



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
CAMPUS ARAGON

SISTEMA DE ADMINISTRACION Y CONTROL DE INVENTARIOS

T E S I S

QUE PARA OBTENER EL TITULO DE
INGENIERO EN COMPUTACION

PRESENTA:

DAVID SOLIS ALCANTAR

ASESOR:

ING. LILIANA HERNANDEZ CERVANTES



MEXICO D. F.

2004



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

ESTA TESIS NO SALE
DE LA BIBLIOTECA

SISTEMA DE ADMINISTRACIÓN Y CONTROL DE INVENTARIOS

Autorizo a la Dirección General de Bibliotecas de la UNAM a difundir en formato electrónico e impreso el contenido de mi trabajo recepcional.

NOMBRE: David Sales

Alcantara

FECHA: 21 / Enero / 2004

FIRMA: DS

*A mis padres y hermanos y a todos los que me han ayudado a ser quien soy ahora y
han moldeado quien espero ser en el futuro*

CONTENIDO

Introducción.....	xi
I Antecedentes.....	1
1.1 Aspectos Generales.....	2
Historia de las Tecnologías de Información.....	2
Utilidad de las Tecnologías de Información.....	2
Inventarios.....	3
Sistema de Administración y Control de Inventarios.....	3
1.2 Conceptos de Sistemas de Información.....	5
Definición y Objetivos de un Sistema.....	5
Actividades Básicas de un Sistema de Información.....	6
Tipos y Usos de los Sistemas de Información.....	7
Metodología para el Desarrollo de Sistemas.....	7
Ciclo de Vida de los Sistemas de Información.....	8
Método del Ciclo de Vida de los Sistemas de Información.....	9
1.3 Conceptos de Bases de Datos.....	13
Definiciones Básicas.....	13
Elementos de la Base de Datos.....	13
1.4 Beneficios de Utilizar Sist. de Información en el Manejo de los Negocios.....	15
Ventajas que brinda el SAI.....	15
1.5 Administración y Control de Inventarios.....	17
Definición de Inventario.....	17
Importancia del Inventario.....	17
Los inventarios y la toma de decisiones.....	18
II. Análisis de Los Requerimientos del Sistema.....	19
II.1 Análisis de Sistemas con UML.....	20
Conceptos Básicos.....	20
UML.....	22
Diagramas de UML.....	23
II.2 Identificación de las necesidades.....	27
Giro de la Empresa.....	27
Necesidades.....	28
Problemática y Requerimientos.....	29
II.3 Objetivos y Plan de Actividades.....	30
Objetivo General.....	30
Objetivos Específicos.....	30
Factores de Éxito.....	31
Plan de Actividades.....	32
II.4 Viabilidad.....	33
Factibilidad Técnica.....	33
Factibilidad Económica.....	35
Factibilidad Operacional.....	36

III. Diseño de la Base de Datos	37
III. 1 Antecedentes de Bases de Datos.....	38
Definición de Bases de Datos.....	38
Base de Datos Relacional.....	39
Ventajas del Enfoque de Base de Datos.....	39
Seguridad de la Base de Datos.....	40
Reglas de Normalización.....	42
Reglas de Codd.....	43
Modelo Relacional.....	44
III. 2 Lenguaje SQL.....	45
Lenguaje de Definición de Datos.....	46
Lenguaje de Manipulación de Datos.....	46
Lenguaje de Control de Datos.....	46
Selección de Registros.....	47
Inserción de Registros.....	51
Eliminación de Registros.....	52
Actualización de Registros.....	53
Integridad de los Datos.....	53
III.3 Análisis y Diseño de la Base de Datos.....	55
Pasos para Modelar la Base de Datos.....	55
Diccionario de Datos del SAI.....	55
Diagrama Entidad-Relación del SAI.....	56
IV. Diseño de la Aplicación	58
IV.1 Diseño del Sistema.....	59
Diseño General.....	59
Diseño de la Ventanas de Captura.....	65
Búsquedas.....	68
Diseño de los Reportes.....	69
IV. 2 Desarrollo de Software.....	71
Lenguaje de Programación Visual Basic.....	71
Ambiente de Desarrollo.....	73
Estructura de un Proyecto de Visual Basic.....	74
Variables y Tipos de Datos.....	75
Procedimientos.....	76
Estructuras de Control.....	77
Objetos.....	77
Programación Orientada a Objetos.....	78
Vistas y Procedimientos Almacenados.....	80
Programa de Instalación.....	80
IV.3 Pruebas.....	81
Prueba del Sistema.....	81
Plan de Pruebas.....	81
Puntos que se revisaron en las Pruebas.....	82

V. Implementación.....	85
V.1 Especificaciones.....	86
Requerimientos de software para instalar el SAI 1.0.....	86
Equipos Cliente.....	86
Servidor.....	87
Requisitos para instalar SQL Server 2000.....	87
V.2 Instalación.....	88
Instalación de SQL Server 2000.....	88
Instalación del SAI 1.0.....	93
V.3 Configuración.....	95
Configurar el Servidor de Base de Datos.....	95
Configurar el cliente.....	97
V.4 Carga de Datos.....	98
Creación de la Base de Datos para SAI.....	98
Recuperación de la Estructura de la Base de Datos.....	99
V.5 Depuración de Errores.....	103
Tipos de Errores.....	103
Manejo de Errores.....	103
Rutina de Errores.....	103
Errores Lógicos.....	105
V.6 Puesta a Punto.....	106
Infraestructura.....	106
Configuraciones.....	106
Asignación de Roles.....	106
Capacitación y Soporte Técnico.....	106
Mejoras Futuras.....	107
Conclusiones.....	108
Apéndice A. Diagramas y Casos de Uso.....	109
Apéndice B. Diccionario de Datos.....	159
Apéndice C. Glosario.....	167
Bibliografía.....	172

INTRODUCCIÓN

OBJETIVO

El propósito principal para desarrollar un sistema de administración de inventarios es mantener al día la información de las existencias de aquellos productos ofrecidos dentro de un establecimiento comercial para, de esta forma, evitar negar bienes, que por no estar en existencia, se dejan de vender.

Uno de los objetivos principales del presente proyecto es integrar la información correspondiente al registro de compras y ventas de “Electromateriales Alfa”, un negocio especializado en la comercialización de materiales eléctricos, para automatizar el manejo de sus inventarios.

JUSTIFICACIÓN

Existen varios problemas por resolver en esta empresa. El primero y más importante es el hecho de que constantemente se presenta la negación de un bien a los consumidores por no tenerlos en existencia. Esto provoca la pérdida de clientes por no cubrir sus necesidades, y se da la impresión de ser un negocio que se preocupa poco por proporcionar un buen servicio. Este problema ocurre, principalmente, por la falta de un control de inventarios por medio del cual sea posible conocer con anticipación los productos próximos a escasear y que, al mismo tiempo, permita tener el conocimiento de cuales son los productos más vendidos para mantener constantemente estos artículos en existencia.

BREVE DESCRIPCIÓN

El presente trabajo fue planteado a partir de los objetivos y requerimientos de esta empresa para agilizar y mejorar sus operaciones comerciales.

En el capítulo I se presenta un panorama general de los sistemas de información, su historia y su importancia para el manejo de inventarios, los conceptos elementales de las bases de datos, los beneficios de utilizar el sistema de Administración y Control de Inventarios y algunas definiciones básicas sobre inventarios.

En el capítulo II se cubren temas relacionados con el análisis del sistema, tales como la descripción general del Lenguaje de Modelado Unificado, la identificación de las necesidades de la empresa a la que se desarrolló el sistema, los objetivos específicos, el plan de actividades y la viabilidad del proyecto.

Dentro del capítulo III se explica el diseño de la base de datos. Se exponen desde los conceptos básicos de las bases de datos hasta llegar al diagrama Entidad-Relación.

En el capítulo IV se detalla la forma en que se realiza el diseño de la aplicación considerando el diseño, el desarrollo del software y las pruebas.

El capítulo V trata sobre la forma en la que se implementó el sistema: las especificaciones, la instalación, la configuración necesaria para el funcionamiento adecuado, la carga de datos inicial, la depuración de errores y una descripción de la

puesta a punto. Finalmente, en el apéndice A se presentan los diagramas de casos de uso, en el apéndice B el diccionario de datos y en el apéndice C el glosario.

HIPÓTESIS

La tarea de crear software no es trivial y se requiere de un análisis minucioso de las necesidades previas para llegar a tener un buen producto, un producto de calidad que satisfaga las necesidades de aquellas personas que lo utilizarán. El desarrollo de un sistema informático es un proceso que requiere de una constante retroalimentación entre los usuarios y los involucrados en su creación para lograr los objetivos para los que fue diseñado.

La problemática que presenta esta empresa servirá para hacer el análisis del sistema de tal forma que se traduzcan en soluciones concretas que ayuden a mejorar la organización de sus procesos internos. La empresa requiere mejorar su proceso de control interno del manejo de sus inventarios y llevar a cabo un mejor registro de sus operaciones diarias de compra y venta de mercancías.

Para conseguir un software o sistema de calidad se requiere utilizar alguna metodología que cubra las etapas por las que pasa todo sistema informático: análisis, diseño, desarrollo, pruebas, implantación y mantenimiento. A lo largo del presente trabajo veremos cada una de estas etapas para el desarrollado del Sistema de Administración y Control de Inventarios (SAI).

I
ANTECEDENTES

I.1 ASPECTOS GENERALES

HISTORIA DE LAS TECNOLOGÍAS DE INFORMACIÓN

A lo largo del siglo XX la computación creció a ritmos realmente sorprendentes, desde los primeras mainframes utilizados para descifrar códigos secretos en la segunda guerra mundial hasta la creación de computadoras capaces de descifrar el genoma humano. Este crecimiento en la tecnología computacional se reflejó tanto en los hogares como en la industria donde día a día se hicieron más comunes estas máquinas para desempeñar todo tipo de tareas.

En el transcurso del siglo anterior también se gestó el desarrollo de los lenguajes de programación, desde el lenguaje C, con la programación estructurada, hasta llegar a lenguajes como C++ con la programación orientada a objetos. De la misma forma, el desarrollo de nuevos sistemas operativos permitió la fácil manipulación de estos equipos.

Por otro lado, los protocolos de comunicación hicieron más eficiente la comunicación de equipos de cómputo con arquitecturas diferentes dando inicio a Internet.

Hoy en día contamos con lenguajes de programación multiplataforma, amplio ancho de banda en las telecomunicaciones y equipos con mayor capacidad de proceso y equipos económicos que nos permite aprovechar estas ventajas tecnológicas para nuestro propio beneficio.

UTILIDAD DE LAS TECNOLOGÍAS DE INFORMACIÓN

En la actualidad el software es utilizado en muchas áreas de la economía diaria, por ejemplo, la utilización de las tecnologías de información en un entorno mercantil nos ayuda a llevar una buena administración de nuestros inventarios ya que nos brinda automatización en muchos de los procesos que involucran el manejo de los recursos disponibles y nos permite analizar la información que generan para aplicarla en la toma de decisiones.

El comercio requiere de un uso adecuado de los inventarios para evitar pérdida de ventas y como consecuencia de clientes por negar un bien o producto. El manejo de inventarios es indispensable para evitar ofrecer productos que en realidad no se tienen en existencia. La negación de un bien, en este contexto, sobre todo cuando el bien se paga con anticipación, puede llevar a consecuencias serias en la continuidad de un negocio.

Para competir exitosamente en el mercado actual se necesita del conocimiento de los clientes, sus gustos y preferencias; la habilidad para recabar y procesar esa información oportunamente y ser lo suficientemente ágil para tomar decisiones se basa en el análisis de los datos que se hace con la ayuda de los sistemas de información. Un sistema de control de inventarios le da al comerciante mayor cercanía al cliente al igual que un mayor control sobre recursos.

Un sistema capaz de realizar las tareas de punto de venta y control de inventarios proporciona un mejor control para los negocios. Éste consiste en un programa integral que gestiona las ventas, las compras, el inventario, los clientes, los proveedores, la facturación y, en general, los aspectos relacionados con la operación del negocio.

INVENTARIOS

Los inventarios son recursos utilizables que se encuentran almacenados para su uso posterior en un momento determinado. Algunos autores los definen simplemente como bienes ociosos almacenados en espera de ser utilizados. Otros autores los definen como un recurso de vital importancia para el funcionamiento de una organización o empresa.

El manejo de inventarios ha sido, a lo largo de la historia, trascendente en el desarrollo de la civilización. Las grandes culturas han dependido de su control para subsistir, tal es el caso de los antiguos egipcios, quienes sabiendo que se acercaban periodos de sequía, almacenaban los granos necesarios para sobrevivir en tiempos difíciles.

El adecuado control de inventarios es un factor de éxito en cualquier organización que requiera de un intercambio de bienes o servicios. Pero este control no sólo es importante a gran escala. Cada día, en cada hogar, se requiere, por lo menos de forma rudimentaria, llevar el control de lo que poseemos y de lo que necesitamos.

En el mundo de los negocios el manejo de los inventarios es, sin lugar a dudas, un elemento indispensable para el buen desarrollo de la empresa, si éste se efectúa incorrectamente la posibilidad de tener problemas de abastecimiento o grandes costos es muy alta. Vivimos en una época donde lo único constante es el cambio y si no asimilamos esta realidad la probabilidad de dejar de ser competitivo y salir del mercado aumenta.

Existe un lema que dice: "quien compra bien, vende o produce bien", esto quiere decir que el tener una buena política de compras, permite un manejo fluido a la empresa y, al mismo tiempo, disminuir sus costos, lo que obviamente mejorará su rentabilidad. Debido a lo anterior es necesario controlar los inventarios desde el momento en que se realiza la compra.

SISTEMA DE ADMINISTRACIÓN Y CONTROL DE INVENTARIOS

El Sistema de Administración y Control de Inventarios (SAI) es un sistema informático que nos ayuda en el manejo de nuestros recursos almacenados apoyado en un uso sencillo y homogéneo de los catálogos involucrados en el proceso de compra-venta.

El SAI nos permite tener un estricto control en las compras y en las ventas realizadas, manejando niveles de seguridad para cada puesto dentro del negocio, esto es, crea perfiles o entornos de trabajo dependiendo del nivel de autorización que se tenga, tales como permisos de vendedor, de capturista o de administrador. Además, el SAI nos brinda un amplio conjunto de reportes en los que podemos analizar la información de una forma sencilla y que apoyan en la toma de decisiones.

A lo largo de estos capítulos analizaremos la forma en la que un comercio puede tener el completo control de sus inventarios aplicando tecnologías de información con el Sistema de Administración y Control de Inventarios (SAI).

I.2 CONCEPTOS DE SISTEMAS DE INFORMACIÓN

DEFINICIÓN Y OBJETIVOS DE UN SISTEMA.

Si definimos un sistema como un conjunto de elementos íntimamente interrelacionados, podemos agregar que los sistemas artificiales casi siempre han sido creados para alcanzar propósitos u objetivos, por lo tanto, un sistema es un plan de diseño o método de organización cuya finalidad es lograr algún propósito.

Dos características fundamentales de los sistemas son las siguientes:

- a) Sinergia, es decir, el sistema tomado globalmente posee propiedades que no tienen sus partes, tomadas individualmente. Esto es, el sistema tiene una capacidad de actuación mayor o menor que la de la suma de sus partes.
- b) Finalidad común, es decir, la posibilidad de utilizar diferentes caminos o medios para llegar a un mismo fin, lo cual le permite una gran variedad de comportamientos.

Se puede definir un sistema desde varios puntos de vista:

Como un conjunto de partes: Un sistema tiene más de un elemento. Un volante no es un sistema, pero es una parte vital de un sistema muy conocido que se llama automóvil.

Cómo partes integradas: Debe existir una relación lógica entre las partes de un sistema. Los sistemas mecánicos y electrónicos, como son las lavadoras los juegos de video, tienen muchos componentes que trabajan juntos. Un sistema de administración de personal cuenta con procedimientos integrados para contratar y adiestrar empleados.

Todos los elementos del sistema deben estar ligados y controlados de manera que se logre el objetivo del sistema. Al final de cuentas el propósito de un sistema es el de lograr algún objetivo en común.

Dentro de los sistemas tenemos a los sistemas de información. Un sistema de información es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio. También se puede definir como el conjunto total de procedimientos, operaciones, funciones y difusión de datos o información en una organización.

Existen partes fundamentales de un sistema de procesamiento electrónico de datos, éstos son de naturaleza diversa:

- El equipo computacional, es decir, el hardware es necesario para que el sistema de información pueda operar. Lo constituyen las computadoras y el equipo periférico que puede conectarse a ellas.
- El recurso humano que interactúa con el sistema de información, el cual está formado por las personas que utilizan el sistema, alimentándolo con datos o utilizando los resultados que genere.

- Los datos o información fuente que son introducidos en el sistema, son todas las entradas que necesita el sistema de información para generar como resultado la información que se desea.
- Los programas que son procesados y producen diferentes tipos de resultados. Los programas son la parte del software del sistema de información que hará que los datos de entrada introducidos sean procesados correctamente y generen los resultados que se esperan.

ACTIVIDADES BÁSICAS DE UN SISTEMA DE INFORMACIÓN

Un sistema de información realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información. A continuación se definirán cada una de estas actividades.

Entrada de Información.

La entrada es el proceso mediante el cual el sistema de información toma los datos que requiere para procesar la información. Las entradas pueden ser manuales o automáticas. Las manuales son aquellas que se proporcionan en forma directa por el usuario, mientras que las automáticas son datos o información que provienen o son tomados de otros sistemas o módulos. Esto último se denomina interfases automáticas. Así, un Sistema de Control de Clientes podrá tener una interfase automática de entrada con el Sistema de Facturación, ya que toma las facturas que genera o elabora el Sistema de Facturación como entrada al Sistema de Control de Clientes.

Las unidades típicas de entrada de datos a las computadoras son las terminales, las cintas magnéticas, las unidades de disquete, los códigos de barras, los escáneres, la voz, los monitores sensibles al tacto, el teclado y el ratón, entre otras.

Almacenamiento de información.

El almacenamiento es una de las actividades o capacidades más importantes que tiene una computadora, ya que a través de esta propiedad el sistema puede recordar la información guardada en la sesión o proceso anterior. Esta información suele ser almacenada en estructuras de información denominadas archivos. La unidad típica de almacenamiento son los discos magnéticos o discos duros, los discos flexibles o disquetes y los discos compactos. Sin embargo, existen otras formas de almacenamiento.

Procesamiento de Información.

Es la capacidad del sistema de información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema o bien con datos que están almacenados. Esta característica de los sistemas permite la transformación de datos fuente en información que puede ser utilizada para la toma de decisiones, lo que hace posible, por ejemplo, entre otras cosas, que un tomador de decisiones genere una proyección financiera a partir de los datos que contiene un estado de resultados o un balance general de un año base.

Salida de Información.

La salida es la capacidad de un sistema de información para sacar la información procesada o bien datos de entrada al exterior. Las unidades típicas de salida son las impresoras, terminales, disquetes, cintas magnéticas, la voz, los graficadores y los plotters, entre otros.

Es importante aclarar que la salida de un Sistema de Información puede constituir la entrada a otro sistema de información o módulo. En este caso, también existe una interfase automática de salida. Por ejemplo, el Sistema de Control de Clientes tiene una interfase automática de salida con el Sistema de Contabilidad, ya que genera las pólizas contables de los movimientos procesados de los clientes.

TIPOS Y USOS DE LOS SISTEMAS DE INFORMACIÓN.

Durante los próximos años, los Sistemas de Información cumplirán tres objetivos básicos dentro de las organizaciones:

1. Automatización de procesos operativos.
2. Proporcionar información que sirva de apoyo al proceso de toma de decisiones.
3. Lograr ventajas competitivas a través de su implantación y uso.

Los Sistemas de Información que logran automatización de procesos operativos dentro de una organización, son llamados frecuentemente Sistemas Transaccionales, ya que su función primordial consiste en procesar transacciones tales como pagos, cobros, pólizas, entradas, salidas, etcétera. Por otra parte, los Sistemas de Información que apoyan el proceso de toma de decisiones son los Sistemas de Soporte a la Toma de Decisiones (DSS), Sistemas para la Toma de Decisiones de Grupo (GDSS), Sistemas Expertos de Soporte a la Toma de Decisiones (EDSS) y Sistemas de Información para Ejecutivos (EIS). El tercer tipo de sistemas, de acuerdo con su uso u objetivos que cumplen, es el de los Sistemas Estratégicos, los cuales se desarrollan en las organizaciones con el fin de lograr ventajas competitivas, a través del uso de la tecnología de información.

METODOLOGÍA PARA EL DESARROLLO DE SISTEMAS

La metodología del desarrollo de sistemas es el camino que siguen los analistas de sistemas al realizar su trabajo. Se emplea el término genérico de analista de sistemas para describir a la persona que tiene la responsabilidad principal de conjuntar los componentes estructurales, dándoles forma y sustancia en conformidad con las fuerzas del diseño para construir sistemas de información exitosos.

En una compañía pequeña, el analista quizás no sólo diseñará el sistema de información, sino que también hará la programación y operará la computadora. En una compañía grande el analista de sistemas puede preparar las especificaciones del diseño que se dan a los técnicos, como los programadores, diseñadores de formas y especialistas en comunicaciones. El analista de sistemas coordinará las tareas de todos los especialistas para la implementación final del sistema completa.

CICLO DE VIDA DE LOS SISTEMAS DE INFORMACIÓN.

El concepto de ciclo de vida de un sistema de información es medular en las investigaciones de sistemas. Durante su desarrollo, cada sistema se mueve a través de varias fases de un ciclo de vida, después del cual sólo funciona por varios años con un mínimo mantenimiento. El sistema se deteriora gradualmente hasta el punto en que cesa de funcionar por completo y se comienza un nuevo ciclo de vida con el desarrollo de un nuevo sistema.

La siguiente figura muestra el ciclo de vida de un sistema. La figura muestra cinco fases. Éstas son: la fase de estudio preliminar, la fase de análisis de sistemas, la fase de diseño de sistemas, la fase de implantación, la cual incluye una actividad separada llamada "auditoría posterior". Los ciclos de vida de sistemas varían en gran manera en términos de longitud, pero por lo regular el ciclo de vida de un sistema de información está en el rango de 3 a 8 años. Las primeras cuatro fases de este ciclo de vida pueden llamarse las "fases de investigación de sistemas".

FASES DE UN CICLO DE VIDA DE UN SISTEMA DE INFORMACIÓN



El concepto de ciclo de vida está relacionado con otro concepto importante, el de grupos profesionales de desarrollo de sistemas de información. Las organizaciones grandes y medianas por lo general tienen especialistas de investigación de sistemas de tiempo completo, incluyendo programadores y analistas. Por lo regular, para todas, excepto para las investigaciones de sistemas menores, se formará un equipo al momento de iniciar la fase de estudio preliminar.

Es probable que el equipo crezca al entrar en la fase del análisis y que su composición se modifique de alguna forma para entrar en la fase del diseño de sistemas. Después el equipo, quizá modificado de nuevo en su composición, implanta el sistema diseñado. En cuanto cada miembro del equipo termina las responsabilidades que le fueron asignadas, se le asignan otras responsabilidades del proyecto o se le asigna a otra investigación de sistemas.

Por tanto, los equipos de proyectos son dinámicos: se forman y reforman de continuo para participar en diferentes aspectos de la investigación de sistemas o en otras investigaciones de sistemas. Existe un reciclamiento continuo del personal profesional de sistemas a través de una serie de investigaciones de sistemas. Es útil mencionar brevemente cada una de las fases de la investigación de sistemas.

Dentro de cada fase se incluyen las principales actividades o tareas. Las primeras cuatro fases están dirigidas a proporcionar valores específicos para los componentes estructurales. La última fase se ocupa de que los componentes estructurales sean operacionales. Independientemente del número o nombres de las fases o etapas, la metodología del desarrollo de sistemas racionaliza y asigna una rutina al proceso de construcción de sistemas de información. Su sello distintivo lo forman sus fases discretas. La meta principal de la metodología del desarrollo de sistemas es reducir los inicios falsos, reciclamiento indebido, y callejones sin salida.

Además, aumenta la probabilidad de que el sistema que se construya e instale finalmente sea el que los usuarios desean y necesitan. Pero de ninguna manera la metodología del desarrollo de sistemas deberá ahogar la creatividad o la innovación. Proporciona un faro guía para el recorrido a lo largo de un sistema de información. El que tan excitante, productivo y creativo desee el analista de sistemas que sea este recorrido depende, en gran medida, del propio analista.

MÉTODO DEL CICLO DE VIDA DE LOS SISTEMAS DE INFORMACIÓN

El método del ciclo de vida para desarrollo de sistemas es el conjunto de actividades que los analistas, diseñadores y usuarios realizan para desarrollar e implantar un sistema de información. Este método consta de las siguientes actividades:

- Investigación preliminar
- Determinación de los requerimientos del sistema
- Diseño del sistema
- Desarrollo de software
- Prueba del sistema.
- Implementación, integración y evaluación de sistemas.

Investigación preliminar

La solicitud para recibir ayuda de un sistema de información puede originarse por varias razones; sin importar cuales sean estas, el proceso se inicia siempre con la petición de una persona (administrador, empleado o especialista en sistemas). La investigación preliminar se divide en tres partes: aclaración de la solicitud, estudio de factibilidad y aprobación de la solicitud.

Aclaración a la solicitud. Antes de considerar cualquier investigación de sistemas, la solicitud de proyecto debe examinarse para determinar con precisión lo que el solicitante desea, ya que el solicitante a veces está confundido y no sabe lo que desea en realidad.

Estudio de factibilidad. Existen tres tipos y son:

1. Factibilidad técnica. Es hacer un estudio tecnológico el cual determinará la tecnología existente en el mercado sobre los requerimientos del sistema.
2. Factibilidad económica. Es el estudio que nos permite conocer el costo del proyecto y determinar si lo podemos finalizar o no.
3. Factibilidad operacional. Consiste en conocer si hay recursos humanos, calificados, (si el sistema instalado va a trabajar en perfectas condiciones).

Aprobación de la solicitud. Después de aprobar la solicitud de un proyecto se estima su costo, el tiempo necesario para terminarlo y las necesidades de personal; con esta información se determina donde ubicarlo dentro de la lista existente de proyectos.

Determinación de los requerimientos del sistema.

En este paso es importante comprender todas las facetas importantes de la parte de la empresa que se encuentra bajo estudio. (Es por esta razón que el proceso de adquirir información se denomina, con frecuencia, investigación detallada.) Los analistas al trabajar con los empleados y administradores, deben estudiar los procesos de una empresa para dar respuesta a las siguientes preguntas:

1. ¿Qué es lo que se hace?
2. ¿Cómo se hace?
3. ¿Con qué frecuencia se presenta?
4. ¿Qué tan grande es el volumen de transacciones o decisiones?
5. ¿Cuál es el grado de eficiencia con el que se efectúan las tareas?
6. ¿Existe algún problema?
7. Si existe un problema, ¿qué tan serio es?
8. Si existe un problema, ¿cuál es la causa que lo origina?

Para contestar estas preguntas, el analista conversa con varias personas para reunir detalles relacionados con los procesos de la empresa, sus opiniones sobre por qué ocurren las cosas, las soluciones que proponen y sus ideas para cambiar el proceso.

Diseño del sistema

Los analistas de sistemas comienzan el proceso de diseño identificando los reportes y demás salidas que debe producir el sistema. Hecho lo anterior se determinan con toda precisión los datos específicos para cada reporte y salida. Es común que los diseñadores hagan un bosquejo del formato o pantalla que esperan que aparezca cuando el sistema esté terminado.

Lo anterior se efectúa en papel o en la pantalla de una terminal utilizando para ello algunas de las herramientas automatizadas disponibles para el desarrollo de sistemas. El diseño de un sistema también indica los datos de entrada, aquellos que serán calculados y los que deben ser almacenados. Así mismo, se escriben con todo detalle los procedimientos de cálculo y los datos individuales. Los diseñadores seleccionan las estructuras de archivo y los dispositivos de almacenamiento, tales como discos y cintas magnéticas o incluso archivos de papel. Los procedimientos que se escriben indican como procesar los datos y producir las salidas. Los documentos que contienen las especificaciones de diseño representan a éste de muchas maneras (diagramas, tablas y símbolos especiales).

La información detallada del diseño se proporciona al equipo de programación para comenzar la fase del desarrollo del software. Los diseñadores son los responsables de dar a los programadores las especificaciones de software completas y claramente delineadas. Una vez comenzada la fase de programación, los diseñadores contestan preguntas, aclaran dudas y manejan los problemas que enfrentan los programadores cuando utilizan las especificaciones de diseño.

Desarrollo de software.

Los encargados de desarrollar software pueden instalar (o modificar y después instalar) software comprado a terceros o escribir programas diseñados a la medida del solicitante. La elección depende del costo de cada alternativa, el tiempo disponible para escribir el software y de la disponibilidad de los programadores.

Por regla general, los programadores (o analistas programadores) que trabajan en las grandes organizaciones pertenecen a un grupo permanente de profesionales. Los programadores también son responsables de la documentación de los programas y de proporcionar una explicación de como y por que ciertos procedimientos se codifican en determinada forma. La documentación es esencial para probar el programa y llevar a cabo el mantenimiento una vez que la aplicación se encuentra instalada.

Prueba del sistema.

En esta fase se verifica que el software cumpla con los objetivos para que fue creado, considerando los errores lógicos.

Implantación

La implantación es el proceso de verificar e instalar nuevo equipo, entrenar a los usuarios, instalar la aplicación y construir todos los archivos de datos necesarios para utilizarla.

Dependiendo del tamaño de la organización que empleara la aplicación y el riesgo asociado con su uso, puede elegirse comenzar la operación del sistema solo en un área de la empresa (prueba piloto), por ejemplo en un departamento o con una o dos personas. Algunas veces se deja que los dos sistemas, el viejo y el nuevo, trabajen forma paralela con la finalidad de comparar los resultados.

En otras circunstancias, el viejo sistema deja de utilizarse determinado día para comenzar a emplear el nuevo al día siguiente. Cada estrategia de implantación tiene sus méritos de acuerdo con la situación que se considere dentro de la empresa. Sin importar cual sea la estrategia utilizada, los encargados de desarrollar el sistema procuran que el uso inicial del sistema se encuentre libre de problemas.

Integración

La integración de los sistemas de información tendrá que diseñarse con un acoplamiento más estrecho entre la oficina y la planta. A decir verdad, el sistema de información llegará a ser tan importante en la planta de la fábrica como en la oficina. Además, la conectividad y las comunicaciones entre y dentro de los departamentos deben ser mejores dentro de la oficina y la fábrica.

La tecnología informática estará inserta en las organizaciones y enlazada para una sincronización completa y una coordinación de las operaciones. El sistema ya no estará separado funcional y espacialmente del lugar de trabajo. Este diseño dará por resultado una malla de información para la organización. La mano derecha sabrá lo que esta haciendo la izquierda y viceversa.

Evaluación

La evaluación de un sistema se lleva a cabo para identificar puntos débiles y fuertes. La evaluación ocurre a lo largo de cualquiera de las siguientes dimensiones:

- Evaluación operacional. Valoración de la forma en que funciona el sistema, incluyendo su facilidad de uso, tiempo de respuesta, lo adecuado de los formatos de información, confiabilidad global y nivel de utilización.
- Impacto organizacional. Identificación y medición de los beneficios para la organización en áreas tales como finanzas (costos, ingresos y ganancias), eficiencia operacional e impacto competitivo. También se incluye el impacto sobre el flujo de información interno y externo.
- Opinión de los administradores. Evaluación de las actitudes de los directivos y administradores dentro de la organización así como de los usuarios finales.
- Desempeño del desarrollo. La evaluación del proceso de desarrollo de acuerdo con criterios tales como tiempo y esfuerzo de desarrollo, concuerdan con presupuestos y estándares, y otros criterios de administración de proyectos. También se incluye la valoración de los métodos y herramientas utilizados en el desarrollo.

La evaluación de sistemas no siempre recibe la atención que merece. Sin embargo, cuando se conduce en forma adecuada proporciona mucha información que puede ayudar a mejorar la efectividad de los esfuerzos de desarrollo de aplicaciones posteriores.

I.3 CONCEPTOS DE BASES DE DATOS

En esta parte veremos algunas de las definiciones del mundo de las bases de datos relacionales.

DEFINICIONES BÁSICAS

Un campo es la unidad más pequeña a la cual uno puede referirse en un programa. Desde el punto de vista del programador representa una característica de un individuo u objeto. En las bases de datos representan los atributos de una tabla, por ejemplo, si existe una tabla llamada Cliente uno de sus atributos o campos es el nombre del cliente.

Un registro es una colección de campos de iguales o de diferentes tipos de dato. Regularmente se refiere a un renglón o tupla dentro de una tabla de la base de datos. Así, continuando con el ejemplo de la tabla Cliente, un registro representa los datos relacionados de un mismo cliente, por ejemplo, su dirección, teléfono, etc., que físicamente, para el usuario, es un renglón de una tabla.

Cuando se habla de "tipo de datos" se hace referencia a las diferentes categorías de datos con los que se fijan las características de un campo dentro de una base de datos. Por ejemplo, existen datos de tipo entero, flotante, carácter, booleano, etc.

Por otro lado tenemos a los archivos que son la colección de registros almacenados siguiendo una estructura homogénea dentro de un sistema operativo. Existen muchos tipos de archivos: archivos que almacenan imágenes, video, texto y cada uno de ellos se identifica comúnmente mediante una extensión que es una serie de caracteres que se escriben en su nombre después de un punto.

ELEMENTOS DE LA BASE DE DATOS

Una base de datos, es una colección de archivos interrelacionados que son creados con un DBMS. El contenido de una base de datos engloba a la información concerniente (almacenadas en archivos) de una organización, de tal manera que los datos estén disponibles para los usuarios, una finalidad de la base de datos es eliminar la redundancia o al menos minimizarla. Los tres componentes principales de un sistema de base de datos son el hardware, el software DBMS y los datos a manejar, así como el personal encargado del manejo del sistema.

Pero una base de datos necesita ser administrada, para esto tenemos los DBMS. Un DBMS es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de una tarea específica. Además es necesario contar con una persona que tenga los conocimientos necesarios para administrar la base de datos. El DBA es la persona o equipo de personas profesionales responsables del control y manejo del sistema de base de datos, generalmente tienen experiencia en DBMS, diseño de bases de datos, sistemas operativos, comunicación de datos, hardware y programación.

El objetivo primordial de un sistema manejador base de datos es proporcionar un contorno que sea a la vez conveniente y eficiente para ser utilizado al extraer, almacenar y manipular información de la base de datos. Todas las peticiones de acceso a la base, se manejan centralizadamente por medio del DBMS, por lo que este paquete funciona como interfase entre los usuarios y la base de datos.

Una base de datos tiene cierta estructura o esquema. La estructura de base de datos se especifica por medio de un conjunto de definiciones que se expresa mediante un lenguaje especial llamado lenguaje de definición de datos (DDL).

Internamente la Base de Datos (BD) se maneja como una especie de lista ligada, pero al crecer la BD se presentan el problema de la estandarización y la redundancia de la información.

La estandarización se refiere a el como se organiza la base de datos, los tamaños de los campos que se tienen, cuantos campos se tienen, etc.

Por otro lado, la redundancia de la información es el problema de como ordenar la base de datos de tal manera que se eviten ciertas redundancias en los campos de datos, si la base de datos llega a crecer demasiado, estas redundancias ocasionan un mayor tamaño en la base de datos almacenada, lo que repercute en problemas de espacio, acceso más lento de la información, etc.

El primer problema no es resuelto de forma tan sencilla, ya que no hay un método de estandarización eficaz que sea adoptado por todos los manejadores de BD existentes, pero en el caso del segundo problema puede resolverse con una Base de Datos Relacional.

I.4 BENEFICIOS DE UTILIZAR SISTEMAS DE INFORMACIÓN EN EL MANEJO DE LOS NEGOCIOS.

El Sistema de Administración y Control de Inventarios (SAI) proporciona el control del negocio al detalle. Consiste en un Programa integrado que controla las ventas, las compras, el inventario, los clientes, los proveedores, la facturación, y en general, los aspectos relacionados con la operación del negocio de una forma automatizada.

VENTAJAS QUE BRINDA EL SAI

- El SAI brinda un conjunto completo de reportes que dan un panorama completo sobre el funcionamiento del negocio, entre los reportes encontramos las ventas realizadas por empleado, las compras hechas a los proveedores por periodos de tiempo, los artículos más vendidos y los menos vendidos en un rango de fechas, reportes de cuentas por pagar, cuentas por cobrar, entre otros más.
- Dos vistas en los reportes de catálogos de artículo, clientes, proveedores, categorías, etc.: detallada o resumida.
- Los reportes se pueden exportar a aplicaciones Web como el Internet Explorer, a hojas de cálculo como Excel y otras de aplicaciones de Windows como Word de una forma fácil y sin pérdida de información.
- Módulo de seguridad para validar los usuarios que tendrán acceso al sistema. Gestión de cuentas con varios niveles de autorización. Los niveles de autorización o roles son: administrador, capturista y vendedor. Cada uno de ellos tiene acceso a la totalidad del sistema (en el caso del administrador) o a diferentes partes del sistema relacionadas con su función dentro del negocio.
- Métodos de acceso directo que facilitan una entrada rápida a las diferentes opciones del programa ya sea por medio del ratón con la barra de herramientas configurable o por medio del teclado utilizando las letras subrayadas en los menús y con la combinación de teclas del método abreviado.
- Varias formas de hacer una misma tarea. Se tienen diferentes vistas para realizar la captura: a través de cuadrículas tipo hoja de cálculo o a través de cuadros de texto tipo formulario.
- Interfaz intuitiva que minimiza la curva de aprendizaje de cada una de las opciones del sistema. Los menús se encuentran organizados por categorías y las opciones más importantes están presentes en la barra de herramientas de la aplicación.
- Métodos de búsqueda homogénea y sencilla que permiten encontrar desde uno hasta un conjunto de registros para trabajar con los resultados de la búsqueda. Las búsquedas son eficaces ya que hacen uso de muchos criterios que se pueden emplear de forma sencilla.

- Personalización de facturas. Son completamente configurables las facturas para su impresión, esto es, se pueden definir las coordenadas en las que se desea que aparezcan los datos del cliente de los movimientos de la venta, los totales o, incluso quitar datos que no se necesiten ajustándose a las necesidades del negocio.
- Ajustable al tipo de negocio. Gracias al diseño de la base de datos y de la aplicación el SAI es adaptable a cualquier tipo de negocios que requiera controlar sus inventarios.
- Validaciones para la introducción de datos numéricos, fechas, tamaño de campos, etc. En todos los módulos del programa se desarrolló la interfaz de tal forma que los datos, antes de ser insertados en la base de datos, estén validados con el tipo de dato que les corresponde.
- Manejo de errores. Si por alguna causa externa al sistema, ya sea por fallas en la infraestructura de red, o incluso en el servidor de la base de datos se llegarán a presentar errores el SAI los detecta y manda una descripción detallada de la causa del error para que se solucione más rápidamente.
- Todas las ventajas de una aplicación cliente/servidor como el manejo de transacciones. Por ejemplo, el registro de una compra no se lleva a cabo hasta que se finaliza completamente la captura de la totalidad de los movimientos. Esto se logra gracias a la integridad de los datos que ofrece el manejador de la base de datos
- Cuenta con ayuda, de todos los módulos en formato HTML y brinda todas las posibilidades de búsqueda e impresión de un documento en este formato.
- Permite actualizar el inventario y da el número exacto de las existencias de forma automática en cualquier momento desde la compra hasta la venta de mercancía.
- Implementa la selección de los productos de múltiples formas en la captura de compras y de ventas y hace los procesos necesarios para actualizar los precios de los artículos cuando estos son adquiridos.
- Controla todos los vendedores que se desee, independiente uno del otro. Lleva el registro de sus ventas, la hora en que se realizó la operación.
- Maneja varios precios de venta por producto si llegara a haber alguna consideración especial y hace uso de descuentos tanto por artículo como por venta en general.
- Manejo de cuentas pendientes por cobrar donde se detalla cada uno de los pagos pendientes y manejo de pago a proveedores.
- Permite las devoluciones o cancelaciones de las compras y las ventas con la actualización automática del inventario, manteniendo el registro de la operación.

I.5 ADMINISTRACIÓN Y CONTROL DE INVENTARIOS

DEFINICIÓN DE INVENTARIO

En la acepción más amplia de la palabra, los inventarios son recursos utilizables que se encuentran almacenados para su uso posterior en un momento determinado. Algunos autores los definen simplemente como bienes ociosos almacenados en espera de ser utilizados. Otros autores los definen como un activo corriente de vital importancia para el funcionamiento de la empresa. Existen múltiples argumentos para justificar la tenencia o no de inventarios, de los cuales mencionaremos tan solo unos pocos.

Los argumentos a favor del manejo de inventarios es que prevén escasez, además de que es preferible ahorrar productos que dinero y permiten obtener ganancias cuando hay alzas. Otro punto importante es que facilitan desfasar (separar) los diferentes procesos de la empresa.

Sin embargo, como en todo, existen argumentos en contra del manejo de inventarios, por ejemplo se dice que inmovilizan recursos que podrían usarse mejor y esconden los problemas de la empresa

IMPORTANCIA DEL INVENTARIO

Los problemas de inventario pueden contribuir, y de hecho lo hacen, a las quiebras de los negocios. Cuando una compañía solo falla en que involuntariamente se queda sin inventario, los resultados no son agradables. Si la firma es una tienda de menudeo, el comerciante pierde la utilidad bruta de este artículo. Si la empresa es un fabricante, la falta de inventario (incapacidad de abastecer un artículo del inventario) podría en casos extremos, hacer que se detenga la producción. Por el contrario si una firma mantiene inventarios excesivos, el costo de mantenimiento adicional puede representar la diferencia entre utilidades y pérdidas. La conclusión debe ser que la administración habilidosa de los inventarios, puede hacer una contribución importante a las utilidades mostradas por la empresa.

En cualquier organización, los inventarios añaden una flexibilidad de operación que de otra manera no existiría. En fabricación, los inventarios de producto en proceso son una necesidad absoluta, a menos que cada parte individual se lleve de maquina a maquina y que estas se preparen para producir una sola parte.

Hay dos decisiones básicas de inventario que los gerentes deben hacer cuando intentan llevar a cabo las funciones de inventario recién revisadas. Estas dos decisiones se hacen para cada artículo en el inventario:

1. Qué cantidad de un artículo ordenar cuando el inventario de ese identificador se va a reabastecer.
2. Cuándo reabastecer el inventario de ese artículo.

En la mayoría de los negocios, los inventarios representan una inversión relativamente alta y producen efectos importantes sobre todas las funciones principales de la empresa. Cada funciones tiende a generar demandas de inventario diferente y a menudo incongruente:

Ventas. Se necesitan inventarios elevados para hacer frente con rapidez a las exigencias del mercado.

Producción. Se necesitan elevados inventarios de materias primas para garantizar la disponibilidad en las actividades de fabricación; y un colchón permisiblemente grande de inventarios de productos terminados facilita niveles de producción estables.

Compras. Las compras elevadas minimizan los costos por unidad y los gastos de compras en general.

Financiación. Los inventarios reducidos minimizan las necesidades de inversión (corriente de efectivo) y disminuyen los costos de mantener inventarios (almacenamiento, riesgos, etc.).

Los propósitos de las políticas de inventarios deben ser:

1. Planificar el nivel óptimo de inversión en inventarios
2. A través de control, mantener los niveles óptimos tan cerca como sea posible de lo planificado.

Los niveles de inventario tienen que mantenerse entre dos extremos: un nivel excesivo que causa costos de operación, riesgos e inversión insostenibles, y un nivel inadecuado que tiene como resultado la imposibilidad de hacer frente rápidamente a las demandas de ventas y producción (alto costo por falta de existencia).

LOS INVENTARIOS Y LA TOMA DE DECISIONES

La toma de decisiones es aún la responsabilidad más importante. A medida que se logra conocer y manejar un mayor número de instrumentos de control de inventarios la probabilidad de éxito en las empresas se incrementa.

Así como en las ventas, los empresarios y aun los comercios, han debido incluir diferentes esquemas de comercialización, donde se mezclan posibilidades de crédito, con descuentos por pago de contado, como única opción de supervivencia; los empresarios en general, deben buscar herramientas para la administración de los inventarios, que les permitan manejar estos desde diversos ángulos y estar preparados a reconsiderar en cualquier momento su uso, ya que, pretender que existe hoy, una única y mejor forma de resolver algo, es síntoma de terquedad y puede llevar a la empresa a complicaciones importantes.

Algunos afirman que se deben manejar grandes cantidades de inventarios, ya que esto significa riqueza y que es preferible guardar en insumos o productos el dinero, ya que con las alzas ocasionadas por la inflación y la devaluación, se consigue una mayor rentabilidad que en otras opciones. En el otro extremo se observan las personas que afirman que los inventarios se deben reducir a cero y manejar una política de justo a tiempo, donde los inventarios son un problema que se genera por ineficiencia gerencial. Los dos extremos pueden estar mal, ya que alguien puede tener inventarios en exceso y no tener como pagar la nómina, mientras que en el otro extremo estaría quien ha tenido que retrasar su proceso de producción por no contar en forma oportuna con los elementos requeridos para el efecto.

II

ANÁLISIS DE LOS REQUERIMIENTOS DEL SISTEMA

II.1 ANÁLISIS DE SISTEMAS CON UML

Después de haber conocido los elementos del inventario en el capítulo anterior veamos ahora algunos conceptos base sobre el análisis de los sistemas que servirán para entender el modelado empleado para crear la aplicación.

CONCEPTOS BÁSICOS

Modelo

Un modelo es una representación de algo real, en él se captan los aspectos importantes de lo que se esta representando desde cierto punto de vista por medio de abstracciones que permiten contemplar la funcionalidad principal de lo que se esta modelando para simplificar u omitir el resto.

Se hace necesario modelar los sistemas de cómputo debido a que para la solución de problemas, en un inicio, es difícil comprender la totalidad y la modelación nos permite dividir un problema complejo en problemas menores que reducen la complejidad. Un modelado nos permite visualizar un sistema desde varios puntos de vista, por ejemplo el punto de vista del cliente, del desarrollador, etc. para poder entender y dimensionar el problema adecuadamente.

Desde la etapa de modelado tenemos la oportunidad comprender la solución y probarla ya que a través de las abstracciones de las características y estructuras principales por construir se detectan fallas, inconsistencias y se pueden proveer cambios.

Cuando se esta haciendo el modelado del sistema generalmente se realiza parte de la documentación de éste para que las diferentes personas involucradas en su desarrollo tengan una fuente de donde obtener la información que se requiere para que la programación se ajuste a los requerimientos del cliente.

Proceso

Un proceso es un conjunto de actividades, métodos y prácticas para desarrollar y mantener el software y los productos asociados. Define qué hacer, cuándo y cómo hacerlo, y quién o quienes deben hacerlo.



Las fases del proceso son las siguientes:

- Inicio: Definir el alcance y objetivos del proyecto
- Elaboración: Analizar el dominio del problema, establecer una arquitectura base, desarrollar el plan y eliminar los elementos de alto riesgo del proyecto
- Construcción: Desarrollo del producto o codificación
- Transición: Introducir el producto a la comunidad de usuarios

Cada fase se concluye en punto en el tiempo en el que se deben tomar ciertas decisiones y lograr ciertas metas.

Los elementos principales que se obtienen de la fase de inicio son:

- Un documento de visión que incluye una visión general de los requerimientos del proyecto, sus características clave y sus principales restricciones
- El contexto del negocio, factores de y estimaciones financieras
- Un plan del proyecto

Los elementos principales que se obtienen de la fase de elaboración son:

- Un modelo de Casos de Uso (al menos 80%). Todos los Casos de Uso y Actores han sido identificados y la mayoría de las descripciones se han desarrollado
- Una descripción del arquitectura de software
- Un prototipo de la arquitectura

Los elementos principales que se obtienen de la fase de construcción son:

- El producto de software integrado en la plataforma adecuada
- Los manuales (de usuario y técnico)
- Una descripción de la versión actual

Los elementos principales que se obtienen de la fase de transición son:

- Una “prueba-beta” para validar el nuevo sistema contra las expectativas del usuario
- Operación paralela del sistema legado que será reemplazado
- Conversión de bases de datos operacionales
- Capacitación de los usuarios y administradores
- Presentación del producto a los equipos relacionados

UML

El Lenguaje de Modelado Unificado (UML) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Fue desarrollado en un esfuerzo para simplificar y consolidar las notaciones de desarrollo orientado a objetos que habían surgido. Surgió gracias al apoyo por el OMG, Rational Software, Microsoft, Hewlett-Packard, Oracle, Texas Instruments, MCI Systemhouse y otros.

Es el lenguaje para crear planos de software que permite visualizar la solución y facilitar la comunicación entre las partes o personas invocadas en la creación de los sistemas. Con él se puede especificar modelos más precisos y construir código base mediante herramientas como Rational Rose en varios lenguajes de programación.

UML incluye una serie de diagramas; especifica la notación para representarlos pero no describe cómo crearlos. Básicamente UML está compuesto por tres componentes que son los Elementos, las Relaciones y los Diagramas.

Modelar con UML ofrece muchas ventajas ya que define una notación expresiva y consistente que facilita la comunicación y permite detectar omisiones o inconsistencias en la etapa del análisis. Además es aplicable a sistemas sencillos y complejos y es muy popular pues se ha convertido en un estándar en la industria de construcción de software.

UML se basa en los requerimientos. Un requerimiento es una condición o característica que se debe satisfacer.

Casos de Uso

La vista de casos de uso captura el comportamiento de un sistema. Un caso de uso representa por lo tanto una secuencia de acciones que un sistema lleva a cabo para ofrecer algún resultado de valor para un actor u objeto externo que interactúa con el sistema. Los casos de uso representan toda la funcionalidad del sistema. El modelo de casos de uso es una especificación completa de todas las formas posibles de utilizar un sistema: requerimientos funcionales.

Los casos de uso describen al sistema, su ambiente, y las relaciones entre estos. Especifican el comportamiento del sistema y sus requerimientos. El comportamiento del sistema se refiere a cómo actúa y reacciona el sistema; la actividad visible y comprobable. En los casos de uso se especifica cómo se relacionan los objetos externos al sistema con él.

Un caso de uso es una unidad coherente de funcionalidad en donde se especifica una secuencia de acciones, incluyendo variantes, que el sistema puede llevar a cabo, y que producen un resultado observable de valor para un actor. Se dice que un caso de uso es una toma instantánea de algún aspecto del sistema pero en conjunto representan toda la funcionalidad del sistema.

Elementos del Caso de Uso

- Un Actor representa cualquier cosa que interactuó con el sistema
- Un Caso de Uso es una interacción típica entre un usuario (Actor) y un sistema



Los actores no son parte del sistema; representan los roles que pueden jugar los usuarios del sistema, éstos pueden intercambiar información activamente con el sistema, y pueden ser receptores pasivos de información o ser una persona, una máquina u otro sistema. Cada actor participa en uno o más casos de uso y un rol puede ser realizado por distintas instancias o actores.

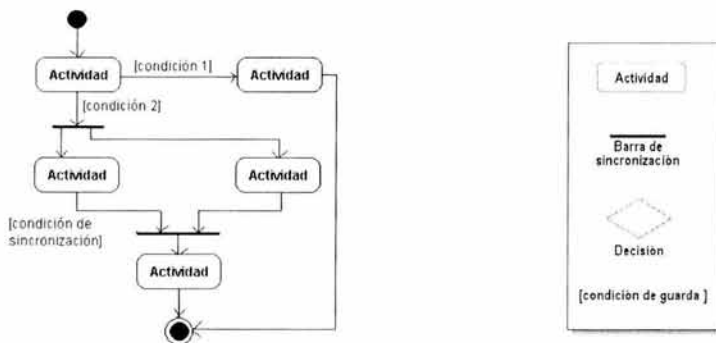
Las fuentes de información para obtener los casos de uso son principalmente la conversación con el cliente o usuario y las entrevistas con expertos, aunque también son importantes fuentes de información la literatura del problema y la experiencia del desarrollo de sistemas anteriores.

Los casos de uso son importantes para cada uno de los diferentes roles que participan en la creación de un sistema: para los clientes porque aprueban lo que el sistema debe hacer, para los usuarios son substanciales debido a que ganan entendimiento del sistema, mientras que a los analistas les proporciona las bases para el análisis y diseño. Por otra parte, a los desarrolladores les especifica el comportamiento del sistema y a los líderes del proyecto les brinda información para la planeación de proyectos. Finalmente a los documentadores les provee la base para la guía de usuario.

DIAGRAMAS DE UML

Diagramas de Actividad

Representan secuencia y acciones paralelas. Son útiles para la descripción del comportamiento que tiene una gran cantidad de procesos paralelos ya que permiten modelar los procesos reales de una organización humana y modelar actividades de software.



La barra de sincronización permite iniciar acciones una vez que se han realizado actividades concurrentes. La decisión es un punto en el que se pueden seguir alternativas distintas de acuerdo al resultado de la actividad anterior. Las condiciones de guarda son los posibles resultados de una acción que servirán como condición para la realización de otra

Durante el análisis se establecen clases (principalmente entity) y conexiones básicas (asociaciones, agregaciones y generalizaciones). Estas conexiones existen debido a la naturaleza de las clases, y no debido a una implementación específica. Se hace una estimación inicial de multiplicidad para exponer hipótesis ocultas

Durante el diseño se consideran otros estereotipos de clase (boundary y control). Se refinan y actualizan las estimaciones de multiplicidad. Se evalúan y refinan las relaciones. Se detallan los tipos y firmas (signature)

Diagramas de Interacción

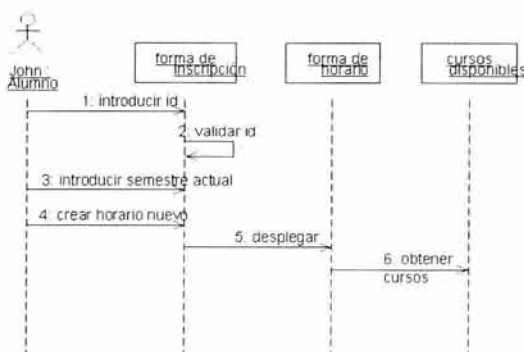
Un diagrama de interacción es una representación gráfica de interacciones entre objetos. Hay dos tipos de diagramas de interacción: diagramas de secuencia y diagramas de colaboración.

Cada uno provee un punto de vista diferente de la misma interacción. Los diagramas de secuencia están ordenados de acuerdo al tiempo. Los diagramas de colaboración pueden incluir flujo de datos.

Diagramas de Secuencia

Un diagrama de secuencia muestra interacciones de objetos ordenados en secuencia de tiempo. El diagrama muestra los objetos que participan en la interacción y la secuencia de mensajes intercambiados.

Un diagrama de secuencia contiene objetos con sus "líneas de vida", mensajes intercambiados entre objetos en orden secuencial y un enfoque de control (opcional).



Diagramas de Estado

Los diagramas de clase representan la estructura y el comportamiento estático de los objetos. Los diagramas de estado se utilizan para mostrar la historia de la vida de una clase dada, los eventos que causan una transición de un estado a otro, y las acciones que resultan de un cambio de estado. El estado de un objeto es una de las condiciones posibles en las que puede existir un objeto.

Los conceptos asociados a los diagramas de estado son:

- Un evento es la ocurrencia de alguna situación que sucede en un punto del tiempo.
- Una transición es un cambio que sucede de un estado original a un estado sucesor como resultado de algunos estímulos
- Una acción es una operación que se asocia a una transición
- Una actividad es una operación que toma tiempo para completarse. Las actividades se asocian con un estado.
- Una condición de guarda es una expresión booleana de valores de atributos que permiten una transición solo si la condición es verdadera.



Los estados pueden distinguirse por los valores de ciertos atributos:



Si el número máximo de alumnos por curso es 10

$\text{numEstudiantes} < 10$ $\text{numEstudiantes} >= 10$



Los estados pueden identificarse también por la existencia de ciertas ligas.

Diagramas de distribución

Los diagramas de distribución son creados para mostrar los diferentes nodos (procesadores y dispositivos) en el sistema.

Los nodos, al igual que los componentes, pertenecen al mundo material y modelan el aspecto físico de un sistema. Los nodos modelan la topología del hardware sobre el que se ejecuta el sistema. Los elementos esenciales de un diagrama de distribución son los nodos y sus conexiones. Un nodo es un objeto físico en tiempo de ejecución que representa recursos de cómputo. Una conexión indica comunicación, usualmente la comunicación directa con el hardware

II.2 IDENTIFICACIÓN DE LAS NECESIDADES

El objetivo central del presente apartado es dar a conocer las necesidades de la empresa para la cual el Sistema de Administración y Control de Inventarios va a ser utilizado. La problemática que presenta esta empresa servirá para hacer el análisis del sistema de tal forma que se traduzcan en soluciones concretas que ayuden a mejorar la organización de sus procesos internos, concretamente se pretende mejorar la forma en que es manejada la información relativa a sus inventarios y por lo tanto su proceso de compra y venta de mercancías.

GIRO DE LA EMPRESA

En la actualidad los pequeños y medianos negocios representan una parte importante del mercado mexicano. La mayoría de las empresas que se crean actualmente dentro del territorio nacional entran dentro de esta clasificación, esto se debe, principalmente, a las pocas oportunidades de trabajo que existen en el mercado laboral donde día a día se requiere de una mayor especialización y conocimientos para ingresar en él; esto provoca que cada vez más personas opten por crear su propio negocio, con la idea de mejorar sus ingresos. Sin embargo, este no es el único motivo por el cual nacen las pequeñas y medianas empresas (PYMES), ya que también encontramos profesionistas que son emprendedores gracias a que en nuestro país esta creciendo la cultura empresarial que brinda la oportunidad de satisfacer nichos de mercado poco explotados.

La empresa “Electromateriales Alfa”, es un negocio mediano, ubicado en Ciudad de México que se dedica a la comercialización de materiales eléctricos necesarios para las instalaciones domésticas e industriales, así como a la compra-venta de los artículos relacionados con la iluminación eléctrica.

Cuenta con varios proveedores que surten desde cable de uso industrial y sistemas completos de iluminación hasta focos para series navideñas. Cada uno de estos proveedores vende a “Electromateriales Alfa” ciertos productos brindando el precio más competitivo, sin embargo, son cinco los principales proveedores de esta empresa por ser ellos los que otorgan los créditos más favorables o los precios más asequibles.

El personal que labora en ella se conforma de cinco personas, el dueño que se encarga de hacer las compras a los proveedores, de asignar los precios de venta de sus productos y en general de su comercialización. Además laboran dos empleados de medio tiempo que son vendedores de las mercancías y se encargan del mantenimiento del local, uno trabaja en el horario matutino y el otro en el vespertino. Las otras dos personas se encargan de la distribución y entrega de mercancía en los pedidos que se hacen fuera de la localidad donde se encuentra físicamente el negocio.

Las mercancías que se comercializan son principalmente marcas reconocidas en el ramo y que cuentan con la garantía del fabricante, son marcas tanto nacionales como extranjeras que suelen variar en su precio, pero, que el cliente ya tiene identificadas como sus favoritas.

El horario de trabajo es de lunes a viernes de 9:00 a 19:00 hrs., los sábados de 9:00 a 18:00 hrs. y los domingos no abre.

En sus mostradores se exhiben principalmente artículos de iluminación como lámparas fluorescentes, lámparas decorativas y toda clase de timbres, contactos y apagadores.

Esta empresa fue creada hace tres años y sus márgenes de ganancia son relativamente bajos si se toma en cuenta la potencialidad que tiene, gracias a su buena ubicación comercial, la promoción que ha adquirido y los precios competitivos que ofrece.

NECESIDADES

La empresa "Electromateriales Alfa" dedica a la comercialización de artículos para las instalaciones del fluido eléctrico requiere mejorar su proceso de control interno del manejo de sus inventarios y llevar a cabo un mejor registro de sus operaciones diarias de compra y venta de mercancías.

Los registros actuales de las "ventas diarias", "artículos pendientes por comprar", "pagos y abonos pendientes", entre otros se realizan manualmente en lista de papel que no proporcionan una información completa para la toma de decisiones.

Requiere adquirir infraestructura para automatizar los procesos previamente citados tales como computadoras que sirvan de punto de venta, pero que al mismo tiempo compartan información centralizada de las operaciones diarias: para ellos es necesario una red de computadoras. Además se hace indispensable contar con el software que facilite la manipulación de esta información, es decir de un sistema informático capaz de procesar la información de la operación del negocio y de un repositorio común que mantenga la integridad, confidencialidad y seguridad de la información, esto es, un sistema manejador de bases de datos. Mas adelante en el capítulo V se detallarán todas las características técnicas de esta infraestructura.

De la misma forma se hace necesario redefinir los procesos para la organización de sus productos que están a la venta, es necesario que estén sincronizados con el funcionamiento del sistema de información, para esto, se requerirá de cierta clasificación física para facilitar una pronta captura en el sistema de cómputo.

Finalmente es importante redefinir las políticas internas dentro del negocio para que el personal que ahí labora en una primera instancia este enterado de las normas requeridas para manipular y alimentar el nuevo sistema y así, posteriormente, llevar a cabo las actividades regulares de comercialización a través del sistema informático. Tales políticas involucraran aspectos de seguridad como tipo de permiso por usuario, reglas de acceso al lugar donde se encuentran el servidor, entre otras.

PROBLEMÁTICA Y REQUERIMIENTOS

Existen varios problemas por resolver en esta empresa. El primero y más importante es el hecho de que constantemente se presenta la negación de un bien a los consumidores por no tenerlos en existencia. Esto provoca la pérdida de clientes por no cubrir sus necesidades y se da la impresión de ser un negocio que se preocupa poco por proporcionar un buen servicio. Este problema ocurre, principalmente, por la falta de un control de inventarios donde sea posible conocer con anticipación los productos próximos a escasear o simplemente tener el conocimiento de cual es el producto más vendido para mantener constantemente este artículo en existencia y no ser nunca negado.

Aún y cuando se llegara a conocer la carencia de un bien en el negocio, sucede con frecuencia que una vez hecha la solicitud de compra al proveedor que regularmente abastece el producto, éste no está disponible por parte del distribuidor y no se cuenta con la información necesaria sobre otros proveedores que ofrezcan el mismo artículo.

Otro problema es que, ocasionalmente, se omiten ciertos productos en la lista de "artículos pendientes por comprar", debido a que no se registra a tiempo cuando se observó de su carencia.

Existe mucha información perdida o que es difícil de conocer por la forma en la que se maneja la información de las facturas y en general de los registros de compra-venta. Estos registros se almacenan en una carpeta conforme se realizan las operaciones comerciales, y generalmente se ordenan por fecha. En ocasiones se desearía conocer rápidamente información valiosa, por ejemplo, que proveedor ofrece el precio más competitivo, cual es el proveedor al que más dinero se le debe, a que proveedor se le compra más, cual es el cliente que tiene la mayor cantidad de deudas con nosotros, quien es el cliente que más compra, cual es el artículo más vendido, cual es el artículo menos vendido, que empleado realiza el mayor número de ventas, en que fechas se da el mayor monto de ventas, en cual el menor, etc.

No se cuenta con la infraestructura necesaria para hacer funcionar un sistema de información capaz de automatizar estos procesos, ni con la capacitación para manipularlo, sin embargo se pretende implantarlo, para aumentar su volumen de ventas.

Este sistema debe ser confiable, seguro, rápido y sencillo de operar. Debe contar con una interfaz intuitiva capaz de ser dominada rápidamente. Se requiere de un sistema capaz de actualizar la información en varias terminales simultáneamente sin perder la integridad de la información y sin crear redundancia innecesaria.

El sistema debe ser capaz de brindar los reportes con la información requerida en todo momento y manejar niveles de seguridad, uno para el administrador, para los vendedores y para los capturistas para que cada uno de ellos, en sus diferentes roles solo tenga acceso a la información requerida para hacer su trabajo. Deberá tener una pantalla donde se pueda configurar el formato de las facturas. La captura de la información en él debe ser ágil, de tal forma que los registros en él se realicen en el mínimo de tiempo. Debe brindar ayuda al usuario.

II.3 OBJETIVOS Y PLAN DE ACTIVIDADES

El propósito principal para desarrollar un sistema de administración de inventarios es mantener al día la información de los inventarios para poder determinar la cantidad de mercancías que se necesitan adquirir para su comercialización en el negocio y para no negar bienes que por no estar en existencia se dejan de vender.

Se pretende hacer un sistema que sea capaz de automatizar los procesos para mantener actualizado las cantidades de inventario que existen en el negocio y que sirva para realizar una mejor toma de decisiones sobre las mercancías que se necesitan adquirir. Se desea que lo puedan emplear varias personas al mismo tiempo sobre una red local de computadoras manejando una base de datos centralizada. Debe ser un sistema que no necesite conocimientos avanzados de computación para poderlo manejar. Es importante que implemente métodos para que la captura de los datos se realice de una manera ágil.

Se emplearán sistemas operativos y programas que cuenten con soporte técnico por parte de sus fabricantes y que no requieran excesiva infraestructura.

El sistema debe mostrar reportes que faciliten la toma de decisiones e implementar módulos de seguridad con varios niveles para las distintas actividades dentro de la organización.

OBJETIVO GENERAL DEL PROYECTO

Integrar la información correspondiente al registro de compras y ventas de "Electromateriales Alfa" permitiendo la diferenciación de tipos de usuarios dentro del sistema para automatizar el control de los inventarios de esta empresa.

OBJETIVOS ESPECÍFICOS

- Registrar, actualizar y consultar información de Clientes, Proveedores y Empleados, así como toda la información referente a los artículos ofrecidos por la empresa.
- Registrar la información detallada de las actividades de compra-venta de mercancía.
- Agilizar la búsqueda y disponibilidad de artículos para su venta.
- Automatizar el control de los inventarios.
- Controlar los abonos hechos a los Proveedores y pagos de los Clientes.
- Permitir la consulta de artículos más y menos vendidos.
- Permitir la consulta de los registros de compra-venta asociados a los Proveedores-Clientes y Empleados.

FACTORES DE ÉXITO

- Comunicación y conocimiento de los objetivos, características y alcances del proyecto por parte de todos los involucrados.
- Contar con personal suficiente y capacitado en las herramientas de análisis, diseño y desarrollo a utilizar
- Contar con la infraestructura de cómputo y telecomunicaciones adecuada, así como con las herramientas y el software de desarrollo necesario
- Eficiente coordinación y seguimiento de las actividades.
- Retroalimentación oportuna de los usuarios
- Identificación oportuna de riesgos y aplicación de acciones preventivas y correctivas

PLAN DE ACTIVIDADES

ETAPA	DURACIÓN (DIAS)	FECHA DE INICIO	FECHA DE TÉRMINO
DISEÑO			
Arquitectura lógica de datos (BD)			
Diagrama Entidad-Relación	15	Oct-01	Oct-15
Diccionario de Datos	15	Oct-16	Oct-31
Componentes y Objetos de la Base de Datos	15	Nov-01	Nov-15
Arquitectura de distribución			
Diagrama de Distribución	14	Nov-16	Nov-30
Especificaciones de Hardware, Software y Conectividad	15	Dic-01	Dic-15
Arquitectura de componentes			
Diagrama modular	15	Dic-16	Dic-31
Diagrama de componentes	15	Ene-01	Ene-15
Especificaciones de componentes y estándares	15	Ene-16	Ene-31
Estructura de Interfaces			
Estándares	15	Feb-01	Feb-15
Diseño Gráfico	15	Feb-16	Feb-28
Revisión del diseño detallado	15	Mar-01	Mar-15
DESARROLLO			
Instalación del software para el desarrollo	15	Mar-16	Mar-31
Creación de la base de datos	15	Abr-01	Abr-15
Modulo de Artículos	14	Abr-16	Abr-30
Modulo de Proveedores	15	May-01	May-15
Modulo de Clientes	15	May-16	May-31
Módulo de Empleados	15	Jun-01	Jun-15
Módulo de Categorías	14	Jun-16	Jun-30
Módulo de Seguridad	15	Jul-01	Jul-15
Modulo de Artículos Alterados	15	Jul-16	Jul-31
Módulo Compras	15	Ago-01	Ago-15
Módulo Ventas	15	Ago-16	Ago-31
Módulo Búsquedas	15	Sep-01	Sep-15
Reportes	14	Sep-16	Sep-30
IMPLEMENTACIÓN			
Instalación	15	Oct-01	Oct-15
Pruebas Internas	15	Oct-16	Oct-31
Pruebas con el cliente	15	Nov-01	Nov-15
Documentación	14	Nov-16	Nov-30
Capacitación	15	Dic-01	Dic-30

II.4 VIABILIDAD

FACTIBILIDAD TÉCNICA

Software

La tecnología de software en la actualidad se compone de dos grandes bloques: la de software libre y la de software propietario.

El software libre representa un nuevo paradigma en los sistemas de información puesto que el desarrollo se hace por todas las personas en el mundo interesadas en incrementar sus capacidades, y lo hacen de una forma gratuita. Existe actualmente toda clase de software libre desde sistemas operativos para grades computadoras como Linux, hasta la más sencilla utilería. El software libre por el lado del precio tiene muchísimas ventajas puesto que no cuesta, como su nombre lo indica y, en general, para el aprendizaje universitario es muy importante, sin embargo, al mismo tiempo, tiene algunas desventajas, por ejemplo la relacionada al escaso soporte técnico existente, aunque algunas empresas están haciendo un gran esfuerzo en este sentido para proporcionar el servicio.

Por otro lado esta el software conocido como propietario. Este es el software que se adquiere de las grandes compañías como Microsoft, Oracle, IBM, etc. A diferencia del software libre, este tipo de software tiene un costo por la licencia de uso, sin embargo proporciona la asistencia técnica necesaria y existe documentación técnica de todo tipo sobre ellos en el mercado.

Es sistema SAI utiliza el software propietario debido a la extensa documentación que existe en el mercado para hacer uso de ella en cualquier eventualidad o, en su defecto, hacer uso de la asistencia técnica por parte del proveedor.

Por el lado del sistema operativo los sistemas desarrollados por Microsoft son los más populares en la actualidad y es más fácil para el usuario adaptarse a un sistema que se ejecuta dentro de un ambiente conocido que si lo trata de hacer en un nuevo sistema operativo.

En cuanto al sistema manejador de la base de datos, se seleccionó SQL Server porque es el manejador que saca mayor provecho de las capacidades para base de datos de Visual Basic, que, por otro lado, es uno de los lenguajes que aprovecha mejor las características del sistema operativo Windows dado que ambos son del mismo fabricante.

Hardware

El hardware no es tan relevante en el sentido de que cualquier sistema que ejecute el sistema operativo Windows y que tenga capacidades para conectarse a una red es útil para ser cliente dentro del SAI.

En cuanto al hardware existen máquinas que son vendidas ensambladas por algunas compañías reconocidas como Dell, IBM, HP, etc. Y otras que se venden por partes, conocidas como genéricas. Las primeras son más caras porque los fabricantes ofrecen

asistencia técnica en todo momento. Las segundas, son más económicas, sin embargo, al igual que las máquinas “de marca”, cuentan con garantía en todos sus componentes.

La consideración especial en hardware es para el equipo que cumpla la función de servidor, porque es el sistema que va a llevar la mayor carga de trabajo. Se recomienda que este equipo tenga componentes de buena calidad y de preferencia que maneje procesamiento en paralelo una amplia capacidad de almacenamiento.

Existe en el mercado una amplia variedad de componentes para configurar el servidor, de muy diversos precios, sin embargo es importante poner la confiabilidad del equipo sobre su costo.

Arquitectura

La arquitectura es la estructura organizativa de un sistema, que incluye su descomposición en partes, conectividad, mecanismos de interacción y Principios de guía que proporcionan información sobre el diseño del mismo.

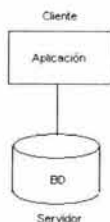
La arquitectura es el espacio donde trabajan los objetos. Con la arquitectura definimos cómo se reparte el mi sistema y que herramientas se ajustan mejor.

Arquitectura Cliente/Servidor

El SAI utiliza una arquitectura Cliente-Servidor.

Desventajas de la arquitectura Cliente/Servidor:

- Puede haber mucha carga en el cliente
- Mantenimiento costoso en cada cliente
- Posibilidad de clientes desfasados



Ventajas de la arquitectura Cliente/Servidor:

- Es menos una arquitectura sencilla
- Se tiene un mayor control sobre los componentes
- Distribución de la carga de trabajo

FACTIBILIDAD ECONÓMICA

El estudio de factibilidad económica nos permite conocer el costo del proyecto y con él podemos determinar si lo podemos finalizar o no.

A continuación se presentan las tablas correspondientes a los costos de recursos humanos en el proyecto del Sistema para la Administración de Inventarios (SQI) de acuerdo a los sueldos vigentes en el mercado.

Cursos	Costo X Curso (20 hrs)
Básico	\$8,000.00
Especializado	\$10,000.00
Muy especializado	\$12,000.00

Perfiles	Costo Hora	Promedio Sueldo X Hr	Sueldo
Soporte Técnico	\$70.00	\$120.00	\$7,144.00
Consultor en Telecomunicaciones	\$300.00	\$289.00	\$16,321.00
Consultor en Data Center	\$300.00	\$289.00	\$16,321.00
Analista	\$200.00	\$189.00	\$8,646.00
Diseñador Gráfico	\$200.00	\$160.00	\$7,860.00
Probador	\$100.00	\$156.00	\$7,144.00
Programador	\$200.00	\$171.00	\$7,860.00
Administrador de Base de Datos	\$250.00	\$189.00	\$8,646.00
Capacitador	\$100.00	\$120.00	\$7,144.00

	MESES / HORAS														Total Hr.	Costo Total	
	Personas	Oct	Nov	Dic	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct			Nov
ASESORÍA INFRAESTRUCTURA																	
Consultor en telecomunicaciones	1	20	20	20	20											80	\$24,000.00
Consultor en Data Center	1	20	20	20	20											80	\$24,000.00
Total Asesoría	2	40	40	40	40											160	\$48,000.00
CAPACITACIÓN																	
SQL	1				20											20	\$8,000.00
SQL Avanzado	1					20										20	\$12,000.00
Administración de Windows 2000	1						20									20	\$12,000.00
Administración de SQL Server 2000	1							20								20	\$12,000.00
Seguridad en Redes	1								20							20	\$12,000.00
Administración en Redes	1									20						20	\$6,000.00
Total Capacitación	6				20	20	20	20	20	20						100	\$56,000.00
DISEÑO DE LA APLICACIÓN																	
Análisis			20	20	20	20										80	\$15,120.00
Diseño						20	20									40	\$6,400.00
Desarrollo								20	20	20	20	20				100	\$17,100.00
Pruebas													20			20	\$3,120.00
Implantación														20	20	40	\$7,560.00
TOTAL																	\$257,300.00

En la tabla anterior se observa que es factible llevar a cabo el proyecto dentro de un año distribuyendo la carga de trabajo, desde la asesoría hasta la implantación y se hace el cálculo con los costos promedio en el mercado

No se considera el hardware ni el equipo de red dado que será adquirido por el cliente de forma independiente

FACTIBILIDAD OPERACIONAL

Consiste en conocer si hay recursos humanos calificados y si el sistema instalado va a trabajar en perfectas condiciones.

En la actualidad no se cuenta con personal capacitado en la empresa que será nuestro cliente para que maneje y administre el SAI. Esto se soluciona con cursos de capacitación antes de la implementación del sistema. Estos cursos están programados para impartirse por personal calificado especializado en el área.

Los cursos que necesarios son los siguientes:

- Administración de Windows 2000
- SQL básico
- SQL avanzado.
- Administración de SQL Server.
- Introducción a SAI

Por otra parte, se dará asistencia técnica en cualquier eventualidad surgida del sistema, principalmente lo concerniente a la base de datos una vez que sea implementado.

Por el lado del desarrollo se cuenta con el equipo, el software y el personal necesarios para desarrollar el sistema en el periodo de un año.

Para el desarrollo se hará uso del siguiente software:

- Lenguaje de Programación: Visual Basic 6.0
- Manejador de la Base de Datos: SQL Server 2000
- Sistema Operativo del Servidor de la Base de Datos: Windows 2000 Server
- Herramientas Case: Rational Rose.

Las instalaciones donde operará el SAI ya se tienen identificadas y se puede decir que cuentan con las condiciones indispensables para que pueda operar una red local de computadoras. Así mismo los equipos están en óptimas condiciones y cuentan con los componentes necesarios que requiere el sistema.

III

DISEÑO DE LA BASE DE DATOS

III.1 ANTECEDENTES DE BASES DE DATOS

DEFINICIÓN DE BASE DE DATOS

Una base de datos es similar a un archivo de datos en que es también un lugar de almacenamiento de datos. Como en un archivo de datos, una base de datos no presenta información directamente al usuario; el usuario ejecuta una aplicación que tiene acceso a los datos de la base de datos y los presenta al usuario en un formato inteligible.

Las bases de datos son una colección de datos integrados, con redundancia controlada y con una estructura que refleje las interacciones y restricciones existentes en el mundo real: los datos que han de ser compartidos por diferentes usuarios y aplicaciones, deben mantenerse independientes de estas, y su definición y descripción, únicas para cada tipo de datos, han de ser almacenadas junto con los mismos. Los procedimientos de actualización y recuperación comunes, y bien determinadas, habrán de ser capaces de conservar la integridad, confidencialidad y seguridad del conjunto de datos.

Los sistemas de bases de datos son más eficaces que los archivos de datos, ya que los datos están mejor organizados. En una base de datos bien diseñada, no hay elementos de datos duplicados que el usuario o la aplicación deban actualizar al mismo tiempo. Los elementos de datos están agrupados en una única estructura o registro, y se pueden definir relaciones entre dichas estructuras y registros.

Cuando se trabaja con archivos de datos, las aplicaciones se tienen que codificar para trabajar con la estructura específica de cada archivo de datos. En contraste, las bases de datos contienen un catálogo que las aplicaciones pueden utilizar para determinar la organización de los datos. Las aplicaciones de bases de datos genéricas pueden utilizar el catálogo para presentar dinámicamente a los usuarios datos de distintas bases de datos, sin tener que depender de formatos de datos específicos.

Generalmente, una base de datos tiene dos partes principales: los archivos que almacenan la base de datos física y el software del sistema de administración de la base de datos (DBMS, Database Management System), que las aplicaciones utilizan para tener acceso a los datos. El DBMS es el responsable de mantener la estructura de la base de datos, lo que incluye:

- El mantenimiento de las relaciones entre los datos de la base de datos.
- La garantía de que los datos estén correctamente almacenados y de que no se infrinjan las reglas que definen las relaciones entre los datos.
- La recuperación de todos los datos hasta un punto coherente en caso de errores del sistema.

BASE DE DATOS RELACIONAL

Hay diferentes maneras de organizar los datos en una base de datos, las bases de datos relacionales son una de las más efectivas. Los sistemas de bases de datos relacionales son una aplicación de la teoría matemática de conjuntos al problema de la organización de datos. En una base de datos relacional, los datos están organizados en tablas (llamadas relaciones en la teoría relacional).

Una tabla representa una clase de objeto que tiene cierta importancia en una organización. Por ejemplo, una organización puede tener una base de datos con una tabla para los empleados, otra tabla para los clientes y otra para los almacenes. Cada tabla se crea con columnas y filas (llamados atributos y tuplas en la teoría relacional). Cada columna representa algún atributo del objeto representado por la tabla. Por ejemplo, una tabla Empleado tendría columnas para los atributos como nombre, apellido, id de empleado, departamento, pago, título. Cada fila representa una instancia del objeto representado por la tabla. Por ejemplo, una fila de la tabla Empleado representa el empleado cuyo Id. de empleado es 12345.

Al organizar los datos en tablas, se pueden encontrar varias formas de definirlos. La teoría de las bases de datos relacionales define un proceso llamado normalización (como se vio en el punto 1 del presente capítulo), que garantiza que el conjunto de tablas que define organizarán los datos de manera eficaz.)

Una base de datos debe cumplir con las condiciones siguientes:

1. Los datos han de estar almacenados juntos
2. Tanto los usuarios finales como las aplicaciones necesitan conocer los detalles de las estructuras de almacenamiento.
3. Los datos son compartidos por diferentes usuarios y programas de aplicación; existe un mecanismo común para inserción, actualización, borrado y consulta de los datos.
4. Los procedimientos de actualización y recuperación, comunes y bien determinados habrán de ser capaces de conservar la integridad, confidencialidad y seguridad del conjunto de datos.
5. Tanto datos como procedimientos pueden ser transportados conceptualmente a través de diferentes DBMS

VENTAJAS DEL ENFOQUE DE BASE DE DATOS

Los siguientes conceptos nos muestran las ventajas de utilizar el esquema de bases de datos relacional.

Redundancia

La redundancia de datos se refiere, a la existencia de información repetida o duplicada en diferentes tablas dentro de una base de datos.

La redundancia conduce a muchos problemas que tienen que ver con la integridad y consistencia de los datos. La redundancia de los datos requiere múltiples procedimientos de entrada y de actualización de los mismos.

Dentro de una base de datos relacional la redundancia debe ser mínima y controlada. En ocasiones existirán motivos válidos de negocios o técnicos para mantener varias copias de los mismo datos almacenados.

Consistencia

Frecuentemente los problemas de consistencias de datos se deben a la redundancia de estos.

Es muy probable que surjan incongruencias al almacenar la misma información en más de un lugar; ya que al modificar, eliminar o agregar un dato, en esas condiciones, debe realizarse en cada una de las instancias del mismo con el riesgo de no ser realizado en su totalidad, generando en este caso datos inconsistentes.

Integridad

La integridad de una base de datos se refiere no sólo al hecho de que los datos sean consistentes dentro de la base, sino además, que los valores que posean los datos sean válidos de acuerdo a las dependencias funcionales entre las tablas y de acuerdo a las políticas del negocio.

La inconsistencia de dos entradas que representan el mismo “hecho” es una falta de integridad que, por supuesto solo ocurre si existe redundancia de los datos almacenados.

- La integridad de la base de datos se puede lograr mediante:
- Manteniendo una redundancia mínima y controlada.
- El establecimiento de llaves primarias o índices primarios.
- Estableciendo reglas de validación durante la creación y edición de los datos
- Estableciendo procedimientos que validen la dependencia funcional entre tablas relacionadas (Integridad referencial)

SEGURIDAD DE LA BASE DE DATOS

Hoy en día se considera la información de una empresa como uno de los activos más valiosos, por lo que la seguridad de la misma es muy importante.

La seguridad de una base de datos se refiere principalmente al control de acceso, modificación y definición, tanto de los datos como la estructura de la base de datos por parte de los diferentes usuarios de la base de datos.

La seguridad implica asegurar que los usuarios estén autorizados para llevar a cabo lo que tratan de hacer.

Algunos sistema operativos brindan algún nivel de seguridad en el control de acceso a los usuarios, sin embargo esta debe radicar principalmente en el sistema manejador de bases de datos (DBMS) o en la aplicación que maneje la base de datos.

Seguridad de los Datos

Un sistema de cómputo es seguro si se puede confiar en que él, y su software se comportaran como se espera que lo hagan, y que la información almacenada en él se mantendrá inalterada y accesible durante tanto tiempo como su dueño lo desee.

El problema de la seguridad tiene muchos aspectos entre ellos los siguientes:

- Aspectos, legales, sociales y éticos (por ejemplo ¿tiene una persona que solicitar el crédito de un cliente, digamos derecho legal a obtener la información solicitada?).
- Controles físicos (por ejemplo, ¿deberá cerrar o resguardar de alguna otra manera el cuarto de la computadora?).
- Cuestiones de política interna (por ejemplo, ¿cómo decide la empresa propietaria del sistema quienes pueden tener acceso a que?).
- Problemas de operación (por ejemplo, si se utiliza un sistema de contraseñas, ¿cómo se mantiene en secreto la contraseña?, ¿con qué frecuencia se cambian?).
- Controles de equipo (por ejemplo, ¿posee el CPU características de seguridad tales como claves para la protección de las áreas de almacenamiento o un modo de operación privilegiado?).
- Seguridad del sistema operativo (por ejemplo, ¿borra el sistema operativo subyacente el contenido de las áreas de almacenamiento los archivos de datos cuando ya no se necesitan?).

Existe otro tipo de seguridad dentro de las bases de datos conocido como seguridad de objetos. Este tipo de seguridad se refiere a los permisos de los diferentes usuarios para poder hacer uso de tablas, procedimientos almacenados, triggers, etc.

Dentro de la seguridad de las bases de datos existen ciertos conceptos que definen las características para que una base de datos sea segura, estos son:

Confidencialidad. Un sistema manejador de base de datos no debe permitir que la información contenida en él no este accesible a nadie que no tenga autorización adecuada

Autenticidad. El sistema debe de ser capaz de impedir los cambios a los datos ya sea por medio de hardware, software o personal no autorizado, además de contar con los medios para verificar que los datos contenidos no son modificados ilegalmente

Disponibilidad. Significa que tanto el hardware, software y sobre todo la información se mantendrán disponibles y funcionando en forma eficiente para su uso por parte de los usuarios y que el sistema sea capaz de recuperarse rápidamente cuando surjan problemas.

Características Técnicas de Seguridad

Los Sistemas manejadores de bases de datos relacionales (RDBMS) se direccionan por medio de direcciones IP y de puertos en el servidor, por lo que hay veces que sin necesidad de tener una cuenta en el servidor a nivel de sistema operativo se puede tener acceso al mismo, únicamente configurando el cliente o el ODBC con estos datos y una cuenta con privilegios es más que suficiente. A veces los fabricantes de RDBMS por

comodidad del usuario usan típicamente el mismo puerto y las mismas cuentas para configurar el servidor, las cuales cuando se instalan tienen la misma contraseña o no la tienen.

En el caso de SQL Server se instala en el servidor la cuenta "sa" que de forma predeterminada no tiene contraseña y utiliza el puerto 1433.

REGLAS DE NORMALIZACIÓN

Las reglas de normalización sirven para mantener la integridad referencial de la base de datos, se describen a continuación cada una de ellas:

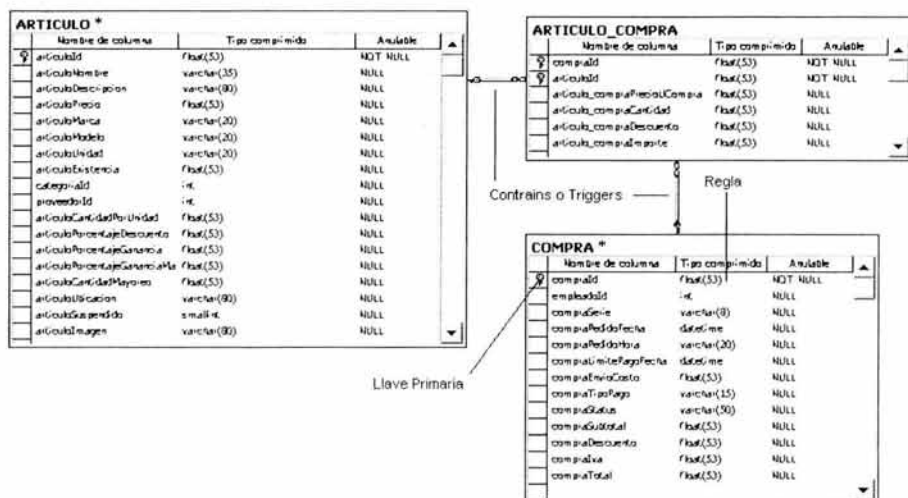
1ª Forma normal. Debemos tener un atributo único que identifique a cada renglón de la tabla. Se asigna una llave primaria a este atributo. Si se tiene una tabla T1 con atributos a1, a2, a3..., se dice que esta en primera forma normal si al como clave identifica de manera única cada renglón de T1.

2ª Forma normal. No deben de existir dependencias parciales en una tabla. Es decir, si una tabla tiene una llave primaria formada por dos o más atributos, los atributos no llave deberán depender de todos los atributos que conforman la llave primaria.

3ª Forma normal. No deben existir dependencias transitivas, es decir, un atributo no llave no debe depender de otro atributo no llave.

Para implementar integridad de la base de datos, se hace uso de algunos de sus componentes tales como llaves primarias y foráneas que surgen en el diseño e indican las dependencias entre las diferentes tablas; los constrains y triggers que son herramientas disponibles en la mayoría de los DBMS y que ayudan a implementar las dependencias antes mencionadas; los índices que garantizan la primera forma normal y, finalmente, las reglas que garantizan que el tipo de datos insertado en un campo estén en la forma y características requeridas.

En el siguiente diagrama podemos apreciar estos componentes:



REGLAS DE CODD

Para crear una base de datos relacional es necesario seguir con las reglas de Codd, estas son las reglas:

1. Cualquier DBMS que proclama ser relacional, deberá manejar completamente, las bases de datos por medio de sus capacidades relacionales.
2. Regla de Información. Toda la información dentro de una base de datos relacional se representa de manera explícita a nivel lógico y exactamente de una sola manera, como valores en una tabla.
3. Regla de acceso garantizado. Se garantiza que todos y cada uno de los datos (valor atómico) en una base de datos relacional pueden ser leídos recurriendo a una combinación del nombre de la tabla, valor de la llave primaria y nombre de la columna.
4. El manejo sistemático de los valores nulos. En un DBMS totalmente relacional se soportan los valores nulos (que es distinta de una cadena de caracteres vacía o de una cadena de caracteres en blanco o de cero o de cualquier otro número), para representar información faltante o no aplicable a una forma consistente, independientemente del tipo de dato.
5. Catálogo dinámico en línea basado en un modelo relacional. La descripción de la base de datos se representa en un nivel lógico de la misma forma que los datos ordinarios, de tal suerte que los usuarios autorizados pueden aplicar el mismo lenguaje relacional para consultarla, que aquel que emplean para con sus datos habituales.
6. Regla del sublenguaje de dato completo. Contempla definición de datos, definición de vistas, manipulación de datos, restricciones de integridad, autorización inicio y fin de una transacción.
7. Regla de actualización de vistas. Todas las vistas que teóricamente sean actualizables deberán ser actualizadas por medio del sistema.
8. Inserción, actualización y eliminación de alto nivel. La posibilidad de manejar una relación base o una relación derivada como un solo operador se aplica a la lectura, inserción, modificación y eliminación de los datos.
9. Independencia física de los datos. Los programas de aplicación y la actividad en las terminales no deberán ser afectadas por cambios en el almacenamiento físico de los datos o en el método de acceso.
10. Independencia lógica de los datos. Los programas de aplicación y la actividad en las terminales no deberán ser afectadas por a cualquier tipo que preserve la información y que teóricamente permitan la afectación en la tabla base.
11. Independencia de la integridad. Las restricciones de integridad de una base de datos deberán poder definirse en el mismo sublenguaje de datos relacional y deberán almacenarse en el catálogo, no en los programas de aplicación.
12. Independencia de la distribución. Un DBMS relacional tiene independencia de la distribución.
13. Regla de la no subversión. Si un sistema relacional tiene u lenguaje de bajo nivel (un solo registro cada vez), ese nivel no puede ser utilizado para suprimir las reglas de integridad y las restricciones expresadas en el lenguaje relacional de nivel superior (múltiples registros a la vez).

MODELO RELACIONAL

Una base de datos relacional es aquella cuyos usuarios la perciben como un conjunto de tablas.

Un DBMS utiliza un modelo de datos para definir la estructura fundamental de los mismos. Un modelo de datos expresa entidades y sus relaciones y es la herramienta utilizada para representar la organización conceptual de los datos, siendo necesario para ello el uso de cierta terminología que se verá a continuación.

Es importante conocer la definición de entidad pues es uno de los conceptos principales del modelo relacional.

Una entidad es una persona, lugar, evento u objeto identificado en forma única y del cual se registra información y que además cae dentro del alcance del sistema por lo que éste debe mantener, correlacionar y desplegar información. En una forma más coloquial podemos decir que una entidad es una representación abstracta de la realidad de la cual queremos obtener información.

Estas pueden ser de dos tipos: Tangible, por ejemplo Empleado, Alumnos, Piezas, artículos o Intangibles como: Sucesos, Actividades, Cuentas de Cliente. Las entidades se representan por medio de tablas en la base de datos.

Las entidades se representan en tablas de la base de datos, las tablas son estructuras bidimensionales formadas por tuplas o campos y registros o atributos. El grado de una tabla es el número de columnas o atributos. Las tablas también tienen cardinalidad, la cardinalidad se refiere al número de registros o tuplas. Cada registro tiene cierto dominio, esto es, el conjunto de datos o valores que posee un campo comúnmente conocido como tipo de dato.

Las tablas pueden tener llaves primarias y llaves foráneas. La llave primaria es aquel atributo o atributos que permiten identificar de manera única a cada registro dentro de una tabla. Por otro lado la llave foránea Es aquel atributo o conjunto de atributos que en la tabla padre forman parte de la llave primaria y en la tabla hija son un atributo más.

III.2 LENGUAJE SQL

En este tema no se pretende dar una descripción detallada del lenguaje SQL sino, más bien, dar una idea general de los aspectos relevantes y sobre todo el concepto principal de cada comando además de presentar un panorama para utilizarlos en el contexto adecuado.

Las bases de datos de un sistema manejador de base de datos están estrechamente relacionadas con este lenguaje, tanto para su creación como para su manipulación. El SQL es la interfaz entre los clientes y el manejador de la base de datos.

El SQL Es un lenguaje semánticamente fácil de entender, ya que las sentencias (instrucciones) parecen sencillas frases en inglés, lo que lo hace fácil de aprender y utilizar. Estas sentencias hacen énfasis en el "qué" consultar o definir y no en el "cómo" el servidor realiza la consulta.

Las sentencias de SQL son como frases escritas en inglés, donde decimos lo que deseamos obtener y donde obtenerlo. Todas las sentencias siguen una sintaxis para que se puedan ejecutar correctamente.

El SQL (Structured Query Language, en español Lenguaje de Consulta Estructurado), es un lenguaje para el acceso a bases de datos relacionales. SQL se ha convertido en la actualidad, en un lenguaje estándar de bases de datos, soportándolo la mayoría de los sistemas de bases de datos. Nació en los laboratorios de investigación de IBM, de un lenguaje, llamado SEQUEL, para la especificación de las características de las bases de datos que adoptaban el modelo relacional. IBM lo adoptó y lo utilizó internacionalmente. Gracias al éxito numerosas compañías, empezaron a comercializar productos basados en SQL, convirtiéndose en el estándar en lo que respecta a las bases de datos relacionales. Tras esto, las organizaciones ANSI e ISO lo adoptaron como estándar para los lenguajes relacionales.

El hecho de tener un estándar definido por un lenguaje para bases de datos relacionales abre potencialmente el camino a la intercomunicación entre todos los productos que se basan en él. Pero a partir del estándar, cada sistema ha desarrollado su propio SQL, por lo que SQL puede variar de un sistema a otro, siendo estos cambios insignificantes por lo que no suponen ningún problema para alguien que conozca un SQL concreto.

SQL nos permite realizar consultas a la base de datos. Pero además también nos permite realizar funciones de definición, control y gestión de la base de datos.

Las sentencias SQL se clasifican por su finalidad dando origen sublenguajes:

- DDL (Data Description Language, en español Lenguaje de Definición de Datos), define, modifica o borra las tablas en las que se almacenan los datos y de las relaciones entre estas.
- DCL (Data Control Language, en español Lenguaje de Control de Datos), sirve para trabajar en un entorno multiusuario, donde es muy importante la protección y la seguridad de los datos y la compartición de datos por parte de usuarios.

- DML (Data Manipulation Language, en español Lenguaje de Manipulación de Datos), permite al usuario recuperar datos almacenados en la base de datos así como actualizarla añadiendo, suprimiendo o modificando datos almacenados.

LENGUAJE DE DEFINICIÓN DE DATOS (DDL - DATA DEFINITION LANGUAGE)

El lenguaje de Definición de Datos es el aspecto del DBMS (Data Base Management System - Sistema de Manipulación de una BD) que especifica la forma en que los datos pueden almacenarse. Este lenguaje tiene dos perspectivas que son las que se mencionan a continuación.

1. Visión Lógica: Es la forma en que el programador percibe la realidad de la BD.
2. Visión Física: Es la forma en que físicamente está almacenada la información en una BD.

El lenguaje de definición de datos tiene como principales funciones:

- Crear, suprimir o modificar la definición de una tabla.
- Definir y suprimir una vista de datos (tabla virtual).
- Definir y suprimir índices de tablas.

LENGUAJE DE MANIPULACIÓN DE DATOS (DML - DATA MANIPULATION LANGUAGE)

Es o consiste en la forma de recuperar datos almacenados en una BD.

Se distinguen dos tipos de operaciones:

- Operaciones de Actualización (actúan sobre una única tabla)

INSERT: inserción de filas
DELETE: eliminación de filas
UPDATE: modificación de filas

- Operaciones de Consulta (actúan sobre varias tablas)

SELECT: consulta sobre la Base de Datos

Para ambas se requiere el lenguaje de manipulación de datos. Entonces, el DBMS se convierte en un intermediario entre el usuario y la BD almacenada, entonces, a través de un programa o una consulta podemos extraer información.

LENGUAJE DE CONTROL DE DATOS (DCL - DATA CONTROL LANGUAGE)

Es el encargado de cuidar la seguridad de los datos. En él se da la concesión de privilegios a los diferentes usuarios de la base de datos. La concesión de privilegios se utiliza para permitir a los usuarios el acceso completo o restringido a las tablas de la base de datos.

SELECCIÓN DE DATOS

Para seleccionar o consultar los registros y campos de una tabla se utiliza el comando "Select".

```
select * | [distinct] <campo>, <campo>
from <nombre tabla>, [ <nombre tabla> ]
[where <condición> ]
[group by <campo>, <campo>,... ]
[having <condición> ]
[order by <campo>, <campo>,.... ]
```

La instrucción "select" tiene dos componentes que son indispensables y deben estar en cualquier instrucción de este tipo; éstas son: las palabras "select" y "from". Los demás componentes son opcionales.

Cláusula Where

La cláusula "Where" sirve para indicar el criterio o condición que se desea aplicar a la consulta. Para esto se utilizan distintos operadores que permiten la comparación de valores. Comúnmente la condición de búsqueda tiene la siguiente forma:

<expresión> <operador> <expresión>

Donde <expresión> puede ser un campo o un valor dado y <operador> puede ser algún operador lógico o comparativo. Los operadores de comparación son: (=, <, >, != ó <>, >=, <=, between, in, like e isnull()). Los operadores lógicos son: (not, !, or u ||, and o &&);

Por ejemplo: sueldoEmpleado < 1000 and tipoEmpleado = "H", donde sueldoEmpleado y tipoEmpleado representan campos de la tabla y 1000 y "H" son la expresión con la que se va a comparar el valor del los campos respectivos.

Comodines

Los comodines se utilizan para sustituir uno o más caracteres dentro de una cadena de caracteres, se cuentan con los siguientes:

- % Substituye cualquier número de caracteres
- _ Substituye un caracter

Ejemplo:

Para seleccionar todos los campos de la tabla empleados, donde el campo sexo_emp = "M" y la primera letra del campo nombre_emp sea igual a "S"

```
select *
from empleados
where sexo_emp = "M"
and nombre_emp like "S%"
```

Valores Nulos

Una expresión contiene un valor NULL cuando esta expresión está vacía, es decir, no contiene ningún valor. No se consideran nulos en el caso de que un campo tipo numérico contenga un 0, o un campo tipo carácter contenga una cadena de espacios.

Para evaluar que un campo se encuentre nulo, se hace uso de los operadores IS NULL e IS NOT NULL.

“Alias” en columnas

Al seleccionar los campos de una tabla es posible identificarlos mediante un “alias”, es decir, con un nombre diferente al nombre físico del campo.

Por ejemplo:

```
select cveAut as “Clave autor” from autor
```

En este ejemplo “Clave autor” es el sustituto para nombrar a cveAut

“Alias” en campos

También se puede identificar una tabla mediante un alias, con la diferencia que sólo tiene efectos prácticos dentro de la sentencia SQL.

Por ejemplo:

```
select l.lib_cve, l.lib_nom, a.aut_nom, a.aut_apat, a.aut_amat  
from autor as a, libros as l  
where a.aut_cve = l.aut_cve
```

Registros únicos

Por definición cada registro de una tabla debe ser único, sin embargo una sentencia SQL puede generar como resultado registros duplicados.

Para indicar que el resultado deseado de una sentencia SQL presente sólo registros únicos, se utiliza la cláusula: Distinct

Por ejemplo:

```
select distinct libro.tema_clave, tema.tema_nombre  
from libro, tema  
where tema.tema_clave = libro.tema_clave
```

Cláusula Order By

Ordena el resultado de la consulta por los campos especificados, puede ser ascendente o descendente.

Por ejemplo:

Para seleccionar todos los campos de la tabla empleados, donde el campo sexo_emp = "F" y ordenar el resultado por los campos depto_emp y tipo_emp.

```
select * from empleados
where sexo_emp = "F"
order by depto_emp, tipo_emp
```

Cláusula Group By

En la cláusula "group by" se indica el o los campos por los cuales se desea agrupar un conjunto de registros. Comúnmente esta agrupación va acompañada con una serie de funciones que realizan ciertas operaciones sobre el valor de los campos indicados.

Algunas funciones de agrupación son las siguientes:

FUNCIÓN	ACCIÓN
Count(*)	Regresa el número de registros encontrados
Count(<campo>)	Regresa el número de registros cuyo valor del campo especificado no es nulo
Sum(<campo>)	Suma el valor del campo especificado
Avg(<campo>)	Promedia el valor del campo especificado
Min(<campo>)	Regresa el valor mínimo del campo especificado
Max(<campo>)	Regresa el valor máximo del campo especificado

Ejemplo:

Para obtener el número de empleados agrupados por cada valor distinto del campo tipo_emp

```
select tipo_emp, count(*)
from empleados
group by tipo_emp
```

Cláusula Having

Esta cláusula es el equivalente a la cláusula "where", es decir, especifica un criterio o condición; en este caso lo aplica al resultado de una función de agrupación utilizando la cláusula "group by".

Por ejemplo:

Para obtener los promedios mayores a 500 del campo sueldo_emp agrupado por cada valor distinto en el campo depto_emp

```
select depto_emp, avg(sueldo_emp)
from nomina
group by depto_emp
having avg(sueldo_emp) > 500
```

Funciones

Existen múltiples funciones en SQL para dar formato a las expresiones de tipo carácter, como "upper()" para unir cadenas como "concat()", etc., además de varias funciones matemáticas para realizar operaciones como "+", "-", "*" y "/", y funciones para manejar datos tipo fecha como "current()", "year()", etc

Cláusula Join

Amplia las capacidades consulta de información a la BD mediante la especificación de criterios de vinculación o relación entre ellas.

Sintaxis:

```
FROM ... nombre_tabla INNER JOIN nombre_tabla ON condición_Join WHERE
FROM... nombre_tabla LEFT [OUTER] JOIN nombre_tabla ON condición_Join
WHERE...
```

Donde condición_Join sigue las mismas características que la cláusula WHERE.

Por ejemplo:

```
Select *
from pais LEFT JOIN disco ON pais.pais_clave = disco.pais_clave
```

```
Select *
from disco RIGHT OUTER JOIN productora ON disco.productora_clave =
productora.productora_clave
```

Cláusula Union

Permite combinar el resultado de dos sentencias SQL en una sola.

```
SELECT
UNION [ALL]
SELECT ....
[UNION
SELECT ...]
```

Restricciones:

- Sólo la última sentencia SQL puede contener la cláusula ORDER BY.
- Si no se incluye la cláusula ALL solo se contemplarán los registros únicos, como si las sentencias tuvieran la cláusula DISTINCT

Por ejemplo:

```
Select * from pais1
UNION
Select * from pais2
```

Sentencias Anidadas

Es la inserción (del resultado) de una sentencia SQL dentro de otra sentencia SQL.

Por ejemplo:

```
Select *
From disco
Where pais_clave in (Select pais_clave from pais where zona = 1)
```

INSERCIÓN DE REGISTROS

Para insertar datos en una tabla se utiliza el comando "insert", el cual se puede utilizar en dos formas:

- Insertando un solo registro a la vez
- Insertando en la misma instrucción uno o más registros, en este caso, se combina con la instrucción SELECT

La sintaxis del comando INSERT tiene dos variantes: la primera de ellas consiste en especificar el nombre de cada uno de los campos y la lista de los valores que se desean insertar en correspondencia a cada uno de los campos especificados.

En este caso la sintaxis es la siguiente:

```
insert into nombre_tabla (campo1, campo2, campo3, ..... )
values (valor1, valor2, valor3.....)
```

En la sintaxis siguiente se omiten los nombres de los campos. En este caso, asume que se van a insertar valores en todos los campos de la tabla, por lo que se tienen que proporcionar valores para todos ellos y en el mismo orden en el cual están definidos en la tabla. En esta variante la sintaxis es la siguiente:

```
INSERT into nombre_tabla values (valor1, valor2, ....)
```

Cuando se desean insertar uno o más registros, estos deben de provenir de otra tabla, para lo cual el comando INSERT se combina con el comando SELECT que será el que seleccione los datos de otra tabla (origen) para insertarlos en la tabla (destino) indicada en el comando INSERT.

Al igual que en el punto anterior, se cuenta con dos variantes, en la primera de ellas se indican los nombres de los campos de la tabla donde se desea insertar los valores que serán seleccionados por la instrucción SELECT.

En este caso, los campos seleccionados en la instrucción SELECT (origen) deben estar en correspondencia con los campos especificados en la instrucción INSERT (destino).

La sintaxis es la siguiente:

```
insert into nombre tabla (campo1,campo2,...) SELECT....
```

ELIMINACIÓN DE REGISTROS

Existen dos maneras de eliminar registros de una tabla:

Delete

Mediante esta instrucción, el usuario puede eliminar registros especificando algún criterio o condición de eliminación; en caso de no hacerlo, se eliminan todos los registros de la tabla.

Sintaxis:

```
Delete nombre tabla  
[ From nombre_tabla1, nombre_tabla2, ... ]  
[ Where <condición> ]
```

Ejemplos:

```
delete venta where venta_clave=1120  
delete empleados  
from empleados, nomina  
where empleados.clave_emp = nomina.clave_emp  
and nomina.sueldo < 2500
```

Truncate

Esta instrucción también borra registros de una tabla, pero a diferencia de la instrucción “delete”, borra todos los registros sin dar la posibilidad de indicar algún criterio o condición de eliminación.

Sintaxis:

```
truncate table nombre_tabla
```

Ejemplo:

```
truncate carrito
truncate cliente
```

ACTUALIZACIÓN DE REGISTROS

Para modificar o actualizar los valores de los campos de una tabla se utiliza el comando UPDATE, en el cual se indica la tabla, los campos a actualizar, los nuevos valores y en su caso, la condición de la actualización. La sintaxis es la siguiente:

```
update nombre_tabla
set campo1 = valor1, campo2 = valor2, ....
[ From nombre_tabla1, .... ]
[ Where <condición> ]
```

Ejemplos:

```
update producto
set precio = precio * 1.1

update producto
set precio = precio * 1.1,
    precio_oferta = precio_oferta * 1.08
where tipo_articulo = "teclado"
```

Existen otros comando relacionado con la creación, actualización o eliminación de los objetos de una base de datos como "alter table", "drop view", etc, y algunos otros que sirven para otorgar permiso o definir roles de los usuarios de la base de datos como "grant" y "revoke".

INTEGRIDAD DE LOS DATOS

Las instrucciones que permiten mantener la integridad de los datos son:

Commit

Los cambios que se puedan estar realizando sobre la base de datos se hacen fijos únicamente al completar la transacción.

(Transacción: secuencia de operaciones tales que cada operación de la secuencia es necesaria para completar un resultado unitario. Todas las operaciones deben completarse para que la base de datos quede consistente)

En la base de datos de ejemplo, si eliminamos un proyecto será necesario eliminar las filas de la tabla "Trabajos" que pertenezcan al proyecto eliminado. Para mantener la integridad de la base de datos o se ejecutan las dos operaciones de borrado o no se debe ejecutar ninguna.

Rollback

Elimina todos los cambios que se hayan podido producir en la base de datos desde la ejecución de la última instrucción commit.

Si se produce un error de programa o un fallo hardware el sistema realiza un rollback automáticamente.

III.3 ANÁLISIS Y DISEÑO DE LA BASE DE DATOS

Para el análisis de la base de datos es necesario seguir los siguientes pasos para obtener, posteriormente, el modelo Entidad-Relación:

PASOS PARA MODELAR LA BASE DE DATOS

1. Identificar las posibles entidades
2. Revisar que las entidades estén en el contexto del sistema
3. Identificar los atributos de las entidades
4. Revisar que los atributos estén en el contexto del sistema.
5. Comenzar a definir las llaves primarias
6. Establecer las relaciones que existen entre las entidades (Asociación y tipos de relación). Vinculación.
7. Revisar que las relaciones estén en el contexto del sistema.
8. Aplicar primera y segunda forma normal. Romper asociaciones de muchos a muchos
9. Documentación. Establecer reglas, defaults, constrains, índices, definir tipos de datos, revisar los rangos, revisar si los campos son o no requeridos.
10. Hacer pruebas de escritorio

Los pasos anteriores no son más que una vía para llegar al modelo final, y no necesariamente se tienen que seguir en un orden secuencial, aunque si es recomendable. El proceso se repite hasta que se logra el resultado esperado: un modelo que represente la realidad de lo que se está analizando.

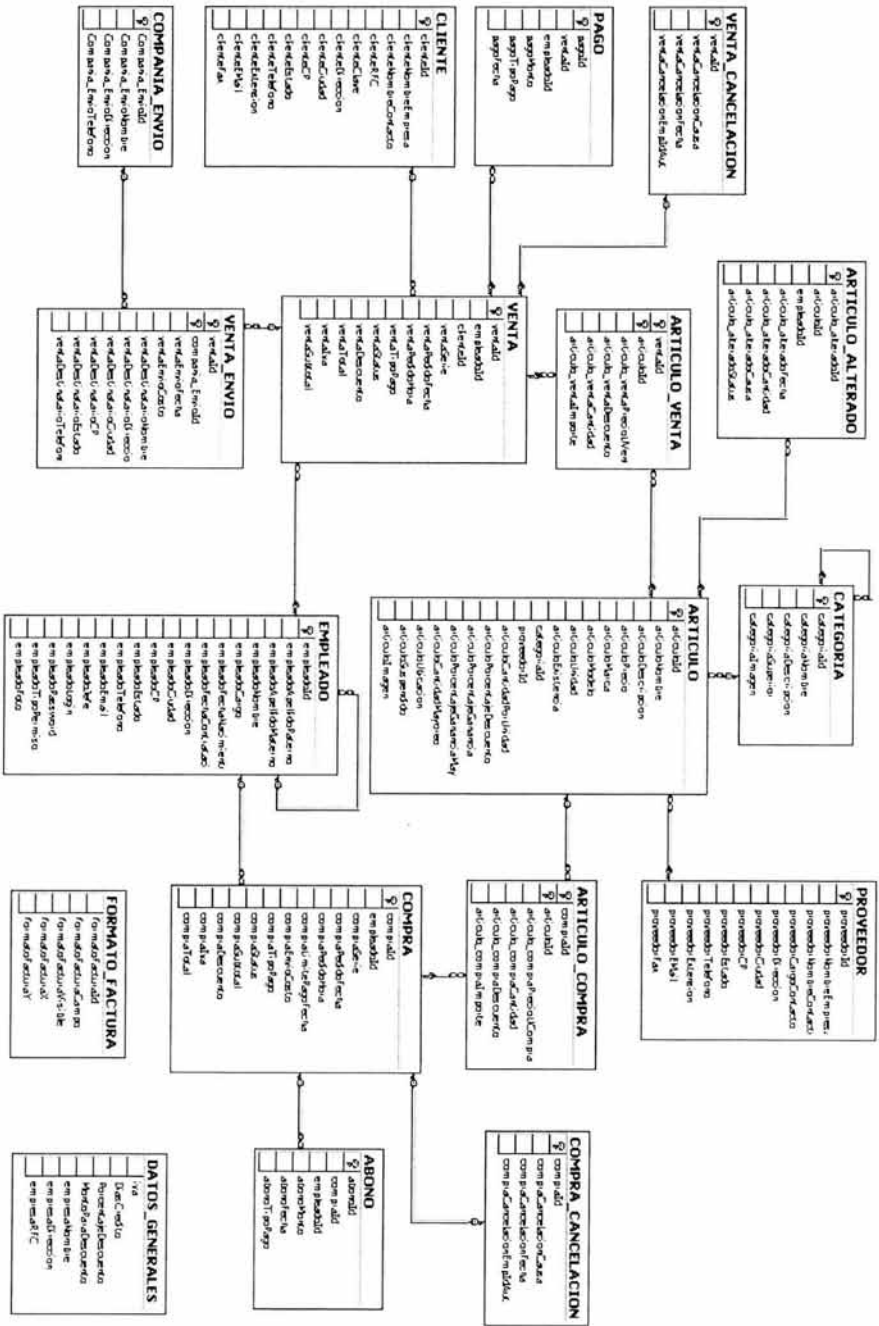
El modelo Entidad-Relación y la mayor parte del Diccionario de Datos se hicieron directamente en la aplicación incluida en SQL Server llamado Administrador Corporativo que permite crear el diagrama Entidad-Relación con todos sus componentes: entidades, atributos, relaciones, llaves primarias, etc.

DICCIONARIO DE DATOS

El contenido del diccionario de datos puede considerarse “datos acerca de los datos” (los cuales en ocasiones reciben el nombre de metadatos), es decir, definiciones de objetos de la base de datos, y no tan solo “datos en bruto”.

En el apéndice B se presenta el diccionario de datos del SAI, con las entidades y su descripción general, el nombre y definición de cada uno de sus atributos: su tipo de dato, longitud, si corresponde a un llave y de que tipo: llave primaria (PK) o llave foránea (FK), si es nulo y, finalmente, un ejemplo de los atributos.

DIAGRAMA ENTIDAD RELACION DEL SAI



La figura de la llave en las relaciones del modelo anterior representa “uno” y el símbolo de infinito “muchos”. Así, vemos que la mayoría de las relaciones son de “uno a muchos”.

Existen tablas llamadas transitivas que son utilizadas en una relación de “muchos a muchos”, por ejemplo, en el caso de la relación entre un artículo y una venta: los artículos pueden estar en una o varias ventas y una venta puede tener uno o más artículos, es decir, es una relación de “muchos a muchos”, por lo que se hace necesaria una tabla transitiva entre ellos, por tal razón se crea la tabla “Articulo_Compra”, que principalmente contiene las llaves primarias de ambas tablas relacionadas.

Hay otro tipo de relación, la relación recursiva. Este tipo de relación lo tenemos en dos tablas, en la tabla de empleado y en la tabla de categoría; se usa para relacionar un registro de una tabla con otro registro de la misma tabla. En el caso de los empleado tenemos que un empleado puede tener un jefe que al mismo tiempo sigue siendo empleado.

Por convención los nombres de las entidades se escriben en mayúsculas y los nombres de los atributos en letras minúsculas. En el Apéndice B se presenta el diccionario de datos.

IV

DISEÑO DE LA APLICACIÓN

IV.1 DISEÑO DEL SISTEMA

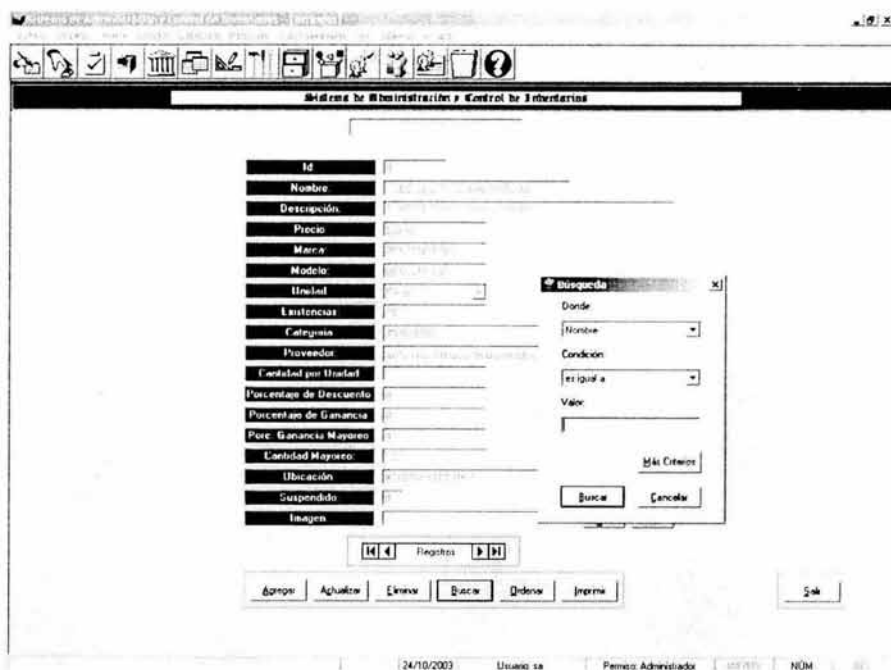
DISEÑO GENERAL

Tipo de Interfaz

En el Sistema de Administración y Control de Inventarios utiliza la interfaz de documentos múltiples (MDI). La interfaz de documentos múltiples permite crear una aplicación que contiene varias ventanas dentro de una única ventana contenedora.

Una aplicación MDI permite que el usuario vea varias ventanas al mismo tiempo. Las ventanas secundarias están contenidas en una ventana primaria, que proporciona un espacio de trabajo para todas las ventanas secundarias de la aplicación.

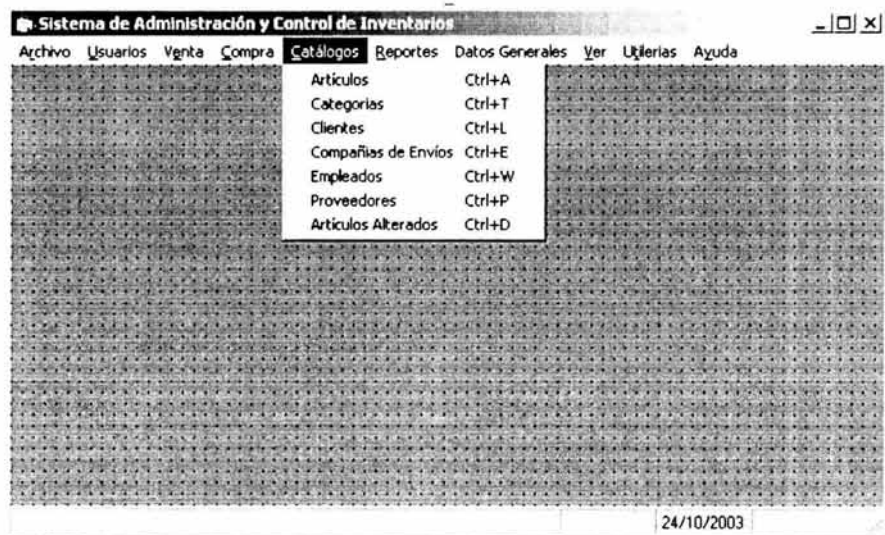
Cuando se minimiza la ventana secundaria, su icono aparece dentro del espacio de trabajo de la ventana MDI en lugar de aparecer en la barra de tareas. Sin embargo, SAI restringe algunas características de esta interfaz, por ejemplo, no permite que las ventanas hijas se puedan minimizar directamente, y sólo permite trabajar con un ventana secundaria de captura por razones de integridad con la información, excepto en los casos de búsquedas de registros y opciones de configuración. Como se ve en la siguiente figura:



Menús

Los menús proporcionan otra forma de acceder a los formularios o realizar alguna acción. Es común que las aplicaciones de Windows contengan una barra de menús donde se agrupan las distintas acciones que se pueden realizar dentro de la aplicación.

En el SAI la barra de menús se compone de varias categorías, con sus respectivas subcategorías como se muestra en la figura:



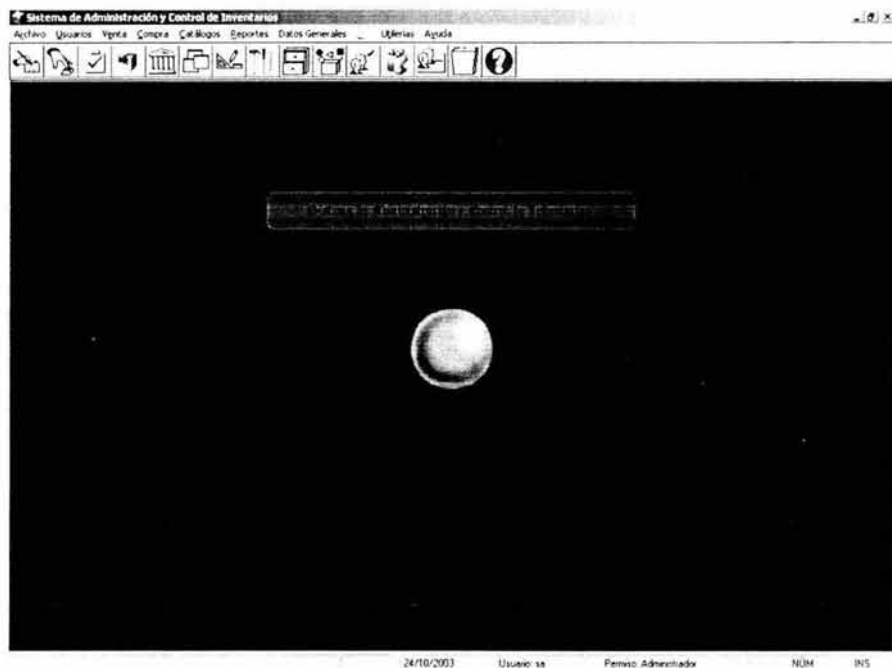
Cada menú tiene una tecla subrayada que combinada con la tecla ALT accede al elemento del menú. El menú se encuentra en la ventana primario o MDI

Las teclas del método abreviado ejecutan un elemento de menú inmediatamente cuando se presionan. Los elementos de menú que se utilizan con más frecuencia tienen una tecla del método abreviado, que proporciona un método de acceso por teclado de un solo paso, en lugar del método de tres pasos de presionar ALT, un carácter de acceso al título de menú y después un carácter de acceso al elemento del menú. Las combinaciones de métodos abreviados incluyen teclas de función y de control, como CTRL-F1-A. Aparecen en el menú a la derecha del elemento de menú correspondiente, como se muestra en la figura:

Venta	
Factura	Ctrl+F
Nota	Ctrl+N
Registro de Ventas	Ctrl+S

Barra de Herramientas y Barra de Estado

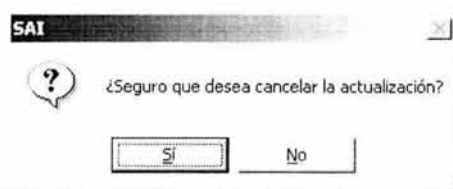
La barra de herramientas contiene botones que proporcionan un acceso rápido a los comandos de la aplicación que se usan con más frecuencia. La barra de herramientas es una característica estándar de muchas aplicaciones de Windows. En la siguiente figura la barra de herramientas aparece como el conjunto de iconos debajo de la barra de menú. Cada icono representa una acción y mientras se pasa el puntero del ratón sobre cualquiera de ellos muestra una descripción de la acción asociada.



La barra de estado se presenta en la parte inferior de la pantalla de la ventana primaria o contenedora. En ella se despliega información de la fecha actual, el usuario activo, el tipo de permiso que tiene y el estado de las teclas MAYUS, NUM e INS.

Cuadros de Diálogo

Los cuadros de diálogo tienen la forma:



Donde se muestra el título del sistema en la parte superior, el mensaje o pregunta en la parte central asociado con el icono correspondiente al mensaje del lado izquierdo, éste último puede ser de pregunta, de información, de exclamación o crítico y, finalmente, los botones de confirmación en la parte inferior

Secuencia en que se muestran las ventanas

Las ventanas hijas se cargan dentro del espacio de trabajo de la ventana padre una a la vez. Así, por ejemplo, si se está trabajando con el catálogo de proveedores, y posteriormente, se quiere capturar a un nuevo empleado, la ventana del catálogo de proveedores se cerrará para poder abrir la ventana del empleado. Este proceso lo hace el sistema automáticamente, pidiendo una previa confirmación para el cierre de la ventana en la que ya no se desea trabajar.

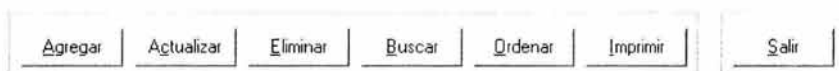
A pesar de esta restricción las ventanas hijas se comunican entre ellas aún y cuando no están cargadas al mismo tiempo. Por ejemplo, si dentro de una factura se quiere registrar a un nuevo cliente, esta ventana contiene un botón con una función para abrir la ventana donde se capture al nuevo cliente. Una vez que se escribió la información del cliente, esta ventana, a su vez, se cierra y carga la ventana de la factura automáticamente.

Existen otro tipo de ventanas hijas que se pueden abrir junto con otras, tal es el caso de las ventanas de búsqueda y las que contienen información que no está relacionada con la integridad de la base de datos. Estas ventanas mantienen el enfoque mientras están cargadas, de tal forma que no se pueda realizar otra acción dentro de la aplicación mientras estén visibles.

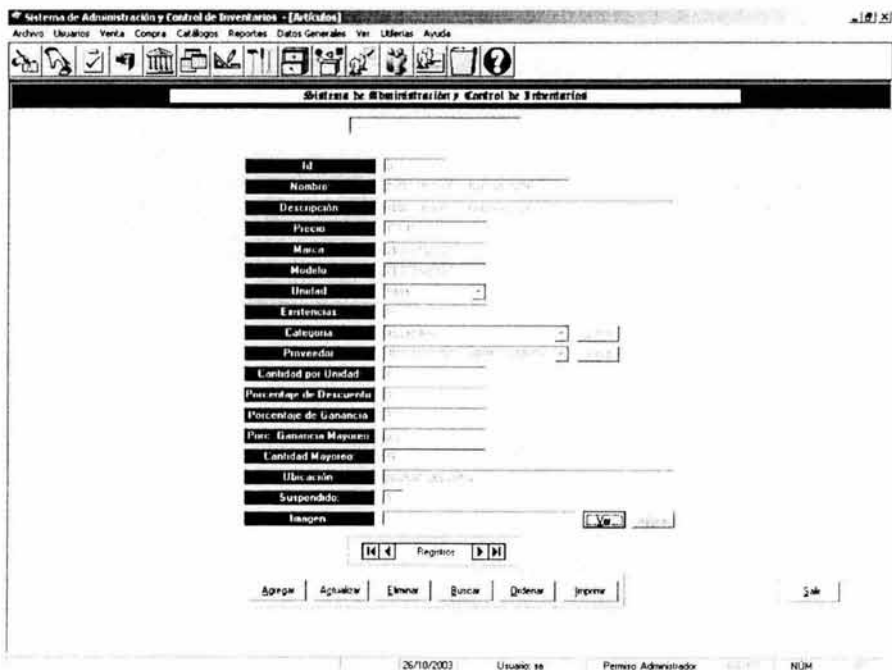
Botones y acciones comunes

Los botones para confirmar alguna acción van del lado izquierdo y para cancelar una acción van del lado derecho. Cada uno de ellos tiene una tecla subrayada para acceder a su función por medio del teclado con la tecla ALT y el carácter subrayado.

Los botones comunes a todos los catálogos de la aplicación son “Agregar”, “Actualizar”, “Eliminar”, “Buscar”, “Ordenar”, “Imprimir” y “Salir”, que corresponden a las acciones del registro mostrado en ese momento. Se muestran en la siguiente figura:



Los botones asociados a todos los registros de la ventana o comunes van en la parte inferior. Mientras que los botones que contienen acciones para un registro específico están al lado izquierdo de éste, como se muestra en la siguiente figura, para categoría, proveedor e imagen:



Navegación a través de los registros

La navegación a través de los registros se hace a través de un control de navegación que no es más que cuatro botones con acciones específicas: los dos botones del lado izquierdo son para ir al primer registro y al registro anterior y los botones del lado derecho son para ir al siguiente registro y al último, respectivamente.



Maestro-Detalle

Existe un diseño para los casos en donde la información es del tipo Maestro-Detalle, es decir, donde a un registro le corresponden varios registros, como en el caso de las compras donde se tienen los datos generales de la compra y sus movimientos.

Para registrar la información de este tipo se hace uso de una interfaz, que en la parte superior de la ventana enmarca la información de los datos del "Maestro" o datos generales, en la parte central los datos del "Detalles" y en la parte inferior los totales correspondientes también a los datos "Maestro".

La captura de los datos generales se realiza una sola vez directamente en los controles asignados a cada campo, mientras que el registro de los movimientos, se hace en la sección adecuada, mediante controles que insertan en la cuadrícula que está debajo de ellos todos los movimientos deseados. El ejemplo lo podemos ver en la siguiente figura:

Sistema de Administración y Control de Inventarios [Compra] [X] [X]

Archivo Usuario Venta Compra Catálogos Reportes Datos Generales Utileras Ayuda

Registro de Compras

Datos de compra:

Pago: Pagos con IVA incluido
 Fecha: Descuento
 Status: Actualizar precios de artículos automáticamente
 Proveedor: MAT ELECTRICOS PATO BAÑERA DE SFOM

Buscar:

Movimientos de compra:

Id	Categoría del Artículo	Precio	Cantidad	Importe
Movimientos de la Compra				
Id	Nombre	Precio Unitario	Cantidad	Importe

26/10/2003 Usuario: sa Permiso: Administrador NÚM: 1

Por otra parte, la consulta de este tipo de información se puede obtener de dos formas diferentes: una es por medio de una ventana común de la aplicación y la otra es por medio de reportes que se verá más adelante en este mismo capítulo.

La consulta por medio de una ventana común es de la siguiente forma: los datos generales están en la cuadrícula de la parte superior, y los detalles se presentan en la cuadrícula inferior. Para actualizar los detalles de cada registro "Maestro" es necesario dar clic en el renglón correspondiente.

Sistema de Administración y Control de Inventarios [Compra] [X] [X]

Archivo Usuario Venta Compra Catálogos Reportes Datos Generales Utileras Ayuda

Registro de Compras

Compras									
Id de Compra	Fecha	Proveedor	Importe	Categoría	Status	Fecha de Pago	Usuario	Permiso	Acción
1	07/10/2003	MAT ELECTRICOS PATO BAÑERA DE SFOM	11.83	PLACA DIVISIONAL ELECTRONICA	Pagado - Entregado	07/10/2003	sa	Administrador	Editar

Movimientos						
Id de Artículo	Artículo	Precio	Importe	Cantidad	Importe	Categoría
1	PLACA DIVISIONAL ELECTRONICA	11.83	11.83	1.00	11.83	PLACA DIVISIONAL ELECTRONICA

26/10/2003 Usuario: sa Permiso: Administrador NÚM: 1

DISEÑO DE LAS VENTANAS DE CAPTURA

Vistas

El SAI maneja, para sus catálogos, el tipo de vista tipo formulario y la vista tipo cuadrícula. Para ambas vistas se maneja el mismo tipo de acciones que puede realizar el usuario. La diferencia principal entre ellas es que mientras en la primera se muestra la información registro por registro, en la segunda se muestra la información de todos los registros en una cuadrícula. Para manipular los datos es más cómoda la vista de formulario, mientras que para consultar información se recomienda la vista tipo cuadrícula.

Vista Formulario

Como se mencionó anteriormente, dentro del tipo de vista formulario la información se despliega registro por registro. Mientras se trate de consultar la información, los cuadros de texto aparecen deshabilitados y se puede navegar a través de ellos. El conjunto de registro se puede ordenar con el botón "Ordenar" y se pueden realizar búsquedas por medio del botón "Buscar" para trabajar solo con aquellos registros que satisfagan los criterios requeridos. Una vez que se accede a la información del registro deseado, este puede ser actualizado o eliminado. Cuando se da clic en los botones de "Agregar" y "Actualizar" los controles se habilitan para la captura.

The screenshot displays the 'Sistema de Administración y Control de Inventarios' (SAI) application window. The title bar reads 'Sistema de Administración y Control de Inventarios - [Escribir...]' and the menu bar includes 'Archivo', 'Usuarios', 'Venta', 'Compras', 'Catálogos', 'Reportes', 'Datos Generales', and 'Ayuda'. The main window title is 'Sistema de Administración y Control de Inventarios'. The form contains the following fields:

- Id
- Apellido Paterno
- Apellido Materno
- Nombre
- Cargo
- Fecha de Nacimiento
- Fecha de Contratación
- Dirección
- Ciudad
- Código Postal
- Estado
- Teléfono
- Correo Electrónico
- Sexo
- Edad
- Password
- Tipo de Permiso
- Fotografía

Navigation and action buttons at the bottom include: 'Registros', 'Agregar', 'Actualizar', 'Eliminar', 'Buscar', 'Ordenar', 'Imprimir', and 'Salir'. The status bar at the bottom shows the date '26/10/2003', the user 'Usuario sa', the role 'Permiso Administrador', and other system information.

Vista Cuadrícula

La vista cuadrícula tiene la peculiaridad de mostrar la totalidad de la información o el resultado de la búsqueda en una sola pantalla. Tiene la ventaja de consultar la información rápidamente y permite al usuario hacer comparaciones fácilmente. Todo lo que se puede hacer en vista formulario también se puede hacer en vista cuadrícula. En esta parte si se desea trabajar con un registro específico basta con hacer clic del lado izquierdo del registro donde aparece el triángulo que representa un apuntador al registro.

Empleados							
Id	Apellido Paterno	Apellido Materno	Nombre	Cargo	Fecha de Nacimiento	Fecha de Contratación	Dirección
1	Ramirez	Garcia	Jose Luis	Empleado	14/07/2003	14/07/2003	
2	Lopez	Ortega	David	Gerente General	14/07/2003	14/07/2003	Bosques de Blas

Facilidad de captura

Una de las características importantes del SAI es que facilita la captura de una manera importante. Hace uso especial del teclado para la introducción de los datos por parte del usuario. Dentro de las ventanas para los catálogos esto es válido principalmente para la vista de formulario en donde el ratón no es requerido puesto que el teclado es capaz de acceder a todas las opciones.

En las ventanas de captura de compras y de ventas se hace un uso aún más interesante del teclado: dado que la tecla ENTER es de las más usadas, el usuario requería que el paso a través de los distintos controles de captura se hiciera a través de ésta tecla de control. De esta forma se agiliza notoriamente la captura.

Enfoque

Cuando un cuadro de texto con información recibe el enfoque y éste contiene información de forma automática los caracteres que contiene se seleccionan. Cada uno de los controles tiene un orden de tabulación, por lo que al utilizar la tecla de tabulador estos recibirán el enfoque en un orden predeterminado excepto aquellos que estén deshabilitados. En el caso de existir algún error de validación el enfoque se establece al control donde existe error.

Validaciones

Las validaciones son indispensables para mantener la consistencia e integridad de la información en cualquier aplicación. Las validaciones son uno de los aspectos más importantes para el adecuado desempeño del SAI.

El SAI cuenta con las siguientes validaciones:

- Cuando un campo es de tipo numérico solo se permite la captura de números, cualquier tecla que no sea número el sistema la cancela automáticamente.
- Validaciones de números positivos. No se permite la captura de número negativos en campos que no los requieren.
- Para la validación de los campos tipo fecha utiliza un control llamado calendario y de otro control que cumple la función de una máscara de tipo fecha: “ / / ” que permite solo caracteres numéricos.
- La información requerida en la base de datos, es decir, la marcada como no nula es pedida por el sistema y no se puede continuar con una siguiente acción hasta que se capture o, en su defecto se cancele la captura.
- Se establece un número máximo de caracteres de acuerdo a la definición del tamaño de dato en la base de datos. Esto es, si un campo tipo cadena tiene una longitud de 20 caracteres en la base de datos, la aplicación ajusta el control asociado a éste campo de tal forma que solo permita al usuario capturar 20 caracteres, se le conoce a esta propiedad como “Longitud Máxima”
- La información relacionada de dos tablas se hace desde el nivel aplicación por medio de listas desplegables. Por ejemplo en una venta no se puede registrar a un cliente que no exista en una lista donde se muestran todos los clientes registrados.

BÚSQUEDAS

Para encontrar un registro o un conjunto de registros se hace uso de la interfaz de búsqueda que es común para todas las ventanas del catálogo

La ventana de búsqueda consta de tres partes principales:

- Campo. Es el nombre del campo que contendrá el criterio
- Valor. Es el valor de la condición que servirá de filtro para la búsqueda
- Condición. Es el operador que relaciona al campo con su valor.

En general, la estructura de la búsqueda es similar a la cláusula WHERE de SQL, del tipo "WHERE campo operador valor". Con la ventaja de que esta interfaz despliega los campos válidos y los operadores con un lenguaje en español.

Internamente, se hace la validación del tipo de dato. Por ejemplo, si se pretende hacer una búsqueda de un campo numérico con un valor alfabético marca un error.

Esta pantalla permite la introducción de hasta dos criterios y permite relacionarlos con los operadores de relación AND u OR que son presentados al usuario como "O" e "Y"

The screenshot shows a search window titled "Búsqueda" with a close button (X) in the top right corner. The window is divided into two columns for search criteria. The left column has a "Donde:" label above a dropdown menu containing "Descripcion", a "Condición:" label above a dropdown menu containing "es como", and a "Valor:" label above a text input field containing "foco". The right column has a "Donde:" label above a dropdown menu containing "Precio", a "Condición:" label above a dropdown menu containing "es mayor que", and a "Valor:" label above a text input field containing "10". Between the two columns is a dropdown menu containing "Y". At the bottom right of the window are two buttons: "Buscar" and "Cancelar".

DISEÑO DE LOS REPORTES

Tipo de Interfaz

Las ventanas de tipo reporte son las únicas que no están dentro del entorno de interfaz de aplicaciones múltiples que maneja el SAI con el propósito de mantener varios reportes cargados en memoria y poder hacer comparaciones. De esta forma cada uno de ellos tiene su propia ventana independiente de la ventana MDI a partir de la cual fueron creados.

Barra de Herramientas

En la barra de herramientas de los reportes se encuentra el control de tamaño que permite a los repostes ajustarse a diferentes dimensiones. También se pueden imprimir o exportar a otras aplicaciones con los iconos respectivos que aparecen en la parte superior de la pantalla.

Información de los reportes

Todos los reportes tienen un encabezado común que contiene del lado izquierdo el nombre del sistema, del lado derecho la fecha actual y el número de página, en la parte central muestra el nombre de la empresa, su dirección y su RFC. Más abajo se presenta el título del reporte específico.

Si se trata de un reporte que muestra información del tipo “Maestro-Detalle” los datos correspondientes al “Maestro” se despliegan en la parte superior, los detalles en la parte central y los totales, en la parte inferior. Cada campo tiene el formato de su tipo de dato, es decir, si es un dato que se trata de dinero, el valor se alinea a la derecha y contiene el símbolo de pesos “\$”. El nombre de los campos se pinta en negritas.

SAI ver: 1.0 Domingo, 26 de Octubre de 2003
Pag: 1

**SISTEMAS ICOM
MEXICO DF
RFC:ICOM**

COMPRA

Id de Compra: 2
Límite de Pago: 01/01/2004 **Status:** Pagado - Entregado
Serie: AA **Condición:** Contado
Costo de Envío: \$0.00 **Fecha de Pedido:** 26/10/2003

Nombre del Proveedor: MAT ELECTRICOS PATO BAEZA, DESCONOCIDO
Dirección del Proveedor: CALZADA DE LA VIGA NO 602 CD DE MEXICO 00200 D.F.
Nombre del Empleado: Administrador Principal

Id	Nombre	P. Unitario	Cantidad	Descuento	Importe
1	FOCO DE HORNO REFRI 250V	\$4.05	10	0%	\$40.50
2	PLACA 2U ANONIZADO ORO	\$11.33	3	0%	\$33.99

Subtotal: \$73.99
Descuento: \$9.98
Iva: \$11.98
Total: \$64.97

Página: 1/4

Para los catálogos existen dos tipos de reportes, los detallados y los que muestran un resumen de los registros, como se ve en las siguientes figuras para el caso del catálogo de artículos:

Formato detallado:

SAI ver: 1.0 Domingo, 26 de Octubre de 2003

**SISTEMAS ICOM
MEXICO DF
IFCICOM**

ARTÍCULOS

Id:	0	Precio:	529.45
Nombre:	TUBO DE 3/4 P.D. GALVANIZADO	Unidad:	Pieza
Marca:	DESCONOCIDA	Modelo:	DESCONOCI
Existencia:	15		
Descripción:	TUBO DE 3/4 P.D. GALVANIZADO		
Ubicación:	BOCOTA SECCION 3		
Porcentaje de Ganancia:	30%	Porcentaje Descuento:	0%
Porcentaje de Gan. May:	30%	Cantidad Mayorero:	12
Cantidad por Unidad:	1	Suspendido:	0
Nombre de Categoría:	PLUMERIA	Id de Categoría:	1
Nombre de Proveedor:	GPO ELECTRICO LARRUS SA DE CV	Id de Proveedor:	1

Número total de Artículos: 1

Page: 1 | 1

Formato resumido:

SAI ver: 1.0 Domingo, 26 de Octubre de 2003

**SISTEMAS ICOM
MEXICO DF
IFCICOM**

ARTÍCULOS

Id	Nombre	Precio	Marca	Modelo
0	TUBO DE 3/4 P.D. GALVANIZADO	529.45	DESCONOCIDA	DESCONOCIDO
1	TUBO DE 1/2 P.D. GALVANIZADO	54.95	DESCONOCIDA	DESCONOCIDO
2	PLACA DE ANCLAJE 20X20	10.00	DESCONOCIDA	DESCONOCIDA
3	PLACA DE ANCLAJE 25X25	12.00	DESCONOCIDA	DESCONOCIDA
4	PLACA DE ANCLAJE 30X30	14.00	DESCONOCIDA	DESCONOCIDA
5	PLACA DE ANCLAJE 35X35	16.00	DESCONOCIDA	DESCONOCIDA
6	PLACA DE ANCLAJE 40X40	18.00	DESCONOCIDA	DESCONOCIDA
7	PLACA DE ANCLAJE 45X45	20.00	DESCONOCIDA	DESCONOCIDA
8	PLACA DE ANCLAJE 50X50	22.00	DESCONOCIDA	DESCONOCIDA
9	PLACA DE ANCLAJE 55X55	24.00	DESCONOCIDA	DESCONOCIDA
10	PLACA DE ANCLAJE 60X60	26.00	DESCONOCIDA	DESCONOCIDA
11	PLACA DE ANCLAJE 65X65	28.00	DESCONOCIDA	DESCONOCIDA
12	PLACA DE ANCLAJE 70X70	30.00	DESCONOCIDA	DESCONOCIDA
13	PLACA DE ANCLAJE 75X75	32.00	DESCONOCIDA	DESCONOCIDA
14	PLACA DE ANCLAJE 80X80	34.00	DESCONOCIDA	DESCONOCIDA
15	PLACA DE ANCLAJE 85X85	36.00	DESCONOCIDA	DESCONOCIDA
16	PLACA DE ANCLAJE 90X90	38.00	DESCONOCIDA	DESCONOCIDA

Número total de Artículos: 11

Inicio | Sistema de | Redccion | Tercer - 16 | IV.3 Desc... | Factura | Artículos | Artículos | 12:19 p.m.

IV.2 DESARROLLO DE SOFTWARE

LENGUAJE DE PROGRAMACIÓN VISUAL BASIC

La palabra "Visual" hace referencia al método que se utiliza para crear la interfaz gráfica de usuario (GUI). En lugar de escribir numerosas líneas de código para describir la apariencia y la ubicación de los elementos de la interfaz, simplemente se agregan objetos prefabricados en su lugar dentro de la pantalla.

La palabra "Basic" hace referencia al lenguaje BASIC (Beginners All-Purpose Symbolic Instruction Code), un lenguaje utilizado por más programadores que ningún otro lenguaje en la historia de la informática o computación. Visual Basic ha evolucionado a partir del lenguaje BASIC original y ahora contiene centenares de instrucciones, funciones y palabras clave, muchas de las cuales están directamente relacionadas con la interfaz gráfica de Windows.

El lenguaje Visual Basic es muy potente: si podemos imaginar una tarea de programación, probablemente se podrá realizar con Visual Basic.

Conceptos de Visual Basic

Para entender el proceso de desarrollo de una aplicación, es útil comprender algunos de los conceptos clave alrededor de los cuales está construido Visual Basic. Puesto que Visual Basic es un lenguaje de desarrollo para Windows, es necesario familiarizarse con el entorno Windows.

Funcionamiento de Windows: ventanas, eventos y mensajes

Una versión reducida del funcionamiento de Windows incluye tres conceptos clave: ventanas, eventos y mensajes. Una ventana es simplemente una región rectangular con sus propios límites.

Hay varios tipos de ventanas: una ventana Explorador en Windows, una ventana de documento dentro de un programa de proceso de textos o un cuadro de diálogo que emerge para recordar una cita. Aunque éstos son los ejemplos más comunes, realmente hay otros muchos tipos de ventanas. Un botón de comando es una ventana. Los iconos, cuadros de texto, botones de opción y barras de menús son todos ventanas.

El sistema operativo Microsoft Windows administra todas estas ventanas asignando a cada una un único número identificador (controlador de ventana o hWnd). El sistema controla continuamente cada una de estas ventanas para ver si existen signos de actividad o eventos. Los eventos pueden producirse mediante acciones del usuario, como hacer clic con el ratón o presionar una tecla, mediante programación o incluso como resultado de acciones de otras ventanas.

Cada vez que se produce un evento se envía un mensaje al sistema operativo. El sistema procesa el mensaje y lo transmite a las demás ventanas. Entonces, cada ventana puede realizar la acción apropiada, basándose en sus propias instrucciones para tratar ese mensaje en particular (por ejemplo, volverse a dibujar cuando otra ventana la ha dejado al descubierto).

Muchos de los mensajes los controla automáticamente Visual Basic, mientras que otros se tratan como procedimientos de evento. Esto permite crear rápidamente eficaces aplicaciones sin tener que tratar detalles innecesarios.

Descripción del modelo controlado por eventos

En las aplicaciones tradicionales o "por procedimientos", la aplicación es la que controla qué partes de código y en qué secuencia se ejecutan. La ejecución comienza con la primera línea de código y continúa con una ruta predefinida a través de la aplicación, llamando a los procedimientos según se necesiten.

En una aplicación controlada por eventos, el código no sigue una ruta predeterminada; ejecuta distintas secciones de código como respuesta a los eventos. Los eventos pueden desencadenarse por acciones del usuario, por mensajes del sistema o de otras aplicaciones, o incluso por la propia aplicación. La secuencia de estos eventos determina la secuencia en la que se ejecuta el código, por lo que la ruta a través del código de la aplicación es diferente cada vez que se ejecuta el programa.

Puesto que no puede predecir la secuencia de los eventos, el código debe establecer ciertos supuestos acerca del "estado del mundo" cuando se ejecute.

El código también puede desencadenar eventos durante la ejecución. Por ejemplo, cambiar mediante programación el texto de un cuadro de texto hace que se produzca el evento Change del cuadro de texto. Esto causaría la ejecución del código (si lo hay) contenido en el evento Change.

El proceso de desarrollo de las aplicaciones tradicionales se puede dividir en tres etapas diferentes: escritura, compilación y comprobación del código. A diferencia de los lenguajes tradicionales, Visual Basic utiliza una aproximación interactiva para el desarrollo, difuminando la distinción entre los tres pasos.

En la mayoría de los lenguajes, si comete un error al escribir el código, el compilador intercepta este error cuando comience a compilar la aplicación. Debe encontrar y corregir el error y comenzar de nuevo con el ciclo de compilación, repitiendo el proceso para cada error encontrado. Visual Basic interpreta el código a medida que lo escribe, interceptando y resaltando la mayoría de los errores de sintaxis en el momento. Es casi como tener un experto vigilando cómo escribe el código.

Para interceptar errores sobre la marcha, Visual Basic compila parcialmente el código según se escribe. Cuando se está preparado para ejecutar y probar la aplicación, es poco el tiempo que tardará en terminar la compilación. Si el compilador encuentra un error, quedará resaltado en el código. Se puede corregir el error y seguir compilando sin tener que comenzar de nuevo.

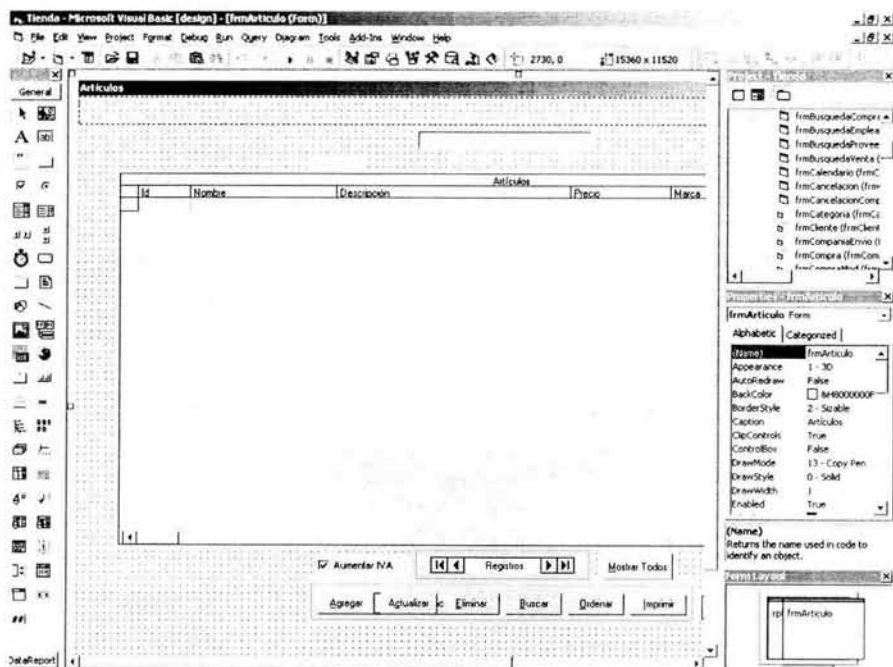
A causa de la naturaleza interactiva de Visual Basic, se es común ejecutar la aplicación frecuentemente a medida que se desarrolla. De esta forma se puede probar los efectos del código según se escriba en lugar de esperar a compilarlo más tarde.

AMBIENTE DE DESARROLLO

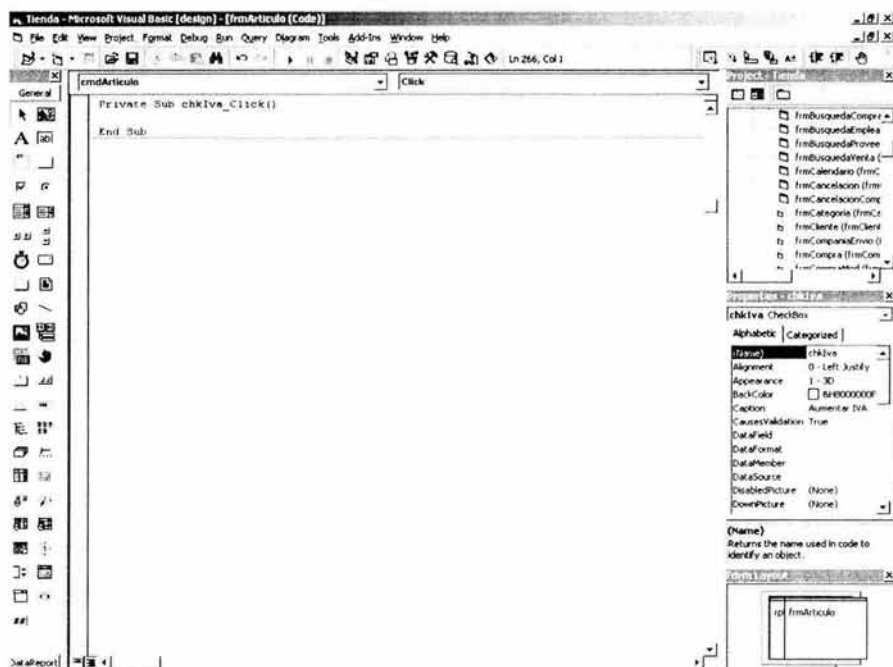
Interfaz gráfica

El ambiente de desarrollo de Visual Basic es un componente de Visual Studio, este ambiente esta compuesto por una interfaz gráfica, mostrado en la siguiente figura, que tiene varios elementos, entre los más importantes tenemos:

- Ventana de los controles del lado derecho en la figura. Aquí están todos los controles que se pueden utilizar para el desarrollo de la aplicación: los cuadros de texto, las etiquetas, los botones, las cuadrículas, las listas desplegables, etc. Basta con arrastrar el objeto deseado y “pintarlo” en el formularios para tener accesos a sus propiedades
- Ventana de formulario en su vista de diseño, la centro de la figura. Aquí se muestra el diseño del formulario.
- Ventana del explorador de proyectos del lado derecho en la parte superior. Esta ventana presenta los archivos que componen el proyecto. Desde aquí se administran todos los archivos requeridos en la aplicación
- Ventana de Propiedades, del lado derecho de la figura en la parte inferior. En esta sección podemos manipular y consultar las propiedades relativas al objeto seleccionado actualmente.



Dentro del entorno de desarrollo podemos, obviamente, escribir el código. Se puede acceder a él de muchas formas, una de ellas es dando doble clic en el objeto del cual se desea trabajar su código, como se ve en la siguiente figura:



ESTRUCTURA DE UN PROYECTO DE VISUAL BASIC

Las siguientes secciones describen los diferentes tipos de archivos usados para el presente proyecto.

Módulos de formulario

Los módulos de formulario (extensión de nombre de archivo .frm) pueden contener texto descriptivo del formulario y sus controles, incluyendo los valores de sus propiedades. También pueden contener declaraciones de formulario de constantes, variables y procedimientos externos, así como procedimientos de evento y procedimientos generales.

Módulos de clase

Los módulos de clase (extensión de nombre de archivo .cls) son similares a los módulos de formulario, excepto en que no tienen interfaz de usuario visible. Puede usar módulos de clase para crear sus propios objetos, incluyendo código para métodos y propiedades.

Módulos estándar

Los módulos estándar (extensión de nombre de archivo .bas) pueden contener declaraciones públicas o a nivel de módulo de tipos, constantes, variables, procedimientos externos y procedimientos públicos.

Archivos de recursos

Los archivos de recursos (extensión de nombre de archivo .res) contienen mapas de bits, cadenas de texto y otros datos que se pueden modificar sin volver a modificar el código. Por ejemplo, si se piensa localizar la aplicación a un idioma extranjero, puede guardar todas las cadenas de texto de la interfaz de usuario y los mapas de bits en un archivo de recursos, y simplemente traducir el archivo de recursos en vez de la aplicación completa. Un proyecto sólo puede contener un archivo de recursos.

Componentes

Además de archivos y módulos, también es posible agregar otro tipo de componentes a un proyecto.

Los controles ActiveX (extensión de nombre de archivo .ocx) son controles opcionales que se pueden agregar al cuadro de herramientas y se pueden usar en formularios. Cuando instala Visual Basic, los archivos que contienen los controles incluidos en Visual Basic se copian a un directorio común (el subdirectorio \Windows\System en Windows). Existen controles ActiveX adicionales disponibles en diversas fuentes.

VARIABLES Y TIPOS DE DATOS

Variables

Declarar una variable es decirle al programa algo de antemano. Se declara una variable mediante la instrucción Dim, proporcionando un nombre a la variable:

```
Dim nombreVariable [As tipo]
```

Los tipos de datos más usados son integer, double, boolean, string, date y currency.

Variables globales

Las variables globales se declaran preferentemente en un módulo estándar y se declaran con la palabra reservada "Public" seguida del nombre de la variable y su tipo de dato.

Declarar variables con tipos de datos

Antes de usar una variable se deben usar las instrucciones Private, Public, Dim o Static para declararla y su tipo. Por ejemplo, la siguiente instrucción declara un tipo Integer, Double, String y Currency, respectivamente:

```
Private I As Integer
Dim Cantidad As Double
Static SuNombre As String
Public PagadoPorJuan As Currency
```

Convenciones de nombres de variables y de objetos

Se usa la notación húngara de prefijo para nombrar variables y objetos, se compone de tres letras en minúscula seguidas del nombre de la variable. Si una variable tiene más de dos palabras éstas se separan con letras mayúsculas de la primera letra de la palabra que continua. La notación para las variables y objetos más utilizados es la siguiente:

```
Integer: intConsecutivo
Double: dblKilometros
String: strNombre
Currency: curPrecio
Recordset: rsArticulo
Connection: cnSqlServer
```

Además si se trata de variables globales se les agrega la letra g al prefijo:

```
Integer: gintIdVentana
```

PROCEDIMIENTOS

Un procedimiento general indica a la aplicación cómo realizar una tarea específica. Una vez que se define un procedimiento general, se le debe llamar específicamente desde la aplicación. Por el contrario, un procedimiento de evento permanece inactivo hasta que se le llama para responder a eventos provocados por el usuario o desencadenados por el sistema.

Un ejemplo de un procedimiento lo tenemos en CrearConjuntoResultados utilizado en la aplicación:

```
Public Sub CrearConjuntoResultados(strQuery As String, rsConjuntoResultados As
Recordset, cn As Connection)
    On Error GoTo error
    If rsConjuntoResultados.State = adStateClosed Then
        rsConjuntoResultados.Open strQuery, cn, adOpenStatic, adLockOptimistic
    End If
Continuar:
    Exit Sub
error:
    MsgBox "Conexión no establecida o error con los registros", vbInformation,
"Aceptar"
    Resume Continuar
End Sub
```

Esta función recibe 3 parámetros: la consulta SQL, un objeto Recordset que es el que contendrá el conjunto de resultados y un objeto Connection que contiene todas las propiedades de la conexión con la base de datos.

La instrucción “On Error GoTo error” tiene la función de atrapar los errores del código contenido entre esta instrucción y la etiqueta que tiene definida y si éstos errores existen manda el flujo ejecución después de la etiqueta, en este caso la etiqueta es “error”, muestra el mensaje de error y sale del procedimiento

Posteriormente compara el estado del recordset y si éste no esta cerrado crea el conjunto de resultados y sale del procedimiento. Aquí es importante señalar que es un procedimiento porque no regresa ningún valor. Si hubiera sido necesario regresar un valor se necesitaría declarar el procedimiento anterior como function y declarar el valor de retorno.

ESTRUCTURAS DE CONTROL

Al igual que la gran mayoría de los lenguajes de programación Visual Basic cuenta con las estructuras de control mas comunes tales como la de decisión como el if .. then y el select similar al if y al swich de “C” respectivamente.

Las estructuras de repetición o bucle son las siguientes:

- Do condición Loop
- Do condición While
Loop
- For contador = iniciar To finalizar [Step incremento]
instrucciones
Next [contador]
- For Each...Next

Cada uno con un propósito similar: de realizar ciclos con las instrucciones contenidas dentro de ellos mientras la condición evaluada sea verdadera. En especial el For Each hace que se recorran los objetos contenidos en una colección de Visual Basic.

OBJETOS

Un objeto es una combinación de código y datos que se puede tratar como una unidad. Un objeto puede ser una parte de una aplicación, como un control o un formulario. Los objetos en Visual Basic los encontramos en los botones, en las bases de datos, en los gráficos, etc. La sintaxis para acceder a las propiedades o procedimientos de un objeto es nombreObjeto.[procedimiento | propiedad]

Los principales objetos utilizados en esta aplicación son los formularios (Form) y los controles asociados como cuadros de texto (Textbox), cuadros de lista (ComboBox) y cuadrículas (DataGrid) para las ventanas de captura, los objetos Connection, los recordset, y ADO para utilizar la base de datos y los DataReport, el DataEnvironment para los reportes

Formularios

Los formularios son los contenedores de los demás controles u objetos dentro de una aplicación de Visual Basic, se asocian directamente a lo que conocemos como una ventana de Windows, ya que con ellos establecemos muchas de las propiedades de las ventanas como su tamaño, color ubicación, etc.. Existen formularios de interfaz de documentos múltiples (MDI) donde existe una ventana contenedora de las demás. La otra interfaz se conoce como interfaz de documento simple (SDI) en donde la aplicación se ejecuta en una sola ventana, como en el caso del bloc de notas de Windows. Para el SAI se utilizaron ambos tipos de formularios, el SDI para los reportes y el MDI para los demás formularios, sin embargo se puede decir que SAI es del tipo MDI.

Conexiones

El objeto Connection de Visual Basic permite, como su nombre lo indica conectarnos a una base de datos especificándole una cadena de conexión que contiene los detalles del servidor al que se va a conectar, el nombre de la base de datos y los parámetros de la seguridad como el nombre de usuario y la contraseña:

Recordset

Los Recordset o conjunto de resultados son los objetos encargados de mantener los datos obtenidos de una base de datos en una variable de tipo objeto, es como un cursor de cualquier manejador de base de datos en los que uno puede navegar a través de los registros. Pueden ser dinámicos o estáticos, esto es, pueden ser manipulados como si estuviera en la base de datos o tomar una fotografía de los datos para consultarlos o para después regresar estos datos modificados a la base de datos.

Una de las propiedades principales de los Recordset es el procedimiento Open que tiene como parámetros la conexión a la base de datos y el query o consulta entre otros.

Módulos

En los archivos de tipo módulo estándar se agrupan las variables y funciones que tienen un alcance global en la aplicación, es decir, que están accesibles desde cualquier parte de la aplicación. Son archivos que tienen la extensión mdl. Para el presente proyecto se hizo uso de dos de estos archivos, uno para las variables globales y otro para las funciones globales. Claro está que para que se den globales tanto las funciones como las variables deben estar declarados como Public

PROGRAMACIÓN ORIENTADA A OBJETOS

Aunque Visual Basic 6.0 no es totalmente orientada a objetos si se hace uso de la mayoría de las características de este paradigma de la programación. Se dice que es un lenguaje orientado a eventos, porque gran parte de su funcionalidad está estrechamente ligada a ellos.

El SAI se desarrolló haciendo uso de los conceptos de la programación orientada a objetos, sobre todo en lo relacionado con la conexión con la base de datos. Por ejemplo

en el siguiente fragmento de código se declaran dos instancias uno de la clase Connection y la otra de la clase Recordset

```
Dim cnLogin As New Connection
Dim rsLogin As New Recordset
```

También se hace uso de clases definidas en tiempo de diseño y se declaran objetos como nuevas instancias de ellos, como el siguiente caso donde frmCliente es un formulario creado en tiempo de diseño y se crea una instancia en tiempo de ejecución.

```
Dim instCliente As New frmCliente
```

Por otro lado se destruyen instancias para que no permanezcan cargadas en memoria cuando se descarga el formulario, esto aplica para todos los formularios

```
Private Sub Form_Unload(Cancel As Integer)
    Set frmLogin = Nothing
End Sub
```

La información privada de los objetos se encapsula por medio de la palabra reservada Private, de hecho todos los procedimientos de forma predeterminada son privados al objeto donde se encuentran declaradas:

```
Private Sub cmdSearh_Click()
    Call llamaBusqueda
End Sub
```

La forma de acceder a los métodos y propiedades de los objetos se realiza por medio del operador punto ".", como en el siguiente ejemplo:

```
Public Sub cmdFind_Click()
    'Llena el cuadro de lista con los campos de la tabla:
    With frmBusqueda.cmbCampo
        .AddItem "Id"
        .AddItem "NombreEmpresa"
        .AddItem "NombreContacto"
        .AddItem "CargoContacto"
        .AddItem "Direccion"
        .AddItem "Telefono"
        .AddItem "Fax"
    End With
    frmBusqueda.Show vbModal
End Sub
```

En este ejemplo además se usa el operador "With" para evitar repetir el mismo objeto con todos sus atributos.

VISTAS Y PROCEDIMIENTOS ALMACENADOS

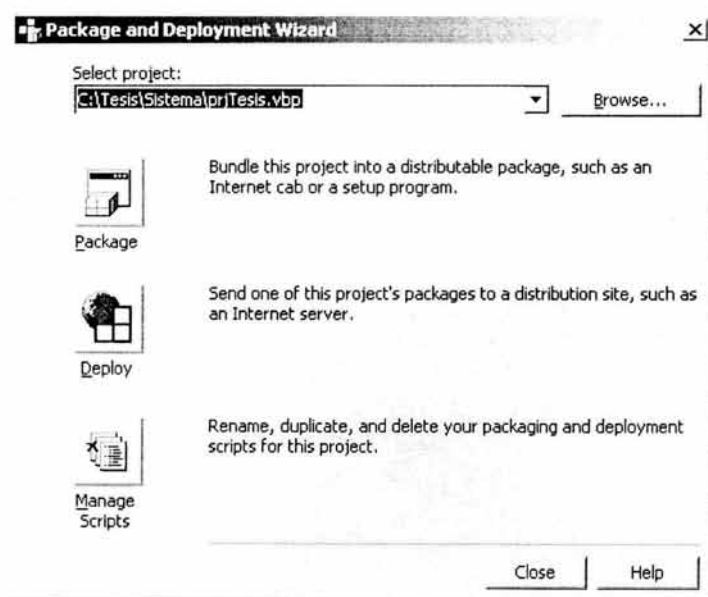
Las vistas que se utilizan en el sistema se crearon en el administrador corporativo de SQL Server. Las vistas no son más que consultas SQL almacenadas en el servidor para evitar repetir una misma consulta en varias partes, de esta forma se hace solo una consulta sencilla a la vista

Los procedimientos almacenados se utilizan principalmente para los reportes que maneja el SAI. Los procedimientos almacenados contienen también consultas SQL al igual que las vistas pero cuentan con la ventaja de que son funciones que reciben parámetros y pueden o no regresar algún valor. Se utilizan además para las operaciones de inserción, modificación y eliminación de datos en la base de datos.

Los procedimientos almacenados para el presente proyecto se crearon y son usados dentro del "DataEnvironment" que es un entorno de trabajo que nos permite definir los procedimientos dentro del mismo ambiente de desarrollo de Visual Basic.

PROGRAMA DE INSTALACIÓN

Dentro de la Suite de Visual Studio, existe una aplicación llamada Asistente para empaquetado y distribución. Este asistente permite crear el programa ejecutable y el programa de instalación del proyecto, para lograrlo basta con indicarle los archivos que se necesitan y algunas otras configuraciones básicas:



IV.3 PRUEBAS

PRUEBA DEL SISTEMA

La prueba de los programas es la parte del proceso de evaluación que suele realizarse durante la programación y también, en una forma distinta, cuando está terminado. La prueba consiste en ejecutar el programa simulando ser el usuario final para determinar si el sistema es amigable y de sencillo manejo y si es homogéneo en toda su presentación, entre otros aspectos.

Utilizando datos similares a los reales que habrán de ser suministrados al sistema, se observan los resultados y se deduce la existencia de errores o insuficiencias del programa a partir de las anomalías de estos resultados.

En ocasiones se piensa que la prueba y la depuración de programas son una misma cosa. Aunque están muy relacionados, en realidad son procesos distintos. La prueba es el proceso de establecer la existencia de errores en el programa; la depuración es el proceso de localizar donde se produjeron esos errores y corregir el código incorrecto, optimizándolo.

Es muy importante comprender que la prueba nunca demuestra que un programa es incorrecto. Siempre es posible que existan errores aún después de la prueba más completa. La prueba de programas sólo puede demostrar la presencia de errores, no su ausencia en el mismo.

PLAN DE PRUEBAS

El objetivo del plan de pruebas es detectar un comportamiento distinto al que necesita el usuario y de lo que se planeó en el análisis, diseño y en la programación.

Las pruebas que se aplican al sistema se basan principalmente en los documentos de análisis y diseño del sistema, así como el manual de usuario y técnico en donde se comprueba que es una exacta referencia de lo que se necesita para el manejo del sistema.

Pruebas especiales del sistema:

Pruebas de carga máxima (conurrencia)

Se basa en la existencia de tiempos críticos en los sistemas en línea, es decir, la respuesta de un sistema en prueba cuando varios usuarios quieren acceder a él. En el caso del SAI esta prueba se hizo con resultados satisfactorios.

Pruebas de almacenamiento

Mediante esta prueba se determina si el sistema realmente soporta la capacidad considerada en su diseño, la misma que debe ser considerada antes de la implantación. Para ello se va almacenando datos en forma continua hasta que se alcance la capacidad teórica. La capacidad real se obtiene al realizar la comprobación respectiva.

Prueba de recuperación

Ésta consiste en crear un evento de fallas o de pérdida de datos, para que los usuarios vuelvan a cargar y recuperar a partir de una copia de respaldo

Prueba de procedimientos

Con esta prueba se determina si los manuales de documentación y ejecución contiene una descripción detallada y si refleja realmente las acciones que se llevan a cabo para el funcionamiento del sistema. Para ello el usuario debe seguir las instrucciones en forma exacta, como se indica en el manual.

PUNTOS QUE SE REVISARON EN LAS PRUEBAS

Acceso al sistema

Por el tipo de datos con los que trabaja el sistema es importante conocer la seguridad para cada persona que lo utiliza (usuario y administrador), por lo tanto, los accesos, permisos, y operaciones que se realizan en el sistema se verificaron de la siguiente manera:

Acceso al sistema con diversos permisos en cada proceso (En cada pantalla se verificó los permisos de los usuarios por cada acción de altas, bajas, modificaciones, consultas, reportes, aceptar, cancelar, ayuda, entrar a las pantallas, etc.).

- Se entró al sistema desde menús secundarios, pantallas de reportes, búsquedas, catálogos, etc.
- Se verificó las restricciones de cada personal
- Se revisó las entradas y salidas del sistema con y sin detectar las acciones realizadas.
- Se verificó de las claves de usuario y administradores
- Se actualizaron de claves de usuario y administradores

Estructura

En esta parte se evaluó la organización, distribución y el orden que guarda todo el sistema

Los puntos revisados fueron:

- La uniformidad del diseño y nomenclatura que permita distinguir las pantallas de captura, catálogos, reportes, menús principales y secundarios, etc.
- Estética y homologación de títulos, tipos de letras, botones, listados, colores, búsquedas, presentación de las pantallas, iconos y botones en las pantallas de función similar
- Entradas y salidas de cada pantalla que presenten la pantalla que se requiera.
- Los mensajes de ayuda correspondan a cada caso que se aplique como son los mensajes de acción realizada, errores, datos a introducir o seleccionar, etc.
- Diseño de reportes (distribución de los datos) desde la pantalla hasta la impresión. Cada reporte tenga una justificación de su utilidad, así como las búsquedas.

Operación del sistema

En esta parte se midió que tan amigable es el sistema para el usuario y se verificó la correcta operación de los procesos.

La primera actividad que se realizó consistió en la secuencia y la navegación entre los registros y las funciones que realiza cada pantalla o el conjunto de controles que tuvieron un proceso, estos son:

- La introducción y selección de texto en cada campo correspondiente a cada tabla
- Secuencia de llenado en orden, por medio de tabuladores, ENTER u otra tecla, el llenado no puede ser aleatorio, debe seguir una secuencia.
- Evaluación de la opción correcta en que los datos son suministrados al proceso o función por medio de cajas de texto, botones de opción, casillas, y listas desplegables.
- La operación que se selecciona es la que se ejecuta (insertar, actualizar, aceptar, cancelar, etc.)
- Confirmación de la operación en los cuadros de diálogo.
- Resultados correctos que pueda generar en cada proceso o función
- Navegación entre los registros de la base de datos, por medio de botones como, ir al primero, al anterior, al siguiente y al último.
- Verificación de las búsquedas de información para que correspondan con lo que se ha solicitado.
- Utilización de las combinaciones de teclas para el uso que se les ha asignado
- Utilización de las combinaciones de teclas para conocer la reacción del sistema.

La segunda actividad es la navegación entre los menús y archivos auxiliares:

- Fácil de usar y amigable
- Verificación de las entradas y salidas (enlaces entre menús y pantallas), que funcionen adecuadamente
- Opciones de navegación en cada menú (entrar, salir y regresar)
- Revisión de las salidas que corresponden a cada parte del sistema
- Si un proceso ocupa más de una forma verificación del enlace entre ellas.

Validación de datos y funcionalidad

El objetivo de esta prueba fue asegurar que los filtros o validación que se requiera para cada dato que se introdujo o la relación entre datos fueran válidos.

En esta prueba se realizan las siguientes acciones:

- Revisar que no se aceptan datos erróneos (numéricos, alfabéticos, caracteres especiales, etc.)
- introducción de mayor y menor cantidad de caracteres, cantidades o rangos.
- Identificación de las fallas posibles al introducir errores intencionales.
- Identificación de cuales son los campos clave y verificación que no se dupliquen.
- Identificación de los campos que pueden quedar sin datos.
- Integridad del sistema después de realizar una operación.

Prueba general del sistema

Aquí se verifico la correcta comunicación entre todos los procesos que componen al sistema que las salidas de un módulo correspondan a las entradas de otro según la finalidad del sistema.

V IMPLEMENTACIÓN

V.1 ESPECIFICACIONES

REQUISITOS DE SOFTWARE PARA INSTALAR EL SAI 1.0

El SAI requiere de SQL Server 2000 Enterprise como servidor de base de datos y este a su vez del sistema operativo Windows 2000 Server para poder funcionar. Estas son algunas especificaciones para el SAI.

EQUIPOS CLIENTE

Hardware

- Unidad central de proceso (CPU) Pentium de 133 MHz o superior.
- Se recomienda 32 megabytes (MB) de RAM como mínimo. Una partición de disco duro con suficiente espacio libre para alojar el proceso de instalación. El espacio mínimo necesario será aproximadamente 50 MB.
- Tarjeta de Red o Fax Modem.

Software

- Sistema Operativo Windows 98 o superior (se recomienda Windows XP)
- Microsoft Office 97 o superior
- Internet Explorer 5.0 o superior

SERVIDOR

Hardware

- Unidad central de proceso (CPU) Pentium de 133 MHz o superior. Se recomienda 256 megabytes (MB) de RAM como mínimo (128 MB es el mínimo admitido; 8 gigabytes (GB) es el máximo).
- Una partición de disco duro con suficiente espacio libre para alojar el proceso de instalación. El espacio mínimo necesario será aproximadamente 1 GB. Es posible que necesite más espacio dependiendo de los factores siguientes:
 - El sistema de archivos que se utiliza: FAT requiere entre 100 y 200 MB más de espacio libre en disco que otros sistemas de archivos.
 - El método utilizado para la instalación: si va a instalar a través de una red, necesitará entre 100 y 200 MB más de espacio que si instala desde el disco compacto. (Es necesario tener disponibles más archivos de controlador durante la instalación a través de una red).
- Monitor VGA o de mayor resolución.
- Teclado.
- Mouse (ratón) u otro dispositivo señalador (opcional).
- Unidad de CD-ROM o DVD.

Software

- Microsoft Windows 2000
- Microsoft SQL Server 2000 Enterprise
- Microsoft Access 2000
- Microsoft Internet Explorer 5.0
- Service Pack 4 para Windows 2000
- Service Pack 3 para SQL Server 2000

REQUISITOS PARA INSTALAR SQL SERVER 2000

Los requisitos mínimos de hardware y software para ejecutar Microsoft SQL Server 2000 se muestran a continuación.

Requisitos de hardware

Esta tabla muestra los requisitos de hardware para instalar Microsoft SQL Server 2000 o las bibliotecas y herramientas de administración de cliente de SQL Server.

- Equipo: Intel o compatible, Pentium 166 MHz o superior.
- Memoria: 64 MB mínimo, se recomienda 128 MB o más.
- Espacio en el disco duro: Componentes de base de datos de SQL Server: De 95 a 270 MB, 250 MB típica.
- Pantalla: Para las herramientas gráficas de SQL Server se requiere una resolución de 800x600 o superior.
- Unidad de CD-ROM: Obligatorio.

Requisitos de software

Microsoft Windows NT, Windows 2000, Windows Me, Windows 98 y Windows 95 tienen software de red integrado.

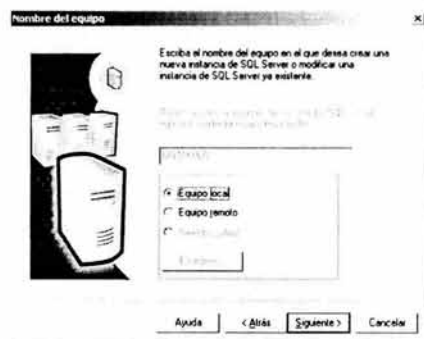
Solo es necesario habilitar TCP/IP al nivel del sistema operativo antes de instalar SQL Server 2000.

V.2 INSTALACIÓN

INSTALACIÓN DE SQL SERVER 2000

Estos son los pasos a seguir para la instalación de SQL Server 2000

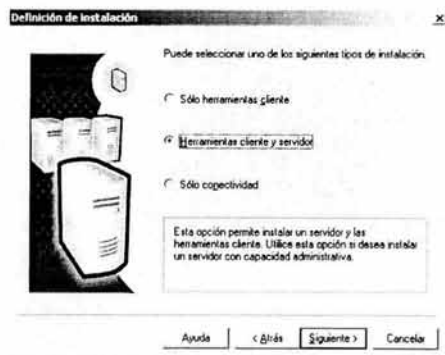
Seleccionar equipo local:



Crear una nueva instancia:



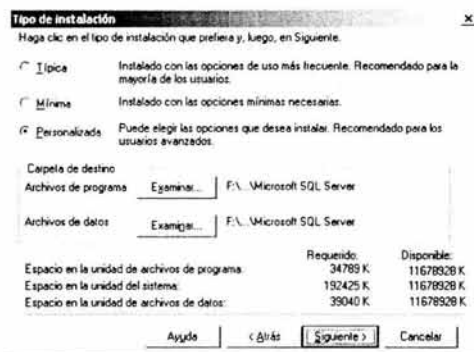
Seleccionar la opción “Herramientas de cliente y servidor”:



Escribir el nombre de la nueva instancia:



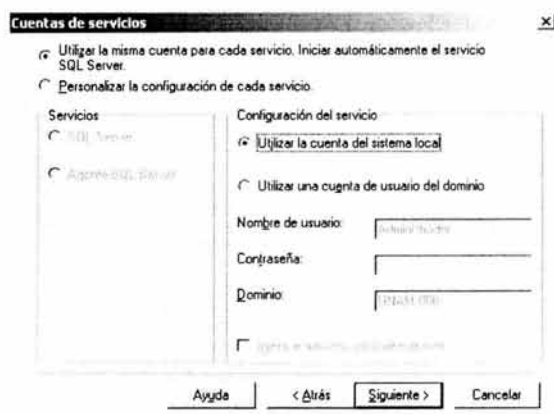
Seleccionar la opción para crear una instalación personalizada:



Seleccionar los componentes necesarios, si no esta seguro deje las opciones predeterminadas:



Seleccionar las opciones mostradas en la ilustración:



Elegir modo de autenticación mixto:



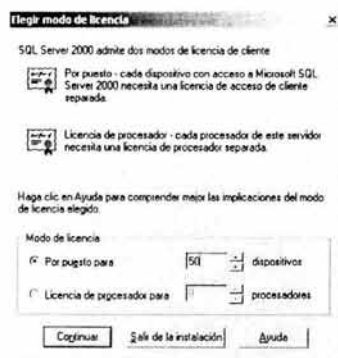
Seleccionar las opciones regionales, de preferencia en español:



Escribir el puerto 1433 para TCP/IP:



Asignar el número de dispositivos necesarios para el funcionamiento del sistema:



Comenzará la copia de archivos:



La instalación se completo satisfactoriamente:



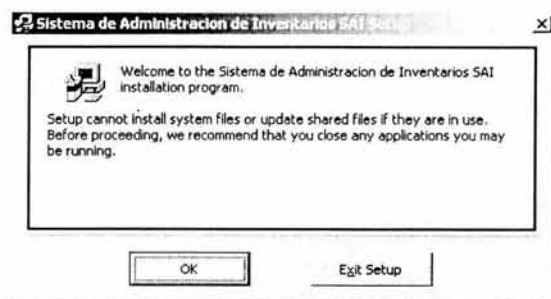
INSTALACIÓN DEL SAI 1.0

El CD en el que viene el SAI tiene “autorun”, esto es, inicia la instalación automáticamente cuando se inserta el disco compacto, sin embargo, algunas computadoras tienen deshabilitada esta opción, de ser así, de clic en el archivo de Instalación “Setup” que viene en la carpeta raíz del CD:

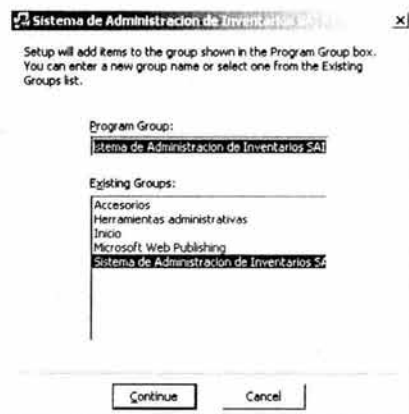
Aparecerá la siguiente ventana:



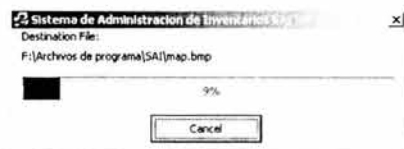
Dé clic en OK y posteriormente de clic en el icono de la computadora, se recomienda dejar la ruta predeterminada, sin embargo si desea cambiarla es irrelevante para el sistema



Después el programa de instalación pregunta la ubicación del acceso directo, de clic, en “Continue”



Aquí comenzará la copia de archivos:



Dé clic en finalizar cuando la instalación haya concluido:

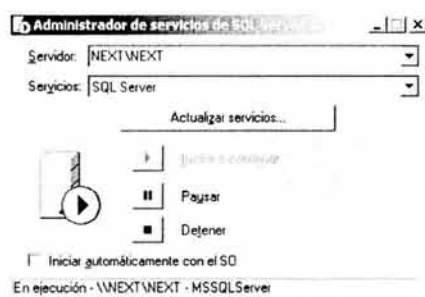


V.3 CONFIGURACIÓN

CONFIGURAR EL SERVIDOR DE BASE DE DATOS

Configurar el Servidor SQL

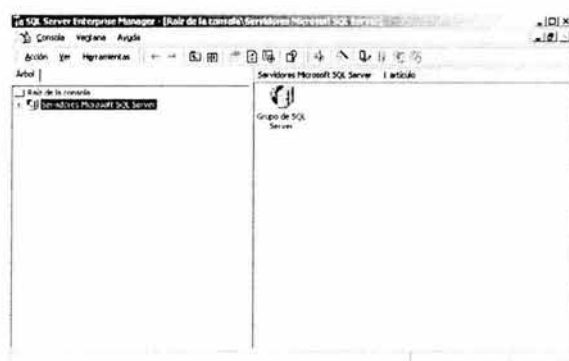
Para iniciar el servidor de base de datos Microsoft SQL Server, es necesario abrir el “Administrador de Servicios” que esta en el menú de Microsoft SQL Server, después (si no esta seleccionado de forma predeterminada seleccione el servidor instalado previamente y seleccione el servicio “SQL Server”). De clic en el botón “iniciar o continuar” para iniciar el servicio. Si lo desea puede activar la casilla “iniciar automáticamente con el SO”



Establecer la Seguridad en el Manejador de la Base de Datos

Es importante tener opciones de seguridad para el manejador de base de datos Microsoft SQL Server, para pedir la autenticación del administrador de sistema (sa), se requiere hace el siguiente proceso, en el servidor:

Abra el “Administrador Corporativo” que esta en el menú de Microsoft SQL Server:



Despliegue los servidores de Microsoft SQL Server dando clic en el signo “+”, y posteriormente abra el grupo de los servidores:



Una vez que aparezca el servidor activo, de clic con el botón en el servidor y seleccione “modificar propiedades del registro del servidor SQL Server”:



En la nueva ventana mostrada seleccione la opción “Utilizar autenticación SQL Server”, escriba la contraseña y deje habilitada la opción de “pedir siempre nombre y contraseña de inicio de sesión”:



CONFIGURAR EL CLIENTE

Indicar la ubicación del Servidor en el Cliente

El cliente necesita saber la localización del servidor SQL, este último puede estar en el mismo equipo, es decir ser un servidor local al cliente o ser un servidor remoto. Estas dos opciones se pueden configurar en el cliente.

Para configurar la localización del servidor se siguen los siguientes pasos:

Abra el SAI en el menú “Programas / Sistema de Administración de Inventarios (SAI)” del botón de inicio de Windows. Proporcione el nombre de usuario “sa” y la clave de acceso para el sistema local, esta clave la crea automáticamente el programa de instalación almacenando un valor en el registro de Windows del cliente, (si no la tiene pregunte al administrador). De clic en el botón “Servidor”, aparecerá la siguiente ventana:

Autenticación

Nombre de usuario: sa

Contraseña: *****

Dirección IP:

Local Servidor: next\nnext

Guardar Configuración

Crear Cuenta

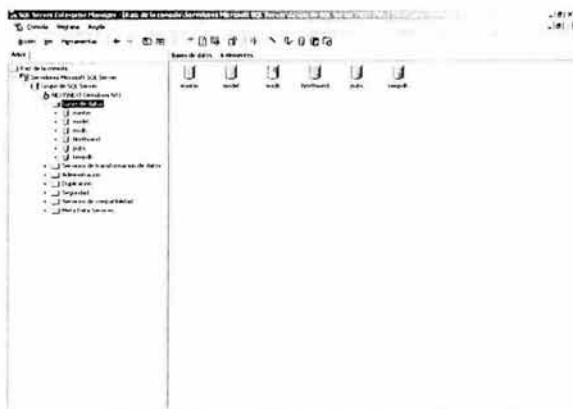
Aceptar Cancelar

Aquí podemos configurar la ubicación del servidor de SQL Server, que puede ser el nombre de la instancia local o una dirección IP del equipo en la red local. Si se activa la casilla “Guardar Configuración”, la próxima vez que entre se guardarán las preferencias seleccionadas en la sesión anterior. Es importante aclarar que la contraseña local para configurar estas opciones es diferente que la usada para establecer una conexión con SQL Server, este último requiere de un registro de la base de datos en SQL Server, mientras que la primera contraseña se encuentra almacenada en el registro de Windows del sistema local (regedit), por lo que al dar clic en “Aceptar”, una vez configuradas estas opciones es necesario escribir la contraseña de un usuario registrado en el SAI.

V.4 CARGA DE DATOS

CREACIÓN DE LA BASE DE DATOS PARA SAI

Si es la primera vez que instalamos SQL Server 2000 en el servidor se tiene que crear la base de datos tienda, para que aparezca en el catalogo de las bases de datos disponibles en el servidor.



Con el botón secundario en la carpeta “Bases de Datos” seleccionar “Nueva Base de Datos”, aparecerá la siguiente ventana:



En el cuadro de texto en donde nos pide el nombre de la base de datos, escribimos el nombre de nuestra base de datos que es “Tienda”

Si por alguna razón, ya existiera la base de datos tienda pero no aparece en la lista de las bases de datos, marcará el siguiente mensaje de error

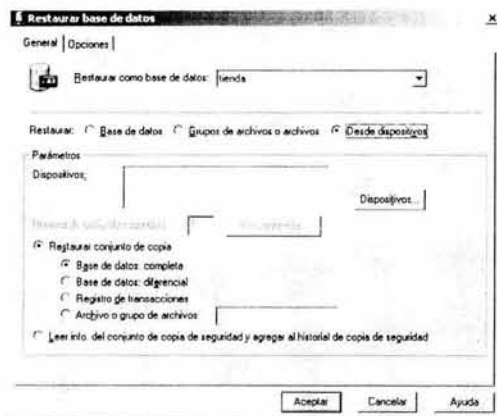


Esto se debe principalmente a que se hizo una instalación previa y no se borro por completo la información relaciona con esta base de datos, en este caso, basta con eliminar del sistema los archivos tienda_log y tienda_data, que generalmente se encuentran en \MSSQL\$(NombreServidor)\Data\” dentro de Microsoft SQL Server dentro del directorio de Archivos de Programa, de no encontrarse aquí puede buscarse en el sistema para eliminarlos, de esta forma se podrá crear la nueva base de datos.

RECUPERACIÓN DE LA ESTRUCTURA DE LA BASE DE DATOS

Una vez creada nuestra base de datos se hace la recuperación del respaldo de la siguiente manera.

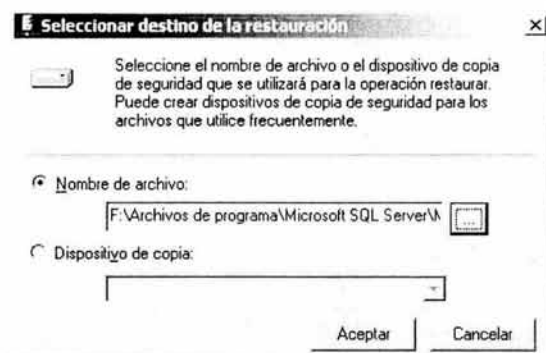
1. Clic derecho en la base de datos “Tienda” que acabamos de crear, seleccionar “Todas las tareas”
2. Seleccionar “Restaurar base de datos...”
3. Si la base de datos se encuentra en el disco duro seleccionar las siguientes opciones:



Dar Clic en “Dispositivos...”, aparecerá la siguiente ventana:



Dar clic en “Agregar...”, y seleccionar el archivo de respaldo de la base de datos con su ruta completa, este archivo se encuentra en la carpeta “support” con el nombre de “resTienda”, en el CD de instalación del SAI.



Seleccionar la ruta:



Dar clic en aceptar en las ventanas abiertas, aparecerá la ventana que confirma el respaldo exitoso:



Ahora podemos ver la información completa de nuestra base de datos restaurada:

Nombre	Propiedades	Tip	Fecha de creación
Adventureworks2008	db	Usuario	1/10/2008 10:28:28
Adventureworks2008_log	db	Usuario	1/10/2008 10:31:05
Adventureworks2008_1	db	Usuario	1/10/2008 10:30:36
Adventureworks2008_2	db	Usuario	1/10/2008 10:31:10
Adventureworks2008_3	db	Usuario	1/10/2008 10:32:14
Adventureworks2008_4	db	Usuario	1/10/2008 10:30:06
Adventureworks2008_5	db	Usuario	1/10/2008 10:30:02
Adventureworks2008_6	db	Usuario	1/10/2008 10:31:17
Adventureworks2008_7	db	Usuario	1/10/2008 10:31:07
Adventureworks2008_8	db	Usuario	1/10/2008 10:30:37
Adventureworks2008_9	db	Usuario	1/10/2008 10:30:32
Adventureworks2008_10	db	Usuario	1/10/2008 10:32:05
Adventureworks2008_11	db	Usuario	1/10/2008 10:30:05
Adventureworks2008_12	db	Usuario	1/10/2008 11:07:00
Adventureworks2008_13	db	Usuario	1/10/2008 10:32:25
Adventureworks2008_14	db	Usuario	1/10/2008 10:30:52
Adventureworks2008_15	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_16	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_17	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_18	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_19	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_20	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_21	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_22	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_23	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_24	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_25	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_26	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_27	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_28	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_29	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_30	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_31	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_32	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_33	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_34	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_35	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_36	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_37	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_38	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_39	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_40	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_41	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_42	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_43	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_44	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_45	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_46	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_47	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_48	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_49	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_50	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_51	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_52	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_53	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_54	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_55	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_56	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_57	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_58	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_59	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_60	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_61	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_62	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_63	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_64	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_65	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_66	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_67	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_68	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_69	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_70	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_71	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_72	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_73	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_74	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_75	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_76	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_77	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_78	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_79	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_80	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_81	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_82	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_83	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_84	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_85	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_86	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_87	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_88	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_89	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_90	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_91	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_92	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_93	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_94	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_95	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_96	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_97	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_98	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_99	db	Usuario	1/10/2008 10:31:12
Adventureworks2008_100	db	Usuario	1/10/2008 10:31:12

Para el llenado de los catálogos es recomendable hacerlo directamente desde la aplicación SAI debido a que en ella se encuentran todas las validaciones correspondientes a las reglas del negocio, sin embargo esto no impide que en casos excepcionales la captura se haga directamente desde el “Administrador Corporativo” de SQL Server 2000.

V.5 DEPURACIÓN DE ERRORES

TIPOS DE ERRORES

Básicamente existen dos tipos de errores en tiempo de ejecución: los errores de operaciones inválidas y los errores lógicos.

Los errores de operaciones inválidas son aquellos en donde una instrucción intenta una operación que es imposible de realizar. Un ejemplo de esto es una división entre cero.

Los errores lógicos se producen cuando una aplicación no actúa de la forma que se pretendía. Una aplicación puede tener código sintácticamente válido, ejecutarse sin llevar a cabo ninguna operación que no sea válida y, aún así, producir resultados incorrectos. Sólo si se prueba la aplicación y analiza los resultados se puede comprobar que la aplicación está actuando correctamente.

MANEJO DE ERRORES

A pesar de que en las fases de programación y pruebas se revisa el sistema para encontrar posibles fallas y evitar cualquier tipo de error en la aplicación, no se descarta que un vez puesto en marcha el sistema surjan errores que posiblemente no fueron contemplados mientras éste se desarrolló. Aun así, para el caso de presentarse errores de operaciones inválidas, se programó código que facilita la rápida detección de éstos en tiempo de ejecución. La función que se programó tiene la ventaja de especificar de que modulo proviene el error, de que procedimiento o función es, en que evento se generó, el número de error y su descripción.

Tanto para el cliente, como para el desarrollador resulta una manera cómoda de corregir las fallas una vez que se sabe exactamente donde ocurrieron. Los errores típicos en un sistema cliente-servidor suelen estar asociados con una conexión fallida a la base de datos o con problemas en la red. El manejo de errores en el SAI tiene además el propósito de mantener el sistema funcionando aún y cuando el error podría cerrar la aplicación.

RUTINA DE ERRORES

Con las siguientes instrucciones de Visual Basic se hace el manejo de errores en el sistema SAI:

- On Error GoTo fin,
- On Error Resume Next

On Error GoTo

Si ocurre algún error en el código contenido entre las palabras clave

On Error GoTo <Etiqueta> y <Etiqueta:>

El error es tratado al final de "<Etiqueta>" y se toma la acción necesaria.

Ejemplo

En el siguiente código, la etiqueta que marca la captura de un error es "fin". Si en las instrucciones ocurrió un error se llama a la función "MostrarError" que muestra la descripción del error y el lugar del programa donde se generó

```
On Error GoTo fin
```

```
.
```

```
.
```

```
.
```

```
Instrucciones
```

```
.
```

```
.
```

```
.
```

```
Exit Function/Procedure
```

```
fin:
```

```
Call MostrarError("validarUpdProveedor", "frmArticulo", "Error con el origen de  
datos")
```

```
*****
```

```
'Procedimiento que genera el mensaje de error que atrapa la aplicación
```

```
'Creada: 23-01-2003
```

```
*****
```

```
Public Sub MostrarError(strProcFunc As String, strForm As String, Optional  
strMensaje As String)
```

```
    If strMensaje = "" Then
```

```
        MsgBox "Datos del error: " & Chr(13) & Chr(13) & "Módulo: " & strForm &  
Chr(13) & "Procedimiento/Función: " & strProcFunc & Chr(13) _  
        & "Descripción del Error: " & Err.Description _  
        & Chr(13) & "Número de Error: " & Err.Number, vbExclamation, "Manejo de  
Errores"
```

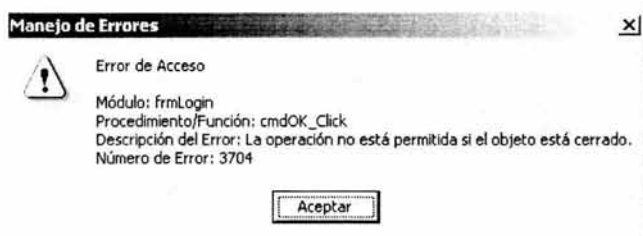
```
    Else
```

```
        MsgBox strMensaje & Chr(13) & Chr(13) & "Módulo: " & strForm & Chr(13) &  
"Procedimiento/Función: " & strProcFunc & Chr(13) _  
        & "Descripción del Error: " & Err.Description _  
        & Chr(13) & "Número de Error: " & Err.Number, vbExclamation, "Manejo de  
Errores"
```

```
    End If
```

```
End Sub
```

Esta es la descripción del error que ve el usuario:



On Error Resume Next

Especifica que, en caso de que se produzca un error en tiempo de ejecución, el control pase a la instrucción que sigue inmediatamente a aquella en la que se ha producido el error donde continúa la ejecución.

ERRORES LÓGICOS

Los errores lógicos son más difíciles de detectar, y solo mediante el uso del sistema se puede comprobar si éstos no existen. La fase de pruebas es fundamental para detectarlos y corregirlos si es que existen para dejar listo al sistema sin errores de este tipo. Una buena programación apegada a las necesidades del cliente y a la documentación de los requerimientos es la mejor solución para evitarlos.

V.6 PUESTA A PUNTO

INFRAESTRUCTURA

Para poder hacer uso del sistema se tiene que verificar la infraestructura de red, revisar el cableado de computadoras y verificar que este correctamente configurado el protocolo TCP/IP en cada una de las terminales que utilizaran el sistema. El equipo más importante, sin duda, es el servidor. De este equipo es necesario comprobar su disponibilidad en todo momento que sea requerido, así como establecer políticas para su mantenimiento continuo, principalmente en lo referente al manejo de la información como los respaldos. Los equipos deben estar en buenas condiciones, y funcionar en un ambiente adecuado (sin exceso de calor, limpio, etc.), para evitar contratiempos en la puesta en marcha del SAI.

CONFIGURACIONES MENORES

El servidor debe soportar los usuarios concurrentes para los que fue diseñado el sistema, esto se configura directamente en el servidor. Aunque SQL Server administra de forma automática el crecimiento de la base de datos, una configuración manual en el Administrador Corporativo puede incrementar el rendimiento del servidor.

Por otro lado, aunque con el programa de instalación se instalan la mayoría de las opciones necesarias para que el sistema funcione de manera adecuada dentro de una red local, existen ciertas configuraciones que ajustar dentro de los equipos cliente, por ejemplo, la ruta de la máquina local en donde se almacenaran las imágenes de las categorías, los artículos y los empleados. Se recomienda que esta ruta sea la misma para todas las máquinas cliente y, por supuesto mantener la carpeta principal de imágenes en el servidor.

ASIGNACIÓN DE ROLES

Cada uno de los usuarios tendrá un rol específico dentro del sistema: administradores, vendedores y capturistas, por lo que es importante establecer estos roles previo a la utilización del sistema. Estos roles los crea aquella persona que tenga permisos de administración dentro del SAI.

CAPACITACIÓN Y SOPORTE TÉCNICO

Si dentro de la organización que utilizará el sistema no existe alguna persona con los conocimientos adecuados para dar mantenimiento al sistema, es necesario llevar a cabo cursos de capacitación de varias sesiones semanales para que en el futuro sea personal del cliente quien administre el servidor.

Por otro lado los cursos para utilizar el sistema a nivel usuario una vez que éste sea implementado. En estas clases se explica el comportamiento básico del sistema, y se explican las ventajas de utilizarlo.

El soporte técnico se proporciona en todo momento que el sistema lo requiera por el periodo de tiempo acordado previamente.

MEJORAS FUTURAS

Es importante mantener una constante comunicación con el cliente para conocer sus impresiones sobre el sistema y reunir aquellos aspectos que le gustaría mejorar para incluirlos en las versiones futuras, de tal forma que la entrega del producto no cierre definitivamente nuestras opciones de venta de nuevos productos.

Una de las mejoras, a mediano plazo, que se tienen contempladas es hacer funcionar al SAI con un lector de código de barras para agilizar la captura de los artículos.

Otra mejora interna será migrar el SAI a versiones futuras de Visual Basic y SQL Server, para que de esta forma el código se mantenga vigente.

Por otro lado, como se mencionó en los capítulos anteriores, se pretende que el SAI sea una plantilla para el desarrollo de programas posteriores de otra índole, como contabilidad y nómina.

CONCLUSIONES

Actualmente, una organización competitiva requiere de un adecuado manejo de su información para la toma de decisiones que le permita continuar a flote y prosperar. En los establecimientos comerciales la información cambia día a día y por esta razón es importante llevar un control centralizado de los datos que fluyen en todas sus transacciones. En el presente trabajo de tesis se abordó esta problemática planteando como solución la organización de los datos de Electromateriales Alfa por medio del Sistema de Administración y Control de Inventarios (SAI) que tiene como tarea principal automatizar el manejo de los inventarios a través del registro de sus operaciones comerciales como lo son las compras y las ventas principalmente.

La creación de este sistema informático no fue una tarea fácil, ya que requirió de la coordinación de múltiples personas que intervinieron en su elaboración: desde aquellos que dieron los requerimientos hasta los usuarios. La necesidad de investigar las condiciones actuales de la empresa y la búsqueda de la solución correcta, de sus objetivos y sus metas fueron parte de la primera etapa de este proceso de desarrollo.

Por otro lado, la adecuada planeación y análisis del sistema fue la base del éxito de este sistema de software. De ahí la necesidad de utilizar una metodología adecuada. La metodología fue fundamental, ya que esta proporcionó la guía y dirección para el desarrollo del sistema.

Una de las enseñanzas del presente trabajo fue que cubrir con las necesidades del cliente es el principal objetivo de un sistema de calidad, considero que se cubrió con este objetivo gracias al conocimiento de la problemática del cliente. Estas necesidades iban desde crear un sistema que tuviera facilidad de uso, una interfaz agradable y que no requiriera de conocimientos avanzados de computación, hasta lograr un sistema que alcanzara un alto desempeño y que cumpliera con determinadas políticas para garantizar la integridad, disponibilidad y confidencialidad de la información.

Otro aspecto importante a considerar es que la estandarización de la programación y homogenización general de este sistema de software fue vital para mantener el código actualizado. Un sistema es valioso cuando se adapta a los cambios de la organización para los que fue creado. Considero que la estandarización del código es una de las bases para lograr esta adaptación sin mayores esfuerzos.

El presente trabajo es un caso de software hecho a la medida del cliente, sin embargo tiene las bases para poder considerarlo como una plantilla para crear aplicaciones futuras.

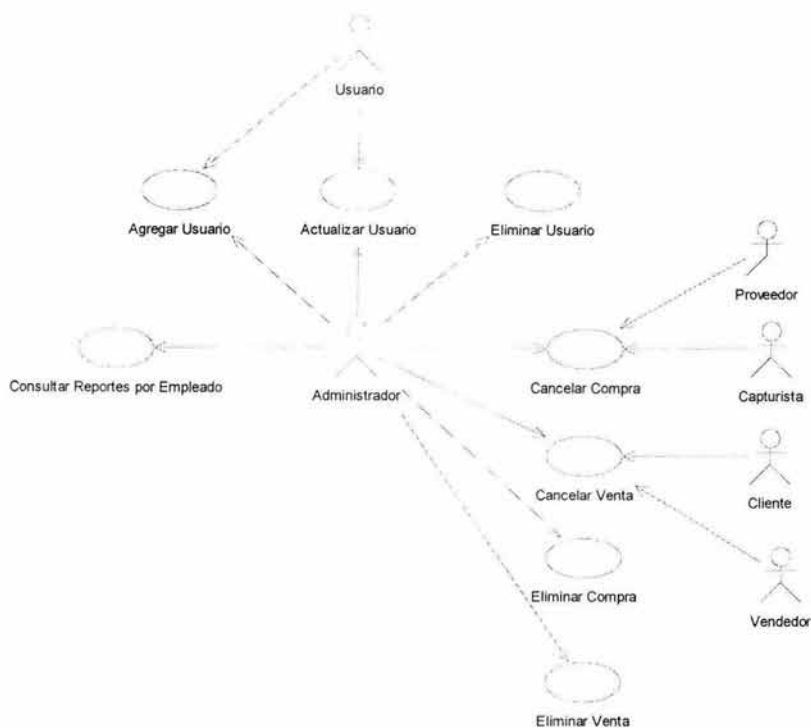
Un detalle relevante a tener en cuenta es que los sistemas informáticos por si solos no son capaces de tomar decisiones, por lo que es indispensable alimentarlos con los datos correctos, de esta forma se conseguirían los resultados esperados.

El SAI es una herramienta de cómputo que utilizado adecuadamente permite que la información de los inventarios este al alcance en todo momento, que sea expedita, rápida, específica, oportuna y sobre todo confiable. De esta forma se logra que la organización que lo maneje alcance un alto grado de competitividad en el mercado.

APÉNDICE A

DIAGRAMAS DE CASOS DE USO DEL SAI

CASOS DE USO DE ADMINISTRADOR

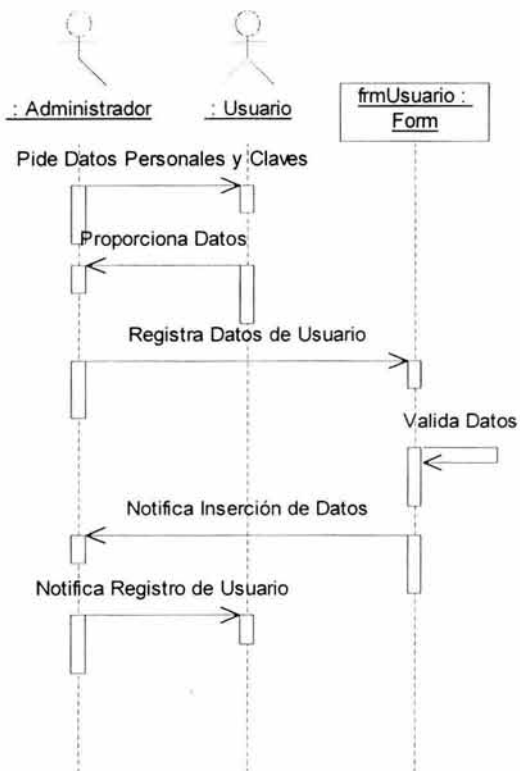


BREVE DESCRIPCIÓN

El administrador es el único encargado de la gestión de los usuarios, es decir puede agregar, actualizar, eliminar y consultar las cuentas de usuario y es el único actor del sistema que puede cancelar y eliminar las compras y las ventas. Para realizar los anteriores procesos se auxilia de actores como usuario, proveedor, vendedor y cliente que participan de forma indirecta proporcionando algunos datos.

DIAGRAMAS DE SECUENCIA PARA CASOS DE USO DE ADMINISTRADOR

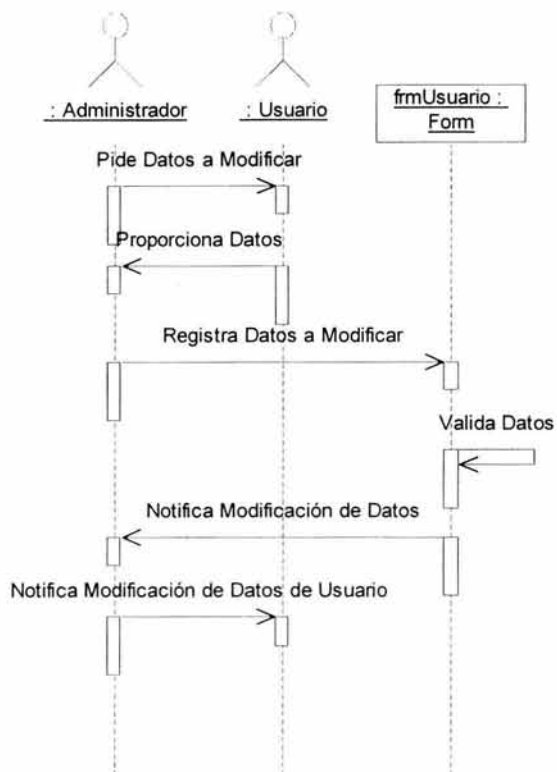
AGREGAR USUARIO



DESCRIPCIÓN

1. El administrador pide sus datos al nuevo usuario y sus clave
2. El nuevo usuario proporcionan sus datos y su nombre de usuario y contraseña
3. El administrador registra en el sistema los datos del usuario y asignan un perfil ya sea administrador, vendedor o capturista.
4. El sistema en la Forma de Usuario valida los datos del usuario, como lo son la contraseña y nombre de usuario.
5. El sistema notifica el registro exitoso del usuario.
6. El administrador notifica la creación de la cuenta al nuevo usuario y le confirma sus claves de acceso.

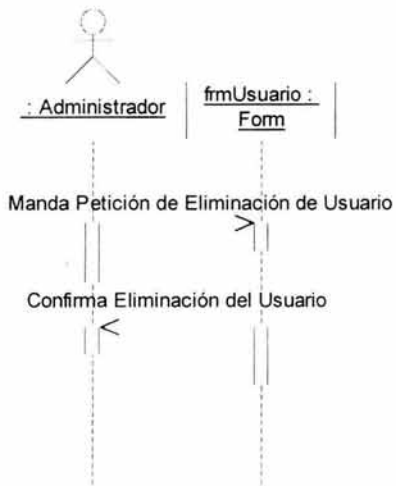
ACTUALIZAR USUARIO



DESCRIPCIÓN

1. El administrador pide los datos del usuario a modificar
2. El usuario proporciona los datos actuales.
3. El administrador registra los datos en la forma de Usuario del Sistema
4. El sistema valida los nuevos datos a través de la forma de Usuario
5. El sistema notifica el registro exitoso del usuario.
6. El administrador notifica la actualización de la cuenta al usuario existente.

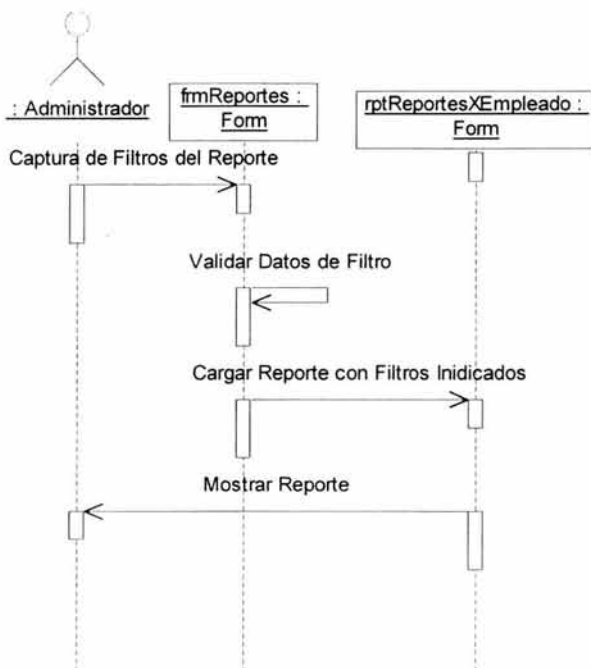
ELIMINAR USUARIO



DESCRIPCIÓN

1. El administrador selecciona el registro del usuario a eliminar y manda la petición al sistema, en la forma de Usuario
2. El sistema a través de la forma de Usuario confirma la eliminación de la cuenta de usuario.

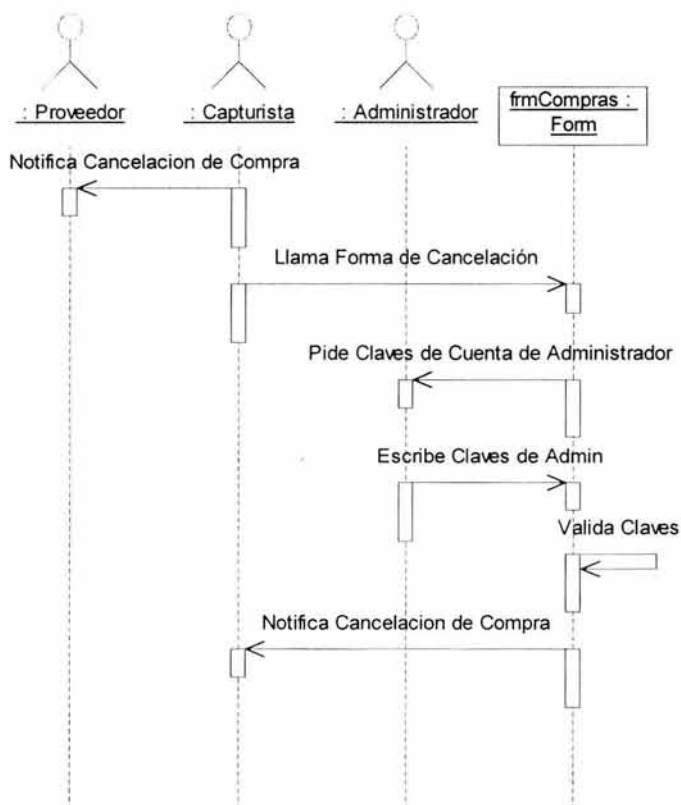
CONSULTAR REPORTES POR EMPLEADO



DESCRIPCIÓN

1. El administrador captura los criterios de búsqueda para obtener los datos del empleado o empleados que requiere.
2. El formulario de reportes del sistema valida los datos introducidos por el administrador
3. Una vez validados los datos la forma de reportes carga el reporte de empleado con los criterios seleccionados previamente.
4. El reporte es mostrado al administrador.

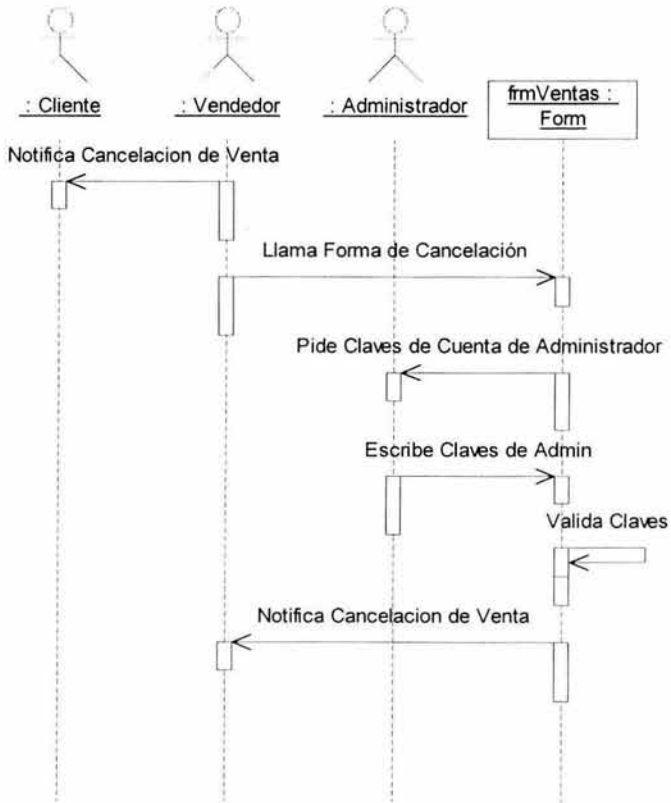
CANCELAR COMPRA



DESCRIPCIÓN.

1. El capturista notifica al proveedor la cancelación de una compra
2. El capturista manda llamar forma de cancelación de compras.
3. El sistema a través de la forma de compra pide claves de administrador para poder cancelar la compra.
4. El administrador teclea sus claves.
5. El sistema valida las claves en la forma de compras.
6. El sistema notifica la cancelación de la compra.

CANCELAR VENTA



DESCRIPCIÓN

1. El vendedor notifica al cliente la cancelación de la venta
2. El vendedor carga la forma de venta para cancelar la venta.
3. El sistema a través de la forma de venta pide, para cancelar la venta, claves de administrador.
4. El administrador escribe las claves para proceder con la cancelación.
5. El sistema valida las claves.
6. El sistema en la forma de Venta notifica al usuario la cancelación de la venta.

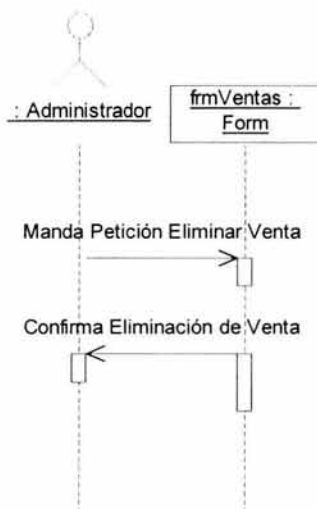
ELIMINAR COMPRA



DESCRIPCIÓN

1. El administrador selecciona el registro la compra a eliminar y manda la petición al sistema, en la forma de Compras
2. El sistema a través de la forma de Compras confirma la eliminación de la compra.

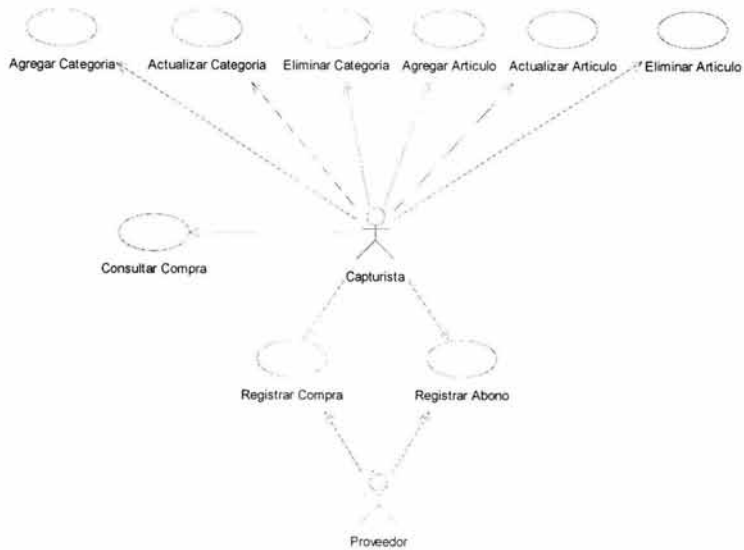
ELIMINAR VENTA



DESCRIPCIÓN

1. El administrador selecciona el registro la compra a eliminar y manda la petición al sistema, en la forma de Ventas
2. El sistema a través de la forma de Ventas confirma la eliminación de la venta.

CASOS DE USO DE CAPTURISTA

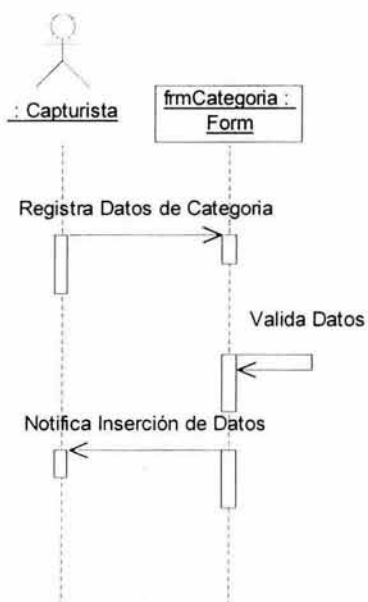


BREVE DESCRIPCIÓN

El capturista es el único encargado de la gestión de los artículos, es decir puede agregar, actualizar, eliminar y consultar los artículos y es el único actor del sistema que puede registrar las compras y los abonos. Para realizar los anteriores procesos se auxilia de los proveedor que participa de forma indirecta proporcionando algunos datos.

DIAGRAMAS DE SECUENCIA PARA CASOS DE USO DE CAPTURISTA

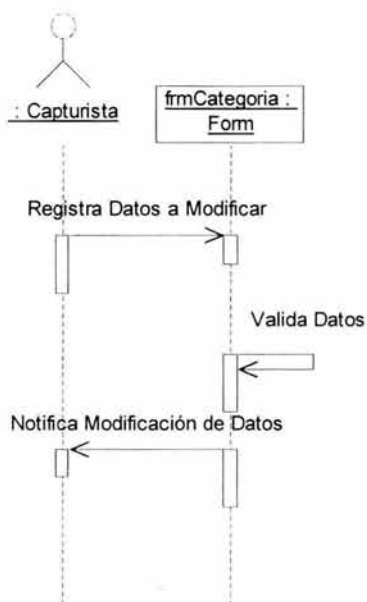
AGREGAR CATEGORÍA



DESCRIPCIÓN

1. El capturista registra los datos de la nueva categoría en la forma de categoría del sistema.
2. El sistema valida los datos en el formulario de categoría
3. El sistema confirma la inserción exitosa del registro.

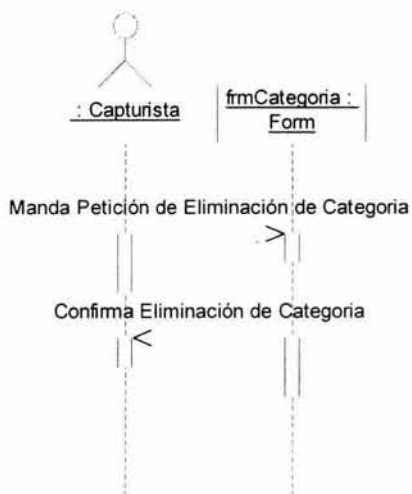
ACTUALIZAR CATEGORÍA



DESCRIPCIÓN

1. El capturista registra los datos a modificar de la categoría en la forma de categoría del sistema.
2. El sistema valida los datos en el formulario de categoría
3. El sistema confirma la modificación del registro.

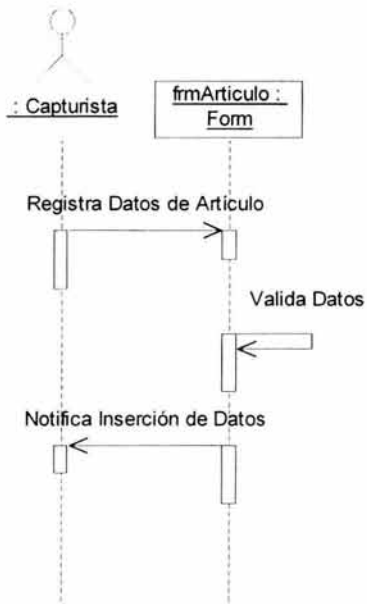
ELIMINAR CATEGORÍA



DESCRIPCIÓN

1. El capturista selecciona el registro de la categoría a eliminar y manda la petición al sistema, en la forma de Categoría.
2. El sistema a través de la forma de Categoría confirma la eliminación del registro.

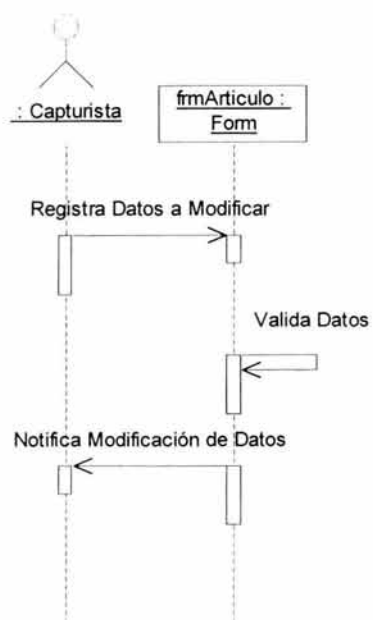
AGREGAR ARTÍCULO



DESCRIPCIÓN

1. El capturista registra los datos del nuevo artículo en la forma de artículo del sistema.
2. El sistema valida los datos en el formulario de usuario.
3. El sistema confirma la inserción exitosa del registro.

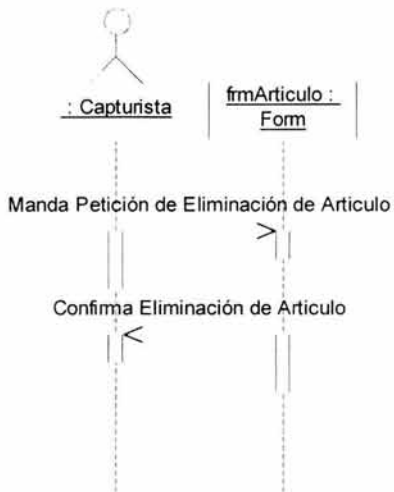
ACTUALIZAR ARTÍCULO



DESCRIPCIÓN

1. El capturista registra los datos a modificar del artículo en la forma de artículo del sistema.
2. El sistema valida los datos en el formulario de artículo
3. El sistema confirma la modificación del registro.

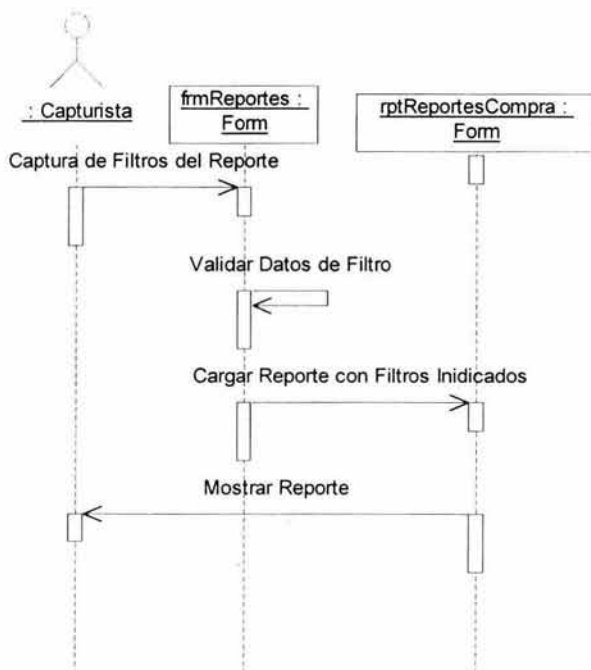
ELIMINAR ARTÍCULO



DESCRIPCIÓN

1. El capturista selecciona el registro del artículo a eliminar y manda la petición al sistema, en la forma de Artículo.
2. El sistema a través de la forma de Artículo confirma la eliminación del registro.

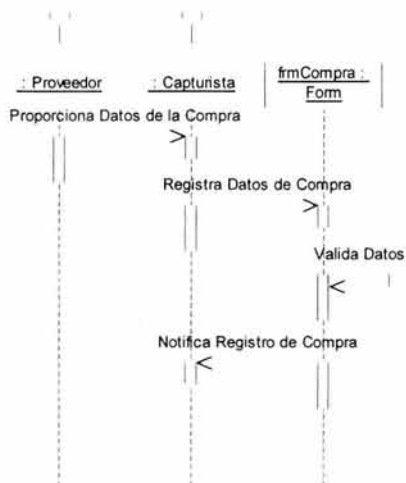
CONSULTAR COMPRA



DESCRIPCIÓN

1. El capturista registra los criterios de búsqueda para obtener los datos de la compra o compras que requiere.
2. El formulario de reportes del sistema valida los datos introducidos por el capturista.
3. Una vez validados los datos la forma de reportes carga el reporte de la compra con los criterios seleccionados previamente.
4. El reporte es mostrado al capturista.

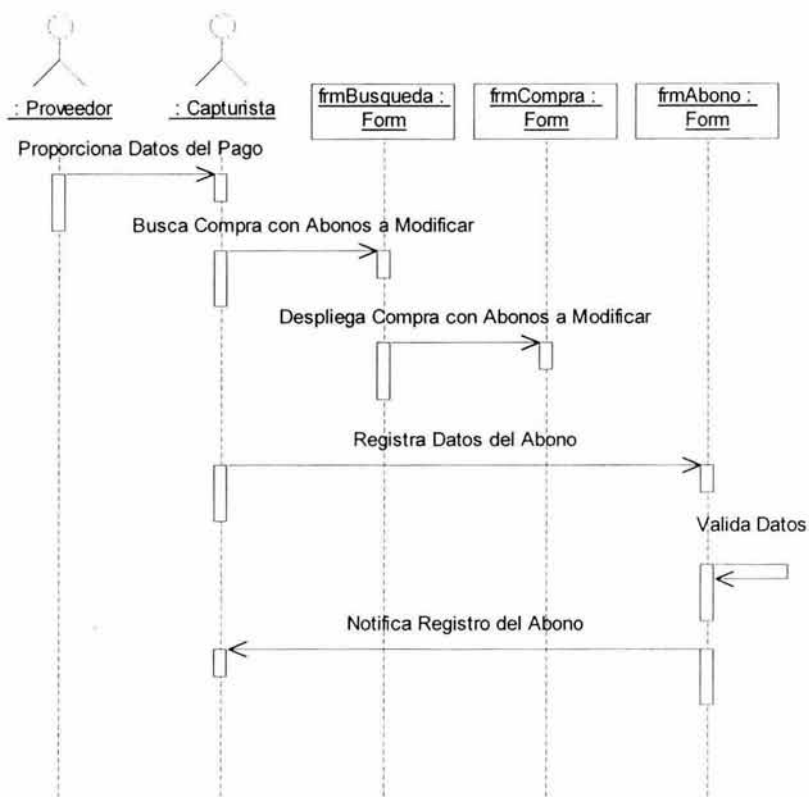
REGISTRAR COMPRA



DESCRIPCIÓN

1. El proveedor proporciona los datos de la compra al capturista
2. El capturista registra los datos de la compra en el formulario de compra del sistema.
3. El formulario de compra valida los datos introducidos.
4. El formulario de compra notifica al usuario el registro exitoso de la compra.

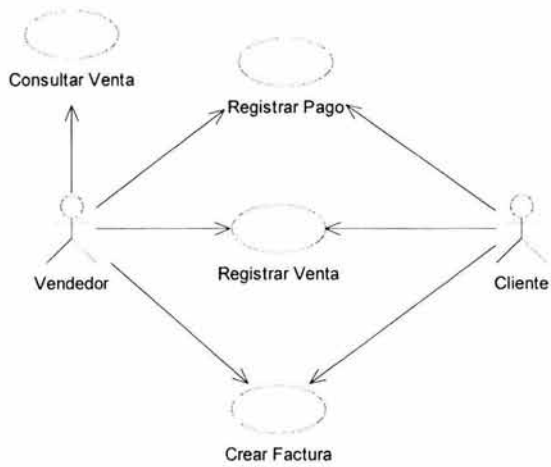
REGISTRAR ABONO



DESCRIPCIÓN

1. El proveedor proporciona los datos de pago que realizó la empresa.
2. El capturista busca el registro de la compra a la cual se le agregarán abonos en el formulario de búsqueda.
3. El formulario de búsqueda muestra en el formulario de compra la compra a la cual se le agregarán abonos.
4. Una vez encontrada la compra el capturista registra en el formulario de abono la cantidad pagada al proveedor.
5. El formulario de abono valida los datos introducidos.
6. El formulario de abono notifica el registro exitoso del abono.

CASOS DE USO DE VENDEDOR

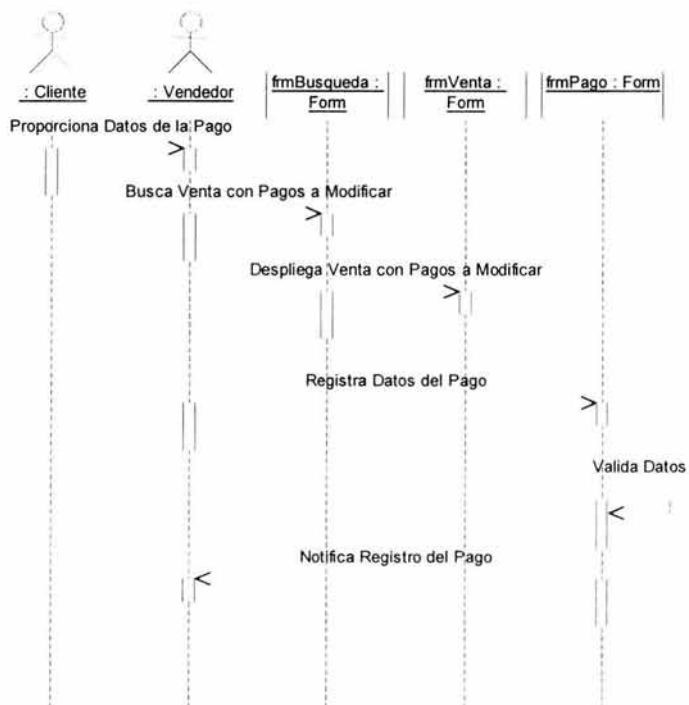


BREVE DESCRIPCIÓN

El vendedor es el encargado especial para registrar ventas, pagos, crear facturas y consultar ventas. Para realizar los anteriores procesos se auxilia de los clientes que participa de forma indirecta proporcionando algunos datos.

DIAGRAMAS DE SECUENCIA PARA CASOS DE USO DE VENDEDOR

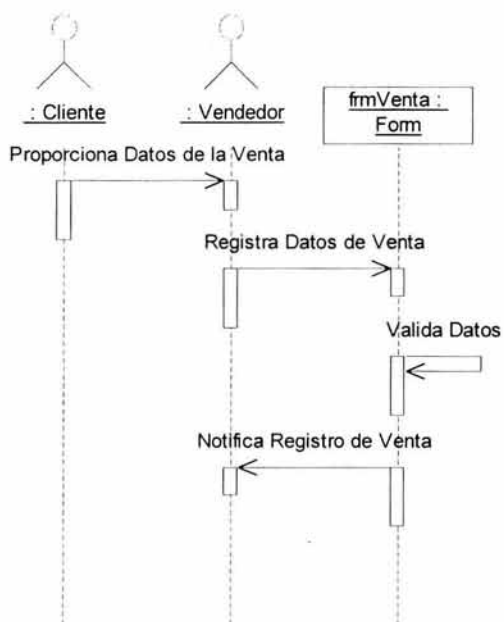
REGISTRAR PAGO



DESCRIPCIÓN

1. El cliente proporciona los datos de pago que realizó.
2. El vendedor busca el registro de la venta a la cual se le agregarán pagos en el formulario de búsqueda.
3. El formulario de búsqueda muestra en el formulario de venta la venta a la cual se le agregaran pagos.
4. Una vez encontrada la venta el vendedor registra en el formulario de pago la cantidad pagada del cliente.
5. El formulario de pago valida los datos introducidos.
6. El formulario de pago notifica el registro exitoso del pago.

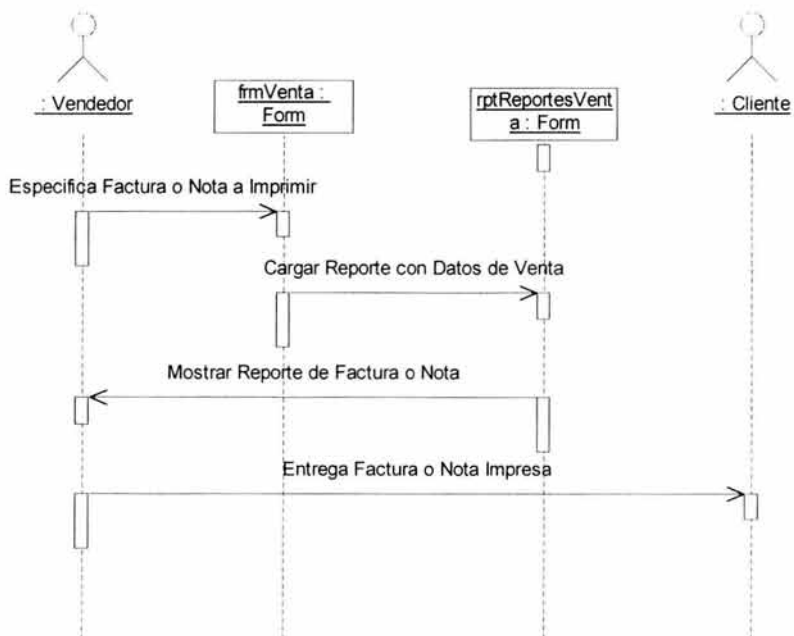
REGISTRAR VENTA



DESCRIPCIÓN

1. El cliente proporciona los datos de los artículos adquiridos
2. El vendedor registra los datos de la venta en el formulario de venta
3. El formulario de venta valida los datos
4. El formulario de venta notifica la inserción exitosa del registro.

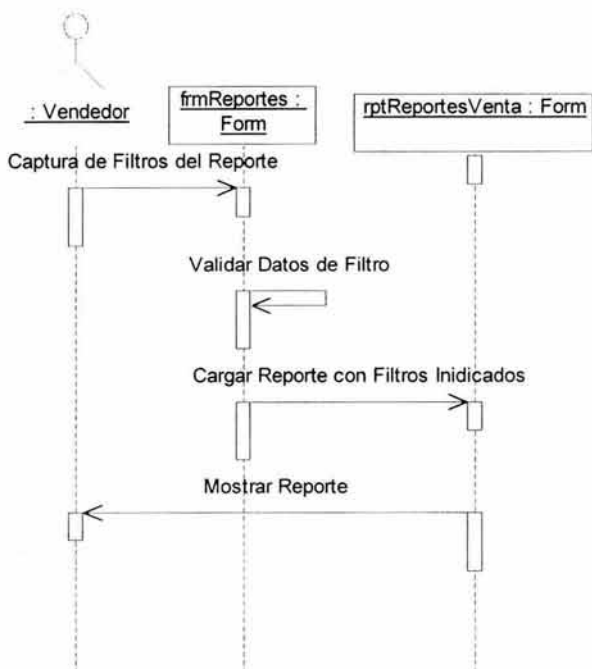
CREAR FACTURA



DESCRIPCIÓN

1. El vendedor especifica la factura o nota a imprimir
2. La forma de venta carga el reporte con los datos de la venta
3. Se muestra el reporte al vendedor desde el control de reportes de venta
4. El vendedor entrega la impresión de la factura al cliente.

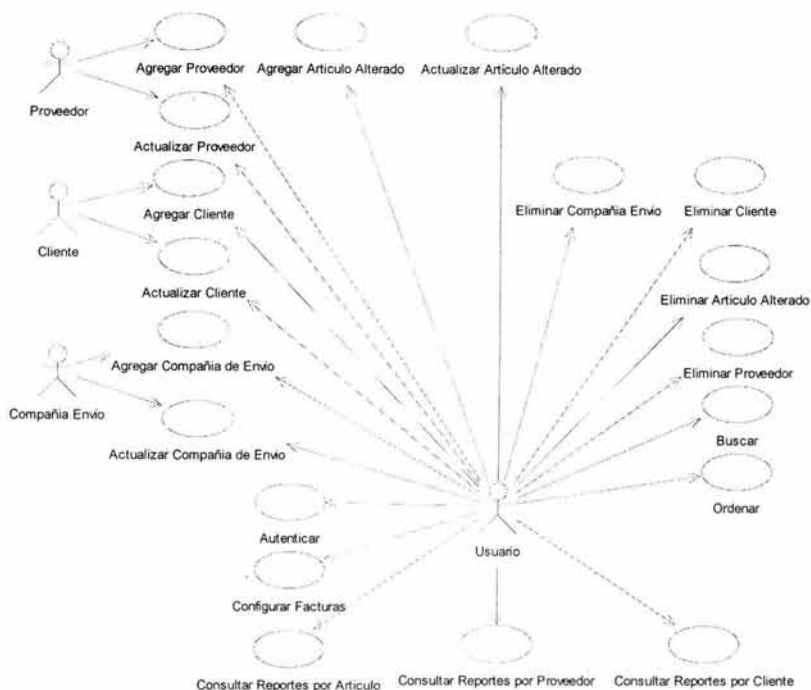
CONSULTAR VENTA



DESCRIPCIÓN

1. El vendedor registra los criterios de búsqueda para obtener los datos de la venta o ventas que requiere.
2. El formulario de reportes del sistema valida los datos introducidos por el vendedor.
3. Una vez validados los datos la forma de reportes carga el reporte de la venta con los criterios seleccionados previamente.
4. El reporte es mostrado al vendedor.

CASOS DE USO DE USUARIO

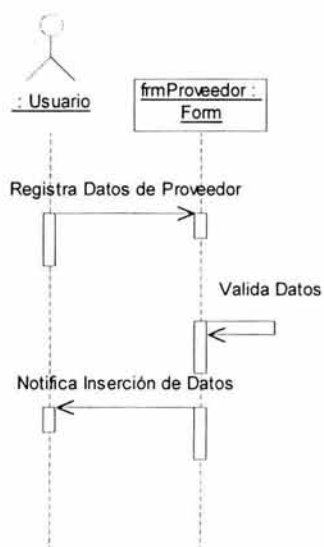


BREVE DESCRIPCIÓN

Por usuario se entiende toda aquella persona que tiene acceso al sistema, es decir: el administrador, el capturista y el usuario, por lo tanto los procesos del usuario son los procesos comunes para todos los roles. Estos procesos corresponden a la gestión de proveedores, clientes, compañías de envío y artículos alterados, además de los procesos de autenticación en el sistema, la configuración de facturas, y la consulta de los reportes por artículo, proveedor y cliente.

DIAGRAMAS DE SECUENCIA PARA CASOS DE USO DE USUARIO (TODOS LOS ROLES)

AGREGAR PROVEEDOR



DESCRIPCIÓN

1. El usuario registra los datos del nuevo proveedor en la forma de proveedores del sistema.
2. El sistema valida los datos en el formulario de proveedor
3. El sistema confirma la inserción exitosa del registro.

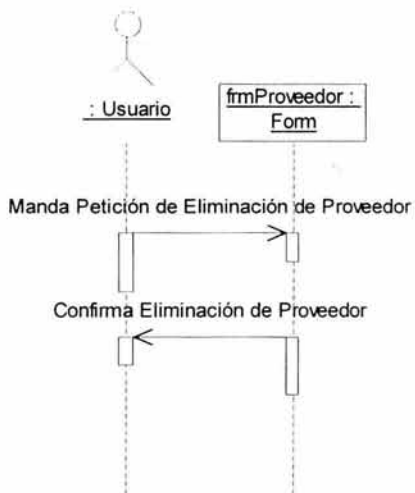
ACTUALIZAR PROVEEDOR



DESCRIPCIÓN

1. El usuario registra los datos a modificar del proveedor en la forma de proveedor del sistema.
2. El sistema valida los datos en el formulario de proveedor.
3. El sistema confirma la modificación del registro.

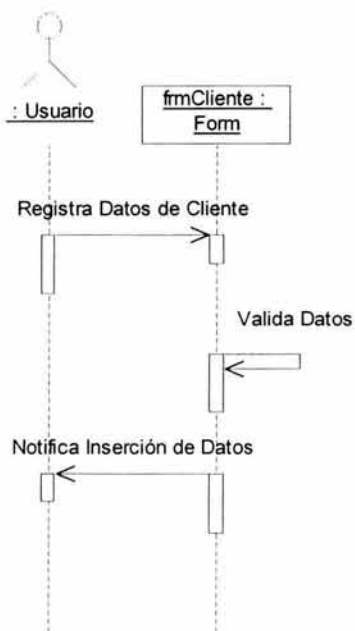
ELIMINAR PROVEEDOR



DESCRIPCIÓN

1. El usuario selecciona el registro de la categoría a eliminar y manda la petición al sistema, en la forma de Categoría.
2. El sistema a través de la forma de Categoría confirma la eliminación del registro.

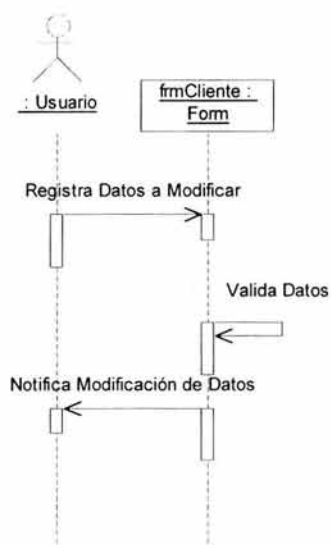
AGREGAR CLIENTE



DESCRIPCIÓN

1. El usuario registra los datos del nuevo cliente en la forma de cliente del sistema.
2. El sistema valida los datos en el formulario de cliente
3. El sistema confirma la inserción exitosa del registro.

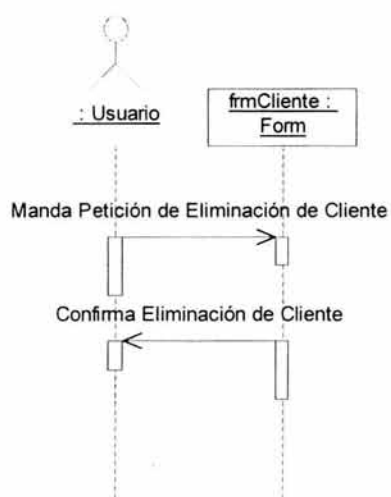
ACTUALIZAR CLIENTE



DESCRIPCIÓN

1. El usuario registra los datos a modificar del cliente en la forma de cliente del sistema.
2. El sistema valida los datos en el formulario de cliente
3. El sistema confirma la modificación del registro.

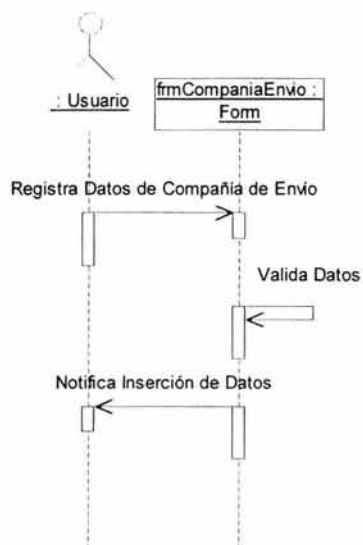
ELIMINAR CLIENTE



DESCRIPCIÓN

1. El usuario selecciona el registro del cliente a eliminar y manda la petición al sistema, en la forma de cliente.
2. El sistema a través de la forma de cliente confirma la eliminación del registro.

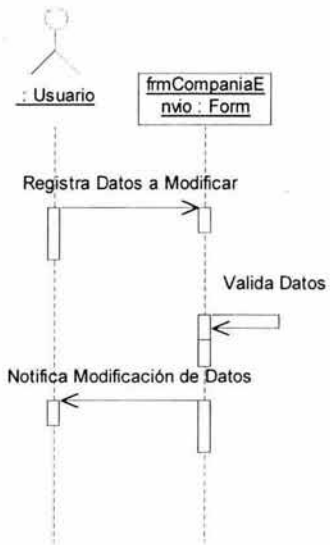
AGREGAR COMPAÑÍA DE ENVÍO



DESCRIPCIÓN

1. El usuario registra los datos de la nueva compañía de envío en la forma de Compañía de Envío del sistema.
2. El sistema valida los datos en el formulario de Compañía de Envío
3. El sistema confirma la inserción exitosa del registro.

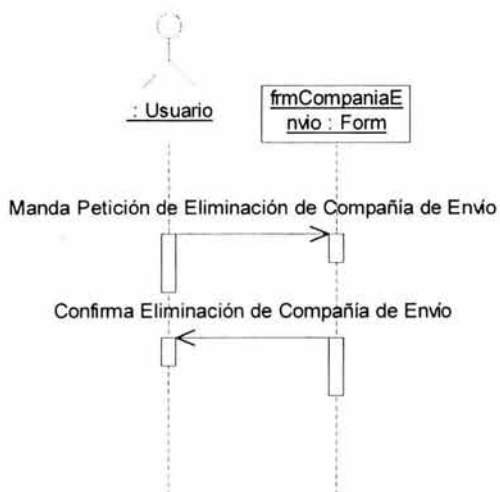
ACTUALIZAR COMPAÑÍA DE ENVÍO



DESCRIPCIÓN

1. El usuario registra los datos a modificar de la compañía de envío en la forma de compañía de envío del sistema.
2. El sistema valida los datos en el formulario de compañía de envío
3. El sistema confirma la modificación del registro.

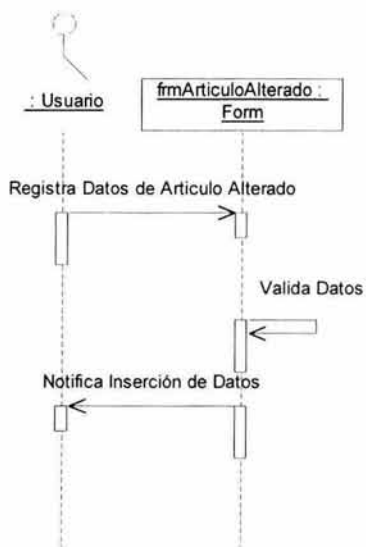
ELIMINAR COMPAÑÍA DE ENVÍO



DESCRIPCIÓN

1. El usuario selecciona el registro de la compañía de envío a eliminar y manda la petición al sistema, en la forma de Compañía de Envío.
2. El sistema a través de la forma de Compañía de Envío confirma la eliminación del registro.

AGREGAR ARTÍCULO ALTERADO



DESCRIPCIÓN

1. El usuario registra los datos del nuevo artículo alterado en la forma de artículo alterado del sistema.
2. El sistema valida los datos en el formulario de artículo alterado
3. El sistema confirma la inserción exitosa del registro.

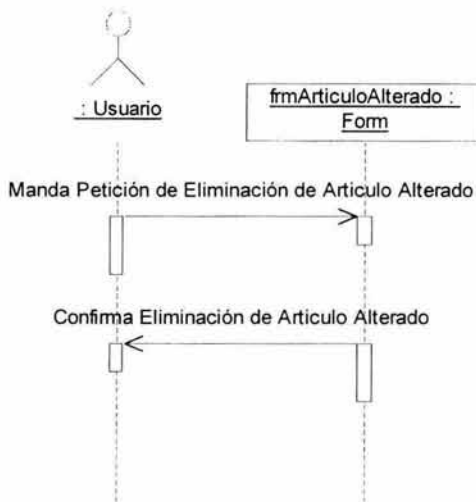
ACTUALIZAR ARTÍCULO ALTERADO



DESCRIPCIÓN

1. El usuario registra los datos a modificar del artículo alterado en la forma de artículo alterado del sistema.
2. El sistema valida los datos en el formulario de artículo alterado.
3. El sistema confirma la modificación del registro.

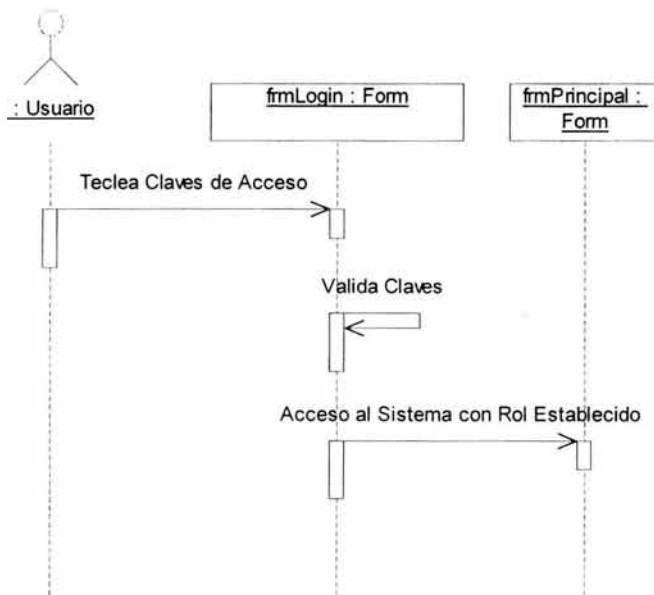
ELIMINAR ARTÍCULO ALTERADO



DESCRIPCIÓN

1. El usuario selecciona el registro del artículo alterado a eliminar y manda la petición al sistema, en la forma de Artículo Alterado.
2. El sistema a través de la forma de Artículo Alterado confirma la eliminación del registro.

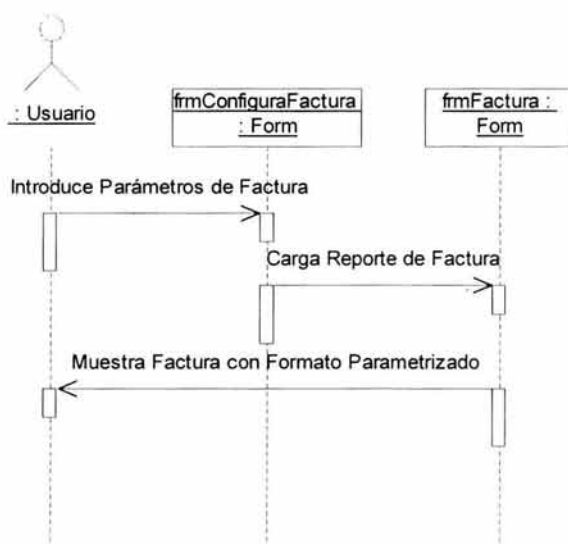
AUTENTICAR



DESCRIPCIÓN

1. El usuario ingresa las claves de acceso
2. El formulario Login del sistema valida las claves ingresadas.
3. El formulario Login carga el formulario principal del sistema.

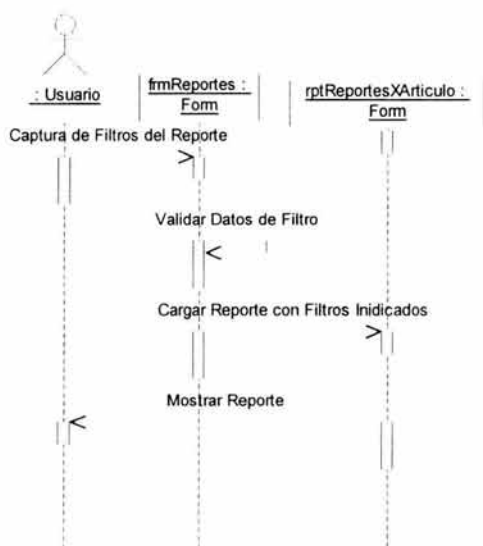
CONFIGURAR FACTURAS



DESCRIPCIÓN

1. El usuario introduce los parámetros de la factura en el formulario de Configuración de Facturas.
2. El formulario de configurar facturas carga el reporte de facturas.
3. El reporte de factura es mostrado al usuario.

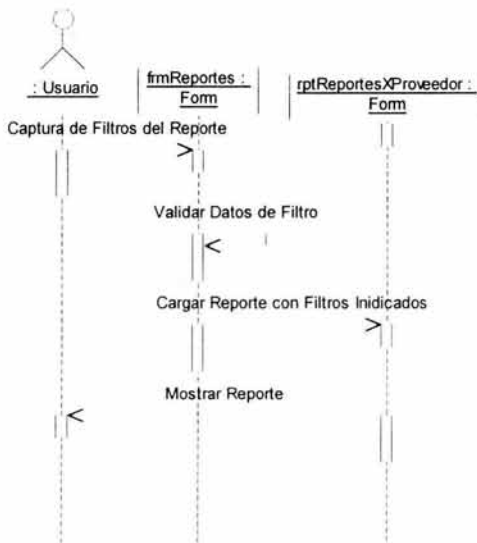
CONSULTAR REPORTES POR ARTÍCULO



DESCRIPCIÓN

1. El usuario captura los filtros del reporte
2. El formulario de reportes valida los datos introducidos.
3. El formulario de reportes carga el reporte por artículo.
4. El reporte es mostrado al usuario

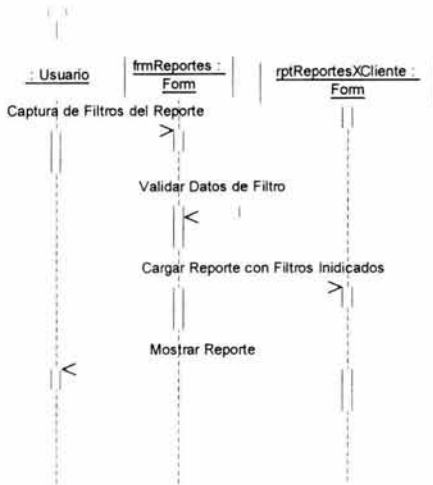
CONSULTAR REPORTES POR PROVEEDOR



DESCRIPCIÓN

1. El usuario captura los filtros del reporte
2. El formulario de reportes valida los datos introducidos.
3. El formulario de reportes carga el reporte por proveedor.
4. El reporte es mostrado al usuario

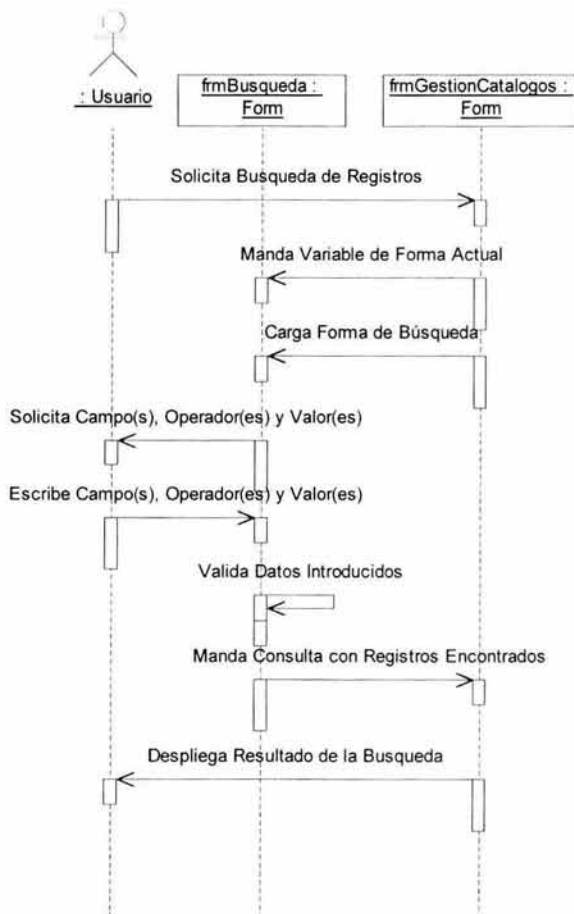
CONSULTAR REPORTES POR CLIENTE



DESCRIPCIÓN

1. El usuario captura los filtros del reporte
2. El formulario de reportes valida los datos introducidos.
3. El formulario de reportes carga el reporte por cliente.
4. El reporte es mostrado al usuario

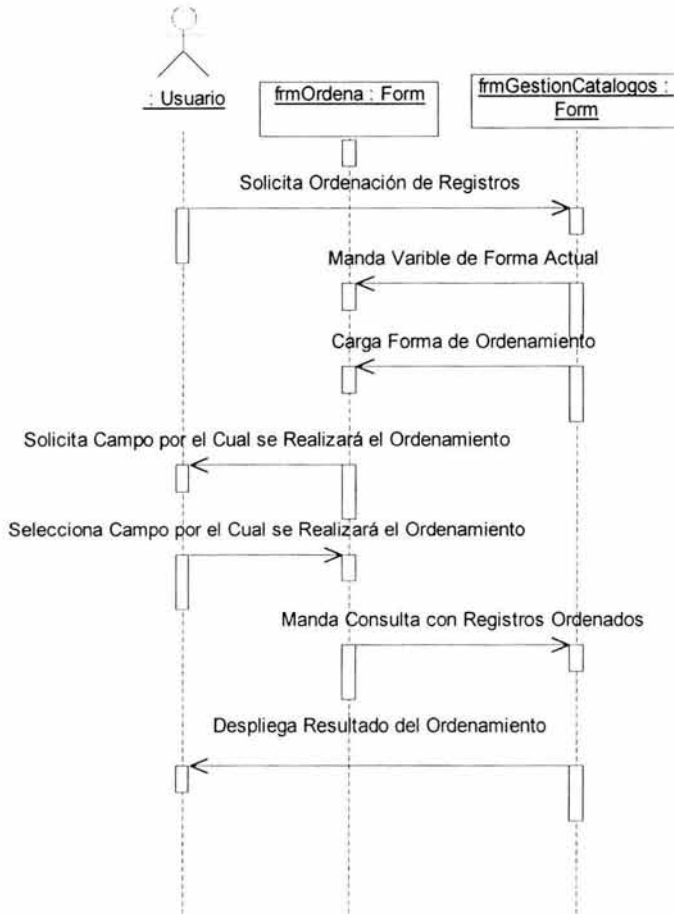
BUSCAR



DESCRIPCIÓN

1. El usuario carga la forma de algún catálogo.
2. La forma del catálogo carga la forma de búsqueda cuando el usuario da click en el botón apropiado y actualiza la variable global para el formulario actual.
3. El formulario de búsqueda pide el filtro de la búsqueda al usuario.
4. El usuario registra los criterios de la búsqueda en el formulario de búsqueda.
5. El formulario de búsqueda valida los datos introducidos.
6. El formulario de búsqueda manda la consulta al formulario del catálogo actual.
7. El formulario del catálogo actual muestra los registros encontrados al usuario.

ORDENAR



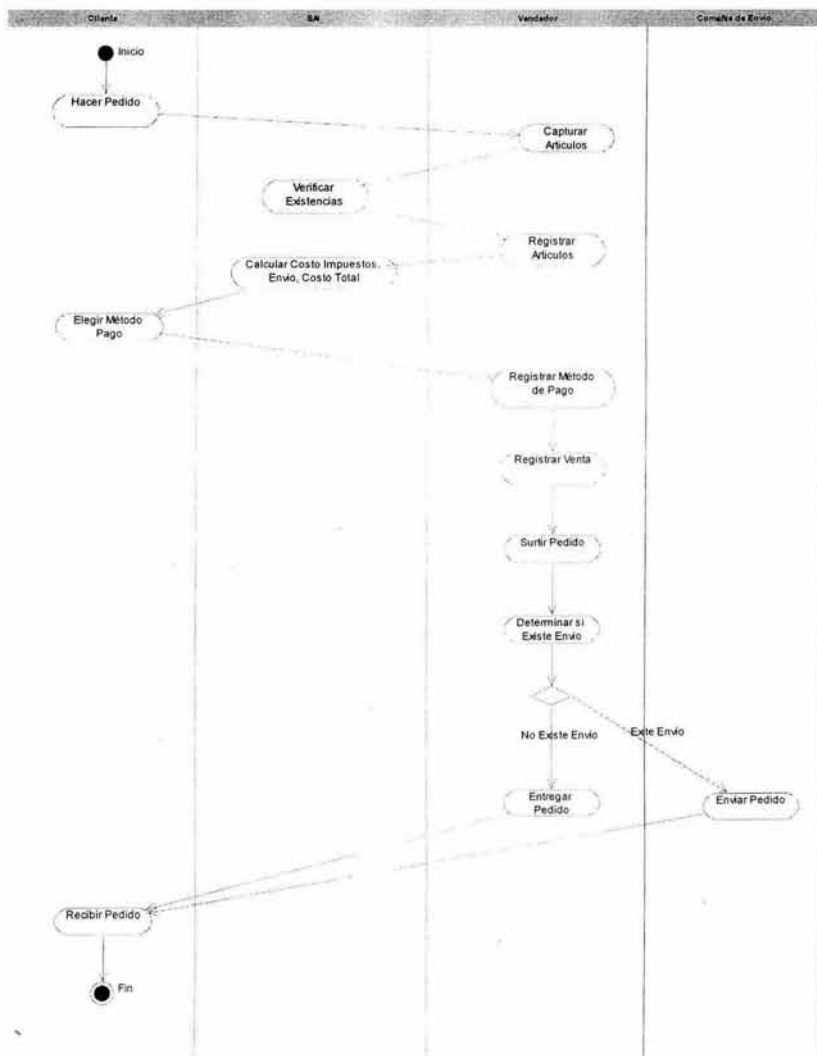
DESCRIPCIÓN

1. El usuario carga la forma de algún catálogo.
2. La forma del catálogo carga la forma de ordenamiento cuando el usuario da click en el botón apropiado y actualiza la variable global para el formulario actual.
3. El formulario de ordenamiento pide el campo de la búsqueda al usuario.
4. El usuario registra el campo de ordenación en el formulario de ordenación.
5. El formulario de ordenación manda la consulta al formulario del catálogo actual.
6. El formulario del catálogo actual muestra los registros ordenados al usuario.

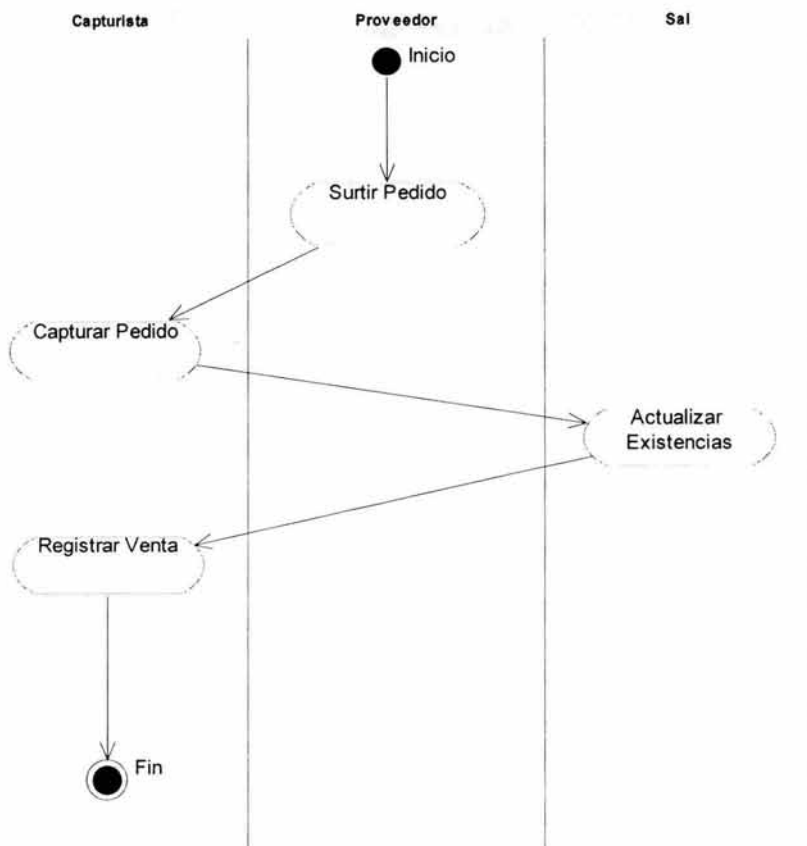
DIAGRAMAS DE ACTIVIDAD

Los siguientes diagramas presentan las actividades más importantes dentro del SAI, como lo son la venta, la compra y los pagos. El renglón superior muestra los objetos participantes y la columna que les corresponde presenta las actividades asociadas (en óvalos), las flechas representan el flujo de la actividad completa.

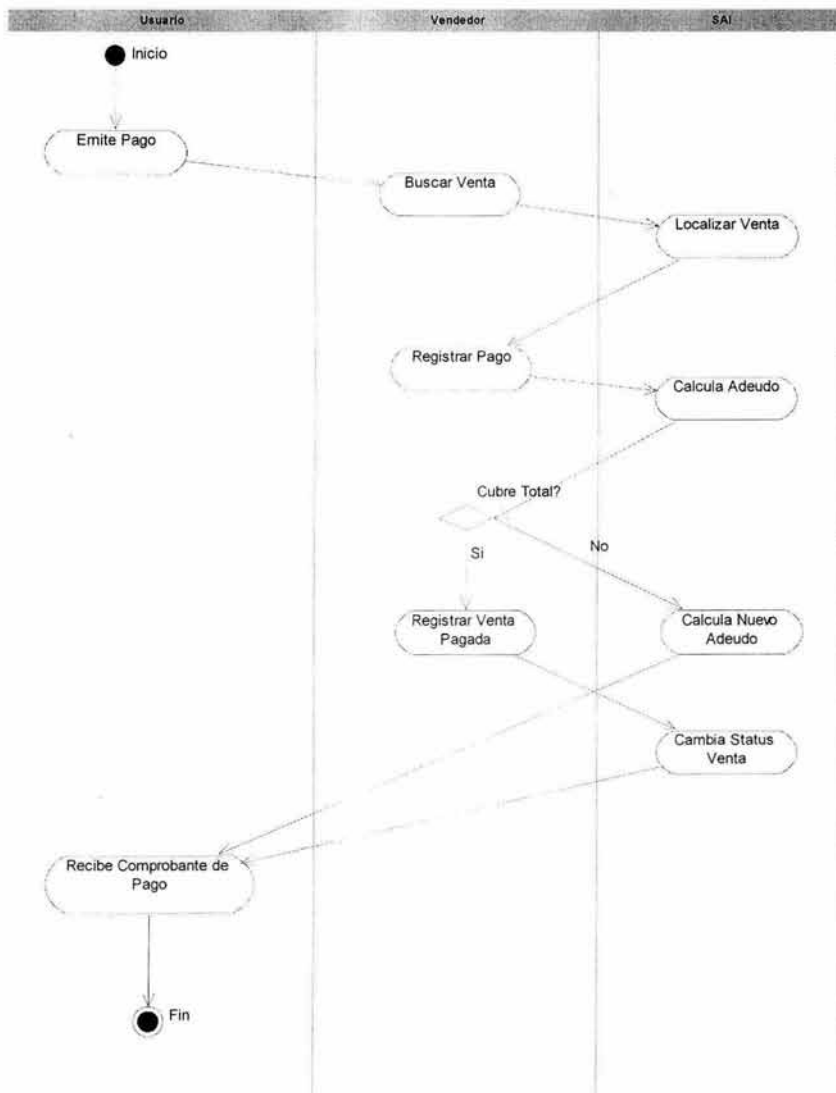
VENTA



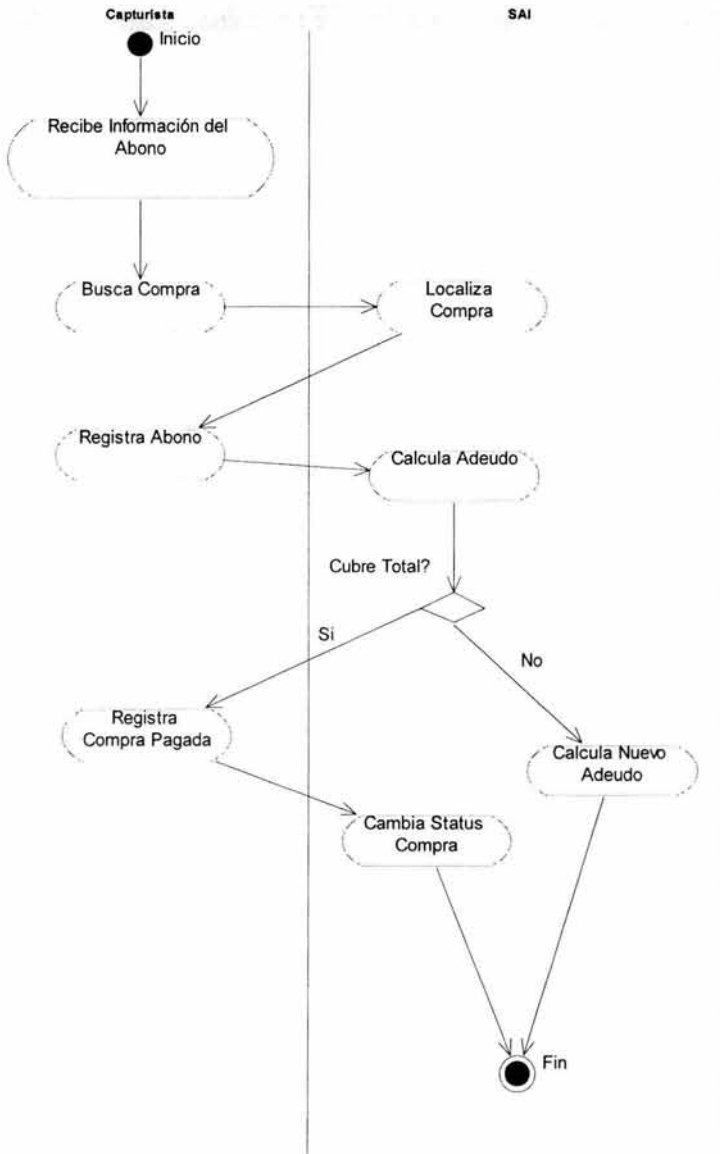
COMPRA



PAGO



ABONO



APÉNDICE B

DICCIONARIO DE DATOS DEL SAI

SISTEMA DE ADMINISTRACIÓN Y CONTROL DE INVENTARIOS							
DICCIONARIOS DE DATOS							

Entidad	Nombre de Atributo	Tipo	Long	Definición	Llave	Nulo	Ejemplo	
ABONO	abonoId	int	4	Identificación del abono que se pago a un proveedor	FK	0	1251	
	Descripción	compralId	int	4	Identificación de la compra al proveedor		1	1000
	Son los abonos o pagos dados a los proveedores	abonoMonto	money	8	Monto del abono		1	\$1,000
		abonoFecha	datetime	8	Fecha en la que se realizo el abono al proveedor		1	12/10/2003
	abonoTipoPago	varchar	15	Tipo de pago del abono		1	Efectivo	
ARTICULO	articuloId	int	4	Identificación del articulo	PK	0	1	
	Descripción	categorialId	int	4	Identificación de la categoría	FK	1	27
	Son los productos que vende la organización	proveedorId	int	4	Identificación del proveedor	FK	1	13
		articuloNombre	char	35	Nombre corto del articulo		1	Foco
		articuloUnidad	varchar	10	Unidad empleada para medir el articulo		1	Metro
		articuloMarca	varchar	20	Marca comercial del articulo		1	Elm
		articuloModelo	varchar	20	Modelo del articulo, definido por el fabricante		1	M-23
		articuloDescripcion	varchar	80	Descripción detallada del articulo		1	Foco fluorescente de 40w. Luz de día
		articuloExistencia	float	8	Número de articulos en existencia		1	150

Entidad	Nombre de Atributo	Tipo	Long	Definición	Llave	Nulo	Ejemplo
ARTICULO	articuloCantidadPorUnidad	float	8	Cantidad de elementos contenidos en un producto unitario que viene empaquetados		1	12
	articuloPrecioUCompra	money	8	Precio unitario del artículo al que es adquirido		1	\$50.00
	articuloPrecioUventa	money	8	Precio unitario del artículo al que es vendido		1	\$75.00
	articuloObservaciones	varchar	80	Las observaciones particulares del artículo		1	Artículo importado. Edición limitada
	articuloResumen	varchar	80	Información adicional del artículo		1	Artículo Alemán de alta calidad. Uso rudo
	articuloSuspendido	tinyint	1	Opción que indica si el artículo está suspendido para la venta		1	0
	articuloImagen	varchar	80	Ruta de la imagen del artículo		1	C:\img\art52.jpg
ARTICULO_COMPRA	compralId	int	4	identificador de la compra	FK	0	65
Descripción	articuloId	int	4	identificador del artículo	FK	0	5266
Es el detalle de los artículos adquiridos en una compra	articulo_compra PrecioUCompra	money	8	Precio unitario al que se adquirió el artículo en una compra específica		1	\$40.50
	articulo_compraCantidad	money	8	Cantidad del artículos comprados en una compra específica		1	100
	articulo_compraDescuento	int	4	Descuento obtenido por cada artículo en una compra específica		1	0
	articulo_compraImporte	money	8	Importe del Artículo comprado		1	\$1,600

Entidad	Nombre de Atributo	Tipo	Long	Definición	Llave	Nulo	Ejemplo	
ARTICULO_VENTA	ventaId	int	4	identificador de la venta	FK	0	899	
	Descripción	articuloId	int	4	identificador del artículo	FK	0	1689
		articulo_ventaPrecioUVenta	money	8	Cantidad del artículo vendidos en una compra específica		1	\$500
	Es el detalle de los artículos producto de en una venta	articulo_ventaCantidad	int	4	Descuento otorgado por cada artículo en una venta específica		1	3
		articulo_ventaDescuento	int	4	Importe del Artículo vendido		1	0.1
		articulo_ventaImporte	real	4			1	10000
ARTICULODETERIORADO	artículoDeterioradoId	char	10	identificador del artículo alterado	PK	0	15	
	Descripción	artículoId	int	4	identificador del artículo	FK	1	1667
	Son los arpiculos que por alguna causa dejan de ser útiles y no se pueden vender	articuloDeterioradoCausa	char	10	Causa por la que se deterioro el artículo		1	Deterioro natural
		articuloDeteriorado Responsable	int	4	identificador del empleado responsable que registró el artículo alterado		1	1
		articuloDeterioradoFecha	datetime	8	Fecha en la que se registro el artículo deteriorado		1	12/12/2003
		articuloDeteriorado Cantidad	float	8	Cantidad de un mismo artículo deteriorado		1	1
CATEGORIA	categoríaId	int	4	identificador de la categoría	PK	0	5	
	Descripción	categoríaNombre	varchar	20	Nombre de la categoría		1	Iluminación
		categoríaDescripción	varchar	80	Descripción detallada de la categoría		1	Todos los artículos de iluminación
	Son las clasificaciones por medio de las cuales se agrupan los artículos	categoríaSuperior	int	4	Categoría a la que pertenece la presente categoría		1	Electricidad
		categoríaImagen	varchar	80	Ruta de la imagen que representa a la categoría		1	C:\img\cat5.jpg

Entidad	Nombre de Atributo	Tipo	Long	Definición	Llave	Nulo	Ejemplo
CLIENTE	clienteId	int	4	identificador del cliente	PK	0	1633
Descripción	clienteNombreEmpresa	varchar	50	Nombre de la empresa a la que pertenece el cliente		1	Elektra
Son los datos de los clientes de la organización	clienteNombreContacto	varchar	50	Nombre del contacto del cliente		1	Pedro López
	clienteRFC	varchar	15	RFC del cliente al que se le va a facturar		1	DALOP751104
	clienteClave	varchar	30	Clave especial que se asigna a un cliente		1	Elektra
	clienteDireccion	varchar	80	Calle, Numero y Colonia del cliente		1	Av 152, 24, Tepeyac
	clienteCiudad	varchar	15	Ciudad del cliente		1	D.F.
	clienteCP	varchar	5	Código postal del cliente		1	58163
	clientePais	varchar	15	Pais del cliente		1	México
	clienteTelefono	varchar	15	Teléfono del cliente		1	59663628
	clienteMail	varchar	50	Correo electrónico del cliente		1	pedro@mail.com
	clienteFax	varchar	15	Fax del Cliente		1	56789615
COMPRA	compraId	int	4	identificador de la compra	PK	0	178
Descripción	proveedorId	int	4	identificador del proveedor	FK	1	18
Son los registros de la operación de una compra realizados a un proveedor	compraPedidoFecha	datetime	8	Fecha en la que se realizó el pedido		1	01/06/2002
	compraLimitePagoFecha	datetime	8	Fecha limite para pagar la compra al proveedor		1	01/07/2002
	compraEnvioFecha	datetime	8	Fecha en la que se envió el pedido por parte del proveedor		1	15-Jun
	compraCancelacionCausa	char	10	Causa por la que se cancelo una compra registrada		1	Retardo del envío
	compraCancelacionFecha	datetime	8	Fecha en la que se canceló una compra registrada		1	28/06/2002

Entidad	Nombre de Atributo	Tipo	Long	Definición	Llave	Nulo	Ejemplo
COMPRA	compraEstatus	varchar	10	Status de la venta actual		I	Cancelado
	compraTotal	money	8	Total de la compra incluyendo impuestos		I	\$1,000
	compraIva	money	8	Impuesto al valor agregado		I	\$150
	compraSubtotal	money	8	Total antes de impuestos		I	\$850
EMPLEADO	empleadoId	int	4	identificador del empleado	PK	0	4
Descripción Son los datos de los trabajadores de la organización	empleadoApellidoPaterno	varchar	20	Apellido paterno del empleado		I	Juárez
	empleadoApellidoMaterno	varchar	20	Apellido materno del empleado		I	Hernández
	empleadoNombre	varchar	20	Nombre del empleado		I	Humberto
	empleadoCargo	varchar	30	Cargo del empleado dentro de la organización		I	Vendedor
	empleadoFechaNacimiento	datetime	8	Fecha de nacimiento del empleado		I	06/05/1972
	empleadoFechaContratacion	datetime	8	Fecha en la que se contrato al empleado		I	13/04/2000
	empleadoDireccion	varchar	50	Calle, Número y Colonia del empleado		I	Bosques de Africa 27, Bosques de Aragón
	empleadoCiudad	varchar	50	Ciudad del Empleado		I	Nezahualcoyotl
	empleadoCP	varchar	7	Código Postal del empleado		I	57181
	empleadoEstado	varchar	20	Estado del Empleado		I	México
	empleadoTelefono	varchar	15	Teléfono del empleado		I	56876115
	empleadoEmail	varchar	40	Correo electrónico del empleado		I	hernandez@usa.net
	empleadoJefe	int	4	identificador del jefe del empleado		I	1
empleadoLogin	varchar	15	Nombre de Usuario del Empleado		I	humberto	
empleadoPassword	varchar	15	Contraseña del empleado		I	hh64558s	

Entidad	Nombre de Atributo	Tipo	Long	Definición	Llave	Nulo	Ejemplo
EMPLEADO	empleadoTipoPermiso	varchar	20	Tipo de permiso del empleado		1	Vendedor
	empleadoFoto	varchar	80	Ruta de la imagen de la foto del empleado		1	C:\img\emp4.jpg
PAGO	pagoid	int	4	Identificador del pago	PK	0	2
Descripción	ventaId	int	4	Identificador de la venta	FK	1	189
Son los pagos cobrados a los clientes	pagoMonto	money	8	Monto de la cantidad pagada		1	\$500
	pagoTipoPago	varchar	10	Tipo de pago		1	Contado
	pagoFecha	datetime	8	Fecha en la que se realizó el pago		1	19/03/2003
PROVEEDOR	proveedorId	int	4	Identificador del proveedor	PK	0	15
Descripción	proveedorNombreContacto	varchar	30	Nombre de contacto del proveedor		1	Adriana Luna
Son los datos de los proveedores de la organización	proveedorCargoContacto	varchar	30	Cargo del proveedor en su organización		1	Vendedor
	proveedorNombreEmpresa	varchar	30	Nombre de la empresa del proveedor		1	Distribuidora Industrial
	proveedorDireccion	varchar	80	Calle, Número y Colonia del proveedor		1	Valle Perdido 78, Valle de Aragón
	proveedorCiudad	varchar	15	Ciudad del proveedor		1	México
	proveedorCP	varchar	5	Código postal del proveedor		1	56123
	proveedorPais	varchar	15	País del proveedor		1	México
	proveedorTelefono	varchar	13	Teléfono del proveedor		1	57661842
	proveedorExtension	varchar	5	Extensión del proveedor		1	15
	proveedorEmail	varchar	40	Correo electrónico del proveedor		1	adrianalu@di.com
proveedorFax	varchar	13	Fax del proveedor		1	57891523	

Entidad	Nombre de Atributo	Tipo	Long	Definición	Llave	Nulo	Ejemplo
VENTA	ventaId	int	4	identificador de la venta	PK	0	1887
Descripción	empleadId	int	4	Identificador del empleado	FK	0	4
Son los registros de la operación de una venta realizada por los clientes	clienteId	int	4	identificador del cliente	FK	0	564
	ventaPedidoFecha	datetime	8	Fecha en la que se realizó el pedido		1	23/08/2003
	ventaPedidoHora	varchar	8	Hora en la que se realizó el pedido		1	04:35
	ventaEnvioFecha	datetime	8	Fecha en la que se realizó el envío del pedido		1	24/08/2003
	ventaEnvioForma	varchar	10	Manera en que fue enviado el pedido		1	Paquetería
	ventaEnvioCosto	money	8	Costo del envío del pedido		1	\$150
	ventaDestinatarioNombre	varchar	20	Nombre del destinatario		1	Pedro Uribe
	ventaDestinatarioDireccion	varchar	30	Calle, Número y Colonia del destinatario		1	Av 402, No. 15, San Juan de Aragón
	ventaDestinatarioCiudad	varchar	10	Ciudad del destinatario		1	D.F.
	ventaDestinatarioCP	varchar	10	Código postal del destinatario		1	52148
	ventaDestinatarioTelefono	varchar	20	Teléfono del destinatario		1	57661625
	ventaCancelacionCausa	varchar	50	Causa de cancelación de la venta		1	Varios artículos deteriorados
	ventaCancelacionFecha	datetime	8	Fecha en que fue cancelada la venta		1	15/09/2003
	ventaCancelacionHora	datetime	8	Hora en que fue cancelada la venta		1	12:30
	ventaEstatus	varchar	10	Status en que se encuentra la venta		1	Cancelado
	ventaTotal	money	8	Total de la venta		1	\$500
	ventaIva	money	8	Impuesto al valor agregado de la venta		1	\$150
	ventaSubtotal	money	8	Total antes de impuestos		1	\$350

APÉNDICE C

GLOSARIO

Arquitectura. Estructura organizativa de un sistema, que incluye su descomposición en partes, conectividad, mecanismos de interacción y principios de guía que proporcionan información sobre el diseño del mismo.

Archivo. Colección de registros almacenados siguiendo una estructura homogénea dentro de un sistema operativo. Cada uno de ellos se identifica comúnmente mediante una extensión que es una serie de caracteres que se escriben en su nombre después de un punto.

Base de datos. Colección de archivos interrelacionados que son creados con un DBMS. El contenido de una base de datos engloba a la información concerniente (almacenadas en archivos) de una organización, de tal manera que los datos estén disponibles para los usuarios, una finalidad de la base de datos es eliminar la redundancia o al menos minimizarla.

Base de datos relacional. Aplicación de la teoría matemática de conjuntos al problema de la organización de datos. En una base de datos relacional, los datos están organizados en tablas (llamadas relaciones en la teoría relacional).

Campo. Es la unidad más pequeña a la cual uno puede referirse en un programa. Desde el punto de vista del programador representa una característica de un individuo u objeto. En las bases de datos representan los atributos de una tabla.

Casos de Uso. Representa una secuencia de acciones que un sistema lleva a cabo para ofrecer algún resultado de valor para un actor u objeto externo que interactúa con el sistema. Los casos de uso representan toda la funcionalidad del sistema. El modelo de casos de uso es una especificación completa de todas las formas posibles de utilizar un sistema.

Ciclo de Vida. Ciclo por el cual atraviesa todo sistema de información. Durante su desarrollo, cada sistema se mueve a través de varias fases de un ciclo de vida, después del cual sólo funciona por varios años con un mínimo mantenimiento. El sistema se deteriora gradualmente hasta el punto en que cesa de funcionar por completo y se comienza un nuevo ciclo de vida con el desarrollo de un nuevo sistema.

DBA. Persona o equipo de personas profesionales responsables del control y manejo del sistema de base de datos, generalmente tienen experiencia en DBMS, diseño de bases de datos, sistemas operativos, comunicación de datos, hardware y programación.

DBMS. Sistema administrador de base de datos. Colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de una tarea específica.

Diccionario de datos. El contenido del diccionario de datos puede considerarse “datos acerca de los datos” (los cuales en ocasiones reciben el nombre de metadatos), es decir, definiciones de objetos de la base de datos, y no tan solo “datos en bruto”.

Estructuras de Control. Estructuras que definen la secuencia de ejecución o control de flujo de un programa tales como las estructuras de decisión y de repetición.

Formulario. Contenedores de los demás controles u objetos dentro de una aplicación de Visual Basic, se asocian directamente a lo que conocemos como una ventana de Windows, ya que con ellos establecemos muchas de las propiedades de las ventanas como su tamaño, color ubicación, etc.

MDI. Interfaz de documento múltiple. Una aplicación MDI permite que el usuario vea varias ventanas al mismo tiempo. Las ventanas secundarias están contenidas en una ventana primaria, que proporciona un espacio de trabajo para todas las ventanas secundarias de la aplicación.

Inventario. Recursos utilizables que se encuentran almacenados para su uso posterior en un momento determinado.

Integridad. Se refiere no al hecho de que los valores que posean los datos sean válidos de acuerdo a las dependencias funcionales entre las tablas y de acuerdo a las políticas del negocio.

Lenguaje de Definición de Datos. Especifica la forma en que los datos pueden almacenarse.

Lenguaje de Manipulación de Datos. Consiste en la forma de recuperar datos almacenados en una BD.

Lenguaje de Control de Datos. Es el encargado de cuidar la seguridad de los datos. En él se da la concesión de privilegios a los diferentes usuarios de la base de datos.

Modelo. Representación de algo real, en él se captan los aspectos importantes de lo que se esta representando desde cierto punto de vista por medio de abstracciones que permiten contemplar la funcionalidad principal de lo que se esta modelando para simplificar u omitir el resto.

Modelo Relacional. Un modelo de datos expresa entidades y sus relaciones y es la herramienta utilizada para representar la organización conceptual de los datos, siendo necesario para ello el uso de cierta terminología que se verá a continuación.

Módulo. Dentro de Visual Basic: archivo donde se agrupan las variables y funciones que tienen un alcance global en la aplicación, es decir, que están accesibles desde cualquier parte de la aplicación.

Objeto. Combinación de código y datos que se puede tratar como una unidad. Un objeto puede ser una parte de una aplicación, como un control o un formulario. Los objetos en Visual Basic los encontramos en los botones, en las bases de datos, en los gráficos, etc.

Procedimiento. Indica a la aplicación cómo realizar una tarea específica. Una vez que se define un procedimiento general, se le debe llamar específicamente desde la aplicación. Por el contrario, un procedimiento de evento permanece inactivo hasta que se le llama para responder a eventos provocados por el usuario o desencadenados por el sistema.

Proceso. Conjunto de actividades, métodos y prácticas para desarrollar y mantener el software y los productos asociados. Define qué hacer, cuándo y cómo hacerlo, y quién o quienes deben hacerlo.

Recordset. Objetos encargados de mantener las datos obtenidos de una base de datos en una variable de tipo objeto, es como un cursor de cualquier manejador de base de datos en los que uno puede navegar a través de los registros. Pueden ser dinámicos o estáticos, esto es, pueden ser manipulados como si estuviera en la base de datos o tomar una fotografía de los datos para consultarlos o para después regresar estos datos modificados a la base de datos.

Redundancia. Se refiere, a la existencia de información repetida o duplicada en diferentes tablas dentro de una base de datos.

Registro. Colección de campos de iguales o de diferentes tipos de dato. Regularmente se refiere a un renglón o tupla dentro de una tabla de la base de datos.

SAI. Sistema de Administración y Control de Inventarios. Sistema informático que ayuda en el manejo de recursos almacenados apoyado en un uso homogéneo de los catálogos involucrados en el proceso de compra-venta.

Sinergia. Sistema tomado globalmente posee propiedades que no tienen sus partes, tomadas individualmente. Esto es, el sistema tiene una capacidad de actuación mayor o menor que la de la suma de sus partes

Sistema. Conjunto de partes que colaboran para alcanzar uno o más objetivos. Todos los elementos del sistema deben estar ligados y controlados de manera que se logre el objetivo.

Sistema de Información. Aquel sistema que realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información.

SQL. Lenguaje estructurado de consulta para manejar las bases de datos relacionales. Las sentencias de SQL son como frases escritas en inglés, donde decimos lo que deseamos obtener y donde obtenerlo. Todas las sentencias siguen una sintaxis para que se puedan ejecutar correctamente.

Tipo de Dato. Se refiere a las diferentes categorías de datos con los que se fijan las características de un campo dentro de una base de datos. Por ejemplo, existen datos de tipo entero, flotante, carácter, booleano, etc.

UML. Lenguaje de Modelado Unificado. Es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Fue desarrollado en un esfuerzo para simplificar y consolidar las notaciones de desarrollo orientado a objetos que habían surgido.

Vista. Consultas SQL almacenadas en el servidor de base de datos para evitar repetir una misma consulta en varias partes, de esta forma se hace solo una consulta sencilla a la vista.

Visual Basic. Lenguaje de programación orientado a eventos, basado en el lenguaje Basic que permite crear interfaces graficas de usuario de una forma sencilla. La palabra "Visual" hace referencia al método que se utiliza para crear la interfaz gráfica de usuario (GUI). En lugar de escribir numerosas líneas de código para describir la apariencia y la ubicación de los elementos de la interfaz, simplemente se agregan objetos prefabricados en su lugar dentro de la pantalla. La palabra "Basic" hace referencia al lenguaje BASIC (Beginners All-Purpose Symbolic Instruction Code), un lenguaje utilizado por más programadores que ningún otro lenguaje en la historia de la informática o computación.

BIBLIOGRAFÍA

Ambler, Scott
¿Qué le falta a UML?
Objet Magazine
Octubre de 1997

Booch, Grady
Análisis y Diseño Orientado a Objetos
Adison – Wesley, Días de Santos
Estados Unidos, 1994

Harmon, Paul
Entendiendo UML: La guía del desarrollador
Morgan Kauffman Publishers, Inc.
1998

John G. Burch
Diseño de Sistemas de Información
Gary Grudnitski
Ed. Megabyte

Kendall
Análisis y Diseño de Sistemas
Prentice-Hall
España, 1992

Lawrence, Shari
Ingeniería de Software, Teoría y Práctica
Pentice Hall
Brasil, 2002

Sin Autor
Microsoft Visual Basic 6.0, Manual del Programador
Microsoft Press, Mc Graw Hill
España, 1998

Vaughn, William R.
Programación de SQL Server con
Visual Basic 6.0
Microsoft Press, Mc Graw Hill
España, 1998

Referencias

www.mkp.com/books_catalog/
www.sigs.com/omo/articles/ambler.html
Ayuda de Microsoft Windows 2000