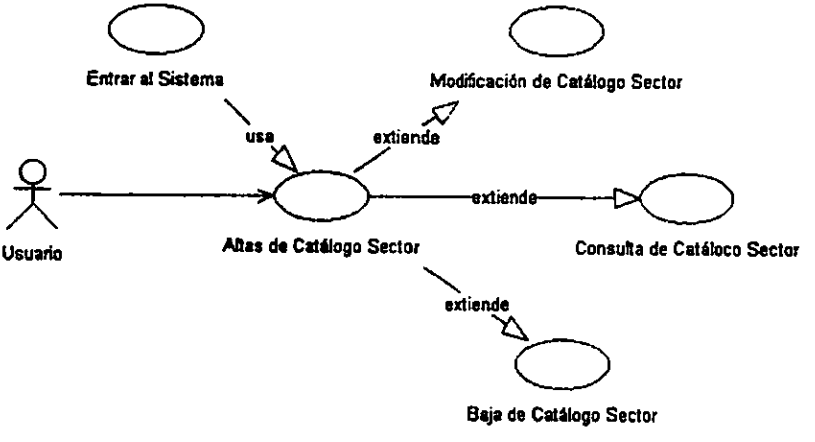
Introduce nuevos artículos en el catálogo equipo-componente.

Altas de Catálogo Marca



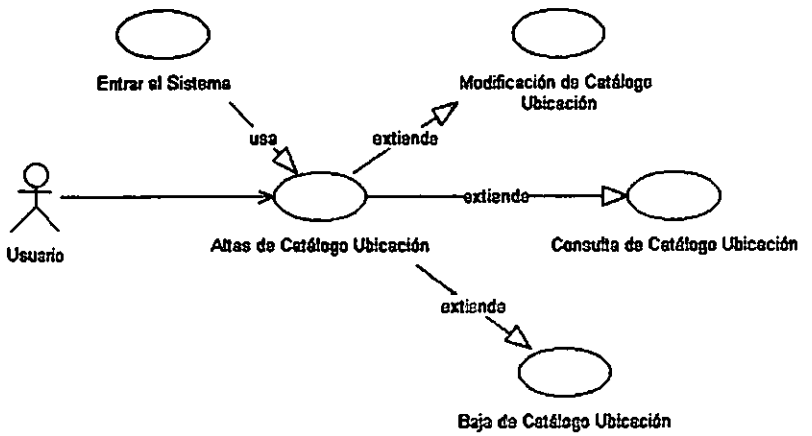
Introduce nuevos registros en el catálogo marca.

Altas de Catálogo Sector



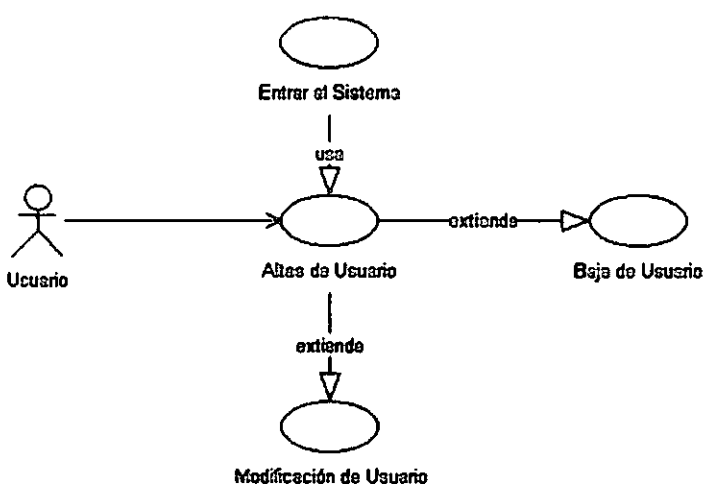
Introduce nuevos elementos en el catálogo sector.

Altas de Catálogo Ubicación

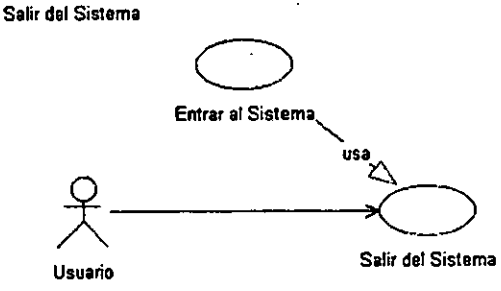


Introduce nuevos elementos en el catálogo ubicación.

Altas de Usuario



Añade usuarios en el sistema.



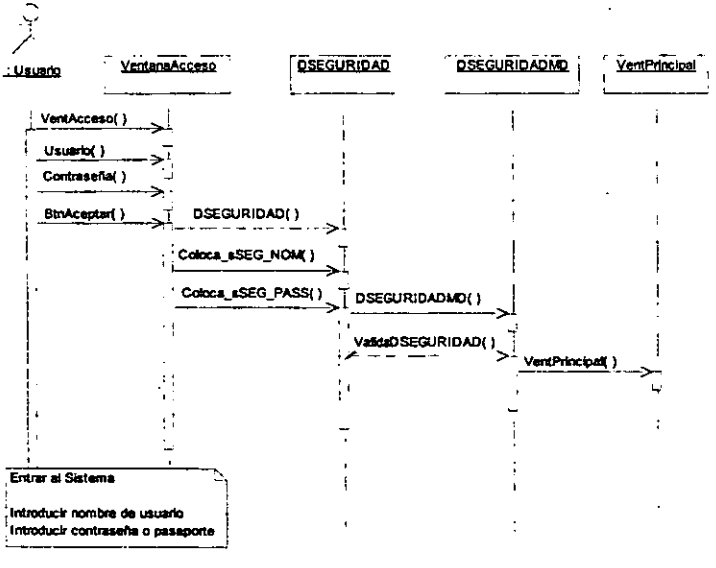
Se desconecta del sistema y se cierra la aplicación

4.1.6. Diagrama de secuencias

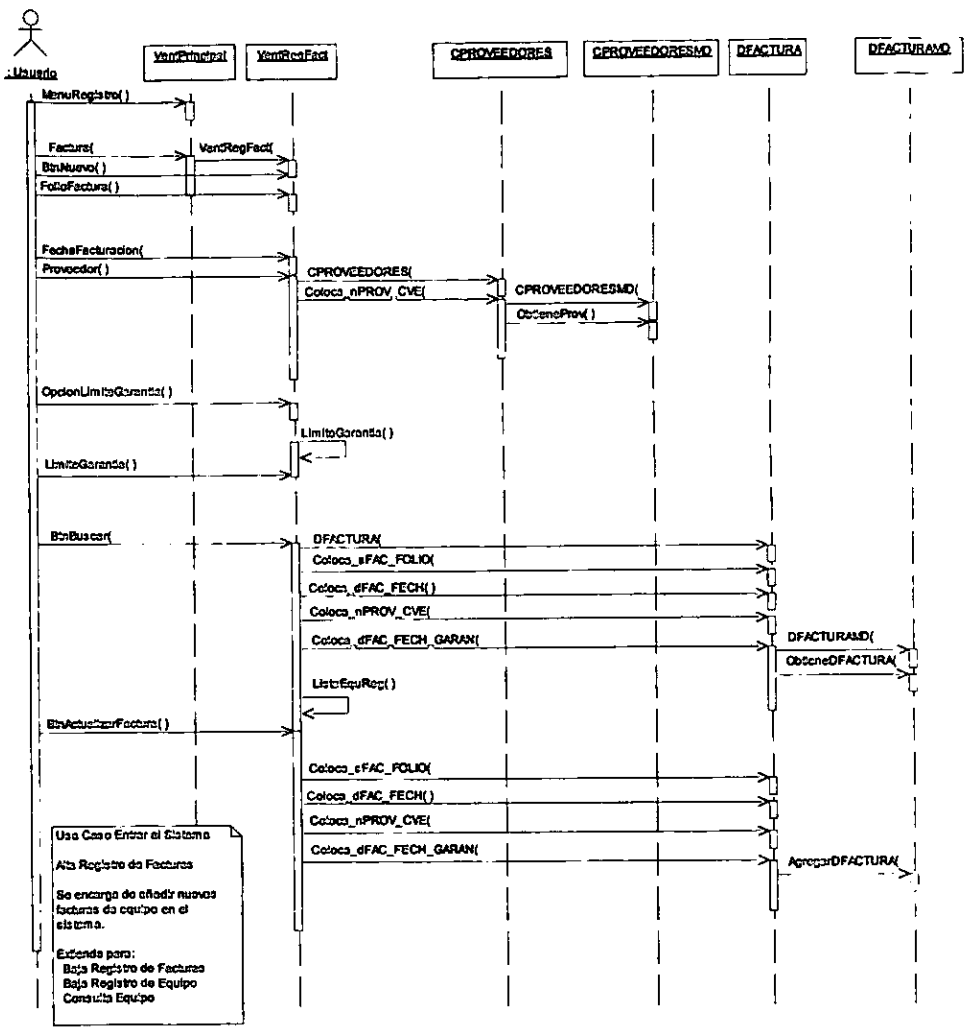
Un diagrama de secuencias se utiliza para definir la lógica de un escenario de un caso de uso. Un diagrama de secuencia sirve para validar casos de uso con la finalidad de comprender la lógica de su aplicación. . (Utilización de estándares y técnicas para el desarrollo del software. Según IEEE Estándar 730 y 983. Herramientas, técnicas y metodologías. Butler Estándares y convenciones.).

A continuación se procederá a mostrar los diagramas de secuencias del sistema de información.

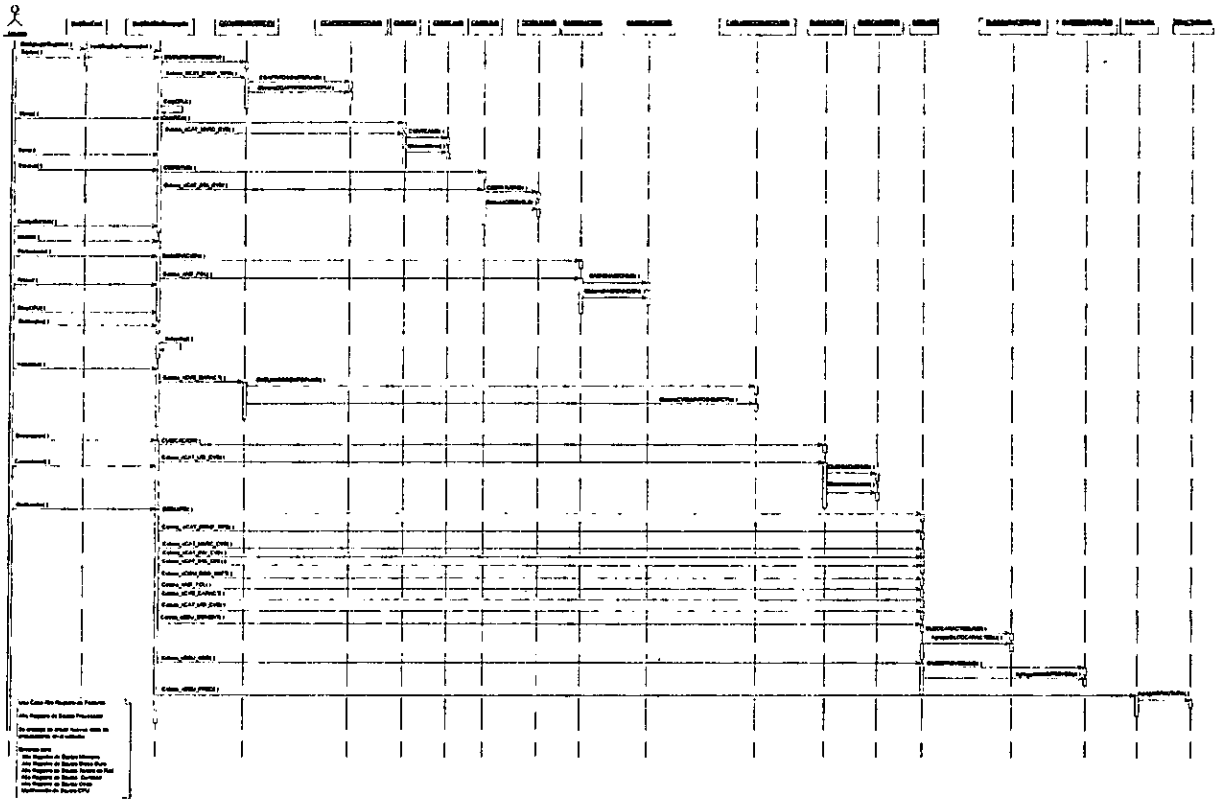
Entrar al Sistema



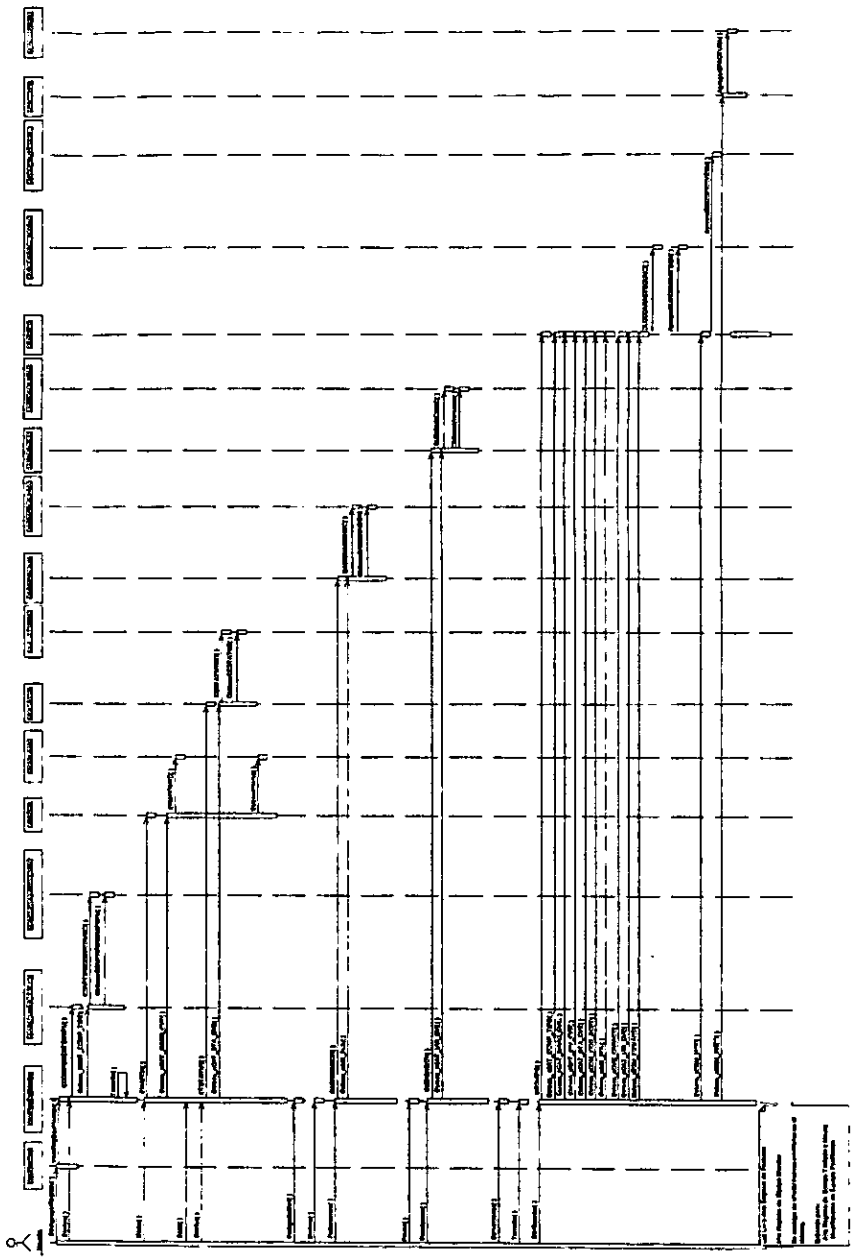
Alta Registro de Facturas



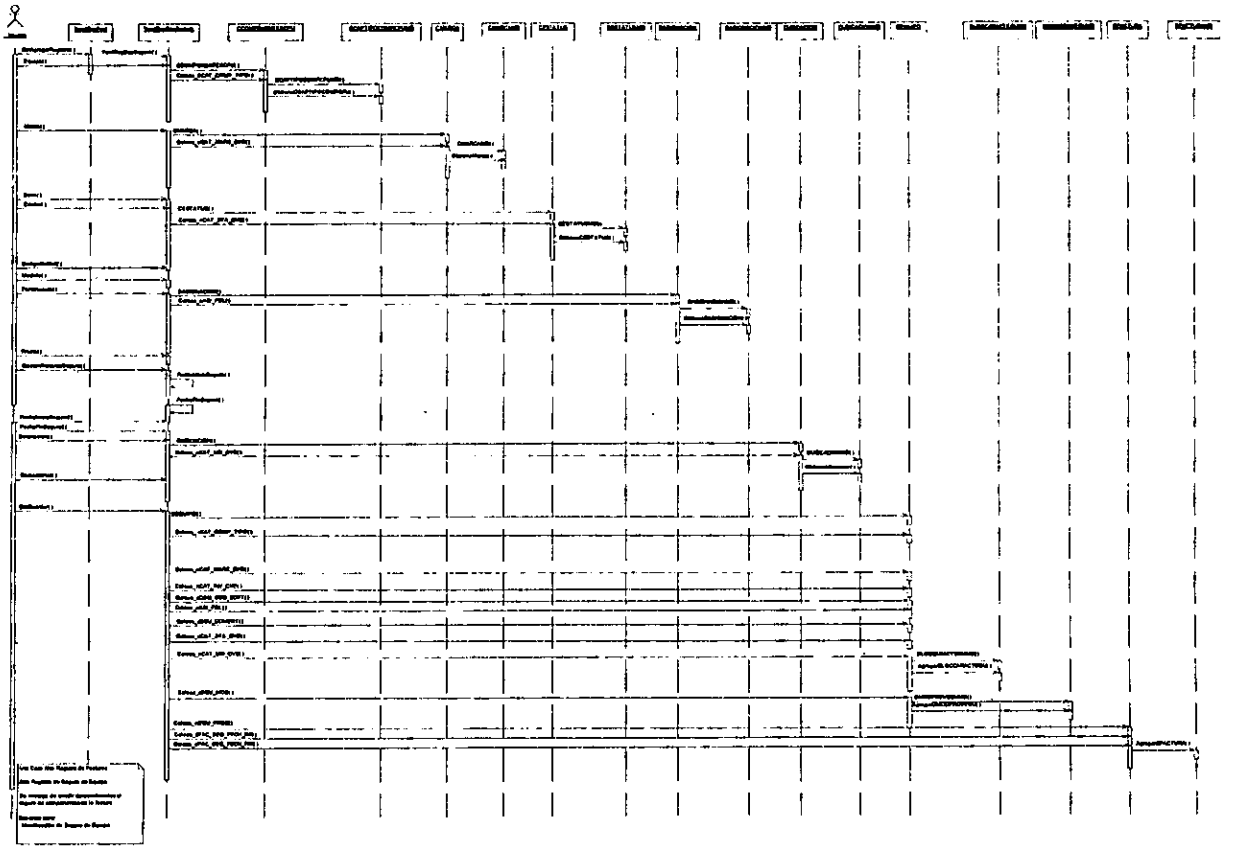
Alta Registro de Equipo Procesador



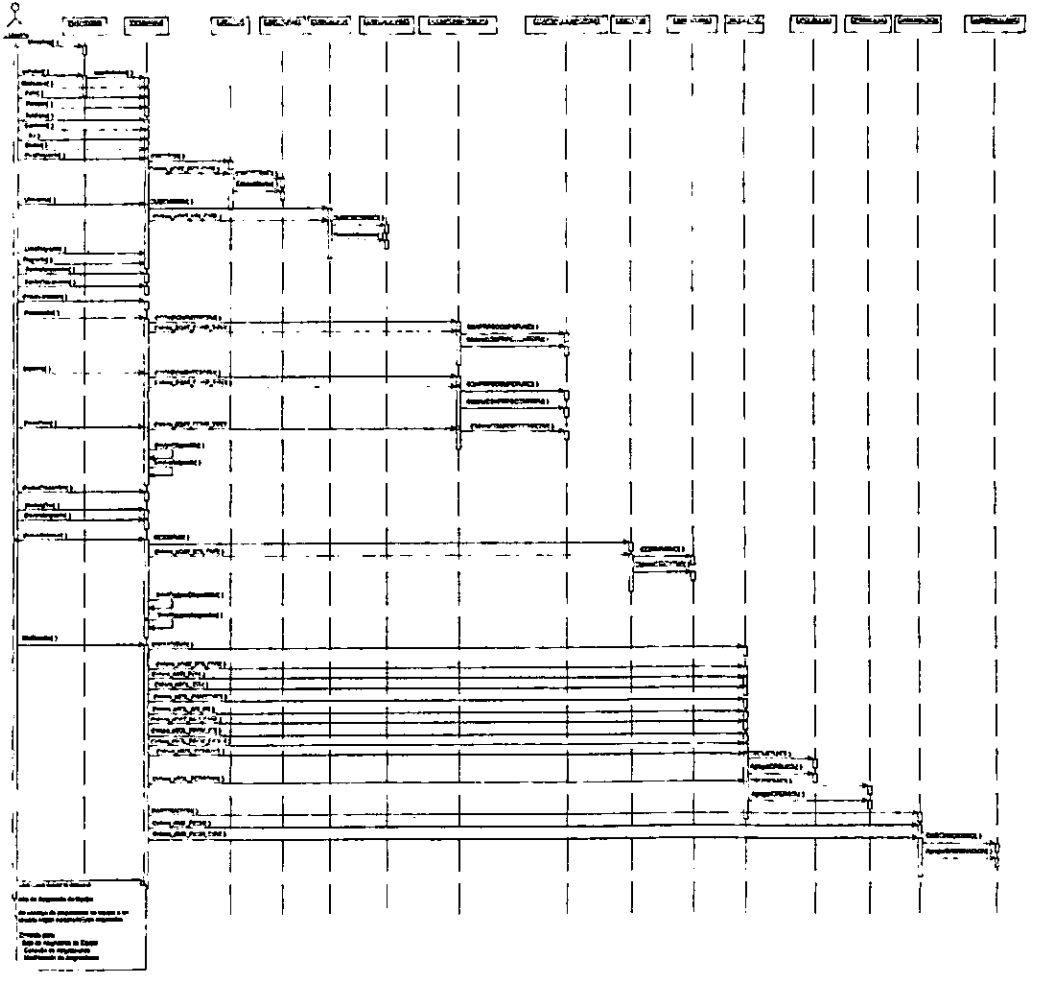
Alta Registro de Equipo Monitor



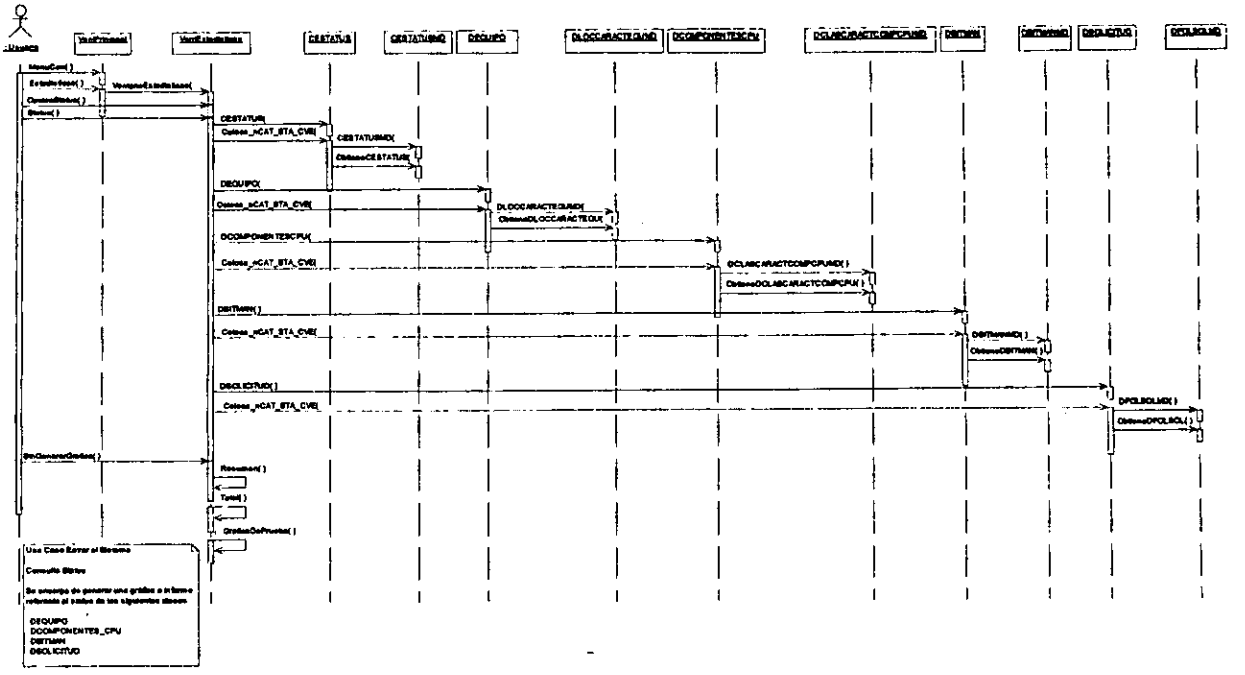
Alta Registro de Seguro de Equipo



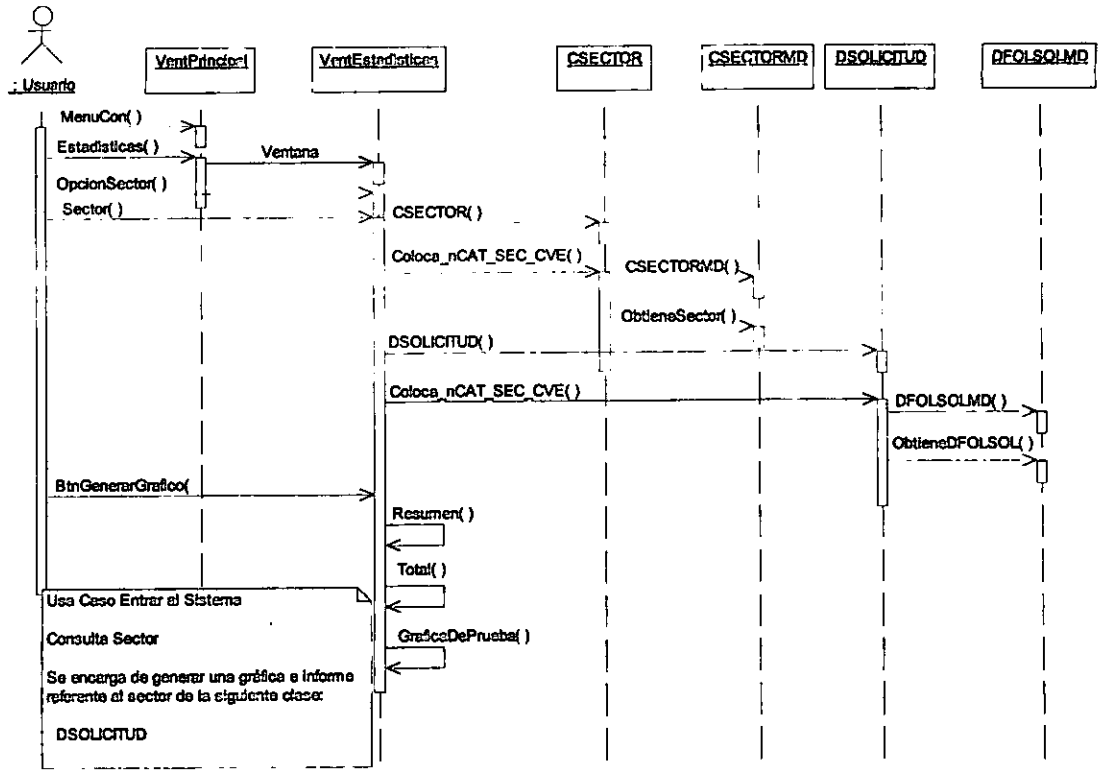
Alta de Asignación de Equipo



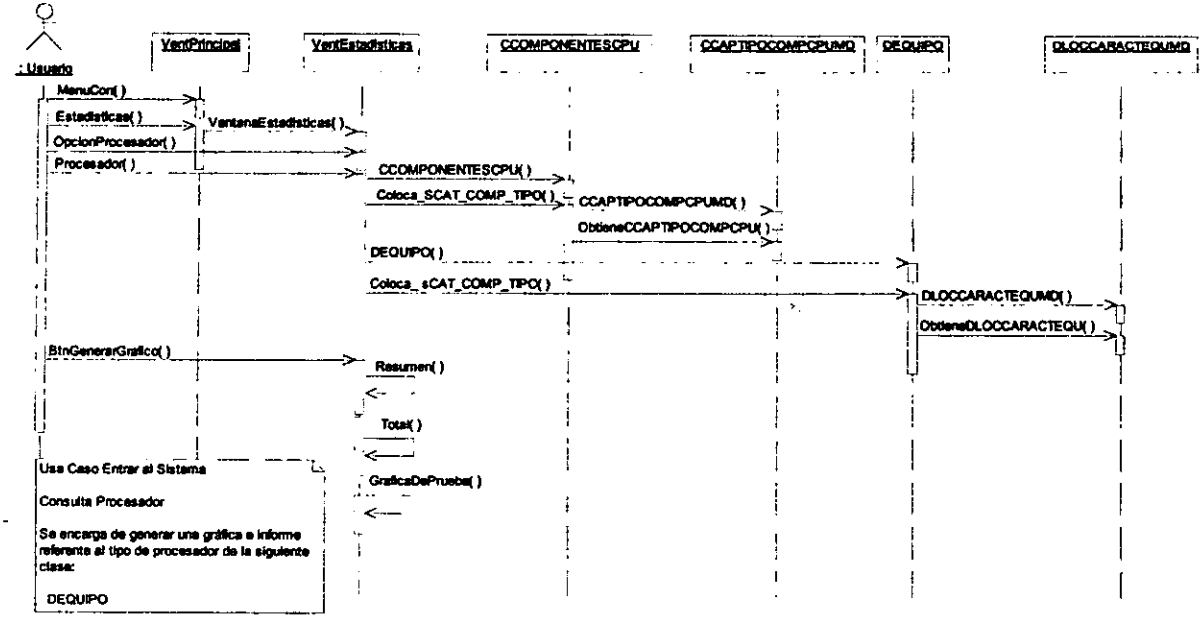
Consulta Status



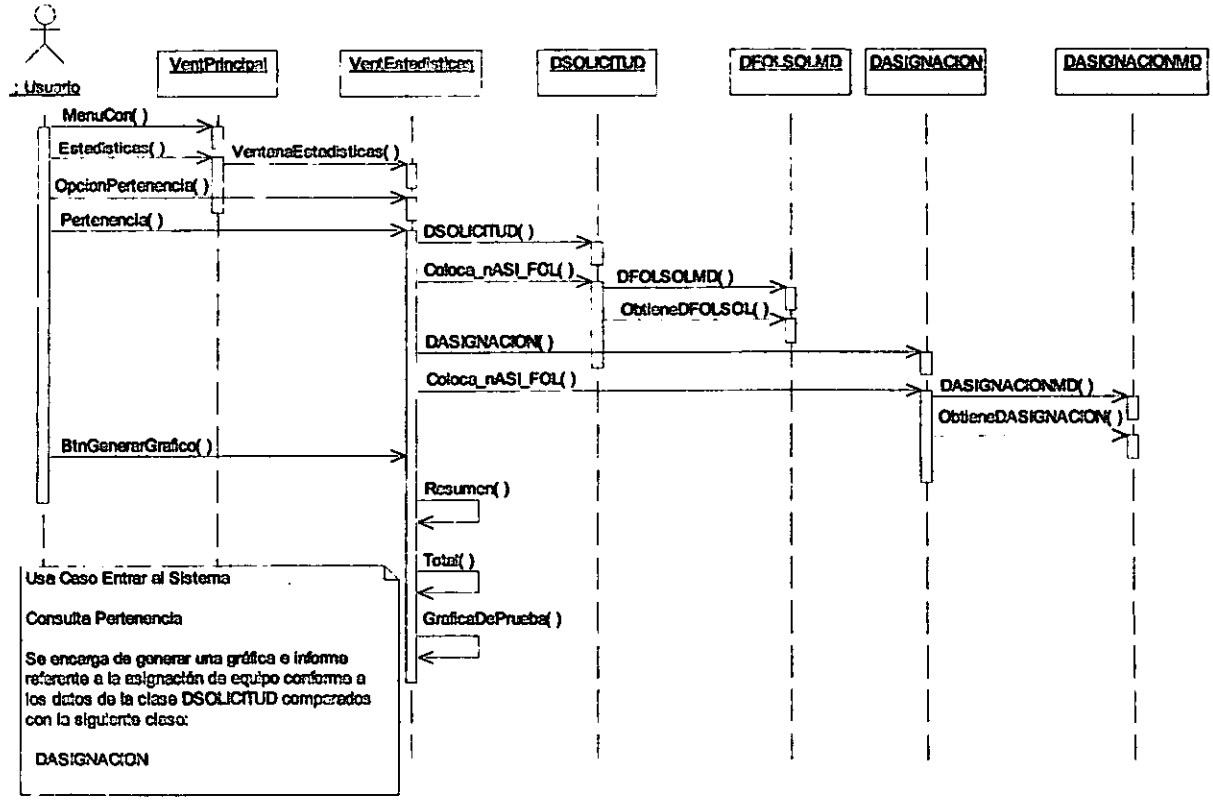
Consulta Sector



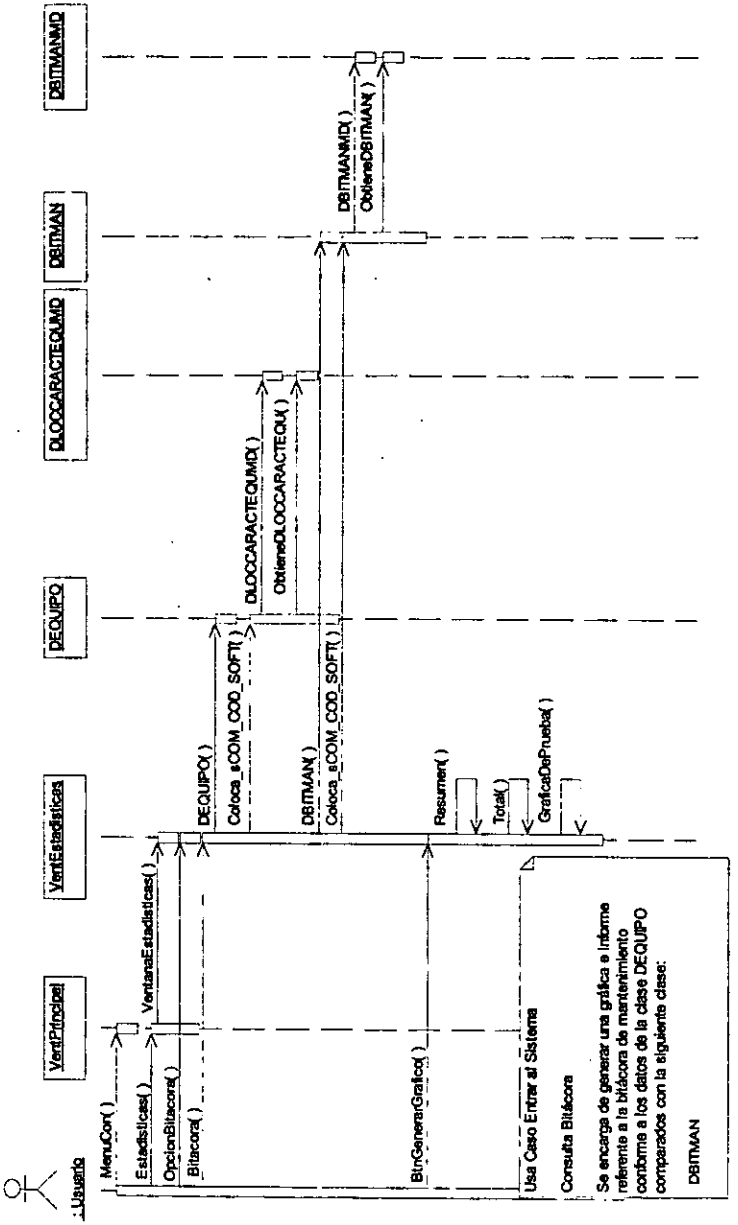
Consulta Procesador

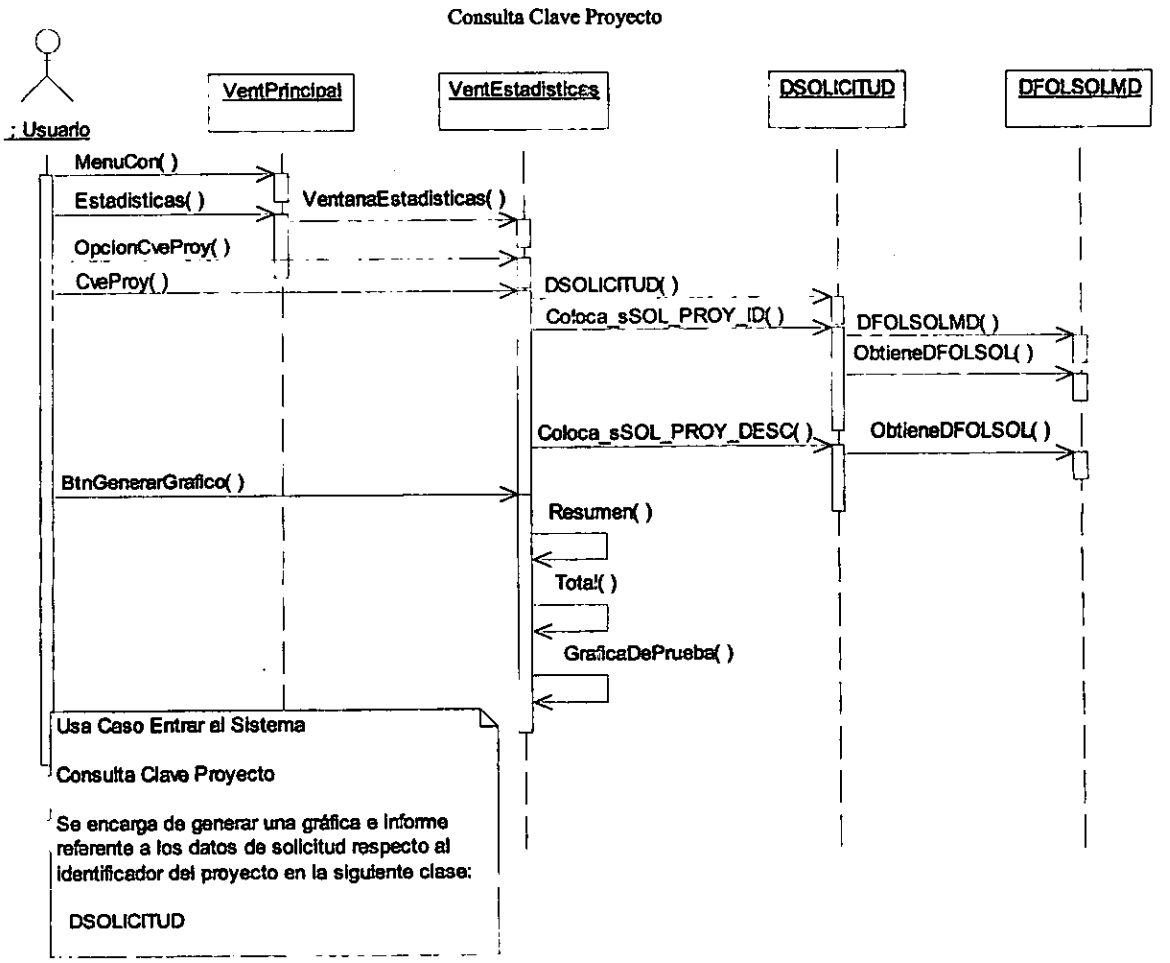


Consulta Pertenencia

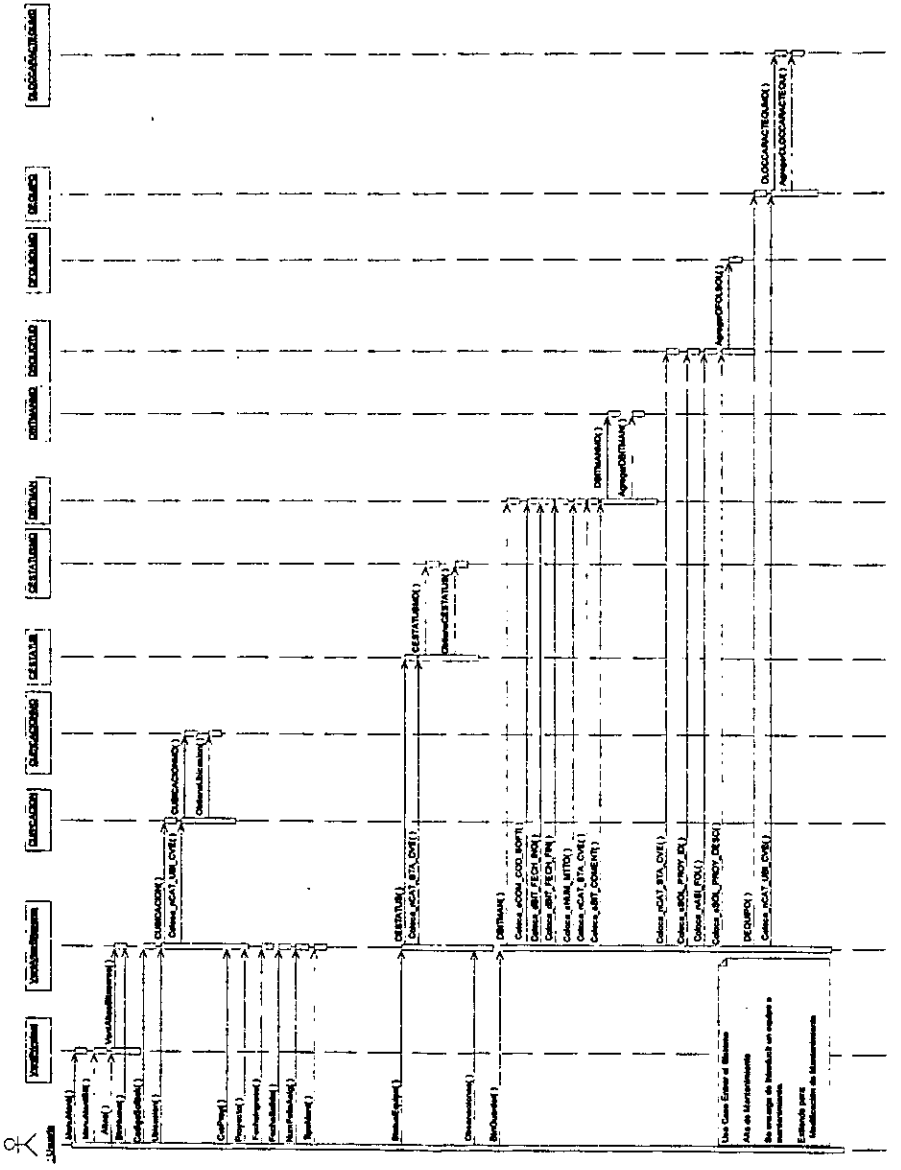


Consulta Bitácora

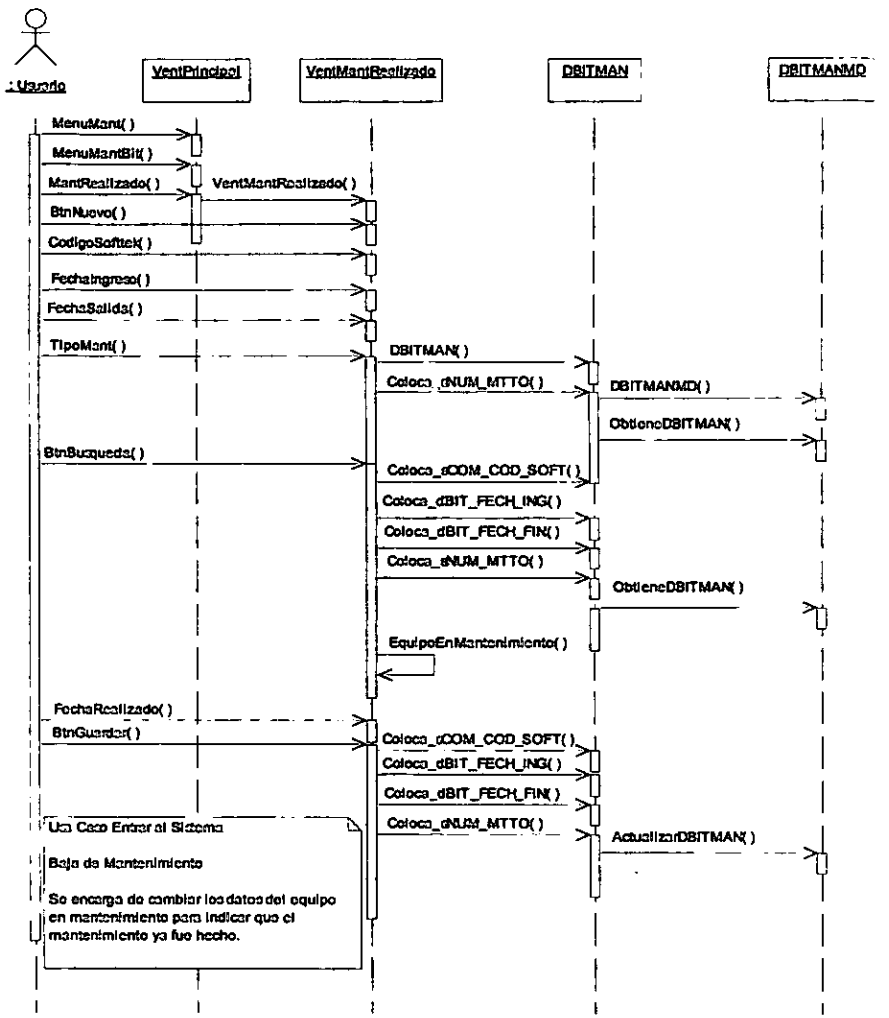



Consulta Clave Proyecto
: Usuario
VentPrincipal
VentEstadisticas
DSOLICITUD
DFOLSOLMD
MenuCon( )
Estadisticas( )
VentanaEstadisticas( )
OpcionCveProy( )
CveProy( )
DSOLICITUD( )
Coloca_sSOL_PROY_ID( )
DFOLSOLMD( )
ObtieneDFOLSOL( )
Coloca_sSOL_PROY_DESC( )
ObtieneDFOLSOL( )
BtnGenerarGrafico( )
Resumen( )
Total( )
GraficaDePrueba( )
Usa Caso Entrar al Sistema
Consulta Clave Proyecto
Se encarga de generar una gráfica e informe referente a los datos de solicitud respecto al identificador del proyecto en la siguiente clase:
DSOLICITUD

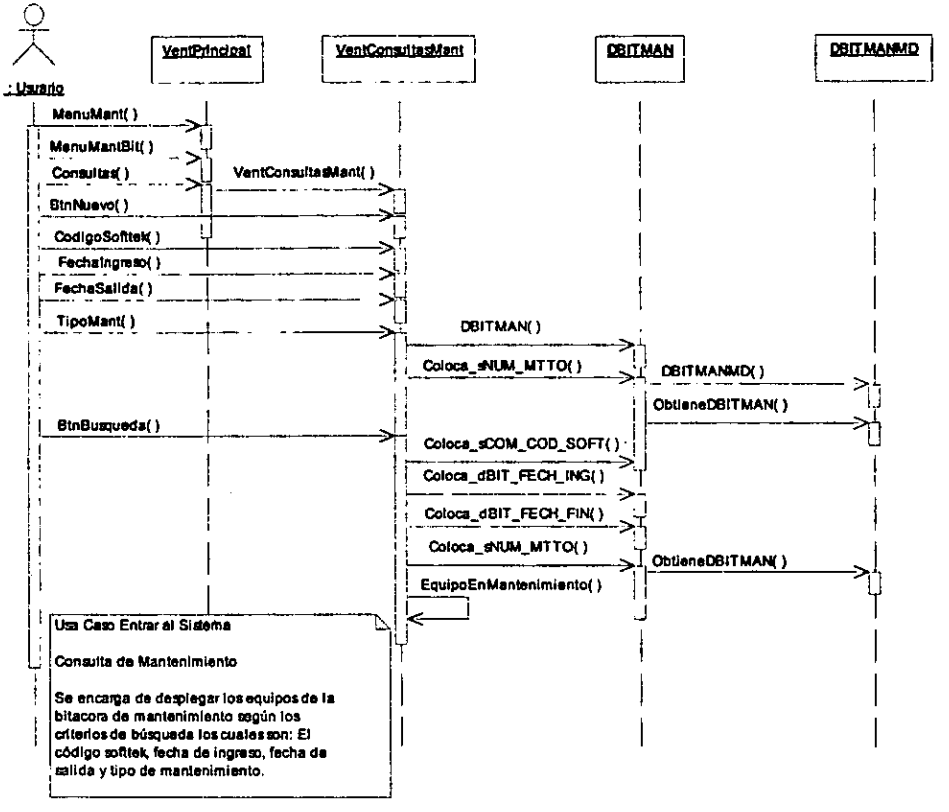
Alta de Mantenimiento



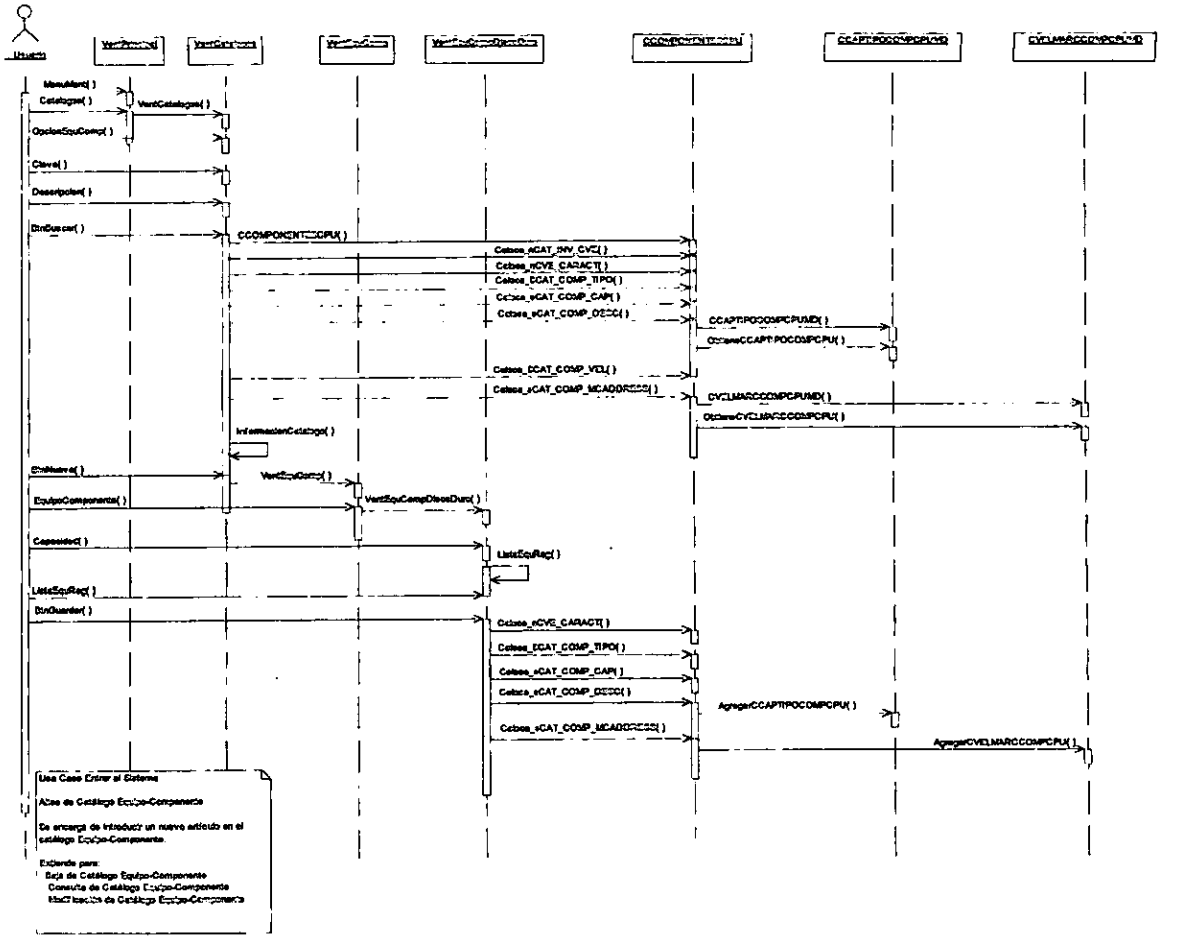
Baja de Mantenimiento



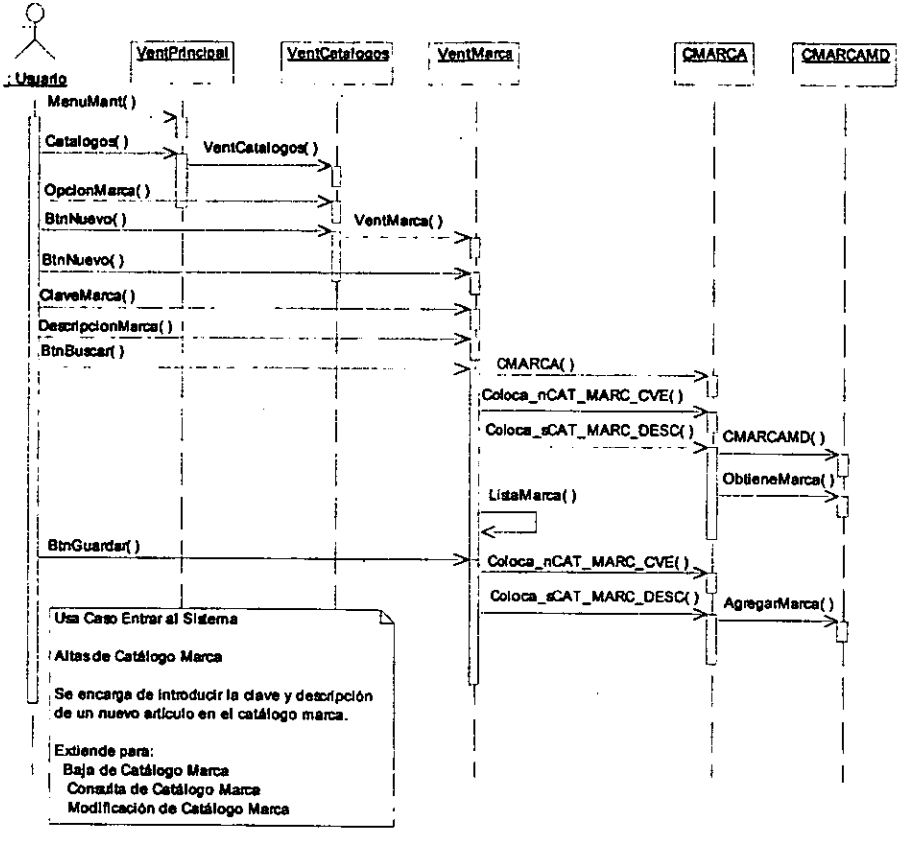
Consulta de Mantenimiento



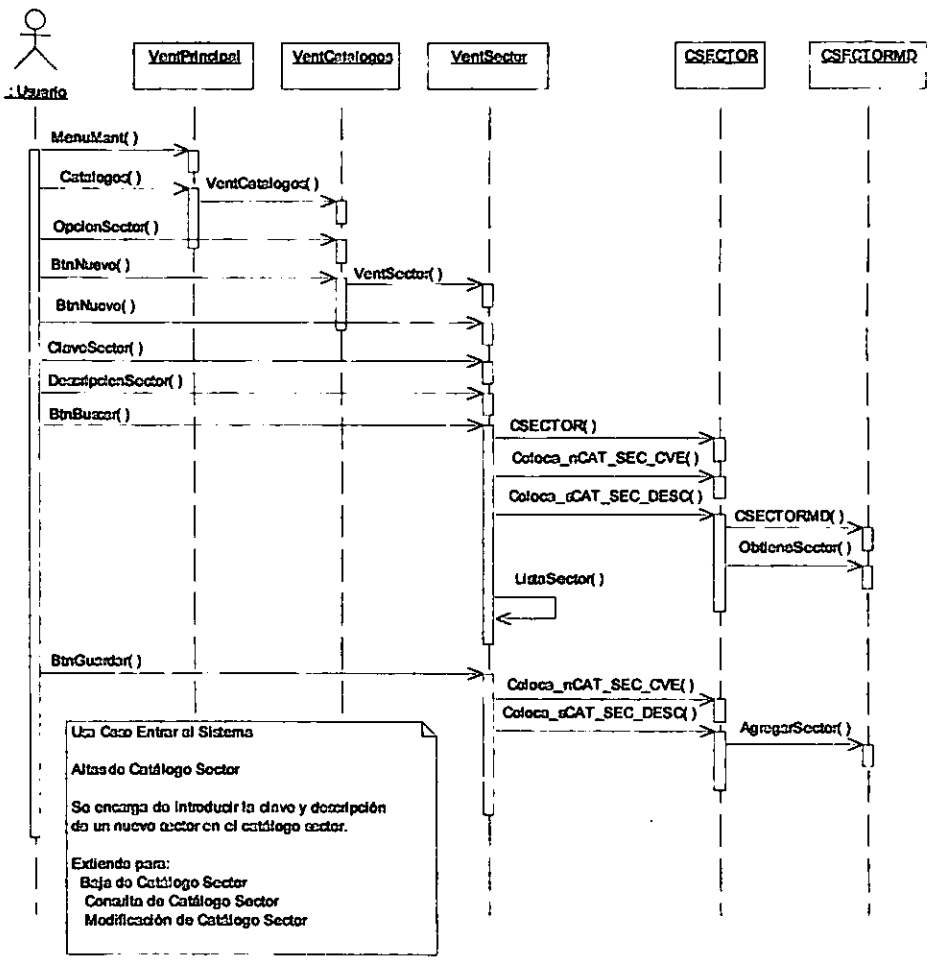
Altas de Catálogo Equipo-Componente



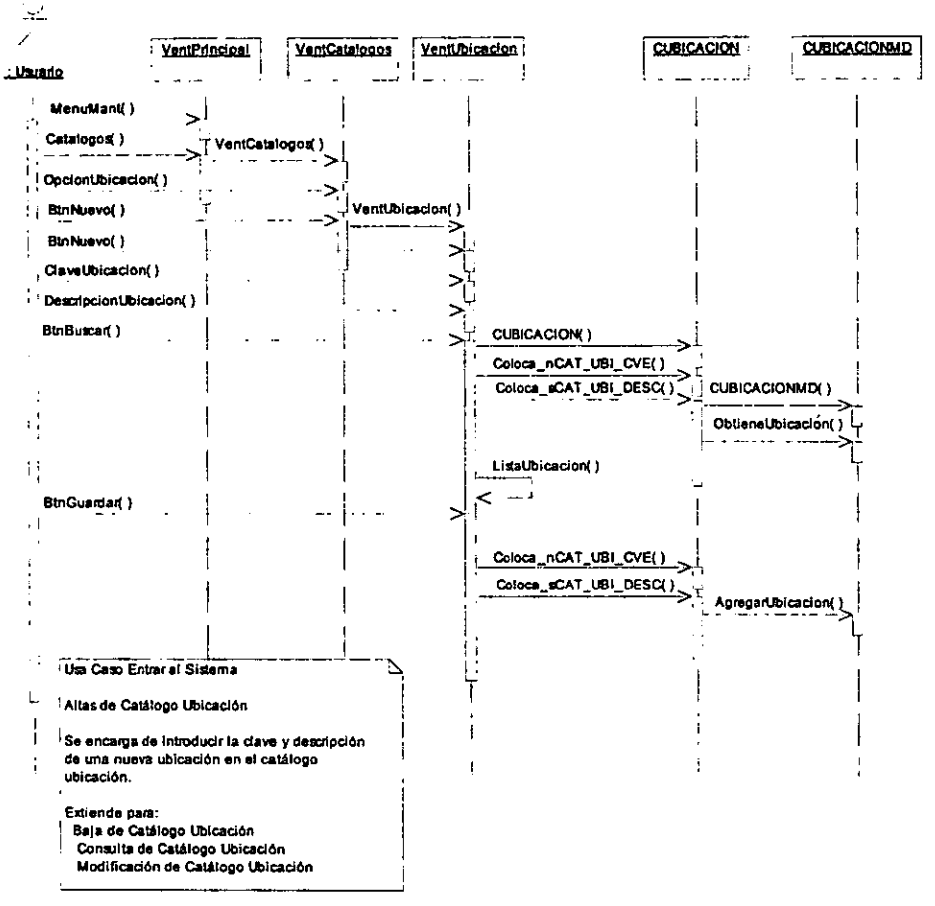
Altas de Catálogo Marca



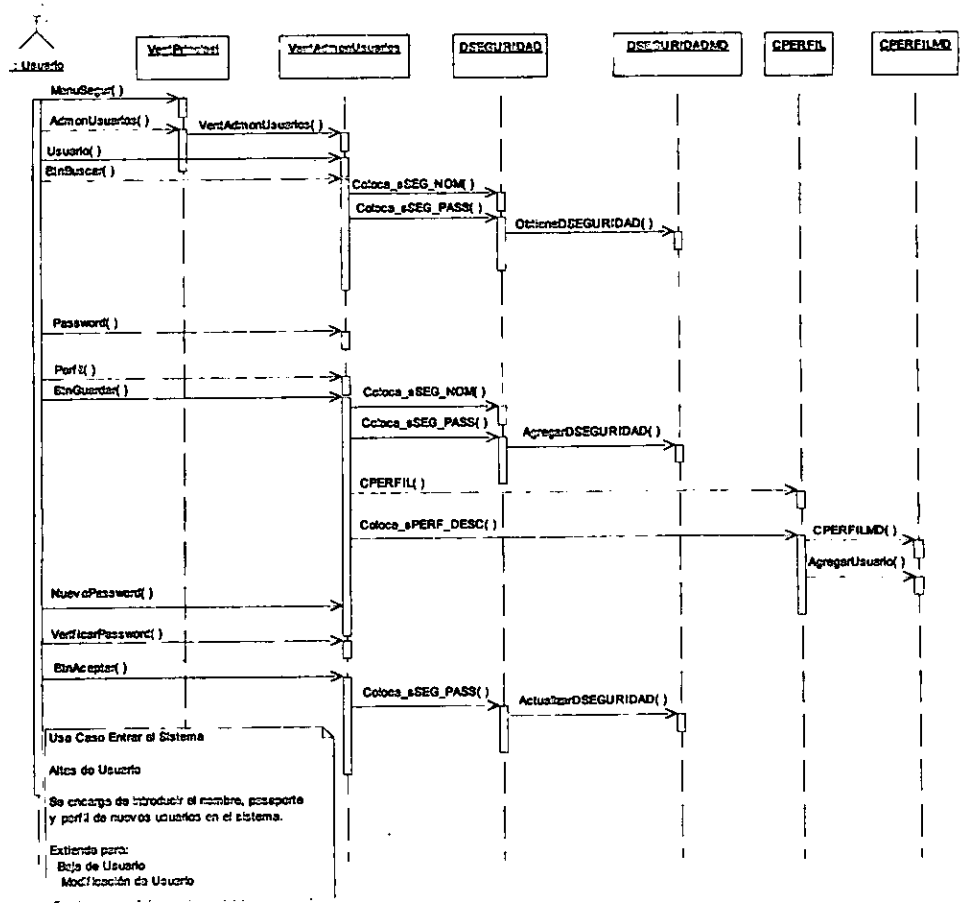
Altas de Catálogo Sector



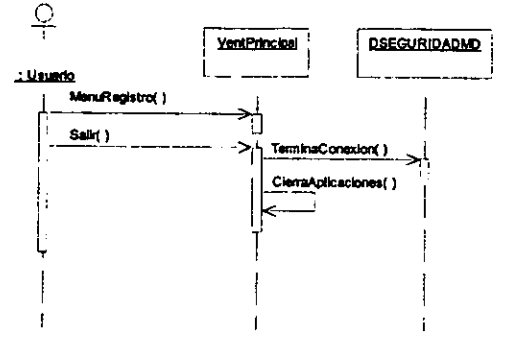
Altas de Catálogo Ubicación



Altas de Usuario



Salir del Sistema



Usa Caso Entrar al Sistema

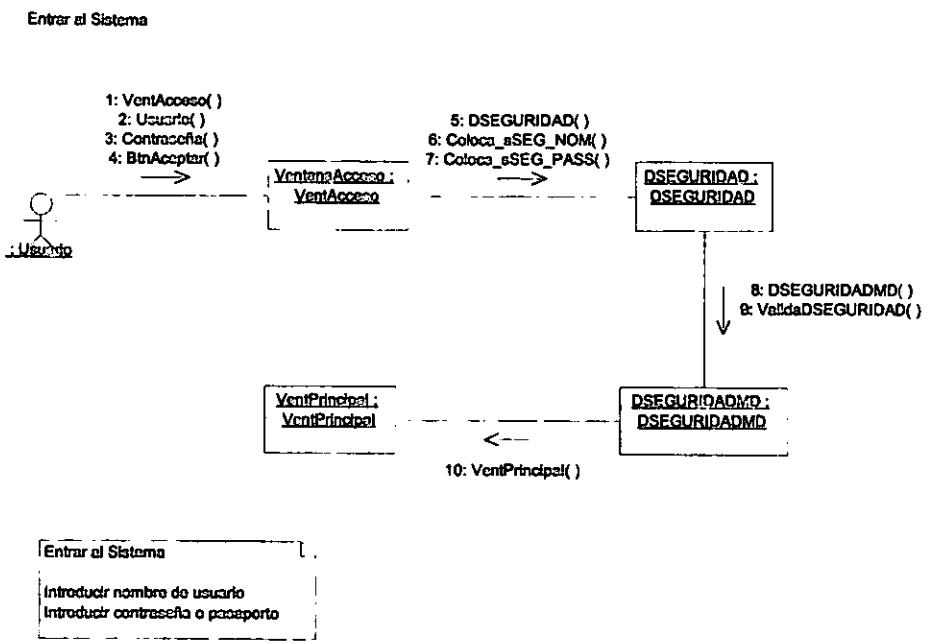
Salir del Sistema

Se encarga de terminar con todas las aplicaciones activas del sistema de información y desconectar al usuario.

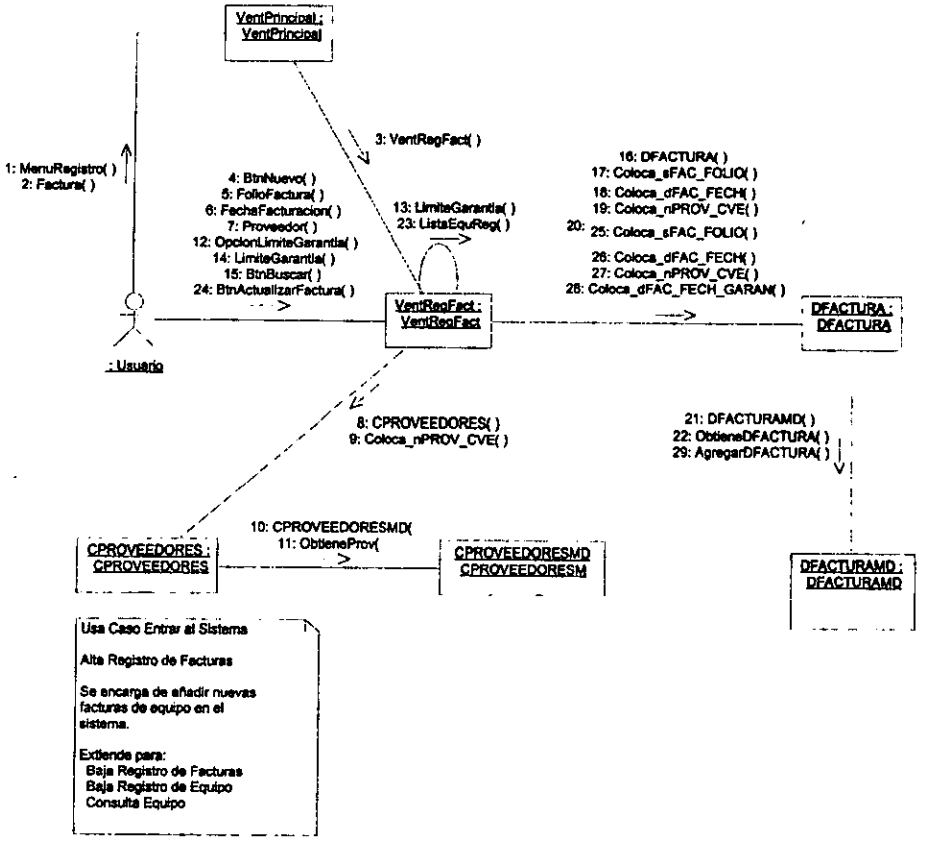
4.1.7. Diagrama de colaboración

Un diagrama de colaboración muestra interacciones entre objetos; es decir, el flujo de mensajes entre objetos en una aplicación orientada a objetos e implica las asociaciones básicas entre los mismos. Un diagrama de colaboración expresa, a la vez, el contexto de un grupo de objetos (a través de objetos y enlaces) y la interacción entre ellos (mediante la notación de envíos de mensajes).

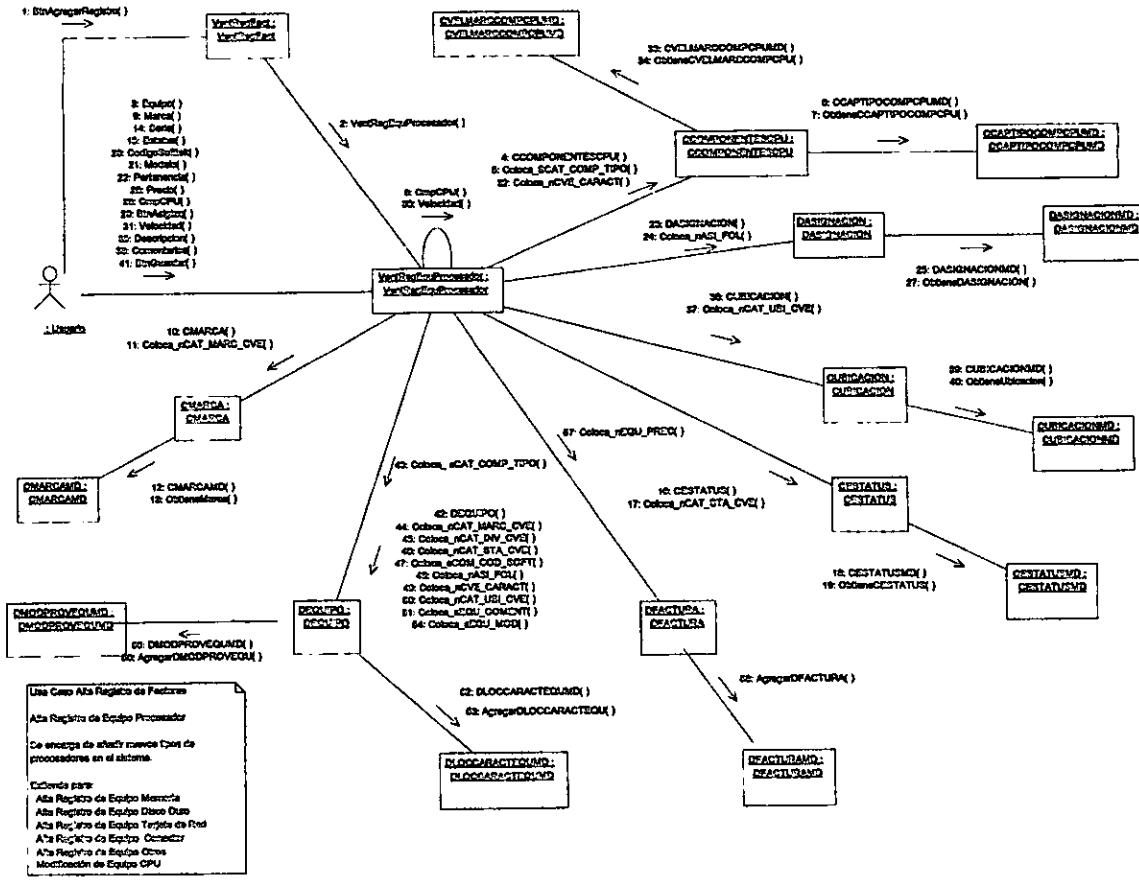
A continuación se procederá a mostrar los diagramas de colaboración del sistema de información. (Utilización de estándares y técnicas para el desarrollo del software Según IEEE Estándar 730 y 983 Herramientas, técnicas y metodologías. Butler Estándares y convenciones.)



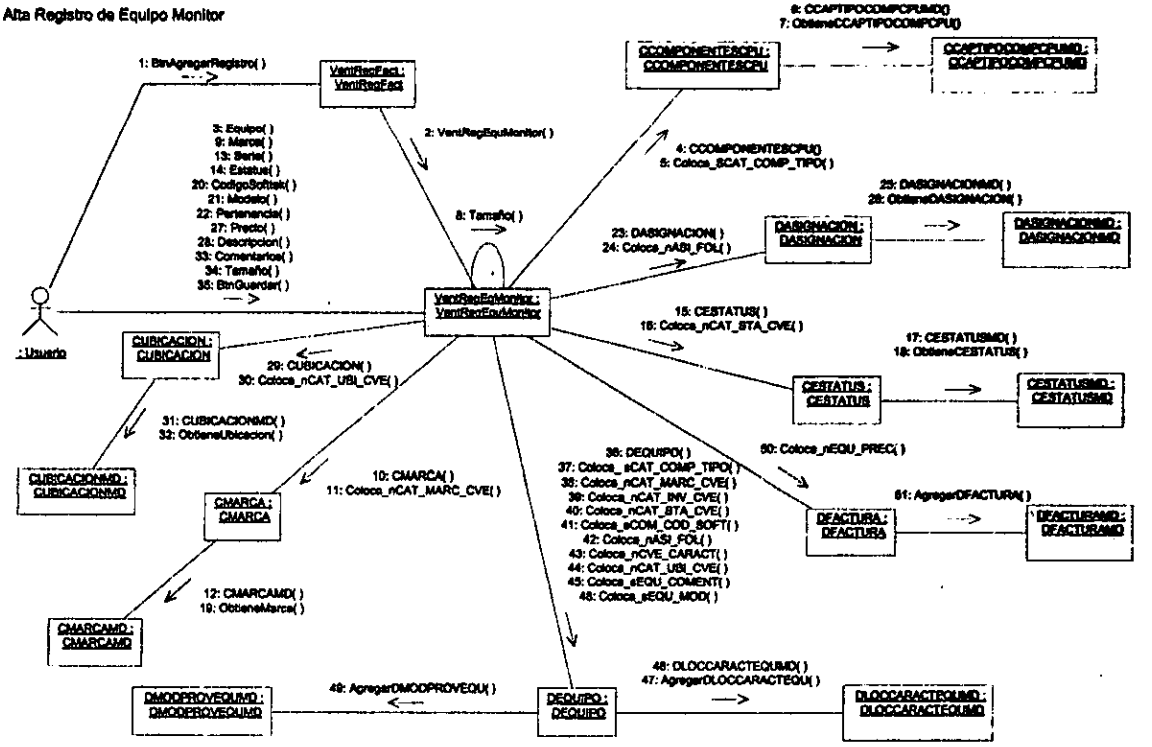
Alta Registro de Facturas



Alta Registro de Equipo Procesador



Alta Registro de Equipo Monitor



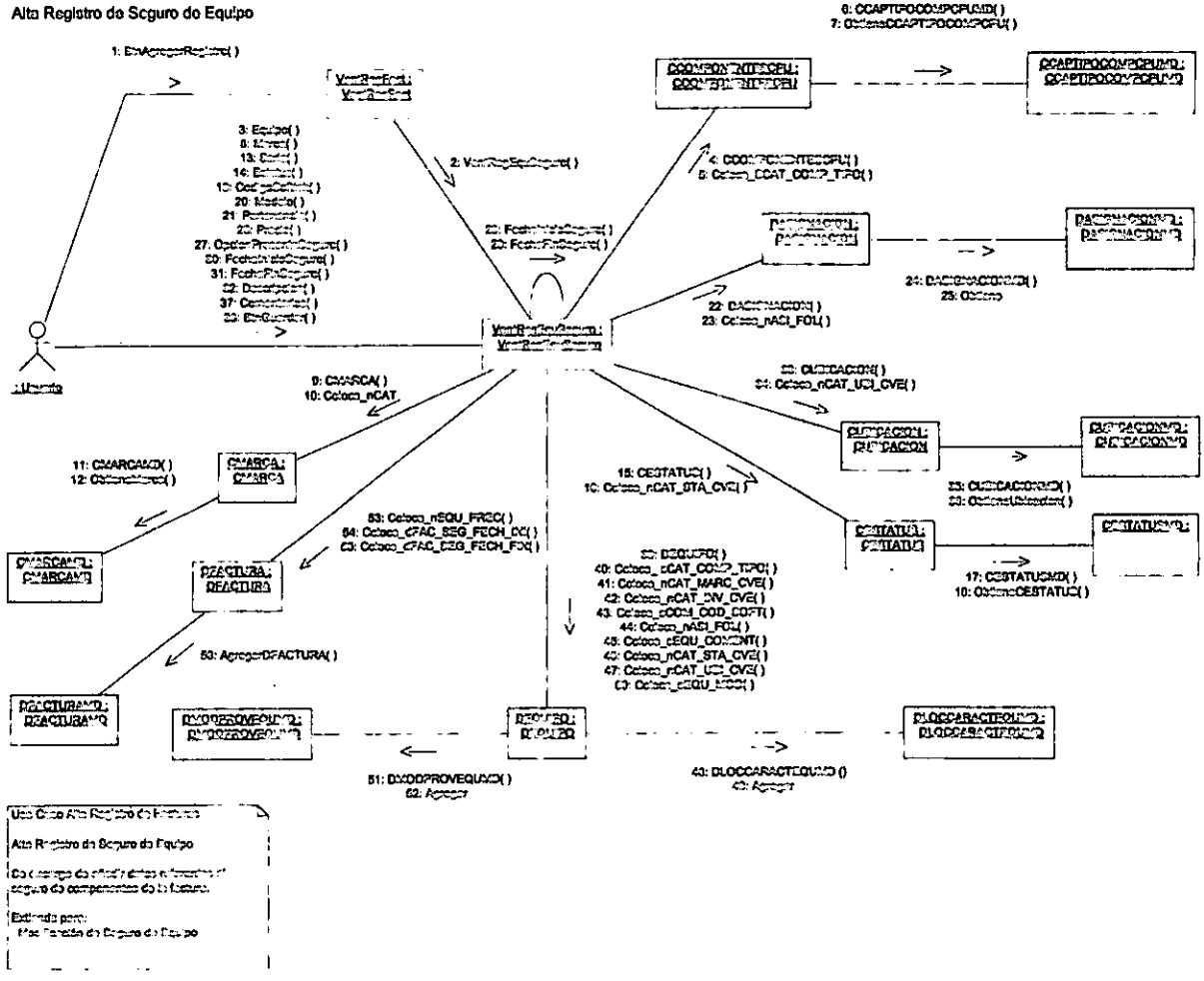
Use Caso Alta Registro de Facturas

Alta Registro de Equipo Monitor

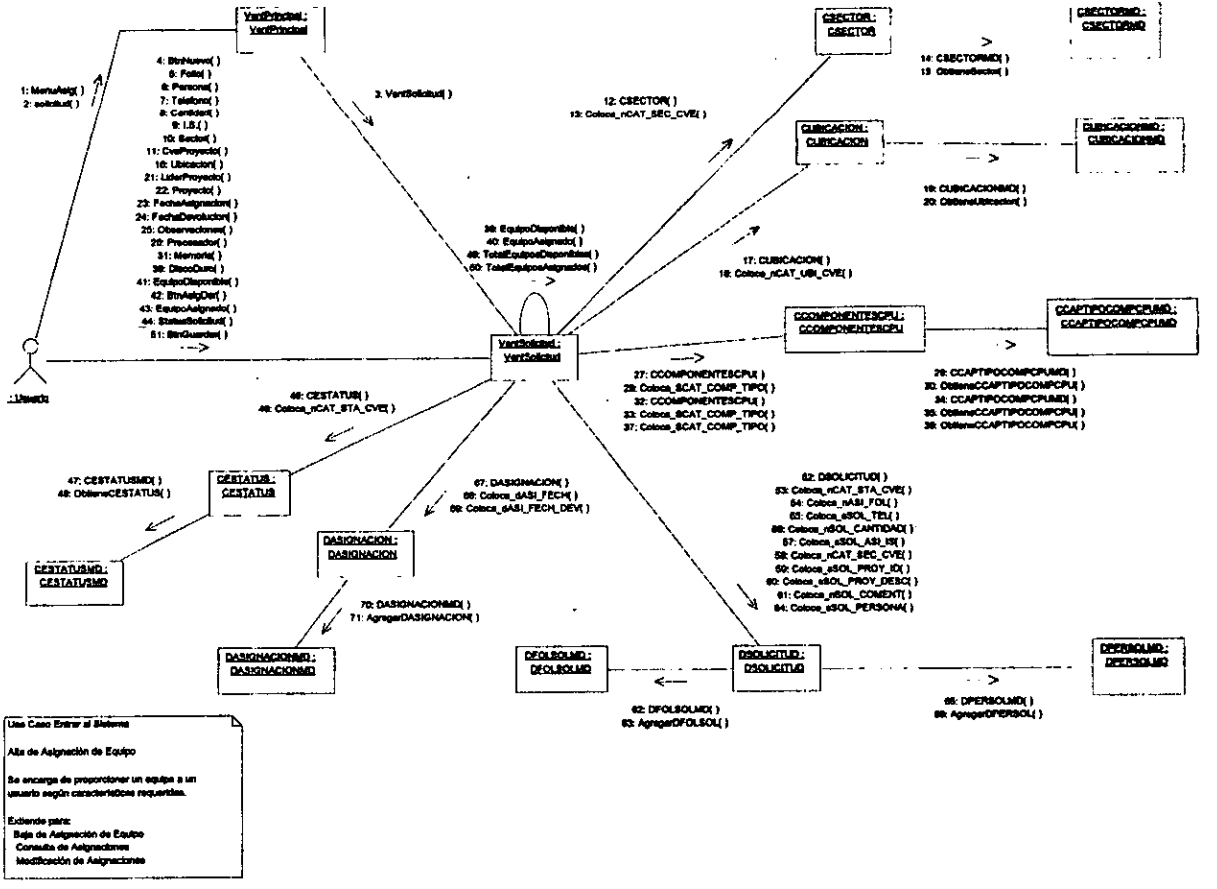
Se encarga de añadir nuevos monitores en el sistema.

Extiende para:
Alta Registro de Equipo Teclado y Mouse
Modificación de Equipo Periféricos

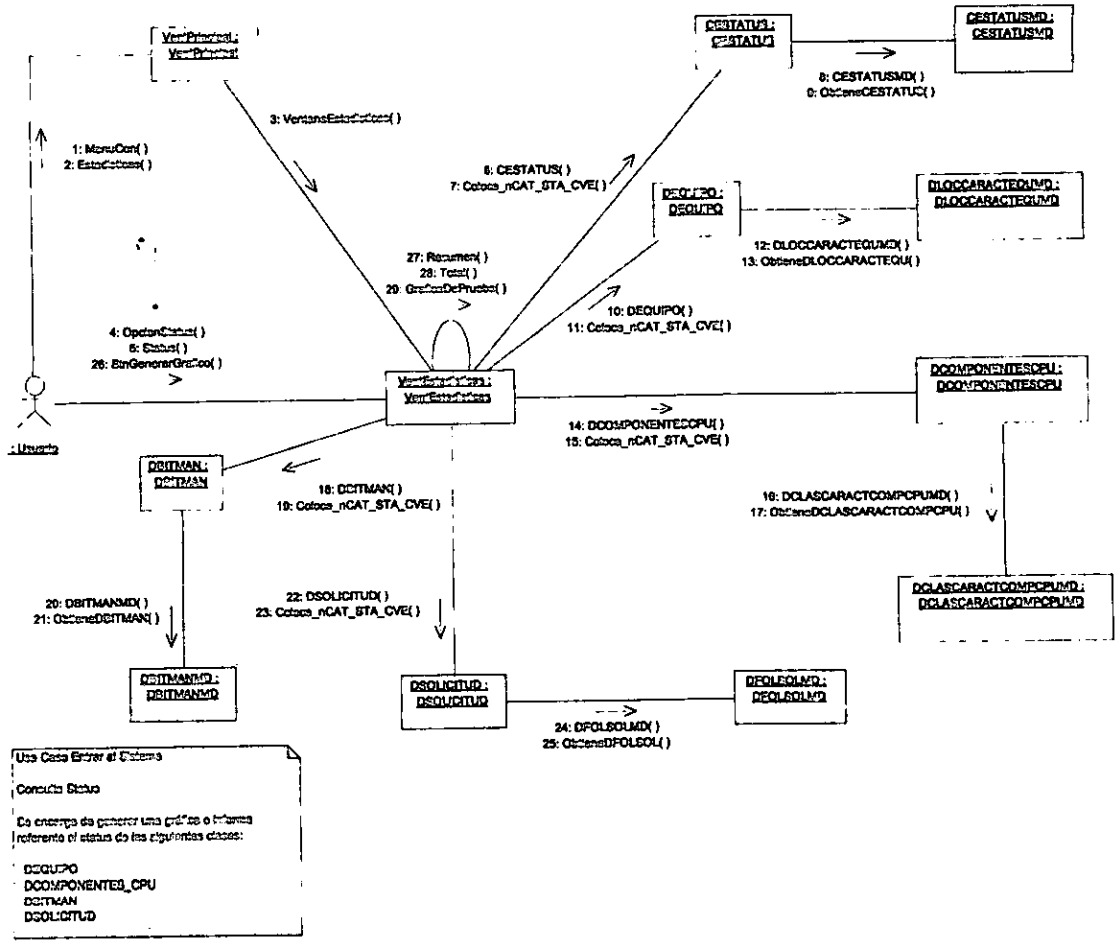
Alta Registro de Seguro de Equipo



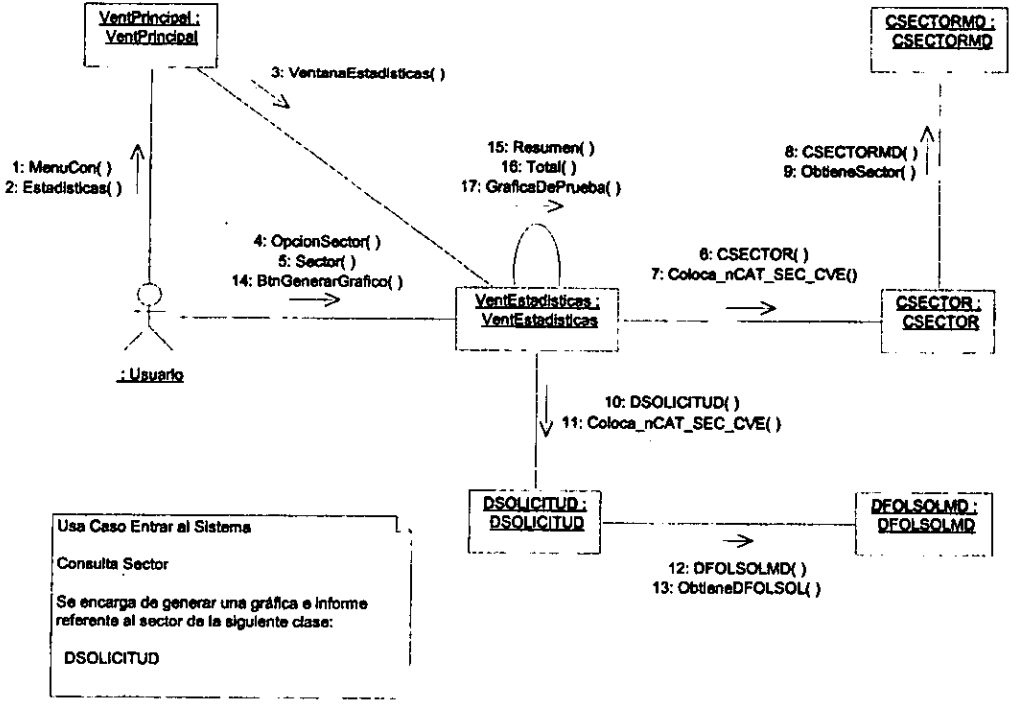
Alta de Asignación de Equipo



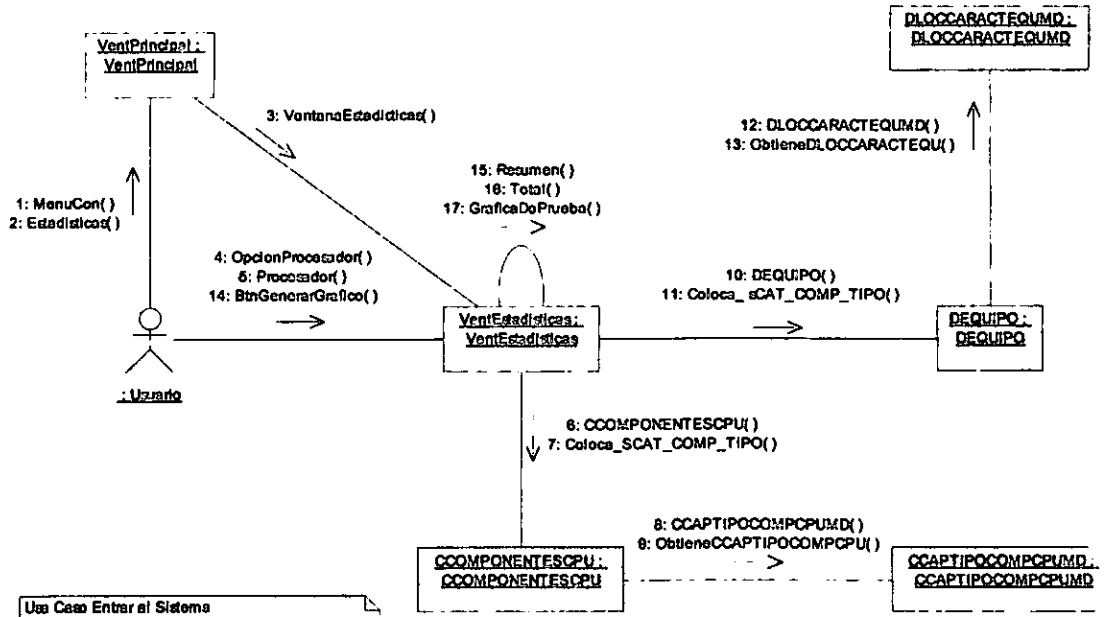
Consulta Status



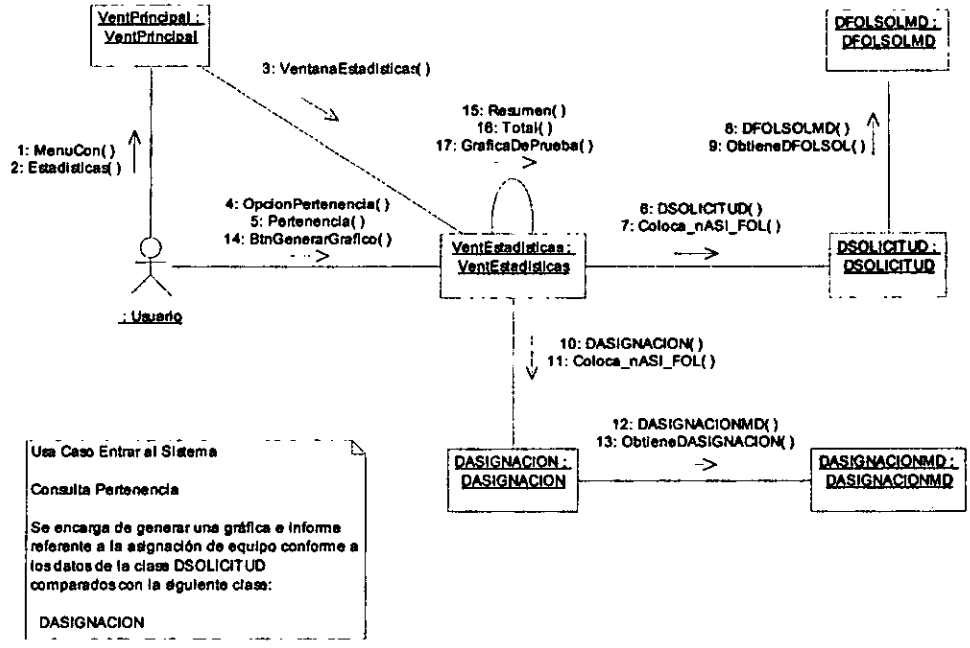
Consulta Sector



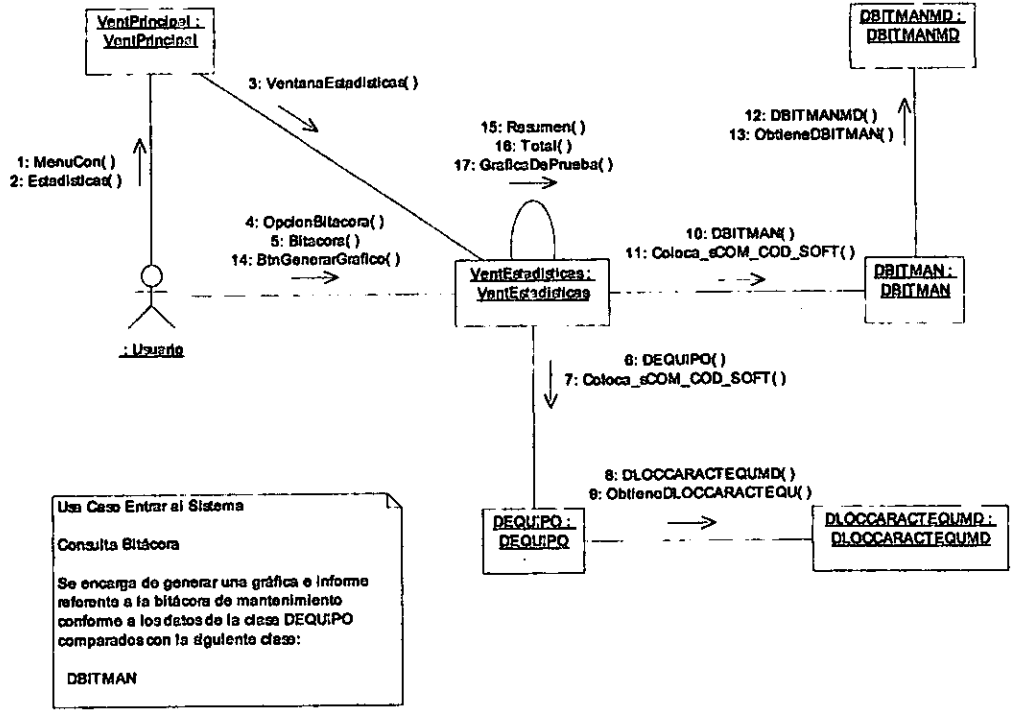
Consulta Procesador



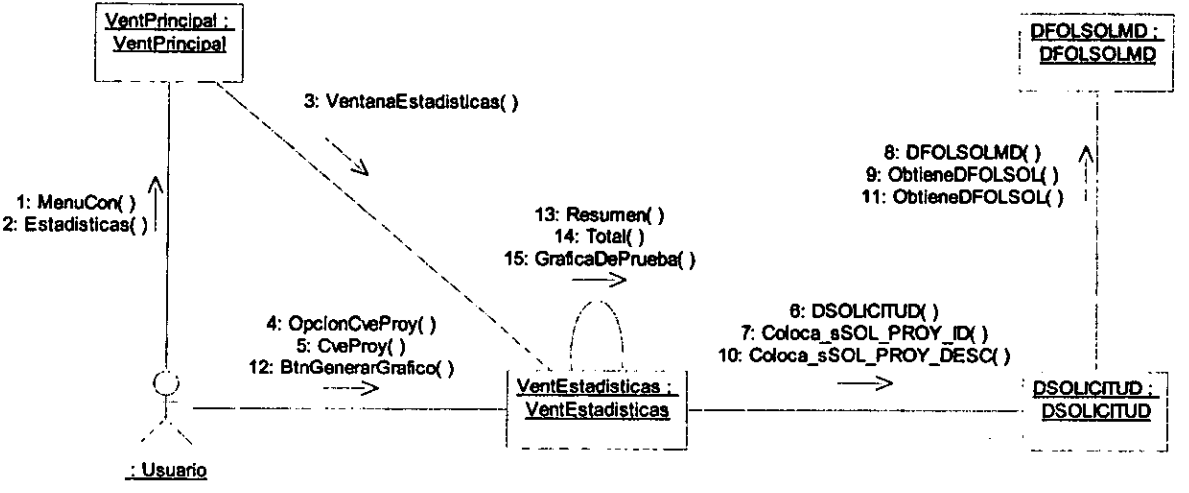
Consulta Pertenencia



Consulta Bitácora



Consulta Clave Proyecto



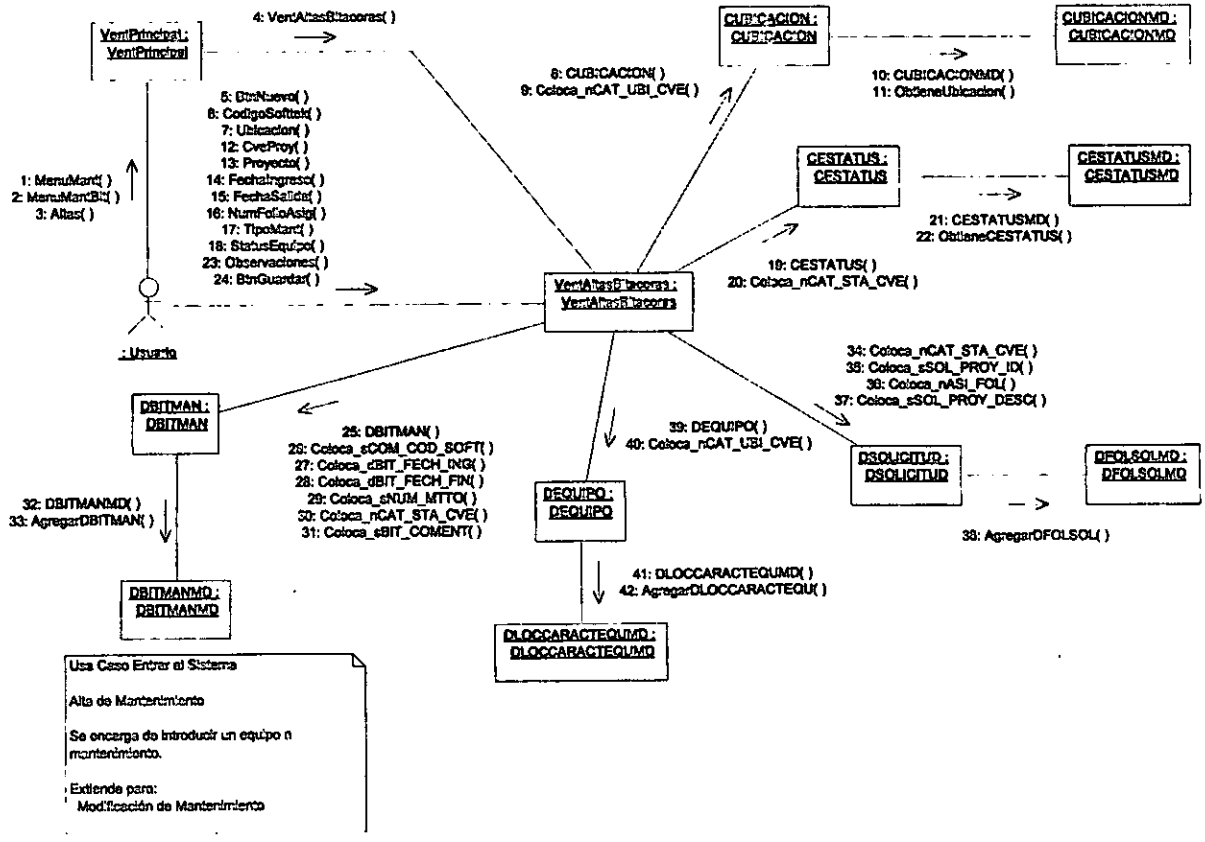
Usa Caso Entrar al Sistema

Consulta Clave Proyecto

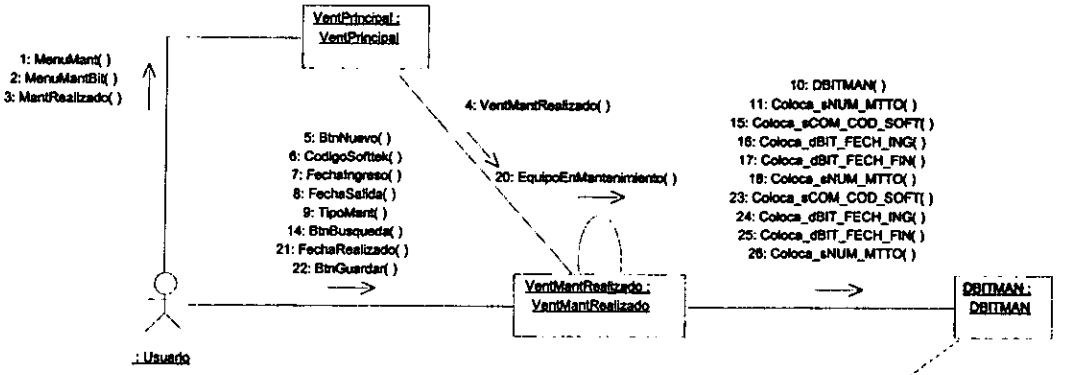
Se encarga de generar una gráfica e informe referente a los datos de solicitud respecto al identificador del proyecto en la siguiente clase:

DSOLICITUD

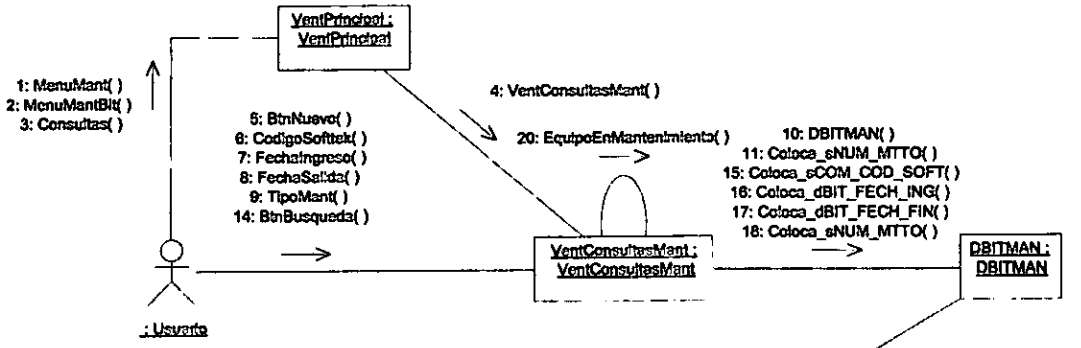
Alta de Mantenimiento



Baja de Mantenimiento



Consulta de Mantenimiento

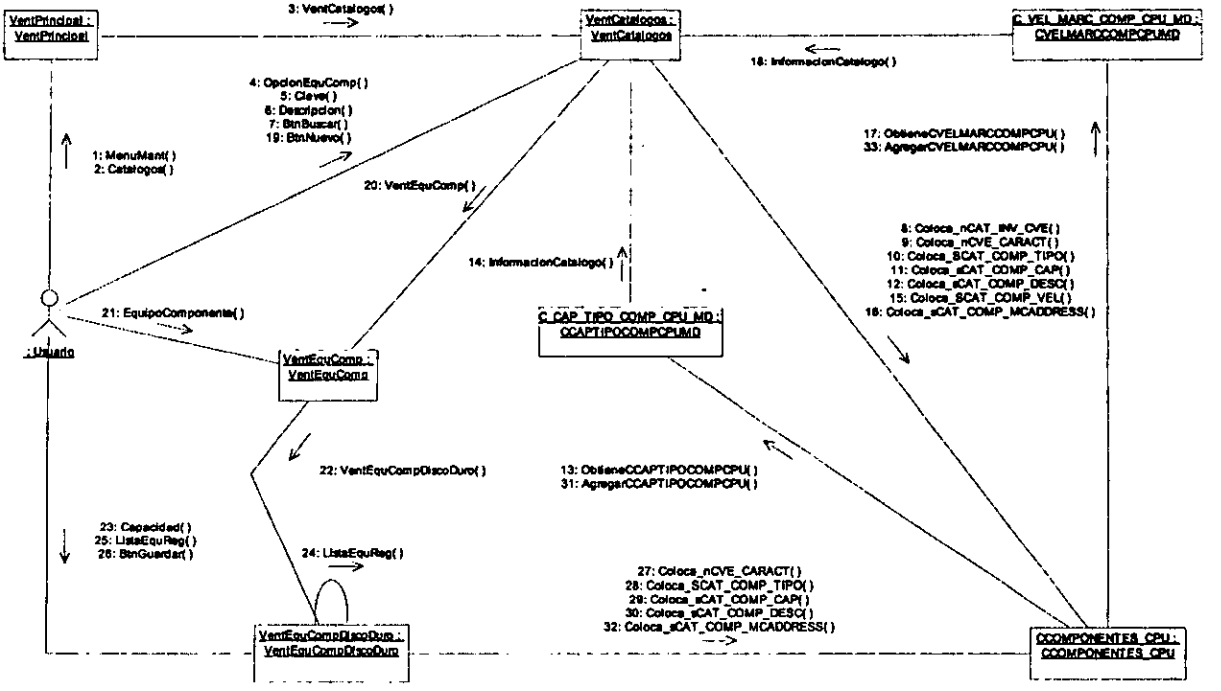


Usa Caso Entrar al Sistema

Consulta de Mantenimiento

Se encarga de desplegar los equipos de la bitacora de mantenimiento según los criterios de búsqueda los cuales son: El código softtek, fecha de ingreso, fecha de salida y tipo de mantenimiento.

Altas de Catálogo Equipo-Componente



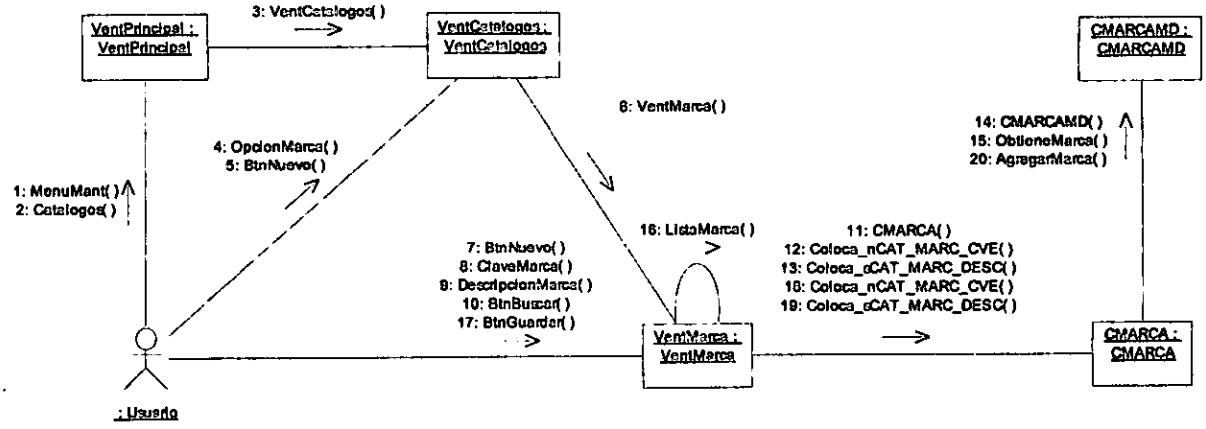
Usa Caso Entrar al Sistema

Altas de Catálogo Equipo-Componente

Se encarga de introducir un nuevo artículo en el catálogo Equipo-Componente..

Extiende para:
Baja de Catálogo Equipo-Componente
Consulta de Catálogo Equipo-Componente
Modificación de Catálogo Equipo-Componente

Altas de Catálogo Marca



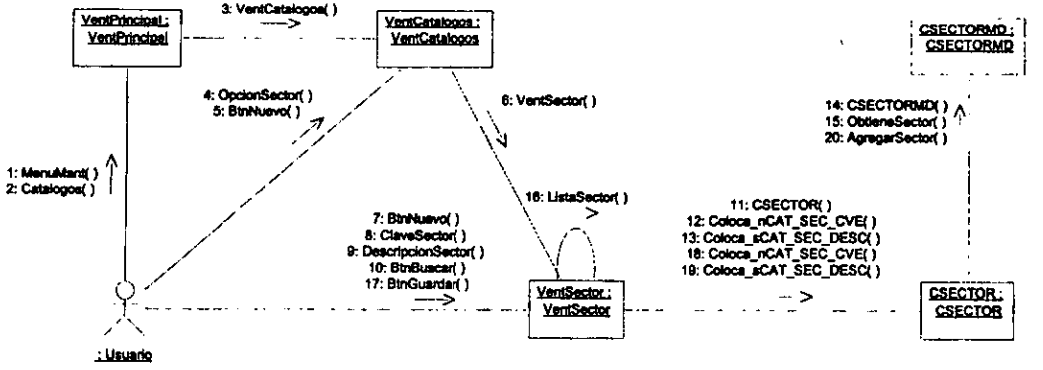
Usa Caso Entrar al Sistema

Altas de Catálogo Marca

Se encarga de introducir la clave y descripción de un nuevo artículo en el catálogo marca.

Extiende para:
- **Baja de Catálogo Marca**
- **Consulta de Catálogo Marca**
- **Modificación de Catálogo Marca**

Altas de Catálogo Sector



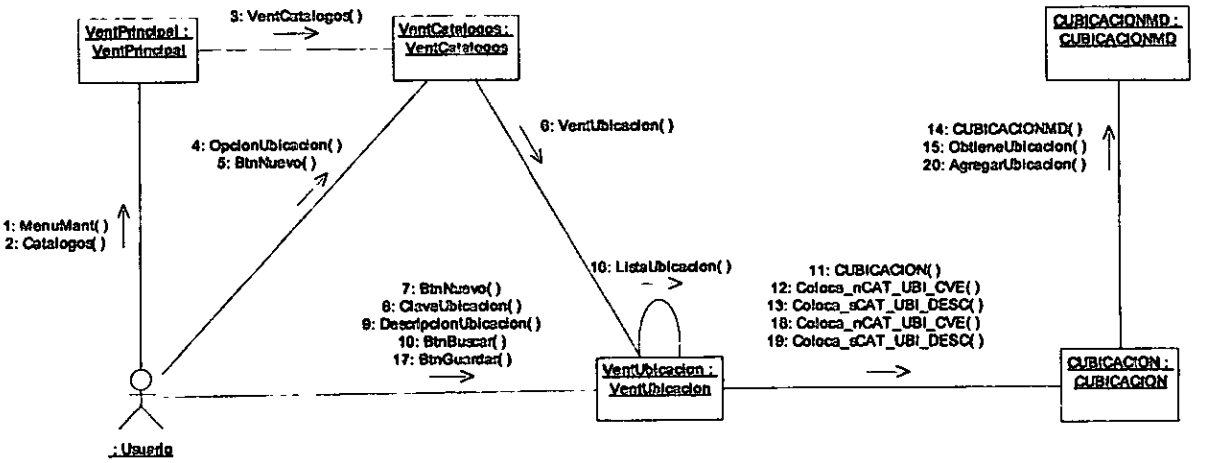
Usa Caso Entrar al Sistema

Altas de Catálogo Sector

Se encarga de introducir la clave y descripción de un nuevo sector en el catálogo sector.

Extiende para:
Baja de Catálogo Sector
Consulta de Catálogo Sector
Modificación de Catálogo Sector

Altas de Catálogo Ubicación



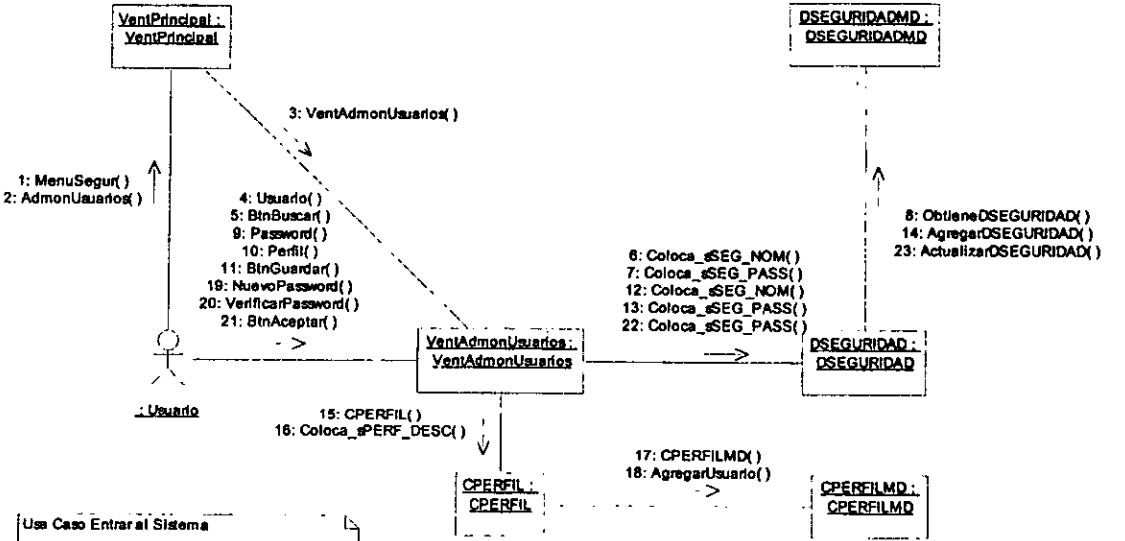
Use Case Entrar al Sistema

Altas de Catálogo Ubicación

Se encarga de introducir la clave y descripción de una nueva ubicación en el catálogo ubicación.

Extiende para:
- Baja de Catálogo Ubicación
- Consulta de Catálogo Ubicación
- Modificación de Catálogo Ubicación

Altas de Usuario



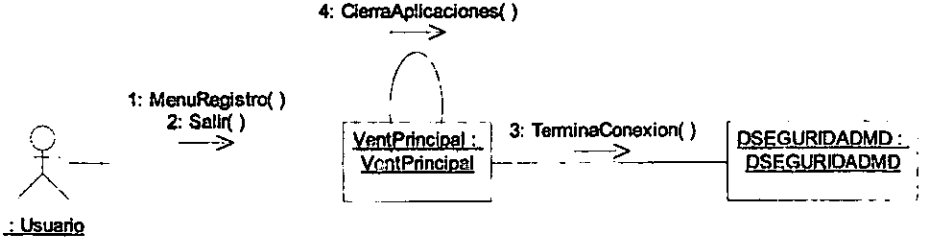
Usa Caso Entrar al Sistema

Altas de Usuario

Se encarga de Introducir el nombre, pasaporte y perfil de nuevos usuarios en el sistema.

Extiende para:
Baja de Usuario
Modificación de Usuario

Salir del Sistema



Usa Caso Entrar al Sistema

Salir del Sistema

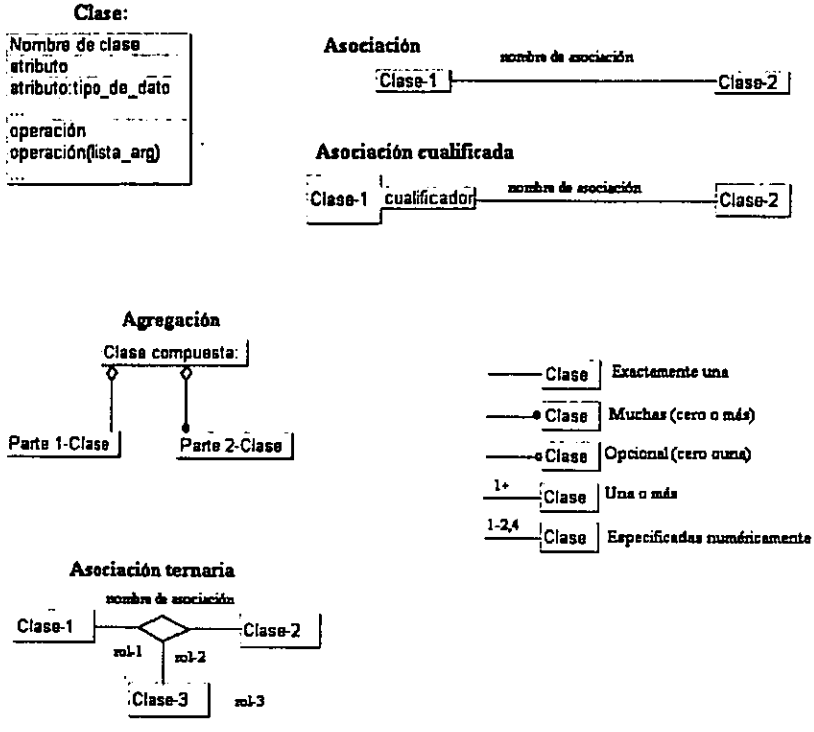
Se encarga de terminar con todas las aplicaciones activas del sistema de información y desconectar al usuario.

4.1.8. Diagramas de clases

El diagrama de clases es usado para mostrar la existencia de clases y sus relaciones en la parte lógica del sistema. Representa la vista de la estructura de clases de un sistema. Durante el análisis, indican las responsabilidades, roles y relaciones con otras clases que determinan el comportamiento del sistema. Durante el diseño, capturan la estructura de las clases que forman la arquitectura del sistema.

A continuación se presenta la simbología utilizada en los diagramas de clases:

Simbología.



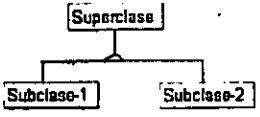
Diagramas de clases

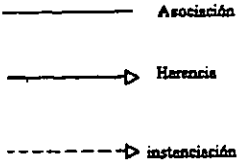
El siguiente diagrama de clases presenta las clases de la definición del problema, del manejo de datos y clases de la interfaz humana.

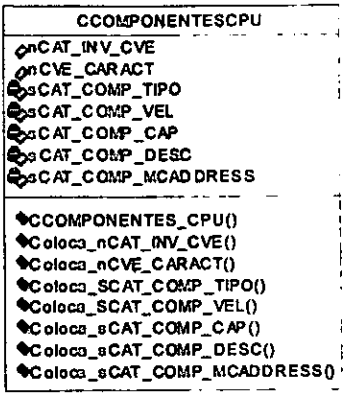
Clases de la definición del problema

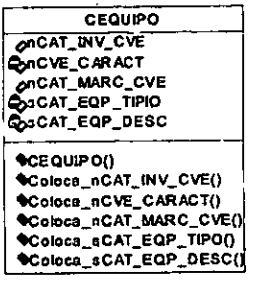
Relaciones de Clase

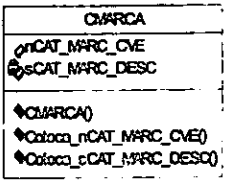
Generalización (herencia)

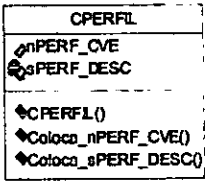


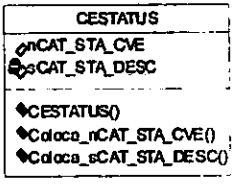


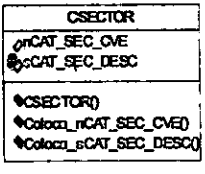


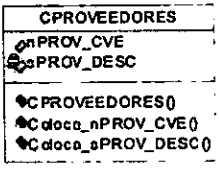


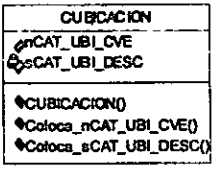


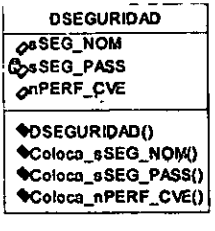


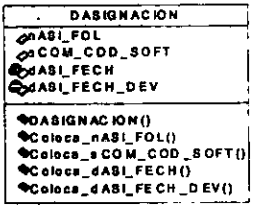


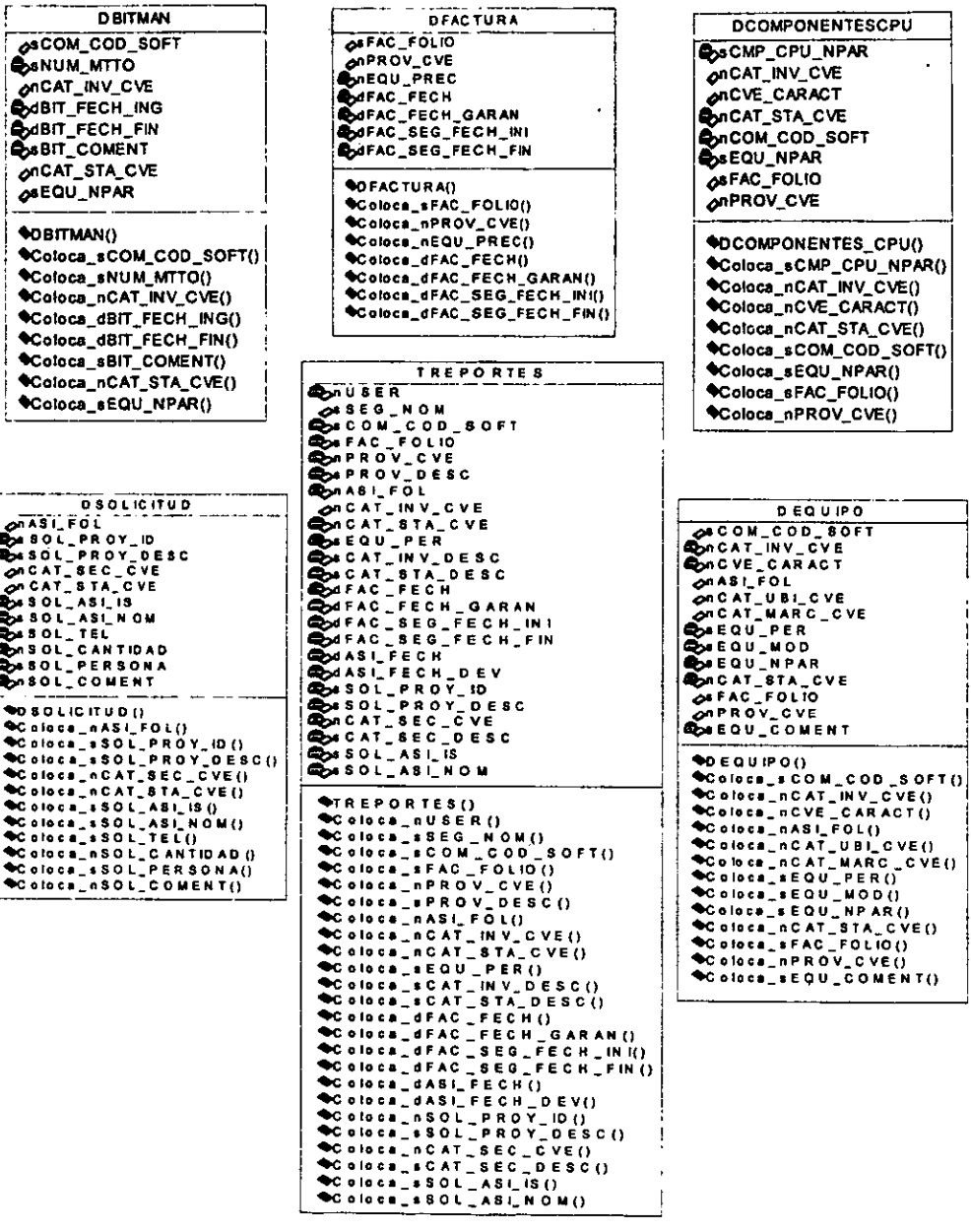




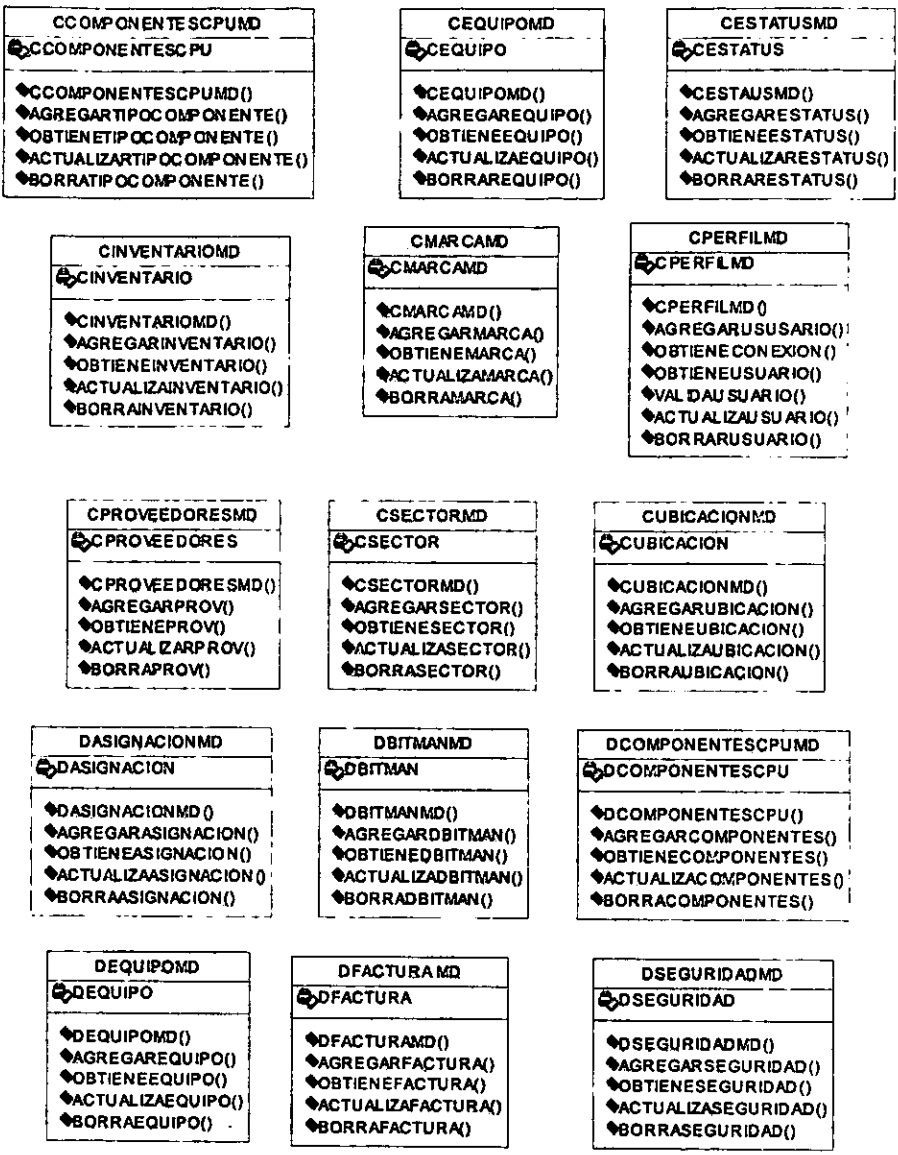


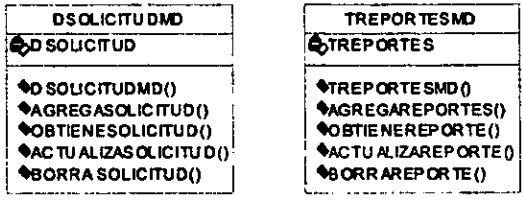




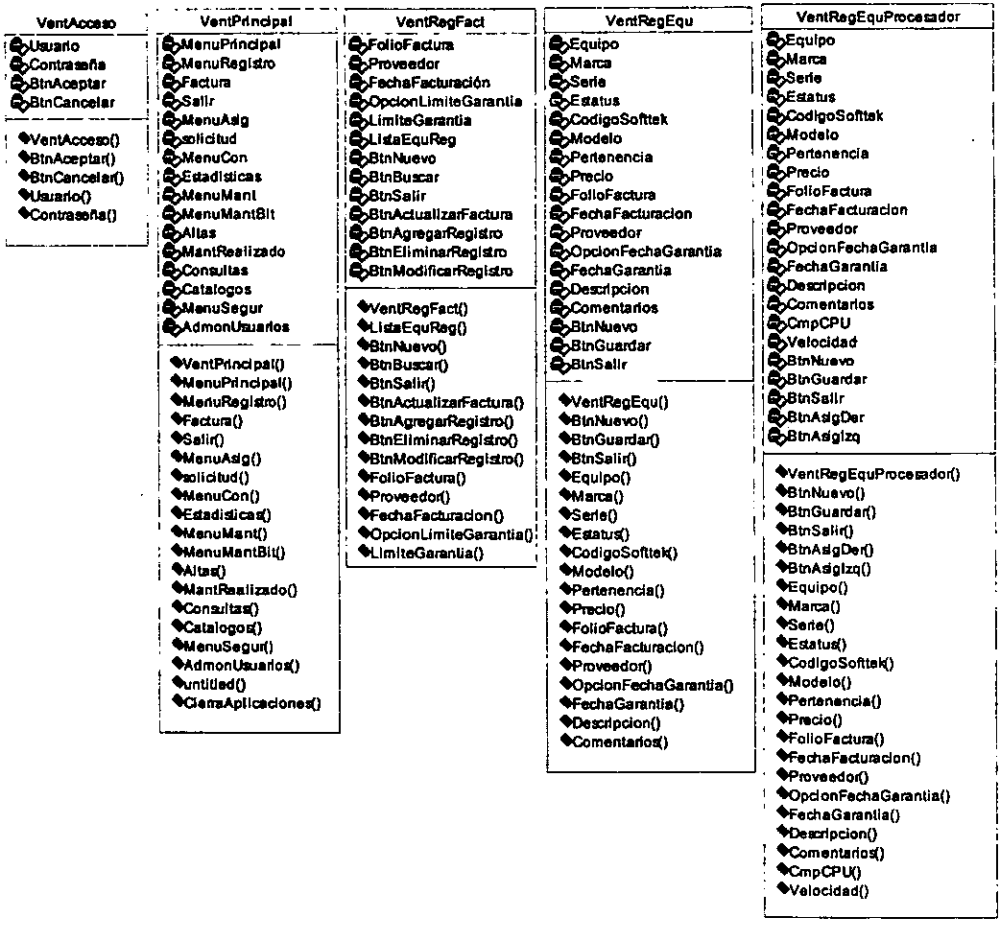

DBITMAN
sCOM_COD_SOFT
sNUM_MTTO
nCAT_INV_CVE
dBIT_FECH_ING
dBIT_FECH_FIN
sBIT_COMENT
nCAT_STA_CVE
sEQU_NPAR
DBITMAN()
Coloca_sCOM_COD_SOFT()
Coloca_sNUM_MTTO()
Coloca_nCAT_INV_CVE()
Coloca_dBIT_FECH_ING()
Coloca_dBIT_FECH_FIN()
Coloca_sBIT_COMENT()
Coloca_nCAT_STA_CVE()
Coloca_sEQU_NPAR()
DFACTURA
sFAC_FOLIO
nPROV_CVE
nEQU_PREC
dFAC_FECH
dFAC_FECH_GARAN
dFAC_SEG_FECH_INI
dFAC_SEG_FECH_FIN
DFACTURA()
Coloca_sFAC_FOLIO()
Coloca_nPROV_CVE()
Coloca_nEQU_PREC()
Coloca_dFAC_FECH()
Coloca_dFAC_FECH_GARAN()
Coloca_dFAC_SEG_FECH_INI()
Coloca_dFAC_SEG_FECH_FIN()
DCOMPONENTESCPU
sCMP_CPU_NPAR
nCAT_INV_CVE
nCVE_CARACT
nCAT_STA_CVE
nCOM_COD_SOFT
sEQU_NPAR
sFAC_FOLIO
nPROV_CVE
DCOMPONENTES_CPU()
Coloca_sCMP_CPU_NPAR()
Coloca_nCAT_INV_CVE()
Coloca_nCVE_CARACT()
Coloca_nCAT_STA_CVE()
Coloca_sCOM_COD_SOFT()
Coloca_sEQU_NPAR()
Coloca_sFAC_FOLIO()
Coloca_nPROV_CVE()
TREPORTES
nUSER
sSEG_NOM
sCOM_COD_SOFT
sFAC_FOLIO
nPROV_CVE
sPROV_DESC
nASI_FOL
nCAT_INV_CVE
nCAT_STA_CVE
sEQU_PER
sCAT_INV_DESC
sCAT_STA_DESC
dFAC_FECH
dFAC_FECH_GARAN
dFAC_SEG_FECH_INI
dFAC_SEG_FECH_FIN
dASI_FECH
dASI_FECH_DEV
sSOL_PROY_ID
sSOL_PROY_DESC
nCAT_SEC_CVE
sCAT_SEC_DESC
sSOL_ASI_IS
sSOL_ASI_NOM
TREPORTES()
Coloca_nUSER()
Coloca_sSEG_NOM()
Coloca_sCOM_COD_SOFT()
Coloca_sFAC_FOLIO()
Coloca_nPROV_CVE()
Coloca_sPROV_DESC()
Coloca_nASI_FOL()
Coloca_nCAT_INV_CVE()
Coloca_nCAT_STA_CVE()
Coloca_sEQU_PER()
Coloca_sCAT_INV_DESC()
Coloca_sCAT_STA_DESC()
Coloca_dFAC_FECH()
Coloca_dFAC_FECH_GARAN()
Coloca_dFAC_SEG_FECH_INI()
Coloca_dFAC_SEG_FECH_FIN()
Coloca_dASI_FECH()
Coloca_dASI_FECH_DEV()
Coloca_nSOL_PROY_ID()
Coloca_sSOL_PROY_DESC()
Coloca_nCAT_SEC_CVE()
Coloca_sCAT_SEC_DESC()
Coloca_sSOL_ASI_IS()
Coloca_sSOL_ASI_NOM()
DSOLICITUD
nASI_FOL
sSOL_PROY_ID
sSOL_PROY_DESC
nCAT_SEC_CVE
nCAT_STA_CVE
sSOL_ASI_IS
sSOL_ASI_NOM
sSOL_TEL
nSOL_CANTIDAD
sSOL_PERSONA
nSOL_COMENT
DSOLICITUD()
Coloca_nASI_FOL()
Coloca_sSOL_PROY_ID()
Coloca_sSOL_PROY_DESC()
Coloca_nCAT_SEC_CVE()
Coloca_nCAT_STA_CVE()
Coloca_sSOL_ASI_IS()
Coloca_sSOL_ASI_NOM()
Coloca_sSOL_TEL()
Coloca_nSOL_CANTIDAD()
Coloca_sSOL_PERSONA()
Coloca_nSOL_COMENT()
DEQUIPO
sCOM_COD_SOFT
nCAT_INV_CVE
nCVE_CARACT
nASI_FOL
nCAT_UBI_CVE
nCAT_MARC_CVE
sEQU_PER
sEQU_MOD
sEQU_NPAR
nCAT_STA_CVE
sFAC_FOLIO
nPROV_CVE
sEQU_COMENT
DEQUIPO()
Coloca_sCOM_COD_SOFT()
Coloca_nCAT_INV_CVE()
Coloca_nCVE_CARACT()
Coloca_nASI_FOL()
Coloca_nCAT_UBI_CVE()
Coloca_nCAT_MARC_CVE()
Coloca_sEQU_PER()
Coloca_sEQU_MOD()
Coloca_sEQU_NPAR()
Coloca_nCAT_STA_CVE()
Coloca_sFAC_FOLIO()
Coloca_nPROV_CVE()
Coloca_sEQU_COMENT()

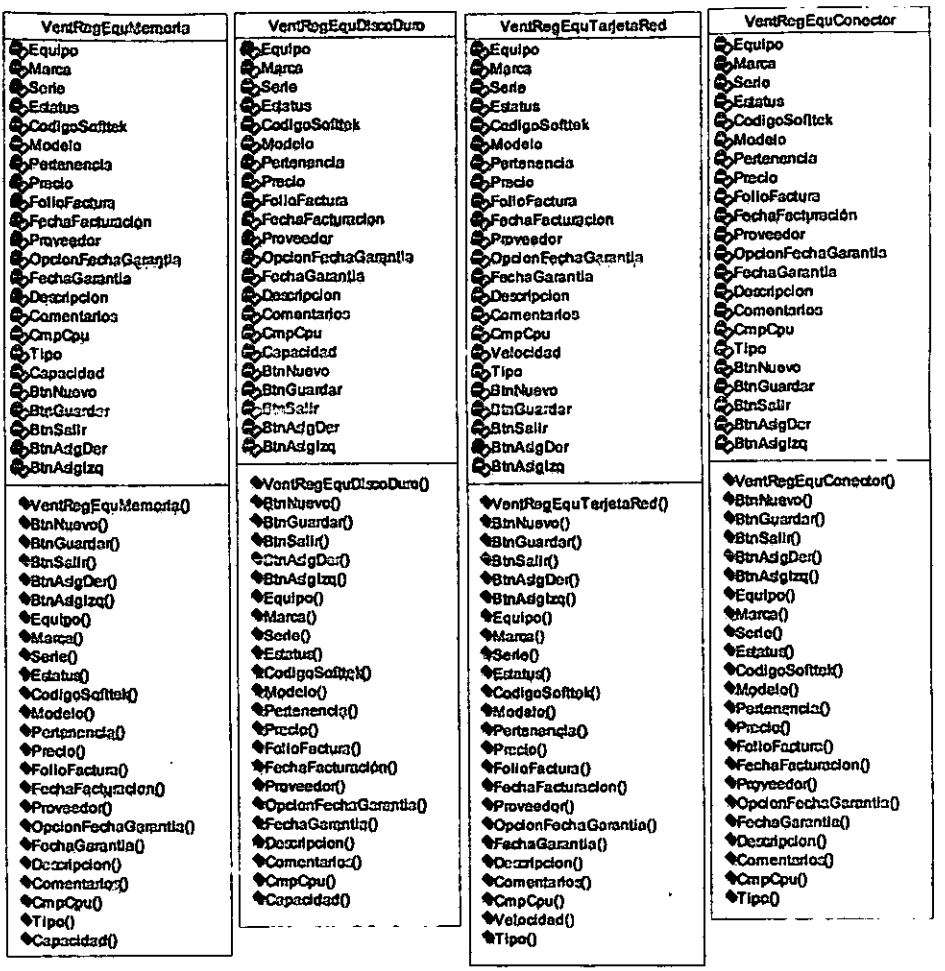
Clases del manejo de datos

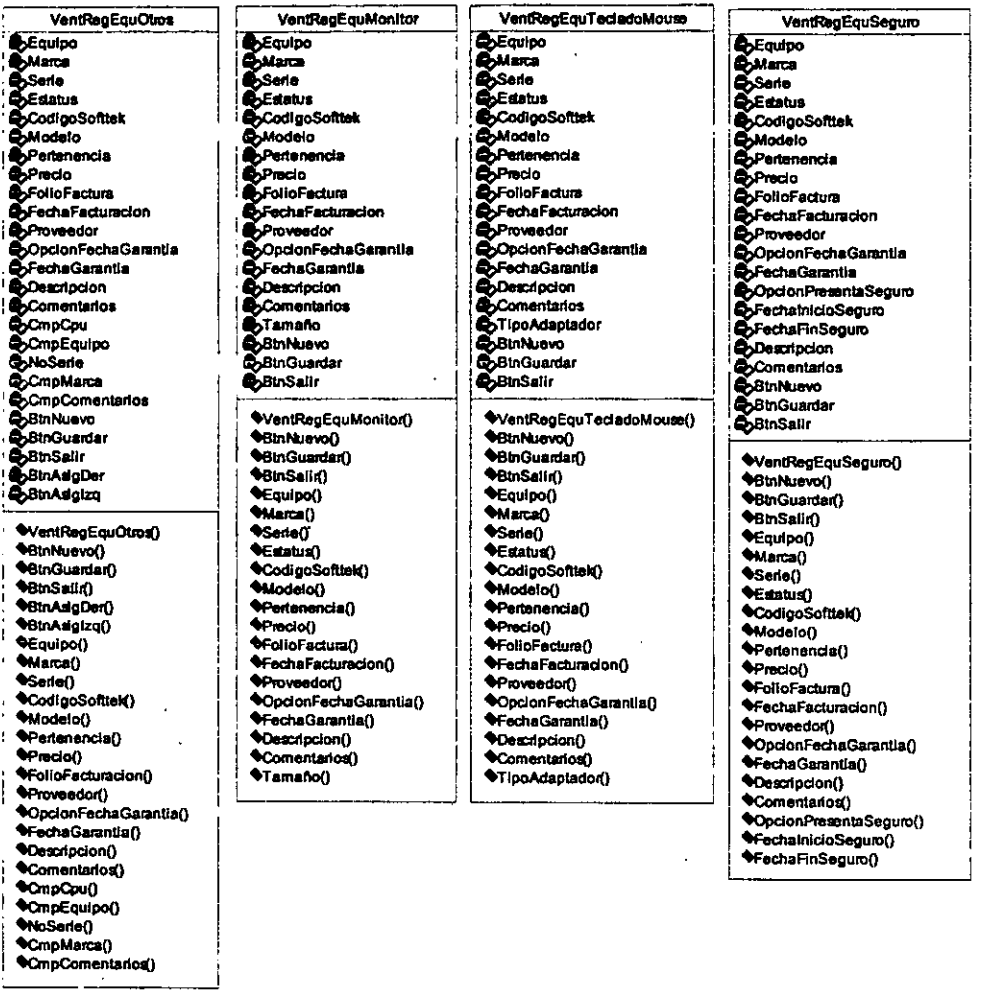


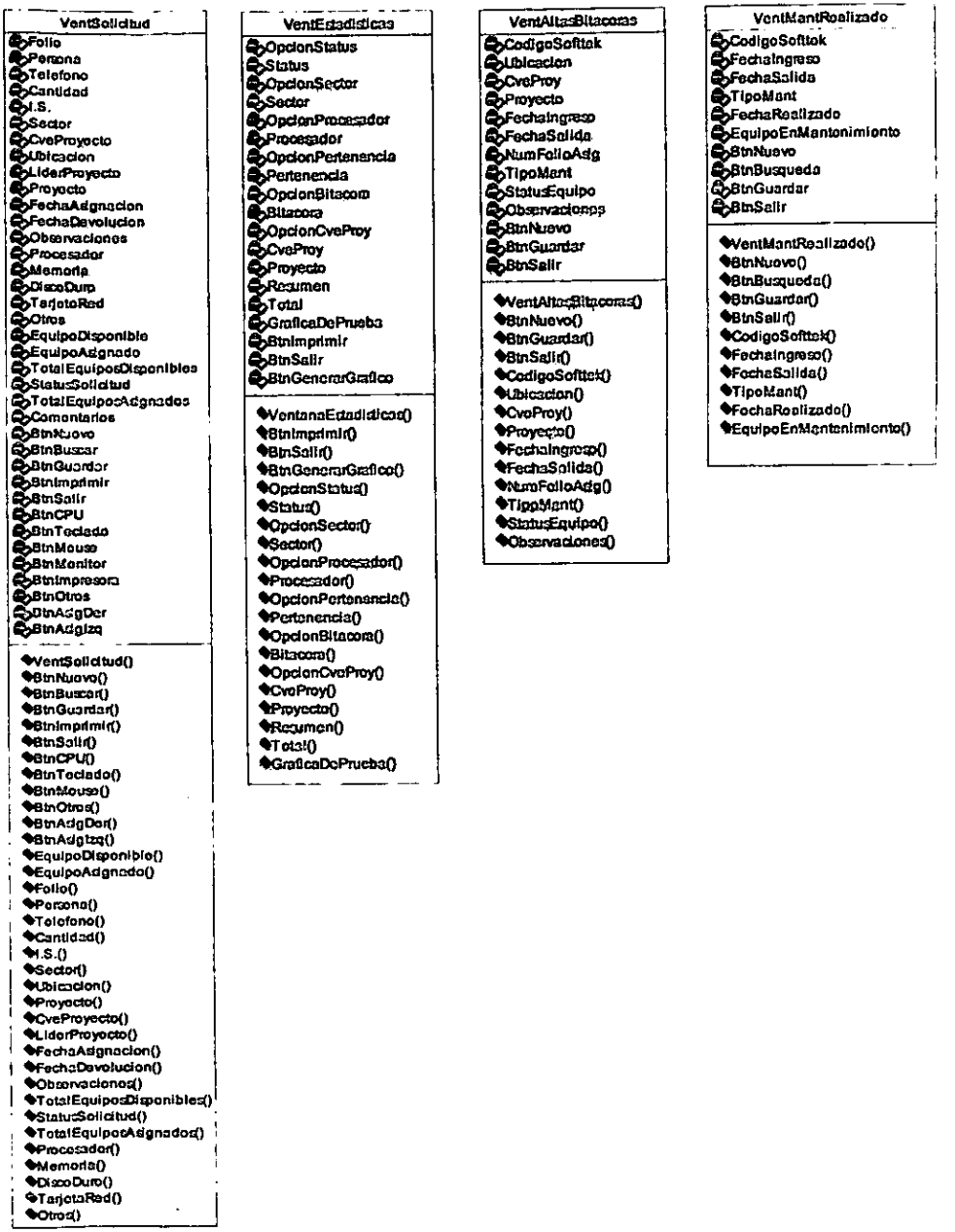


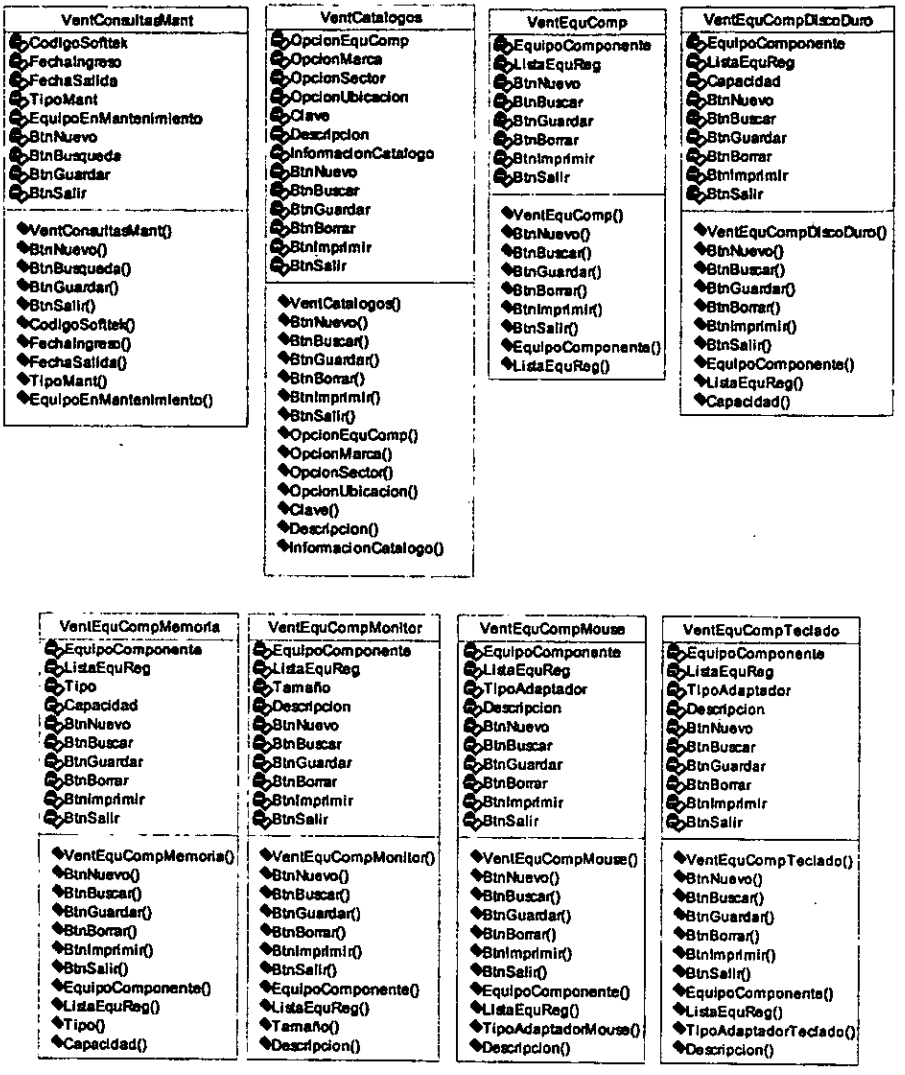
Clases de la interfaz humana

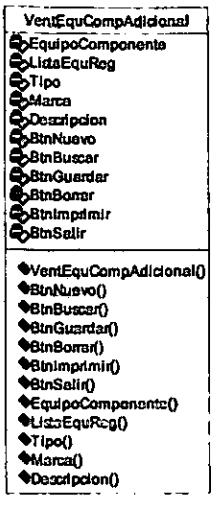


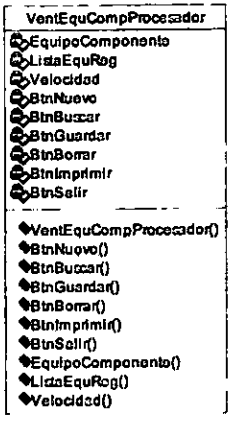

VentRegEquMemoria
Equipo
Marca
Serie
Estatus
CodigoSoftek
Modelo
Pertenencia
Precio
FolioFactura
FechaFacturacion
Proveedor
OpcionFechaGarantia
FechaGarantia
Descripcion
Comentarios
CmpCpu
Tipo
Capacidad
BtnNuevo
BtnGuardar
BtnSalir
BtnAdgDer
BtnAdgIzq
VentRegEquMemoria()
BtnNuevo()
BtnGuardar()
BtnSalir()
BtnAdgDer()
BtnAdgIzq()
Equipo()
Marca()
Serie()
Estatus()
CodigoSoftek()
Modelo()
Pertenencia()
Precio()
FolioFactura()
FechaFacturacion()
Proveedor()
OpcionFechaGarantia()
FechaGarantia()
Descripcion()
Comentarios()
CmpCpu()
Tipo()
Capacidad()
VentRegEquDiscoDuro
Equipo
Marca
Serie
Estatus
CodigoSoftek
Modelo
Pertenencia
Precio
FolioFactura
FechaFacturacion
Proveedor
OpcionFechaGarantia
FechaGarantia
Descripcion
Comentarios
CmpCpu
Capacidad
BtnNuevo
BtnGuardar
BtnSalir
BtnAdgDer
BtnAdgIzq
VentRegEquDiscoDuro()
BtnNuevo()
BtnGuardar()
BtnSalir()
BtnAdgDer()
BtnAdgIzq()
Equipo()
Marca()
Serie()
Estatus()
CodigoSoftek()
Modelo()
Pertenencia()
Precio()
FolioFactura()
FechaFacturación()
Proveedor()
OpcionFechaGarantia()
FechaGarantia()
Descripcion()
Comentarios()
CmpCpu()
Capacidad()
VentRegEquTarjetaRed
Equipo
Marca
Serie
Estatus
CodigoSoftek
Modelo
Pertenencia
Precio
FolioFactura
FechaFacturacion
Proveedor
OpcionFechaGarantia
FechaGarantia
Descripcion
Comentarios
CmpCpu
Velocidad
Tipo
BtnNuevo
BtnGuardar
BtnSalir
BtnAdgDer
BtnAdgIzq
VentRegEquTarjetaRed()
BtnNuevo()
BtnGuardar()
BtnSalir()
BtnAdgDer()
BtnAdgIzq()
Equipo()
Marca()
Serie()
Estatus()
CodigoSoftek()
Modelo()
Pertenencia()
Precio()
FolioFactura()
FechaFacturacion()
Proveedor()
OpcionFechaGarantia()
FechaGarantia()
Descripcion()
Comentarios()
CmpCpu()
Velocidad()
Tipo()
VentRegEquConector
Equipo
Marca
Serie
Estatus
CodigoSoftek
Modelo
Pertenencia
Precio
FolioFactura
FechaFacturación
Proveedor
OpcionFechaGarantia
FechaGarantia
Descripcion
Comentarios
CmpCpu
Tipo
BtnNuevo
BtnGuardar
BtnSalir
BtnAdgDer
BtnAdgIzq
VentRegEquConector()
BtnNuevo()
BtnGuardar()
BtnSalir()
BtnAdgDer()
BtnAdgIzq()
Equipo()
Marca()
Serie()
Estatus()
CodigoSoftek()
Modelo()
Pertenencia()
Precio()
FolioFactura()
FechaFacturacion()
Proveedor()
OpcionFechaGarantia()
FechaGarantia()
Descripcion()
Comentarios()
CmpCpu()
Tipo()

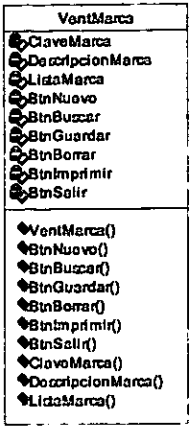

VentRegEquOtros
Equipo
Marca
Serie
Estatus
CodigoSoftek
Modelo
Pertenencia
Precio
FolioFactura
FechaFacturacion
Proveedor
OpcionFechaGarantia
FechaGarantia
Descripcion
Comentarios
CmpCpu
CmpEquipo
NoSerie
CmpMarca
CmpComentarios
BtnNuevo
BtnGuardar
BtnSalir
BtnAsigDer
BtnAsigIzq
VentRegEquOtros()
BtnNuevo()
BtnGuardar()
BtnSalir()
BtnAsigDer()
BtnAsigIzq()
Equipo()
Marca()
Serie()
CodigoSoftek()
Modelo()
Pertenencia()
Precio()
FolioFacturacion()
Proveedor()
OpcionFechaGarantia()
FechaGarantia()
Descripcion()
Comentarios()
CmpCpu()
CmpEquipo()
NoSerie()
CmpMarca()
CmpComentarios()
VentRegEquMonitor
Equipo
Marca
Serie
Estatus
CodigoSoftek
Modelo
Pertenencia
Precio
FolioFactura
FechaFacturacion
Proveedor
OpcionFechaGarantia
FechaGarantia
Descripcion
Comentarios
Tamaño
BtnNuevo
BtnGuardar
BtnSalir
VentRegEquMonitor()
BtnNuevo()
BtnGuardar()
BtnSalir()
Equipo()
Marca()
Serie()
Estatus()
CodigoSoftek()
Modelo()
Pertenencia()
Precio()
FolioFactura()
FechaFacturacion()
Proveedor()
OpcionFechaGarantia()
FechaGarantia()
Descripcion()
Comentarios()
Tamaño()
VentRegEquTecladoMouse
Equipo
Marca
Serie
Estatus
CodigoSoftek
Modelo
Pertenencia
Precio
FolioFactura
FechaFacturacion
Proveedor
OpcionFechaGarantia
FechaGarantia
Descripcion
Comentarios
TipoAdaptador
BtnNuevo
BtnGuardar
BtnSalir
VentRegEquTecladoMouse()
BtnNuevo()
BtnGuardar()
BtnSalir()
Equipo()
Marca()
Serie()
Estatus()
CodigoSoftek()
Modelo()
Pertenencia()
Precio()
FolioFactura()
FechaFacturacion()
Proveedor()
OpcionFechaGarantia()
FechaGarantia()
Descripcion()
Comentarios()
TipoAdaptador()
VentRegEquSeguro
Equipo
Marca
Serie
Estatus
CodigoSoftek
Modelo
Pertenencia
Precio
FolioFactura
FechaFacturacion
Proveedor
OpcionFechaGarantia
FechaGarantia
OpcionPresentaSeguro
FechaInicioSeguro
FechaFinSeguro
Descripcion
Comentarios
BtnNuevo
BtnGuardar
BtnSalir
VentRegEquSeguro()
BtnNuevo()
BtnGuardar()
BtnSalir()
Equipo()
Marca()
Serie()
Estatus()
CodigoSoftek()
Modelo()
Pertenencia()
Precio()
FolioFactura()
FechaFacturacion()
Proveedor()
OpcionFechaGarantia()
FechaGarantia()
Descripcion()
Comentarios()
OpcionPresentaSeguro()
FechaInicioSeguro()
FechaFinSeguro()

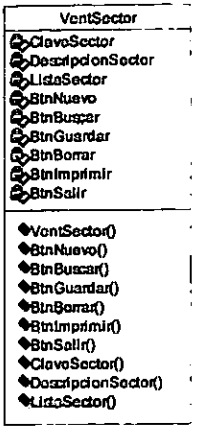

VentSolicitud
Folio
Persona
Telefono
Cantidad
H.S.
Sector
CveProyecto
Ubicacion
LiderProyecto
Proyecto
FechaAsignacion
FechaDevolucion
Observaciones
Procesador
Memoria
DiscoDuro
TarjetaRed
Otros
EquipoDisponible
EquipoAsignado
TotalEquiposDisponibles
StatusSolicitud
TotalEquiposAsignados
Comentarios
BtnNuevo
BtnBuscar
BtnGuardar
BtnImprimir
BtnSalir
BtnCPU
BtnTeclado
BtnMouse
BtnMonitor
BtnImpresora
BtnOtros
BtnAdgDer
BtnAdgIzq
VentSolicitud()
BtnNuevo()
BtnBuscar()
BtnGuardar()
BtnImprimir()
BtnSalir()
BtnCPU()
BtnTeclado()
BtnMouse()
BtnOtros()
BtnAdgDer()
BtnAdgIzq()
EquipoDisponible()
EquipoAsignado()
Folio()
Persona()
Telefono()
Cantidad()
H.S.()
Sector()
Ubicacion()
Proyecto()
CveProyecto()
LiderProyecto()
FechaAsignacion()
FechaDevolucion()
Observaciones()
TotalEquiposDisponibles()
StatusSolicitud()
TotalEquiposAsignados()
Procesador()
Memoria()
DiscoDuro()
TarjetaRed()
Otros()
VentEstadisticas
OpcionStatus
Status
OpcionSector
Sector
OpcionProcesador
Procesador
OpcionPertenencia
Pertenencia
OpcionBitacora
Bitacora
OpcionCveProy
CveProy
Proyecto
Resumen
Total
GraficaDePrueba
BtnImprimir
BtnSalir
BtnGenerarGrafico
VentanaEstadisticas()
BtnImprimir()
BtnSalir()
BtnGenerarGrafico()
OpcionStatus()
Status()
OpcionSector()
Sector()
OpcionProcesador()
Procesador()
OpcionPertenencia()
Pertenencia()
OpcionBitacora()
Bitacora()
OpcionCveProy()
CveProy()
Proyecto()
Resumen()
Total()
GraficaDePrueba()
VentAltasBitacoras
CodigoSoftek
Ubicacion
CveProy
Proyecto
FechaIngreso
FechaSalida
NumFolioAdg
TipoMant
StatusEquipo
Observacionpp
BtnNuevo
BtnGuardar
BtnSalir
VentAltasBitacoras()
BtnNuevo()
BtnGuardar()
BtnSalir()
CodigoSoftek()
Ubicacion()
CveProy()
Proyecto()
FechaIngreso()
FechaSalida()
NumFolioAdg()
TipoMant()
StatusEquipo()
Observaciones()
VentMantRealizado
CodigoSoftek
FechaIngreso
FechaSalida
TipoMant
FechaRealizado
EquipoEnMantenimiento
BtnNuevo
BtnBusqueda
BtnGuardar
BtnSalir
VentMantRealizado()
BtnNuevo()
BtnBusqueda()
BtnGuardar()
BtnSalir()
CodigoSoftek()
FechaIngreso()
FechaSalida()
TipoMant()
FechaRealizado()
EquipoEnMantenimiento()

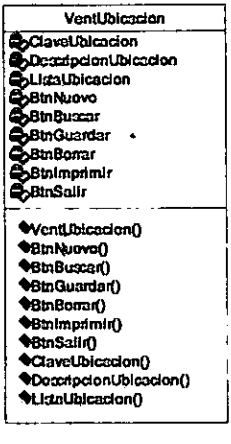

VentConsultasMant
CodigoSoftek
FechaIngreso
FechaSalida
TipoMant
EquipoEnMantenimiento
BtnNuevo
BtnBusqueda
BtnGuardar
BtnSalir
VentConsultasMant()
BtnNuevo()
BtnBusqueda()
BtnGuardar()
BtnSalir()
CodigoSoftek()
FechaIngreso()
FechaSalida()
TipoMant()
EquipoEnMantenimiento()
VentCatalogos
OpcionEquComp
OpcionMarca
OpcionSector
OpcionUbicacion
Clave
Descripcion
InformacionCatalogo
BtnNuevo
BtnBuscar
BtnGuardar
BtnBorrar
BtnImprimir
BtnSalir
VentCatalogos()
BtnNuevo()
BtnBuscar()
BtnGuardar()
BtnBorrar()
BtnImprimir()
BtnSalir()
OpcionEquComp()
OpcionMarca()
OpcionSector()
OpcionUbicacion()
Clave()
Descripcion()
InformacionCatalogo()
VentEquComp
EquipoComponente
ListaEquReg
BtnNuevo
BtnBuscar
BtnGuardar
BtnBorrar
BtnImprimir
BtnSalir
VentEquComp()
BtnNuevo()
BtnBuscar()
BtnGuardar()
BtnBorrar()
BtnImprimir()
BtnSalir()
EquipoComponente()
ListaEquReg()
VentEquCompDiscoDuro
EquipoComponente
ListaEquReg
Capacidad
BtnNuevo
BtnBuscar
BtnGuardar
BtnBorrar
BtnImprimir
BtnSalir
VentEquCompDiscoDuro()
BtnNuevo()
BtnBuscar()
BtnGuardar()
BtnBorrar()
BtnImprimir()
BtnSalir()
EquipoComponente()
ListaEquReg()
Capacidad()
VentEquCompMemoria
EquipoComponente
ListaEquReg
Tipo
Capacidad
BtnNuevo
BtnBuscar
BtnGuardar
BtnBorrar
BtnImprimir
BtnSalir
VentEquCompMemoria()
BtnNuevo()
BtnBuscar()
BtnGuardar()
BtnBorrar()
BtnImprimir()
BtnSalir()
EquipoComponente()
ListaEquReg()
Tipo()
Capacidad()
VentEquCompMonitor
EquipoComponente
ListaEquReg
Tamaño
Descripcion
BtnNuevo
BtnBuscar
BtnGuardar
BtnBorrar
BtnImprimir
BtnSalir
VentEquCompMonitor()
BtnNuevo()
BtnBuscar()
BtnGuardar()
BtnBorrar()
BtnImprimir()
BtnSalir()
EquipoComponente()
ListaEquReg()
Tamaño()
Descripcion()
VentEquCompMouse
EquipoComponente
ListaEquReg
TipoAdaptador
Descripcion
BtnNuevo
BtnBuscar
BtnGuardar
BtnBorrar
BtnImprimir
BtnSalir
VentEquCompMouse()
BtnNuevo()
BtnBuscar()
BtnGuardar()
BtnBorrar()
BtnImprimir()
BtnSalir()
EquipoComponente()
ListaEquReg()
TipoAdaptadorMouse()
Descripcion()
VentEquCompTeclado
EquipoComponente
ListaEquReg
TipoAdaptador
Descripcion
BtnNuevo
BtnBuscar
BtnGuardar
BtnBorrar
BtnImprimir
BtnSalir
VentEquCompTeclado()
BtnNuevo()
BtnBuscar()
BtnGuardar()
BtnBorrar()
BtnImprimir()
BtnSalir()
EquipoComponente()
ListaEquReg()
TipoAdaptadorTeclado()
Descripcion()

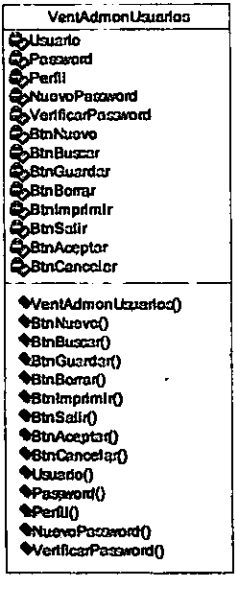

VentEquCompAdicional
EquipoComponente
ListaEquReg
Tipo
Marca
Descripcion
BtnNuevo
BtnBuscar
BtnGuardar
BtnBorrar
BtnImprimir
BtnSalir
VentEquCompAdicional()
BtnNuevo()
BtnBuscar()
BtnGuardar()
BtnBorrar()
BtnImprimir()
BtnSalir()
EquipoComponente()
ListaEquReg()
Tipo()
Marca()
Descripcion()


VentEquCompProcesador
EquipoComponente
ListaEquReg
Velocidad
BtnNuevo
BtnBuscar
BtnGuardar
BtnBorrar
BtnImprimir
BtnSalir
VentEquCompProcesador()
BtnNuevo()
BtnBuscar()
BtnGuardar()
BtnBorrar()
BtnImprimir()
BtnSalir()
EquipoComponente()
ListaEquReg()
Velocidad()


VentMarca
ClaveMarca
DescripcionMarca
ListaMarca
BtnNuevo
BtnBuscar
BtnGuardar
BtnBorrar
BtnImprimir
BtnSalir
VentMarca()
BtnNuevo()
BtnBuscar()
BtnGuardar()
BtnBorrar()
BtnImprimir()
BtnSalir()
ClaveMarca()
DescripcionMarca()
ListaMarca()

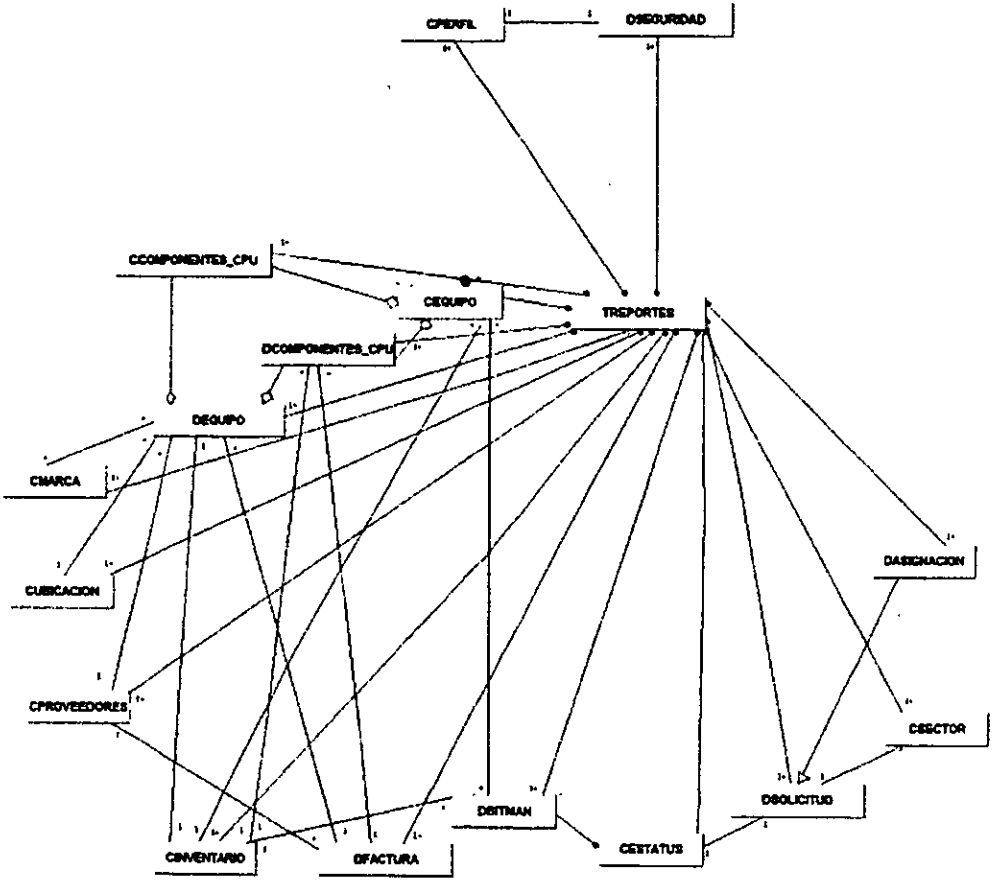

VentSector
ClaveSector
DescripcionSector
ListaSector
BtnNuevo
BtnBuscar
BtnGuardar
BtnBorrar
BtnImprimir
BtnSalir
VentSector()
BtnNuevo()
BtnBuscar()
BtnGuardar()
BtnBorrar()
BtnImprimir()
BtnSalir()
ClaveSector()
DescripcionSector()
ListaSector()


VentUbicacion
ClaveUbicacion
DescripcionUbicacion
ListaUbicacion
BtnNuevo
BtnBuscar
BtnGuardar
BtnBorrar
BtnImprimir
BtnSalir
VentUbicacion()
BtnNuevo()
BtnBuscar()
BtnGuardar()
BtnBorrar()
BtnImprimir()
BtnSalir()
ClaveUbicacion()
DescripcionUbicacion()
ListaUbicacion()


VentAdmonUsuarios
Usuario
Password
Perfil
NuevoPassword
VerificarPassword
BtnNuevo
BtnBuscar
BtnGuardar
BtnBorrar
BtnImprimir
BtnSalir
BtnAceptar
BtnCancelar
VentAdmonUsuarios()
BtnNuevo()
BtnBuscar()
BtnGuardar()
BtnBorrar()
BtnImprimir()
BtnSalir()
BtnAceptar()
BtnCancelar()
Usuario()
Password()
Perfil()
NuevoPassword()
VerificarPassword()

4.1.9. Diagrama principal

El siguiente diagrama muestra las 17 clases y su relación entre ellas de acuerdo a la simbología especificada anteriormente. Solamente se muestran los nombres de las clases, los métodos y los atributos ya se señalaron.



4.1.10. Diagrama de estados

Un diagrama de estados sirve para mostrar los sucesos enviados y recibidos por un objeto, con lo que se tiene el comportamiento dinámico de dicho objeto. Un diagrama de estados representa el comportamiento dinámico de todas las clases de objetos que intervienen en él. (Utilización de estándares y técnicas para el desarrollo del software. Según IEEE Estándar 730 y 983. Herramientas, técnicas y metodologías. Butler Estándares y convenciones.

A continuación se procederá a mostrar los diagramas de estados del sistema de información referentes a sus clases.

Diagrama de la Clase CCOMPONENTES_CPU

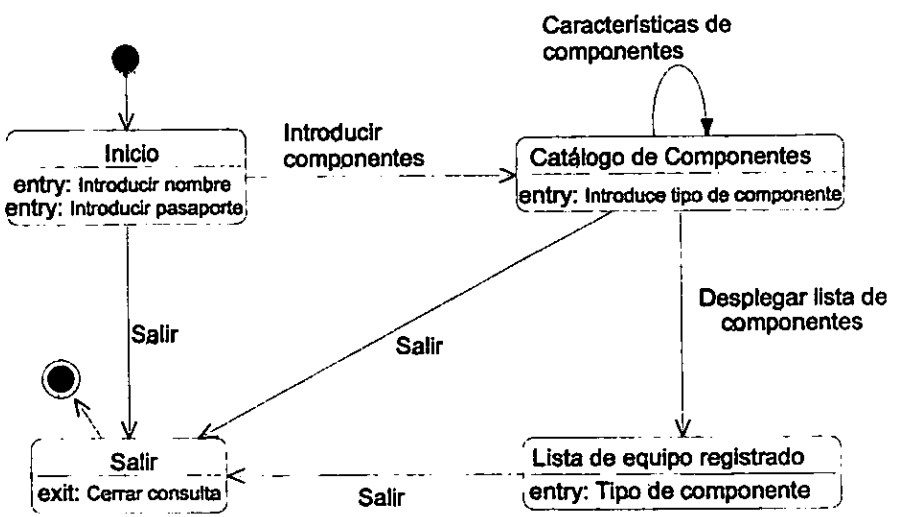


Diagrama de la Clase CEQUIPO

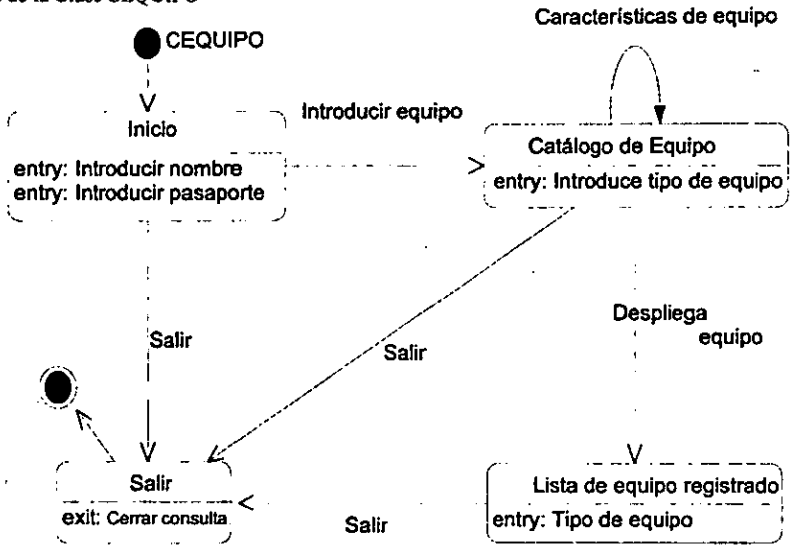


Diagrama de la Clase CESTATUS

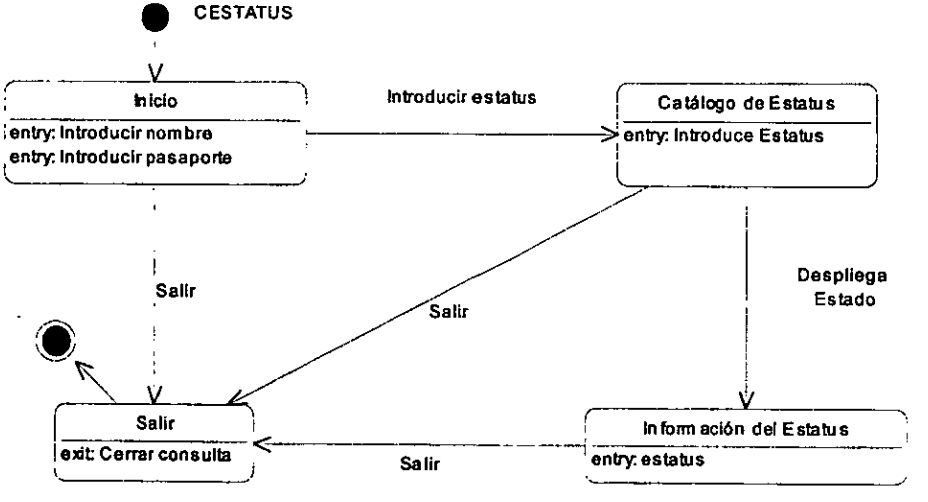


Diagrama de la Clase CINVENTARIO

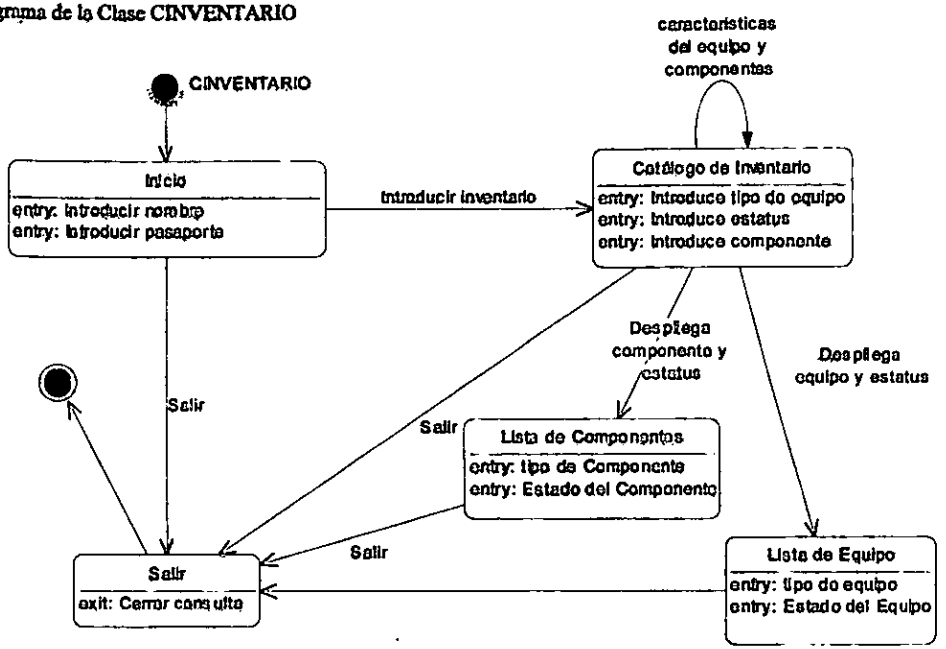


Diagrama de la Clase MARCA

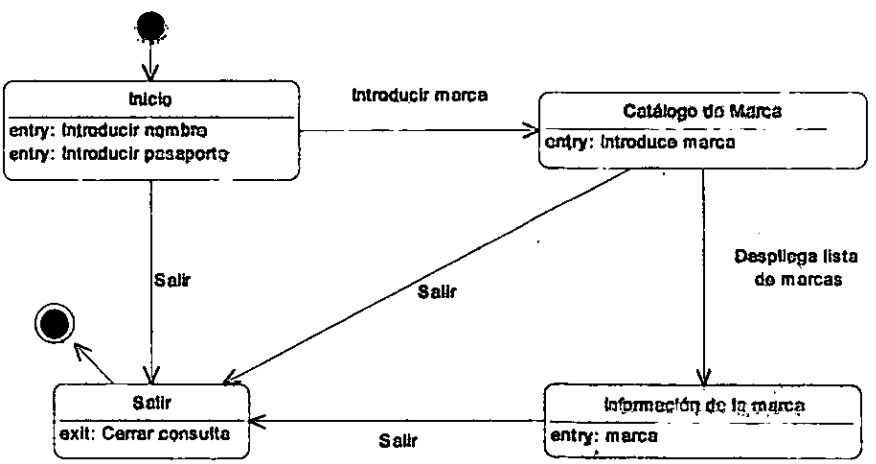


Diagrama de la Clase PERFIL

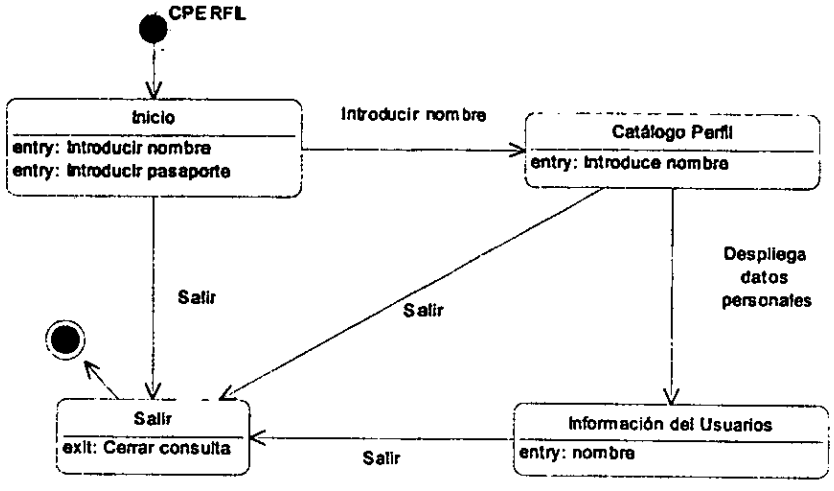


Diagrama de la Clase CPROVEEDORES

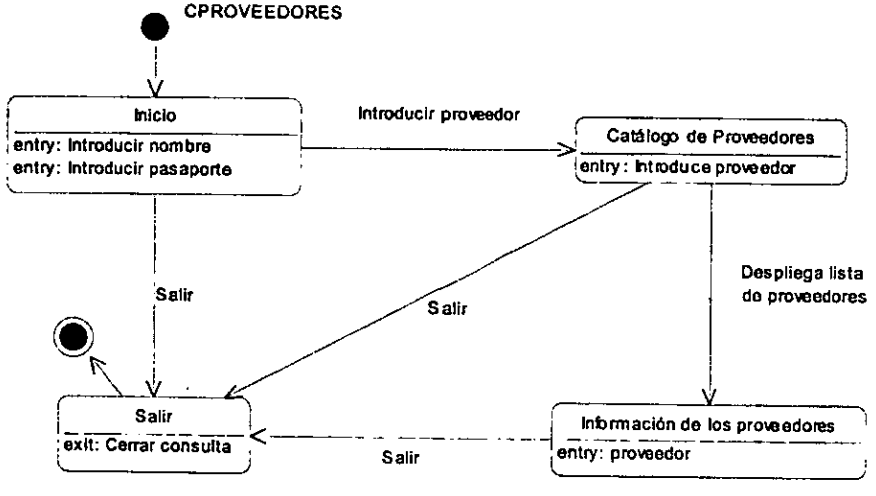


Diagrama de la Clase CSECTOR

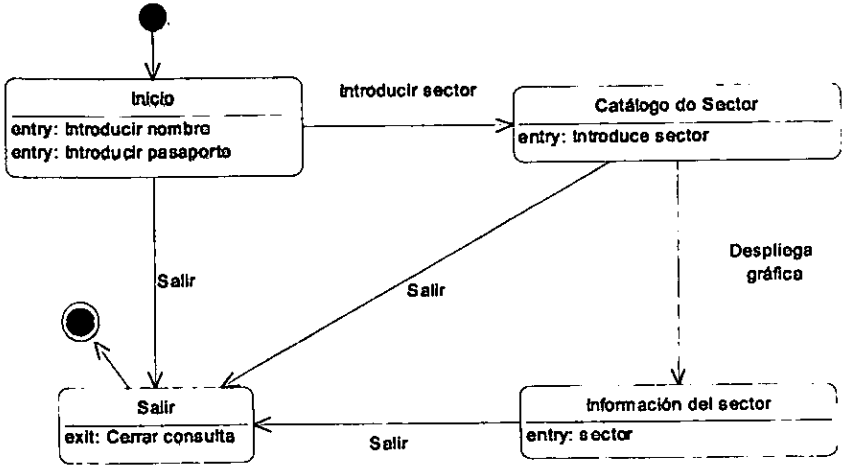


Diagrama de la Clase CUBICACIÓN

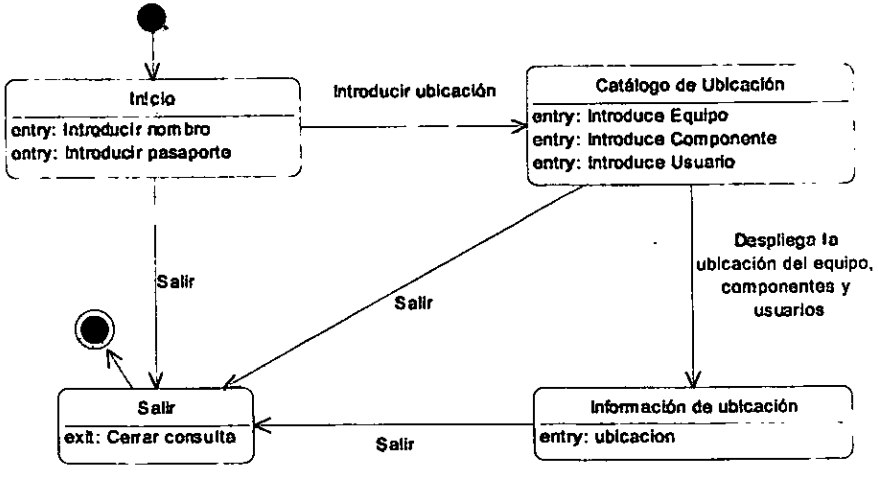


Diagrama de la Clase DASIGNACION

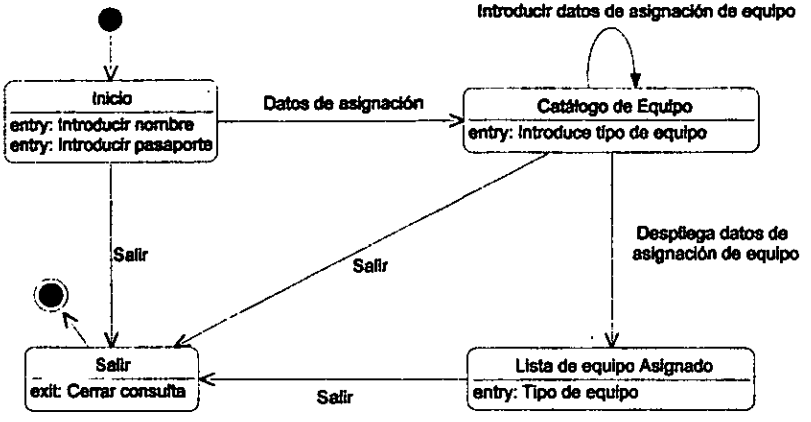


Diagrama de la Clase DBITMAN

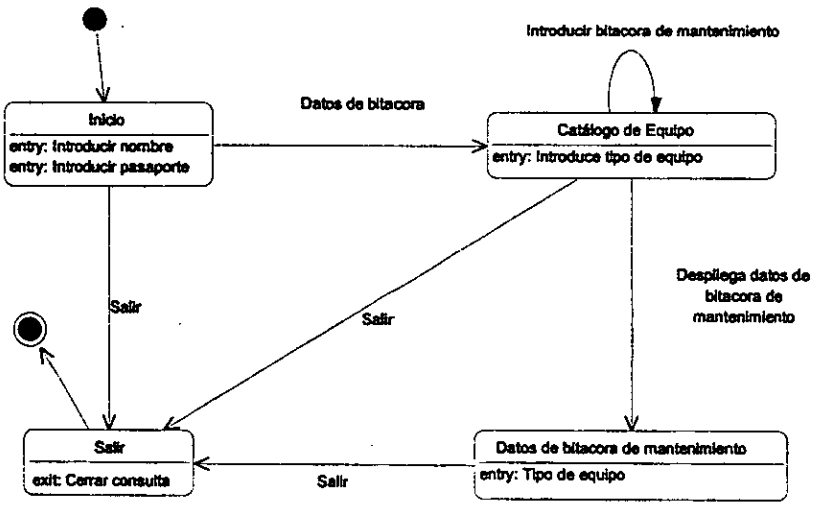


Diagrama de la Clase DCOMPONENTESCPU

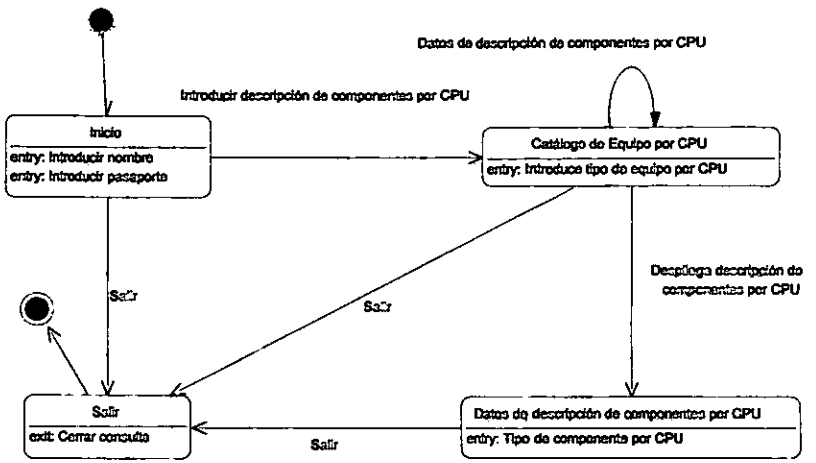


Diagrama de la Clase DEQUIPO

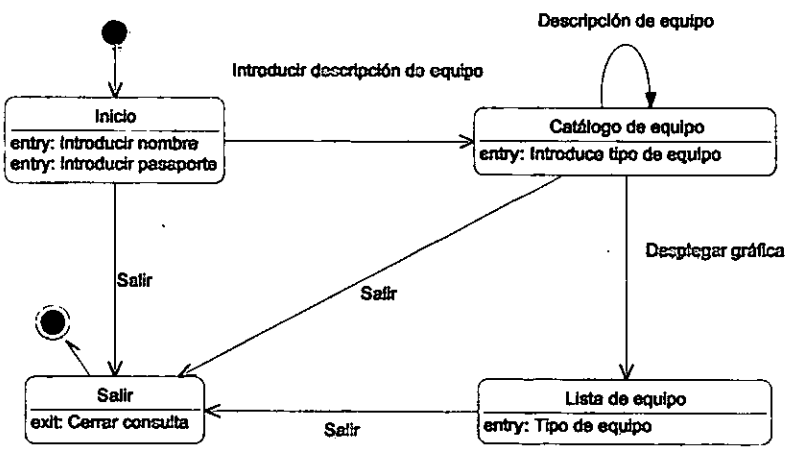


Diagrama de la Clase DFACTURA

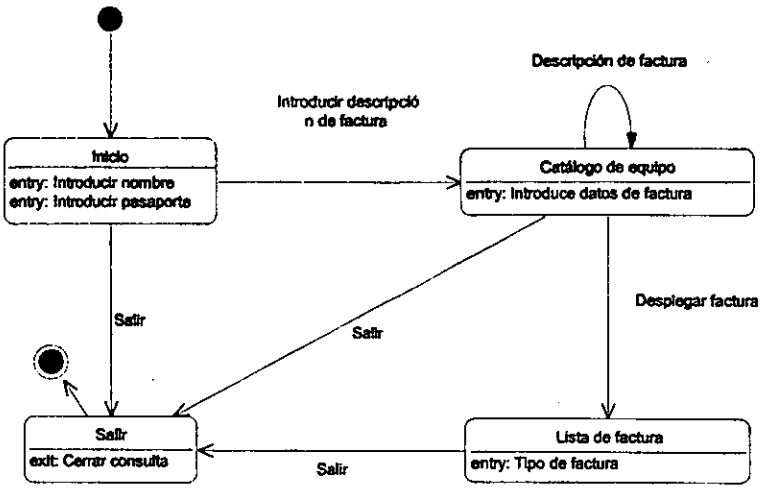


Diagrama de la Clase DSEGURIDAD

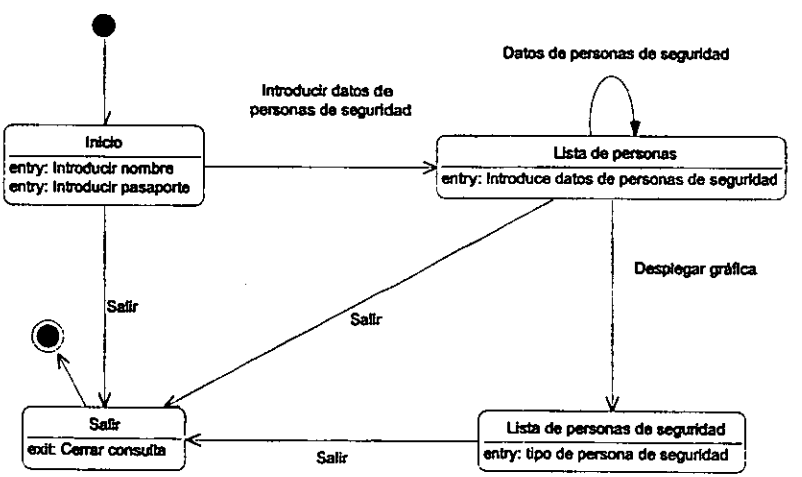


Diagrama de la Clase DSOLICITUD

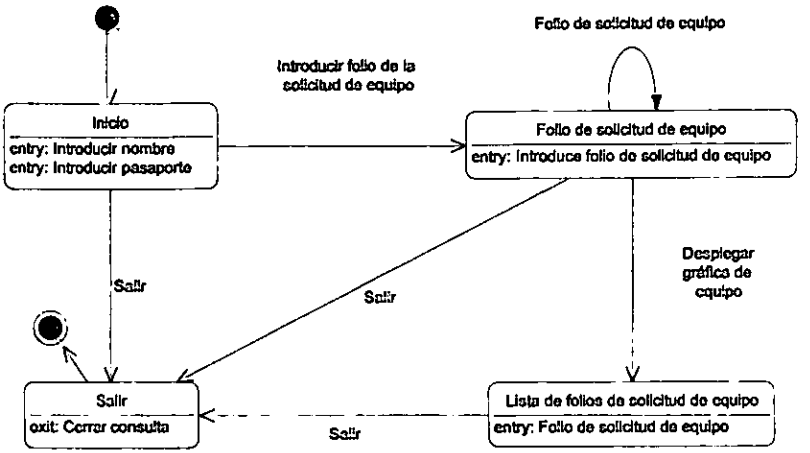
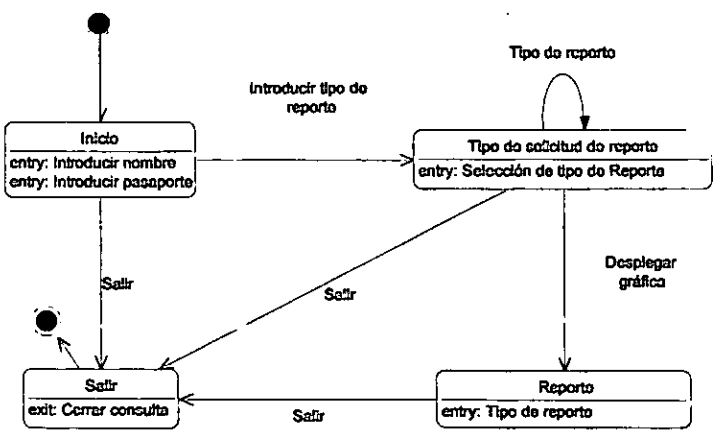


Diagrama de la Clase TREPORTES



4.2. Diseño

La fase de diseño de objetos determina las definiciones completas de las clases y asociaciones que se utilizarán en la implementación, así como las interfaces y algoritmos de los métodos utilizados para implementar las operaciones. La fase de diseño de objetos añadirá objetos internos para la implementación. (Se contemplan los métodos, técnicas , herramientas estándares para desarrollo de información. Según. IEEE Estándar 730 y 983 Estándares, Prácticas y convenciones. Butler (1995) Estándares y convenciones.)

En las siguientes secciones se desarrollará lo que es la etapa de diseño del sistema de información.

4.2.1. Patrones de Diseño

Para comenzar con esta sección, se iniciará por definir que es un marco de trabajo (framework), y un patrón de diseño.

Marco de trabajo (framework):

Es un subsistema expandible de un conjunto de servicios afines [AIS, 1977].

Patrón de diseño:

Es la descripción de un problema el cual ocurre una y otra vez durante el desarrollo de un sistema, y entonces describe el núcleo de la solución de dicho problema, en cuyo caso se puede usar dicha solución en forma repetitiva sin tener que volverla a desarrollar [AIS, 1977].

Con las dos definiciones anteriormente mencionadas se procederá a dar características más específicas de los marcos de trabajo y patrones de diseño.

En términos generales, un marco de trabajo:

- Es un conjunto cohesivo de clases que colaboran para prestar servicios a la parte fundamental e invariable de un subsistema lógico.
- Contiene clases concretas y, especialmente, abstractas que definen las interfaces a las cuales se conformarán, las interacciones en que participarán y otras invariantes.
- En términos generales, se requiere (aunque no necesariamente) que el usuario defina subclases de las actuales del marco de trabajo para que utilice, adapte y amplíe los servicios que le ofrece.
- Posee clases abstractas que pueden incluir métodos abstractos y concretos.
- Se basa en el **Principio de Hollywood**: " *No nos llame, nosotros le llamamos.* " Ello significa que las clases definidas por el usuario (entre ellas, las clases nuevas) recibirán mensajes de las clases previamente definidas del marco de trabajo. Suelen ser manejadas al implementar los métodos abstractos de las superclases.

En términos generales, un patrón de diseño consta de cuatro elementos fundamentales:

1. El **nombre del patrón** el cual sirve para describir un problema de diseño, sus soluciones, y consecuencias en una o dos palabras.

2. El **problema** descrito cuando se aplica el patrón. Se explica el problema y su contexto. Debe describir problemas de diseño específico así como la forma de presentar algoritmos como objetos.

3. La **solución** describe los elementos que constituyen el diseño, sus relaciones, responsabilidades, y colaboraciones. El patrón provee una descripción abstracta de un problema de diseño y como un arreglo general de elementos (clases y objetos en este caso) lo resuelven.

4. Las **consecuencias** son los resultados y el balance comparativo de aplicar el patrón. Aunque las consecuencias son a menudo ignoradas cuando describimos decisiones referentes al diseño, ellas resultan críticas para evaluar alternativas de diseño y entender los costos y beneficios de aplicar el patrón.

Diferencias entre marcos de trabajo (frameworks) y patrones de diseño

- *Los patrones de diseño son más abstractos que los marcos de trabajo.* Esto quiere decir que es más factible implementar y reutilizar un patrón de diseño que un marco de trabajo porque el patrón de diseño explica el propósito, el balance comparativo, y las consecuencias de un diseño.

- *Los patrones de diseño son elementos arquitectónicos menores que los marcos de trabajo.* Un marco de trabajo típico contiene varios patrones de diseño, pero lo contrario nunca ocurre.

- *Los patrones de diseño son menos especializados que los marcos de trabajo.* Los marcos de trabajo siempre tienen un dominio particular de aplicación. En contraste, los marcos de diseño pueden ser usados en cualquier tipo de aplicación.

4.2.1.1. Patrón de diseño a utilizar

Ya establecidas las características de los marcos de trabajo y patrones de diseño, así como sus diferencias, se expondrán las principales características del patrón de diseño a utilizar como parámetro para desarrollar el marco de trabajo del sistema de información.

Patrón de Diseño "Adaptador"

Propósito:

Convierte la interfaz de una clase en otra interfaz cliente deseada. Adaptador permite a las clases trabajar conjuntamente que en otras condiciones no podrían por la incompatibilidad en interfaces. También a este patrón se le conoce como "envolvente".

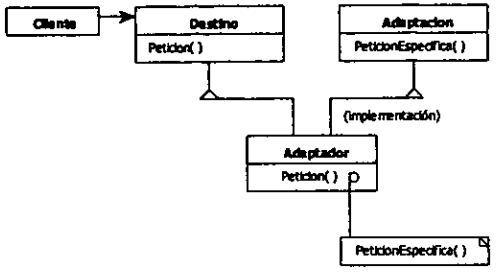
Aplicabilidad:

Se usa el patrón adaptador cuando:

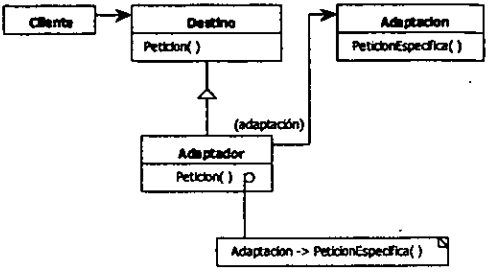
- Se necesita usar alguna clase existente, y sus interfaces no son compatibles con los requerimientos.
- Se requiere crear alguna clase reutilizable que coopere con clases no relacionadas o imprevistas, esto es, clases que no necesariamente tengan interfaces compatibles.
- (únicamente para objeto adaptador) se requiere el uso de varias clases existentes, pero resulta impractico el adaptar sus interfaces para cada subclase. Un objeto adaptador puede adaptar la interfaz de la clase origen de estas subclases.

Estructura:

Clase adaptadora usando herencia múltiple para adaptar de una interfaz a otra.



Un objeto adaptador implementado sobre un objeto compositor.



4.2.1.2.- Planteamiento del marco de trabajo (framework)

Se eligió el patrón de diseño adaptador como parámetro en el marco de trabajo del sistema de información porque, con su mecanismo de herencia múltiple, permite usar clases ya existentes sin tener que adaptar sus interfaces para cada subclase, con lo que dicho patrón de diseño resulta el más adecuado para aplicar en el marco de trabajo (framework) del sistema de información. (Estándares especificados que definen un conjunto de criterios o procedimientos de desarrollo que guíen la forma correcta para la elaboración del software. Según IEEE estándar 730 y 983. Documentación de requerimientos, diseño, verificación, gestión, desarrollo y procedimientos estándares. Herramientas técnicas y metodologías, Butler. Estándares y auditorias).

4.2.1.3.- Desarrollo del marco de trabajo (framework)

El marco de trabajo se desarrollará en tres secciones, la primera abarcará las clases del dominio del problema, la segunda será de clases referentes al manejo de datos, y la tercera se referirá a clases de la interfaz humana.

Por tratarse de metaclases generadoras de todo el sistema de información, se antepondrá el prefijo "meta" y posteriormente la denominación de la clase en el nombre. En el caso de las clases del manejo de datos se tendrá también el sufijo "MD" para indicar "Manejo de Datos". En cada metaclase se dará una explicación general de su propósito así como también se mencionarán las clases generadas al implementarse por medio del patrón de diseño adaptador.

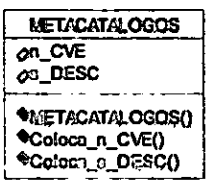
A continuación se procederá a desarrollar el marco de trabajo del sistema de información.

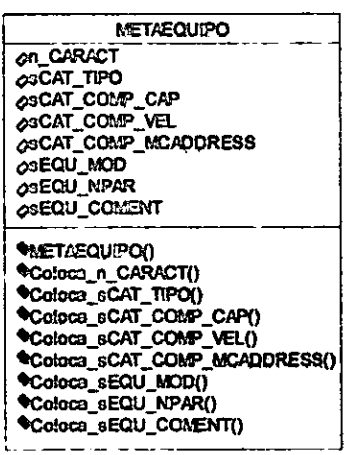
Marco de trabajo del dominio del problema

Se encarga de proporcionar los elementos necesarios en cada clase para la implementación de la información referente a catálogos del sistema.

Hereda a las siguientes clases:

CCAPTIPOCOMPCPU; CVELMARCCOMPCPU; CEQUIPO; CMARCA; CINVENTARIO; ESTATUS; CPERFIL; CPROVEEDORES; CSECTOR; CUBICACION; DCLASCARACTCOMPCPU; DLOCCARACTEQU; DBITMAN; DFACPROVCOMPCPU; DMODPROVEQU; DFACTURA; DSEGURIDAD; DFOLSOL; TREPINVGAREQU; TREPASIGDEVEQU; CDESCEQU; CMARC_EQU; CCOMPNOM; CCOMPINV; CMARCCOMP; CUBIEQU; DFACEQU; DCPUFAC; DESTBIT; DEQUBIT; DPROVFAC; CESTSOL; CSECSOL; CPERFILSEG; TEQUREP.

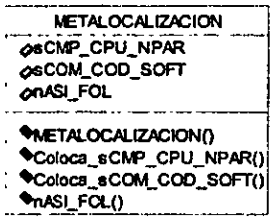




Se encarga de proporcionar los elementos necesarios en cada clase para la implementación de la información referente a las características del equipo.

Hereda a las siguientes clases:

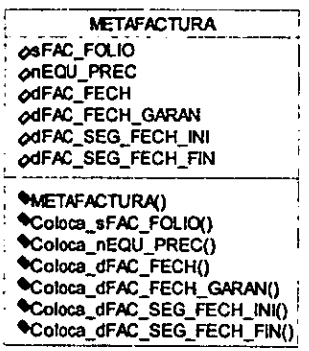
CCAPTIPOCOMPCPU; CVELMARCCOMPCPU; CEQUIPO; DCLASCARACTCOMPCPU; DLOCCARACTEQU; DBITMAN; DFACPROVCOMPCPU; DMODPROVEQU; CCOMPNOM; DEQUBIT.



Se encarga de proporcionar los elementos necesarios en cada clase para la implementación de la información referente a la clasificación y asignación de equipo.

Hereda a las siguientes clases:

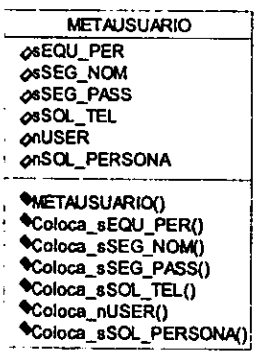
DCLASCARACTCOMPCPU; DASIGNACION; DLOCCARACTEQU; DBITMAN; DFACPROVCOMPCPU; DMODPROVEQU; TREPINVGAREQU; DEQUBIT; DASIGEQU.



Se encarga de proporcionar los elementos necesarios en cada clase para la implementación de la información referente a facturas de equipo.

Hereda a las siguientes clases:

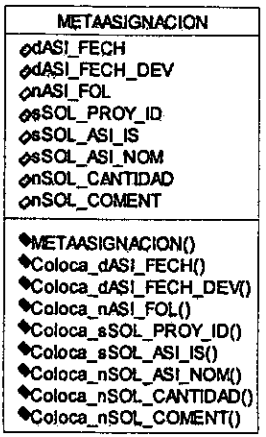
DFACTURA; DLOCCARACTEQU; DFACPROVCOMPCPU; TREPINVGAREQU; DFACEQU; DCPUFAC.



Se encarga de proporcionar los elementos necesarios en cada clase para la implementación de la información referente a los usuarios registrados dentro del sistema de información.

Hereda a las siguientes clases:

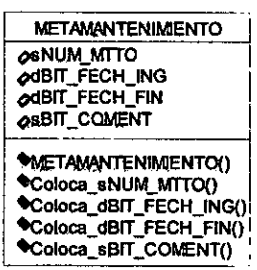
DLOCCARACTEQU; DSEGURIDAD; DFOLSOL; TREPINVGAREQU; TREPASIGDEVEQU; DPERSOL; TREPSEG.



Se encarga de proporcionar los elementos necesarios en cada clase para la implementación de la información referente a dónde y con quién estará el equipo, así como el tiempo de asignación.

Hereda a las siguientes clases:

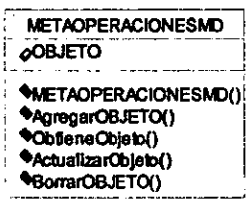
DASIGNACION; DFOLSOL; TREPASIGDEVEQU; DPERSOL; DSOLASIG.



Se encarga de proporcionar los elementos necesarios en cada clase para la implementación de la información referente a la reparación y reacondicionamiento del equipo.

Hereda a la siguiente clase:

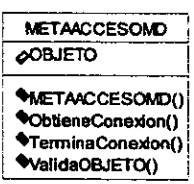
DBITMAN.



Se encarga de proporcionar a cada clase las funciones ú operaciones fundamentales de manipulación y manejo de la base de datos del sistema de información.

Hereda a las siguientes clases:

CCAPTIPOCOMPCPUMD; CVELMARCCOMPCPUMD;CEQUIPOMD; CESTATUSMD; CINVENTARIOMD; CMARCAMD; CPERFILMD; CPROVEEDORESMD; CSECTORMD; CUBICACIONMD; DASIGNACIONMD; DBITMANMD; DCLASCARACTCOMPCPUMD; DFACPROVCOMPCPUMD; DLOCCARACTEQUMD; DMODPROVEQUMD; DFACTURAMD; DSEGURIDADMD; DFOLSOLMD; DPERSOLMD; TREPINVGAREQUMD; TREPASIGDEVEQUMD; CDESCEQUMD; CMARCEQUMD; CCOMPINVMD; CMARCCOMPMD; CCOMPNOMMD; CUBIEQUMD; DFACEQUMD; DCPUFACMD; DEQUBITMD; DPROVFACMD; DESTBITMD; DESTSOLMD; DSOLASIGMD; DASIGEQUMD; CPERFILSEGMD; TREPSEGMD; TEQUREPMD; CSECSOLMD.

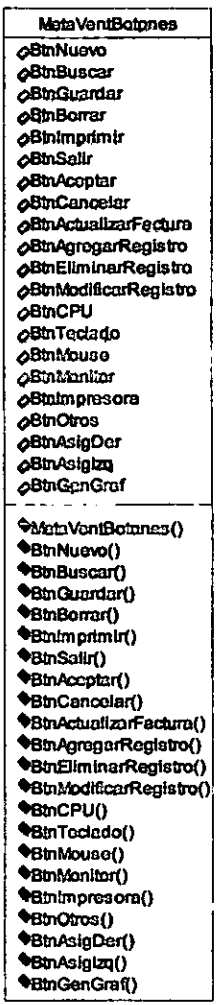


Se encarga de proporcionar las funciones ú operaciones fundamentales referentes a la comunicación y seguridad con la base de datos del sistema de información.

Hereda a las siguientes clases:

CPERFILMD; DSEGURIDADMD; CPERFILSEGMD; TREPSEGMD.

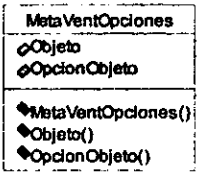
Marco de trabajo de la interfaz humana



Se encarga de proporcionar los elementos necesarios en cada clase para la implementación de funciones por medio de botones de acción.

Hereda a las siguientes clases:

VentAcceso;
VentRegFact;
VentRegEqu;
VentRegEquProcesador;
VentRegEquMemoria;
VentRegEquDiscoDuro;
VentRegEquTarjetaRed;
VentRegEquConector;
VentRegEquOtros;
VentRegEquMonitor;
VentRegEquTecladoMouse;
VentRegEquSeguro;
VentSolicitud;
VentEstadisticas;
VentAltasBitacoras;
VentMantRealizado;
VentConsultasMant;
VentCatalogos;
VentEquComp;
VentEquCompDiscoDuro;
VentEquCompMemoria;
VentEquCompMonitor;
VentEquCompMouse;
VentEquCompTeclado;
VentEquCompAdicional;
VentEquCompProcesador;
VentMarca;
VentSector;
VentUbicacion;
VentAdmonUsuarios.

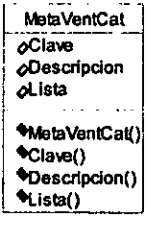


Se encarga de proporcionar los elementos necesarios en cada clase para implementar la activación o desactivación de una alternativa de llenado de información.

Hereda a las siguientes clases:

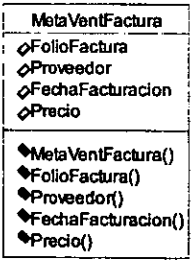
VentRegFact; VentRegEqu; VentRegEquProcesador; VentRegEquMemoria; VentRegEquDiscoDuro; VentRegEquTarjetaRed; VentRegEquConector; VentRegEquOtros; VentRegEquMonitor; VentRegEquTecladoMouse; VentRegEquSeguro; VentEstadisticas; VentCatalogos.

Se encarga de proporcionar los elementos necesarios en cada clase para manipular y visualizar información referente a catálogos.



Hereda a las siguientes clases:

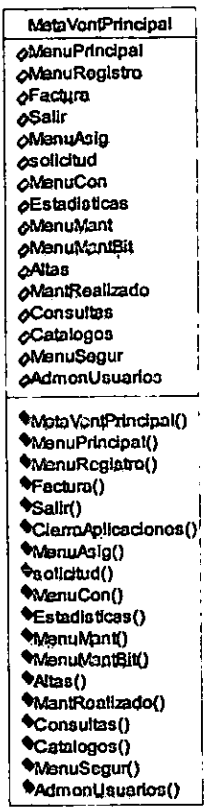
VentUbicacion; VentSector; VentMarca; VentEquCompProcesador; VentEquCompAdicional; VentEquCompTeclado; VentEquCompMouse; VentEquCompMonitor; VentEquCompMemoria; VentEquCompDiscoDuro; VentEquComp; VentCatalogos; VentRegEquSeguro; VentRegEquTecladoMouse; VentRegEquMonitor; VentRegEquOtros; VentRegEquConector; VentRegEquTarjetaRed; VentRegEquDiscoDuro; VentRegEquMemoria; VentRegEquProcesador; VentRegEqu; VentRegFact.



Se encarga de proporcionar los elementos en cada clase para la implementación de la visualización y llenado de los datos de las facturas de equipo.

Hereda a las siguientes clases:

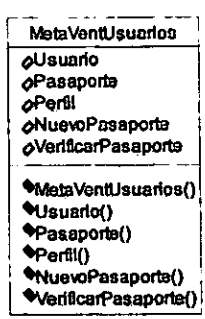
VentRegFact; VentRegEqu; VentRegEquProcesador; VentRegEquMemoria; VentRegEquDiscoDuro; VentRegEquTarjetaRed; VentRegEquConector; VentRegEquOtros; VentRegEquMonitor; VentRegEquTecladoMouse; VentRegEquSeguro.



Se encarga de proporcionar la accesibilidad a lo largo de las distintas aplicaciones del sistema de información.

Hereda a la siguiente clase:

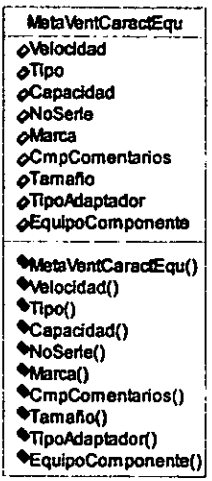
VentPrincipal.



Se encarga de proporcionar a las clases la capacidad de visualizar y manipular la información de los usuarios registrados en el sistema.

Hereda a las siguientes clases:

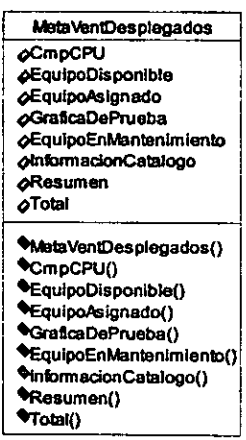
VentAdmonUsuarios; VentAcceso.



Se encarga de proporcionar elementos a las clases para visualizar y manipular las cualidades de los distintos componentes referentes al equipo.

Hereda a las siguientes clases:

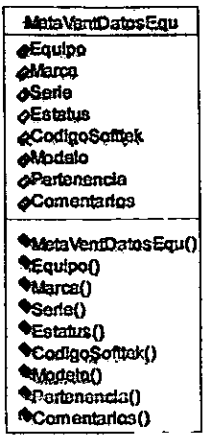
VentRegEquProcesador; VentRegEquMemoria;
VentRegEquDiscoDuro; VentRegEquTarjetaRed;
VentRegEquConector; VentRegEquOtros;
VentRegEquMonitor; VentRegEquTecladoMouse;
VentEquComp; VentEquCompDiscoDuro;
VentEquCompMemoria; VentEquCompMonitor;
VentEquCompMouse; VentEquCompTeclado;
VentEquCompAdicional; VentEquCompProcesador.



Se encarga de proporcionar a las clases la capacidad de desplegar varios elementos de equipo simultáneamente para su visualización y manipulación.

Hereda a las siguientes clases:

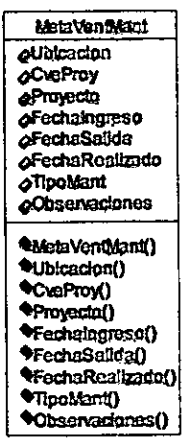
VentRegEquProcesador; VentRegEquMemoria;
VentRegEquDiscoDuro; VentRegEquTarjetaRed;
VentRegEquConector; VentRegEquOtros;
VentSolicitud; VentEstadisticas;
VentMantRealizado; VentConsultasMant;
VentCatalogos.



Se encarga de proporcionar a las clases la capacidad de visualizar e introducir datos referentes al equipo.

Hereda a las siguientes clases:

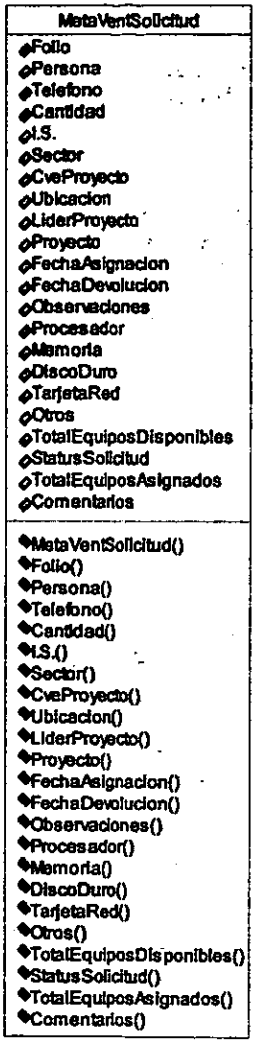
VentRegEqu;
VentRegEquMemoria;
VentRegEquTarjetaRed;
VentRegEquOtros;
VentRegEquTecladoMouse;
VentAltasBitacoras;
VentConsultasMant.
VentRegEquProcesador;
VentRegEquDiscoDuro;
VentRegEquConector;
VentRegEquMonitor;
VentRegEquSeguro;
VentMantRealizado;



Se encarga de proporcionar elementos a las clases para visualizar y manipular la información referente al equipo en mantenimiento.

Hereda a las siguientes clases:

VentAltasBitacoras; VentMantRealizado;
VentConsultasMant.



Se encarga de proporcionar elementos a las clases para visualizar y manipular la información referente al registro de asignación de equipo.

Hereda a las siguientes clases:

VentSolicitud; VentEstadisticas;
VentAltasBitacoras; VentMantRealizado;
VentConsultasMant.

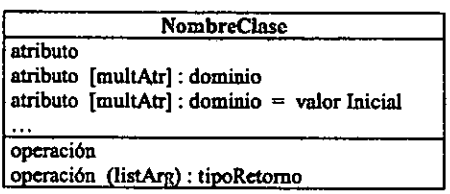
4.2.2.- Diseño de clases y objetos

La siguiente sección se desarrollará con el método OOA/OOD (Object-Oriented Analysis/Object-Oriented Design) y la notación UML (Unified Modeling Language). (Utilización de métodos, técnicas y herramientas que existen para el desarrollo del sistema de información. Según IEEE Estándar 730 y 983, Butler Herramientas, técnicas y metodologías. Stamm. Metodología de ingeniería del software. ISO 9000 – 3 Diseño e implementación).Del método OOA/OOD se va a usar únicamente OOD para identificar los siguientes grupos de componentes o clases:

- Clases del Dominio del Problema
- Clases del Manejo de Datos
- Clases de la Interfaz Humana

Utilización de métodos, técnicas y herramientas que existen para el desarrollo del sistema de información. Según IEEE Estándar 730 y 983, Butler Herramientas, técnicas y metodologías. Stamm. Metodología de ingeniería del software. ISO 9000 – 3 Diseño e implementación.

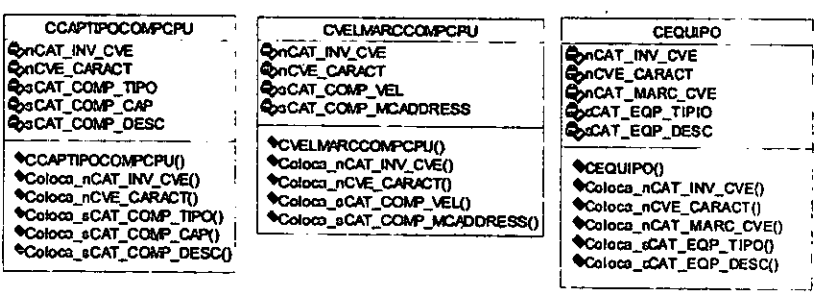
Y de UML se utilizará la notación gráfica mostrada a continuación para representar las clases:

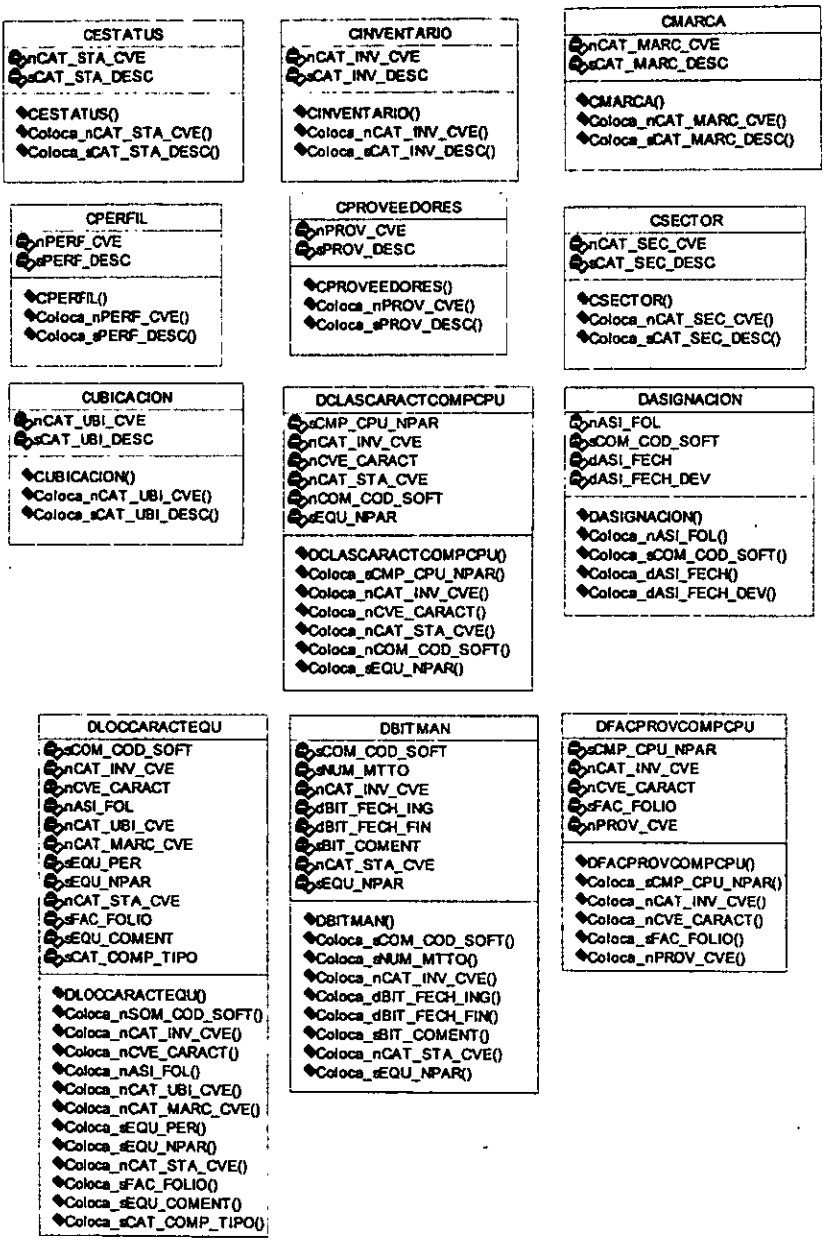


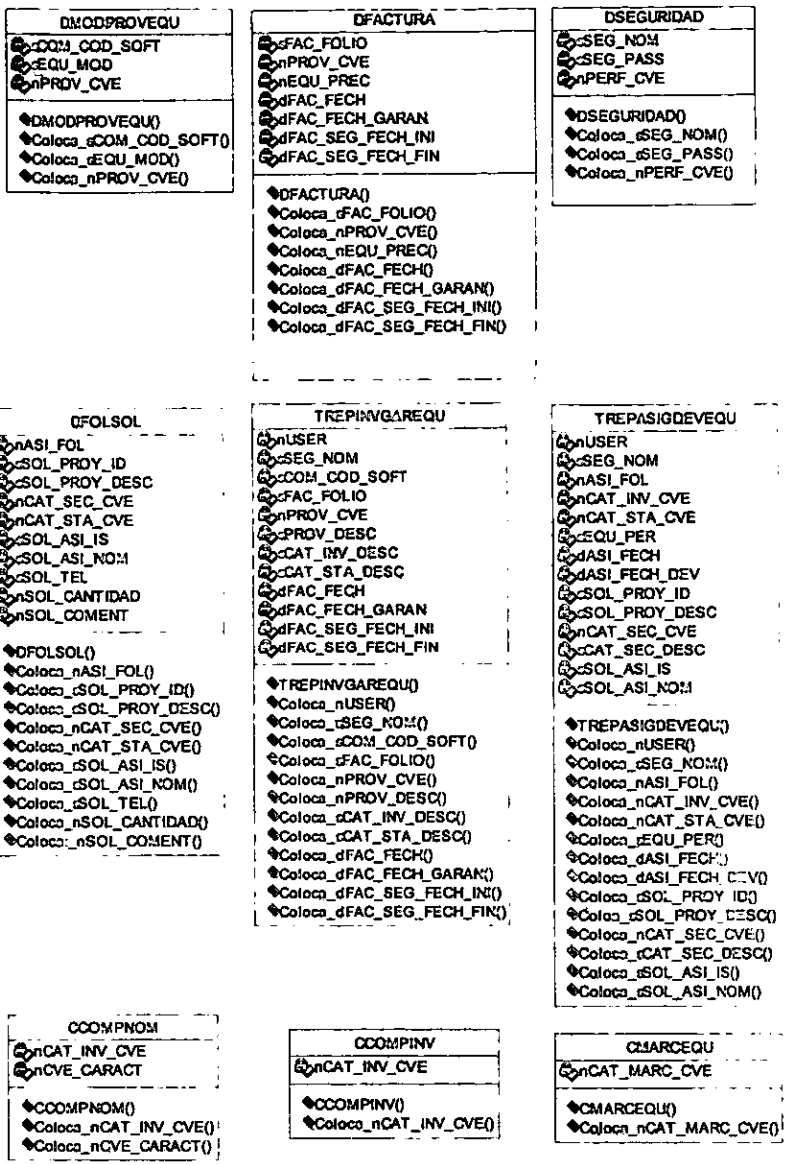
Notación Gráfica de Clases

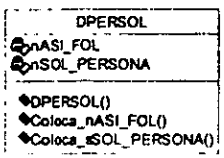
4.2.2.1.- Clases del Dominio del Problema

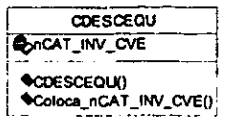
Se contendrán las clases ú objetos que directamente corresponden al problema que esta siendo modelado. Estos objetos son neutrales ya que tienen poco o ningún conocimiento acerca de los objetos contenidos en los demás componentes. También los atributos, los servicios o nuevas clases tienen que agregarse a las clases del espacio del problema. A continuación se mostrarán las clases del dominio del problema correspondientes al sistema de información.

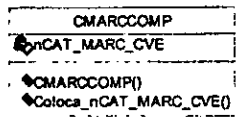


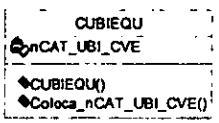

CESTATUS
nCAT_STA_CVE
sCAT_STA_DESC
CESTATUS()
Coloca_nCAT_STA_CVE()
Coloca_sCAT_STA_DESC()
CINVENTARIO
nCAT_INV_CVE
sCAT_INV_DESC
CINVENTARIO()
Coloca_nCAT_INV_CVE()
Coloca_sCAT_INV_DESC()
CMARCA
nCAT_MARC_CVE
sCAT_MARC_DESC
CMARCA()
Coloca_nCAT_MARC_CVE()
Coloca_sCAT_MARC_DESC()
CPERFIL
nPERF_CVE
sPERF_DESC
CPERFIL()
Coloca_nPERF_CVE()
Coloca_sPERF_DESC()
CPROVEEDORES
nPROV_CVE
sPROV_DESC
CPROVEEDORES()
Coloca_nPROV_CVE()
Coloca_sPROV_DESC()
CSECTOR
nCAT_SEC_CVE
sCAT_SEC_DESC
CSECTOR()
Coloca_nCAT_SEC_CVE()
Coloca_sCAT_SEC_DESC()
CUBICACION
nCAT_UBI_CVE
sCAT_UBI_DESC
CUBICACION()
Coloca_nCAT_UBI_CVE()
Coloca_sCAT_UBI_DESC()
DCLASCARACTCOMPCPU
sCMP_CPU_NPAR
nCAT_INV_CVE
nCVE_CARACT
nCAT_STA_CVE
nCOM_COD_SOFT
sEQU_NPAR
DCLASCARACTCOMPCPU()
Coloca_sCMP_CPU_NPAR()
Coloca_nCAT_INV_CVE()
Coloca_nCVE_CARACT()
Coloca_nCAT_STA_CVE()
Coloca_nCOM_COD_SOFT()
Coloca_sEQU_NPAR()
DASIGNACION
nASI_FOL
sCOM_COD_SOFT
dASI_FECH
dASI_FECH_DEV
DASIGNACION()
Coloca_nASI_FOL()
Coloca_sCOM_COD_SOFT()
Coloca_dASI_FECH()
Coloca_dASI_FECH_DEV()
DLOCCARACTEQU
sCOM_COD_SOFT
nCAT_INV_CVE
nCVE_CARACT
nASI_FOL
nCAT_UBI_CVE
nCAT_MARC_CVE
sEQU_PER
sEQU_NPAR
nCAT_STA_CVE
sFAC_FOLIO
sEQU_COMENT
sCAT_COMP_TIPO
DLOCCARACTEQU()
Coloca_nSOM_COD_SOFT()
Coloca_nCAT_INV_CVE()
Coloca_nCVE_CARACT()
Coloca_nASI_FOL()
Coloca_nCAT_UBI_CVE()
Coloca_nCAT_MARC_CVE()
Coloca_sEQU_PER()
Coloca_sEQU_NPAR()
Coloca_nCAT_STA_CVE()
Coloca_sFAC_FOLIO()
Coloca_sEQU_COMENT()
Coloca_sCAT_COMP_TIPO()
DBITMAN
sCOM_COD_SOFT
sNUM_MTTO
nCAT_INV_CVE
dBIT_FECH_ING
dBIT_FECH_FIN
sBIT_COMENT
nCAT_STA_CVE
sEQU_NPAR
DBITMAN()
Coloca_sCOM_COD_SOFT()
Coloca_sNUM_MTTO()
Coloca_nCAT_INV_CVE()
Coloca_dBIT_FECH_ING()
Coloca_dBIT_FECH_FIN()
Coloca_sBIT_COMENT()
Coloca_nCAT_STA_CVE()
Coloca_sEQU_NPAR()
DFACPROVCOMPCPU
sCMP_CPU_NPAR
nCAT_INV_CVE
nCVE_CARACT
sFAC_FOLIO
nPROV_CVE
DFACPROVCOMPCPU()
Coloca_sCMP_CPU_NPAR()
Coloca_nCAT_INV_CVE()
Coloca_nCVE_CARACT()
Coloca_sFAC_FOLIO()
Coloca_nPROV_CVE()

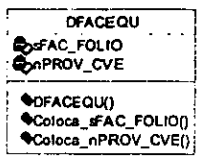

DMODPROVEQU
cCOM_COD_SOFT
cEQU_MOD
nPROV_CVE
DMODPROVEQU()
Coloca_cCOM_COD_SOFT()
Coloca_cEQU_MOD()
Coloca_nPROV_CVE()
DFACTURA
cFAC_FOLIO
nPROV_CVE
nEQU_PREC
dFAC_FECH
dFAC_FECH_GARAN
dFAC_SEG_FECH_INI
dFAC_SEG_FECH_FIN
DFACTURA()
Coloca_cFAC_FOLIO()
Coloca_nPROV_CVE()
Coloca_nEQU_PREC()
Coloca_dFAC_FECH()
Coloca_dFAC_FECH_GARAN()
Coloca_dFAC_SEG_FECH_INI()
Coloca_dFAC_SEG_FECH_FIN()
DSEGURIDAD
cSEG_NOM
cSEG_PASS
nPERF_CVE
DSEGURIDAD()
Coloca_cSEG_NOM()
Coloca_cSEG_PASS()
Coloca_nPERF_CVE()
DFOLSOL
nASI_FOL
cSOL_PROY_ID
cSOL_PROY_DESC
nCAT_SEC_CVE
nCAT_STA_CVE
cSOL_ASI_IS
cSOL_ASI_NOM
cSOL_TEL
nSOL_CANTIDAD
nSOL_COMENT
DFOLSOL()
Coloca_nASI_FOL()
Coloca_cSOL_PROY_ID()
Coloca_cSOL_PROY_DESC()
Coloca_nCAT_SEC_CVE()
Coloca_nCAT_STA_CVE()
Coloca_cSOL_ASI_IS()
Coloca_cSOL_ASI_NOM()
Coloca_cSOL_TEL()
Coloca_nSOL_CANTIDAD()
Coloca_nSOL_COMENT()
TREPINVGAREQU
nUSER
cSEG_NOM
cCOM_COD_SOFT
cFAC_FOLIO
nPROV_CVE
cPROV_DESC
cCAT_INV_DESC
cCAT_STA_DESC
dFAC_FECH
dFAC_FECH_GARAN
dFAC_SEG_FECH_INI
dFAC_SEG_FECH_FIN
TREPINVGAREQU()
Coloca_nUSER()
Coloca_cSEG_NOM()
Coloca_cCOM_COD_SOFT()
Coloca_cFAC_FOLIO()
Coloca_nPROV_CVE()
Coloca_nPROV_DESC()
Coloca_cCAT_INV_DESC()
Coloca_cCAT_STA_DESC()
Coloca_dFAC_FECH()
Coloca_dFAC_FECH_GARAN()
Coloca_dFAC_SEG_FECH_INI()
Coloca_dFAC_SEG_FECH_FIN()
TREPASIGDEVEQU
nUSER
cSEG_NOM
nASI_FOL
nCAT_INV_CVE
nCAT_STA_CVE
cEQU_PER
dASI_FECH
dASI_FECH_DEV
cSOL_PROY_ID
cSOL_PROY_DESC
nCAT_SEC_CVE
cCAT_SEC_DESC
cSOL_ASI_IS
cSOL_ASI_NOM
TREPASIGDEVEQU()
Coloca_nUSER()
Coloca_cSEG_NOM()
Coloca_nASI_FOL()
Coloca_nCAT_INV_CVE()
Coloca_nCAT_STA_CVE()
Coloca_cEQU_PER()
Coloca_dASI_FECH()
Coloca_dASI_FECH_DEV()
Coloca_cSOL_PROY_ID()
Coloca_cSOL_PROY_DESC()
Coloca_nCAT_SEC_CVE()
Coloca_cCAT_SEC_DESC()
Coloca_cSOL_ASI_IS()
Coloca_cSOL_ASI_NOM()
CCOMPNOM
nCAT_INV_CVE
nCVE_CARACT
CCOMPNOM()
Coloca_nCAT_INV_CVE()
Coloca_nCVE_CARACT()
CCOMPINV
nCAT_INV_CVE
CCOMPINV()
Coloca_nCAT_INV_CVE()
CMARCEQU
nCAT_MARC_CVE
CMARCEQU()
Coloca_nCAT_MARC_CVE()

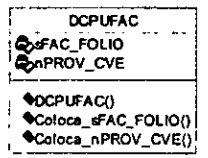

DPERSOL
nASI_FOL
nSOL_PERSONA
DPERSOL()
Coloca_nASI_FOL()
Coloca_sSOL_PERSONA()

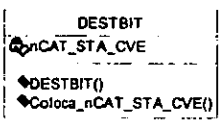

CDESCEQU
nCAT_INV_CVE
CDESCEQU()
Coloca_nCAT_INV_CVE()

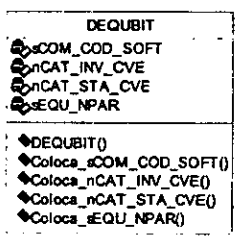

CMARCCOMP
nCAT_MARC_CVE
CMARCCOMP()
Coloca_nCAT_MARC_CVE()

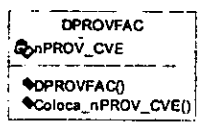

CUBIEQU
nCAT_UBI_CVE
CUBIEQU()
Coloca_nCAT_UBI_CVE()

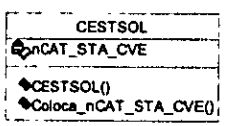

DFACEQU
sFAC_FOLIO
nPROV_CVE
DFACEQU()
Coloca_sFAC_FOLIO()
Coloca_nPROV_CVE()

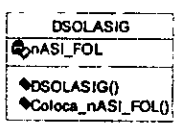

DCPUFAC
sFAC_FOLIO
nPROV_CVE
DCPUFAC()
Coloca_sFAC_FOLIO()
Coloca_nPROV_CVE()

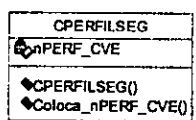

DESTBIT
nCAT_STA_CVE
DESTBIT()
Coloca_nCAT_STA_CVE()

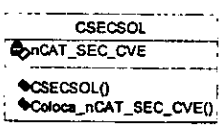

DEQUBIT
sCOM_COD_SOFT
nCAT_INV_CVE
nCAT_STA_CVE
sEQU_NPAR
DEQUBIT()
Coloca_sCOM_COD_SOFT()
Coloca_nCAT_INV_CVE()
Coloca_nCAT_STA_CVE()
Coloca_sEQU_NPAR()

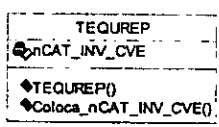

DPROVFAC
nPROV_CVE
DPROVFAC()
Coloca_nPROV_CVE()

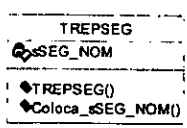

CESTSOL
nCAT_STA_CVE
CESTSOL()
Coloca_nCAT_STA_CVE()


DSOLASIG
nASI_FOL
DSOLASIG()
Coloca_nASI_FOL()


CPERFILSEG
nPERF_CVE
CPERFILSEG()
Coloca_nPERF_CVE()

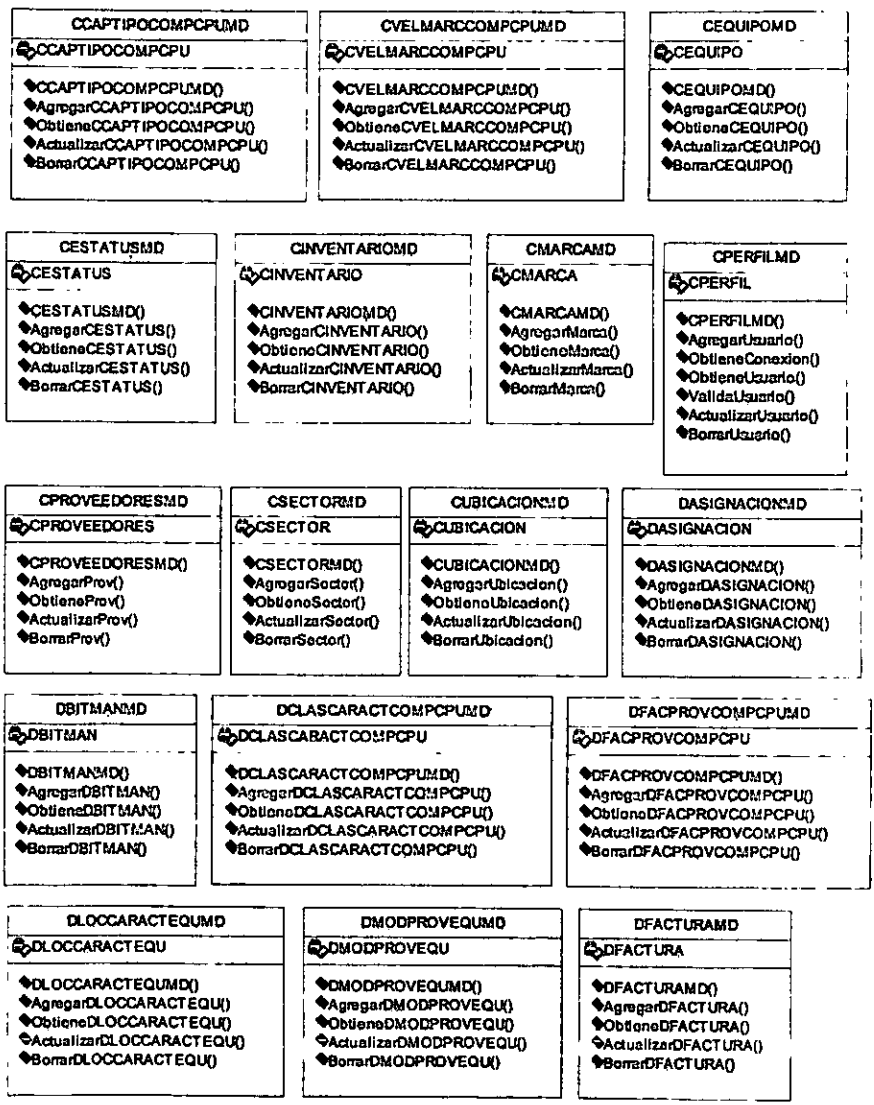

CSECSOL
nCAT_SEC_CVE
CSECSOL()
Coloca_nCAT_SEC_CVE()

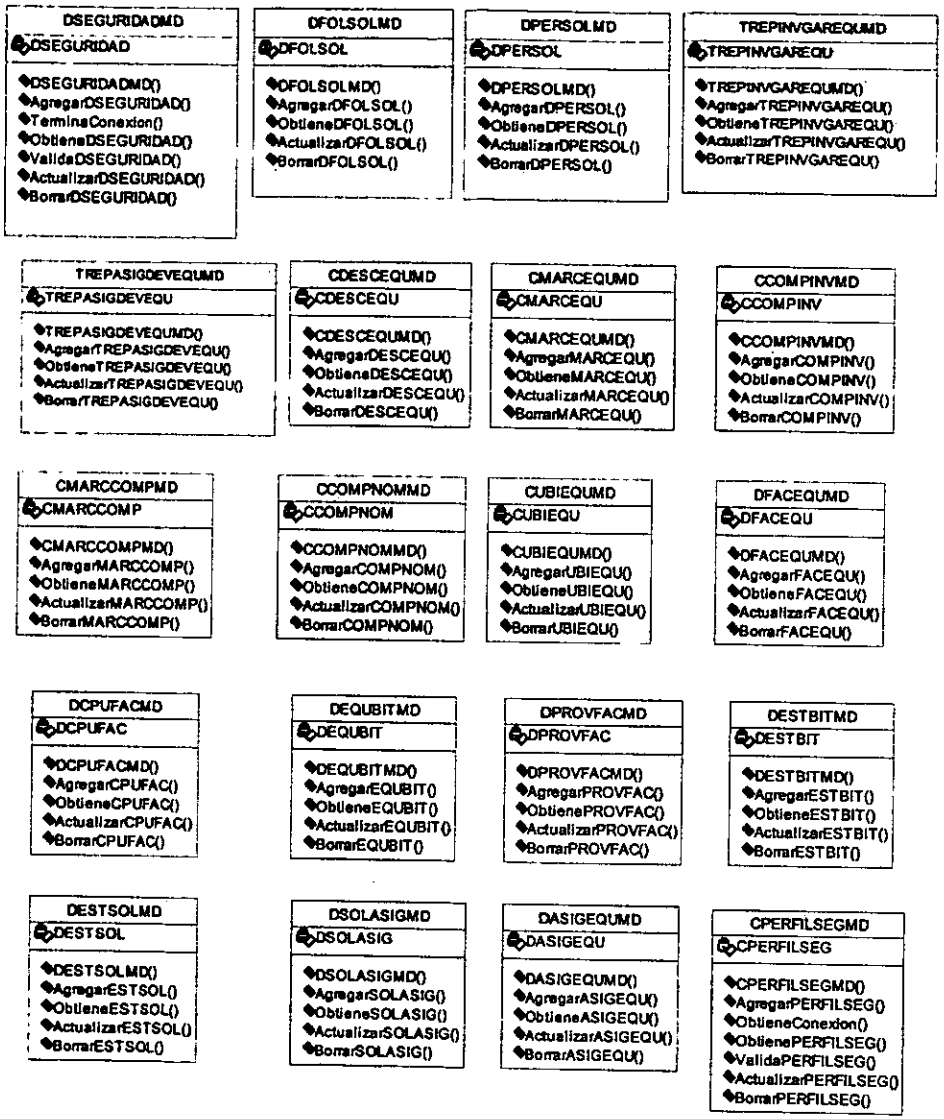

TEQUREP
nCAT_INV_CVE
TEQUREP()
Coloca_nCAT_INV_CVE()

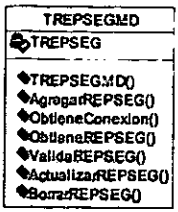

TREPSEG
sSEG_NOM
TREPSEG()
Coloca_sSEG_NOM()

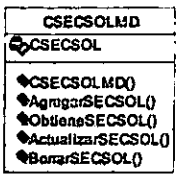
4.2.2.2.- Clases del Manejo de Datos

Se contendrán las clases ú objetos que proveen una interfaz entre las clases ú objetos del dominio del problema y una base de datos o un sistema de manejo de archivos. Enseguida se mostrarán las clases del manejo de datos referentes al sistema de información.



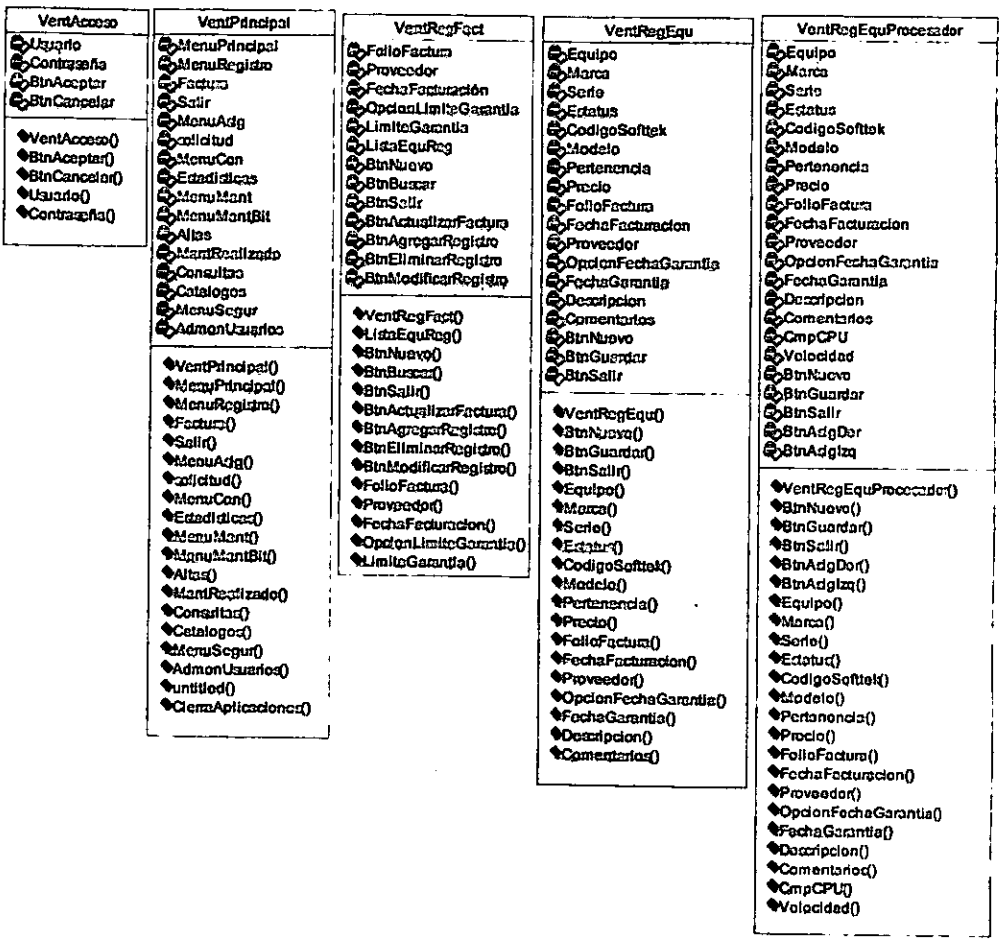

DSEGURIDADMD
DSEGURIDAD
DSEGURIDADMD()
AgregarDSEGURIDAD()
TerminaConexion()
ObtieneDSEGURIDAD()
ValidaDSEGURIDAD()
ActualizarDSEGURIDAD()
BorrarDSEGURIDAD()
DFOLSOLMD
DFOLSOL
DFOLSOLMD()
AgregarDFOLSOL()
ObtieneDFOLSOL()
ActualizarDFOLSOL()
BorrarDFOLSOL()
DPERSOLMD
DPERSOL
DPERSOLMD()
AgregarDPERSOL()
ObtieneDPERSOL()
ActualizarDPERSOL()
BorrarDPERSOL()
TREPINVGAREQUMD
TREPINVGAREQU
TREPINVGAREQUMD()
AgregarTREPINVGAREQU()
ObtieneTREPINVGAREQU()
ActualizarTREPINVGAREQU()
BorrarTREPINVGAREQU()
TREPASIGDEVEQUMD
TREPASIGDEVEQU
TREPASIGDEVEQUMD()
AgregarTREPASIGDEVEQU()
ObtieneTREPASIGDEVEQU()
ActualizarTREPASIGDEVEQU()
BorrarTREPASIGDEVEQU()
CDESCEQUMD
CDESCEQU
CDESCEQUMD()
AgregarDESCEQU()
ObtieneDESCEQU()
ActualizarDESCEQU()
BorrarDESCEQU()
CMARCEQUMD
CMARCEQU
CMARCEQUMD()
AgregarMARCEQU()
ObtieneMARCEQU()
ActualizarMARCEQU()
BorrarMARCEQU()
CCOMPINVMD
CCOMPINV
CCOMPINVMD()
AgregarCOMPINV()
ObtieneCOMPINV()
ActualizarCOMPINV()
BorrarCOMPINV()
CMARCCOMPMD
CMARCCOMP
CMARCCOMPMD()
AgregarMARCCOMP()
ObtieneMARCCOMP()
ActualizarMARCCOMP()
BorrarMARCCOMP()
CCOMPNOMMD
CCOMPNOM
CCOMPNOMMD()
AgregarCOMPNOM()
ObtieneCOMPNOM()
ActualizarCOMPNOM()
BorrarCOMPNOM()
CUBIEQUMD
CUBIEQU
CUBIEQUMD()
AgregarUBIEQU()
ObtieneUBIEQU()
ActualizarUBIEQU()
BorrarUBIEQU()
DFACEQUMD
DFACEQU
DFACEQUMD()
AgregarFACEQU()
ObtieneFACEQU()
ActualizarFACEQU()
BorrarFACEQU()
DCPUFACMD
DCPUFAC
DCPUFACMD()
AgregarCPUFAC()
ObtieneCPUFAC()
ActualizarCPUFAC()
BorrarCPUFAC()
DEQUBITMD
DEQUBIT
DEQUBITMD()
AgregarEQUBIT()
ObtieneEQUBIT()
ActualizarEQUBIT()
BorrarEQUBIT()
DPROVFACMD
DPROVFAC
DPROVFACMD()
AgregarPROVFAC()
ObtienePROVFAC()
ActualizarPROVFAC()
BorrarPROVFAC()
DESTBITMD
DESTBIT
DESTBITMD()
AgregarESTBIT()
ObtieneESTBIT()
ActualizarESTBIT()
BorrarESTBIT()
DESTSOLMD
DESTSOL
DESTSOLMD()
AgregarESTSOL()
ObtieneESTSOL()
ActualizarESTSOL()
BorrarESTSOL()
DSOLASIGMD
DSOLASIG
DSOLASIGMD()
AgregarSOLASIG()
ObtieneSOLASIG()
ActualizarSOLASIG()
BorrarSOLASIG()
DASIGEQUMD
DASIGEQU
DASIGEQUMD()
AgregarASIGEQU()
ObtieneASIGEQU()
ActualizarASIGEQU()
BorrarASIGEQU()
CPERFILSEGMD
CPERFILSEG
CPERFILSEGMD()
AgregarPERFILSEG()
ObtieneConexion()
ObtienePERFILSEG()
ValidaPERFILSEG()
ActualizarPERFILSEG()
BorrarPERFILSEG()

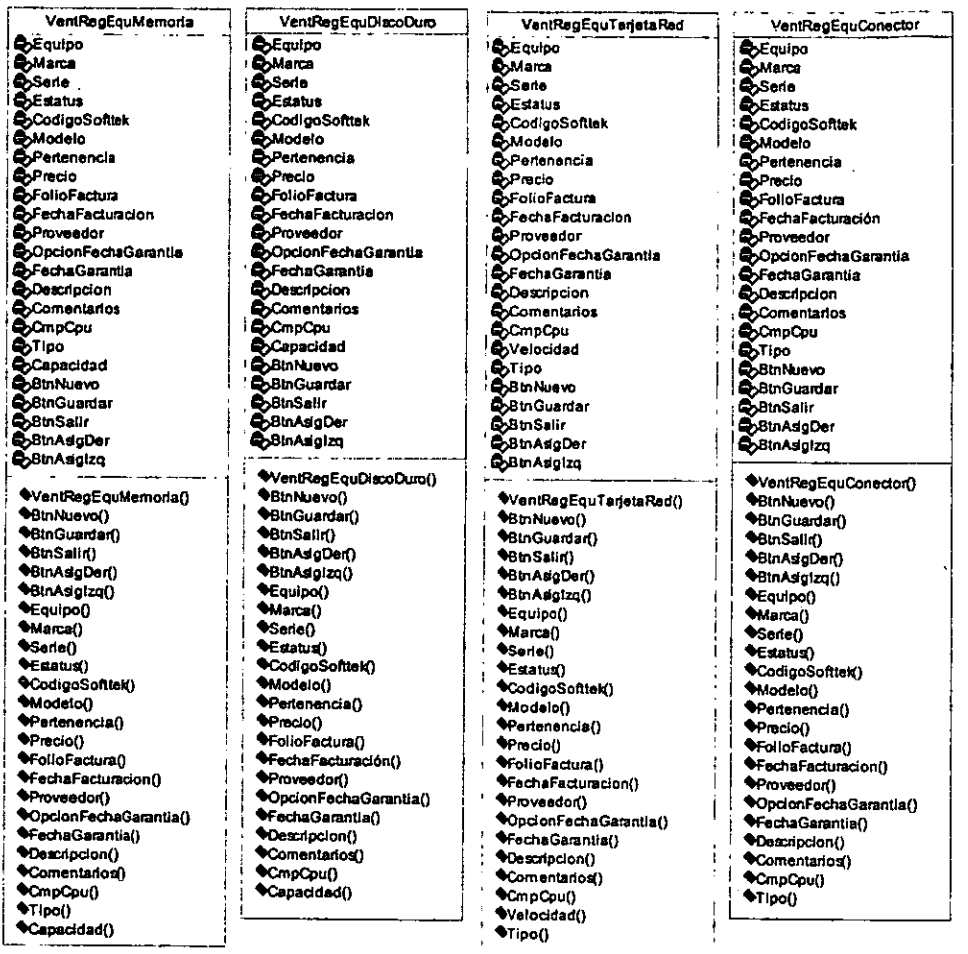


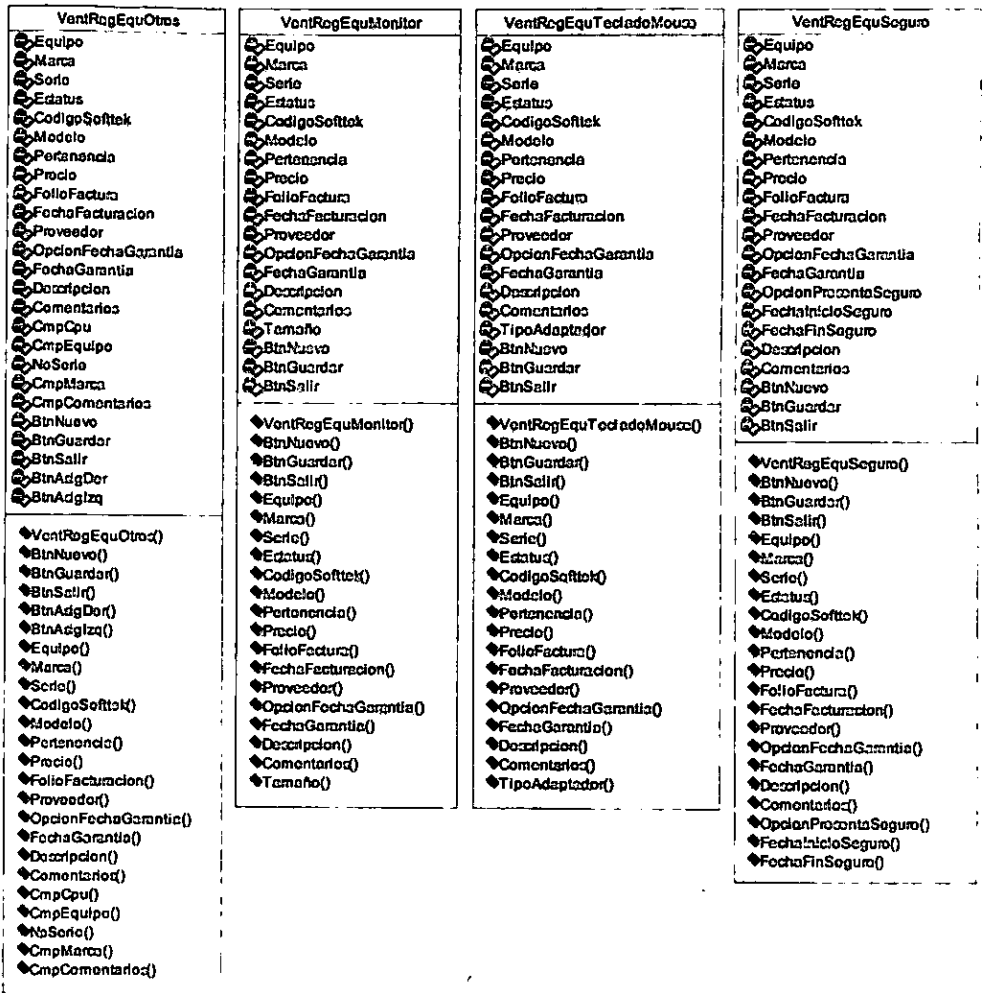


4.2.2.3. Clases de la Interfaz Humana

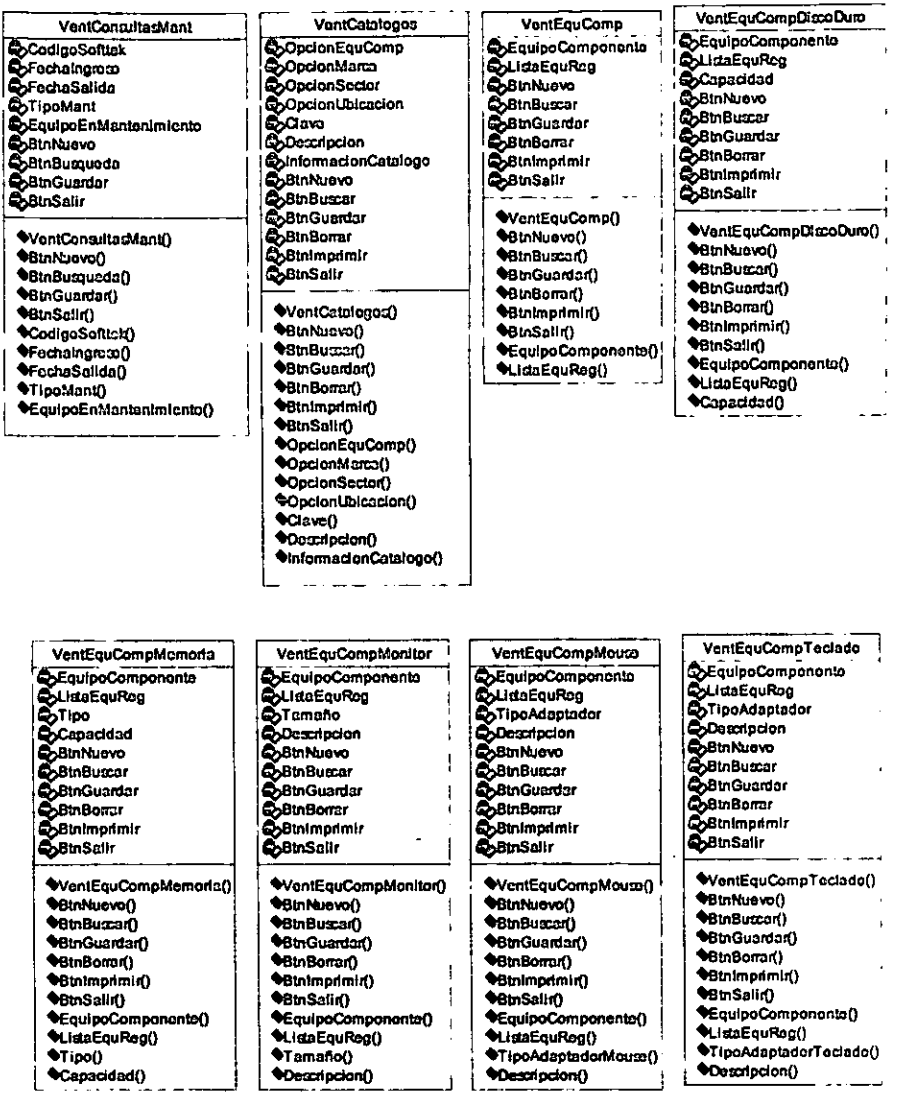
Se contendrán las clases ú objetos que proveen una interacción entre las clases ú objetos del dominio del problema y los usuarios como ventanas y reportes. A continuación se muestran las clases de la interfaz humana referentes al sistema de información.

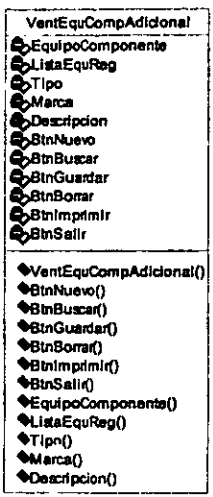


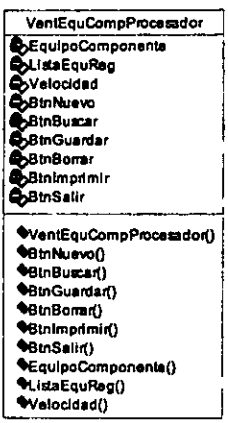

VentRegEquMemoria
Equipo
Marca
Serie
Estatus
CodigoSoftek
Modelo
Pertenencia
Precio
FolioFactura
FechaFacturacion
Proveedor
OpcionFechaGarantia
FechaGarantia
Descripcion
Comentarios
CmpCpu
Tipo
Capacidad
BtnNuevo
BtnGuardar
BtnSalir
BtnAsigDer
BtnAsigIzq
VentRegEquMemoria()
BtnNuevo()
BtnGuardar()
BtnSalir()
BtnAsigDer()
BtnAsigIzq()
Equipo()
Marca()
Serie()
Estatus()
CodigoSoftek()
Modelo()
Pertenencia()
Precio()
FolioFactura()
FechaFacturacion()
Proveedor()
OpcionFechaGarantia()
FechaGarantia()
Descripcion()
Comentarios()
CmpCpu()
Tipo()
Capacidad()
VentRegEquDiscoDuro
Equipo
Marca
Serie
Estatus
CodigoSoftek
Modelo
Pertenencia
Precio
FolioFactura
FechaFacturacion
Proveedor
OpcionFechaGarantia
FechaGarantia
Descripcion
Comentarios
CmpCpu
Capacidad
BtnNuevo
BtnGuardar
BtnSalir
BtnAsigDer
BtnAsigIzq
VentRegEquDiscoDuro()
BtnNuevo()
BtnGuardar()
BtnSalir()
BtnAsigDer()
BtnAsigIzq()
Equipo()
Marca()
Serie()
Estatus()
CodigoSoftek()
Modelo()
Pertenencia()
Precio()
FolioFactura()
FechaFacturación()
Proveedor()
OpcionFechaGarantia()
FechaGarantia()
Descripcion()
Comentarios()
CmpCpu()
Capacidad()
VentRegEquTarjetaRed
Equipo
Marca
Serie
Estatus
CodigoSoftek
Modelo
Pertenencia
Precio
FolioFactura
FechaFacturacion
Proveedor
OpcionFechaGarantia
FechaGarantia
Descripcion
Comentarios
CmpCpu
Velocidad
Tipo
BtnNuevo
BtnGuardar
BtnSalir
BtnAsigDer
BtnAsigIzq
VentRegEquTarjetaRed()
BtnNuevo()
BtnGuardar()
BtnSalir()
BtnAsigDer()
BtnAsigIzq()
Equipo()
Marca()
Serie()
Estatus()
CodigoSoftek()
Modelo()
Pertenencia()
Precio()
FolioFactura()
FechaFacturacion()
Proveedor()
OpcionFechaGarantia()
FechaGarantia()
Descripcion()
Comentarios()
CmpCpu()
Velocidad()
Tipo()
VentRegEquConector
Equipo
Marca
Serie
Estatus
CodigoSoftek
Modelo
Pertenencia
Precio
FolioFactura
FechaFacturación
Proveedor
OpcionFechaGarantia
FechaGarantia
Descripcion
Comentarios
CmpCpu
Tipo
BtnNuevo
BtnGuardar
BtnSalir
BtnAsigDer
BtnAsigIzq
VentRegEquConector()
BtnNuevo()
BtnGuardar()
BtnSalir()
BtnAsigDer()
BtnAsigIzq()
Equipo()
Marca()
Serie()
Estatus()
CodigoSoftek()
Modelo()
Pertenencia()
Precio()
FolioFactura()
FechaFacturacion()
Proveedor()
OpcionFechaGarantia()
FechaGarantia()
Descripcion()
Comentarios()
CmpCpu()
Tipo()

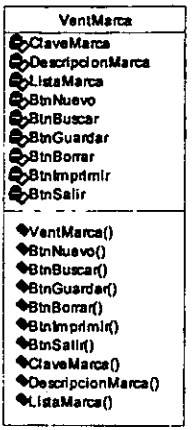

VentRegEquOtros
Equipo
Marca
Serie
Estatus
CodigoSoftek
Modelo
Pertenencia
Precio
FolioFactura
FechaFacturacion
Proveedor
OpcionFechaGarantia
FechaGarantia
Descripcion
Comentarios
CmpCpu
CmpEquipo
NoSerie
CmpMarca
CmpComentarios
BtnNuevo
BtnGuardar
BtnSalir
BtnAdgDer
BtnAdgIzq
VentRegEquOtros()
BtnNuevo()
BtnGuardar()
BtnSalir()
BtnAdgDer()
BtnAdgIzq()
Equipo()
Marca()
Serie()
CodigoSoftek()
Modelo()
Pertenencia()
Precio()
FolioFactura()
FechaFacturacion()
Proveedor()
OpcionFechaGarantia()
FechaGarantia()
Descripcion()
Comentarios()
CmpCpu()
CmpEquipo()
NoSerie()
CmpMarca()
CmpComentarios()
VentRegEquMonitor
Equipo
Marca
Serie
Estatus
CodigoSoftek
Modelo
Pertenencia
Precio
FolioFactura
FechaFacturacion
Proveedor
OpcionFechaGarantia
FechaGarantia
Descripcion
Comentarios
Tamaño
BtnNuevo
BtnGuardar
BtnSalir
VentRegEquMonitor()
BtnNuevo()
BtnGuardar()
BtnSalir()
Equipo()
Marca()
Serie()
Estatus()
CodigoSoftek()
Modelo()
Pertenencia()
Precio()
FolioFactura()
FechaFacturacion()
Proveedor()
OpcionFechaGarantia()
FechaGarantia()
Descripcion()
Comentarios()
Tamaño()
VentRegEquTecladoMouse
Equipo
Marca
Serie
Estatus
CodigoSoftek
Modelo
Pertenencia
Precio
FolioFactura
FechaFacturacion
Proveedor
OpcionFechaGarantia
FechaGarantia
Descripcion
Comentarios
TipoAdaptador
BtnNuevo
BtnGuardar
BtnSalir
VentRegEquTecladoMouse()
BtnNuevo()
BtnGuardar()
BtnSalir()
Equipo()
Marca()
Serie()
Estatus()
CodigoSoftek()
Modelo()
Pertenencia()
Precio()
FolioFactura()
FechaFacturacion()
Proveedor()
OpcionFechaGarantia()
FechaGarantia()
Descripcion()
Comentarios()
TipoAdaptador()
VentRegEquSeguro
Equipo
Marca
Serie
Estatus
CodigoSoftek
Modelo
Pertenencia
Precio
FolioFactura
FechaFacturacion
Proveedor
OpcionFechaGarantia
FechaGarantia
OpcionProcentaSeguro
FechaInicioSeguro
FechaFinSeguro
Descripcion
Comentarios
BtnNuevo
BtnGuardar
BtnSalir
VentRegEquSeguro()
BtnNuevo()
BtnGuardar()
BtnSalir()
Equipo()
Marca()
Serie()
Estatus()
CodigoSoftek()
Modelo()
Pertenencia()
Precio()
FolioFactura()
FechaFacturacion()
Proveedor()
OpcionFechaGarantia()
FechaGarantia()
Descripcion()
Comentarios()
OpcionProcentaSeguro()
FechaInicioSeguro()
FechaFinSeguro()

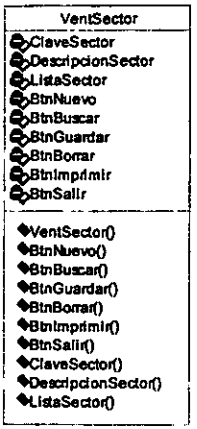

VentSolicitud
Folio
Persona
Telefono
Cantidad
I.S.
Sector
CveProyecto
Ubicacion
LiderProyecto
Proyecto
FechaAsignacion
FechaDevolucion
Observaciones
Procesador
Memoria
DiscoDuro
TarjetaRed
Otros
EquipoDisponible
EquipoAsignado
TotalEquiposDisponibles
StatusSolicitud
TotalEquiposAsignados
Comentarios
BtnNuevo
BtnBuscar
BtnGuardar
BtnImprimir
BtnSalir
BtnCPU
BtnTeclado
BtnMouse
BtnMonitor
BtnImpresora
BtnOtros
BtnAsigDer
BtnAsigIzq
VentSolicitud()
BtnNuevo()
BtnBuscar()
BtnGuardar()
BtnImprimir()
BtnSalir()
BtnCPU()
BtnTeclado()
BtnMouse()
BtnOtros()
BtnAsigDer()
BtnAsigIzq()
EquipoDisponible()
EquipoAsignado()
Folio()
Persona()
Telefono()
Cantidad()
I.S.()
Sector()
Ubicacion()
Proyecto()
CveProyecto()
LiderProyecto()
FechaAsignacion()
FechaDevolucion()
Observaciones()
TotalEquiposDisponibles()
StatusSolicitud()
TotalEquiposAsignados()
Procesador()
Memoria()
DiscoDuro()
TarjetaRed()
Otros()
VentEstadisticas
OpcionStatus
Status
OpcionSector
Sector
OpcionProcesador
Procesador
OpcionPertenencia
Pertenencia
OpcionBitacora
Bitacora
OpcionCveProy
CveProy
Proyecto
Resumen
Total
GraficaDePrueba
BtnImprimir
BtnSalir
BtnGenerarGrafico
VentanaEstadisticas()
BtnImprimir()
BtnSalir()
BtnGenerarGrafico()
OpcionStatus()
Status()
OpcionSector()
Sector()
OpcionProcesador()
Procesador()
OpcionPertenencia()
Pertenencia()
OpcionBitacora()
Bitacora()
OpcionCveProy()
CveProy()
Proyecto()
Resumen()
Total()
GraficaDePrueba()
VentAltasBitacoras
CodigoSoftek
Ubicacion
CveProy
Proyecto
FechaIngreso
FechaSalida
NumFolioAsig
TipoMant
StatusEquipo
Observaciones
BtnNuevo
BtnGuardar
BtnSalir
VentAltasBitacoras()
BtnNuevo()
BtnGuardar()
BtnSalir()
CodigoSoftek()
Ubicacion()
CveProy()
Proyecto()
FechaIngreso()
FechaSalida()
NumFolioAsig()
TipoMant()
StatusEquipo()
Observaciones()
VentMantRealizado
CodigoSoftek
FechaIngreso
FechaSalida
TipoMant
FechaRealizado
EquipoEnMantenimiento
BtnNuevo
BtnBusqueda
BtnGuardar
BtnSalir
VentMantRealizado()
BtnNuevo()
BtnBusqueda()
BtnGuardar()
BtnSalir()
CodigoSoftek()
FechaIngreso()
FechaSalida()
TipoMant()
FechaRealizado()
EquipoEnMantenimiento()

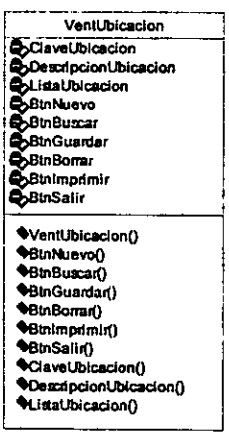

VentConsultasMant
CodigoSoftek
FechaIngreso
FechaSalida
TipoMant
EquipoEnMantenimiento
BtnNuevo
BtnBusqueda
BtnGuardar
BtnSalir
VentConsultasMant()
BtnNuevo()
BtnBusqueda()
BtnGuardar()
BtnSalir()
CodigoSoftek()
FechaIngreso()
FechaSalida()
TipoMant()
EquipoEnMantenimiento()
VentCatalogos
OpcionEquComp
OpcionMarca
OpcionSector
OpcionUbicacion
Clave
Descripcion
InformacionCatalogo
BtnNuevo
BtnBuscar
BtnGuardar
BtnBorrar
BtnImprimir
BtnSalir
VentCatalogos()
BtnNuevo()
BtnBuscar()
BtnGuardar()
BtnBorrar()
BtnImprimir()
BtnSalir()
OpcionEquComp()
OpcionMarca()
OpcionSector()
OpcionUbicacion()
Clave()
Descripcion()
InformacionCatalogo()
VentEquComp
EquipoComponente
ListaEquReg
BtnNuevo
BtnBuscar
BtnGuardar
BtnBorrar
BtnImprimir
BtnSalir
VentEquComp()
BtnNuevo()
BtnBuscar()
BtnGuardar()
BtnBorrar()
BtnImprimir()
BtnSalir()
EquipoComponente()
ListaEquReg()
VentEquCompDiscoDuro
EquipoComponente
ListaEquReg
Capacidad
BtnNuevo
BtnBuscar
BtnGuardar
BtnBorrar
BtnImprimir
BtnSalir
VentEquCompDiscoDuro()
BtnNuevo()
BtnBuscar()
BtnGuardar()
BtnBorrar()
BtnImprimir()
BtnSalir()
EquipoComponente()
ListaEquReg()
Capacidad()
VentEquCompMemoria
EquipoComponente
ListaEquReg
Tipo
Capacidad
BtnNuevo
BtnBuscar
BtnGuardar
BtnBorrar
BtnImprimir
BtnSalir
VentEquCompMemoria()
BtnNuevo()
BtnBuscar()
BtnGuardar()
BtnBorrar()
BtnImprimir()
BtnSalir()
EquipoComponente()
ListaEquReg()
Tipo()
Capacidad()
VentEquCompMonitor
EquipoComponente
ListaEquReg
Tamaño
Descripcion
BtnNuevo
BtnBuscar
BtnGuardar
BtnBorrar
BtnImprimir
BtnSalir
VentEquCompMonitor()
BtnNuevo()
BtnBuscar()
BtnGuardar()
BtnBorrar()
BtnImprimir()
BtnSalir()
EquipoComponente()
ListaEquReg()
Tamaño()
Descripcion()
VentEquCompMouse
EquipoComponente
ListaEquReg
TipoAdaptador
Descripcion
BtnNuevo
BtnBuscar
BtnGuardar
BtnBorrar
BtnImprimir
BtnSalir
VentEquCompMouse()
BtnNuevo()
BtnBuscar()
BtnGuardar()
BtnBorrar()
BtnImprimir()
BtnSalir()
EquipoComponente()
ListaEquReg()
TipoAdaptadorMouse()
Descripcion()
VentEquCompTeclado
EquipoComponente
ListaEquReg
TipoAdaptador
Descripcion
BtnNuevo
BtnBuscar
BtnGuardar
BtnBorrar
BtnImprimir
BtnSalir
VentEquCompTeclado()
BtnNuevo()
BtnBuscar()
BtnGuardar()
BtnBorrar()
BtnImprimir()
BtnSalir()
EquipoComponente()
ListaEquReg()
TipoAdaptadorTeclado()
Descripcion()

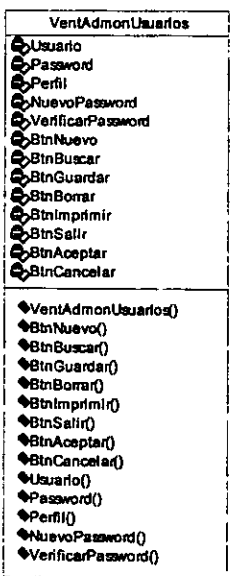

VentEquCompAdicional
EquipoComponente
ListaEquReg
Tipo
Marca
Descripcion
BtnNuevo
BtnBuscar
BtnGuardar
BtnBorrar
BtnImprimir
BtnSalir
VentEquCompAdicional()
BtnNuevo()
BtnBuscar()
BtnGuardar()
BtnBorrar()
BtnImprimir()
BtnSalir()
EquipoComponente()
ListaEquReg()
Tipo()
Marca()
Descripcion()


VentEquCompProcesador
EquipoComponente
ListaEquReg
Velocidad
BtnNuevo
BtnBuscar
BtnGuardar
BtnBorrar
BtnImprimir
BtnSalir
VentEquCompProcesador()
BtnNuevo()
BtnBuscar()
BtnGuardar()
BtnBorrar()
BtnImprimir()
BtnSalir()
EquipoComponente()
ListaEquReg()
Velocidad()


VentMarca
ClaveMarca
DescripcionMarca
ListaMarca
BtnNuevo
BtnBuscar
BtnGuardar
BtnBorrar
BtnImprimir
BtnSalir
VentMarca()
BtnNuevo()
BtnBuscar()
BtnGuardar()
BtnBorrar()
BtnImprimir()
BtnSalir()
ClaveMarca()
DescripcionMarca()
ListaMarca()


VentSector
ClaveSector
DescripcionSector
ListaSector
BtnNuevo
BtnBuscar
BtnGuardar
BtnBorrar
BtnImprimir
BtnSalir
VentSector()
BtnNuevo()
BtnBuscar()
BtnGuardar()
BtnBorrar()
BtnImprimir()
BtnSalir()
ClaveSector()
DescripcionSector()
ListaSector()


VentUbicacion
ClaveUbicacion
DescripcionUbicacion
ListaUbicacion
BtnNuevo
BtnBuscar
BtnGuardar
BtnBorrar
BtnImprimir
BtnSalir
VentUbicacion()
BtnNuevo()
BtnBuscar()
BtnGuardar()
BtnBorrar()
BtnImprimir()
BtnSalir()
ClaveUbicacion()
DescripcionUbicacion()
ListaUbicacion()


VentAdmonUsuarios
Usuario
Password
Perfil
NuevoPassword
VerificarPassword
BtnNuevo
BtnBuscar
BtnGuardar
BtnBorrar
BtnImprimir
BtnSalir
BtnAceptar
BtnCancelar
VentAdmonUsuarios()
BtnNuevo()
BtnBuscar()
BtnGuardar()
BtnBorrar()
BtnImprimir()
BtnSalir()
BtnAceptar()
BtnCancelar()
Usuario()
Password()
Perfil()
NuevoPassword()
VerificarPassword()

4.2.3. Modelo Entidad-Relación

Se utilizará el modelo entidad-relación para describir el sistema de información ya que permite de manera sencilla esquematizar las características más importantes de dicho sistema. Posteriormente se aplicará la cuarta forma normal al modelo entidad-relación y se indicará como pasarlo a un modelo equivalente orientado a objetos en notación OMT.

Para obtener un prototipo de bases de datos se hará uso de una técnica de modelado conocida como Entidad-Relación (ER). (Utilización de técnicas métodos y herramientas para el desarrollo de sistemas de información. Según IEEE Estándar 730 y 983 (1984),Documentación y Estándares, prácticas y convenciones, Butler (1995).Herramientas técnicas y metodologías, Pressman (1995) Aplicación de métodos técnicos. Stamm 1981 Metodología de ingeniería del software. ISO 9000 -3 Actividades del ciclo de vida (Diseño e implantación).

Los modelos ER son parte de las herramientas conceptuales que existen para describir datos y las relaciones que existen entre ellos. El modelo ER está basado en una percepción del mundo como si consistiese de una colección de objetos básicos (entidades), representados por una colección finita de tablas de dos dimensiones (columnas y renglones) y las relaciones que existen entre estos objetos, las cuales representan acciones o situaciones reales.

Características del modelo relacional.

1. Simplicidad: Las tablas son una forma familiar y explicables por sí mismas para representar datos. La mayoría de la gente ha utilizado datos en forma de tabla, no se requiere de un entrenamiento especial para entender o utilizar los datos que se representan en las tablas.
2. Precisión: Las tablas correctamente diseñadas mantienen un rigor matemático, dicen lo que significan y significan lo que dicen. Pueden ser implantadas y procesadas por una gran variedad de configuraciones de hardware y software.
3. Flexibilidad. Las tablas no solamente muestran la estructura de los datos sino pueden mostrar los datos también.

El modelado ER nos dice que podemos dividir el análisis en 3 fases:

a) Entidades
b) Relaciones
c) Atributos

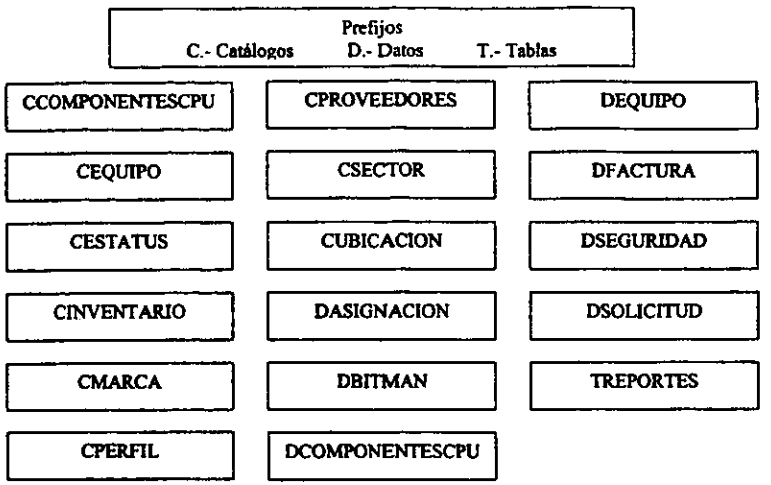
4.2.3.1.- Definición de Entidades

El modelado de Entidades sigue los siguientes pasos:

1.- Descubrir entidades
2.- Definir el alcance de la entidad
3.- Definir un índice
4.- Documentar

Mediante la utilización de técnicas métodos y herramientas para el desarrollo de sistemas de información. Según IEEE Estándar 730 y 983 (1984), Documentación y Estándares y convenciones. Butler (1995).Herramientas técnicas y metodologías Pressman (1995) Aplicación de métodos técnicos. Stamm 1981 Metodología de ingeniería del software. ISO 9000 -3 Actividades del ciclo de vida (Diseño e implantación).

A continuación se muestran las entidades definidas para la creación del sistema y también se incluye una tabla con las definiciones de cada entidad y su indización.



Modelado de Entidades

#	Entidad	Descripción	Indice
1	CCOMPONENTESCPU	Catálogo con todos los nombres de los componentes de una computadora.	nCAT_INV_CVE nCVE_CARACT
2	CEQUIPO	Catálogo con las características del equipo existente.	nCAT_INV_CVE nCVE_CARACT
3	CESTATUS	Catálogo que indica la situación actual del equipo.	nCAT_STA_CVE
4	CINVENTARIO	Catálogo para clasificar el equipo.	nCAT_INV_CVE
5	CMARCA	Catálogo para registrar la marca de los componentes.	nCAT_MARC_CVE
6	CSECTOR	Catálogo de sectores ó actividades por regiones.	nCAT_SEC_CVE
7	CPERFIL	Catálogo con la descripción del perfil del usuario.	nPERF_CVE
8	CPROVEEDORES	Catálogo con la información de proveedores de equipo.	nPROV_CVE
9	CUBICACION	Catálogo en el que se indica donde será asignado el equipo.	nCAT_UBI_CVE
10	DASIGNACION	Datos necesarios para saber en dónde y con quién fue asignado un equipo.	nASI_FOL sCOM_COD_SOFT
11	DBITMAN	Datos referentes a la bitácora de mantenimiento.	sCOM_COD_SOFT nNUM_MTTO
12	DCOMPONENTESCPU	Datos referentes a la clasificación y descripción de los componentes de una computadora.	sCMP_CPU_NPAR nCAT_INV_CVE nCVE_CARACT
13	DEQUIPO	Datos sobre la clasificación y descripción del equipo.	sCOM_COD_SOFT
14	DFACTURA	Datos generales de la factura como fecha, folio, proveedor.	nFAC_FOLIO nPROV_CVE
15	DSEGURIDAD	Datos del personal autorizado para acceder al sistema SIESM.	sSEG_NOM
16	DSOLICITUD	Datos sobre el folio de la solicitud de equipo.	nASI_FOL
17	TREPORTES	Tabla referente al movimiento y características del equipo.	

Definición e indización de Entidades

4.2.3.2.- Definición de Relaciones

Las relaciones se obtienen de los siguientes pasos:

1. Descubrir relaciones.
2. Definir el alcance de la relación.
3. Definir el tipo de relación.
4. Documentar en el diagrama ER.
5. Documentar en tablas.

Por medio de técnicas métodos y herramientas para el desarrollo de sistemas de información. Según IEEE Estándar 730 y 983 (1984), Documentación y Estándares y convenciones. Butler (1995).Herramientas técnicas y metodologías Pressman (1995) Aplicación de métodos técnicos. Stamm 1981 Metodología de ingeniería del software. ISO 9000 –3 Actividades del ciclo de vida (Diseño e implantación). Las relaciones de acuerdo al número de elementos que se involucran de ambos lados de las entidades que participan se clasifican en tres tipos: Uno a Uno (1-1), Uno a Muchos (1-M) y Muchos a Muchos (M-M).

En la siguiente tabla se muestra la lista de relaciones definidas para el sistema y posteriormente se muestra el diagrama Entidad-Relación que se obtiene después de completar esta definición al igual que una descripción del mismo.

Relación	Nombre	Descripción	Entidades
R1	CDESCEQU	Presenta descripción del equipo.	E2-E4
R2	CMARCEQU	Clasifica equipo por marca.	E2-E5
R3	CCOMPINV	Clasifica equipo por componentes.	E4-E1
R4	CMARCCOMP	Clasifica y describe el equipo por marca de los componentes.	E5-E13
R5	CCOMPNOM	Presenta clasificación y descripción de los componentes de una computadora por nombre.	E1-E12
R6	CUBIEQU	Asignación de equipo.	E13-E9
R7	DFACEQU	Presenta descripción con datos generales de la factura de equipo.	E13-E14
R8	DCPUFAC	Presenta datos generales de clasificación, descripción y factura de los componentes de una computadora.	E12-E14
R9	DEQUBIT	Clasifica y describe el equipo a partir de la bitácora de mantenimiento.	E11-E13
R10	CPROVFAC	Presenta información de proveedores de equipo según datos de factura.	E14-E8
R11	DESTBIT	Presenta situación actual del equipo según la bitácora de mantenimiento.	E11-E3
R12	DASIGEQU	Clasifica y describe el equipo según dónde y con quién fue asignado.	E10-E13
R13	DSOLASIG	Presenta folio de asignación de equipo según donde y con quien fue asignado.	E10-E16
R14	CESTSOL	Presenta la situación actual del equipo según su asignación en el folio.	E16-E3
R15	CSECSOL	Clasifica sectores ó actividades por regiones según la asignación en el folio.	E16-E6
R16	CPERFILSEG	Presenta el perfil del usuario según su nivel de autorización.	E15-E7
R17	TEQUREP	Presenta el movimiento con las características del equipo existente.	E17-E2
R18	TREPSEG	Clasifica datos del personal autorizado para ver movimiento y características de equipo.	E15-E17

Relaciones definidas para el sistema

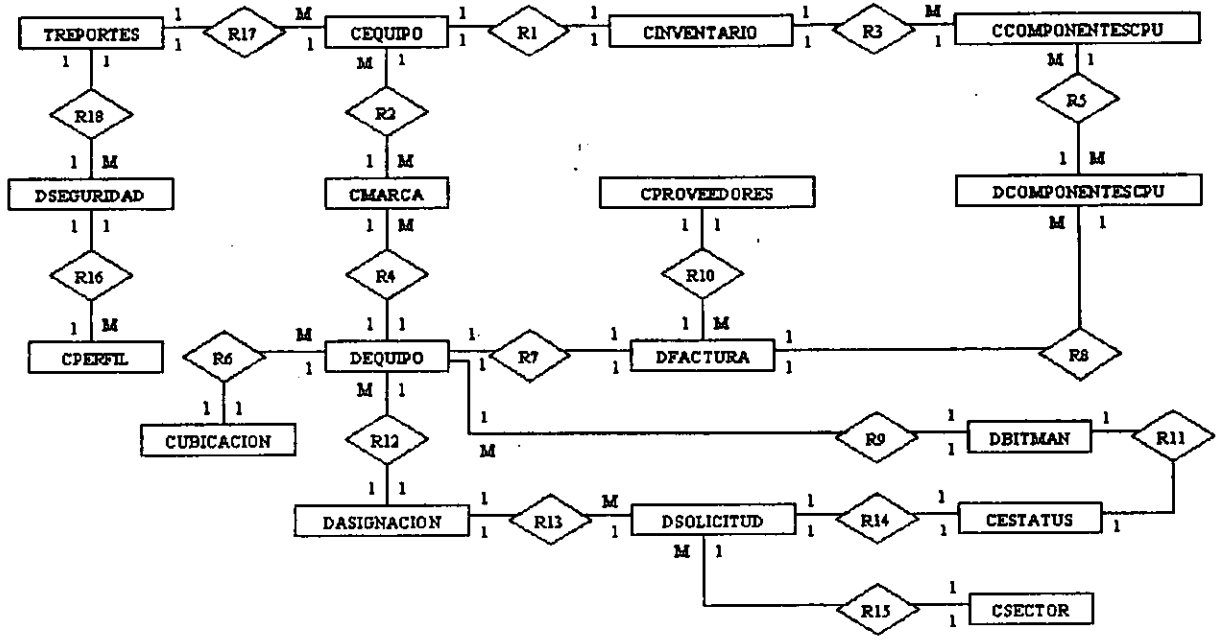


Diagrama Entidad-Relación

Descripción del Diagrama Entidad-Relación

A continuación se describirá el diagrama entidad-relación anteriormente desarrollado.

La relación CPERFIL se asocia con DSEGURIDAD por medio del perfil del usuario según su nivel de autorización (R16); la relación DSEGURIDAD se asocia con TREPORTES por medio del personal autorizado para ver movimiento y características de equipo (R18); la relación TREPORTES se asocia con CEQUIPO por el movimiento con las características del equipo existente (R17); la relación CEQUIPO se asocia con CINVENTARIO por medio de la descripción del equipo (R1); la relación CEQUIPO se asocia con CMARCA por medio de la clasificación del equipo por marca (R2); la relación CINVENTARIO se asocia con CCOMPONENTESCPU por medio de la clasificación del equipo por componentes (R3); la relación CMARCA se asocia con DEQUIPO por medio de la clasificación y descripción del equipo por marca de los componentes (R4); la relación CCOMPONENTESCPU se asocia con DCOMPONENTESCPU por medio de la clasificación y descripción de los componentes de una computadora por nombre (R5); la relación DCOMPONENTESCPU se asocia con DFACTURA al presentar datos generales de clasificación, descripción y factura de los componentes de una computadora (R8); la relación CPROVEEDORES se asocia con DFACTURA por medio de la información de proveedores de equipo según datos de factura (R10); la relación CUBICACION se asocia con DEQUIPO por medio de la asignación de equipo (R6); la relación DFACTURA se asocia con DEQUIPO por la presentación de la descripción con datos generales de la factura de equipo (R7); la relación DEQUIPO se asocia con DBITMAN por medio de la clasificación y descripción del equipo a partir de la bitácora de mantenimiento (R9); la relación DASIGNACION se asocia con DEQUIPO por medio de la clasificación y descripción del equipo según dónde y con quién fue asignado (R12); la relación CSECTOR se asocia con DSOLICITUD al clasificar sectores ó actividades por regiones según la asignación en el folio (R15); la relación DBITMAN se asocia con CESTATUS por medio de la situación actual del equipo según la bitácora de mantenimiento (R11); la relación CESTATUS se asocia a DSOLICITUD por medio de la situación actual del equipo según su asignación en el folio (R14); la relación DSOLICITUD se asocia a DASIGNACION por medio del folio de asignación de equipo según donde y con quien fue asignado (R13).

4.2.3.3. Definición de Atributos

La definición de atributos se obtiene con los siguientes pasos:

1. Descubrir atributos.
2. Definir el alcance del atributo.
3. Documentar el atributo en la entidad.

Utilizando técnicas métodos y herramientas para el desarrollo de sistemas de información. Según IEEE Estándar 730 y 983 (1984) Documentación, Butler (1995).Herramientas técnicas y metodologías Stamm 1981 Metodología de ingeniería del software. ISO 9000 -3 Actividades del ciclo de vida (Diseño e implantación)

Para colocar los atributos de manera correcta en una entidad se utiliza el método de normalización, el cual se aplica sobre las tablas (también llamadas relaciones) del sistema para lograr el correcto funcionamiento del mismo. Se procederá a mostrar cuales fueron las relaciones resultantes junto con la descripción de sus atributos, posteriormente se van a normalizar.

Relación: CCOMPONENTESCPU

Esta relación contiene el catálogo con las características de los componentes de una computadora.

Atributos:

Descripción	Campo	Tipo	Índice
Clave de inventario según catálogo	nCAT_INV_CVE	Numeric(5)	Llave primaria
Clave de características	nCVE_CARACT	Numeric(5)	Llave foránea
Tipo de componente según catálogo	sCAT_COMP_TIPO	VarChar(35)	
Velocidad de componente según catálogo	sCAT_COMP_VEL	VarChar(10)	
Capacidad de componente según catálogo	sCAT_COMP_CAP	VarChar(15)	
Descripción de componente según catálogo	sCAT_COMP_DESC	VarChar(30)	
Marca y dirección de componente según catálogo	sCAT_COMP_MCADDRESS	VarChar(20)	

Relación: CEQUIPO

Esta relación contiene el catálogo con las características del equipo existente.

Atributos:

Descripción	Campo	Tipo	Índice
Clave de inventario según catálogo	nCAT_INV_CVE	Numeric(5)	Llave primaria
Clave de características	nCVE_CARACT	Numeric(5)	Llave primaria
Clave de marca según catálogo	nCAT_MARC_CVE	Numeric(5)	
Tipo de equipo según catálogo	sCAT_EQP_TIPO	VarChar(20)	
Descripción de equipo según catálogo	sCAT_EQP_DESC	VarChar(20)	

Relación: CESTATUS

Esta relación contiene el catálogo que indica la situación actual del equipo.

Atributos:

Descripción	Campo	Tipo	Índice
Clave del estatus según catálogo	nCAT_STA_CVE	Numeric(5)	Llave foránea
Descripción del estatus según catálogo	sCAT_STA_DESC	VarChar(50)	

Relación: CINVENTARIO

Esta relación contiene el catálogo para clasificar el equipo.

Atributos:

Descripción	Campo	Tipo	Índice
Clave de inventario según catálogo	nCAT_INV_CVE	Numeric(5)	Llave foránea
Descripción de inventario según catálogo	sCAT_INV_DESC	VarChar(50)	

Relación: CMARCA

Esta relación contiene el catálogo para registrar la marca de los componentes.

Atributos:

Descripción	Campo	Tipo	Índice
Clave de marca según catálogo	nCAT_MARC_CVE	Numeric(5)	Llave foránea
Descripción de marca según catálogo	sCAT_MARC_DESC	VarChar(50)	

Relación: CPERFIL

Esta relación contiene el catálogo con la descripción del perfil del usuario.

Atributos:

Descripción	Campo	Tipo	Índice
Clave del perfil	nPERF_CVE	Numeric(5)	Llave foránea
Descripción del perfil	sPERF_DESC	VarChar(50)	

Relación: CPROVEEDORES

Esta relación contiene el catálogo con la información de proveedores de equipo.

Atributos:

Descripción	Campo	Tipo	Índice
Clave de proveedores	nPROV_CVE	Numeric(5)	Llave foránea
Descripción de proveedores	sPROV_DESC	VarChar(50)	

Relación: CSECTOR

Esta relación contiene el catálogo de sectores ó actividades por regiones.

Atributos:

Descripción	Campo	Tipo	Índice
Clave de sector según catálogo	nCAT_SEC_CVE	Numeric(5)	Llave foránea
Descripción de sector según catálogo	sCAT_SEC_DESC	VarChar(50)	

Relación: CUBICACION

Esta relación contiene el catálogo en el que se indica donde será asignado el equipo.

Atributos:

Descripción	Campo	Tipo	Índice
Clave de la ubicación según catálogo	nCAT_UBI_CVE	Numeric(5)	Llave foránea
Descripción de la ubicación según catálogo	sCAT_UBI_DESC	VarChar(50)	

Relación: DASIGNACION

En esta relación se dan los datos necesarios para saber en dónde y con quién fue asignado un equipo.

Atributos:

Descripción	Campo	Tipo	Índice
Número de folio de asignación	nASI_FOL	Integer(4)	Llave primaria
Código de serie Softtek	sCOM_COD_SOFT	VarChar(8)	Llave primaria
Fecha de asignación	dASI_FECH	DateTime(8)	
Fecha de devolución	dASI_FECH_DEV	DateTime(8)	

Relación: DBITMAN

En esta relación se dan los datos referentes a la bitácora de mantenimiento.

Atributos:

Descripción	Campo	Tipo	Índice
Código de serie Softtek	sCOM_COD_SOFT	VarChar(8)	Llave primaria
Número de mantenimiento	sNUM_MTTO	Integer(4)	Llave primaria
Clave de inventario según catálogo	nCAT_INV_CVE	Numeric(5)	
Fecha de ingreso en la bitácora	dBIT_FECH_ING	DateTime(8)	
Fecha de realización del mantenimiento	dBIT_FECH_FIN	DateTime(8)	
Comentarios en la bitácora	sBIT_COMENT	VarChar(100)	
Clave del estatus según catálogo	nCAT_STA_CVE	Numeric(5)	
Código de partes de un equipo	sEQU_NPAR	VarChar(8)	

Relación: DCOMPONENTESCPU

Esta relación contiene datos referentes a la clasificación y descripción de los componentes de una computadora.

Atributos:

Descripción	Campo	Tipo	Índice
Código de partes de una computadora	sCMP_CPU_NPAR	VarChar(15)	Llave primaria
Clave de inventario según catálogo	nCAT_INV_CVE	Numeric(5)	Llave primaria
Clave de características	nCVE_CARACT	Numeric(5)	Llave primaria
Clave de catálogo de status de equipo	nCAT_STA_CVE	Numeric(5)	
Código de serie Softtek	sCOM_COD_SOFT	VarChar(8)	
Código de partes de un equipo	sEQU_NPAR	VarChar(8)	
Número de folio de la factura	sFAC_FOLIO	VarChar(15)	
Clave del proveedor	nPROV_CVE	Numeric(5)	

Relación: DEQUIPO

Esta relación contiene datos sobre la clasificación y descripción del equipo.

Atributos:

Descripción	Campo	Tipo	Índice
Código de serie Softtek	sCOM_COD_SOFT	VarChar(8)	Llave primaria
Clave de inventario según catálogo	nCAT_INV_CVE	Numeric(5)	
Clave de características	nCVE_CARACT	Numeric(5)	
Número de folio de asignación	nASI_FOL	Integer(4)	
Clave de la ubicación según catálogo	nCAT_UBI_CVE	Numeric(5)	
Clave de marca según catálogo	nCAT_MARC_CVE	Numeric(5)	
Persona	sEQU_PER	Char(3)	
Modelo del equipo	sEQU_MOD	VarChar(20)	
Número de partes del equipo	sEQU_NPAR	VarChar(15)	
Clave del estatus según catálogo	nCAT_STA_CVE	Numeric(5)	
Número de folio de la factura	sFAC_FOLIO	VarChar(15)	
Clave de proveedores	nPROV_CVE	Numeric(5)	
Comentarios referentes al equipo	sEQU_COMENT	VarChar(60)	
Tipo de componente según catálogo	sCAT_COMP_TIPO	VarChar(35)	

Relación: DFACTURA

Esta relación contiene datos generales de la factura como fecha, folio, proveedor.

Atributos:

Descripción	Campo	Tipo	Índice
Número de folio de la factura	sFAC_FOLIO	VarChar(15)	Llave foránea
Clave del proveedor	nPROV_CVE	Numeric(5)	Llave foránea
Precio del equipo	nEQU_PREC	Decimal(9)	
Fecha de la factura	dFAC_FECH	DateTime(8)	
Fecha de garantía	dFAC_FECH_GARAN	DateTime(8)	
Fecha de inicio de seguro	dFAC_SEG_FECH_INI	DateTime(8)	
Fecha de fin de seguro	dFAC_SEG_FECH_FIN	DateTime(8)	

Relación: DSEGURIDAD

En esta relación se dan los datos del personal autorizado para acceder al sistema SIESM.

Atributos:

Descripción	Campo	Tipo	Índice
Nombre	sSEG_NOM	VarChar(20)	Llave primaria
Pasaporte	sSEG_PASS	VarChar(14)	
Clave del perfil	nPERF_CVE	Numeric(5)	

Relación: DSOLICITUD

Esta relación contiene datos sobre el folio de la solicitud de equipo.

Atributos:

Descripción	Campo	Tipo	Índice
Número de folio de asignación	nASI_FOL	Integer(4)	Llave foránea
Identificador del proyecto solicitante	sSOL_PROY_ID	VarChar(7)	
Descripción del proyecto solicitante	sSOL_PROY_DESC	VarChar(50)	
Clave del sector según catálogo	nCAT_SEC_CVE	Numeric(5)	
Clave del estatus según catálogo	nCAT_STA_CVE	Numeric(5)	
Asignación de la solicitud	sSOL_ASI_IS	VarChar(4)	
Nombre de la asignación solicitada	sSOL_ASI_NOM	VarChar(50)	
Teléfono del solicitante	sSOL_TEL	VarChar(20)	
Cantidad solicitada	nSOL_CANTIDAD	Integer(4)	
Persona que solicita	nSOL_PERSONA	VarChar(50)	
Comentarios de la solicitud	nSOL_COMENT	VarChar(60)	

Relación: TREPORTES

Relación referente al movimiento y características del equipo.

Atributos:

Descripción	Campo	Tipo	Índice
Número de usuario	nUSER	Integer(4)	Llave primaria
Nombre	sSEG_NOM	VarChar(20)	Llave foránea
Código de serie Softtek	sCOM_COD_SOFT	VarChar(8)	
Número de folio de la factura	sFAC_FOLIO	VarChar(15)	
Clave del proveedor	nPROV_CVE	Numeric(5)	
Descripción del proveedor	sPROV_DESC	VarChar(50)	
Número de folio de asignación	nASI_FOL	Integer(4)	
Clave de inventario según catálogo	nCAT_INV_CVE	Numeric(5)	
Clave del estatus según catálogo	nCAT_STA_CVE	Numeric(5)	
Persona	sEQU_PER	Char(3)	
Descripción de inventario según catálogo	sCAT_INV_DESC	VarChar(50)	
Descripción del estatus según catálogo	sCAT_STA_DESC	VarChar(50)	
Fecha de la factura	dFAC_FECH	DateTime(8)	
Fecha de garantía	dFAC_FECH_GARAN	DateTime(8)	
Fecha de inicio de seguro	dFAC_SEG_FECH_INI	DateTime(8)	
Fecha de fin de seguro	dFAC_SEG_FECH_FIN	DateTime(8)	
Fecha de asignación	dASI_FECH	DateTime(8)	
Fecha de devolución	dASI_FECH_DEV	DateTime(8)	
Identificador del proyecto solicitante	sSOL_PROY_ID	VarChar(7)	
Descripción del proyecto solicitante	sSOL_PROY_DESC	VarChar(50)	
Clave del sector según catálogo	nCAT_SEC_CVE	Numeric(5)	
Descripción de sector según catálogo	sCAT_SEC_DESC	VarChar(50)	
Asignación de la solicitud	sSOL_ASI_IS	VarChar(4)	
Nombre de la asignación solicitada	sSOL_ASI_NOM	VarChar(50)	

4.2.3.4. Normalización

La normalización tiene como fundamento el concepto de formas normales. Se dice que una relación está en una determinada forma normal si satisface un cierto conjunto de restricciones. La normalización nos permite convertir una relación determinada en otras más deseables por medio de proyecciones de la misma, con lo cual se conserva la integridad de la base de datos a través de las distintas operaciones que se ejecuten sobre la misma. (Estándares especificados que definen que definen un conjunto de criterios o procedimientos de desarrollo que guíen la forma correcta para la elaboración del software. Calidad. Según IEEE Estándar 730 y 983 (1984), Documentación y Estándares y convenciones. Butler (1995).Herramientas técnicas y metodologías Pressman (1995) Aplicación de métodos técnicos. Stamm 1981 Metodología de ingeniería del software. ISO 9000 –3 Actividades del ciclo de vida (Diseño e implantación)

A continuación se normalizarán las relaciones del sistema de información, para lo cual se usarán las características de la cuarta forma normal (4NF), ya que por tratarse de un inventario existen dependencias multivaluadas y la cuarta forma normal es la que mejor resuelve dicha situación al buscar aplanar el modelo de datos. En las proyecciones de una relación se indicará de donde provienen con respecto a la relación no normalizada, dicha indicación estará en la descripción de cada relación, en el caso de las relaciones que ya se encuentren normalizadas no se indicará una relación origen.

Relación: CCAPTIPOCOMPCPU

Esta relación contiene el catálogo con características sobre capacidad y tipo de componentes en una computadora. Esta relación es una proyección de la relación CCOMPONENTESCPU.

Atributos:

Descripción	Campo	Tipo	Índice
Clave de inventario según catálogo	nCAT_INV_CVE	Numeric(5)	Llave primaria
Clave de características	nCVE_CARACT	Numeric(5)	Llave foránea
Tipo de componente según catálogo	sCAT_COMP_TIPO	VarChar(35)	
Capacidad de componente según catálogo	sCAT_COMP_CAP	VarChar(15)	
Descripción de componente según catálogo	sCAT_COMP_DESC	VarChar(30)	

Relación: CVELMARCCOMPCPU

Esta relación contiene el catálogo con características sobre velocidad y marca de componentes en una computadora. Esta relación es una proyección de la relación CCOMPONENTESCPU.

Atributos:

Descripción	Campo	Tipo	Índice
Clave de inventario según catálogo	nCAT_INV_CVE	Numeric(5)	Llave primaria
Clave de características	nCVE_CARACT	Numeric(5)	Llave foránea
Velocidad de componente según catálogo	sCAT_COMP_VEL	VarChar(10)	
Marca y dirección de componente según catálogo	sCAT_COMP_MCADDRESS	VarChar(20)	

Relación: CEQUIPO

Esta relación contiene el catálogo con las características del equipo existente.

Atributos:

Descripción	Campo	Tipo	Índice
Clave de inventario según catálogo	nCAT_INV_CVE	Numeric(5)	Llave primaria
Clave de características	nCVE_CARACT	Numeric(5)	Llave primaria
Clave de marca según catálogo	nCAT_MARC_CVE	Numeric(5)	
Tipo de equipo según catálogo	sCAT_EQP_TIPO	VarChar(20)	
Descripción de equipo según catálogo	sCAT_EQP_DESC	VarChar(20)	

Relación: CESTATUS

Esta relación contiene el catálogo que indica la situación actual del equipo.

Atributos:

Descripción	Campo	Tipo	Índice
Clave del estatus según catálogo	nCAT_STA_CVE	Numeric(5)	Llave foránea
Descripción del estatus según catálogo	sCAT_STA_DESC	VarChar(50)	

Relación: CINVENTARIO

Esta relación contiene el catálogo para clasificar el equipo.

Atributos:

Descripción	Campo	Tipo	Índice
Clave de inventario según catálogo	nCAT_INV_CVE	Numeric(5)	Llave foránea
Descripción de inventario según catálogo	sCAT_INV_DESC	VarChar(50)	

Relación: CMARCA

Esta relación contiene el catálogo para registrar la marca de los componentes.

Atributos:

Descripción	Campo	Tipo	Índice
Clave de marca según catálogo	nCAT_MARC_CVE	Numeric(5)	Llave foránea
Descripción de marca según catálogo	sCAT_MARC_DESC	VarChar(50)	

Relación: CPERFIL

Esta relación contiene el catálogo con la descripción del perfil del usuario.

Atributos:

Descripción	Campo	Tipo	Indice
Clave del perfil	nPERF_CVE	Numeric(5)	Llave foránea
Descripción del perfil	sPERF_DESC	VarChar(50)	

Relación: CPROVEEDORES

Esta relación contiene el catálogo con la información de proveedores de equipo.

Atributos:

Descripción	Campo	Tipo	Índice
Clave de proveedores	nPROV_CVE	Numeric(5)	Llave foránea
Descripción de proveedores	sPROV_DESC	VarChar(50)	

Relación: CSECTOR

Esta relación contiene el catálogo de sectores ó actividades por regiones.

Atributos:

Descripción	Campo	Tipo	Índice
Clave de sector según catálogo	nCAT_SEC_CVE	Numeric(5)	Llave foránea
Descripción de sector según catálogo	sCAT_SEC_DESC	VarChar(50)	

Relación: CUBICACION

Esta relación contiene el catálogo en el que se indica donde será asignado el equipo.

Atributos:

Descripción	Campo	Tipo	Índice
Clave de la ubicación según catálogo	nCAT_UBI_CVE	Numeric(5)	Llave foránea
Descripción de la ubicación según catálogo	sCAT_UBI_DESC	VarChar(50)	

Relación: DASIGNACION

En esta relación se dan los datos necesarios para saber en dónde y con quién fue asignado un equipo.

Atributos:

Descripción	Campo	Tipo	Índice
Número de folio de asignación	nASI_FOL	Integer(4)	Llave primaria
Código de serie Softtek	sCOM_COD_SOFT	VarChar(8)	Llave primaria
Fecha de asignación	dASI_FECH	DateTime(8)	
Fecha de devolución	dASI_FECH_DEV	DateTime(8)	

Relación: DBITMAN

En esta relación se dan los datos referentes a la bitácora de mantenimiento.

Atributos:

Descripción	Campo	Tipo	Índice
Código de serie Softtek	sCOM_COD_SOFT	VarChar(8)	Llave primaria
Número de mantenimiento	sNUM_MTTO	Integer(4)	Llave primaria
Clave de inventario según catálogo	nCAT_INV_CVE	Numeric(5)	
Fecha de ingreso en la bitácora	dBIT_FECH_ING	DateTime(8)	
Fecha de realización del mant.	dBIT_FECH_FIN	DateTime(8)	
Comentarios en la bitácora	sBIT_COMENT	VarChar(100)	
Clave del estatus según catálogo	nCAT_STA_CVE	Numeric(5)	
Código de partes de un equipo	sEQU_NPAR	VarChar(8)	

Relación: DCLASCARACTCOMPCPU

Esta relación contiene datos referentes a la clasificación y características de los componentes en una computadora. Esta relación es una proyección de la relación DCOMPONENTESCPU.

Atributos:

Descripción	Campo	Tipo	Índice
Código de partes de una computadora	sCMP_CPU_NPAR	VarChar(15)	Llave primaria
Clave de inventario según catálogo	nCAT_INV_CVE	Numeric(5)	Llave primaria
Clave de características	nCVE_CARACT	Numeric(5)	Llave primaria
Clave de catálogo de status de equipo	nCAT_STA_CVE	Numeric(5)	
Código de serie Softtek	sCOM_COD_SOFT	VarChar(8)	
Código de partes de un equipo	sEQU_NPAR	VarChar(8)	

Relación: DFACPROVCOMPCPU

Esta relación contiene datos referentes a la factura y proveedor de los componentes de una computadora. Esta relación es una proyección de la relación DCOMPONENTESCPU.

Atributos:

Descripción	Campo	Tipo	Índice
Código de partes de una computadora	sCMP_CPU_NPAR	VarChar(15)	Llave primaria
Clave de inventario según catálogo	nCAT_INV_CVE	Numeric(5)	Llave primaria
Clave de características	nCVE_CARACT	Numeric(5)	Llave primaria
Número de folio de la factura	sFAC_FOLIO	VarChar(15)	
Clave del proveedor	nPROV_CVE	Numeric(5)	

Relación: DLOCCARACTEQU

Esta relación contiene datos sobre la localización y características del equipo. Esta relación es una proyección de la relación DEQUIPO.

Atributos:

Descripción	Campo	Tipo	Índice
Código de serie Softtek	sCOM_COD_SOFT	VarChar(8)	Llave primaria
Clave de inventario según catálogo	nCAT_INV_CVE	Numeric(5)	
Clave de características	nCVE_CARACT	Numeric(5)	
Número de folio de asignación	nASI_FOL	Integer(4)	
Clave de la ubicación según catálogo	nCAT_UBI_CVE	Numeric(5)	
Clave de marca según catálogo	nCAT_MARC_CVE	Numeric(5)	
Persona	sEQU_PER	Char(3)	
Número de partes del equipo	sEQU_NPAR	VarChar(15)	
Clave del estatus según catálogo	nCAT_STA_CVE	Numeric(5)	
Número de folio de la factura	sFAC_FOLIO	VarChar(15)	
Comentarios referentes al equipo	sEQU_COMENT	VarChar(60)	
Tipo de componente según catálogo	sCAT_COMP_TIPO	VarChar(35)	

Relación: DMODPROVEQU

Esta relación contiene datos sobre el modelo y proveedores del equipo. Esta relación es una proyección de la relación DEQUIPO.

Atributos:

Descripción	Campo	Tipo	Índice
Código de serie Softtek	sCOM_COD_SOFT	VarChar(8)	Llave primaria
Modelo del equipo	sEQU_MOD	VarChar(20)	
Clave de proveedores	nPROV_CVE	Numeric(5)	

Relación: DFACTURA

Esta relación contiene datos generales de la factura como fecha, folio, proveedor.

Atributos:

Descripción	Campo	Tipo	Índice
Número de folio de la factura	sFAC_FOLIO	VarChar(15)	Llave foránea
Clave del proveedor	nPROV_CVE	Numeric(5)	Llave foránea
Precio del equipo	nEQU_PREC	Decimal(9)	
Fecha de la factura	dFAC_FECH	DateTime(8)	
Fecha de garantía	dFAC_FECH_GARAN	DateTime(8)	
Fecha de inicio de seguro	dFAC_SEG_FECH_INI	DateTime(8)	
Fecha de fin de seguro	dFAC_SEG_FECH_FIN	DateTime(8)	

Relación: DSEGURIDAD

En esta relación se dan los datos del personal autorizado para acceder al sistema SIESM.

Atributos:

Descripción	Campo	Tipo	Índice
Nombre	sSEG_NOM	VarChar(20)	Llave primaria
Pasaporte	sSEG_PASS	VarChar(14)	
Clave del perfil	nPERF_CVE	Numeric(5)	

Relación: DFOLSOL

Esta relación contiene datos sobre el folio de la solicitud de equipo. Esta relación es una proyección de la relación DSOLICITUD.

Atributos:

Descripción	Campo	Tipo	Índice
Número de folio de asignación	nASI_FOL	Integer(4)	Llave foránea
Identificador del proyecto solicitante	sSOL_PROY_ID	VarChar(7)	
Descripción del proyecto solicitante	sSOL_PROY_DESC	VarChar(50)	
Clave del sector según catálogo	nCAT_SEC_CVE	Numeric(5)	
Clave del estatus según catálogo	nCAT_STA_CVE	Numeric(5)	
Asignación de la solicitud	sSOL_ASI_IS	VarChar(4)	
Nombre de la asignación solicitada	sSOL_ASI_NOM	VarChar(50)	
Teléfono del solicitante	sSOL_TEL	VarChar(20)	
Cantidad solicitada	nSOL_CANTIDAD	Integer(4)	
Comentarios de la solicitud	nSOL_COMENT	VarChar(60)	

Relación: DPERSOL

Esta relación contiene datos sobre la persona que solicita el equipo. Esta relación es una proyección de la relación DSOLICITUD.

Atributos:

Descripción	Campo	Tipo	Índice
Número de folio de asignación	nASI_FOL	Integer(4)	Llave foránea
Persona que solicita	nSOL_PERSONA	VarChar(50)	

Relación: TREPINVGAREQU

Relación referente al reporte del inventario y garantía del equipo. Esta relación es una proyección de la relación TREPORTES.

Atributos:

Descripción	Campo	Tipo	Índice
Número de usuario	nUSER	Integer(4)	Llave primaria
Nombre	sSEG_NOM	VarChar(20)	Llave foránea
Código de serie Softtek	sCOM_COD_SOFT	VarChar(8)	
Número de folio de la factura	sFAC_FOLIO	VarChar(15)	
Clave del proveedor	nPROV_CVE	Numeric(5)	
Descripción del proveedor	sPROV_DESC	VarChar(50)	
Descripción de inventario según catálogo	sCAT_INV_DESC	VarChar(50)	
Descripción del estatus según catálogo	sCAT_STA_DESC	VarChar(50)	
Fecha de la factura	dFAC_FECH	DateTime(8)	
Fecha de garantía	dFAC_FECH_GARAN	DateTime(8)	
Fecha de inicio de seguro	dFAC_SEG_FECH_INI	DateTime(8)	
Fecha de fin de seguro	dFAC_SEG_FECH_FIN	DateTime(8)	

Relación: TREPASIGDEVEQU

Relación referente al reporte de la asignación y devolución del equipo. Esta relación es una proyección de la relación TREPORTES.

Atributos:

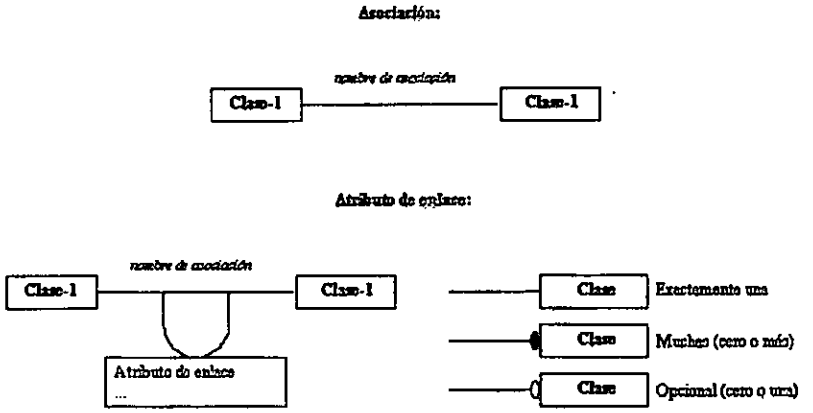
Descripción	Campo	Tipo	Índice
Número de usuario	nUSER	Integer(4)	Llave primaria
Nombre	sSEG_NOM	VarChar(20)	Llave foránea
Número de folio de asignación	nASI_FOL	Integer(4)	
Clave de inventario según catálogo	nCAT_INV_CVE	Numeric(5)	
Clave del estatus según catálogo	nCAT_STA_CVE	Numeric(5)	
Persona	sEQU_PER	Char(3)	
Fecha de asignación	dASI_FECH	DateTime(8)	
Fecha de devolución	dASI_FECH_DEV	DateTime(8)	
Identificador del proyecto solicitante	sSOL_PROY_ID	VarChar(7)	
Descripción del proyecto solicitante	sSOL_PROY_DESC	VarChar(50)	
Clave del sector según catálogo	nCAT_SEC_CVE	Numeric(5)	
Descripción de sector según catálogo	sCAT_SEC_DESC	VarChar(50)	
Asignación de la solicitud	sSOL_ASI_IS	VarChar(4)	
Nombre de la asignación solicitada	sSOL_ASI_NOM	VarChar(50)	

4.2.3.5. Paso del Modelo Entidad-Relación al Modelo de Objetos

Una vez obtenido el modelo entidad-relación con sus entidades, asociaciones, atributos, y normalizado, se procederá a convertir dicho modelo a un modelo de objetos. Las relaciones obtenidas se pasarán a su equivalente en clases por medio de la notación OMT. (Utilización de técnicas métodos y herramientas para el desarrollo de sistemas de información. Según IEEE Estándar 730 y 983 (1984), Documentación y Estándares y convenciones. Butler (1995).Herramientas técnicas y metodologías Pressman (1995) Aplicación de métodos técnicos. Stamm 1981 Metodología de ingeniería del software. ISO 9000 –3 Actividades del ciclo de vida (Diseño e implantación).

Reglas de correspondencia entre modelos de objetos y relaciones

- Cada clase se corresponde con una o más relaciones. (Similarmente, una relación puede corresponder a más de una relación si están conectadas mediante una asociación uno-a-uno o bien uno-a-muchos).
- Cada asociación uno-a-muchos se corresponde con una relación diferente, o bien se puede incluir en forma de clave externa dentro de la relación para la clase "muchos".
- Cada asociación uno-a-uno se corresponde con una relación distinta, o bien puede ser incluida en forma de clave externa dentro de la relación de cualquiera de las clases.
- Para las asociaciones uno-a-muchos y uno-a-uno, si no hay ciclos, se tiene la opción adicional de almacenar la asociación y ambos objetos relacionados dentro de una misma relación.
- Los nombres de rol se incorporan como parte del nombre de atributo de las claves externas.



Notación gráfica para el modelo de objetos en OMT

Una vez establecidas las reglas de correspondencia necesarias para la conversión entre las relaciones y modelos de objetos, al igual que la notación gráfica OMT a utilizar, se procederá a realizar el modelo de objetos en base al modelo entidad-relación y normalización antes desarrollados.

A continuación se muestra el modelo de objetos resultante con la notación gráfica OMT.

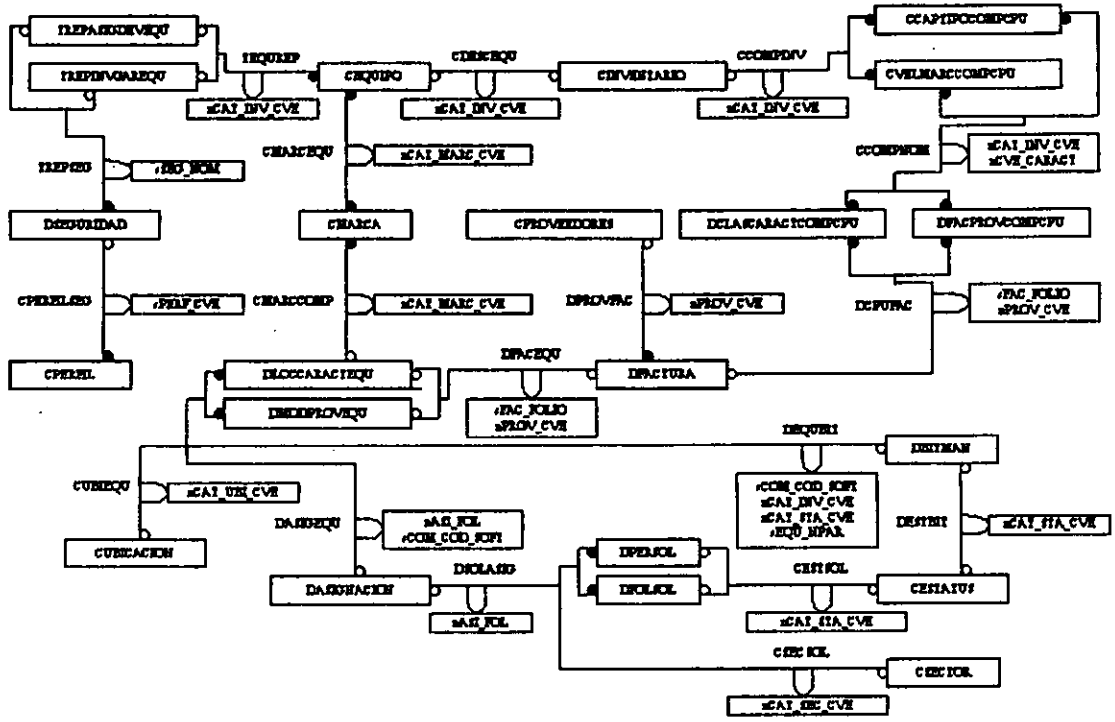
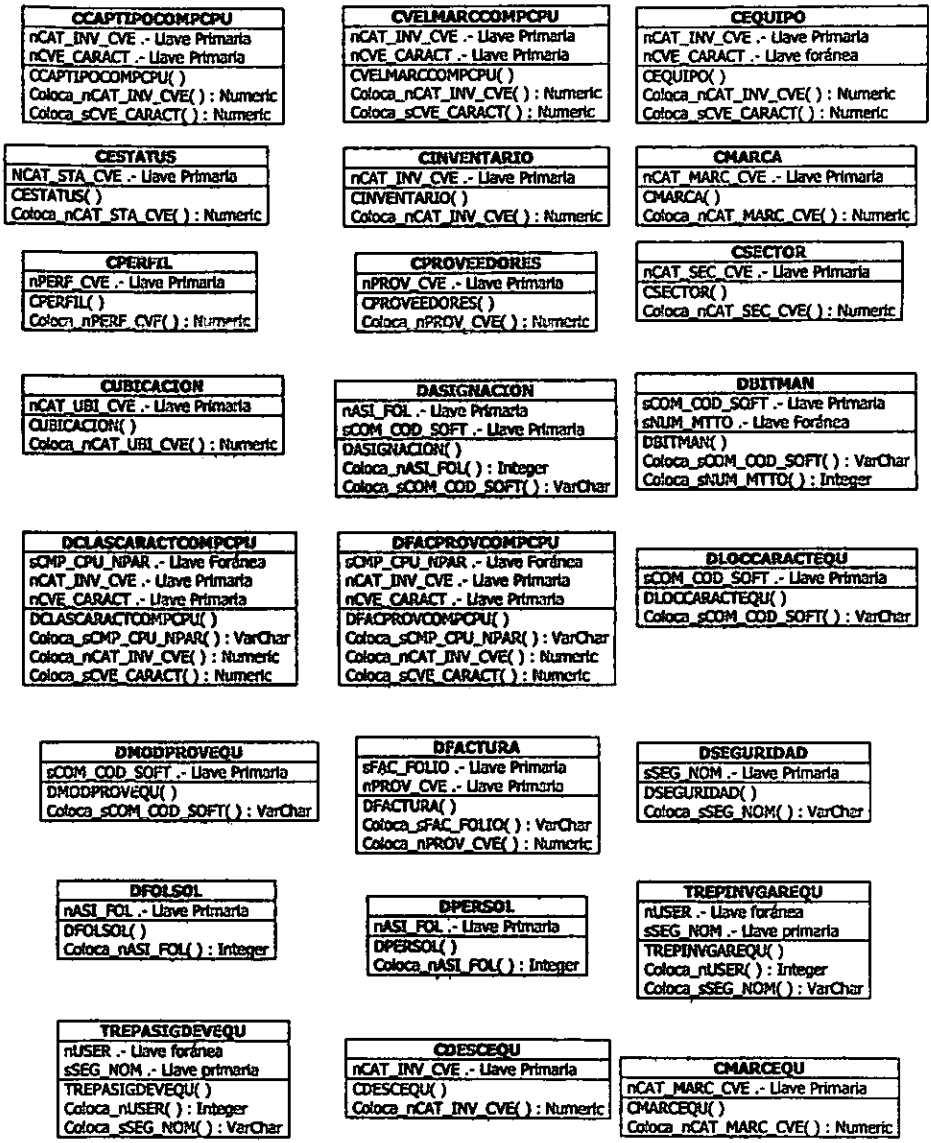
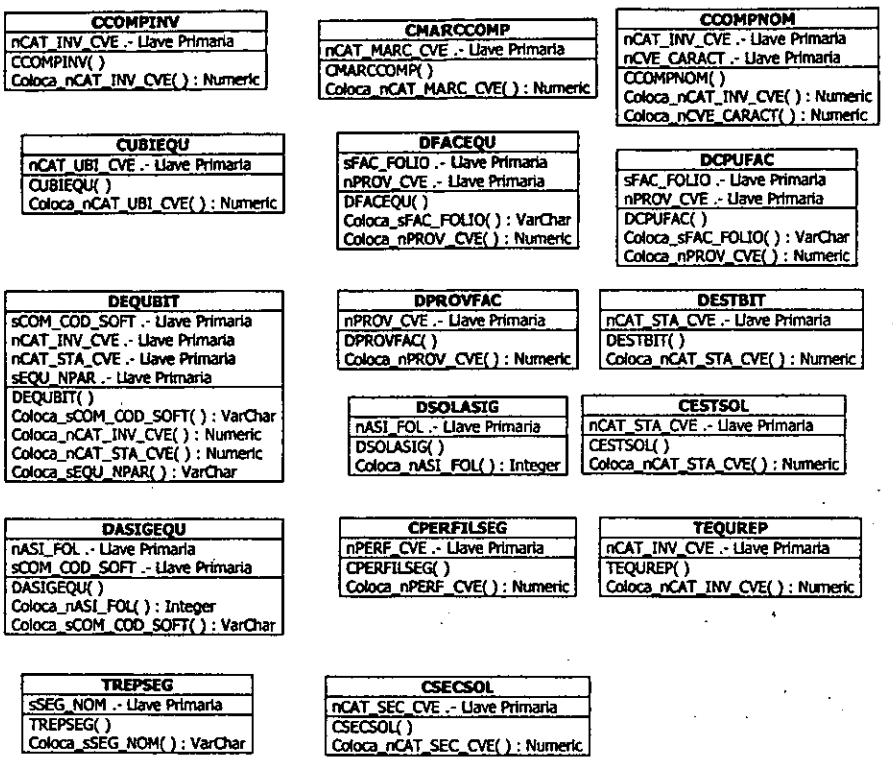


Diagrama de Objetos

Clases del modelo de objetos

Ya establecido el modelo de objetos, se procederá a desplegar sus clases con atributos y operaciones referentes a sus índices.





4.2.4. Diccionario de datos

En seguida se dará una breve descripción de los términos más relevantes usados dentro del sistema de información. (Estándares específicos para la definición de criterios de desarrollo para la guía en la elaboración del software. Según IEEE Estándar 730 y 983 (1984), Documentación y Estándares y convenciones. Stamm 1981 Metodología de ingeniería del software. ISO 9000 –3 Actividades del ciclo de vida (Diseño e implantación)

AbreConexion.- Establece una conexión a la base de datos SIESM.

ActualizarCaracter.- Actualiza registros en el Catálogo de Características de Cmp. de CPU.

ActualizarDatosBitMtto.- Actualiza los datos del equipo para indicar cuando está o no en Mantenimiento.

ActualizarEquipo.- Actualiza registros en el catálogo de Equipos.

ActualizarEstatus.- Actualiza registros en el catálogo de estatus de equipo.

ActualizarMarca.- Actualiza registros en el catálogo de Marcas.

ActualizarProyecto.- Actualiza registros en el catálogo de Proyectos.

ActualizarRegAsig.- Actualiza los datos del la asignación y el equipo asignado para una solicitud.

ActualizarRegistro.- Actualiza las relaciones de factura y equipo.

ActualizarSector.- Actualiza registros en el catálogo de Sectores.

ActualizarUbicacion.- Actualiza registros en el catálogo de Ubicaciones.

ActualizarUsuario.- Actualiza registros en la relación Seguridad.

AgregarBitMtto.- Registra en la relación Bitácora de Mantenimientos los equipos que entran a mantenimiento.

AgregarCaracter.- Inserta registros en el Catálogo de Características de Cmp. de CPU.

AgregarDatos.- Inserta registros en las relaciones factura y equipo.

AgregarEquipo.- Inserta registros en el catálogo de Equipos.

AgregarEstatus.- Inserta registros en el catálogo de estatus de equipo.

AgregarMarca.- Inserta registros en el catálogo de Marcas.

AgregarProyecto.- Inserta registros en el catálogo de Proyectos.

AgregarRegAsig.- Registra en la relación Asignación las fechas, en la tabla de equipo registra el equipo asignado a una solicitud y en la de Solicitud registra los datos referentes a la solicitud; solo que con un folio asignado.

AgregarSector.- Inserta registros en el catálogo de Sectores.

AgregarUbicacion.- Inserta registros en el catálogo de Ubicaciones.

AgregarUsuario.- Inserta registros en la relación Seguridad.

ASIGNACION.- La asignación contiene todos los datos necesarios para saber en donde y con quien fue asignado un equipo, así como la descripción del equipo asignado y la cantidad.

BITMAN.- Contiene la información de la bitácora de mantenimiento.

BorrarCaracter.- Elimina registros en el Catálogo de Características de Cmp. de CPU.

BorrarEquipo.- Elimina registros en el catálogo de Equipos.

BorrarMarca.- Elimina registros en el catálogo de Marcas.

BorrarProyecto.- Elimina registros en el catálogo de Proyectos.

BorrarRegAsig .- Elimina los registros de la relación Solicitud, Asignación y cambia el estatus del equipo en la relación equipo; todos correspondientes a un mismo folio.

BorrarRegistro.- Elimina registros de las relaciones factura y equipo.

BorrarSector.- Elimina registros en el catálogo de Sectores.

BorrarStatus.- Elimina registros en el catálogo de estatus de equipo.

BorrarUbicacion.- Elimina registros en el catálogo de Ubicaciones.

BorrarUsuarior.- Elimina registros en la relación Seguridad.

CATALOGO.- Catálogo de tipos de memoria, tipos de tarjeta de red, de tipos de discos duros, de tipos de procesadores, de tipos de conectores.

CATCOMPONENTE.- Catálogo con todos los nombres de los componentes que pueden formar parte de una computadora, dispositivos internos y externos.

CCAPTIPOCOMPCPU.- Catálogo con características sobre capacidad y tipo de componentes en una computadora.

CCOMPINV.- Clasifica equipo por componentes.

CCOMPNOM.- Presenta clasificación y descripción de los componentes de una computadora por nombre.

CCOMPONENTESCPU.- Catálogo con todos los nombres de los componentes de una computadora.

CDESCEQU.- Presenta descripción del equipo.

CEQUIPO.- Catálogo con las características del equipo existente.

CerrarConexion.- Cierra la conexión global a la base de datos SIESM.

CESTATUS.- Catálogo que indica la situación actual del equipo.

CESTSOL.- Presenta la situación actual del equipo según su asignación en el folio.

CINVENTARIO.- Catálogo para clasificar el equipo.

claseConcluir.- Cierra la conexión al momento en el que se destruye la clase.

CMARCA.- Catálogo para registrar la marca de los componentes.

CMARCCOMP.- Clasifica y describe el equipo por marca de los componentes.

CMARCEQU.- Clasifica equipo por marca.

ColocaRegistrosSeleccionados.- Hace una actualización masiva a partir de un criterio de selección.

COMPONENTE.- En esta relación se registrarán todos los componentes existentes, desde equipos completos hasta componentes solos.

CPERFIL.- Catálogo con la descripción del perfil del usuario.

CPERFILSEG.- Presenta el perfil del usuario según su nivel de autorización.

CPROVEEDORES.- Catálogo con la información de proveedores de equipo.

CPROVFAC.- Presenta información de proveedores de equipo según datos de factura.

CPU.- Contiene la información del cpu formado por tarjeta de red, disco duro, memoria, procesador.

CSECSOL.- Clasifica sectores ó actividades por regiones según la asignación en el folio.

CSECTOR.- Catálogo de sectores ó actividades por regiones.

CUBICACION.- Catálogo en el que se indica donde será asignado el equipo.

CUBIEQU.- Asignación de equipo.

CVELMARCCOMPCPU.- Catálogo con características sobre velocidad y marca de componentes de una computadora.

dASI_FECH.- Fecha de asignación.

dASI_FECH_DEV.- Fecha de devolución.

DASIGEQU.- Clasifica y describe el equipo según dónde y con quién fue asignado.

DASIGNACION.- Datos necesarios para saber en dónde y con quién fue asignado un equipo.

dBIT_FECH_FIN.- Fecha de realización del mantenimiento.

dBIT_FECH_ING.- Fecha de ingreso en la bitácora.

DBITMAN.- Datos referentes a la bitácora de mantenimiento.

DCLASCARACTCOMPCPU.- Datos referentes a la clasificación y características de los componentes de una computadora.

DCOMPONENTESCPU.- Datos referentes a la clasificación y descripción de los componentes de una computadora.

DCPUFAC.- Presenta datos generales de clasificación, descripción y factura de los componentes de una computadora.

DEQUBIT.- Clasifica y describe el equipo a partir de la bitácora de mantenimiento.

DEQUIPO.- Datos sobre la clasificación y descripción del equipo.

DESTBIT.- Presenta situación actual del equipo según la bitácora de mantenimiento.

dFAC_FECH.- Fecha de la factura.

dFAC_FECH_GARAN.- Fecha de garantía.

dFAC_SEG_FECH_INI.- Fecha de inicio de seguro.

dFAC_SEG_FECH_FIN.- Fecha de fin de seguro.

DFACEQU.- Presenta descripción con datos generales de la factura de equipo.

DFACPROVCOMPCPU.- Datos referentes a la factura y proveedor de los componentes de una computadora.

DFACTURA.- Datos generales de la factura como fecha, folio, proveedor.

DFOLSOL.- Datos sobre el folio de la solicitud de equipo.

DLOCCARACTEQU.- Datos sobre la localización y características del equipo.

DMODPROVEQU.- Datos sobre el modelo y proveedores del equipo.

DPERSOL.- Datos sobre la persona que solicita el equipo.

DSEGURIDAD.- Datos del personal autorizado para acceder al sistema SIESM.

DSOLASIG.- Presenta folio de asignación de equipo según donde y con quien fue asignado.

DSOLICITUD.- Datos sobre el folio de la solicitud de equipo.

EjecutaConsulta.- Ejecuta una consulta de acción (inserta, actualiza, borra).

ESTATUS.- Nivel que puede tomar el componente con valores tales como: Disponible, reparación con garantía, reparación sin garantía, asignación, baja, equipo de SOFTEK, personal, arrendado, solicitud parcialmente atendida, solicitud atendida, solicitud no atendida.

FACTURA.- Datos generales de la factura como fecha, folio, proveedor.

ImprimeAsig.- Imprime el equipo asignado a una solicitud con la respectiva responsiva.

ImprimeCaracter.- Genera un reporte del Catálogo de Características de Cmp. de CPU.

ImprimeCons.- Imprime la consulta realizada previamente.

ImprimeEquipo.- Genera un reporte del catálogo de Equipos.

ImprimeMarca.- Genera un reporte del catálogo de Marcas.

ImprimeSector.- Genera un reporte del catálogo de Sectores.

ImprimeStatus.- Genera un reporte del catálogo de status de equipo.

ImprimeUbicacion.- Genera un reporte del catálogo de Ubicaciones.

ImprimeUser.- Genera un reporte de la relación Seguridad.

MARCA.- Catálogo de marcas.

Mensajes.- Maneja los mensajes del sistema.

nASI_FOL.- Número de folio de asignación.

nCAT_INV_CVE.- Clave de inventario según catálogo.

nCAT_MARC_CVE.- Clave de marca según catálogo.

nCAT_SEC_CVE.- Clave de sector según catálogo.

nCAT_STA_CVE.- Clave del estatus según catálogo.

nCAT_UBI_CVE.- Clave de la ubicación según catálogo.

nCVE_CARACT.- Clave de características.

nEQU_PREC.- Precio del equipo.

nPERF_CVE.- Clave del perfil.

nPROV_CVE.- Clave de proveedores.

nSOL_CANTIDAD.- Cantidad solicitada.

nSOL_COMENT.- Comentarios de la solicitud.

nSOL_PERSONA.- Persona que solicita.

nUSER.- Número de usuario.

ObtieneAsigEquipo.- Obtiene datos de la relación Equipo.

ObtieneBitacora.- Obtiene datos de la relación BITMAN.

ObtieneBitMtto.- Obtiene datos de la relación Bitácora de Mantenimientos.

ObtieneCaracter.- Consulta al Catálogo de Características de Componentes de CPU.

ObtieneConexion.- Realizar la conexión a la base de datos SIESM.

ObtieneConsultaReg.- Consulta por su llave (relación EQUIPO).

ObtieneDatosBitMtto.- Obtiene los datos relacionados con un equipo (Ubicación, Estatus, Proyecto, etc.).

ObtieneDatosEquipo.- Obtiene los datos relacionados con el equipo solicitado; en la BD ASIGNACIONES.

ObtieneDatosFolio.- Obtiene los datos relacionados con el folio asignado a una solicitud; en la BD ASIGNACIONES.

ObtieneDatosSolicitud.- Obtiene los datos relacionados con el solicitante de equipo; en la BD ASIGNACIONES.

ObtieneEquipo.- Consulta al Catálogo de Equipos.

ObtieneEstatus.- Consulta al Catálogo de estatus de equipo.

ObtieneEstatusEquipo.- Obtiene los datos relacionados con el equipo que tengan cierto estatus; en la BD SIESM.

ObtieneEstatusSolicitud.- Obtiene el estatus de la relación SOLICITUD.

ObtieneFolioIntrasoft.- Obtiene los datos relacionados con la solicitud de las tablas de Intrasoft (Sector, Estatus, Proyecto, etc.).

ObtieneFolioSIESM.- Obtiene datos de las relaciones Solicitud, Asignación y equipo de la BD SIESM.

ObtieneInfoGral.- Obtiene los datos relacionados con las características del equipo; en la BD SIESM.

ObtieneMarca.- Consulta al Catálogo de Marcas.

ObtienePertenencia.- Obtiene datos de pertenencia de la relación EQUIPO de la BD SIESM.

ObtieneProcesador.- Obtiene datos de la relación CMP_CPU.

ObtieneProyecto.- Consulta al Catálogo de Proyectos.

ObtieneSolicitud.- Obtiene datos de la relación SOLICITUD.

ObtieneRegistrosSeleccionados.- Ejecuta una consulta en base a un criterio de selección.

ObtieneSector.- Consulta al Catálogo de Sectores.

ObtieneUbicacion.- Consulta al Catálogo de Ubicaciones.

ObtieneUsuario.- Consulta a la relación Seguridad.

sBIT_COMENT.- Comentarios en la bitácora.

sCAT_COMP_CAP.- Capacidad de componente según catálogo.

sCAT_COMP_DESC.- Descripción de componente según catálogo.

sCAT_COMP_MCADDRESS.- Marca y dirección de componente según catálogo.

sCAT_COMP_TIPO.- Tipo de componente según catálogo.

sCAT_COMP_VEL.- Velocidad de componente según catálogo.

sCAT_EQP_DESC.- Descripción de equipo según catálogo.

sCAT_EQP_TIPO.- Tipo de equipo según catálogo.

sCAT_INV_DESC.- Descripción de inventario según catálogo.

sCAT_MARC_DESC.- Descripción de marca según catálogo.

sCAT_SEC_DESC.- Descripción de sector según catálogo.

sCAT_STA_DESC.- Descripción del estatus según catálogo.

sCAT_UBI_DESC.- Descripción de la ubicación según catálogo.

sCMP_CPU_NPAR.- Código de partes de una computadora.

sCOM_COD_SOFT.- Código de serie Softtek.

SECTOR.- Catálogo de sectores como TELCOS, FINANCIERO, INDUSTRIA, SERVICIOS, GOBIERNO, SOLUCIONES, REGION NORTE, etc.

SEGURIDAD.- Contiene la información de los datos del personal autorizado para acceder al sistema SIESM.

sEQU_COMENT.- Comentarios referentes al equipo.

sEQU_MOD.- Modelo del equipo.

sEQU_NPAR.- Código de partes de un equipo.

sEQU_PER.- Persona

sFAC_FOLIO.- Número de folio de la factura.

SIESM.- Sistema de Inventario de Equipo Softtek México.

sNUM_MTTO.- Número de mantenimiento.

sPERF_DESC.- Descripción del perfil.

sPROV_DESC.- Descripción de proveedores.

sSEG_NOM.- Nombre.

sSEG_PASS.- Pasaporte.

sSOL_ASI_IS.- Asignación de la solicitud.

sSOL_ASI_NOM.- Nombre de la asignación solicitada.

sSOL_PROY_DESC.- Descripción del proyecto solicitante.

sSOL_PROY_ID.- Identificador del proyecto solicitante

sSOL_TEL.- Teléfono del solicitante.

TEQUREP.- Presenta el movimiento con las características del equipo existente.

TEQUSEG.- Clasifica datos del personal autorizado para ver movimiento y características de equipo.

TREPASIGDEVEQU.- Tabla referente al reporte de la asignación y devolución del equipo.

TREPINVGAREQU.- Tabla referente al reporte del inventario y garantía del equipo.

TREPORTES.- Tabla referente al movimiento y características del equipo.

UBICACION.- Contiene la información de la ubicación donde será asignado el equipo, esta formada por el piso y la oficina.

validaUsuario.- Realiza la validación del usuario que desea ingresar al sistema.

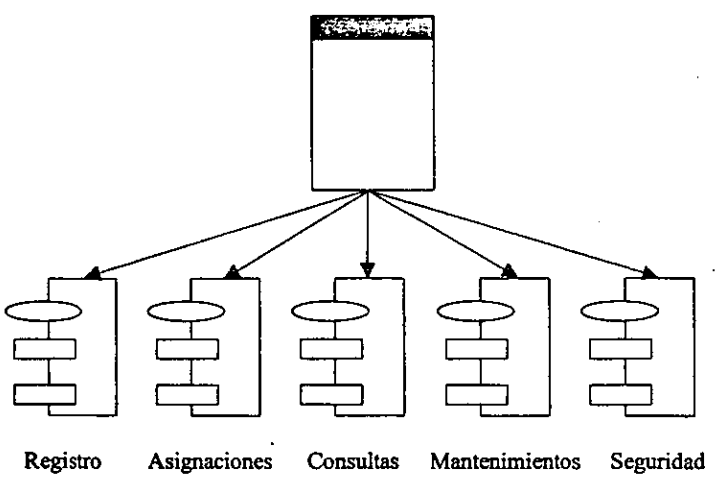
4.2.5. Diseño de pantallas

El diseño de pantallas es una parte esencial en el diseño del sistema, sólo es necesario basarse en lineamientos básicos:

- Mantener la pantalla simple
- Mantener consistente la presentación de la pantalla
- Facilitar al usuario el movimiento entre pantallas
- Crear una pantalla atractiva

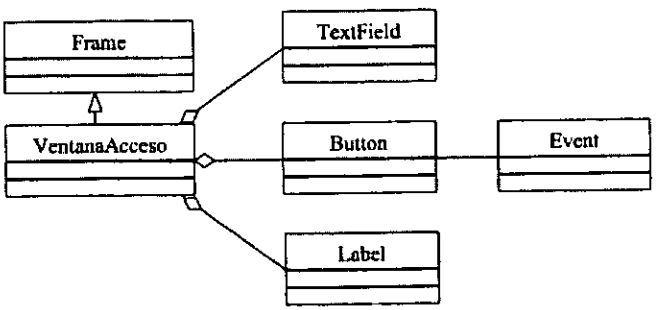
Para la mayoría de los usuarios, la interfaz o pantallas son el sistema, bien o mal, se muestran como la representación del sistema. El objetivo del diseño de las pantallas es que ayuden a los usuarios a proporcionar la información que necesitan

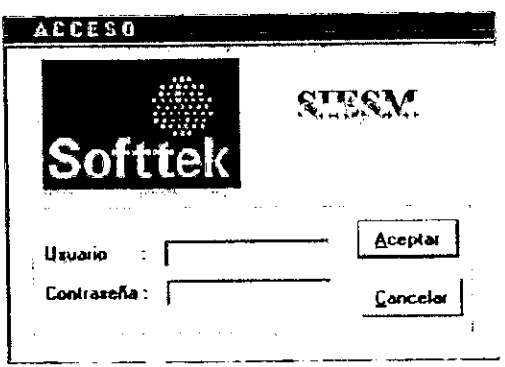
Diagrama de Módulos



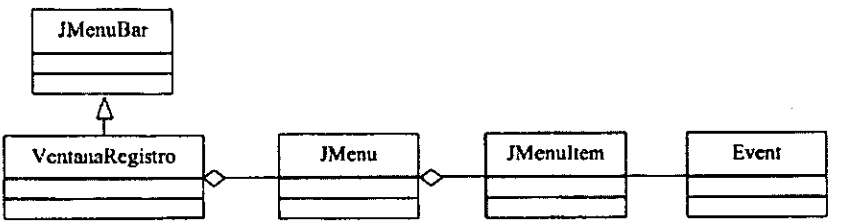
Diseño detallado

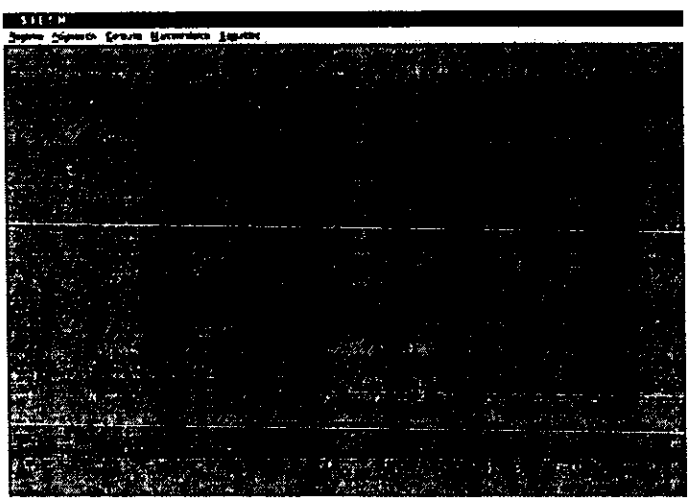
Ventana Menú



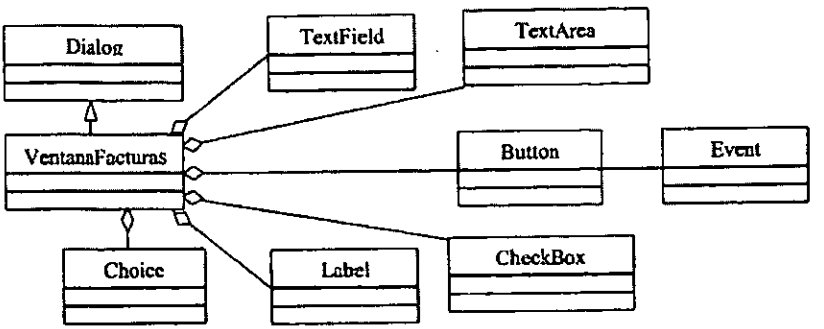


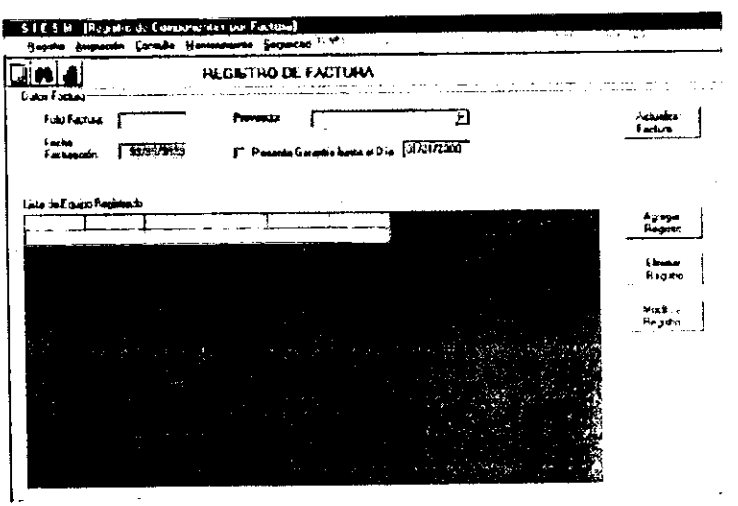
Ventana Principal



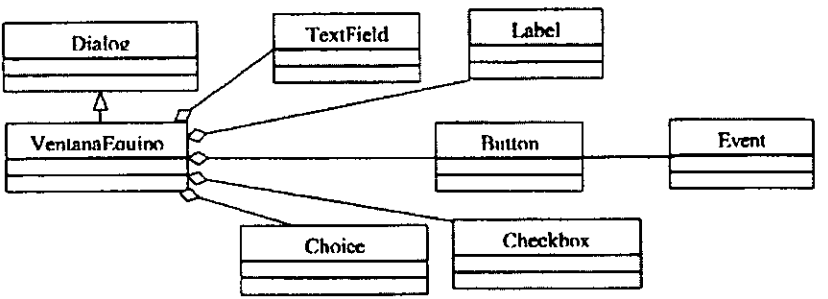


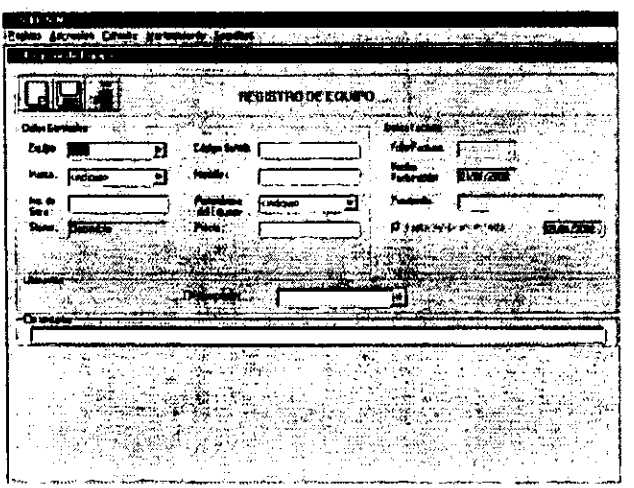
Ventana Registro de Facturas



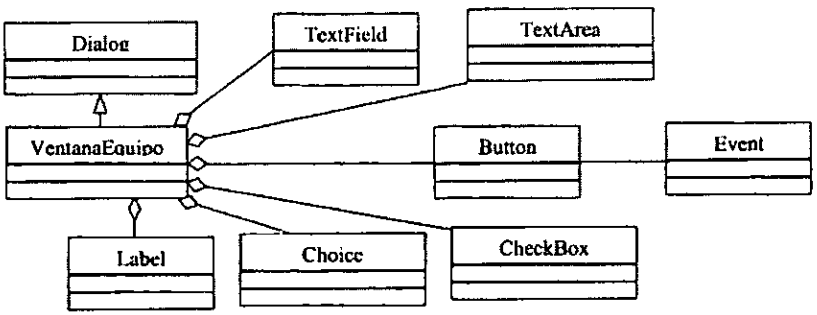


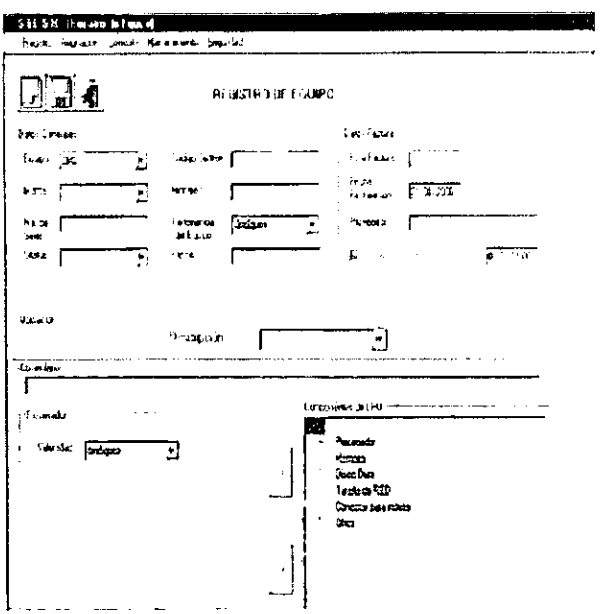
Ventana Registro de Equipo



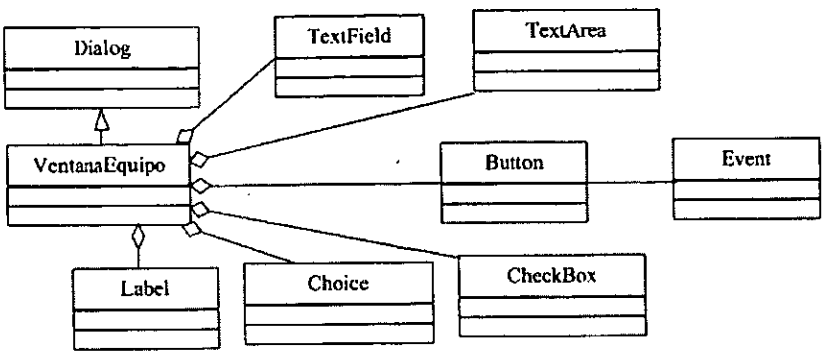


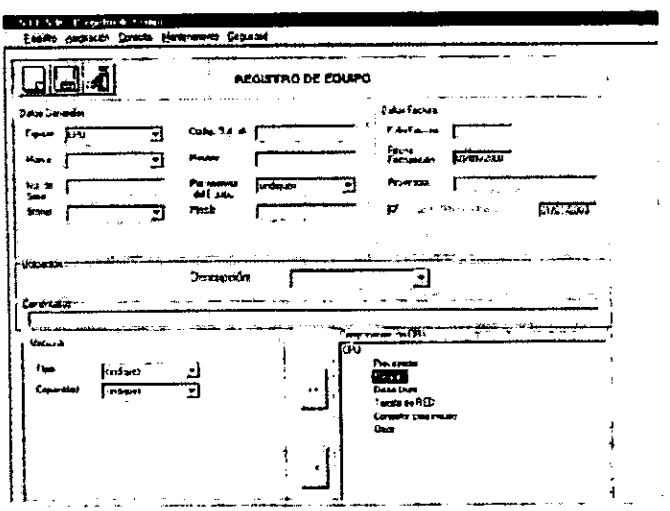
Ventana Registro de Equipo CPU (Procesador)



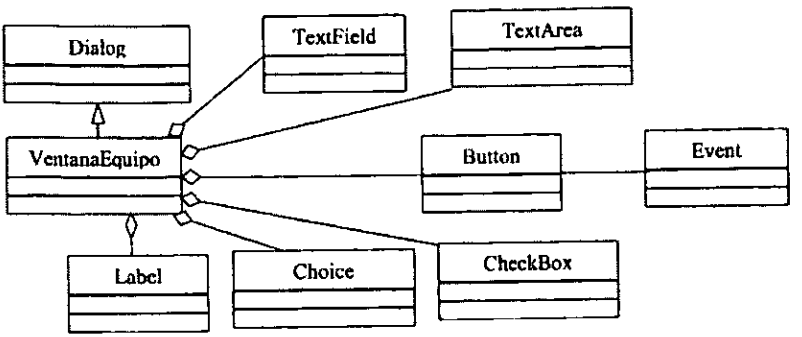


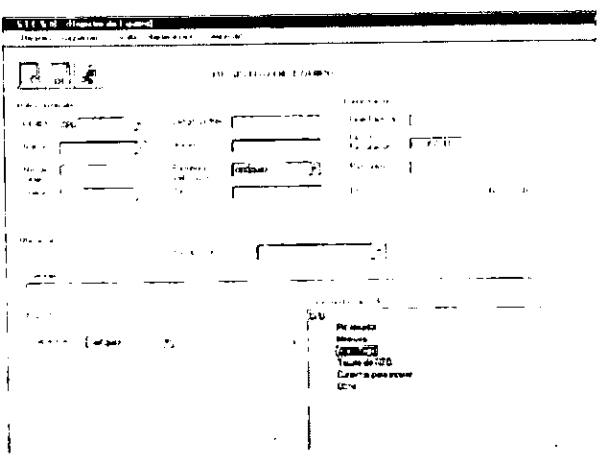
Ventana Registro de Equipo CPU (Memoria)



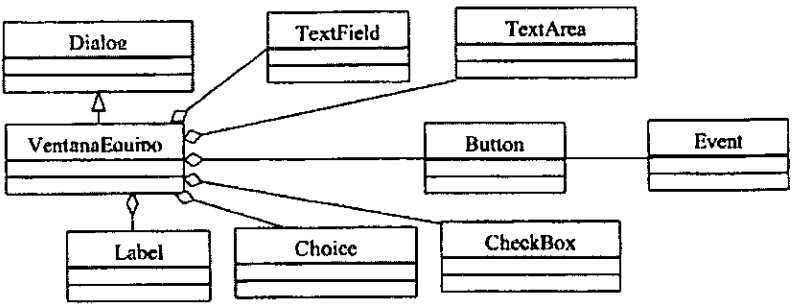


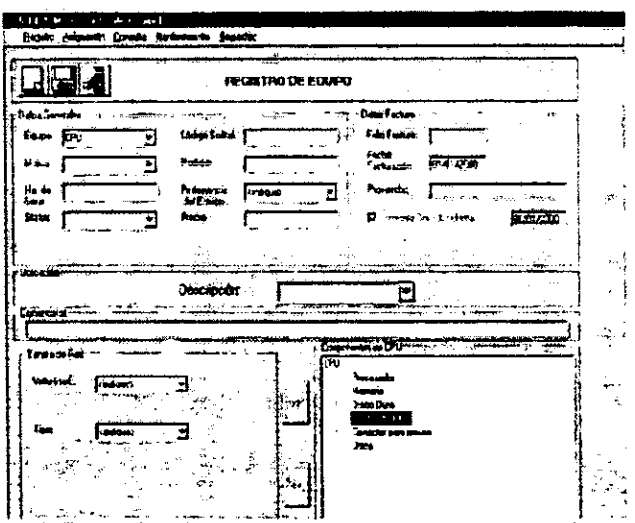
Ventana Registro de Equipo CPU (Disco Duro)



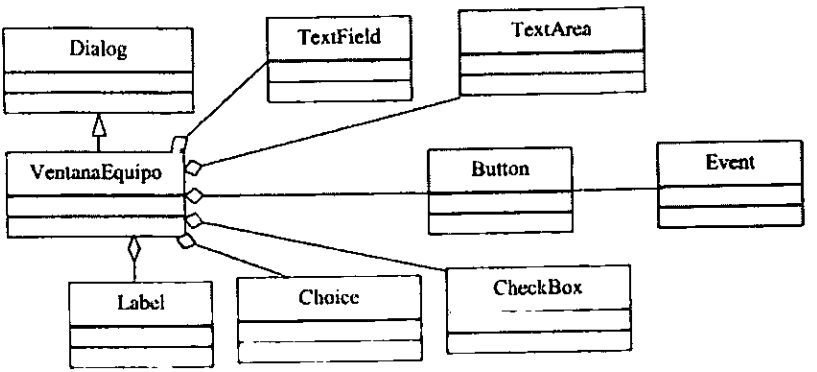


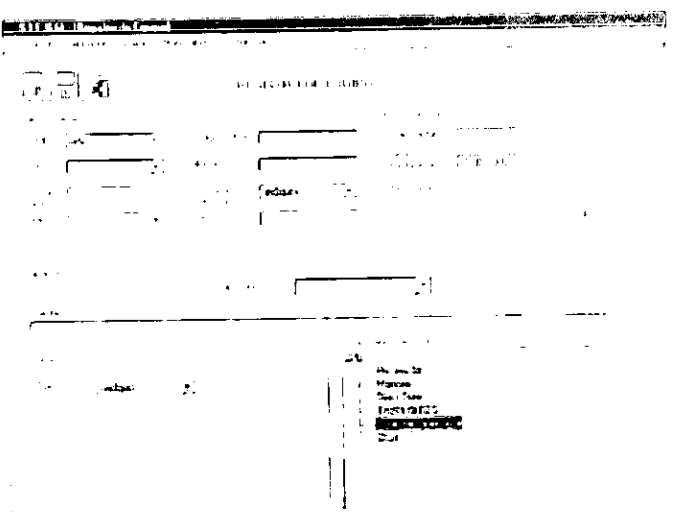
Ventana Registro de Equipo CPU (Tarjeta de Red)



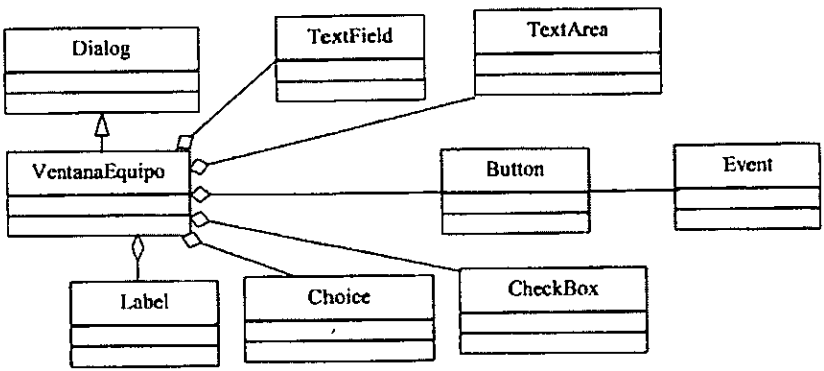


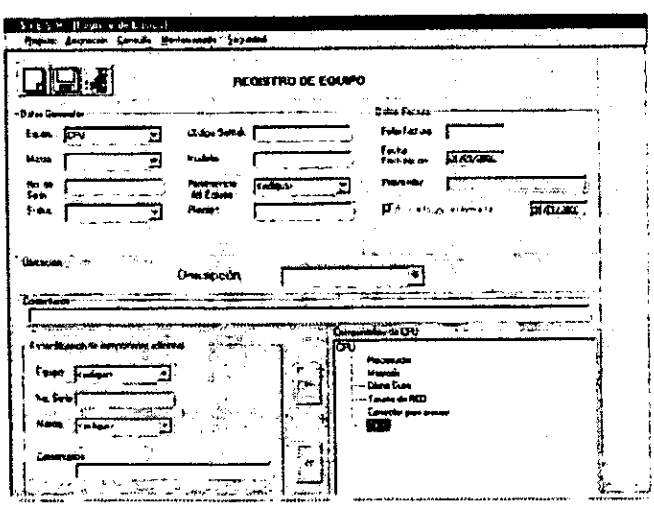
Ventana Registro de Equipo CPU (Conector)



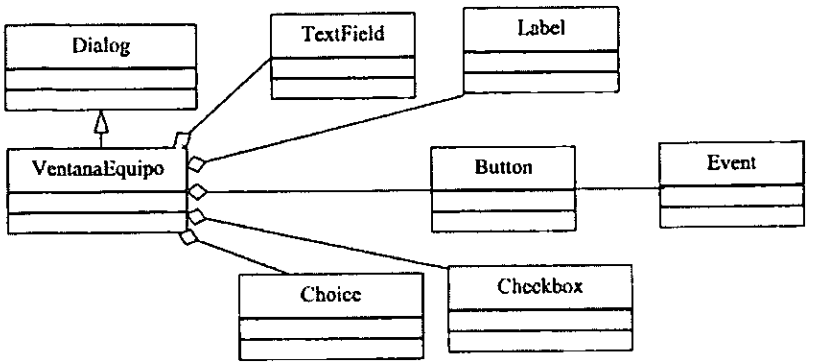


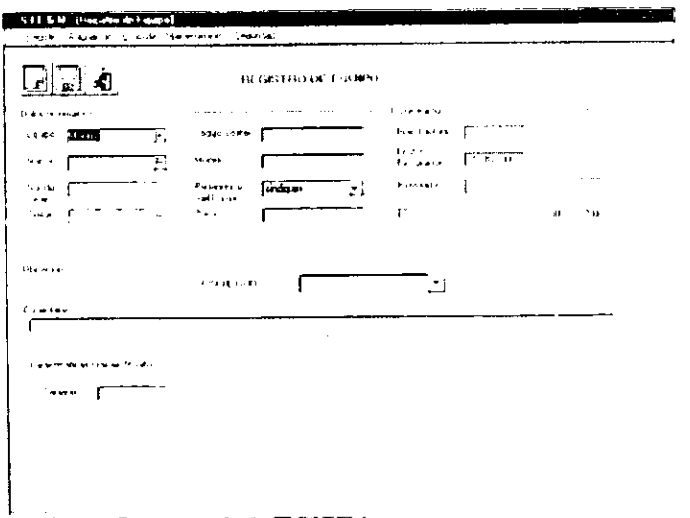
Ventana Registro de Equipo de Otros Dispositivos



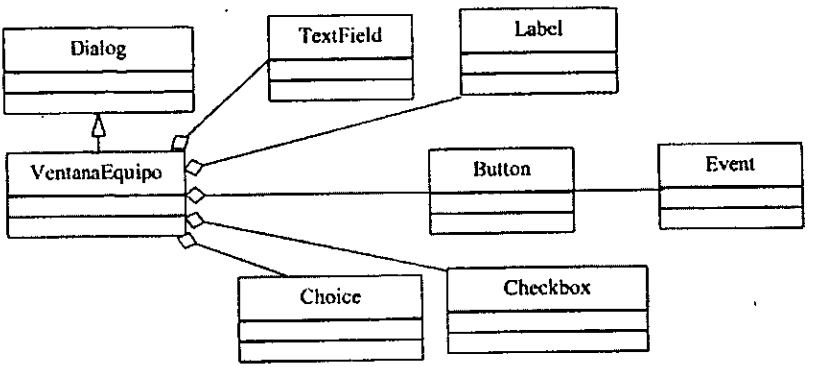


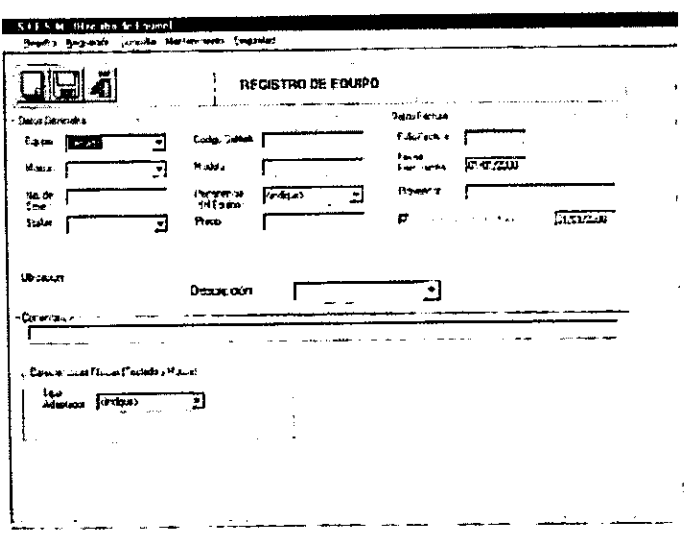
Ventana Registro de Equipo de Monitor



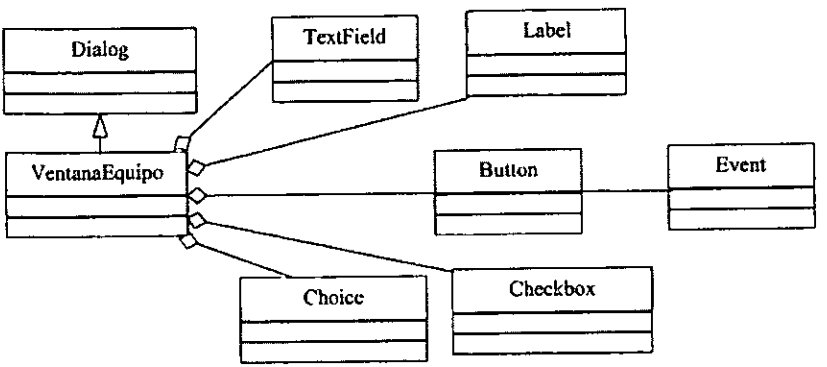


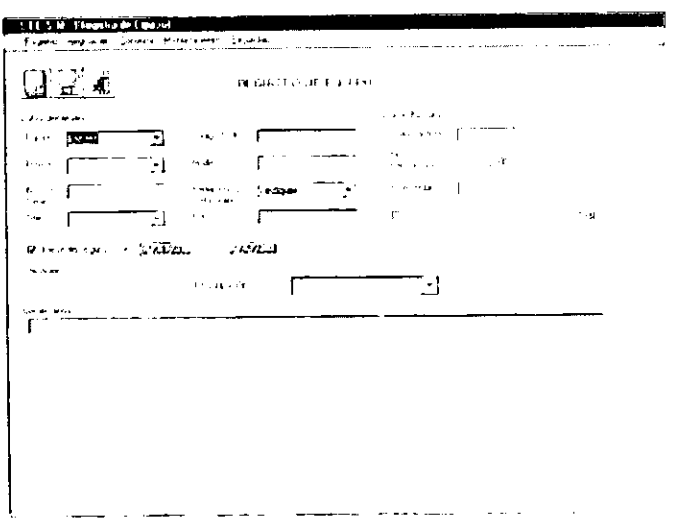
Ventana Registro de Equipo de Teclado y Mouse



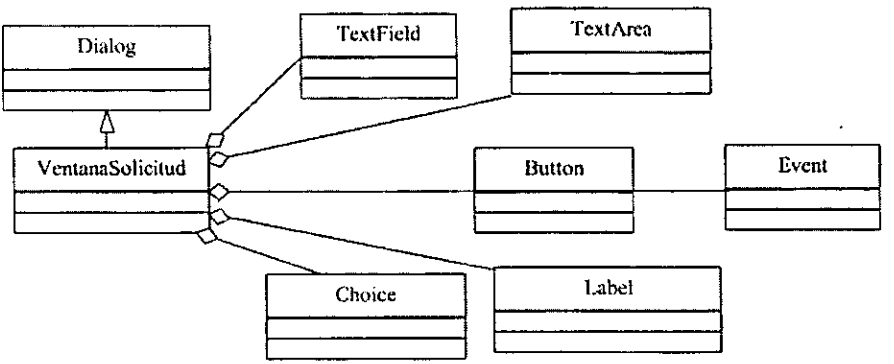


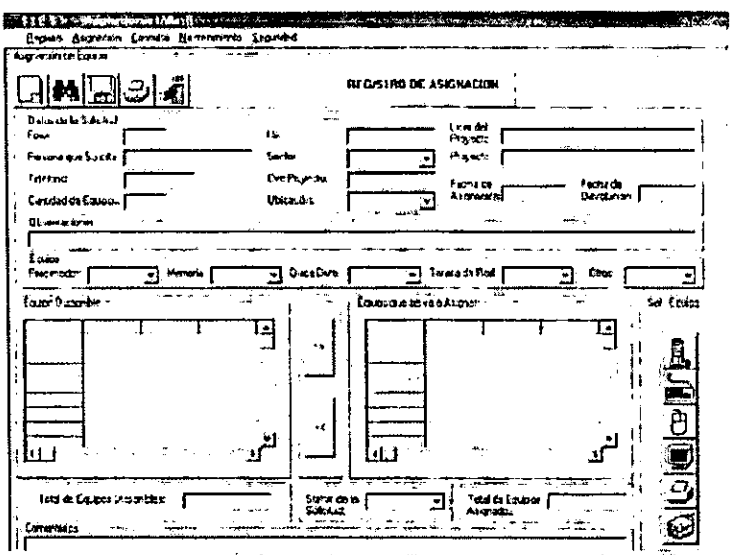
Ventana Registro de Equipo de Seguro



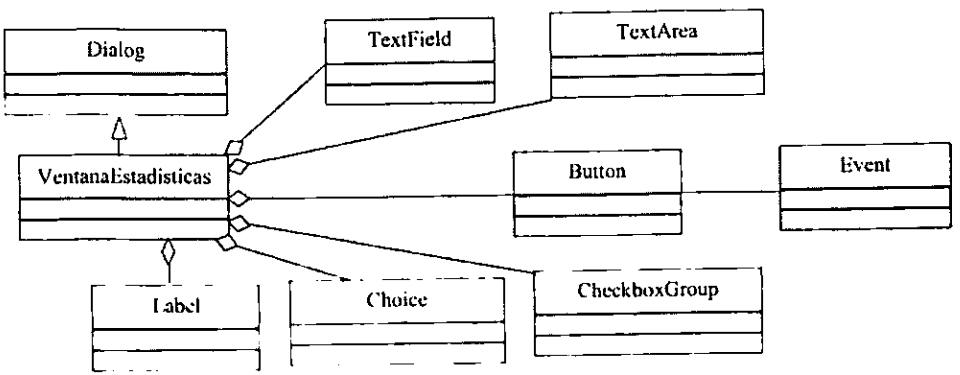


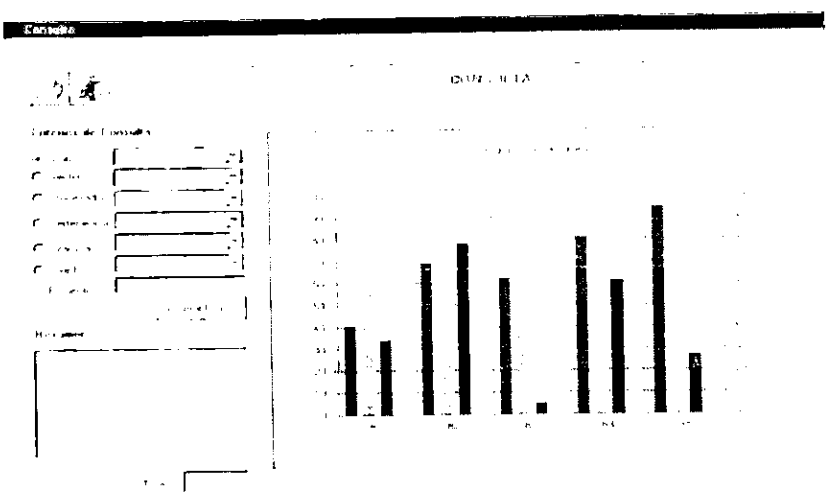
Ventana Solicitud



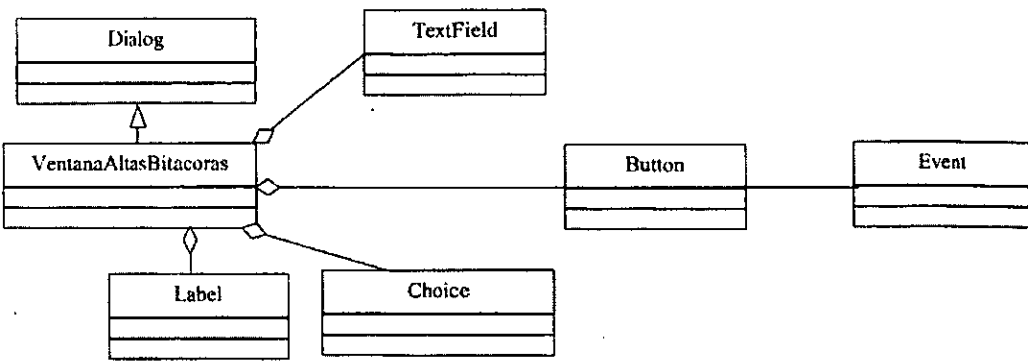


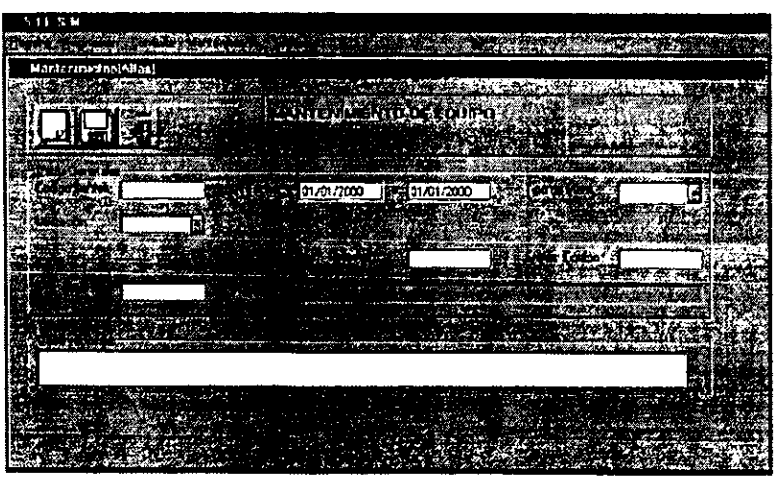
Ventana Estadísticas



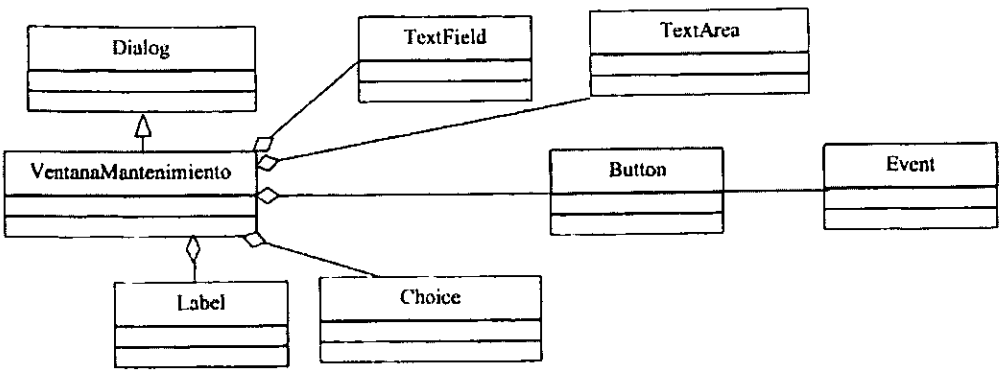


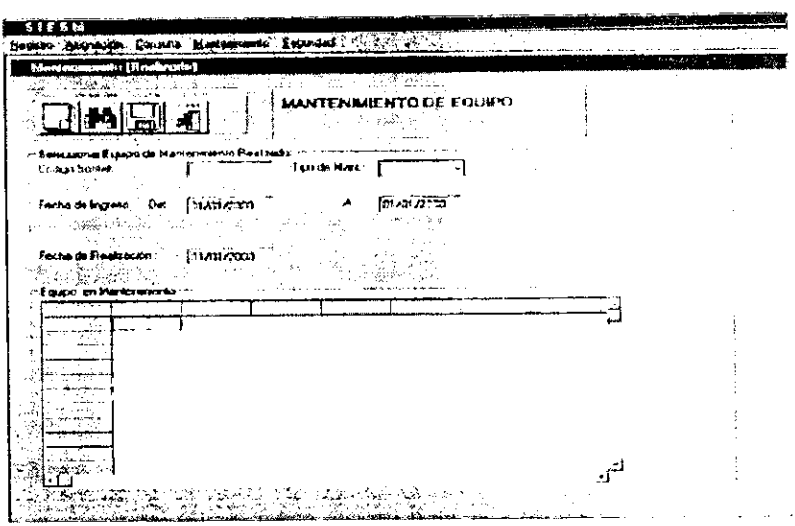
Ventana Altas de Bitácoras



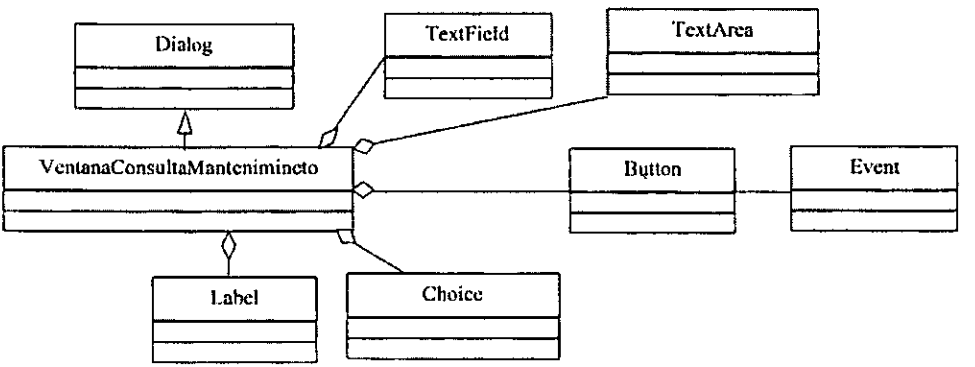


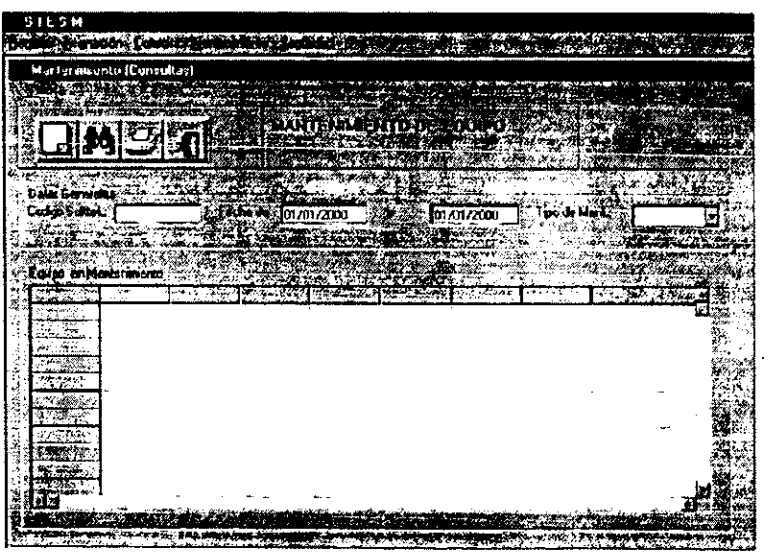
Ventana Mantenimiento Realizado



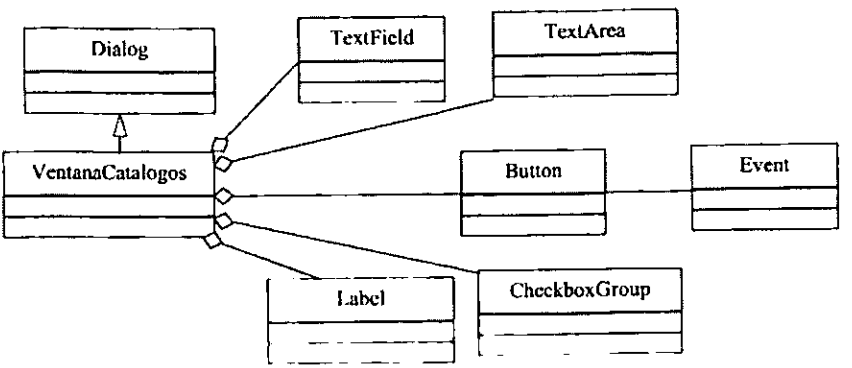


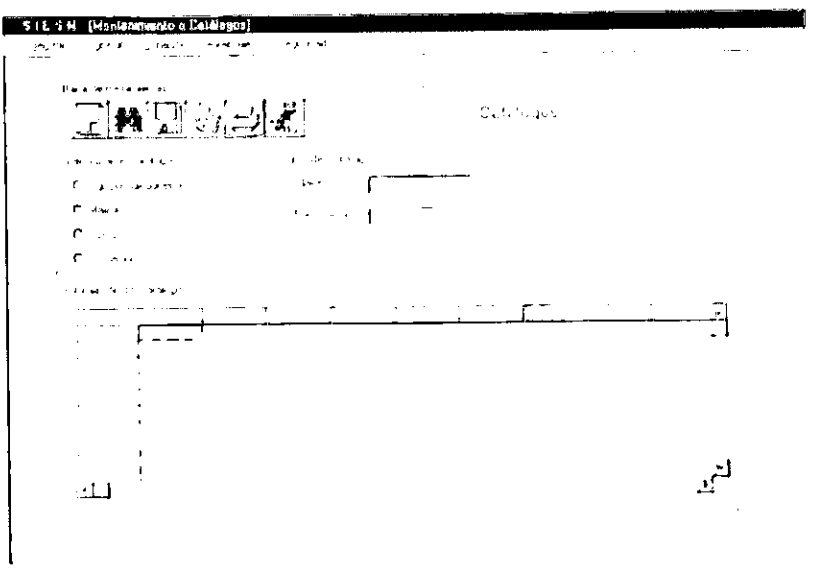
Ventana Consultas de Mantenimiento



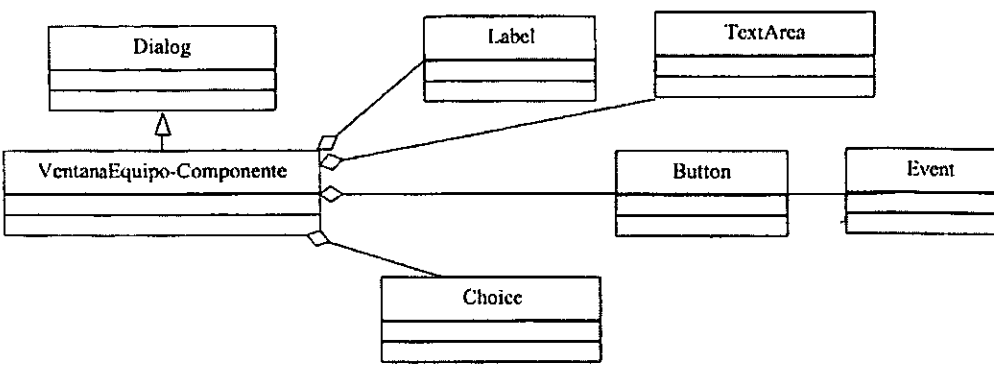


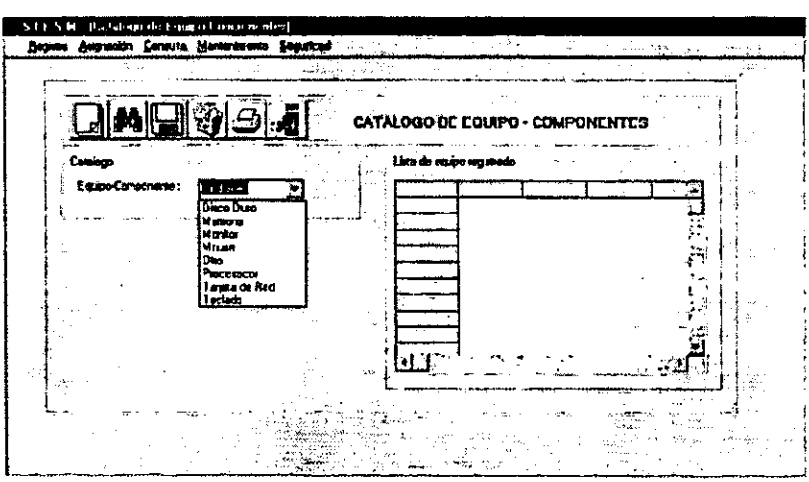
Ventana Catálogos



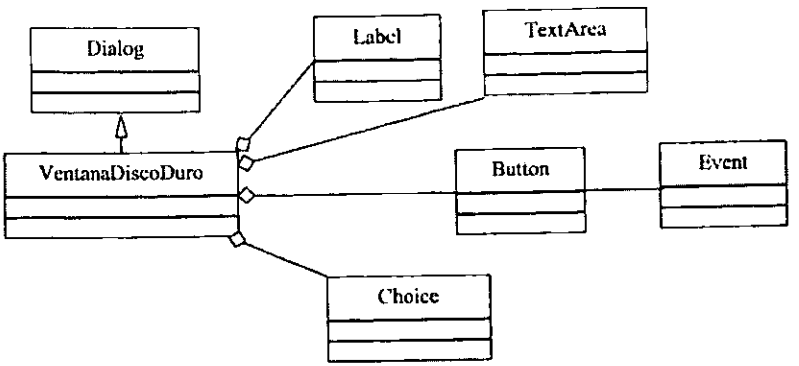


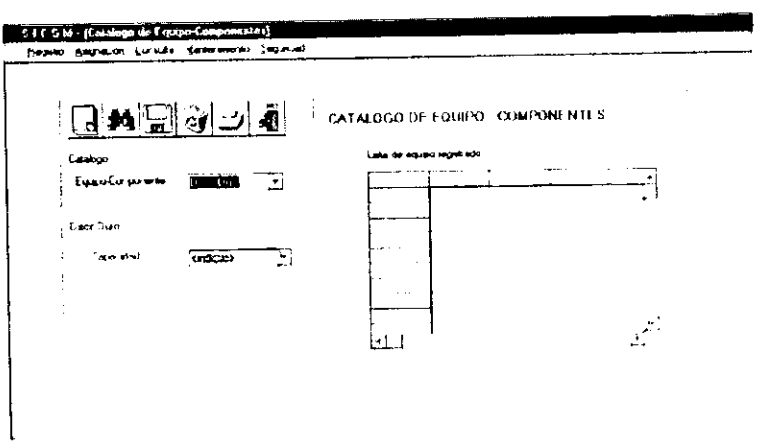
Ventana Equipo-Componente



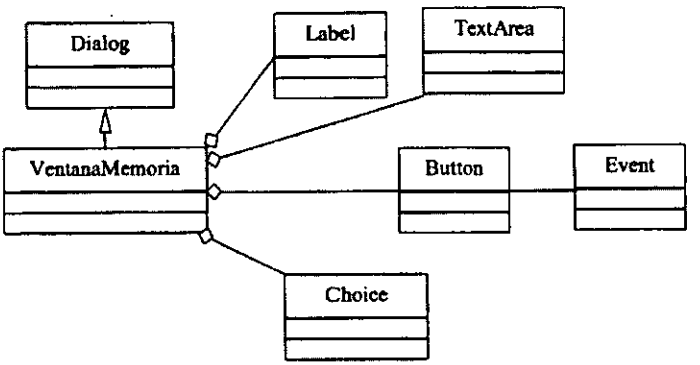


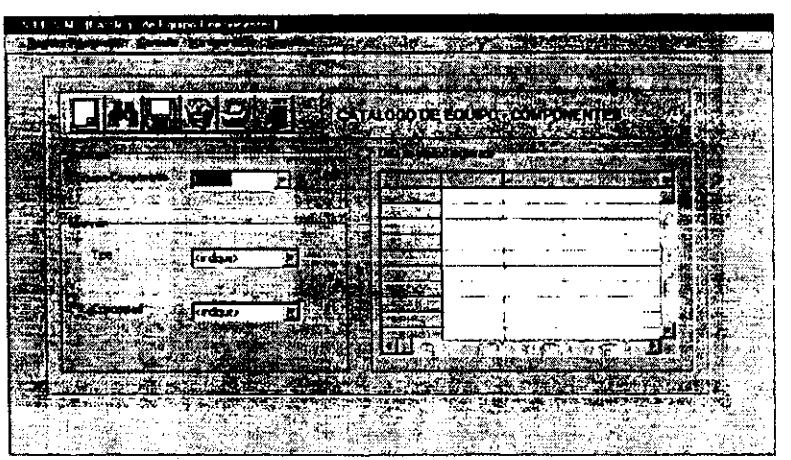
Ventana Equipo-Componente Disco Duro



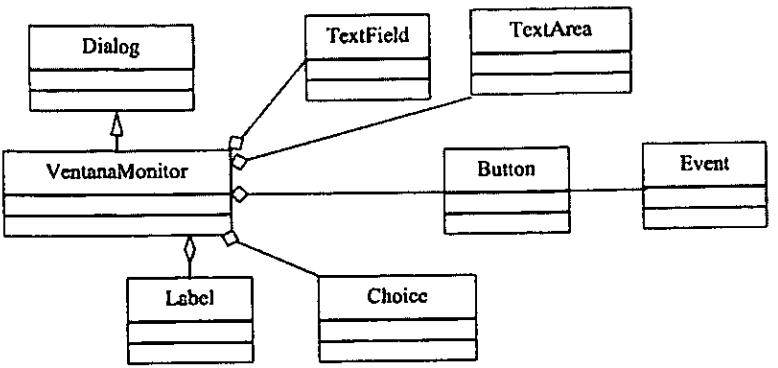


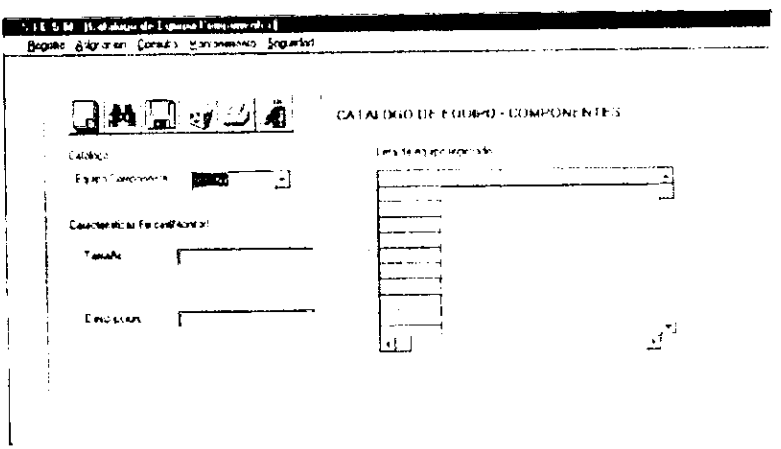
Ventana Equipo-Componente Memoria



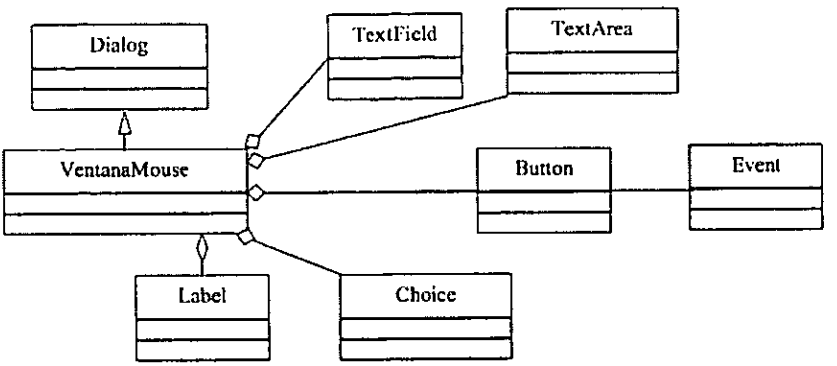


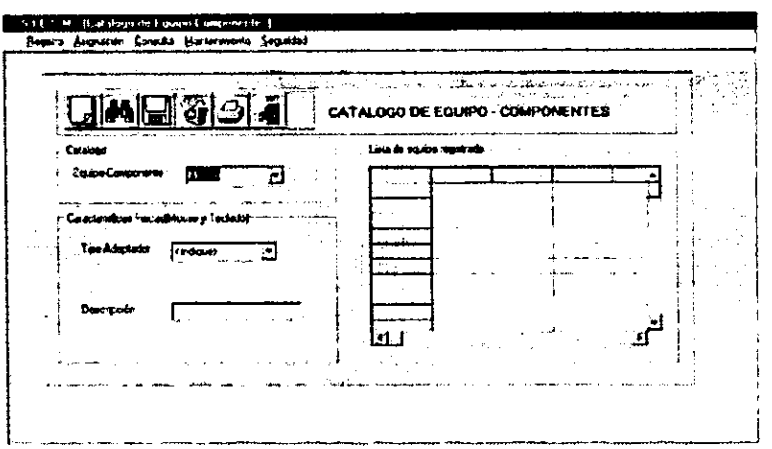
Ventana Equipo-Componente Monitor



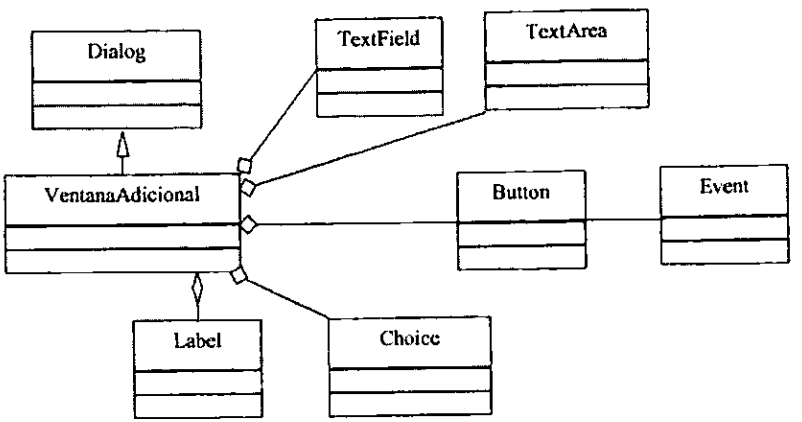


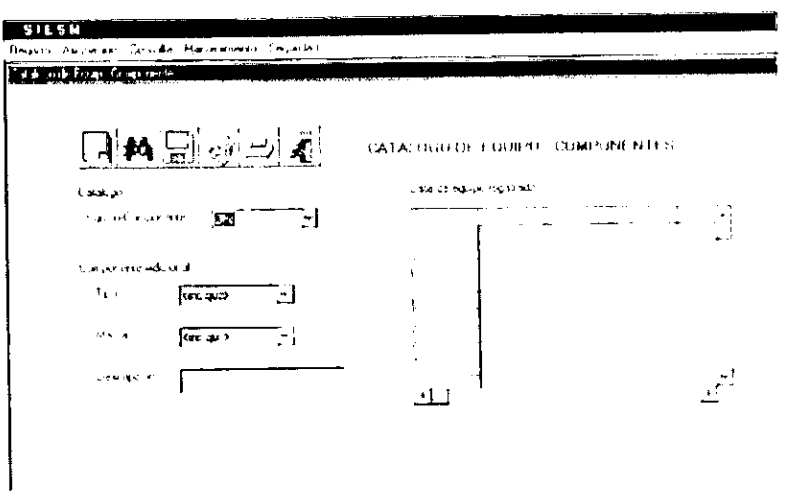
Ventana Equipo-Componente Mouse



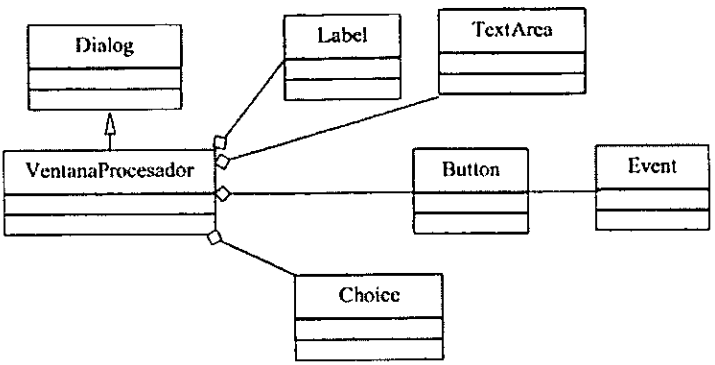


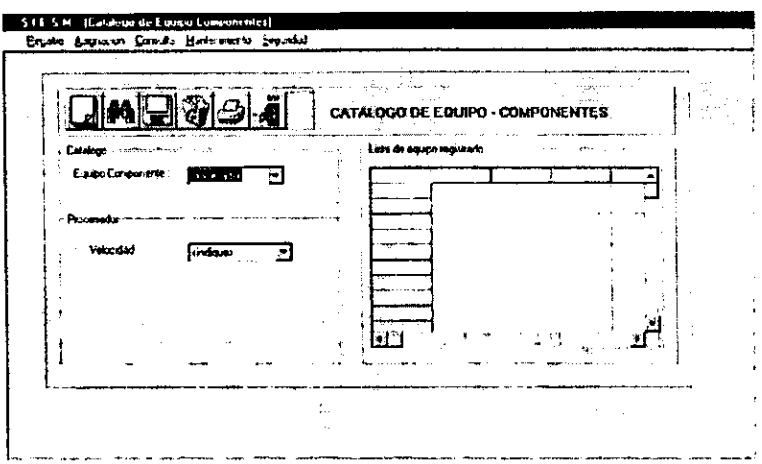
Ventana Equipo-Componente Adicional



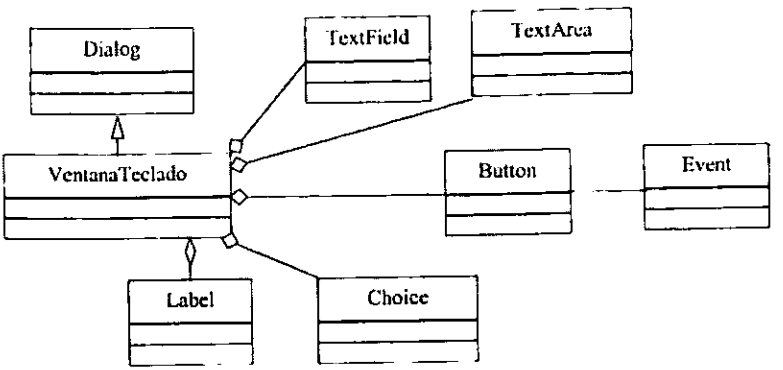


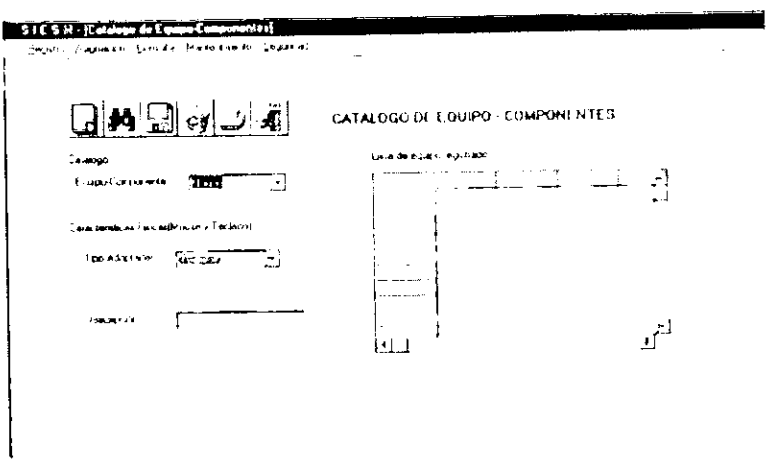
Ventana Equipo-Componente Procesador



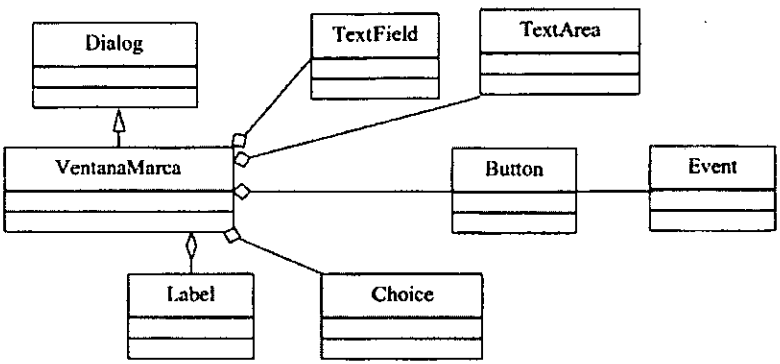


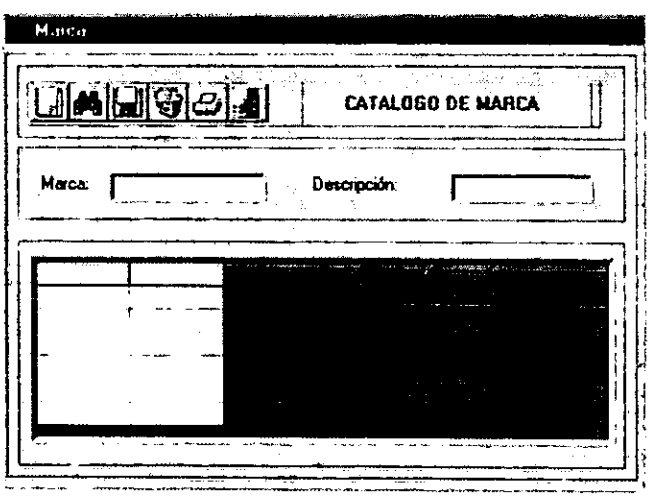
Ventana Equipo-Componente Teclado



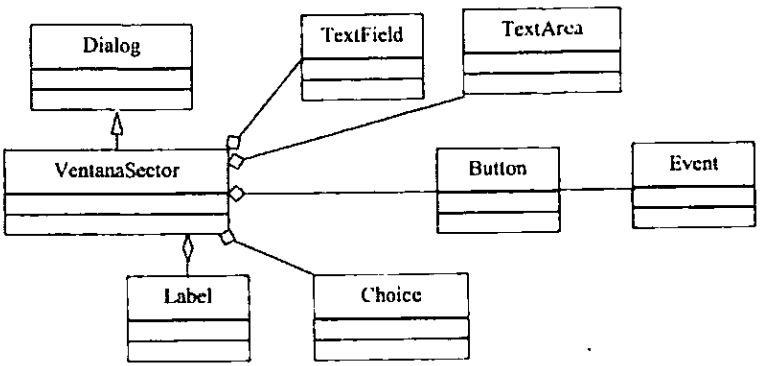


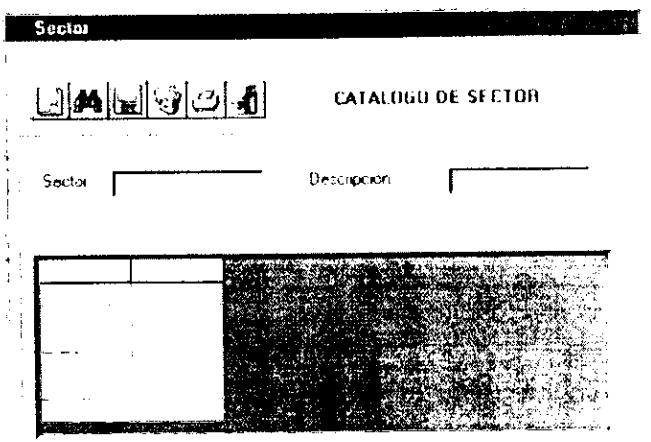
Ventana Marca



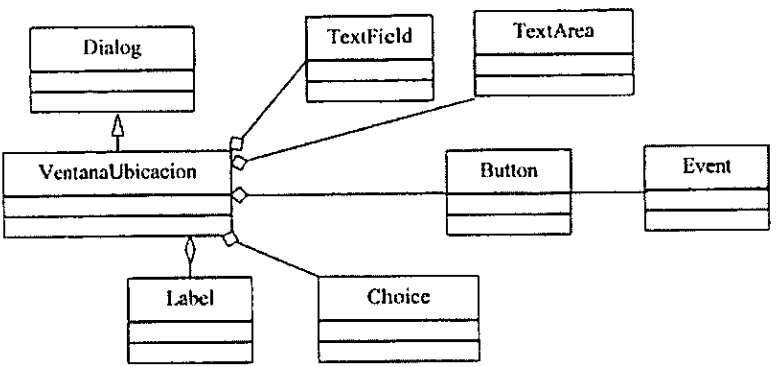


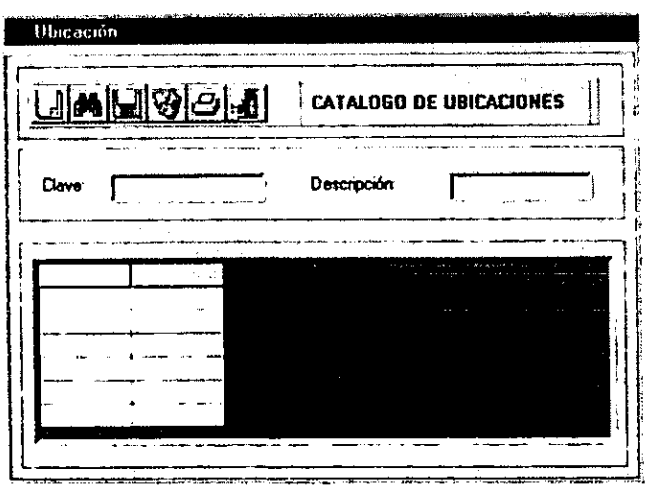
Ventana Sector



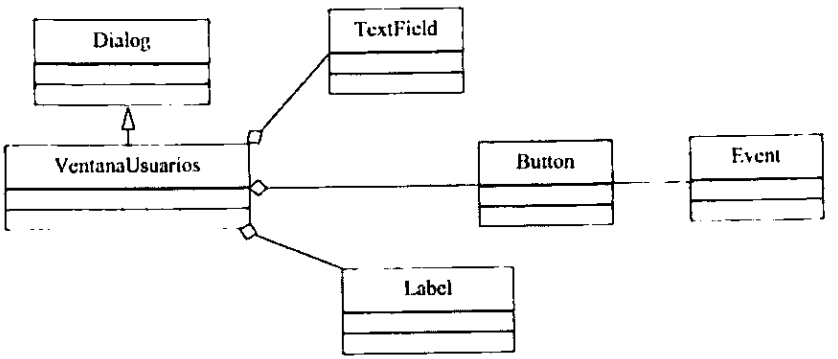


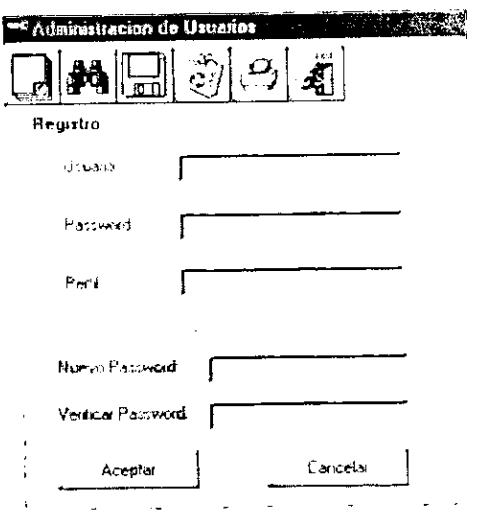
Ventana Ubicaciones





Ventana Administración de Usuarios





4.2.6. Especificación de reportes.

El diseño del sistema de información considera muy importante el modulo de reportes para facilitar el control en los procesos de asignación e inventario de equipo de cómputo, el cual puede ser visualizado gráficamente a través de la pantalla de Consulta, o bien llevado a papel para efectos correspondientes de cada uno de los tipos de reportes, el cual opera de la siguiente manera:

Cuando se carga la forma (pantalla Consulta) los criterios de selección (radio bottom) están todos habilitados y los parámetros de selección (combos) se encuentran deshabilitados. Una vez seleccionado el Radio Bottom del cual se requiere generar el gráfico se habilitará inmediatamente el combo quedando de esta forma los radio bottom no seleccionados también deshabilitados, si se desea cambiar el criterio de selección solo basta con hacer clic en el radio botton del tipo de reporte requerido y automáticamente la pantalla vuelve a adquirir la condición solicitada habilitando el combo del criterio seleccionado.

A continuación se dará una breve descripción de cada uno de los tipos de reportes que el sistema de información generará, mencionando los tipos de criterios de selección ubicados en los radio botton así como sus respectivos parámetros ubicados en su correspondiente combo.

Los criterios de selección así como sus parámetros correspondientes se agrupan de la siguiente manera:

Criterio de selección	Parámetros
Status	• Asignado • Baja • Disponible • Entregado • Mto. Preventivo • Mto. Correctivo • Mto. Sin garantía • Mto. Con garantía
Sector	• Financiero • Telecomunicaciones • Servicios • Soluciones
Procesador	• 200 Mhz • 266 Mhz • 300 Mhz Nota: en este rubro, de acuerdo al avance tecnológico se agregarán o eliminarán parámetros constantemente.
Pertenencia	• Softtek • Arrendado
Bitácora	• Mto. Correctivo • Mto. Sin garantía • Mto. Con garantía
Cve. Proy	• (Clave de proyectos)

Es importante mencionar que los reportes son generados de acuerdo a un criterio de selección con uno y solo uno de sus parámetros correspondientes.

4.2.7. Diseño de pruebas

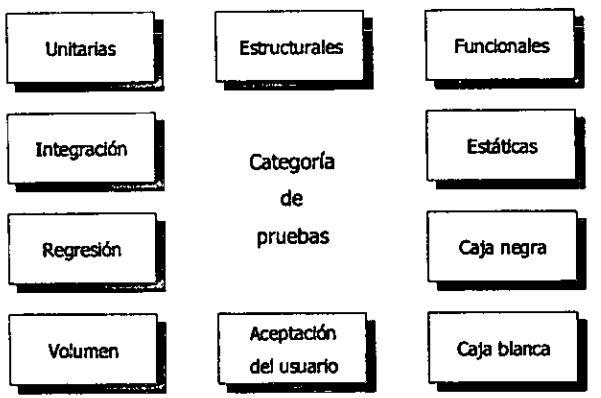
Las pruebas son parte integral y vital del ciclo de vida del desarrollo de sistemas. Se realizan con el propósito de encontrar fallas y se establecen para asegurar la calidad del sistema. Las pruebas requieren que se descarten las ideas acerca de lo correcto que es el software desarrollado y que al descubrir los errores, además de que se logre superar cualquier conflicto en el sistema. Pruebas que permitan establecer que el software puede funcionar correctamente. Según Pressman (1995) Prueba de software.

Las pruebas permiten:

- Determinar las bases para encontrar los objetivos y un plan específico de pruebas.
- Asegurar la obtención y formalización de los requerimientos del usuario y verificar que son adquiridos de una manera completa, correcta y consistente.
- Verificar los requerimientos operacionales y estructurales para establecerlos como fundamento para la realización de las pruebas.
- Buscar y registrar fallas o defectos asociados a los requerimientos establecidos.
- Documentar los reportes para las pruebas realizadas.

4.2.7.1. Categoría de pruebas

Las pruebas aplicables a un sistema de información se dividen en las categorías mostradas en la siguiente figura:



Utilización de métodos y técnicas para la implementación de pruebas a las que debe someterse el software en desarrollo Stamm (1981) Programa de pruebas.

Unitarias

Son las pruebas realizadas sobre un programa ú objeto con la finalidad de encontrar problemas operacionales en la lógica y problemas técnicos en el código. La prueba unitaria centra el proceso de verificación en la menor unidad del diseño del software (el objeto y sus métodos). Usando la descripción del diseño detallado como guía, se prueban los caminos posibles de control importantes, con el fin de descubrir errores dentro del ámbito del objeto y sus métodos.

La complejidad relativa de las pruebas y de los errores descubiertos esta limitada por el alcance estricto establecido por la prueba de unidad. La prueba de unidad siempre esta orientada a la caja blanca y este paso se puede llevar a cabo en paralelo para múltiples objetos con sus métodos. (Utilización de métodos y técnicas para la implementación de pruebas a las que debe someterse el software en desarrollo Stamm (1981) Programa de pruebas).

Integración

Las pruebas realizadas a un grupo de objetos son para asegurar que los mensajes sean pasados adecuadamente entre objetos. La prueba de integración es una técnica sistemática para construir la estructura de la aplicación mientras que, al mismo tiempo, se llevan a cabo para detectar errores asociados con la interacción, el objetivo es tomar los objetos probados en unidad y construir una estructura de aplicación que este de acuerdo con lo que dicta el diseño.

Regresión

Son pruebas selectivas para detectar fallas que se hayan introducido durante las modificaciones a un sistema o componente, que permitan verificar que estas modificaciones no impacten en forma negativa y que se siga cumpliendo con los requerimientos planteados.

Volumen

Son pruebas realizadas para verificar el comportamiento adecuado y eficiente de una aplicación bajo condiciones de volumen (número de operaciones), competencia de recursos (concurrencia) y carga máxima (velocidad de petición de ejecución de una operación) así como el comportamiento eficiente bajo las condiciones de volumen máximo (cantidad de datos) en las aplicaciones.

Aceptación del usuario

Son las pruebas finales ejecutadas por el usuario, para asegurar que el sistema satisfaga las necesidades de la organización o usuario final (validan que el sistema construido es el correcto).

Caja blanca

Son pruebas basadas en el conocimiento sobre la lógica y estructura internas. Usualmente dirigidas a la lógica.

Caja negra

Son pruebas funcionales basadas en los requerimientos sin conocimiento sobre cómo fue construido el sistema y usualmente dirigidas a los datos.

Estáticas

Consiste en la revisión y validación de los documentos generados en las distintas fases de la vida de un proyecto. Verificación realizada sin ejecutar el código del sistema.

Funcionales

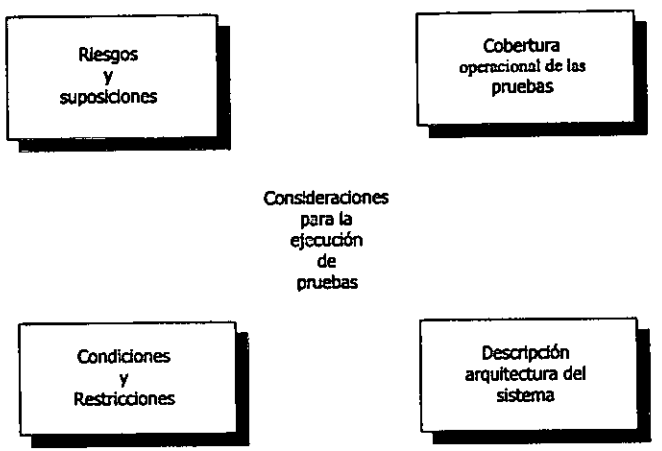
Validan los requerimientos de la organización (lo que se supone que el sistema debe hacer), pretenden descubrir errores cometidos en la implantación de dichos requerimientos.

Estructurales

Validan la arquitectura del sistema confirmando que todas sus partes funcionen sincronizadamente y que la tecnología esta siendo usada apropiadamente. Se refieren a las características técnicas, como su comportamiento con grandes volúmenes de información, tiempos de respuesta, etc.

4.2.7.2. Consideraciones importantes para la ejecución de pruebas

Para la ejecución de pruebas se toman en consideración los puntos mostrados en la siguiente figura:



Riesgos y suposiciones

Los riesgos son aquellos factores que pueden afectar negativamente la ejecución de las pruebas. Las suposiciones son las premisas que pueden afectar positiva o negativamente la ejecución de las pruebas complicando o facilitando las actividades de las mismas.

Condiciones y restricciones

Generalmente son limitaciones o problemas de naturaleza técnica y están relacionadas con el desarrollo del proyecto en sí, la tecnología de pruebas, el estado de los ambientes de pruebas, etc.

Cobertura operacional de las pruebas

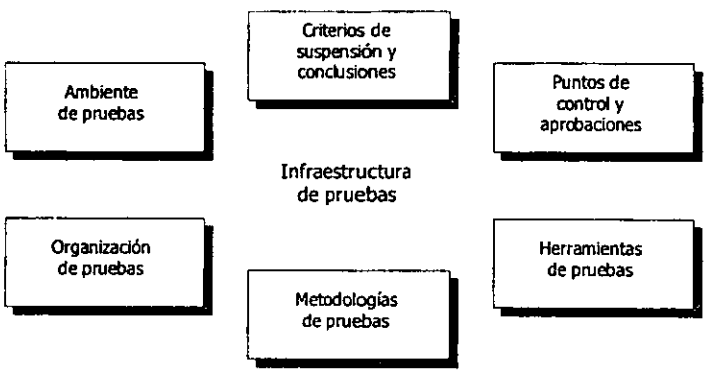
Dentro de la cobertura operacional de las pruebas se deben describir y listar de manera clara y concisa las operaciones a probar, así como aquellas operaciones a no ser probadas aún siendo parte del proyecto, ya que son necesarias especialmente cuando se requiere explicar el porqué de su exclusión, definiendo el alcance de las pruebas y delimitando responsabilidades. Además se debe documentar el ciclo del sistema a ser simulado con el objeto de ejecutar cada una de las operaciones objeto de las pruebas. Esta simulación suele ser realizada con muestras de datos fuera de especificaciones.

Descripción de la arquitectura del sistema

Para ello se consideran las especificaciones del software base sobre el cual esta construido el sistema tal como la plataforma, el software de base de datos, el sistema operativo, el lenguaje de programación, etc.

4.2.7.3.- Infraestructura de pruebas

A continuación se van a considerar los aspectos mostrados en la siguiente figura referentes a la infraestructura de pruebas:



Ambiente de pruebas

Se identifican los ambientes donde se ejecutan las pruebas, así como las características generales de los datos de prueba (qué datos necesitan y cómo se obtendrán). Tomando como base el modelo de datos del proyecto. Es importante saber cuántos y cuáles datos serán seleccionados, y para la estimación de la carga de trabajo necesaria para generarlos.

Organización de las pruebas

Se define la organización (puestos y responsabilidades) que es requerida para la construcción y ejecución de las pruebas.

Metodología de las pruebas

Es importante determinar si existe un procedimiento de pruebas dentro de la organización o si es necesario elaborarlo, y en qué medida este procedimiento está integrado con el resto de los sistemas de desarrollo y mantenimiento.

Herramientas de pruebas

Se identifican los productos a utilizar y el uso específico que se hace con ellos. Se debe determinar si es preciso vigilar todos los componentes o solamente algunos, el interés de vigilar a determinados componentes es justificado por la necesidad de verificar cuál es el comportamiento interno de dichos componentes, esto es, como se realiza el procesamiento de la información. En el caso de estar solamente interesados en las entradas y salidas de los procesos, es suficiente muchas veces el verificar estas entradas y salidas sin tener en cuenta exactamente cómo se leen y cómo se generan.

Puntos de control y aprobaciones

Se deben especificar los puntos de control en el transcurso de la construcción y ejecución de las pruebas, tal como el determinar los puestos de las personas que tendrán que autorizar la continuación de las pruebas acorde con el plan original o asumiendo las variaciones incorporadas del mismo.

Criterios de suspensión y conclusión de las pruebas

Estos criterios se refieren a la suspensión o terminación de la ejecución de los casos de prueba cuando son necesarios otros componentes que no tienen listos, o cuando el número de los defectos encontrados sobrepasa el límite de los esperados, para lo cual es necesario regresar a la etapa de desarrollo y verificar las especificaciones.

4.3. Desarrollo del Sistema

El desarrollo del sistema quedará pendiente, dado que por sus características, plataformas de operación (Java, Oracle y Solaris) y funcionamiento no es posible su implementación durante el desarrollo de este trabajo, además de que el requerimiento por parte de la empresa Softtek México sólo comprende el análisis y diseño, por lo que sólo se plantea un prototipo viable para su posterior desarrollo e implementación del sistema.

Para poder tener un aseguramiento en la calidad del software que se plantea es necesario contar con una buena planeación, organización y preparación de la información, partiendo de la norma IEEE estándar 730 y 983 . Gestión. Referida a organización y asignación de tareas y responsabilidades. Stamm (1994) Definición de requerimientos.

Análisis de desarrollo

El alcance funcional del sistema se ha planteado es los temas anteriores tomando en cuenta las normas de calidad norma IEEE estándar 730 y 983. Gestión. Referida a organización y asignación de tareas y responsabilidades, Stamm (1994) Definición de requerimientos; podemos obtener la siguiente información:

Tiempo estimado de realización del proyecto:
Según el alcance funcional se estima un total de 1424 horas de esfuerzo.

Costo del desarrollo del sistema:

El esfuerzo requerido para desarrollar el sistema de inventario sería:

A	Líder de Proyecto	13.0 semanas – 480 horas hombre
B	Analista Diseñador	12.0 semanas – 416 horas hombre
C	Programador	11.0 semanas – 400 horas hombre

Esfuerzo total de desarrollo **1424 horas**

La duración del desarrollo se estima en 8 semanas con la siguiente calendarización:

		Semanas												
Producto	Responsable	1	2	3	4	5	6	7	8	9	10	11	12	13
Administración del Proyecto	Líder													
Análisis	Líder													
Infraestructura	Analista Diseñador													
Programación	Programador													
Pruebas Integrales	Analista Diseñador													
Capacitación	Analista Diseñador													
Documentación	Programador													

La inversión recurrida para el desarrollo del sistema es la siguiente:

Nivel	Tarifa/Hora	Horas Estimadas	Costo
A	264 (45%)	520	$ 61 776.00
B	214 (40%)	464	$ 55 123.00
C	380 (35%)	440	$ 52 272.00
Total		**1424**	**$ 169 171.00**

Esta tarifa de referencia está basada en una utilidad cero, ya que se maneja como un proyecto interno. La fórmula de donde se derivan estos costos no se incluyó dentro de este trabajo por razones de confidencialidad de la empresa.

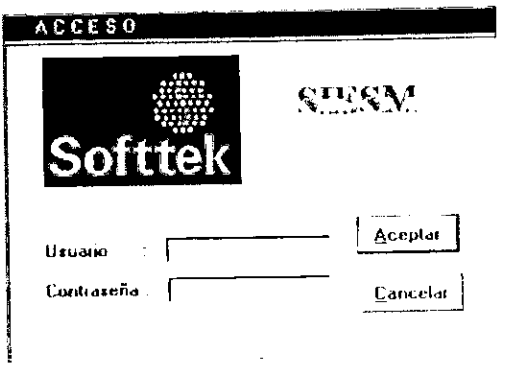
4.3.1. Prototipo

La elaboración de un prototipo de un sistema de información es una técnica valiosa para la recopilación rápida de la información específica acerca de los requerimientos de información de los usuarios.

El prototipo utilizado en este caso es conocido como *Prototipo No Operacional*, que consiste en un modelo no funcional con el objeto de probar determinados aspectos del diseño, este prototipo del sistema de información se aplica debido a que la codificación del sistema es muy amplia, pero sirve para obtener una idea útil del sistema.

En el prototipo siguiente se plantean los módulos principales del sistema donde se detallan las características y funciones principales de las pantallas que contaran con una medida de 800 X 600 píxeles, contando también con pantallas autoajustables dependiendo de las características de la pantalla del monitor. Según IEEE estándar 730 y 983 Estándares, prácticas y convenciones, Buttler Estándares y convenciones e ISO 9000-3 diseñó e implementación; así como la secuencia de las mismas en el sistema.

Módulo	Acceso
Página	Pantalla de acceso



Objetivo:

Permite el acceso al Sistema de Inventarios de Equipo Softtek México.

Especificación Funcional

Esta pantalla es invocada en el momento que se seleccione del menú de Programas la aplicación SIESM.

La pantalla de Acceso se mostrará en el centro y aplicará solamente cuando se desee ingresar al sistema SIESM.

Se desplegarán unos campos tales como el campo de Usuario: que se refiere al personal autorizado que tiene acceso al sistema y el campo de Password que se refiere a la contraseña válida para acceder al sistema.

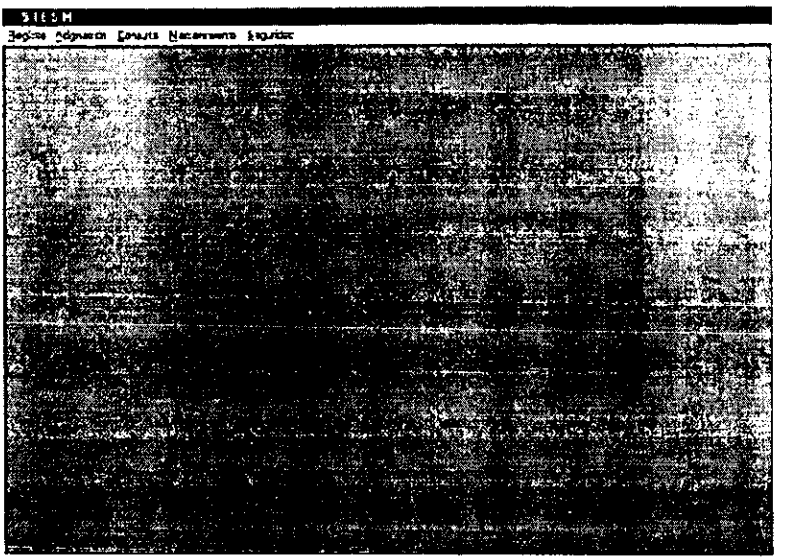
Con todos los datos anteriores se realizará un procedimiento de armado de acceso, que consiste en realizar una conexión a la base de datos y ejecutar el Query formado con todos los datos anteriores para verificar que el usuario se encuentra registrado en el sistema.

Si cualquiera de los datos que se proporcionen no es correcto y se selecciona el botón de Aceptar, inmediatamente se desplegará un mensaje de error indicando "error con la base de datos" y no será proporcionado el acceso al sistema.

En el caso de que los datos proporcionados sean los adecuados y se oprima el botón Aceptar inmediatamente se permitirá el acceso al sistema y se desplegará la pantalla principal del sistema SIESM.

Al seleccionar el Botón de Cancelar, automáticamente se elige no trabajar con el sistema SIESM. Todas las pantallas contaran con el nombre de la pantalla de que se trate en la parte superior izquierda, después aparecerá un menú con las diferentes opciones para cada pantalla en particular. Según IEEE estándar 730 y 983 Estándares, prácticas y convenciones, Buttler Estándares y convenciones e ISO 9000-3 Diseñó e implementación, Acis. Documentación del software.

Módulo	Principal
Página	Pantalla del Menú Principal



Objetivo:

Presentar las diferentes opciones que se tienen en el Sistema de Inventario de Equipo Softtek México.

Especificación Funcional

Esta pantalla es invocada después de la pantalla de Acceso al sistema SIESM.
La pantalla muestra un menú con las siguientes opciones:

- Registro
- Equipo
- Salir
- Asignación
  Solicitud
- Consulta
- Estadistica
- Mantenimiento
  Bitácora
- Dar Mantenimiento
- Mantenimiento Realizado
- Consultas
- Catálogo
- Seguridad
  - Administración de Usuarios.

Registro

- Equipo

Cuando se selecciona la opción de Registro-Equipo es invocada la pantalla de Facturas donde se registrarán todos los datos contenidos en la factura.

- Salir

Esta opción permite salir totalmente de la aplicación SIESM.

Asignación

Al seleccionar esta opción se muestra la pantalla Asignación en la cual se podrán realizar: altas, bajas, modificaciones y consultas de la asignación del equipo asignado a una solicitud.

Consulta

Cuando se selecciona la opción de Consulta-Estadísticas es invocada la pantalla de Estadísticas. Aquí se podrá consultar de manera gráfica la Bitácora de Mantenimiento de los diferentes componentes registrados en el sistema.

Mantenimiento.

- Bitácora

Alta

Al seleccionar esta opción se introduce un equipo a la bitácora para que lleve su respectivo seguimiento

Mantenimiento realizado

Al elegir esta opción el equipo es dado de baja en la bitácora de mantenimiento.

Consultas

Al elegir esta opción se muestran los equipos que están en la bitácora de mantenimiento; con diversos criterios de selección.

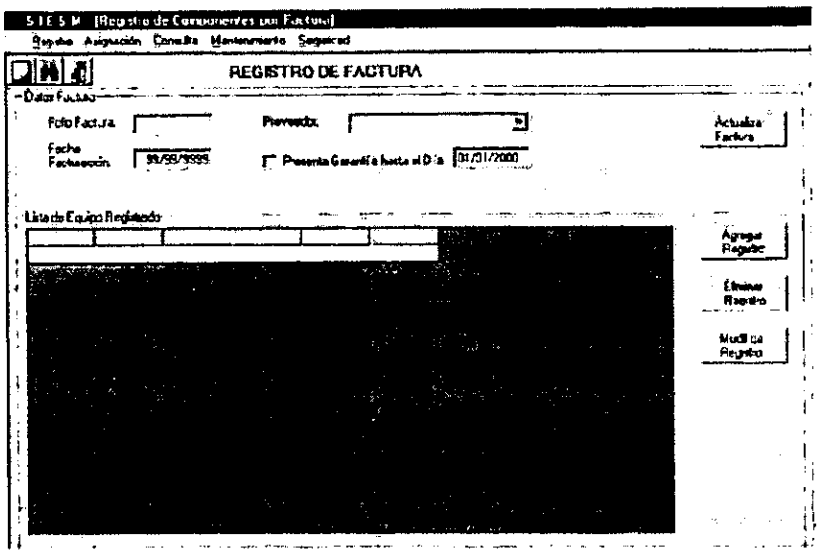
- **Catálogos**

Cuando se selecciona la opción de Mantenimiento-Catálogos es invocada la pantalla de Catálogos. Aquí se podrán llevar a cabo los procedimientos de Altas, Bajas y Modificaciones de todos los catálogos de Sector, Marca y Ubicación teniendo una liga a la pantalla de Catálogo de equipo-componente

Seguridad

Cuando se selecciona la opción de Seguridad-Administración de usuarios es invocada la pantalla de Administración de Usuarios. Aquí se podrá llevar a cabo las Altas, Bajas, Modificación y Consulta de los usuarios autorizados para acceder el sistema SIESM. Todos los datos forman el menú de acceso al sistema SIESM, se podrá elegir cualquiera de las opciones anteriormente descritas y en base a lo elegido se presentara la pantalla respectiva. Las pantallas contarán con botones para realizar las operaciones más comunes como son nuevo, buscar, imprimir, salir, agregar, borrar, los datos que llevaran serán datos generales, y datos específicos que dependerán de la tarea que se este realizando en cada pantalla. Según IEEE estándar 730 y 983 Estándares, prácticas y convenciones Buttler Estándares y convenciones e ISO 9000-3 Diseñó e implementación, Acis Documentación del software.

Módulo	Registro
Página	Registro y Consulta por Factura/Equipo registrado



Objetivo:

Realizar el proceso de registro facturas con el desplegado de detalle de los componentes adquiridos.

Especificación Funcional

Esta pantalla es invocada en el momento que se seleccione del menú principal la opción Factura.

Esta pantalla tiene la posibilidad de actualizar, agregar y eliminar una factura con todo el equipo registrado, así mismo, se puede agregar o quitar equipo que se encuentre registrado en esta factura, siempre y cuando no se encuentre registrado este equipo.

La pantalla esta dividida en dos partes, "Datos de la factura" y "Lista de Equipo Registrado"

En la parte de "Datos de la factura" se muestra información clave de la factura, la cual se registra en una nueva factura, consultar una factura y actualizar los datos de la factura.

En la parte de "Lista de Equipo Registrado" se muestra un formato con los campos de:

- Código Softtek
- Descripción (CPU, Mouse, Teclado, Monitor)
- No. Serie
- Precio

La funcionalidad del Grid de "Lista de Equipo Registrado" es mostrar el equipo que en ese momento se encuentra registrado para la factura.

Esta lista puede ser incrementada o decrementada, según la opción que se elija en la selección de los botones que están a su derecha ("Agregar Registro", "Eliminar Registro", "Modificar Registro").

El botón de "Agregar Registro" llama a la forma "Registro Equipo", enviando los datos de la factura, para dar de alta un nuevo componente a la factura

El botón de "Eliminar Registro", remueve un elemento de la lista y de la Base de Datos para esta factura

El botón de "Modificar Registro" llama a la forma "Registro Equipo", donde se puede apreciar el detalle del equipo que se registro, y modificarlo si se desea.

Los títulos de las columnas del Grid de la lista de equipo registrado, el cual es indicado a continuación, en todas las pantallas para facilitar el funcionamiento para el usuario se contara con botones de selección y grids para consultas, selección e introducción de datos. Según IEEE estándar 730 y 983 Estándares, prácticas y convenciones, Buttler Estándares y convenciones e ISO 9000-3 Diseñó e implementación, Acis. Documentación del software.

Para la 1ª columna: Código Softtek.
Para la 2ª columna: Descripción (puede ser: CPU, Teclado, Mouse, Monitor)
Para la 3ª columna: No de Serie.
Para la 3ª columna: Precio.

Proceso para agregar una nueva factura.

Las cajas de texto, deben de estar sin información, así como, la lista en el grid es limpiada (sin ningún registro), y esperara para ingresar los datos principales de la factura, o nota de compra.

Para comodidad del usuario, la caja de texto para la fecha de facturación tomara el valor de la fecha del sistema con el formato en formato "DD/MM/AAAA".

Una vez ingresado los datos de la factura, permite agregar el equipo adquirido.

Una vez que se termine de agregar el equipo a la tabla FACTURA, se actualizará el grid, leyendo de la base de datos la tabla de EQUIPO.

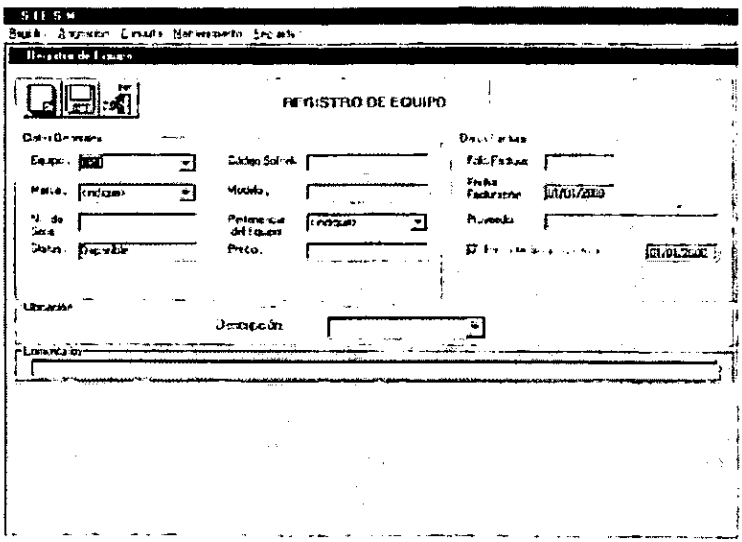
Proceso para consultar y modificar una factura

Existen dos formas de modificar los datos de una factura:
La primera es: Una vez ingresados los datos de la factura, se solicita que se busquen los datos relacionados con esa factura, oprimiendo el botón de buscar del menú, e inmediatamente mostrará la información correspondiente a la factura y el equipo registrado a esta factura.

En el caso de que se cambien datos de la factura, oprima el botón refrescar factura y se actualizaran los datos en la tabla respectiva..

La otra forma de modificar o actualizar una factura es eligiendo del grid un equipo registrado y se desee modificar, una vez elegido se selecciona el botón Modificar y aquí se podrán realizar los cambios correspondientes.

Módulo	Registro
Página	Registro, Consulta y Modificaciones de equipo por factura



Objetivo:

Realizar el proceso de registro de equipo para una factura.

Especificación Funcional

Esta pantalla es invocada en el momento que se seleccione de la pantalla de Registro de facturación "Agregar Registro" o "Modificar Registro" equipo, del botón de la pantalla previa.

Esta pantalla tiene la posibilidad de agregar equipo a una factura indicada, así como mostrar el equipo de una factura y modificar sus características.

La pantalla esta dividida en: "Datos generales", "Datos de la factura", "Ubicación" "Comentarios", "Especificación del equipo" y "Características de CPU".

Al inicio se muestra en la forma la parte superior que corresponde a los Datos generales, Datos de la factura, ubicación y los comentarios.

Una vez capturados todos los campos, se despliega la parte inferior que muestra las especificaciones del equipo seleccionado, y si se trata de un CPU entonces despliega las características propias del CPU

En el frame de "Datos Generales" se captura la información referente al equipo que se está registrando, "Código Softtek", "Marca", "Modelo", "Numero de serie", "Pertenencia ", "Status" y "Precio del equipo".

En la parte de "Datos de la factura" se muestra información clave de la factura, siendo esta: Folio de la factura, fecha de facturación, proveedor y fecha de garantía (si es que la tiene), la cual hace referencia al equipo que se está registrando.

En la parte de "Comentarios" se captura algún comentario adicional del equipo.

En la parte de "Especificación de equipo diferente" se capturan características físicas del equipo como pueden ser:

- Tipo de conector (aplica para CPU, mouse, y teclado)
- Tamaño (Aplica para Monitor)
- Tipo de memoria (aplica para CPU)
- Velocidad del procesador y memoria (aplica para CPU)
- Capacidad de Disco Duro (aplica para CPU)
- Tarjeta de RED (aplica para CPU)

Nota: Estas características aparecen o desaparecen dependiendo del equipo que aplique.

La funcionalidad del TreeView es mostrar todos los componentes con los que cuenta el CPU, que pueden ser incrementados o decrementados con los botones de ">>" o "<<" respectivamente.

El TreeView tiene los siguientes rubros (o nodos) indicados a continuación:

- CPU
- Procesador
- Memoria
- Disco Duro
- Tarjeta de RED
- Conectores
- Otros

Proceso para Registrar un equipo.

Seleccione del menú "nuevo", y la forma mostrará los campos habilitados listos para la captura de información.

Mostrándose únicamente la parte superior de la forma.

Al seleccionar el equipo que se va a registrar, se debe acceder a la tabla Componente_CPU ya que de ahí se toma la información para mostrarse en el combo respectivo.

La parte superior de la forma siempre se mostrará, para cualquier equipo, la parte inferior variará según sea el equipo elegido.

El código Softtek será tecleado por el usuario y guardado en la tabla EQUIPO. Para el combo que hace referencia a la marca se tiene que acceder a la tabla de MARCA El usuario debe teclear el Modelo del equipo que esta registrando, este dato se inserta en la taba EQUIPO.

El número de serie es tecleado por el usuario y registrado en la tabla EQUIPO. Para obtener la pertenencia del equipo se debe acceder a la tabla EQUIPO. Para obtener la información del combo de status se debe acceder a la tabla STATUS.

El precio será capturado por el usuario y esa información debe insertarse en la tabla de EQUIPO.

En el área de comentarios se recibe la información que el usuario crea necesaria, y dicha información se concentra en la tabla EQUIPO.

Si en equipo se eligió un Mouse o Teclado:

En la parte inferior del lado izquierdo se muestra un frame con las características físicas de teclado y mouse, esto es, el tipo de adaptador; que para obtenerlo se debe de acceder y tomar la información.

Si en equipo se eligió un CPU:

Si es necesario sólo se pueden llenar los campos de la parte superior de la forma, al elegir grabar todos estos datos se insertan en las tablas correspondientes de la base de datos.

Proceso para agregar un componente al CPU (aplica solo a CPU)

En la parte inferior del lado derecho se muestra un árbol como el siguiente:

CPU
- Procesador
- Memoria
- Disco Duro
- Tarjeta de Red
- Conector de Mouse
- Otros

En el que se pueden elegir:

Procesador. Si se elige esta opción se muestra del lado izquierdo un frame con la velocidad del procesador, contenida en un combo box.

Si se elige memoria el frame despliega dos combos, tipo y capacidad.

Si se elige disco duro en el frame del lado izquierdo se despliega un combo con las capacidades de disco duro.

Si la elección es en tarjeta de red, se despliega un combo box con los diferentes tipos de tarjetas.

Si se selecciona la opción otros, entonces el frame de la parte izquierda contará con cuatro campos; Equipo y Marca, el Número de serie y los comentarios, son datos que podrá teclear el usuario.

Al elegir cada característica del CPU (si es que la tiene) y seleccionar el botón agregar ">>", en el frame de Características del CPU se agrega cada una de estas y se insertan abajo de la rama que deban de ir, ejemplo: velocidad (500 Mhz) se coloca después de procesador.

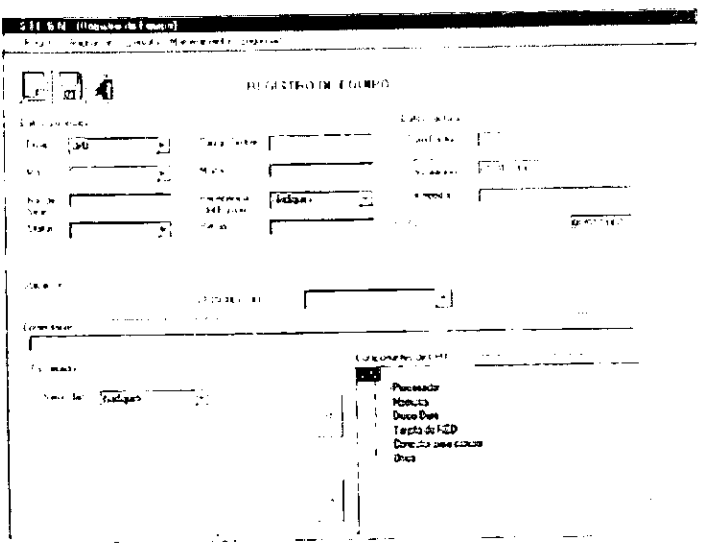
CPU
- Procesador 500 Mhz
- Memoria
- Disco Duro
- Tarjeta de Red
- Conector para mouse
- Otros

Cuando toda la información este completa, se selecciona la opción del menú "Grabar" y en este momento será grabado en las respectivas tablas de la base de datos.

Proceso para eliminar un componente agregado al CPU (solo aplica para CPU).

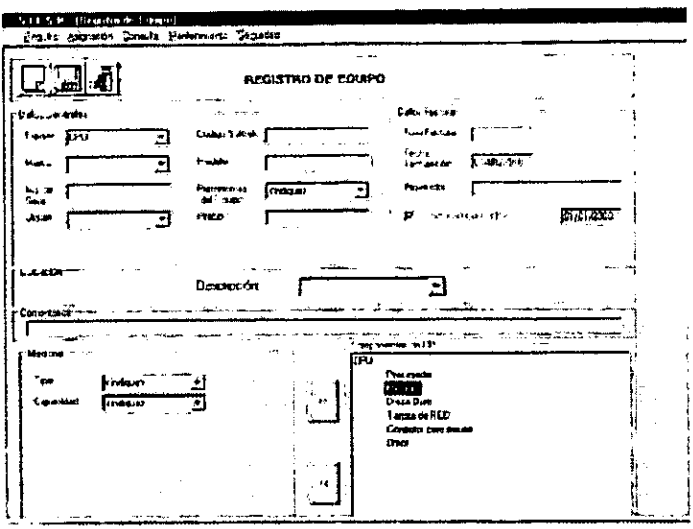
Si el usuario desea eliminar un componente del TreeView; necesita oprimir el botón "<<", y se removerá el componente de la lista para este CPU.

Al oprimir el botón grabar para actualizar el cambio.

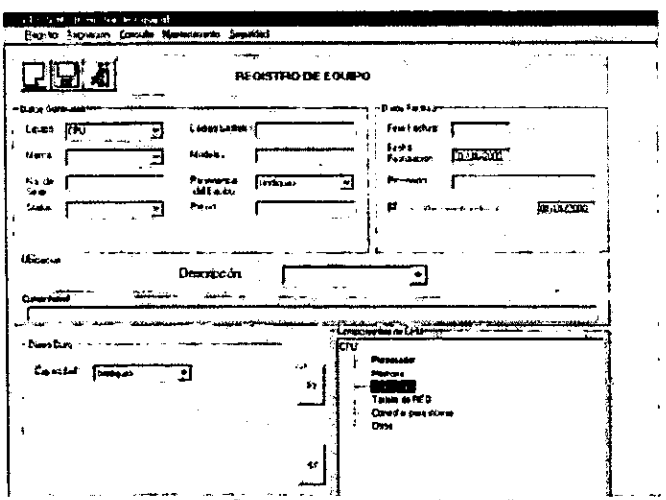


En las ventas que sean necesarias se contara con la opción de TreeView para hacer más fácil la selección de componentes. Según IEEE estándar 730 y 983 Estándares, prácticas y convenciones, Buttler Estándares y convenciones e ISO 9000-3 Diseñó e implementación, Acis. Documentación del software.

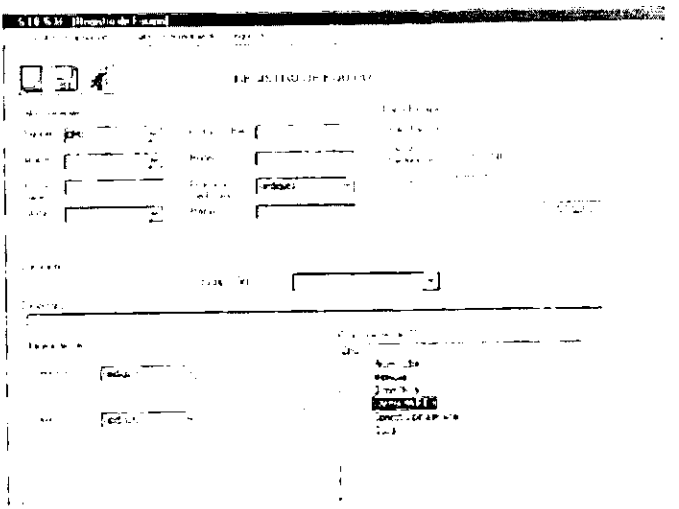
Al momento de elegir opción del combo Equipo, se desplegaran los diferentes tipos de equipo registrados.



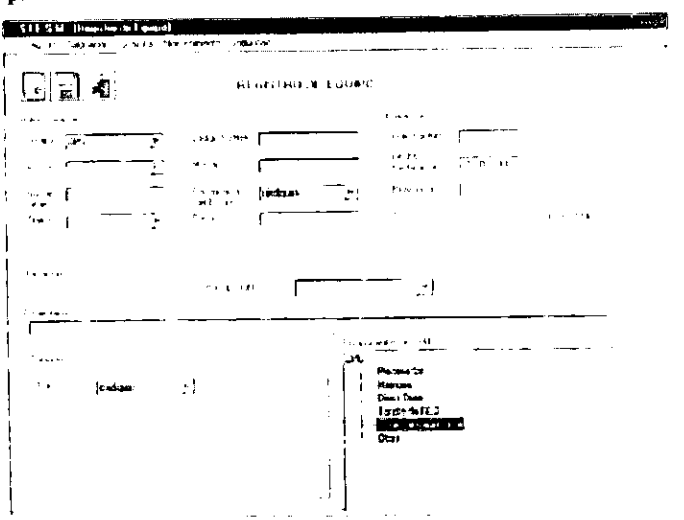
Al momento de seleccionar la opción de Memoria, se mostrarán dos combos: Tipo y Velocidad, para que el usuario elija la opción con las características correspondientes.



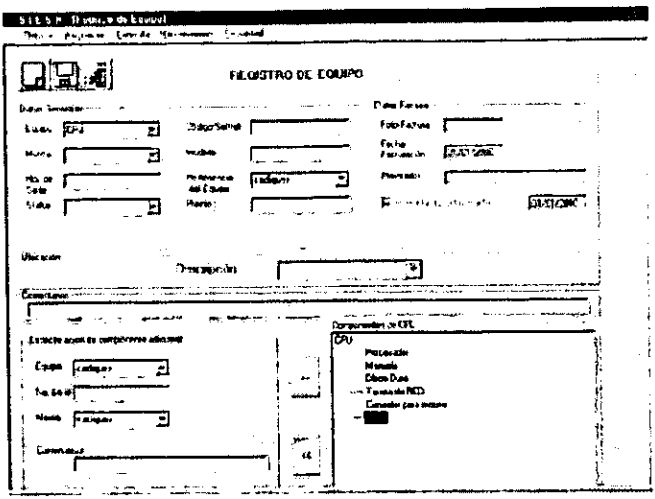
Al momento de seleccionar la opción de Disco Duro, se mostrará el combo de Capacidad, para que el usuario elija la opción correspondiente.



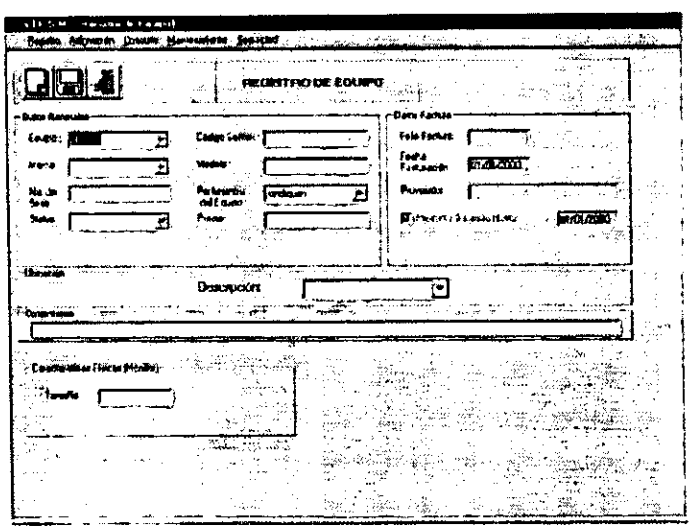
Al momento de seleccionar la opción de Tarjeta de Red, se mostrará el combo de Tarjeta de Red, para que el usuario elija la opción correspondiente.



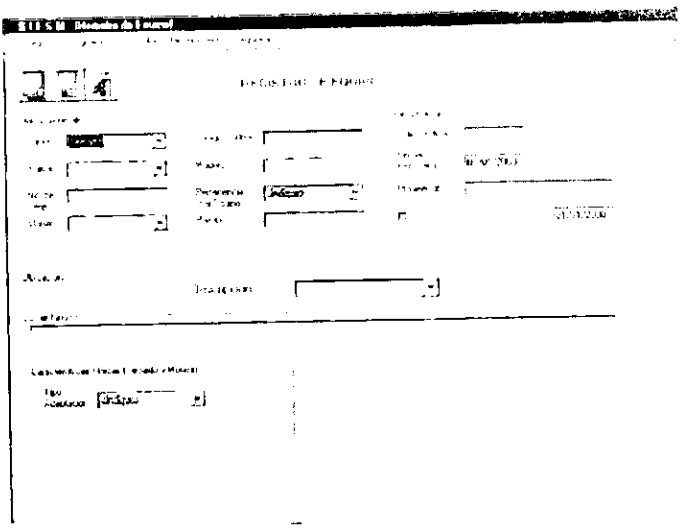
Al momento de seleccionar la opción de Mouse, se mostrará el combo de Tipo, para que el usuario elija la opción correspondiente.



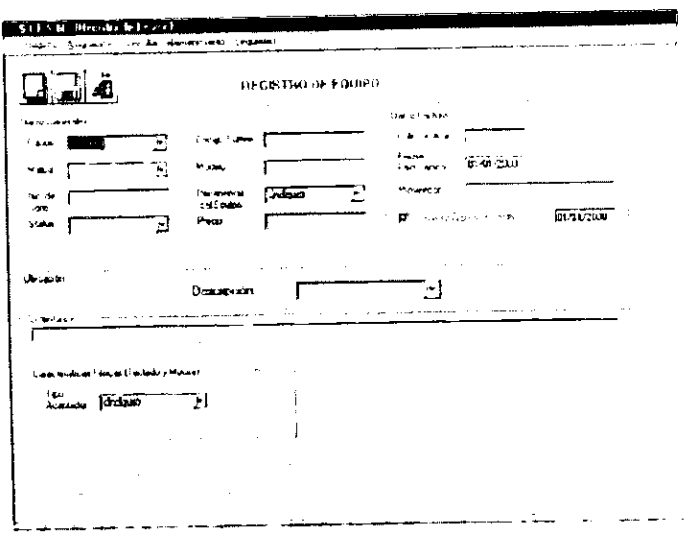
Al momento de seleccionar la opción de Otro, se mostrarán 2 combos: Equipo y Marca, para que el usuario elija la opción correspondiente, además de dos cuadros de texto donde se podrá ingresar el numero de serie del equipo así como un comentario respecto al mismo.



Al momento de seleccionar la opción de Monitor, se mostrará un cuadro de texto para que el usuario ingrese el tamaño del monitor en pulgadas.

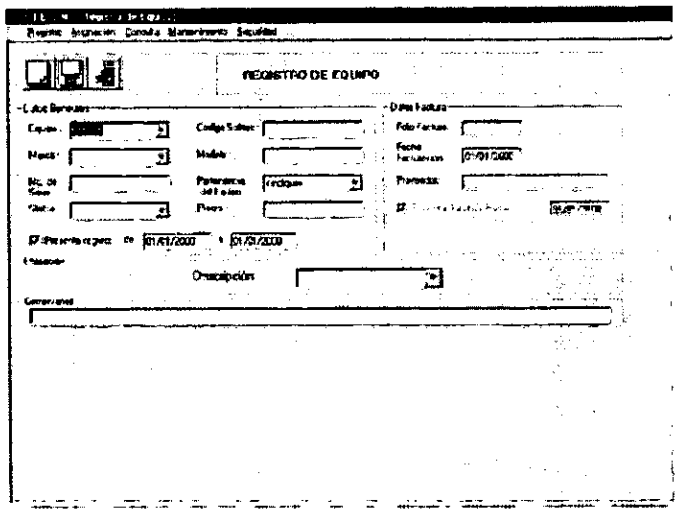


Al momento de seleccionar la opción de Teclado, se mostrará un combo para que el usuario elija el Tipo de Adaptador que corresponda.



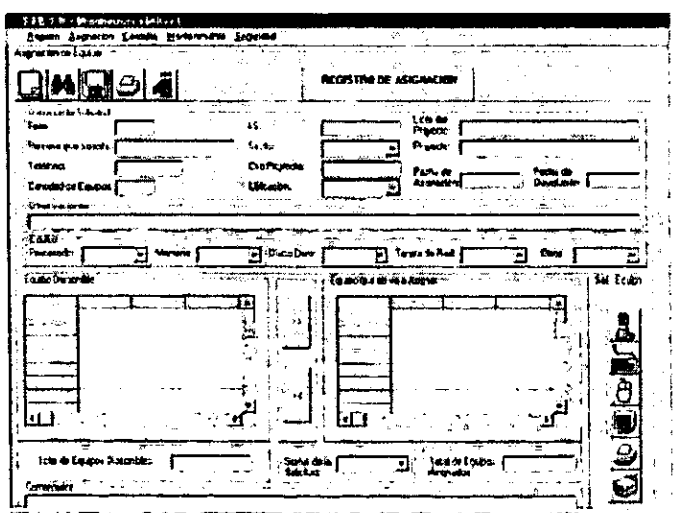
Al momento de seleccionar la opción de Mouse, se mostrará un combo para que el usuario elija el Tipo de Adaptador que corresponda.

También contarán con un campo de comentarios donde, donde se pondrá poner alguna observación si es que es necesaria. Según IEEE estándar 730 y 983 Estándares, prácticas y convenciones, Buttler Estándares y convenciones e ISO 9000-3 Diseñó e implementación, Acis. Documentación del software.



Al momento de seleccionar la opción de Lap-top, se mostrará un checkbox para que el usuario indique si la Lap-top está asegurada. En caso de estar asegurada se deberá indicar la vigencia del seguro.

Módulo	Asignación
Página	Asignaciones Altas, Bajas, Modificaciones y Consultas



Objetivo:

Realizar los procedimientos de Asignación de equipo de Cómputo para llevar los datos Administrativos con un mejor control

Especificación Funcional

Esta pantalla es invocada en el momento que se seleccione del menú la opción Asignación-Solicitud

La pantalla muestra un formato que dado el número de Folio el cual es igual al número de Folio de la ASP, por lo que se llenaran automáticamente los campos de:

- IS
- Lider de Proyecto
- Persona que solicita
- Sector
- Proyecto
- Teléfono
- Ubicación Interna o Externa
- Piso
- Oficina
- Cantidad de equipos
- Otros
- Fecha de Asignación
- Fecha de Devolución
- Procesador
- Memoria
- Disco Duro
- Tarjeta de Red

Proceso de Altas

Todos los datos anteriores son cargados automáticamente en base a una conexión con la tabla de Solicitudes de Help Desk que contiene todos estos datos.

Los campos de:

- Fecha de Asignación
- Fecha de Devolución
- Sector
- Piso
- Oficina

Se pueden editar; los demás campos de Solicitud permanecerán inhabilitados.

Con respecto al Equipo como es:

- Procesador
- Memoria
- Disco Duro
- Tarjeta de Red

Serán cargados también automáticamente en base a lo solicitado, estos datos podrán ser modificados.
Existen dos Grid:

Equipo Disponible
Equipo que se va a Asignar

La funcionalidad del Grid de Equipo disponible es mostrar el equipo que en ese momento existe disponible en base a una selección de un icono que se encuentra en una barra de herramientas (Selección de equipo) del lado derecho de ambos grid's, mostrando los iconos de:

- CPU
- Teclado
- Mouse
- Monitor
- Impresora
- Otros

El icono de CPU realizara una búsqueda en base a la configuración que se realizó en la parte de Equipo y se mostrará en el Grid el Resultado de tal búsqueda, teniendo como encabezados las columnas: Procesador, Memoria, Disco Duro y Tarjeta de Red.

En base al resultado obtenido se podrá seleccionar a través de un botón (>>) el equipo más conveniente y pasarlo al Grid de Equipo que se va a asignar, el cual tiene la funcionalidad de mostrar el equipo que se asignara finalmente. Si se desea también se puede regresar el equipo del Grid de equipo asignado al Grid de Equipo disponible, con el botón (<<).

El botón de Teclado de la barra de selección de equipo tiene como funcionalidad mostrar en el Grid de Equipo Disponible todos los teclados disponibles en ese momento.

El botón con el icono de Mouse en la barra de selección de equipo tiene como funcionalidad mostrar en el Grid de Equipo Disponible todos los mouse disponibles en el momento.

El botón con el icono de Monitores en la barra de selección de equipo tiene como funcionalidad mostrar en el Grid de Equipo Disponible todos los monitores disponibles en el momento.

El botón con el icono de Impresoras en la barra de selección de equipo tiene como funcionalidad mostrar en el Grid de Equipo Disponible todas las impresoras disponibles en el momento.

Por último el botón con el icono de Dispositivo (otros) en la barra de herramientas tiene como funcionalidad mostrar en el Grid de Equipo Disponible todos los dispositivos diferentes a los anteriores disponibles en el momento.

Como se indico anteriormente el Grid de Equipo a Asignar se irá llenando en base a la selección de los botones >> y si ocurre algún error se tiene la opción de regresar ese equipo con el botón << ; ambos botones se encuentran entre los dos Grid.

Cada vez que se realice una búsqueda de equipo disponible se mostrará el total de componentes encontrados.

El campo de status de la solicitud puede tomar valores tales como: "Solicitud Atendida", " Solicitud Atendida Parcialmente " ó " Solicitud No Atendida ", este campo su valor de Default será en Blanco.

Una vez que se termino con el proceso de Equipo a Asignar se mostrará el total de los equipos que serán asignados.

El campo de comentarios es un campo que se despliega para hacer ciertas indicaciones u observaciones para la asignación.

Los títulos de las columnas del Grid de Equipo disponible van a variar en base a lo que se busque.

Para el caso de CPU será: Procesador, Memoria, Disco Duro, Tarjeta de Red, Código Softtek

Para el caso de Teclado, Mouse: Marca, No de Serie, Código Softtek.

Para el caso de Monitor: Marca, No de serie, Pulgadas(tamaño), Código Softtek.

Para el caso de Impresora: Marca, Modelo, No de serie, Código Softtek.

Para el caso de Otros: Marca, Modelo, No de Serie, Código Softtek.

Cuando todos los datos que se han seleccionado son los adecuados y cumplen las necesidades, se oprime el botón guardar, para insertar los nuevos registros en las tablas de ASIGNACION y SOLICITUD; regresará a la misma pantalla con los campos en blanco, hasta que se decida oprimir el botón salir; al seleccionar salir el sistema despliega la pantalla del menú principal. Si los datos no son correctos, se puede utilizar el botón de Limpiar, con lo que se pondrán todos los campos del formulario en blanco.

Proceso de Bajas

Para poder dar una baja equipo de una asignación se teclea el Folio de la solicitud; dicho folio sirve de llave para acceder a la tabla de SOLICITUD; consulta los datos de asignación y equipo para poder llenar los campos correspondientes a la forma, en el grid del lado derecho aparecen los equipos con que cuenta la asignación.

Seleccionar el registro del equipo que fue devuelto al personal de help desk, elegir el botón eliminar "<<" que se encuentra entre los dos grid's, al guardar este procedimiento se borrará de la solicitud y cambiará su status a disponible en la tabla de equipo; la selección realizada se verá reflejada en el grid.

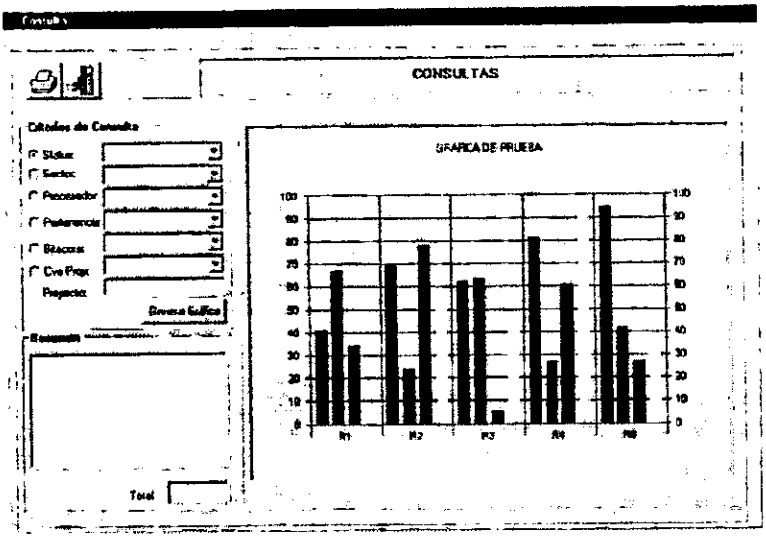
La solicitud va a manejar tres status dependiendo de que si el equipo se regresó completamente "Totalmente devuelto" o no "Parcialmente devuelto", para poder liberar la solicitud.

Proceso de Modificaciones

El proceso es similar al de bajas; se teclea el Folio de la solicitud; dicho folio sirve de llave para acceder a la tabla de SOLICITUD; consulta los datos de asignación y equipo para poder llenar los campos correspondientes a la forma, en el grid del lado derecho aparecen los equipos con que cuenta la asignación Y son los campos que se podrán manipular, dependiendo de la operación a realizar eliminar "<<" o agregar ">>" más equipos a la solicitud; al guardar el cambio se realiza la actualización en la tabla equipo

En todas las pantallas para facilitar el funcionamiento para el usuario se contara con botones de selección y grids para consultas y selección de datos. Según IEEE estándar 730 y 983 Estándares, prácticas y convenciones, Buttler Estándares y convenciones e ISO 9000-3 Diseñó e implementación, Acis. Documentación del software.

Módulo	Consultas
Página	Consulta de Estadísticas en forma Gráfica



Objetivo:

Realizar el procedimiento de Consulta en forma Gráfica en base a diferentes criterios de selección.

Especificación Funcional

Esta pantalla es invocada en el momento que se seleccione del menú la opción Consultas.

La pantalla muestra un formato con los siguientes campos:

- Status
- Sector
- Procesador
- Pertenencia
- Bitácora.
- Cve. Proyecto
- Descripción del proyecto
- Lista de Resumen
- Total

Cuando se carga la forma los radio bottom están todos habilitados y los combos se encuentran deshabilitados, una vez seleccionado el radio bottom del cual se quiere generar el gráfico se habilitará inmediatamente el combo quedando de esta forma los radio bottom no seleccionados también deshabilitados, si se desea cambiar el criterio de selección solo basta con oprimir el botón de Limpiar del menú y automáticamente la pantalla vuelve a adquirir la condición inicial de todo habilitado.

Al seleccionar el status, se activa otro combo en el cual se selecciona el equipo a graficar con su respectivo porcentaje, según el status elegido

Si la búsqueda se elige por Sector, es decir todos los equipos que estén asignados a un sector; la búsqueda se realizará en la tabla de ASIGNACION.

Si la consulta se desea realizar por tipo de Procesador, es decir, todos los equipos existentes con un procesador determinado entonces la consulta se realizará en la tabla de Componente CPU

Si la consulta se realiza por el criterio de pertenencia, es decir todos los equipos dan una pertenencia determinada entonces la búsqueda se realizará en la tabla de COMPONENTE.

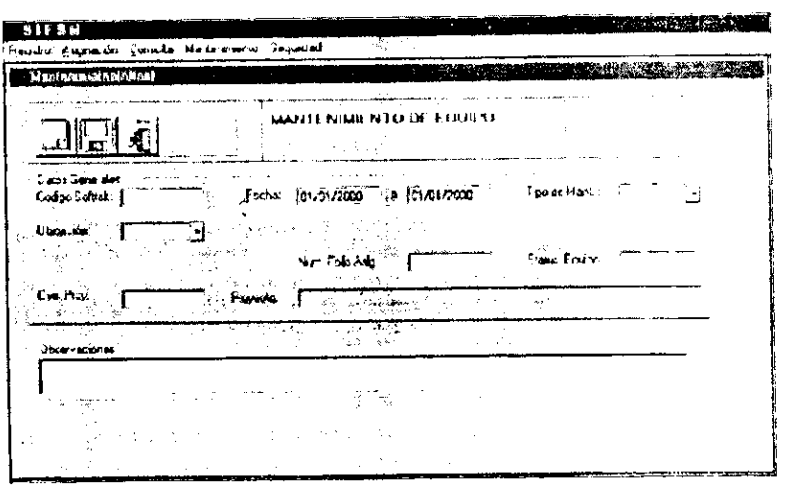
Si se elige la opción de bitácora se representarán gráficamente todos los equipos con un status de mantenimiento preventivo y todos los equipos con un status de mantenimiento correctivo.

Si se elige la opción de Proyecto la consulta se realizará sobre la vista V1_Proyecto, es decir, se representarán gráficamente los componentes que estén asignados a determinado proyecto, desplegándose al mismo tiempo su respectiva descripción (tomada de la misma vista).

Todos los movimientos anteriores generarán la gráfica en el momento de Seleccionar el botón "Generar gráfico", ese momento se genera la gráfica y en el frame de Resumen se obtendrán los resultados en forma de texto, es decir, se hará un reporte de lo que se está observando en la gráfica, mostrándose un total por criterio.

Para hacer más fácil las consultas para se podrán hacer de manera gráfica para una rápida interpretación de la información necesitada (Según IEEE estándar 730 y 983 Estándares, prácticas y convenciones, Buttler Estándares y convenciones e ISO 9000-3 Diseñó e implementación, Acis. Documentación del software.)

Módulo	Mantenimiento
Página	Altas de Mantenimiento de Equipo



Objetivo:

Realizar los procedimientos de Altas para la Bitácora de Mantenimiento de un equipo

Especificación Funcional

Esta pantalla es invocada en el momento que se seleccione del menú la opción Bitácora, Mantenimiento.
La pantalla muestra un formato con los campos de:

- Código Softtek
- Fecha inicial
- Fecha Final
- Tipo de Mantenimiento (Preventivo, Correctivo)
- Ubicación (Interna, Externa)
- Piso (Aplica para ubicación interna)
- Oficina (Aplica para ubicación interna)
- Número de folio asignado (Aplica para equipo asignado)
- Status Equipo (asignado, disponible)
- Clave de Proyecto (Aplica para equipo asignado)
- Proyecto (Aplica para equipo asignado)
- Observaciones

Proceso de Registro de Bitácora:

1.- Proporcionar la siguiente información en la pantalla:

- Código Softtek
- Fecha inicial de mantenimiento (tiene por default el valor de la fecha del sistema)
- Fecha tentativa de cuando termina el mantenimiento
- Tipo de Mantenimiento que puede ser Correctivo o Preventivo.
- Observaciones

2.- De la BD extraer la siguiente información para llenar los campos restantes:

Con el código Softtek se realizará una búsqueda en la tabla de EQUIPO y si existe ese código se validará el Status que tiene, en caso de tener un status de asignado entonces se extraerá la información necesaria de la tabla de SOLICITUD para llenar los campos de Clave de Proyecto, Ubicación Interna o externa según sea el caso, si la ubicación es interna se llenará el campo de piso y oficina. Con la clave del proyecto se extraerá el campo Nombre de Proyecto de la tabla PROYECTO y se llenará la descripción.

En caso de que el Componente aplique todavía Garantía (Fecha de Garantía Menor o Igual a la de Fecha de Mantenimiento) se llenará el campo Status Equipo con la clave de "Con Garantía" en caso contrario con la clave "Sin Garantía" y se continuará con la Alta.

3.- Efectuar las siguientes validaciones:

En caso de que el status del Equipo nos indique que esta dado de baja se enviará mensaje con la leyenda "El componente ya está dado de baja" y no se Continuará con la Alta.

Solo se aceptarán componentes cuyo status sea "disponible" o "asignado". Con cualquier status diferente no se continuará con la Alta.

Si el Campo de Observaciones está vacío. Al oprimir el Botón de Guardar se mandará un mensaje de error indicando "Campo de Observaciones Requerido" de lo contrario no se dará de alta.

Si el equipo tiene status de que se encuentra en mantenimiento se enviará un mensaje con la leyenda "El equipo ya está en mantenimiento".

4.- Consideraciones:

Si el tipo de mantenimiento es Correctivo se modificará el status del equipo en la tabla EQUIPO con la clave "Rep. Con garantía" o "Rep. Sin garantía".

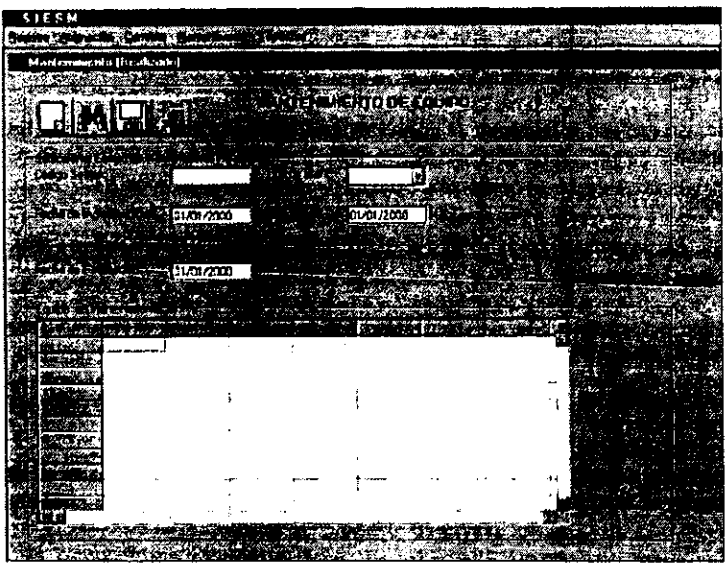
Si el Mantenimiento es Correctivo y es un componente Asignado se eliminará el registro correspondiente a dicho equipo de la tabla ASIGNACION.

Posteriormente se continuará con la Alta de bitácora de mantenimiento que registrará los datos necesarios en la Tabla de BITMAN.

Si el Mantenimiento es Correctivo y es un componente asignado, después de efectuar el alta en BITMAN se desplegará la pantalla de modificación de asignación para sustituir el equipo que esta entrando en mantenimiento correctivo.

Para dar de alta un nuevo registro basta con oprimir el botón de Limpiar y los campos se limpiarán automáticamente. En algunas pantallas solo es necesario llenar algunos datos para poder extraer los datos restantes. (Estándares, prácticas y convenciones Buttler Estándares y convenciones e ISO 9000-3 Diseñó e implementación.)

Módulo	Mantenimiento
Página	Baja de Mantenimiento de Equipo



Objetivo:

Dar por terminado el mantenimiento a un equipo/componente.

Especificación Funcional

Para establecer la terminación de mantenimiento de un equipo se podrá solicitar un equipo especifico por su código de inventario Softtek o seleccionar varios equipos a través de diferentes criterios de selección, por rangos de fecha o por tipo de mantenimiento (correctivo o preventivo).

Esta pantalla es invocada en el momento que se seleccione del menú la opción Bitácora- Mantenimiento Realizado.

La pantalla muestra un formato con los campos de:

- Código Softtek
- Tipo de mantenimiento (Preventivo, Correctivo)
- Fechas de ingreso a la bitácora de mantenimiento (rango de fechas)
- Fecha de realización (terminación del mantenimiento)
- Grid conteniendo los equipos que serán seleccionados para cambiarles el status (indicando que ya se terminó de realizar el mantenimiento)

Proceso de Baja:

1.- Proporcionar la siguiente información en la pantalla:

- Código Softtek
- Tipo de Mantenimiento
- Fecha de Ingreso (fecha inicial del rango)
- Fecha de Ingreso (fecha final del rango)

Después de proporcionar la información anterior se seleccionará el botón "BUSCAR"

2.- Efectuar las siguientes validaciones:

Si teclearon el Código Softtek la selección de los equipos a dar de baja se concretará únicamente al equipo que se nos esta indicando.

Si no teclearon el Código Softtek la selección de los equipos a dar de baja se efectuará a través del tipo de mantenimiento y/o las fechas de ingreso que estén en el rango especificado.

La fecha de Realización será obligatoria que se teclee.

3.- Proceso de Búsqueda:

Con los criterios de selección se buscarán los registros en la tabla BITMAN y el resultado se mostrará en el Grid de Equipo en Mantenimiento donde se podrá seleccionar el o los equipos que se deseen dar de baja del mantenimiento.

Los títulos de las columnas del Grid serán: Código Softtek, Fecha de Mantenimiento, Observaciones, Ubicación, Status del Mantenimiento, las cuales serán llenadas extrayendo la información de la tabla de BITMAN.

4.- Seleccionar los equipos a dar de baja.

Seleccionar del Grid los equipos que se vayan a dar de baja.
Los equipos que hayan sido seleccionados del Grid se les deberá indicar con que status quedarán (Disponible o Baja).

Después de proporcionar la información anterior se seleccionara el botón "GUARDAR"

5.- Proceso de Baja.

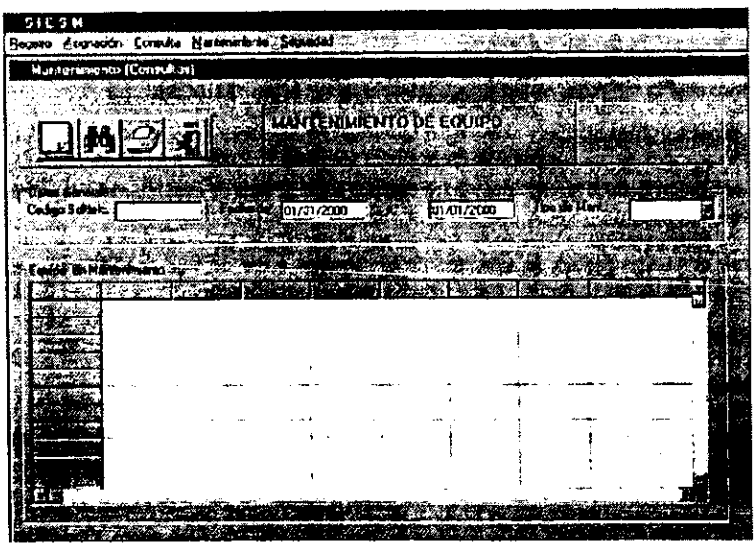
Con los datos de los registros seleccionados para dar de baja, se actualizará la tabla EQUIPO y se le cambiará el status que tenían de que estaban en mantenimiento a "Disponibles" o "Baja". Se desplegará un mensaje indicando que ya se efectuó la actualización y se volverá a desplegar el Grid, pero en esta ocasión sin los registros que fueron actualizados.

Los registros no se borran físicamente de la tabla BITMAN, únicamente se les graba la fecha en que fue realizado su mantenimiento (fecha fin).

6.- Consideraciones.

Para realizar una nueva búsqueda solo basta con oprimir el botón Limpiar(nuevo) y automáticamente todos los campos de la pantalla se limpian y se pueden dar de nuevo los criterios a consultar para realizar una nueva baja de la bitácora.

Módulo	Mantenimiento
Página	Consulta de Mantenimiento de Equipo



Objetivo:

Realizar las Consultas a bitácora de mantenimiento de equipos (componentes)

Especificación Funcional

Para realizar una consulta de equipos en mantenimiento podrá solicitar un equipo especifico por su código de inventario Softtek o seleccionar varios equipos a través de diferentes criterios de selección, por rangos de fecha o por tipo de mantenimiento (correctivo o preventivo).

Esta pantalla es invocada en el momento que se seleccione del menú la opción Bitácora-Consultas.
La pantalla muestra un formato con los campos de:

- Código Softtek
- Fecha inicial
- Fecha Final
- Tipo de Mantenimiento (Preventivo, Correctivo)
- Grid conteniendo los equipos que cumplen los criterios de la consulta

Proceso de Consulta:

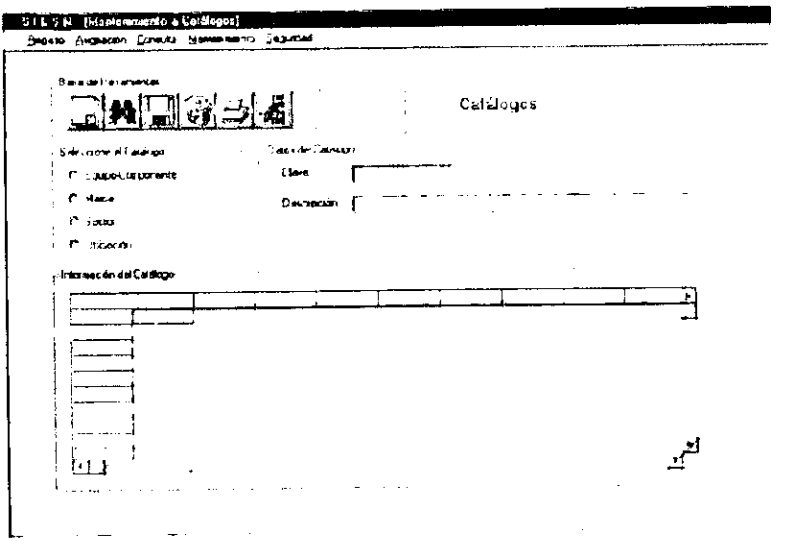
Para realizar el proceso de Consulta solo basta teclear cualquiera de los campos de la pantalla (Datos generales) y seleccionar el botón de buscar, el cual realizará una búsqueda en la tabla de BITMAN, en base al criterio dado.

El resultado se mostrará en el Grid de Equipo en Mantenimiento el cual no podrá ser editable, solo su función es mostrar el resultado, para realizar una nueva búsqueda solo basta con oprimir el botón de Limpiar(nuevo) y automáticamente todos los campos de la pantalla se limpian y se pueden dar de nuevo los criterios a consultar para realizar una nueva consulta de la bitácora.

Los criterios de selección son por Código Softtek, rango de fechas y tipo de mantenimiento.

Los títulos de las columnas del Grid serán: Código Softtek, Fecha de Mantenimiento, Observaciones, Ubicación, Status del Mantenimiento, las cuales serán llenadas extrayendo la información de la tabla de BITMAN.

Módulo	Catálogo
Página	Catálogos de Equipo-Componente, Marca, Sectores, Ubicación



Objetivo:

Realizar transacciones de Alta,.Baja, Modificaciones y Consulta de los catálogos de Equipo-Componente, Marcas, Sectores y Ubicación.

Especificación Funcional

Esta pantalla es invocada en el momento que se seleccione del menú la opción de Mantenimiento-Catálogos; se compone de:

- Selección del catálogo
- Datos del catálogo
- Información del catálogo

En la Selección del catálogo aparecen cuatro criterios:

- Equipo-componente
- Marca
- Sector
- Ubicación

Al elegir equipo-componente, el control envía a otra pantalla Marca, Sector y Ubicación activan el frame de Datos de catálogo con los campos:

- Clave
- Descripción

Clave y descripción se hacen editables al seleccionar Marca o Sector; siempre y cuando se trate de una alta o modificación, para este catálogo.

Piso y oficina se hacen editables al elegir ubicación; siempre y cuando se trate de una alta o modificación, para este catálogo.

El grid que se encuentra en la parte de Información del catálogo es activado cuando se guarda nueva información (este nuevo registro se ve reflejado en el grid); al realizar una consulta o un cambio.

Los campos desplegados en el grid son: clave, descripción (para MARCA y SECTOR); piso y oficina (para Ubicación).

El estado inicial de la pantalla es con Selección del catálogo habilitado.

Proceso de Alta:

Seleccionar el catálogo a manipular (se habilita la barra de herramientas).

Elegir de la barra de herramientas el botón de nuevo (limpia y se habilitan los campos de Datos del catálogo).

Capturar la información referente a la clave y descripción del Sector o la Marca; Si se elige Ubicación entonces habilitar los campos de piso y oficina.

Una vez proporcionados los datos se procederá a oprimir el botón de Guardar que se encuentra en la barra de herramientas y el registro se dará de alta en la tabla de MARCA, SECTOR o Ubicación.

Si por algún motivo no se proporciono la clave y se procedió a oprimir el botón de guardar, se desplegará un mensaje informando que no se puede dar de alta el registro.

Si al realizarse la transacción los datos en los campos de Clave y Descripción (piso y oficina) ya existen, no se podrá dar de alta el registro, es decir los datos que se proporcionen deberán ser únicos e irrepetibles.

Una vez cumplidas las condiciones anteriores el Grid mostrará automáticamente los datos que se están dando de Alta. Estos datos que se muestran no se podrán editar en el Grid, solo tiene la función de ser informativo.

Proceso de Baja:

El proceso de Eliminación se puede realizar a través de criterios diferentes como es proporcionando la Descripción, la clave o solo el catalogo a manipular y seleccionando en la barra de herramientas el botón Buscar, entonces automáticamente se desplegará en el Grid todos los registros que cumplieron la condición de búsqueda.

Una vez mostrados los datos se procederá a elegir en el Grid los registros que se deseen dar de Baja (la información del registro se desplegará en la sección de Datos del catálogo) y se oprimirá el Botón de Eliminar que se encuentra en la barra de herramientas. Inmediatamente se actualizará el Grid.

En caso de que no existan registros con esas características se mandará un mensaje informando que no existen registros.

Este proceso de Eliminación de Registros se realizará en las tablas de MARCA, SECTOR y UBICACION.

Proceso de Modificaciones:

Una vez dados los criterios de Búsqueda se seleccionará el botón de Buscar que se encuentra en la barra de herramientas y automáticamente en el Grid se reflejarán los resultados de la búsqueda, se seleccionará el registro que se desea modificar (los datos se mostrarán en la sección de Datos del catálogo), de tal forma que estos valores se puedan modificar.

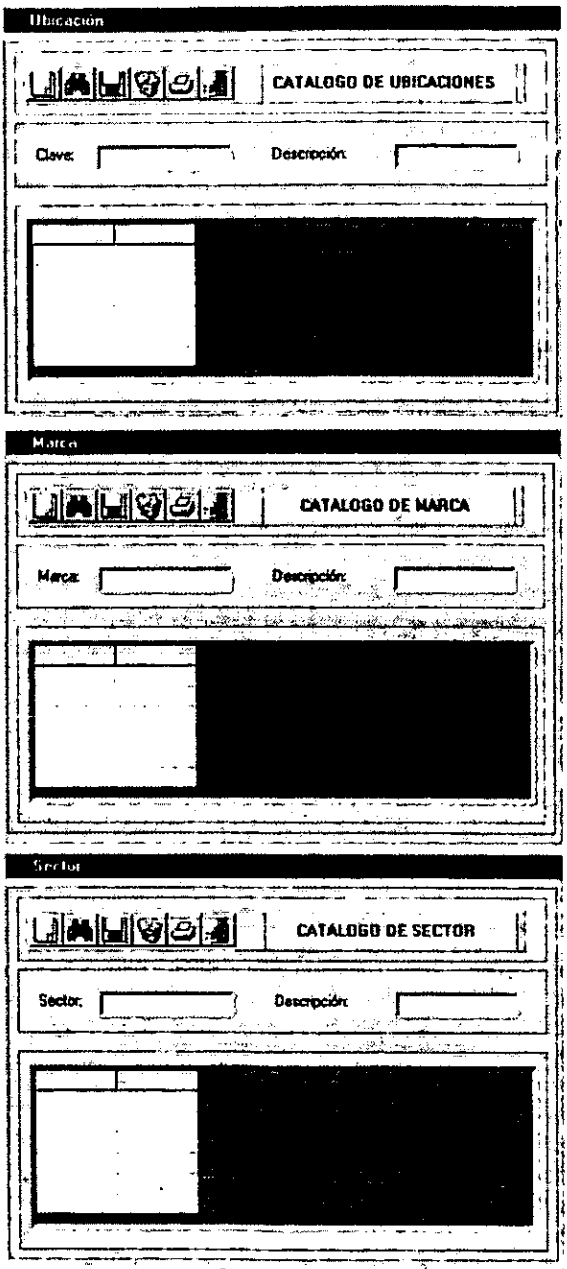
El campo de Clave solo mostrará el valor pero no se podrá Editar.

Una vez realizadas las modificaciones el botón de Guardar será seleccionado para actualizar el registro con los cambios hechos, todo este proceso se lleva a cabo en la tabla de MARCA, SECTOR y UBICACION.

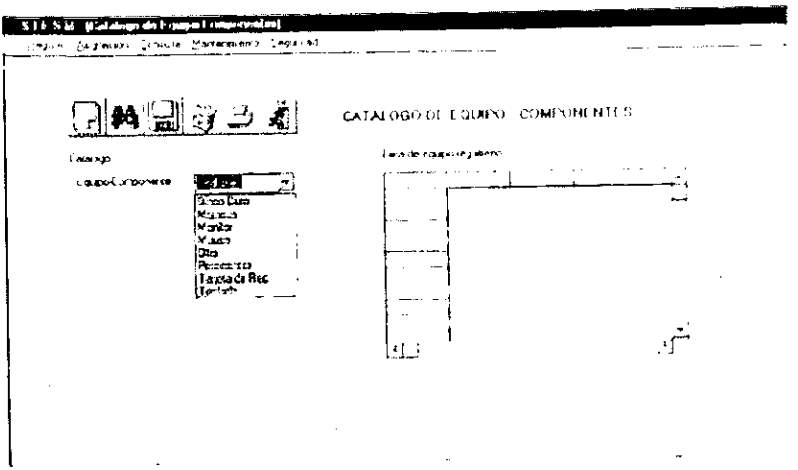
Proceso de Consulta:

El proceso de Consulta se realiza a través de los diferentes criterios proporcionados en la pantalla.

Una vez dados los criterios de Consulta se seleccionará el botón de Buscar que se encuentra en la barra de herramientas y automáticamente en el Grid se reflejarán los resultados de la búsqueda, estos datos no se podrán editar, es decir solo serán informativos. En algunos grids solo se podrán utilizar como consulta sin poder modificar los datos. (Buttler Estándares y convenciones e ISO 9000-3 Diseñó e implementación).


Ubicación
CATALOGO DE UBICACIONES
Clave:
Descripción:
Marca
CATALOGO DE MARCA
Marca:
Descripción:
Sector
CATALOGO DE SECTOR
Sector:
Descripción:

Módulo	Catálogo
Página	Catálogo de Equipo-componentes, Altas, bajas y cambios



Objetivo:

Realizar los procedimientos de Altas, bajas y cambios a los equipos y componentes registrados.

Especificación Funcional

Esta pantalla es invocada en el momento que se seleccione de la pantalla catálogos la opción "Equipo-Componentes" .

La pantalla muestra un formato en que se selecciona el equipo o componente, y dependiendo de la selección muestra los siguientes campos:

- Tipo (conector, tarjeta de red o adaptador)
- Velocidad
- Capacidad o tamaño
- Número de serie

En la parte inferior de la pantalla, aparece un grid con los diferentes elementos que son cargados automáticamente al seleccionar un componente, mostrando las siguientes columnas:

- Tipo (tarjeta de red o adaptador)
- Velocidad
- Capacidad o tamaño
- Número de serie

Proceso para Registrar un nuevo tipo de componente

Al seleccionar del menú la opción de Nuevo, en el campo componente se puede seleccionar el tipo de equipo a dar de alta, posteriormente se llenan los campos del área de características del componente.

Se selecciona del menú la opción de Guardar, para que quede actualizada con el nuevo registro.

Proceso para Modificar un tipo de componente

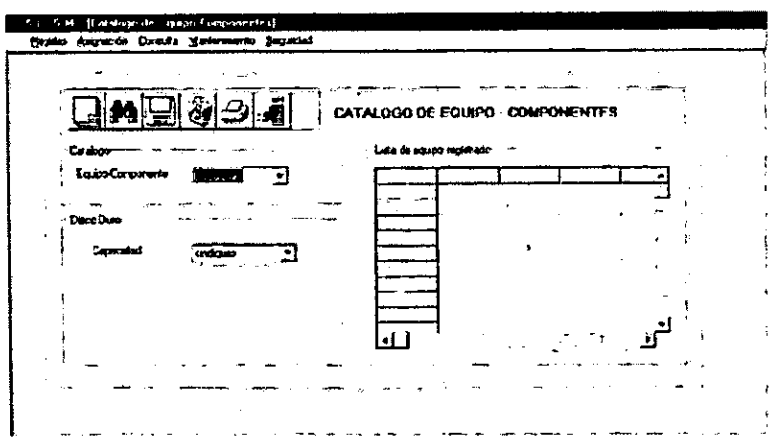
Para modificar determinada característica de un componente, se selecciona del grid el elemento a modificar, al ser seleccionado se hacen editables los campos del área de las características del componente, para que puedan ser modificados al elegir del menú la opción de Guardar, el registro quedará actualizado.

Proceso para Consultar los tipos de componentes

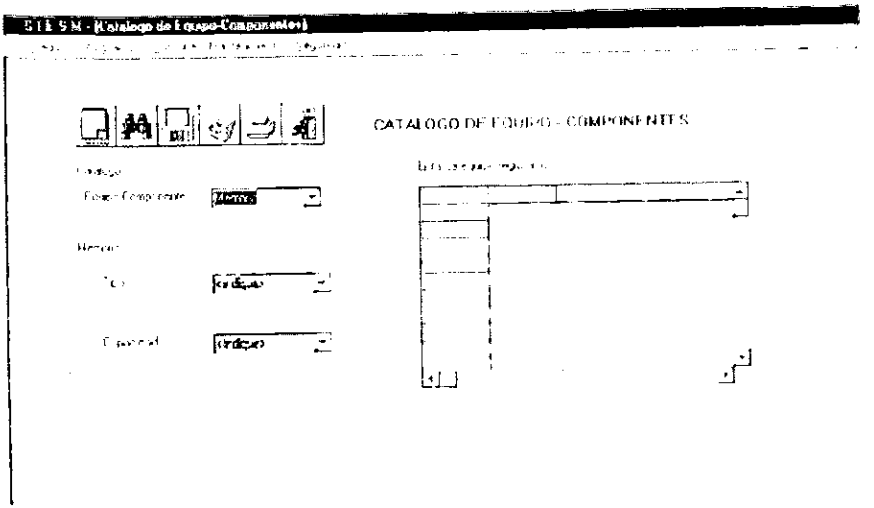
Al seleccionar uno de los componentes que se despliegan en las opciones del combo, se mostrarán en el grid, todos los tipos de registro correspondientes al componente elegido.

Proceso para Eliminar un tipo de componente

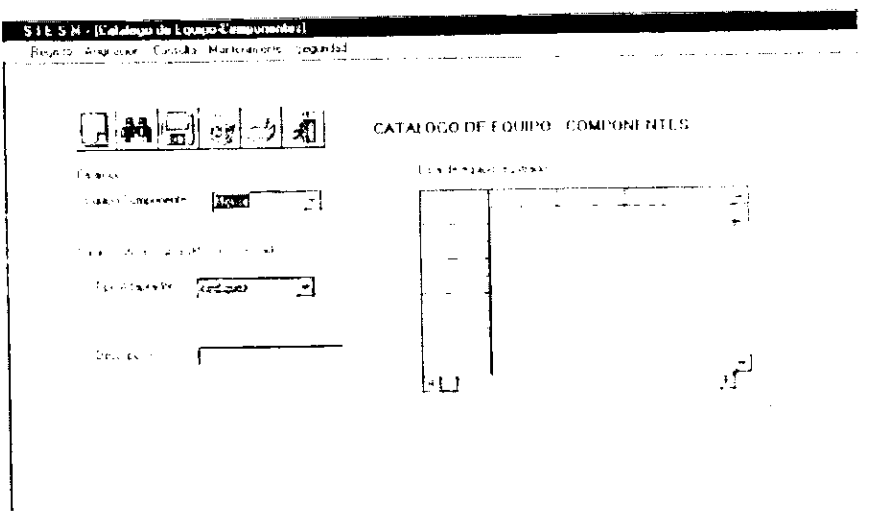
Al elegir buscar del menú; se activa el combo de selección de equipo-componente en el cual se elige el equipo a buscar y el grid es llenado con la información referente a la búsqueda solicitada, posteriormente se elige el registro a borrar y del menú se selecciona la opción eliminar.



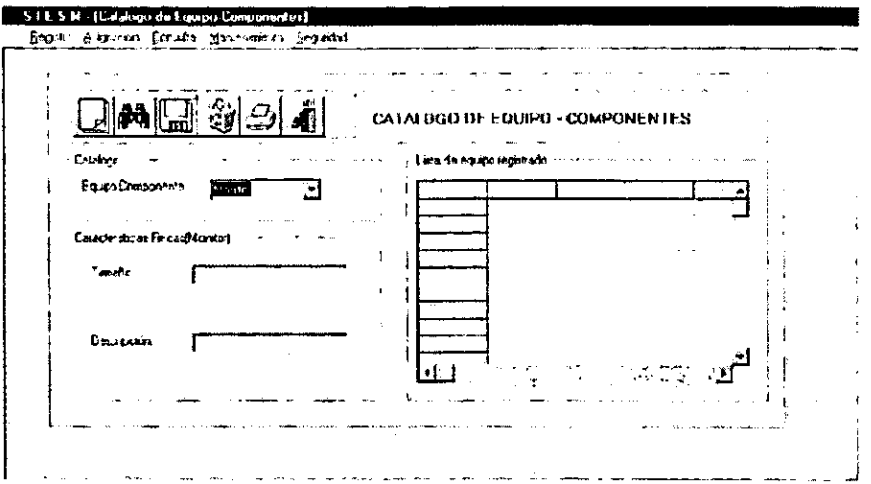
Al momento de seleccionar la opción de Disco Duro, se mostrará un cuadro de texto para que el usuario ingrese la capacidad del disco duro.



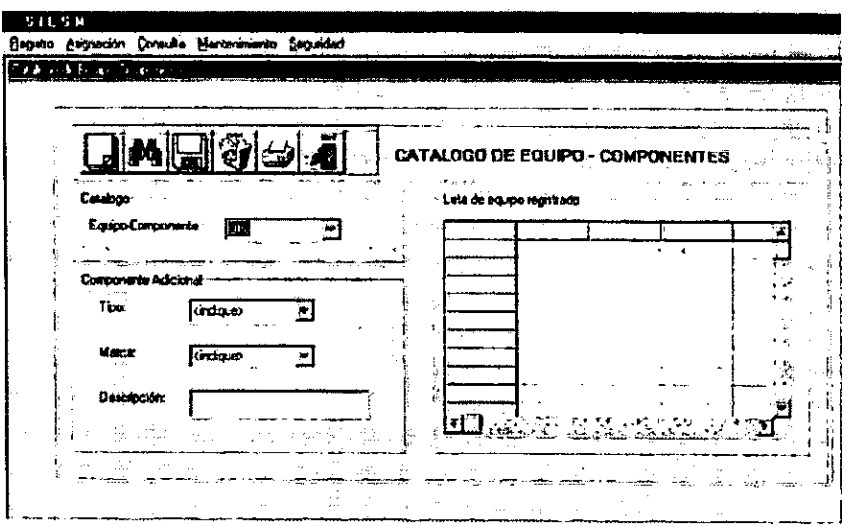
Al seleccionar la opción de Memoria, se mostrarán un par de cuadros de texto para ingresar el tipo de memoria y la capacidad de la misma.



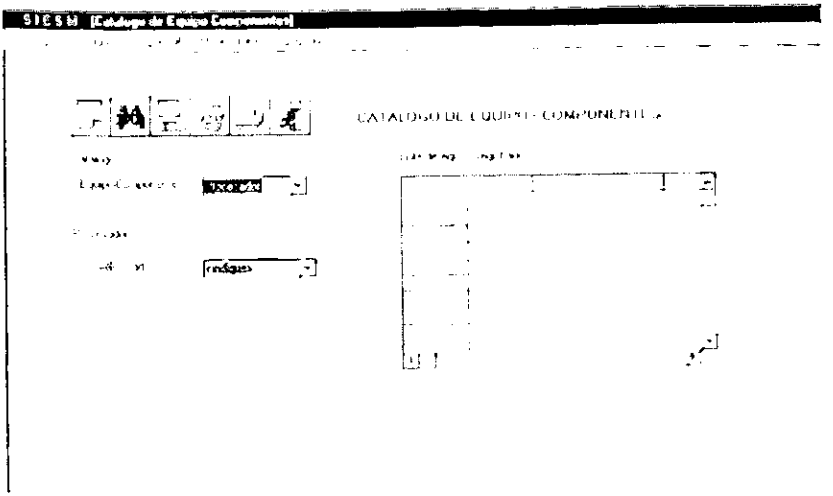
La opción del Mouse, mostrará dos cuadros de texto donde se ingresan el tipo de conector o adaptador y un cuadro para algún comentario adicional.



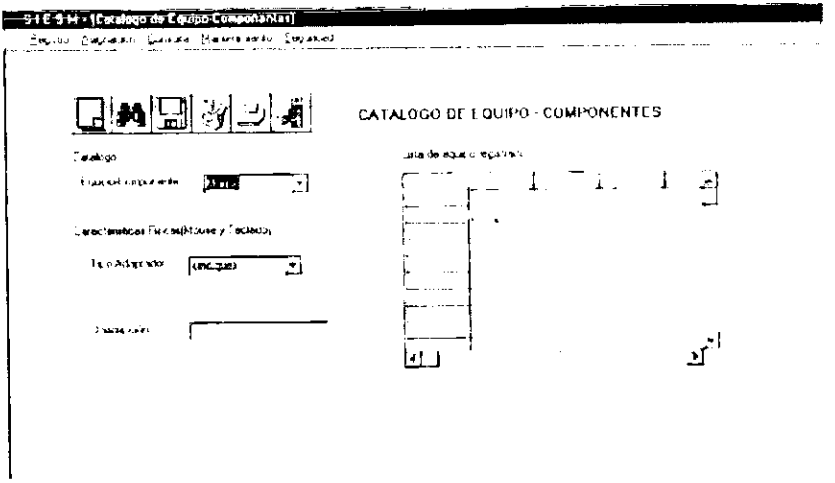
Al momento de seleccionar la opción de Monitor, se mostrará un cuadro de texto para que el usuario ingrese el tamaño del monitor en pulgadas y otro cuadro para agregar alguna descripción adicional.



Al seleccionar la opción de Otro, se mostrarán cuadros de texto para que se ingrese el tipo la marca y descripción del dispositivo.

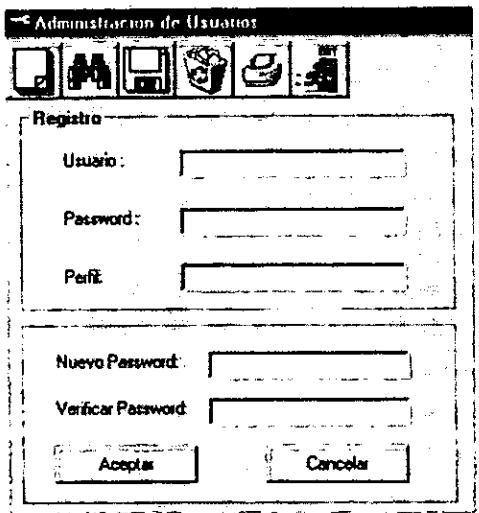


Al momento de seleccionar la opción de Procesador, el usuario ingresará la velocidad.



Al momento de seleccionar la opción de Teclado, se mostrará un cuadro de texto para que el usuario ingrese el tipo de adaptador y otro cuadro más para la descripción.

Módulo	Seguridad
Página	Altas, Bajas y Cambios de Usuarios al sistema



Objetivo:

Realizar el proceso de registro, consulta, modificación y eliminación de Usuarios

Especificación Funcional

Esta pantalla es invocada en el momento que se seleccione de la pantalla principal la opción de Administración de usuarios. Para poder acceder al sistema se necesitará dar de alta primero a los usuarios que podrán acceder al sistema . Según IEEE estándar 730 y 983 Gestión de configuración del software, ISO 9000-3 Diseñó e implementación, Acis).

Al entrar a esta pantalla, será cargado el combo con todos los usuarios que se encuentren en la tabla de usuarios permitidos para acceder al sistema, además se ilustrará en el texto del combo la leyenda de
"**< indique >**".

Proceso para Registrar un nuevo usuario

Seleccione el menú "nuevo", y la pantalla se pondrá con dos TextBox en donde se podrá ingresar los datos del nuevo usuario: el nombre y el password. Haga click en el menú en la opción guardar y esos datos serán registrados en la tabla de Seguridad.

Proceso para borrar un usuario.

Seleccione el usuario.
Seleccione el botón de borrar en el menú y el usuario será eliminado del sistema.

Proceso para consultar y modificar un usuario.

Seleccione el usuario.
Cambie su password (si lo desea).
Seleccione el botón de grabar en el menú, para actualizar los datos modificados.
Para poder acceder al sistema se necesitará dar de alta primero a los usuarios que podrán acceder al sistema . (Según IEEE estándar 730 y 983 Gestión de configuración del software, ISO 9000-3 Diseñó e implementación, Acis. Documentación del software.)

4.3.2. Pruebas y Resultados

En este mismo capítulo se han planteado y fijado todos los antecedentes y las bases para realizar el desarrollo del sistema de inventarios para la empresa Softtek de México.

Una vez que el sistema sea desarrollado será necesario aplicar diferentes tipos de pruebas al mismo, para determinar o encontrar fallas funcionales, lógicas o de diseño. Es necesario realizar las pruebas necesarias y evaluar los resultados esperados. Según Presuman 1995 Prueba del Software Acis (1994) Pruebas del software, Dobbins (1990) Actividades de pruebas. ISO 9000 – 3 Pruebas y validación). Por lo que se recomienda aplicar algunas pruebas el sistema como:

Es necesario realizar las pruebas necesarias y evaluar los resultados esperados. Según Presuman 1995 Prueba del Software. Acis (1994) Pruebas del software, Dobbins (1990) Actividades de pruebas. ISO 9000 – 3 Pruebas y validación.

Pruebas de Caja Blanca. Que permitirán entender y validar la lógica del sistema así como para entender las estructura interna del sistema.

Pruebas de Caja Negra. Pruebas funcionales de los datos basadas en los requerimientos del cliente.

Funcionales. Validaran el funcionamiento del sistema para los usuarios basándose en los requerimientos de los mismos, por lo que es importante aplicar este tipo de pruebas (Es necesario realizar las pruebas necesarias y evaluar los resultados esperados. Stamm Programa de pruebas, Revisiones formales. ISO 9000 – 3 Pruebas y validación.) basándose como ejemplo en las matrices de prueba que se describen a continuación.

Sistema: Sistema de Inventario de Equipo Softtek México (SIESM).				
Módulo: Registro		Pantalla: Alta de Facturas		
Caso	Descripción del caso	**Resultado Esperado**	**Resultado Obtenido**	**Observaciones**
1	Acceder a la pantalla de registro de facturas.	Barra de herramientas /a excepción del botón buscar), opciones de equipo comprado o arrendado, proveedor, folio y monto, habilitados. El resto de controles deshabilitados.		
2	Seleccionar equipo comprado.	Deshabilitar la opción de equipo arrendado. Habilitar los controles correspondientes a fechas factura (excepto fin de garantía)		
3	Seleccionar equipo arrendado.	Deshabilitar la opción de equipo comprado. Habilitar datos de fechas arrendamiento.		
4	Seleccionar equipo comprado. Seleccionar con garantía = si.	Deshabilitar la opción con garantía = no y la fecha de fin de garantía.		

5	Teclear un caracter diferente de número en las fechas.	No permitir la captura del caracter.		
6	Capturar un folio de más de 15 caracteres.	No permitir la captura de más de 15 caracteres.		
7	Capturar un monto de más de 12 caracteres.	No permitir la captura de más de 12 caracteres.		
8	Seleccionar el botón "guardar" de la barra de herramientas. Sin seleccionar si es equipo comprado o arrendado.	Desplegar un mensaje solicitando la selección de equipo comprado o arrendado.		
9	Seleccionar el botón "guardar" de la barra de herramientas. Sin seleccionar un proveedor.	Desplegar un mensaje solicitando la selección de un proveedor.		
10	Seleccionar el botón "guardar" de la barra de herramientas. Sin capturar un folio de factura o contrato de arrendamiento.	Desplegar un mensaje solicitando un número de folio.		
11	Seleccionar el botón "guardar" de la barra de herramientas. Con un monto no numérico.	Desplegar un mensaje indicando que el monto debe ser numérico.		
12	Seleccionar el botón "guardar" de la barra de herramientas. Con un monto mayor a 9 enteros.	Desplegar un mensaje indicando que el monto debe ser de a lo más 9 enteros.		
13	Seleccionar el botón "guardar" de la barra de herramientas. Con un monto de más de 2 decimales.	Desplegar un mensaje indicando que el monto puede tener a lo más 2 decimales.		
14	Seleccionar el botón "guardar" de la barra de herramientas. Con alguna fecha no válida.	Desplegar un mensaje indicando que la fecha debe ser capturada con formato dd/mm/aaaa.		
15	Seleccionar el botón "guardar" de la barra de herramientas. Con equipo comprado seleccionado. Con garantía = no. Sin capturar fecha de facturación.	Desplegar un mensaje indicando que se debe capturar la fecha de facturación.		
16	Seleccionar el botón "guardar" de la barra de herramientas. Con equipo comprado seleccionado. Con garantía = si. Fecha fin de garantía no capturada.	Desplegar un mensaje indicando que se debe capturar la fecha fin de garantía.		
17	Seleccionar el botón "guardar" de la barra de herramientas. Con equipo comprado seleccionado. Con garantía = si. Fecha fin de garantía menor a la fecha de facturación.	Desplegar un mensaje indicando que la fecha fin de garantía debe ser mayor a la fecha de facturación.		
18	Seleccionar el botón "guardar" de la barra de herramientas. Con equipo arrendado seleccionado. Sin capturar las fechas de inicio y fin de arrendamiento.	Desplegar un mensaje indicando que se deben capturar las fechas de inicio y fin de arrendamiento.		
19	Seleccionar el botón "guardar" de la barra de herramientas. Con equipo arrendado seleccionado. Con la fecha de fin de arrendamiento menor a la de inicio.	Desplegar un mensaje indicando que la fecha de fin debe ser mayor a la de inicio.		

20	Seleccionar el botón "guardar" de la barra de herramientas. Con un folio-proveedor ya existente.	Desplegar un mensaje indicando que el folio ya existe registrado.		
21	Seleccionar el botón "guardar" de la barra de herramientas. Con todos los datos necesarios correctos.	Preguntar si se quiere dar de alta la factura. Si se confirma: Registrar la factura en la tabla DFACTURA (verificar los datos del registro en la tabla). Deshabilitar los controles a excepción de los botones "limpiar" y "salir" de la barra de herramientas. Habilitar el botón de registro de equipo.		
22	Oprimir el botón registrar equipo.	Desplegar la pantalla de registro de equipo con los datos de la factura registrada.		
23	Seleccionar el botón "limpiar" de la barra de herramientas.	Regresar la pantalla a su estado inicial.		
24	Seleccionar el botón "salir" de la barra de herramientas.	Regresar al menú principal.		

Sistema: Sistema de Inventario de Equipo Softtek México (SIESM).				
Módulo: Registro		**Pantalla:** Registro de equipo		
Caso	**Descripción del caso**	**Resultado Esperado**	**Resultado Obtenido**	**Observaciones**
1	Acceder a la pantalla de registro de equipo.	Barra de herramientas y opciones de equipo y componente habilitados; el resto de los controles inhabilitados. Presentar los datos de la factura o contrato de arrendamiento según sea el caso.		
2	Seleccionar la opción de equipo.	Habilitar combo de equipo, código Softtek (número), modelo y número de serie.		
3	Capturar más de 5 caracteres en el código Softtek.	No permitir la captura de más de 5 caracteres.		
4	Capturar más de 20 caracteres en el modelo.	No permitir la captura de más de 20 caracteres.		
5	Capturar más de 15 caracteres en el número de serie.	No permitir la captura de más de 15 caracteres.		
6	Seleccionar un equipo.	Asignar el prefijo del código Softtek. Habilitar el combo de marca. Si es LAP TOP habilitar el checkbox del seguro. Si es CPU habilitar los controles referentes a componentes de CPU.		
7	Seleccionar una marca.	Habilitar el combo de tipo.		
8	Seleccionar un tipo.	Habilitar el combo de descripción.		
9	Seleccionar con seguro.	Habilitar los campos donde se deberán registrar las fechas del seguro.		
10	Capturar caracteres no numéricos en alguna fecha.	No permitir la captura de caracteres no numéricos.		
11	Seleccionar la opción de componente.	Habilitar los controles referentes a componentes de CPU.		
12	Seleccionar el botón "guardar" de la barra de herramientas. Sin seleccionar una opción de equipo o componente.	Desplegar un mensaje solicitando la selección de equipo o componente.		

13	Seleccionar el botón "guardar" de la barra de herramientas. Seleccionar la opción equipo. No seleccionar un tipo de equipo, Código Softtek, marca tipo y/o descripción.	Desplegar un mensaje solicitando estos datos.		
14	Seleccionar el botón "guardar" de la barra de herramientas. Seleccionar la opción equipo. Seleccionar LAP TOP. Seleccionar con seguro, sin capturar fechas.	Desplegar un mensaje indicando que se deben capturar las fechas del seguro.		
15	Seleccionar el botón "guardar" de la barra de herramientas. Seleccionar la opción equipo. Seleccionar LAP TOP. Seleccionar con seguro, capturar fechas no válidas.	Desplegar un mensaje indicando que las fechas deben ser capturadas con formato dd/mm/aaaa.		
16	Seleccionar el botón "guardar" de la barra de herramientas. Seleccionar la opción equipo. Seleccionar LAP TOP. Seleccionar con seguro, capturar fecha de fin menor a la de inicio.	Desplegar un mensaje indicando que la fecha de fin no puede ser menor a la de inicio.		
17	Seleccionar la opción procesador.	Habilitar el campo número de serie del componente. Presentar los datos a capturar para el procesador, con el combo "velocidad" habilitado.		
18	Seleccionar una velocidad del procesador.	Habilitar el combo "descripción" del procesador.		
19	Seleccionar una descripción del procesador.	Habilitar el botón ">".		
20	Seleccionar la opción memoria.	Habilitar el campo número de serie del componente. Presentar los datos a capturar para la memoria, con el combo "tipo" habilitado.		
21	Seleccionar un tipo de memoria.	Habilitar el combo "capacidad" de memoria.		
22	Seleccionar una capacidad de memoria.	Habilitar el combo "descripción" de memoria.		
23	Seleccionar una descripción de memoria.	Habilitar el botón ">".		
24	Seleccionar la opción disco duro.	Habilitar el campo número de serie del componente. Presentar los datos a capturar para el disco duro, con el combo "capacidad" habilitado.		
25	Seleccionar una capacidad de disco duro.	Habilitar el combo "descripción" de disco duro.		
26	Seleccionar una descripción de disco duro.	Habilitar el botón ">".		
27	Seleccionar la opción tarjeta de red.	Habilitar el campo número de serie del componente. Presentar los datos a capturar para la tarjeta de red, con el combo "tipo" habilitado.		
28	Seleccionar un tipo de tarjeta de red.	Habilitar el combo "capacidad" de tarjeta de red.		
29	Seleccionar una capacidad de tarjeta de red.	Habilitar el combo "descripción" de tarjeta de red.		
30	Seleccionar una descripción de tarjeta de red.	Habilitar el botón ">".		
31	Seleccionar la opción otros.	Habilitar el campo número de serie del componente. Presentar los datos a capturar para otros componentes, con el combo "componente" habilitado.		

32	Seleccionar un componente de otros.	Habilitar el combo "tipo" de otros.		
33	Seleccionar un tipo de otros.	Habilitar el combo "descripción" de otros.		
34	Seleccionar una descripción de otros.	Habilitar el botón ">".		
35	Oprimir el botón ">" con los datos del componente seleccionados.	Agregar un registro en el grid con los datos del componente incluyendo su número de serie. Si se está registrando un CPU, inhabilitar la opción para capturar dos veces un componente del mismo tipo, excepto la opción otros.		
36	Seleccionar un registro del grid.	Habilitar el botón "<".		
37	Oprimir el botón "<" .	Eliminar el registro del grid y presentar los datos a capturar del componente. Si se está registrando un CPU, habilitar la opción correspondiente.		
38	Seleccionar el botón "guardar" de la barra de herramientas. Seleccionar la opción componente.	Registrar todos los componentes del grid en la tabla DCOMPONENTES_CPU. Desplegar un mensaje indicando que los componentes fueron registrados.		
39	Seleccionar el botón "guardar" de la barra de herramientas. Seleccionar la opción equipo.	Si todos los datos son correctos, enviar un mensaje de confirmación de registro del equipo, si se confirma registrar el equipo en la tabla DEQUIPO e inhabilitar el botón "guardar". Si el equipo es un CPU, también guardar todos sus componentes. Desplegar un mensaje indicando que el equipo (y sus componentes si es CPU) fue registrado.		
40	Seleccionar el botón "limpiar" de la barra de herramientas.	Regresar la pantalla a su estado inicial.		
41	Seleccionar el botón "regresar" de la barra de herramientas.	Regresar a la pantalla de registro o modificación de facturas, según sea la que mandó llamar a la pantalla de registro de equipo.		
42	Seleccionar el botón "salir" de la barra de herramientas.	Regresar al menú principal.		

Sistema: Sistema de Inventario de Equipo Softtek México (SIESM).				
Módulo: Mantenimiento		**Pantalla:** Alta		
Caso	**Descripción**	Resultado esperado	**Resultado obtenido**	**Observaciones**
1	Acceder a la pantalla de Alta de Mantenimiento (esta pantalla solo se accede mediante la consulta de mantenimiento)	1. Los botones del menú habilitados 2. Cargar todos los datos de la sección de Equipo y de Solicitud/Asignación del equipo seleccionado en la consulta de mantenimiento (las dos secciones deshabilitadas). 3. Los campos de la sección de Mantenimiento habilitados.		

2	Dar clic en el botón de guardar	1. Validar los campos correspondientes a la sección de Mantenimiento como: a. Fecha de inicio valida b. Se haya seleccionado el tipo de mantenimiento c. Se haya dado las observaciones 2. Si no son correctos se enviara un mensaje de error posicionándose en el campo correspondiente. 4. Si no hubo problema al dar de alta el mantenimiento: a. se presentara un mensaje de éxito, b. se deshabilitara todos los campos de la pantalla y el botón de guardar. c. Se espera que se actualice en la B.D. el estatus (en la tabla Dequipo) y guardar el estatus anterior (en Dbitman). 5. Los botones de regresar y salir quedan habilitados. Nota: Todos los campos de la sección de Mantenimiento son obligatorios.		
3	Dar clic en el botón de regresar	1. Cerrar la pantalla de alta y 2. Regresar a la pantalla de consulta.		
4	Al dar clic en el botón de salir	Cerrar la pantalla.		

Sistema: Sistema de Inventario de Equipo Softtek México (SIESM).				
Módulo: Consultas		**Pantalla:** Consulta de estadísticas en forma gráfica		
Caso	**Descripción del caso**	**Resultado Esperado**	**Resultado Obtenido**	**Observaciones**
1	Acceder a la pantalla de consultas.	Barra de herramientas, todas las opciones de selección (option buttons), comandos generar gráfico y generar reporte, habilitados. El resto de controles deshabilitados.		
2	Oprimir generar gráfico sin seleccionar criterio de consulta.	Desplegar mensaje solicitando un criterio de consulta.		
3	Oprimir generar reporte sin seleccionar criterio de consulta.	Desplegar mensaje solicitando un criterio de consulta.		
4	Seleccionar la opción consulta por estatus sin seleccionar un estatus, oprimir generar gráfico.	Inhabilitar opciones. Desplegar mensaje solicitando la selección de un estatus. Mostrar combo para seleccionar tipo de equipo.		
5	Seleccionar la opción consulta por estatus sin seleccionar un estatus, oprimir generar reporte.	Inhabilitar opciones. Desplegar mensaje solicitando la selección de un estatus. Mostrar combo para seleccionar tipo de equipo.		
6	Seleccionar la opción consulta por estatus, seleccionar un estatus sin seleccionar un equipo, oprimir generar gráfico.	Inhabilitar opciones. Desplegar mensaje solicitando la selección de un equipo.		
7	Seleccionar la opción consulta por estatus, seleccionar un estatus sin seleccionar un equipo, oprimir generar reporte.	Inhabilitar opciones. Desplegar mensaje solicitando la selección de un equipo.		

8	Seleccionar la opción consulta por estatus, seleccionar un estatus y un tipo de equipo, oprimir generar gráfico.	Inhabilitar opciones. Generar un gráfico de "pie" de número de equipos seleccionado con el estatus seleccionado vs. Total de equipos seleccionado. Desplegar en el cuadro resumen el número de equipos con el estatus seleccionado y el número de equipos con estatus diferente al seleccionado. Desplegar en el cuadro de totales el número total de equipos (del tipo seleccionado).		
9	Seleccionar la opción consulta por estatus, seleccionar un estatus y un tipo de equipo, oprimir generar reporte.	Inhabilitar opciones. Generar un reporte con los datos generales de cada uno de los equipos que cumplen con el tipo de equipo y estatus seleccionado.		
10	Seleccionar la opción consulta por sector sin seleccionar un sector, oprimir generar gráfico.	Inhabilitar opciones. Desplegar mensaje solicitando la selección de un sector.		
11	Seleccionar la opción consulta por sector sin seleccionar un sector, oprimir generar reporte.	Inhabilitar opciones. Desplegar mensaje solicitando la selección de un sector.		
12	Seleccionar la opción consulta por sector, seleccionar un sector, oprimir generar gráfico.	Inhabilitar opciones. Generar un gráfico de "pie" de número de equipos del sector seleccionado vs. El número total de equipos. Desplegar en el cuadro resumen el número de equipos del sector seleccionado y el número de equipos de los sectores diferentes al seleccionado. Desplegar en el cuadro de totales el número total de equipos en todos los sectores.		
13	Seleccionar la opción consulta por sector, seleccionar un sector, oprimir generar reporte.	Inhabilitar opciones. Generar un reporte con los datos generales de cada uno de los equipos del sector seleccionado.		
14	Seleccionar la opción consulta por procesador sin seleccionar un tipo de procesador, oprimir generar gráfico.	Inhabilitar opciones. Desplegar mensaje solicitando la selección de un tipo de procesador.		
15	Seleccionar la opción consulta por procesador sin seleccionar un tipo de procesador, oprimir generar reporte.	Inhabilitar opciones. Desplegar mensaje solicitando la selección de un tipo de procesador.		
16	Seleccionar la opción consulta por procesador, seleccionar un tipo de procesador, oprimir generar gráfico.	Inhabilitar opciones. Generar un gráfico de "pie" de número de CPU's con el procesador seleccionado vs. El número total de CPU's. Desplegar en el cuadro resumen el número de CPU's con el procesador seleccionado y el número de los CPU's de tipos diferentes al seleccionado. Desplegar en el cuadro de totales el número total de CPU's.		
17	Seleccionar la opción consulta por procesador, seleccionar un procesador, oprimir generar reporte.	Inhabilitar opciones. Generar un reporte con los datos generales de cada uno de los CPU's con el tipo de procesador seleccionado.		
18	Seleccionar la opción consulta por pertenencia sin seleccionar una pertenencia, oprimir generar gráfico.	Inhabilitar opciones. Desplegar mensaje solicitando la selección de un tipo de pertenencia.		
19	Seleccionar la opción consulta por pertenencia sin seleccionar una pertenencia, oprimir generar reporte.	Inhabilitar opciones. Desplegar mensaje solicitando la selección de un tipo de pertenencia.		

20	Seleccionar la opción consulta por pertenencia, seleccionar un tipo de pertenencia, oprimir generar gráfico.	Inhabilitar opciones. Generar un gráfico de "pie" de número de equipos de la pertenencia seleccionada vs. El número total de equipos. Desplegar en el cuadro resumen el número de equipos de la pertenencia seleccionada y el número de equipos de pertenencia diferente a la seleccionada. Desplegar en el cuadro de totales el número total de equipos.		
21	Seleccionar la opción consulta por pertenencia, seleccionar un tipo de pertenencia, oprimir generar reporte.	Inhabilitar opciones. Generar un reporte con los datos generales de cada uno de los equipos de la pertenencia seleccionada.		
22	Seleccionar la opción consulta por bitácora sin seleccionar un tipo de mantenimiento, oprimir generar gráfico.	Inhabilitar opciones. Desplegar mensaje solicitando la selección de un tipo de mantenimiento.		
23	Seleccionar la opción consulta por bitácora sin seleccionar un tipo de mantenimiento, oprimir generar reporte.	Inhabilitar opciones. Desplegar mensaje solicitando la selección de un tipo de mantenimiento.		
24	Seleccionar la opción consulta por bitácora, seleccionar un tipo de mantenimiento, oprimir generar gráfico.	Inhabilitar opciones. Generar un gráfico de "pie" de número de equipos del tipo de mantenimiento seleccionado vs. El número total de equipos en mantenimiento. Desplegar en el cuadro resumen el número de equipos con el tipo de mantenimiento seleccionado y el número de equipos en otro tipo de mantenimiento diferente al seleccionado. Desplegar en el cuadro de totales el número total de equipos en mantenimiento.		
25	Seleccionar la opción consulta por bitácora, seleccionar un tipo de mantenimiento, oprimir generar reporte.	Inhabilitar opciones. Generar un reporte con los datos generales de cada uno de los equipos en el tipo de mantenimiento seleccionado.		
26	Seleccionar la opción consulta por proyecto sin seleccionar un proyecto, oprimir generar gráfico.	Inhabilitar opciones. Desplegar mensaje solicitando la selección de un proyecto.		
27	Seleccionar la opción consulta por proyecto sin seleccionar un proyecto, oprimir generar reporte.	Inhabilitar opciones. Desplegar mensaje solicitando la selección de un proyecto.		
28	Seleccionar la opción consulta por proyecto, seleccionar un tipo de proyecto, oprimir generar gráfico.	Inhabilitar opciones. Mostrar un cuadro de texto con el nombre del proyecto de la clave seleccionada. Generar un gráfico de "pie" de número de CPU's asignados al proyecto seleccionado vs. El número total de CPU's. Desplegar en el cuadro resumen el número de CPU's asignados al proyecto seleccionado y el número de CPU's no asignados al proyecto seleccionado. Desplegar en el cuadro de totales el número total de CPU's		
29	Seleccionar la opción consulta por proyecto, seleccionar un tipo de proyecto, oprimir generar reporte.	Inhabilitar opciones. Generar un reporte con los datos generales de cada uno de los CPU's asignados al proyecto seleccionado.		
30	Seleccionar el botón "limpiar" de la barra de herramientas.	Regresar la pantalla al estado inicial para realizar una nueva consulta.		
31	Seleccionar el botón "salir" de la barra de herramientas.	Regresar al menú principal.		

Sistema: Sistema de Inventario de Equipo Softtek México (SIESM).				
Módulo: Catálogos		**Pantalla:** Mantenimiento a Catálogos.		
Caso	**Descripción del caso**	**Resultado Esperado**	**Resultado Obtenido**	**Observaciones**
1	Acceder a la pantalla de mantenimiento a catálogos.	Botón "buscar" y "salir" de la barra de herramientas, así como las opciones de catálogos habilitados; los demás controles inhabilitados.		
2	Seleccionar la opción Equipo-Componente	Desplegar la pantalla Catálogo de Equipo-Componentes.		
3	Seleccionar una opción diferente de Equipo-Componente.	Limpiar grid de información del catálogo.		
4	Oprimir el botón "buscar" de la barra de herramientas. Seleccionar una opción diferente a Equipo-Componente.	Habilitar todos los botones de la barra de herramientas. Desplegar los registros existentes en el catálogo seleccionado. Habilitar el grid de información del catálogo.		
5	Seleccionar un registro del grid de información del catálogo.	Desplegar la información del registro seleccionado en la parte de datos del catálogo. Habilitar el campo descripción (piso/ oficina en el caso de ubicación).		
6	Oprimir el botón "nuevo" de la barra de herramientas.	Habilitar los campos clave y descripción (piso/ oficina en el caso de ubicación).		
7	Capturar una clave de más de 5 dígitos.	No permitir la captura de más de 5 dígitos.		
8	Capturar una descripción de más de 50 caracteres.	No permitir la captura de más de 50 caracteres.		
9	Capturar una ubicación de más de 15 caracteres.	No permitir la captura de más de 20 caracteres.		
10	Seleccionar el botón "guardar" de la barra de herramientas. Capturar una clave alfanumérica.	Desplegar un mensaje indicando que la clave debe ser numérica.		
11	Seleccionar el botón "guardar" de la barra de herramientas. No capturar descripción.	Desplegar un mensaje indicando que se debe capturar una descripción.		
12	Seleccionar el botón "guardar" de la barra de herramientas. Clave ya existente.	Desplegar un mensaje preguntando si se quiere actualizar la clave existente. Si se confirma, actualizar los cambios del registro en el catálogo. Si se cancela, no alterar información.		
13	Seleccionar el botón "guardar" de la barra de herramientas. Clave nueva.	Insertar el nuevo registro al catálogo.		
14	Seleccionar el botón "eliminar" de la barra de herramientas sin seleccionar ningún registro del grid.	Desplegar un mensaje indicando que se debe seleccionar un registro del grid.		
15	Seleccionar el botón "eliminar" de la barra de herramientas seleccionando un registro del grid.	Desplegar un mensaje preguntando si se quiere borrar el registro. Si se confirma, eliminar el registro del catálogo. Si se cancela, el catálogo no sufre cambios. Si el elemento está relacionado con otras aplicaciones, éste no será borrado.		
16	Seleccionar el botón "imprimir" de la barra de herramientas.	Generar un reporte con los elementos de los catálogos.		
17	Seleccionar el botón "salir" de la barra de herramientas.	Regresar al menú principal.		

Sistema: Sistema de Inventario de Equipo Softtek México (SIESM).				
Módulo: Catálogos		**Pantalla:** Mantenimiento a Catálogos.		
Caso	**Descripción del caso**	**Resultado Esperado**	**Resultado Obtenido**	**Observaciones**
1	Acceder a la pantalla de mantenimiento a catálogos.	Botón "buscar" y "salir" de la barra de herramientas, así como las opciones de catálogos habilitados; los demás controles inhabilitados.		
2	Seleccionar la opción Equipo-Componente	Desplegar la pantalla Catálogo de Equipo-Componentes.		
3	Seleccionar una opción diferente de Equipo-Componente.	Limpiar grid de información del catálogo.		
4	Oprimir el botón "buscar" de la barra de herramientas. Seleccionar una opción diferente a Equipo-Componente.	Habilitar todos los botones de la barra de herramientas. Desplegar los registros existentes en el catálogo seleccionado. Habilitar el grid de información del catálogo.		
5	Seleccionar un registro del grid de información del catálogo.	Desplegar la información del registro seleccionado en la parte de datos del catálogo. Habilitar el campo descripción (piso/ oficina en el caso de ubicación).		
6	Oprimir el botón "nuevo" de la barra de herramientas.	Habilitar los campos clave y descripción (piso/ oficina en el caso de ubicación).		
7	Capturar una clave de más de 5 dígitos.	No permitir la captura de más de 5 dígitos.		
8	Capturar una descripción de más de 50 caracteres.	No permitir la captura de más de 50 caracteres.		
9	Capturar una ubicación de más de 15 caracteres.	No permitir la captura de más de 20 caracteres.		
10	Seleccionar el botón "guardar" de la barra de herramientas. Capturar una clave alfanumérica.	Desplegar un mensaje indicando que la clave debe ser numérica.		
11	Seleccionar el botón "guardar" de la barra de herramientas. No capturar descripción.	Desplegar un mensaje indicando que se debe capturar una descripción.		
12	Seleccionar el botón "guardar" de la barra de herramientas. Clave ya existente.	Desplegar un mensaje preguntando si se quiere actualizar la clave existente. Si se confirma, actualizar los cambios del registro en el catálogo. Si se cancela, no alterar información.		
13	Seleccionar el botón "guardar" de la barra de herramientas. Clave nueva.	Insertar el nuevo registro al catálogo.		
14	Seleccionar el botón "eliminar" de la barra de herramientas sin seleccionar ningún registro del grid.	Desplegar un mensaje indicando que se debe seleccionar un registro del grid.		
15	Seleccionar el botón "eliminar" de la barra de herramientas seleccionando un registro del grid.	Desplegar un mensaje preguntando si se quiere borrar el registro. Si se confirma, eliminar el registro del catálogo. Si se cancela, el catálogo no sufre cambios. Si el elemento está relacionado con otras aplicaciones, éste no será borrado.		
16	Seleccionar el botón "imprimir" de la barra de herramientas.	Generar un reporte con los elementos de los catálogos.		
17	Seleccionar el botón "salir" de la barra de herramientas.	Regresar al menú principal.		

Aceptación del Usuario. Son las pruebas más importantes, ya que tiene que ver con que el usuario realice las pruebas del sistema, con el fin de que satisfaga sus necesidades o de la organización.

Puntos Esenciales para la Evaluación del Sistema:

Se llevan a cabo para identificar puntos débiles y fuertes del Sistema . La evaluación ocurre a lo largo de cualquiera de las siguientes cuatro dimensiones:

Evaluación operacional

Es el momento en que sé evalúa la manera en que funciona el Sistema, esto incluye su facilidad de uso, Tiempo de respuesta ante una necesidad o proceso, como se adecuan los formatos en que se presenta la Información, contabilidad global y su nivel de utilidad.

Impacto Organizacional

Para identificar y medir los beneficios operacionales para la Empresa, eficiencia en el desempeño laboral e impacto competitivo, impacto, rapidez y organización en el flujo de información interna y externa.

Desempeño del Desarrollo

Es la evaluación del Proceso de desarrollo adecuado tomando en cuentas ciertos criterios como, Tiempo y esfuerzo en el desarrollo concuerden con presupuesto y estándares y otros criterios de Administración de Proyectos. Además se incluyen la valoración de los métodos y herramientas utilizados durante el desarrollo del Sistema.

Pruebas del Sistemas

Se atenderá el riesgo asociado a su uso, puede hacerse la elección de comenzar la operación del sistema solo con una o dos personas del área (como una Prueba piloto).

Es necesario realizar las pruebas necesarias y evaluar los resultados esperados. ISO 9000 – 3 Pruebas y validación

Durante el Proceso de Implantación y Prueba se deben implementar todas las estrategias posibles para garantizar que en el uso inicial del Sistema éste se encuentre libre de problemas lo cual se puede descubrir durante este proceso y llevar a cabo las correcciones de lugar para su buen funcionamiento. (Es necesario realizar las pruebas necesarias y evaluar los resultados esperados. ISO 9000 – 3 Pruebas y validación).

Desafortuandamente la evaluación de Sistemas no siempre recibe la atención que merece, sin embargo cuando se lleva a cabo de manera adecuada proporciona muchas informaciones que pueden ayudar a mejorar la efectividad de los esfuerzos de desarrollo de aplicaciones futuras.

Conclusiones

Después del esfuerzo y la dedicación invertidos a lo largo del desarrollo de la tesis, llegar a concluir algo sobre la misma resulta difícil, especialmente por la capacidad crítica-objetiva necesaria para dicha conclusión, pero a continuación se procederá al desglose de las conclusiones obtenidas, cuya meta es apegarse a criterios objetivos, en base a las experiencias académicas y laborales aplicadas por los miembros del equipo de tesistas en este trabajo profesional.

El giro de la empresa Softtek México es el desarrollo y consultoría de sistemas, para lo cual brinda servicio a distintas empresas e instituciones. La administración y localización de equipo de cómputo para laborar resulta difícil de controlar, por lo que surgió la necesidad del manejo de información referente al inventario de la compañía. La empresa carecía del personal necesario en ese momento para hacer dicho sistema de información, por lo que se hizo la propuesta de la realización del mismo, acordándose que sólo se abarcaría el análisis, diseño y prototipo de desarrollo.

Se analizaron las necesidades de administración de asignación e inventario de equipo de cómputo del área de sistemas en la empresa de Softtek de México y se propuso la alternativa de solución mediante el desarrollo de un sistema de información que les facilite los procesos administrativos de esta área.

Se inició el desarrollo de dicho trabajo a partir de la filosofía de la empresa, valores, misión, visión, objetivos y metas, los cuales proporcionaron la panorámica de la misma, estando en contacto con las áreas involucradas en los procesos del manejo de los equipos, por lo que se recabó la información de sus requerimientos para traducirlas a sistemas de tipo computacional.

Con la aplicación de la metodología orientada a objetos en el ciclo de vida del software se logró modelar dicho sistema de información. El principal reto consistió en el manejo de los conceptos orientados a objetos, porque era la primera vez que eran tratados por parte del grupo de tesistas, pero se lograron comprender y aplicar exitosamente.

Las herramientas utilizadas para la realización de este trabajo fueron las más adecuadas después del estudio realizado por el grupo, con la orientación y apoyo de la asesora de tesis, considerando la herramientas de la metodología orientada a objetos, que nos permitieron modelar dicho sistema, especialmente el programa Rational Rose (con el que se desarrollaron la mayor parte de los diagramas), y JBuilder (con el que se construyó el front-end del sistema).

Las herramientas JBuilder y Rational Rose, nos permitieron diseñar parte del sistema, con todas las ventajas referentes a la capacidad de llegar posteriormente a la implementación de la lógica del negocio mediante el DBMS de Oracle, de tal forma que los requerimientos quedaron cubiertos, proporcionando un sistema centralizado, seguro, fácil de usar y automático. Partiendo del análisis-diseño hasta el prototipo de implementación que se planteó en el trabajo.

Durante el proceso del análisis de esta tesis se tuvo contacto frecuente con el usuario, para entender sus requerimientos, con base en ello se diseñó este sistema apegándose lo más posible a sus necesidades dado que hay procesos en el área que están en estudio y no se contemplaron en dicho sistema como son cartas responsivas con aspecto legal y la administración y préstamo de software, ya que los parámetros de dichas actividades no se han especificado en el área.

Por una parte se le presentaron las pantallas directamente al usuario, donde se estudiaron sus características, funciones y secuencia de las mismas, concluyendo que el sistema es operable y amigable para el manejo de información.

El trabajo operacional concluyente de este trabajo de tesis se le presentó al líder del proyecto, quien examinó los procesos con base en el análisis, diseño y prototipo, con lo cual verificó que el sistema en efecto puede desarrollarse por personal de la empresa y posteriormente implementarse en el área de soporte técnico.

Con la propuesta de solución, se garantiza que el sistema podrá correr sin dificultad sobre las plataformas planteadas y manejar grandes volúmenes de información.

La propuesta del sistema representa grandes ventajas para Softtek, ya que proporcionará el control del manejo de inventario del que actualmente carece. Esto da una poderosa herramienta de análisis para la gerencia y las personas que se ven involucradas en la toma de decisiones en la distribución y asignación de equipos de cómputo.

Al aplicar normas de calidad, se logra que el sistema cumpla con estándares internacionales para los procesos de manejo de información, así mismos con los estándares de comunicación, lo cual asegura la persistencia del mismo, así como su adaptabilidad a futuras necesidades.

El sistema es de calidad, porque satisface las necesidades en el manejo de la información de asignación e inventario de equipo de cómputo en la empresa Softtek México.

El propósito de realizar una tesis no es únicamente el de cubrir con un requisito que la Facultad de Ingeniería y nuestra Universidad solicitan, sino también concluir con la meta de terminar una carrera profesional y tener presente que cuando algo se empieza se debe concluir, un hábito que no debe olvidarse.

El Programa de Apoyo a la Titulación (PAT) es un camino positivo para que los alumnos de la Facultad de Ingeniería puedan titularse, aunque no sólo debería de aplicarse esta técnica a la gente que ya tiene tiempo de haber terminado sus estudios, sino también, a quienes se encuentran en los últimos semestres de la carrera; para que deje de ser el obstáculo final.

El haber estado en un equipo que va cumpliendo metas para cada semana, es una buena metodología, ya que se tiene una responsabilidad con el grupo de cumplir la parte que a cada uno corresponde, al mismo tiempo que estimula a cada integrante en el buen desempeño de su trabajo, así como también fue posible aplicar el concepto de sinergia para mejorar los resultados obtenidos a lo largo de la elaboración del trabajo de tesis.

Referencias

- Ávila González, Javier; 1990; Teoría de Inventarios y su Aplicación; Ed. Pax-México; Primera Edición; México.
- [AIS'77] Christopher Alexander, Sara Ishikawa, Murray Silverstein, Max Jacobson, Ingrid Fiksdahl-King and Shlomo Angel. A Pattern Language. Oxford University Press, New York, 1977.
- Becher, Hal B.; 1977; Análisis Funcional de redes de información; Ed. Limusa; México.
- Berge, Claude; 1962; Teoría de Redes y sus Aplicaciones; Ed. Continental; México.
- Black, Uyless D.; 1997; Redes de Computadores; Ed. Alfaomega; segunda edición; México.
- Booch G.; 1994; Object-Oriented Analysis and Design; Segunda Edición; Ed. Benjamín / Cummings Publishing Company, Inc; E.E. U.U.
- Date C. J.; 1998; Sistemas de bases de datos, volumen 1; Quinta Edición; Ed. Addison Wesley Longman de México S. A. de C. V.
- Edwards S., King D., Winblad A.; 1993; Software orientado a objetos; Primera Edición; Ed. Addison-Wesley Publishing Company, Inc.; E.E. U.U.
- E. Kendall, Kennet & E. Kendall, Julie; Análisis y Diseño de Sistemas; Ed. Prentice Hall; Tercera Edición.
- Fairley, R. ; 1998; Ingeniería de Software; Ed. McGraw-Hill; México.
- Gail, Linda/Christie, John; Enciclopedia de Términos de Computación; Ed. Prentice Hall.
- García Cantú, Alfonso; 1990; Enfoques Prácticos para la Planeación y Control de Inventarios; Ed. Trillas; Tercera Edición; México.
- González Sainz, Nestor; 1987; Comunicaciones y Redes de Procesamiento de Datos; Ed. McGraw-Hill; Bogotá; México.
- K. Starr, Martín, W. Miller David; 1973; Control de Inventarios Teoría y Práctica; Ed. Diana; Primera Edición; México.
- Kort, Henry F. & Silberschatz Abraham; Análisis y Diseño de Sistemas; Ed. McGraw-hill; Segunda; México.
- Leffingwell,Dean/Widrig, Don; Maging Software Requeriments; Ed. Addison-Wesley.
- L. Windblad, Ann/D. Edwars, Samuel/R. King, David; Software Orientado a Objetos; Ed. Addison-Wesley/Diaz de Santos.
- McClure, C,; 1993; Case la Automatización del Software; Ed. Addison-Wesley Iberoamericana; México.

- Mylls R.; 1994; Information Engineering, practices and techniques; Ed. J. Wiley; New York, New York, USA.
- NMX-CC-001: 1995 IMNC; Administración de la Calidad y Aseguramiento de la Calidad
- Pressman, R.; 1995; Ingeniería del Software, un Enfoque Práctico; Ed. McGraw-Hill, Tercera Edición; México.
- Rambaux, A.; 1969; Gestión Económica de los Stocks; Ed. Hispano Europea; Segunda Edición; España.
- Romero López, Álvaro Javier; 1993 Actualización de Inventarios y Costo de Ventas; Ed. Ecasa; Primera Edición; México.
- Rumbaugh J., Blaha M., Premerlani W., Eddy F., Lorensen W.; 1999; Modelado y Diseño Orientado a Objetos; Cuarta Reimpresión; Ed. Prentice Hall Internacional; España.
- S. Pressman, Roger; Ingeniería del Software; Ed. McGraw-Hill; Cuarta Edición.
- Sunshine, Carl A.; Computer Network Architectures and Protocols; Ed. Plenum Press; New York.
- SYBASE; 1995; Introduction to SYBASE Client/Server Architecture; Student guide; United States of America.
- Tanenbaum, Andrews; 1981; Redes de Ordenadores; Ed. Prentice may; segunda edición; México.
- Ullman J. D.; 1982; Principles of Database Systems; Second Edition; Computer Science Press, Inc.; United States of America.
- Yourdon E; 1996; Análisis estructurado moderno; Ed. Prentice Hall; México.
- http://www.cs.ualberta.ca/~pfiguero/soo
- http://gidis.ing.unlpam.edu.ar/personas/glafuente
- http://www.rational.com/rose/
- http://www.softtek.com
- http://www.monografias.com
- http://www.sun.com