

UNIVERSIDAD LASALLISTA BENAVENTE



ESCUELA DE INGENIERÍA EN COMPUTACIÓN

Con estudios incorporados a la
Universidad Nacional Autónoma de México

CLAVE: 8793-16

“DESARROLLO DE UN SISTEMA ADMINISTRATIVO REMOTO”

TESIS

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN

PRESENTA:

JUAN GERARDO CÁRDENAS TRIGUEROS

ASESOR: **ING. ALEJANDRO GUZMÁN ZAZUETA**

Celaya, Gto.

Noviembre 2008



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



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.

AGRADECIMIENTOS PERSONALES

A Dios por iluminarme y guiarme en cada momento de mi vida

A mis padres. Por enseñarme lo que sé de la vida y seguir enseñándome, apoyándome en todo momento.

A mis hermanas, por ser los mejores compañeros y amigos en los que siempre puedo confiar.

A mis maestros, por compartirme sus conocimientos de los que ahora gozo.

A mi asesor, por el apoyo y confianza con la que siempre me ha tratado

ÍNDICE

INTRODUCCIÓN

Capítulo 1 Compupaq S.A de C.V	1
1.1 Antecedentes	2
1.2 Negocio y participación del mercado	2
1.3 Objetivo del proyecto	3
1.4 Situación actual	3
1.5 Sistemas de Información (SI)	4
1.5.1 Introducción	4
1.5.2 Tipos de sistemas de información	6
1.5.3 Evolución de los sistemas de información	8
Capítulo 2 Metodología usada para la implementación de un SI	11
2.1 Ciclo en espiral para el desarrollo de sistemas	12
2.1.1 Objetivos de cada fase	15
2.2 Herramientas de análisis y diseño de sistemas	17
2.2.1 Planteamiento de objetivos	17
2.2.2 Entrevista con el usuario	17
2.2.3 Rentabilidad de los objetivos	23
2.2.4 Elaboración de diagramas de flujo	24
2.2.5 Determinación de requerimientos	27
2.2.6 Diccionario de datos	31
2.2.7 Modelo entidad relación	33
Capítulo 3 Herramientas de informática para el desarrollo de un sistema	42
3.1 Introducción a la plataforma .NET	43
3.1.1 ¿Qué es .net?	43
3.1.2 El CLR	45
3.1.3 Ensamblados	46
3.2 Visual Studio .Net	48
3.2.1 Introducción	48
3.2.2 Entorno Intellisense	48
3.2.3 Programación orientada a objetos	49

3.2.3.1	Conceptos fundamentales	50
3.2.3.2	Características de la POO	51
3.3	Instalación de Visual Studio.Net	53
3.4	Visual Basic	57
3.4.1	Historia	57
3.4.2	Módulos y clases	57
3.4.3	Procedimientos	58
3.4.4	Aplicaciones MDI	62
3.4.5	MySQL Connector/Net	64
3.4.5.1	Introducción	64
3.4.5.2	Descargando e instalando	65
3.4.5.3	Arquitectura del Connector/Net	66
3.4.5.4	MySqlConnection	67
3.4.5.5	Mysqlcommand	68
3.4.6	Datagridview	71
3.5	Mysql	74
3.5.1	Introducción	74
3.5.2	Panorámica del sistema de gestión de base de datos mysql	74
3.5.3	Principales características de mysql	75
3.5.4	Instalar mysql en Windows	78
Capítulo 4 Implementación del sistema compuPAQ remoto		94
4.1	Implementación del sistema	95
4.2	Reportes	98
4.3	Asignación de reporte	104
4.4	Productos	106
4.5	Requisición de productos o servicios	108

Conclusiones

Bibliografía

INTRODUCCIÓN

En la actualidad los sistemas de información (SI) y las tecnologías de información (TI) han forzado a que las organizaciones actuales operen de forma diferente. A través de su uso se logran importantes mejoras, suministran información necesaria para la toma de decisiones, automatizan procesos operativos y lo más importante logran ventajas competitivas o reducir la ventaja de los rivales.

Las Tecnologías de la Información han sido definida como la integración de la computación, las telecomunicaciones y la técnica para el procesamiento de datos, donde sus principales componentes son: el factor humano, los contenidos de la información, el equipamiento, la infraestructura, el software y los mecanismos de intercambio de información, los elementos de política y regulaciones, además de los recursos financieros.

El objetivo del proyecto es el de desarrollar un sistema de información, con el fin de agilizar y llevar un mejor control de las funciones administrativas de la empresa compupaq, con esto se pretende tener un mejor control de las funciones que se desempeñan dentro de la organización y dar a conocer las ventajas de los sistemas de información en la actualidad.

Para el presente trabajo se hará uso del ciclo de vida conocido como método ciclo en espiral. En este se incluye el análisis y diseño de un punto de venta y control de reportes para equipos de reparación, en el cual se muestran sus ventanas de captura, consultas y sus reportes. En el cual estará conformado de cuatro capítulos que se describirán a continuación.

CAPÍTULO 1

Se describe la empresa en la cual se basa el análisis, mencionando los procesos que se manejan, y dar a conocer la historia de los sistemas de información dando a conocer su función y partes que lo componen y como este va evolucionando.

CAPÍTULO 2

Se explica la forma en que se lleva a cabo el ciclo de vida de desarrollo de los sistemas y mostrando cada una de las partes que forman parte de él y las herramientas para su correcto análisis.

CAPÍTULO 3

Se presentan las características del sistema de programación usado, el servidor de base de datos, la estructura cliente servidor y la programación orientada a objetos.

CAPÍTULO 4

Se presenta la implantación del sistema, las ventanas de captura, consultas y reportes que conforman todo el sistema.

Por último se encuentran las conclusiones a las que se llegó con el desarrollo del trabajo.

CAPÍTULO I

Compupaq S.A de C.V

1.1 ANTECEDENTES

Distribuidora CompuPAQ, fundada en 1998 es una empresa dedicada a la distribución de equipo de cómputo y servicios informáticos avanzados (Mantenimiento, desarrollo, diseño, E-Commerce) al alcance de las micro, pequeñas y medianas empresas.

El propósito de la empresa es el de consolidarse como líder, con productos y servicios informáticos de la más alta calidad, para satisfacer las necesidades de las micro, pequeñas y medianas empresas.

1.2 NEGOCIO Y PARTICIPACIÓN DEL MERCADO

CompuPAQ ofrece desarrollo de software, servicios de mantenimiento y vende:

- Desarrollo de software basado en la tecnología. net el lenguaje en el que desarrollan es Vb.net y C++. net ofreciendo con este el desarrollo de base de datos y el hospedaje en su servidor principal rentándolo a un precio accesible a sus clientes, también ofrecen la implementación de E-commerce a través del Oscommerce.
- Servicios de mantenimiento que ofrece son a nivel software y hardware son : reparación de laptops, monitores, impresoras, mantenimiento preventivo, instalación de redes y configuraciones de servidores en Windows 2003
- El catalogo de ventas de CompuPAQ es extenso ya que ofrece una gama muy variada de sus productos como redes, servidores, juegos, software etc. Para más información de lo que ofrece puede visitar su página Web en <http://compupaq.homelinux.net>.

1.3 OBJETIVO DEL PROYECTO

El objetivo del proyecto es el de implantar un sistema que automatice y agilice

varias tareas tales como: control de inventario, reportes, compra-venta de productos, asignación de reportes y alta de proveedores y clientes.

Con ello se espera el ahorro de tiempo en procesos como el del inventario y dar una respuesta más rápida a los problemas teniendo que el estado de los reportes de a conocer un panorama de cómo se encuentra un equipo sin necesidad de tener contacto directo con el técnico o persona responsable.

También se espera el economizar con el ahorro de papel y tiempo de atención, con esto prestar un mejor servicio al cliente.

1.4 SITUACIÓN ACTUAL

La situación en la que actualmente se encuentra CompuPAQ es la de querer mejorar su control Administrativo dentro de la organización ya que con esto pretende tener una ventaja competitiva y dar un servicio de calidad al cliente.

El organigrama de la empresa de CompuPAQ está estructurado de la siguiente manera (Figura 1.1) ¹:

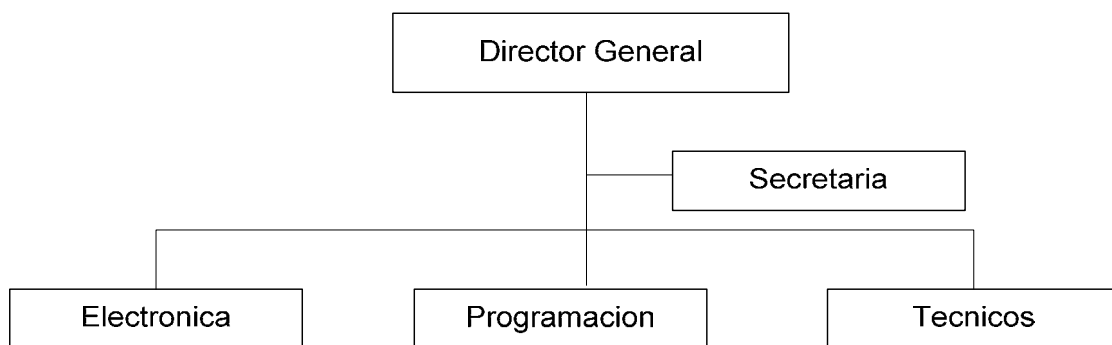


Figura 1.1 Organigrama de la empresa

1.5 SISTEMAS DE INFORMACIÓN

¹ Figura realizada por el autor

1.5.1 INTRODUCCIÓN

Desde los años setenta, los sistemas de bases de datos han ido reemplazando a los sistemas de ficheros en los sistemas de información de las empresas. Al mismo tiempo, se ha ido reconociendo la gran importancia que tienen los datos

que éstas manejan, convirtiéndose en uno de sus recursos más importantes.

Esto ha hecho que muchas empresas tengan departamentos que se encarguen de gestionar toda su información, que estará almacenada en una base de datos. Aparecen los papeles de *administrador de datos* y *administrador de la base de datos*, que son las personas encargadas de supervisar y controlar todas las actividades relacionadas con los datos de la empresa y con el ciclo de vida de las aplicaciones de bases de datos, respectivamente.

Un sistema de información (SI) es el sistema de personas, datos y actividades que procesan los datos y la información con el fin de apoyar u optimizar el desarrollo en una organización. Tales procesos consisten en la recolección de datos, agruparlos, analizarlos y dar a conocer los resultados con el propósito de realizar actividades de control y la toma de decisiones.

Los sistemas de información usan como herramientas los equipo de computación o cualquier componente que pueda enviar o recibir información a la computadora tales como las miniprintes, código de barras, detector de huellas, etc. Un sistema de información realiza cuatro actividades básicas que son: entrada, almacenamiento, procesamiento y salida de información.

Entrada de procesamiento: Es el proceso en el cual el sistema toma la información para el procesamiento de esta. Las entradas pueden ser de manera manual o automática, la manual requiere intervención directa del usuario y la automática proviene o son tomados de otros sistemas o módulos.

Almacenamiento de información: Esta es la parte más importante ya que esta propiedad del sistema permite guardar la información, suele tener varios medios de almacenamiento tales como discos magnéticos, discos duros, unidades de CD-ROM.

Procesamiento de información: Es la capacidad del sistema para efectuar una serie de cálculos con una secuencia de operaciones. Estos datos pueden ser tomados de información anterior o reciente y esta característica permite la

transformación de datos en información que puede ser utilizada para la toma de decisiones.

Salida de información: Consiste en sacar la información recolectada al exterior del sistema, las unidades típicas de salida son las impresoras, terminales, diskettes, cintas magnéticas entre otros.

Un sistema de información está formado por los siguientes componentes:

- La base de datos.
- El SGBD (Sistema Gestión de Base de Datos).
- Los programas de aplicación.
- Los dispositivos físicos (ordenadores, dispositivos de almacenamiento, etc.).
- El personal que utiliza y que desarrolla el sistema.

1.5.2 TIPOS DE SISTEMAS DE INFORMACIÓN

Desde hace tiempo los sistemas de información han estado cumpliendo los objetivos básicos (Ver Figura 1.2) que se necesitan en una organización los cuales son:

1.- Automatización de procesos operativos

2.- Almacena la información de manera clara y precisa para la toma de decisiones².

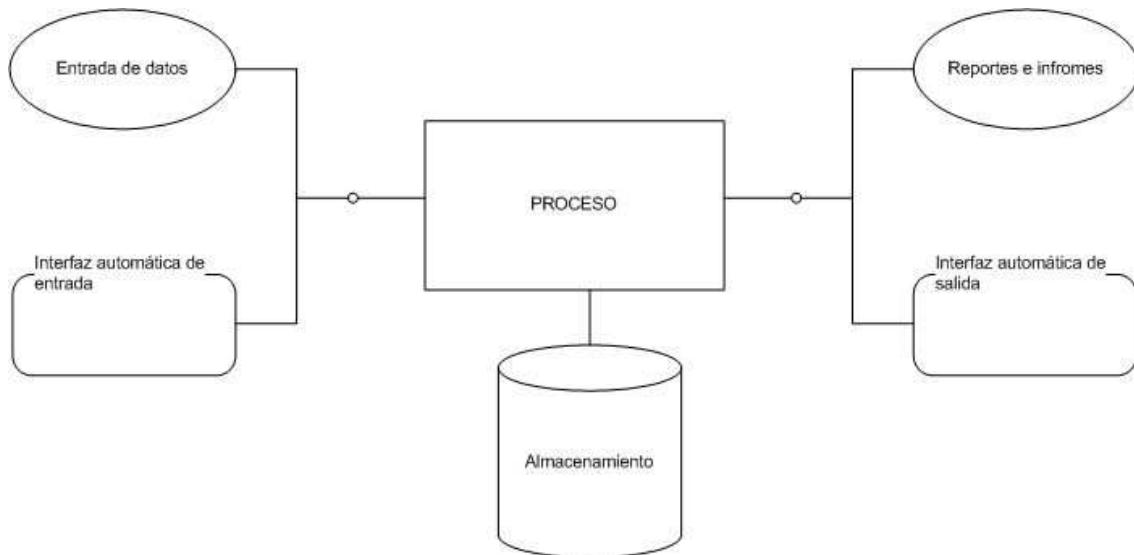


Figura 1.2 Proceso en los sistemas de información

3.- Obtener ventajas competitivas a través de su implantación y uso.

Los sistemas de información que logra la automatización de procesos en una organización tales como pagos, cobros, entradas, salidas, etc. Son llamados **SISTEMAS TRANSACCIONALES** en los que sus principales características son:

- Ahorro significativo de mano de obra.
² Figura realizada por el autor
- Con frecuencia son el primer tipo de sistema que se implantan en la organización.
- Son usados con mucha frecuencia en la entrada y salida de información: Sus cálculos son simples.
- Tienen la propiedad de ser recolectores de información.

El segundo tipo son los **SISTEMAS DE SOPORTE DE TOMA DE DECISIONES** son los sistemas que procesan la información y ayudan a la toma de decisiones (es el proceso en el cual la persona debe escoger entre dos o más alternativas) las características son:

- Suelen introducirse después de haberse implantado los sistemas transaccionales.
- Suelen ser intensivos en cálculos y escasos en entradas y salidas de información.
- Suelen ser sistemas interactivos y amigables, con altos estándares de diseño gráfico y visual.
- Por su misma naturaleza suelen ser repetitivos y de decisiones no estructuradas que no se repiten.
- Este tipo de sistemas puede incluir la programación de la producción, compra de materiales, flujo de fondos, proyecciones financieras, modelos de simulación de negocios, modelos de inventarios, etc.

El tercer tipo de sistema, 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.

- Su función primordial no es apoyar la automatización de procesos operativos ni proporcionar información para apoyar la toma de decisiones.
- Suelen desarrollarse in opuse (dentro de la organización).
- Su función es lograr ventajas que las competencias no posean.
- Apoyan el proceso de innovación de productos y procesos dentro de la empresa.

1.5.3 EVOLUCIÓN DE LOS SISTEMAS DE INFORMACIÓN

En la década de los setenta, Richard Nolan, un conocido autor y profesor de la escuela de Negocios de Harvard, desarrolló una teoría que impactó el proceso de planeación de los recursos y las actividades de la informática.

Según Nolan, la función de la Informática en las organizaciones evoluciona a través de ciertas etapas de crecimiento, las cuales se explican a continuación:

- Etapa inicial: Comienza con la adquisición de la primera computadora y normalmente se justifica por el ahorro de mano de obra y el exceso de papeles.
- Las aplicaciones típicas que se implantan son los Sistemas Transaccionales tales como nominas o contabilidad.
- El pequeño Departamento de Sistemas depende en la mayoría de los casos del área de contabilidad.
- El tipo de administración empleada es escaso y la función de los sistemas suele ser manejada por un administrador.
- El personal que labora en este pequeño departamento consta a lo sumo de un operador y/o un programador.
- En esta etapa es importante estar consciente de la resistencia de cambio del personal y usuario que están involucrados en los primeros sistemas que se desarrollan, ya que estos sistemas son importantes en el ahorro de mano de obra.
- Esta etapa termina con la implantación exitosa del primer Sistema de Información.

Etapa de contagio o expansión. Los aspectos sobresalientes que permiten diagnosticar rápido que una empresa se encuentra en esta etapa son:

- Se inicia con la implantación exitosa del primer Sistema de Información en la organización.
- Las aplicaciones que con frecuencia se implantan en esta etapa son el resto de los Sistemas Transaccionales no desarrollados en la etapa de inicio, tales como facturación, inventarios, control de pedidos de clientes y proveedores,

cheques, etc.

- El tipo de administración empleado está orientado hacia la venta de aplicaciones a todos los usuarios de la organización; en este punto suele contratarse a un especialista de la función con preparación académica en el área de sistemas.
- Se inicia la contratación de personal especializado y nacen puestos tales como analista de sistemas, analista-programador, programador de sistemas, jefe de desarrollo, jefe de soporte técnico, etc.
- Las aplicaciones desarrolladas carecen de interfaces automáticas entre ellas, de tal forma que las salidas que produce un sistema se tienen que alimentar en forma manual a otro sistema, con la consecuente irritación de los usuarios.

Etapa de administración de datos. Entre las características que destacan en esta etapa están las siguientes:

- El departamento de Sistemas de Información reconoce que la información es un recurso muy valioso que debe estar accesible para todos los usuarios.
- Para poder cumplir con lo anterior resulta necesario administrar los datos en forma apropiada, es decir, almacenarlos y mantenerlos en forma adecuada para que los usuarios puedan utilizar y compartir este recurso.
- El usuario de la información adquiere la responsabilidad de la integridad de la misma y debe manejar niveles de acceso diferentes.

CAPÍTULO II

Metodología usada para la implementación de un SI

2.1 CICLO EN ESPIRAL PARA EL DESARROLLO DE SISTEMAS

La metodología empleada para el desarrollo de presente proyecto fue el del ciclo en espiral, la cual establece y define con claridad los siguientes aspectos.

- El alcance del ciclo dependiendo de hasta dónde llegue el proyecto correspondiente. Un proyecto puede comprender un simple estudio de viabilidad del desarrollo de un producto, o su desarrollo completo o, llevando la cosa al extremo, toda la historia del producto con su desarrollo, fabricación, y modificaciones posteriores hasta su retirada del mercado.
- Las características (contenidos) de las fases en que dividen el ciclo. Esto puede depender del propio tema al que se refiere el proyecto (no son lo mismo las tareas que deben realizarse para proyectar un avión que un puente), o de la organización (interés de reflejar en la división en fases aspectos de la división interna o externa del trabajo).
- La estructura de la sucesión de las fases que puede ser lineal, con prototipado, o en espiral. Veámoslo con más detalle:

Al ciclo de vida de los sistemas de información también se le denomina ciclo de vida de desarrollo del software.

Boehm, autor de diversos artículos de ingeniería sobre modelos de estimación de esfuerzo y tiempo que se consume en hacer productos sobre los Modelos de Ciclo de Vida, promulgó un modelo desde un enfoque distinto al modelo tradicional en Cascada.

Su Modelo de Ciclo de Vida en Espiral tiene en cuenta el riesgo que aparece a la hora de desarrollar Software. Para ello, se comienza mirando las posibles

alternativas de desarrollo, se opta por la del riesgo más asumible y se hace un ciclo de la espiral. Si el futuro cliente quiere seguir haciendo mejoras en el Software se vuelve a evaluar las distintas nuevas alternativas y se realiza otra vuelta de la espiral hasta que llegue un momento en el que el producto Software desarrollado sea aceptado y no merezca la pena seguir mejorándolo con otro nuevo ciclo.

Este modelo fue propuesto por Boehm en 1988 puede considerarse como una generalización Ciclo de vida con prototipado para los casos en que no basta con una sola evaluación de un prototipo para asegurar la desaparición de incertidumbres y/o ignorancias. El propio producto a lo largo de su desarrollo puede así considerarse como una sucesión de prototipos que progresan hasta llegar a alcanzar el estado deseado. En cada ciclo (espirales) las especificaciones del producto se van resolviendo paulatinamente.

A menudo la fuente de incertidumbres es el propio cliente, que aunque sepa en términos generales lo que quiere, no es capaz de definirlo en todos sus aspectos sin ver como unos influyen en otros. En estos casos la evaluación de los resultados por el cliente no puede esperar a la entrega final y puede ser necesaria repetidas veces.

El esquema del ciclo de vida (Ver Fig. 2.1) para estos casos puede representarse por un bucle en espiral, donde los cuadrantes son, habitualmente, fases de especificación, diseño, realización y evaluación³.

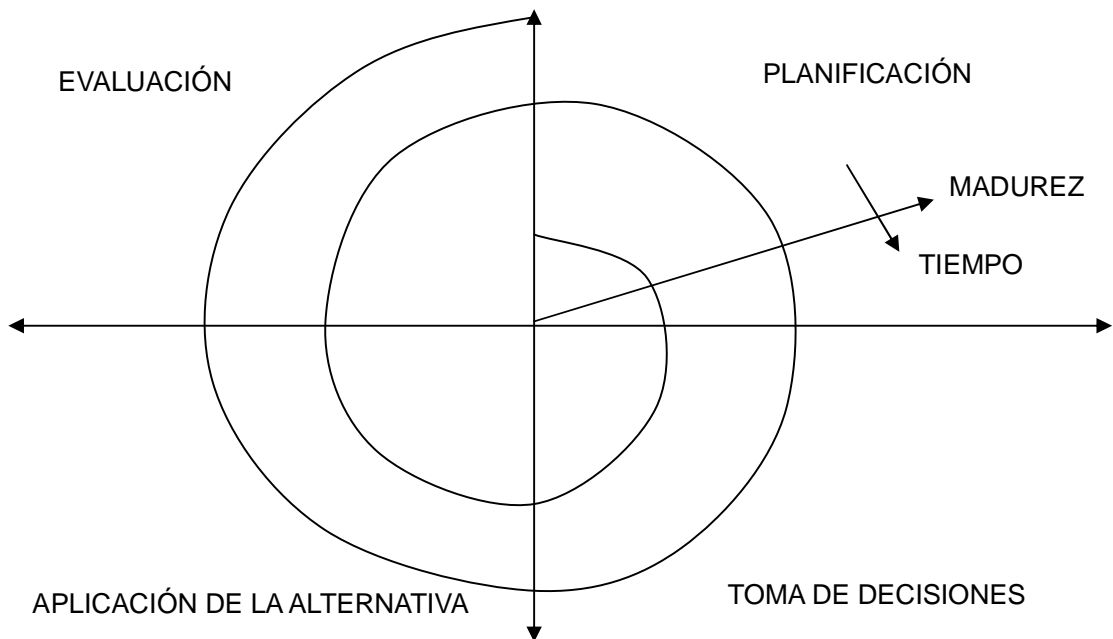


Figura 2.1 Ciclo de vida en espiral

En cada vuelta hay que tener en cuenta:

1. Los Objetivos: ¿Qué necesidad debe cubrir el programa?
2. Alternativas: Las diferentes formas de conseguir los objetivos de forma exitosa, desde diferentes puntos de vista como pueden ser:
3. Características: experiencia del personal, requisitos a cumplir.
4. Formas de gestión del programa.
5. Riesgo asumido con cada alternativa.
6. Desarrollar y Verificar: Programar y probar el programa.

³ Boehm, B., "A Spiral Model For Software Development And Enhancement", 1988 p.p 15

Si el resultado no es el adecuado o se necesita implementar mejoras o funcionalidades. Se planificarán los siguientes pasos y se volverá a empezar la espiral. La espiral tiene una forma de caracola y se dice que mantiene dos dimensiones la radial y la angular:

Angular=Avance del proyecto software dentro de un ciclo.

Radial=Aumento del coste del proyecto, ya que con cada nueva iteración se pasa más tiempo desarrollando.

Este sistema es muy utilizado en proyectos largos y que necesitan constantes cambios. Al ser un modelo de Ciclo de Vida orientado al riesgo se dice que uno de los aspectos fundamentales de su éxito radica en que el equipo que lo aplique sea capaz de detectar y catalogar correctamente dicho riesgo.

2.1.1 OBJETIVOS DE CADA FASE

Dentro de cada fase general de un modelo de ciclo de vida, se pueden establecer una serie de objetivos y tareas que lo caracterizan.

Fase de definición (¿qué hacer?)

- Estudio de viabilidad.
- Conocer los requisitos que debe satisfacer el sistema (funciones y limitaciones de contexto).
- Asegurar que los requisitos son alcanzables.
- Formalizar el acuerdo con los usuarios.
- Realizar una planificación detallada.

Fase de diseño (¿cómo hacerlo? Soluciones en coste, tiempo y calidad)

- Identificar soluciones tecnológicas para cada una de las funciones del sistema.
- Asignar recursos materiales para cada una de las funciones.
- Proponer (identificar y seleccionar) subcontratas.
- Establecer métodos de validación del diseño.
- Ajustar las especificaciones del producto.

Fase de construcción

- Generar el producto o servicio pretendido con el proyecto.
- Integrar los elementos subcontratados o adquiridos externamente.
- Validar que el producto obtenido satisface los requisitos de diseño previamente definidos y realizar, si es necesario, los ajustes necesarios en dicho diseño para corregir posibles lagunas, errores o inconsistencias.

Fase de mantenimiento y operación

- Operación: asegurar que el uso del proyecto es el pretendido.
- Mantenimiento (nos referimos a un mantenimiento no habitual, es decir, aquel que no se limita a reparar averías o desgastes habituales -este es el caso del mantenimiento en productos software, ya que en un programa no cabe hablar de averías o de desgaste).

2.2 HERRAMIENTAS DE ANÁLISIS Y DISEÑO DE SISTEMAS

En el desarrollo de sistemas de información el uso de herramientas de análisis y diseño de sistemas es de uso cotidiano, a continuación se detallan algunas de las más usadas:

2.2.1 PLANTEAMIENTO DE OBJETIVOS

El analista de sistemas debe determinar cuáles son los objetivos que se persiguen, para lo cual es preciso identificar las necesidades de los usuarios. Dichos objetivos deben ser claros, precisos y concisos, pues de éstos dependerá en gran medida el éxito del producto final, ya que si éstos no son los adecuados o se prestan a malas interpretaciones el producto final no cubrirá las necesidades de los usuarios y en vano será todo el trabajo.

2.2.2 ENTREVISTA CON LOS USUARIOS

La herramienta más importante en la etapa de planteamiento de objetivos es la entrevista con los usuarios, pues por medio de esta el analista de sistemas se da cuenta de cómo operan las cosas actualmente y como les gustaría a los usuarios que operaran en el futuro.

La forma más atinada para la realización de entrevistas es aplicarlas de arriba hacia abajo, es decir, comenzar por los niveles gerenciales y terminar con los trabajadores operacionales que participen en el sistema que se está estudiando.

A continuación se nombran una serie de aspectos que hay que tomar en cuenta si se desea tener éxito en una entrevista:

a) Antes de realizar cualquier entrevista es necesario solicitar autorización para

la realización de la misma, esto se debe hacer ante el jefe inmediato superior de la persona a ser entrevistada, y de ser posible que sea éste quien presente al analista con su subordinado.

- b) Toda entrevista debe ser planeada con anterioridad con el fin de tener bien claro cuáles son los objetivos que se persiguen con ésta, es decir, realizar un esbozo de las preguntas que se van a hacer al entrevistado, así como el enfoque que se le va a dar a cada una de ellas.

En realidad, no siempre es necesario determinar con anterioridad las preguntas que se van a plantear, pues a veces el analista decide que es preferible realizar una entrevista menos formal en la que solamente se vaya dirigiendo al entrevistado hacia el tema o los temas que se desean conocer.

La decisión de realizar una entrevista bien planeada o una menos formal se toma basándose en la labor que el entrevistado realiza dentro de la organización y la información que se desea obtener.

- c) Un punto importante para el éxito de la entrevista es buscar un horario oportuno para que el entrevistado no se esté distraendo, conteste todas las preguntas rápidamente sin analizarlas por tener trabajo pendiente, o se presente cualquier otra situación por la que no preste la atención que se necesita. Por lo anterior se recomienda hablar antes con el entrevistado y darle la oportunidad de determinar el día y la hora en que se realizará la entrevista, aclarándole que necesitamos de su completa atención.
- d) El analista de sistemas debe causar buena impresión al entrevistado, debe ser cortés, nunca prepotente, debe transmitir confianza al entrevistado para que éste no sienta que se le está juzgando o fiscalizando; el analista debe presentarse vestido de una manera adecuada pues es difícil que alguien

acceda a cooperar y poner su confianza en una persona sucia y desaliñada.

- e) Se deben evitar las entrevistas largas, pues llega un momento en que el entrevistado se cansa y empieza a distraerse; por lo cual es preferible tener varias entrevistas cortas en lugar de una muy larga ⁴.
- f) Propiciar un ambiente adecuado de tal manera que la persona a entrevistar se sienta bien, para ello es necesario considerar ciertos aspectos tales como el dejar hablar al entrevistado, no interrumpirlo a menos de que se esté desviando demasiado del tema. El analista no debe suponer nada ni dar algo por hecho, pues el entrevistado se puede confundir o sentir mal. Hay que respetar a cualquier persona sea cual fuere su puesto dentro de la organización. Es imprescindible también tratar de hablarle a cada entrevistado en su lenguaje, pues no podemos utilizar el mismo con un gerente que con el trabajador de menor nivel.
- g) El analista debe concretarse al tema, no empezar a tratar asuntos que no tienen nada que ver con el estudio del sistema.
- h) El analista debe tener despiertos sus cinco sentidos, debe estar pendiente de lo que pasa alrededor y del lenguaje corporal del entrevistado, pues en muchas ocasiones estos aspectos revelan cosas importantes.
- i) Inmediatamente después de terminada la entrevista ésta se debe transcribir y documentar, incluso se pueden elaborar diagramas que contribuyan a una mejor comprensión de lo investigado.

Una grabación de la entrevista es de gran ayuda para la documentación, pero ésta se debe hacer sólo con el consentimiento del entrevistado. Un resumen y auto evaluación podría servir de gran ayuda para entrevistas posteriores.

⁴ Jerry Fitzgerald, Stallings Warren. "Fundamentos de Análisis de Sistemas". México, Continental 1989 p.110.

Preguntas en la entrevista

Las preguntas de la entrevista deben estar encaminadas a conocer perfectamente todas las actividades que se realizan en el sistema, así como las entradas requeridas y las salidas que éste produce; y por supuesto determinar cuáles son los cambios que necesita el actual sistema. A continuación se presentan algunas preguntas que no deben faltar en una entrevista bien planeada.

El analista deber agregar aquéllas que considere necesarias. Primero se aplican las preguntas que ayudan al conocimiento del sistema actual.

Generales

- ¿Qué actividades se realizan?
- ¿Quién lo hace?
- ¿Cómo lo hace?
- ¿Cuándo, dónde, cómo, por qué, para qué lo hace?
- ¿Cuánto dura cada una de estas actividades?
- ¿Qué políticas de decisión se siguen?
- ¿Qué costumbres se tienen en el sistema?
- ¿Cuáles son los controles con los que se cuenta?

Entradas

- ¿Qué entradas hay al sistema?
- ¿De dónde vienen?
- ¿Cuándo vienen?
- ¿En qué formato?
- ¿Cómo se procesan las entradas?
- ¿Qué control de entradas se tiene?

Salidas

- ¿Qué salidas hay?
- ¿De dónde salen?
- ¿Cuándo salen?
- ¿En qué formato salen?
- ¿A dónde van?
- ¿Qué control de salidas se tiene?

Equipo de procesamiento

- ¿Cuáles son las características del equipo?
- ¿Qué aplicaciones se corren en el equipo?
- ¿Es seguro el equipo?
- ¿Qué controles se tiene?
- ¿Hay integridad?
- ¿En qué ambiente se trabaja - red, independiente, etc.?
- ¿Qué capacidad tiene el equipo?

Una vez que se conoce cómo trabaja el sistema actual, se procede a investigar los posibles cambios al sistema, para lo cual proponemos las siguientes preguntas:

- ¿Cuáles son los problemas más comunes dentro del sistema?
- ¿Qué agregaría a todo el proceso?
- ¿Qué eliminaría?
- ¿Qué cambios le haría?

2.2.3 RENTABILIDAD DE LOS OBJETIVOS

Cuando se han terminado las entrevistas con los usuarios y se conocen las necesidades y opiniones de los usuarios, el analista de sistemas está en condiciones de plantear ya en forma escrita los objetivos del "nuevo sistema".

Una vez redactados los objetivos se debe llevar a cabo un estudio de su rentabilidad, el cual debe ser revisado y aprobado por los altos mandos de la empresa, quienes decidirán si se realizará o no el desarrollo del sistema. Esto con el objeto de tener la seguridad de lo que se está haciendo y por qué se está haciendo, pues sería frustrante que en etapas posteriores, o incluso una vez terminado el desarrollo, nos diéramos cuenta de que dicho desarrollo no proporciona ningún beneficio económico a la empresa, pues bien sabemos que el objetivo principal de cualquier empresa es "ganar dinero".

Si se tuvieran subprogramas que dependan de uno de los programas del segundo nivel, se anotarían abajo de éste en el siguiente nivel, y así sucesivamente.

Cuando el analista de sistemas ha terminado la elaboración de árboles Modulares ya está en condiciones de realizar la "organización de Módulos".

2.2.4 ELABORACIÓN DE DIAGRAMAS DE FLUJO

Los diagramas de flujo son una herramienta muy útil en la etapa del Análisis. Para un sistema proponemos dos tipos de diagramas de flujo:

Diagramas de Flujo de Datos.

Diagramas de Flujo del Programa.

Este tipo de diagramas muestran cuáles son las entradas, salidas y procesos del sistema; así como los orígenes y destinos de los datos.

Es recomendable no hacer muy detallado ese tipo de diagrama, pues el objetivo principal es tener una visión general del sistema sin profundizar mucho en cada una de las operaciones que este realiza. Se puede elaborar un diagrama en el que se observe cómo está ubicado el sistema dentro de la organización y las relaciones que tiene con los demás sistemas en caso de que existan y otro en el que se muestre cómo intercalan los diferentes módulos que forman parte del sistema. Los diagramas de Flujo de Datos cuentan con cuatro tipos de gráficos que son explicados a continuación ⁵:



Figura 2.2 Gráficos usados para los diagramas de flujo de datos

a) Flujo de datos.- Son los datos que fluyen en las diferentes operaciones del sistema. Sobre la flecha debe tener el nombre con que se identifican los datos.

⁵ Senn James A., “Análisis y Diseño de Sistemas de Información”, México, Mc Graw Hill, 1990, p.118

- b) Procesos.- Representan la conversión de datos de entrada en datos de salida. Cada proceso debe tener un número y un nombre.
- c) Origen o destino externo de datos.- Representan personas u organismos que no interesan en el estudio del sistema, pero operan como un origen o destino de los datos.
- d) Archivos.- Son almacenes de datos.

Cuando una flecha de flujo de datos apunta hacia el símbolo de archivo, se indica que se está almacenando información en el archivo, pero si la flecha sale de él, es indicio de que se está obteniendo información del usuario.

A continuación en la Fig. 2.2 se muestra un ejemplo de un Diagrama de Flujo de Datos.

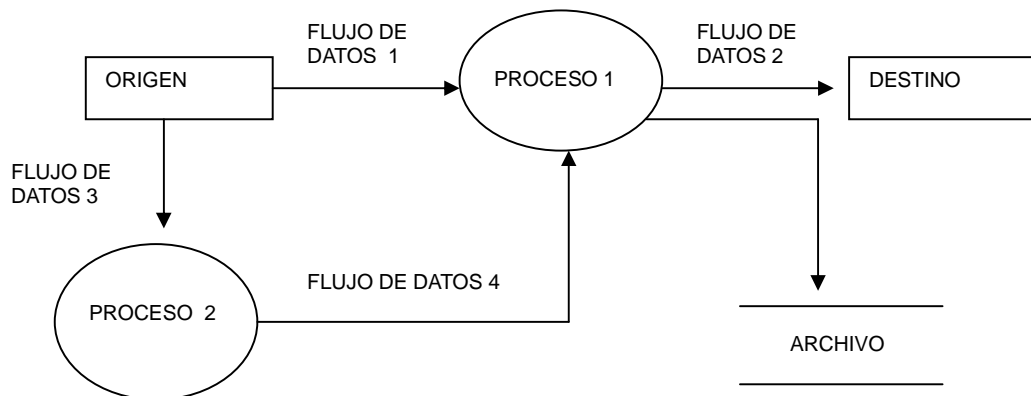


Figura 2.2 Ejemplo de un diagrama de flujo

⁶ Senn James A., “Análisis y Diseño de Sistemas de Información”, México, Mc Graw Hill, 1990, p.130

Diagrama de flujo

Uno de los problemas más grandes a los que uno se enfrenta cuando se tiene que hacer el desarrollo de un sistema es la dificultad para entender la lógica de programación con que se elaboró el sistema actual, pues con frecuencia ocurre que la persona que se encargó de la programación ya no labora en la organización.

Para poder realizar los cambios requeridos por el sistema es necesario que se comprenda perfectamente todo el proceso que se sigue en los programas; sin embargo no es recomendable que se realice un diagrama de flujo para cada uno de los módulos del sistema, pues habrá programas que por su facilidad de comprensión no será necesario representarlos gráficamente, y aunque lo ideal sería que existiera un diagrama de flujo para cada uno de ellos, habrá ocasiones en las que esto es casi imposible por las limitantes de tiempo que existen. Los símbolos más empleados en este tipo de diagramas son los siguientes:

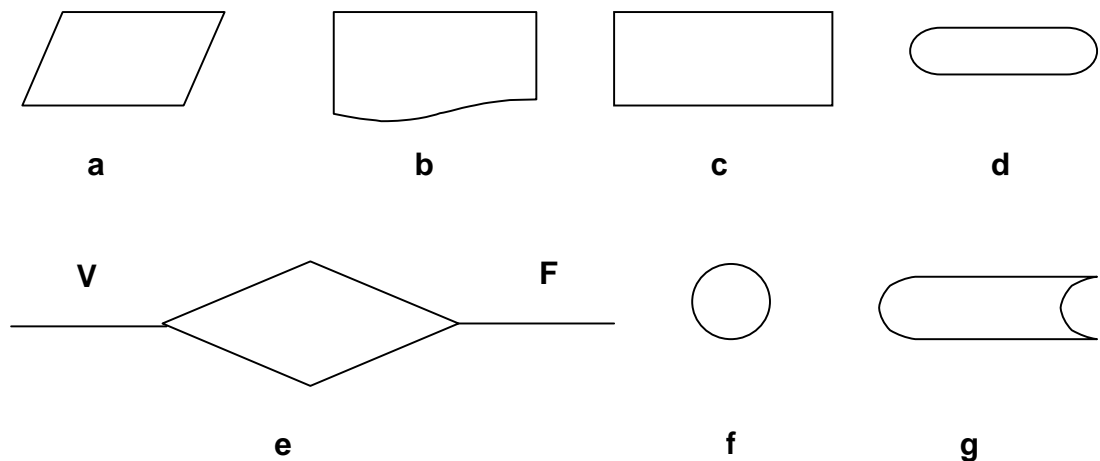


Figura 2.2 Símbolos usados en los diagramas de flujo

a) Entrada o salida de datos procesados o información.

⁷ Senn James A., "Análisis y Diseño de Sistemas de Información", México, Mc Graw Hill, 1990, p.132

- b) Entrada o salida en un formato impreso.
- c) Realización de una operación.
- d) Punto en que inicia o termina un programa.
- e) Punto en que se tiene que tomar una decisión a partir de una condición.
- f) Entrada o salida de otra parte del diagrama de flujo, generalmente cuando se cambia de página.
- g) Entrada o salida de datos en un archivo.

2.2.5 DETERMINACIÓN DE REQUERIMIENTOS

El objeto de la etapa de determinación de requerimientos es definir lo que el sistema debe ser capaz de realizar, para ello es necesario determinar cuáles serán las entradas, salidas, operaciones y recursos que necesitará el sistema para operar adecuadamente y cubrir las necesidades de la organización. No obstante debe quedar bien claro que en esta etapa aún no estamos rediseñando el sistema, sino determinando cuáles son los criterios generales de funcionamiento, mismos que nos ayudarán en el rediseño del sistema ⁸.

Para la determinación de requerimientos el analista debe basarse principalmente en los datos obtenidos en las etapas de Planteamiento de Objetivos y Análisis del Sistema, pues éstas nos dicen cuáles son las necesidades de los usuarios. Pero también se deben tomar muy en cuenta los planes futuros de la organización para lograr que el "nuevo sistema" se ajuste a dichos planes.

⁸ Víctor Gerez, "Desarrollo y administración de programas de computadora", México, Continental 1984 p.51

El analista de sistemas debe determinar los requerimientos del "nuevo sistema" en el siguiente orden:

- 1.- Salidas que debe producir el sistema, como son reportes, documentos, desplegados, etc.
- 2.- Entradas necesarias para producir las salidas esperadas, las cuales pueden ser tomadas de lentos fuentes o ser introducidas directamente al sistema.
- 3.- Todas las operaciones que debe realizar el sistema para producir las salidas esperadas.
- 4.- Los recursos que se necesitan para la operación del sistema, como son: hardware, software, recursos humanos, materiales y técnicos.

Es necesario que el analista determine primero cuáles serán las salidas que debe producir el sistema, y basándose en éstas podrá determinar cuáles son las entradas que se requerirán, qué operaciones deben llevarse a cabo y con qué recursos se debe contar para producir las salidas deseadas. Así como definir los controles con los que se debe contar. Existe una nomenclatura utilizada dentro del análisis de sistemas la cual consiste en especificar los documentos ya sean de entrada o salida con una letra antecediendo al requerimiento que sea para indicar el tipo de documento que se está utilizando. A continuación se especifican los valores y su significado.

- O : Original
- N : Nuevo
- C : Cambios
- B : Baja

Requerimiento 1:O: Reporte

Se requiere que en el sistema se capturen los datos del cliente y la falla del equipo, esta captura se hará en él y se complementará con los datos de la persona que da de alta el reporte como los datos del folio y fecha.

El formato de recibo de equipo es el que se muestra a continuación ⁹ en la Fig. 2.3.

FOLIO: 230 **FECHA:** 2008-4-20 **HORA:** 03:26:23 p.m.

ELABORO: JUAN GERARDO CARDENAS TRIGUEROS

DATOS DEL CLIENTE

CLIENTE: 1	NOMBRE: CLIENTE COMPUPAQ - USO INTERNO
DOMICLIO: HERMENEGILDO GALEANA NO. 346	COLONIA: CENTRO
CIUDAD: CELAYA	ESTADO: GTO CP: 38000
TELEFONO: 461-616-04-29	CONTACTO:

DATOS DEL EQUIPO

TIPO DE EQUIPO: ADAPTADOR DE CORRIEN	MARCA: 3COM	MODELO:
NO. SERIE:		
OBSERVACIONES:		
FALLA:		

DATOS DEL SERVICIO

TIPO DE SERVICIO: GARANTIA
OBSERVACIONES:

POLÍTICAS DEL DEPTO. DE SOPORTE TÉCNICO:

- El diagnóstico de cada servicio tendrá un costo determinado por el tabulador correspondiente.
- El costo de la reparación será determinado por el depto. De soporte y cotizado al cliente por el depto. De ventas. En caso de que por cualquier motivo la reparación no se efectúe, se hará el cargo correspondiente de la revisión. En caso de que la reparación proceda, la revisión será gratuita.
- Una vez notificado el diagnóstico y/o reparación al cliente, deberá recoger su equipo en un plazo no mayor a 30 días naturales. A partir de ese momento se hará un cargo de \$15.00 m.n IVA incluido por cada semana de almacenaje.
- Si el cliente no atiende a la recolección de su equipo, pasados 6 meses después de la fecha de notificación, CompuPAQ podrá disponer del equipo para recuperar el costo de diagnóstico y/o reparación, perdiendo el cliente cualquier derecho de reclamar su equipo total o parcialmente.
- Para servicios urgentes se hará un cargo adicional determinado por el departamento de soporte técnico.
- Para servicios a domicilio se hará un cargo adicional determinado por el departamento de soporte técnico.
- Todo servicio tendrá una garantía de 30 días naturales en mano de obra.
- La garantía de las refacciones será la ofrecida por el fabricante.
- El departamento de soporte técnico determinará si es necesario un anticipo y cual sería su porcentaje.
- El departamento de soporte técnico determinará el tiempo requerido para el diagnóstico y la reparación.
- Para tramitar una garantía será necesario presentar el comprobante de compra.

FIRMA DE CONFORMIDAD DEL CLIENTE

Hermenegildo Galeana #346 - Zona Centro - C.P 38000 - Celaya, Gto - Tel/Fax 01(461) 616-04-29 y 018007010631

Figura 2.2 Formato del recibo de equipo

2.2.6 DICCIONARIO DE DATOS

El diccionario de datos contiene las características de las entidades y atributos, que definen la estructura de la Base de Datos (Meta - Base de Datos). El objetivo del diccionario de datos es:

- Apoyar el diseño de la Base de Datos
- Facilitar el control de cada una de las entidades y atributos que forman parte de la estructura de Base de Datos del Sistema.
- Controlar dinámicamente. la estructura de la interface al usuario, para las diferentes pantallas del sistema.

En la tabla 2.1 se muestra un ejemplo del diccionario de datos.

Ejemplo de diccionario de datos por entidad de la tabla de clientes

Tabla	Campo	Descripción	Nulos	Índice	Tipo de dato
Cliente	Id cliente	Identificador del cliente	No	PRIMARY	Smallint(5)
Cliente	Nombre	Nombre del cliente	Si		Varchar(128)
Cliente	Domicilio	Domicilio del cliente	Si		Varchar(200)
Cliente	Colonia	Colonia del cliente	Si		Varchar(64)
Cliente	Tel1	Teléfono del cliente	Si		Varchar(13)
Cliente	Tel2	Segundo teléfono del cliente	Si		Varchar(13)
Cliente	Fax	Fax del cliente	Si		Varchar(13)
Cliente	Correo	Correo del cliente	Si		Varchar(64)
Cliente	Ciudad	Ciudad en la que reside	Si		Varchar(64)
Cliente	Entidad _ federativa	Entidad federativa del cliente	Si	INDEX	Char (3)
Cliente	Rfc	Rfc del cliente	Si		Varchar(13)
Cliente	Facturable	Se marca que el cliente pide factura	Si		SET('SI', 'NO')
Cliente	Contacto _ ventas	Si es empresa con el encargado de ventas	Si		Varchar(64)
Cliente	Contacto _ pagos	Si es empresa con el encargado de ventas	Si		Varchar(64)
Cliente	Correo_contacto_ventas	Si es empresa con el encargado de ventas	Si		Varchar(64)
Cliente	Correo_contacto_pagos	Si es empresa con el encargado de ventas	Si		Varchar(64)
Cliente	Revisión y pagos		Si		Varchar(64)
Cliente	Acumulado _ ventas		No		Double
Cliente	Saldo		No		Double
Cliente	Crédito		No		Double
Cliente	Observaciones		No		Varchar(128)

Tabla 2.1 Diccionarios de datos de la tabla

2.27 MODELO ENTIDAD – RELACIÓN

El modelo Entidad – Relación fue propuesto por Peter Chen, en 1976, usado como el modelo sobre el cual se soporta el diseño de una base de datos. Permite crear un modelo de datos en términos de: Entidades, sus Atributos y Relaciones entre las entidades.

Componentes del Modelo Entidad Relación

- A) Entidades
- B) Atributos
- C) Relaciones

A) Entidades

- Son los objetos principales acerca de los cuales se almacena información.
- Son cosas de importancia o interés para un área de negocios o para un sistema que requiere del almacenamiento de datos.
- Ejemplos: Personas, Lugares, Cosas o Eventos de interés.

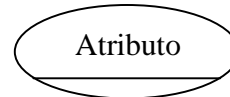
Nombre Entidad

B) Atributos.

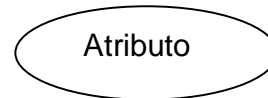
- Son las características de las entidades.
- Describen a las entidades.
- Representan características o cualidades de una entidad.

Ejemplos: Nombre de una persona, Nombres de ciudades, Número de empleado, Fecha de contratación, Monto a pagar.

Atributo identificador:
Identifican de manera única a cada
Ocurrencia de la entidad.



Descriptor:
Describen una característica o
cualidad de la Entidad.



C) Relaciones

Permiten representar diferentes tipos de relaciones entre entidades.

Tienen "semántica", es decir, almacenan información acerca de la forma en que se asocian las entidades.

Se dibujan así:



Modelación gráfica de datos.

El propósito de los diagramas es mostrar las Entidades de datos y cómo éstas se relacionan.

El diagrama entidad relación se concentra sólo en las entidades de datos para construir un diagrama entidad relación se mencionan los siguientes pasos.

Pregunta inicial: Identificación de entidades

¿Cuáles son las entidades de interés acerca de las cuales se desea almacenar datos?

Para un negocio comercial podrían ser: PRODUCTOS, INVENTARIO, PROVEEDORES, FACTURAS, ÓRDENES DE COMPRA.

Construcción.

- Dibujar un bloque para cada entidad identificada.
- Para el nombre de las entidades, se recomienda que sean en singular. Ejemplo: CLIENTE en lugar de CLIENTES, ARTICULO en lugar de ARTÍCULOS.

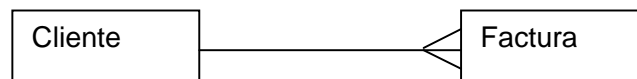
Segunda pregunta: Identificación de relaciones

¿Cuál relación existe entre cada par de entidades?

Para identificar las relaciones entre dos entidades se realizan dos preguntas:

- Primera pregunta de izquierda a derecha.
- Segunda pregunta de derecha a izquierda.

Ejemplo de Cliente - Factura



¿Un cliente cuantas facturas puede tener? Muchas.

¿Una factura cuantos clientes tiene? Una.

Después se tiene que analizar si la relación es obligatoria y se representa así: —+
Se revisa en ambos sentidos a través de dos preguntas.

¿Una factura puede existir sin un cliente? No, ya que es obligatorio tener un cliente.

¿Puede un cliente existir sin factura? Si, ya que no es obligatorio tener una factura.

Ejemplo de Producto - Factura



¿Un producto puede estar involucrado en muchas facturas? Sí.

¿Una factura cuantos productos puede tener? Muchos.

¿Una factura puede existir sin producto? Sí ya que no existe obligatoriedad.

¿Un producto puede existir sin una factura? Sí ya que no existe obligatoriedad.

En una relación de muchos a muchos se le debe asignar un nombre a la relación.

Tomando la relación entre la tabla PRODUCTO Y PROVEEDOR



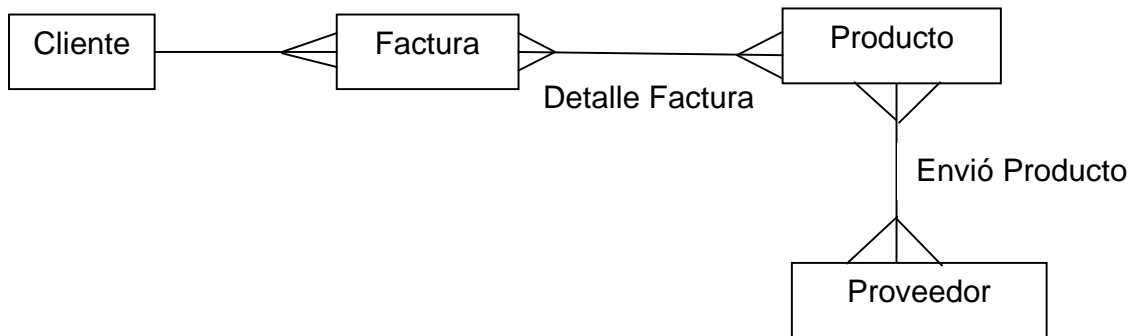
¿Un producto cuantos proveedores puede tener? Varios

¿Un proveedor cuantos productos puede tener? Muchos

¿Un proveedor puede existir sin producto? Sí ya que no tiene obligatoriedad
 ¿Un producto puede existir sin un proveedor? Sí ya que no tiene obligatoriedad

Se colocan las entidades en un solo diagrama:

Sistema completo



Para romper una relación de muchos a muchos se crea una tercera entidad con el nombre de la relación, esta tendrá una relación de uno a muchos con las entidades originales (el lado de muchos al lado de la nueva entidad y el lado de uno tendrá que ser obligatorio con las entidades originales).

Ejemplo:

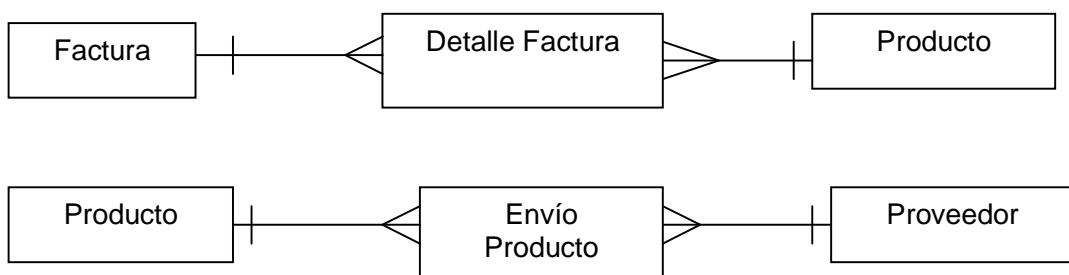
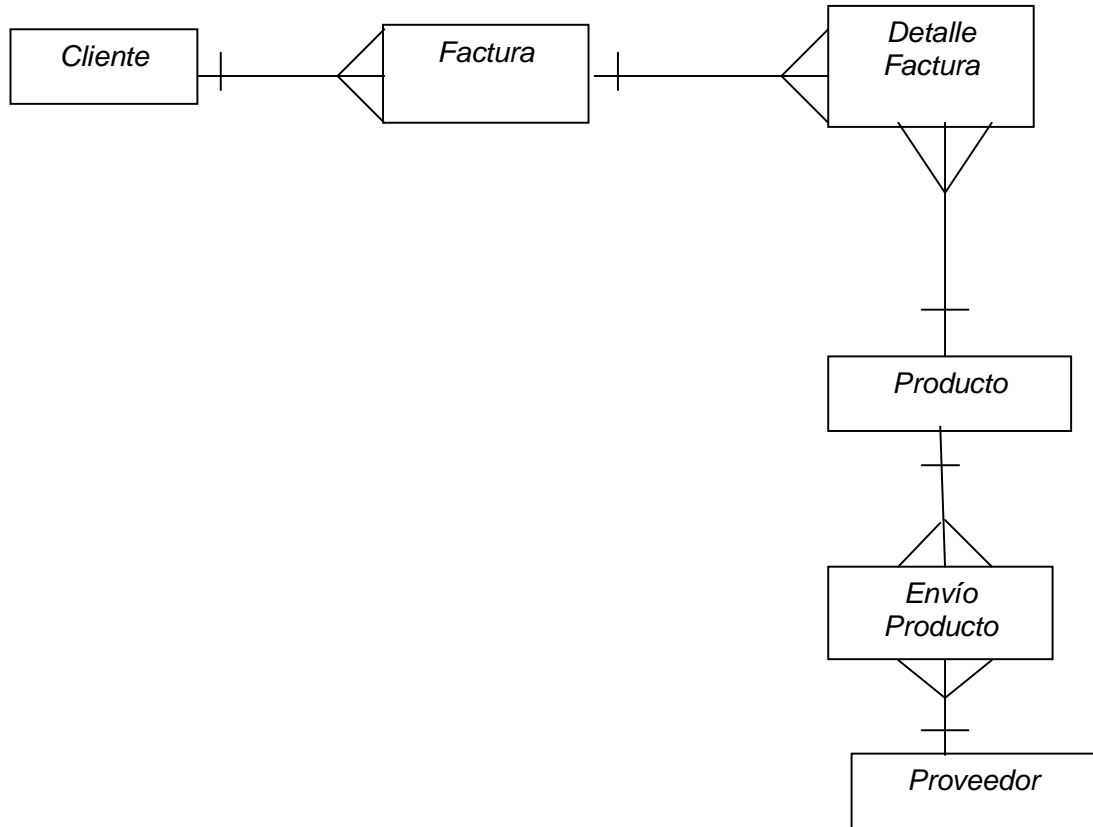


Diagrama completo



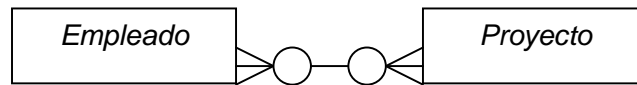
- Relación Opcional:

Dos entidades pueden estar relacionadas, pero no en todas sus ocurrencias.

Ejemplo: EMPLEADO y PROYECTO.

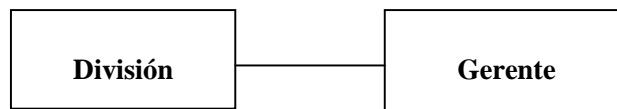
Un empleado puede estar asignado a un proyecto, a algunos proyectos o a ninguno. Un proyecto puede estar autorizado, y no tener empleados asignados

Representación:



- Relación Uno a Uno

Es cuando el atributo identificador de una entidad es el mismo de otra, a este tipo de relación se le llama de uno a uno.

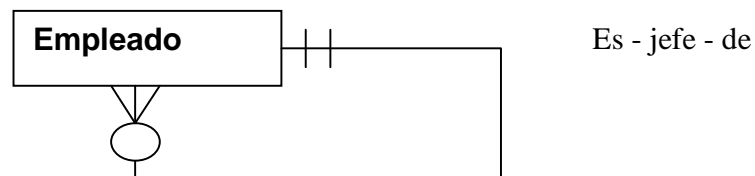


Una División es atendida por un solo Gerente y un Gerente atiende a una solo División.

- Relación Unitaria

Una entidad puede estar relacionada consigo misma.

Ejemplo:



Este diagrama entidad relación puede ser leído así:

“Un empleado puede ser jefe de cero o más empleados” o “Un empleado siempre reporta a otro empleado.”

Comentarios del Modelo Entidad - Relación

El modelo Entidad – Relación apoya al análisis en los siguientes aspectos.

- Definir los requerimientos de la empresa.
- Escribir la información acerca de las entidades y sus relaciones, requeridas para modelar esos requerimientos.
- Determinar los tipos de transacciones que se busca ejecutar sobre la Base de Datos.

El modelo Entidad - Relación apoya el diseño.

- “Mejora” la habilidad del diseñador de Base de Datos.
- Permite definir los requerimientos de información del mundo real de la manera precisa.
- Define la semántica de las relaciones entre los datos.

A continuación se muestra parte del modelo entidad relación de la base de datos (Ver Fig. 2.3) ¹⁰.

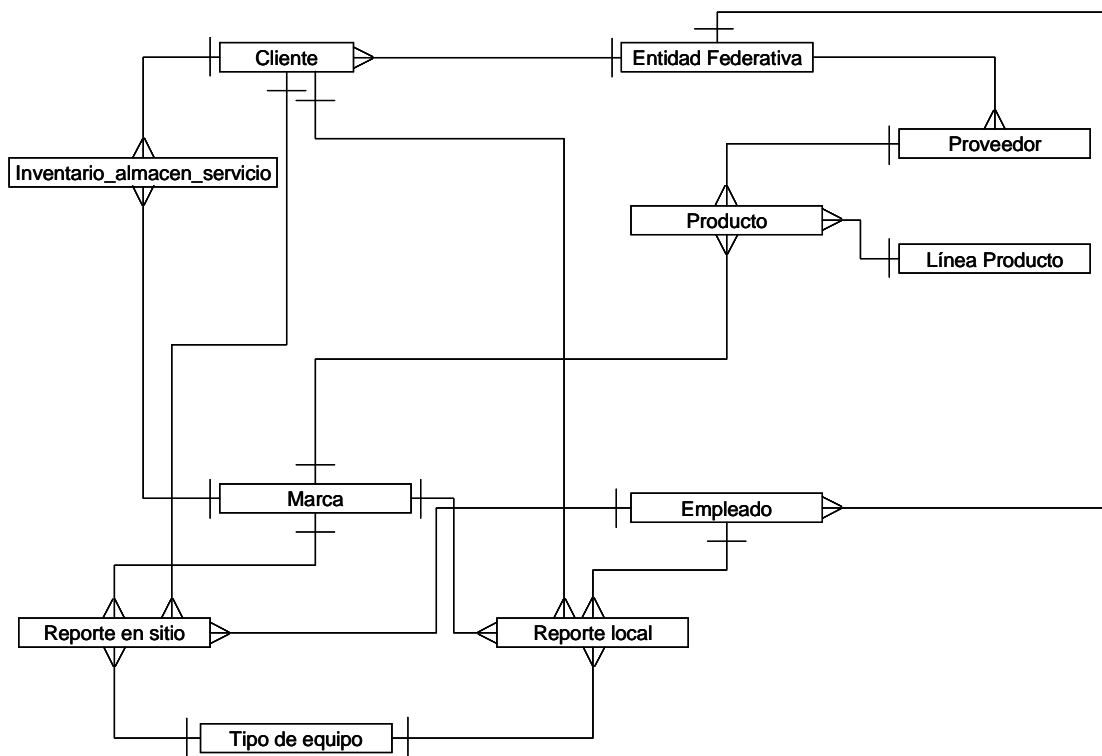


Figura 2.3 Diagrama entidad relación

¹⁰ Figura realizada por el autor

CAPÍTULO III

Herramientas de informática

3.1 INTRODUCCIÓN A LA PLATAFORMA .NET

3.1.1 ¿QUÉ ES .NET?

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

Las características principales que conforman .NET son las siguientes:

- La plataforma .NET Framework, proporciona la infraestructura para crear aplicaciones y el entorno de ejecución para las mismas.
- Los productos de Microsoft enfocados hacia .NET, entre los que se encuentran Windows .NET Server, como sistema operativo que incluirá de forma nativa la plataforma .NET Framework; Visual Studio .NET, como herramienta integrada para el desarrollo de aplicaciones; Office .NET, etc.
- Servicios para .NET desarrollados por terceros fabricantes, que podrán ser utilizados por otras aplicaciones que se ejecuten en Internet.

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

La Fig. 3.1 muestra un diagrama con la distribución de elementos dentro del entorno de .NET Framework ¹⁰.

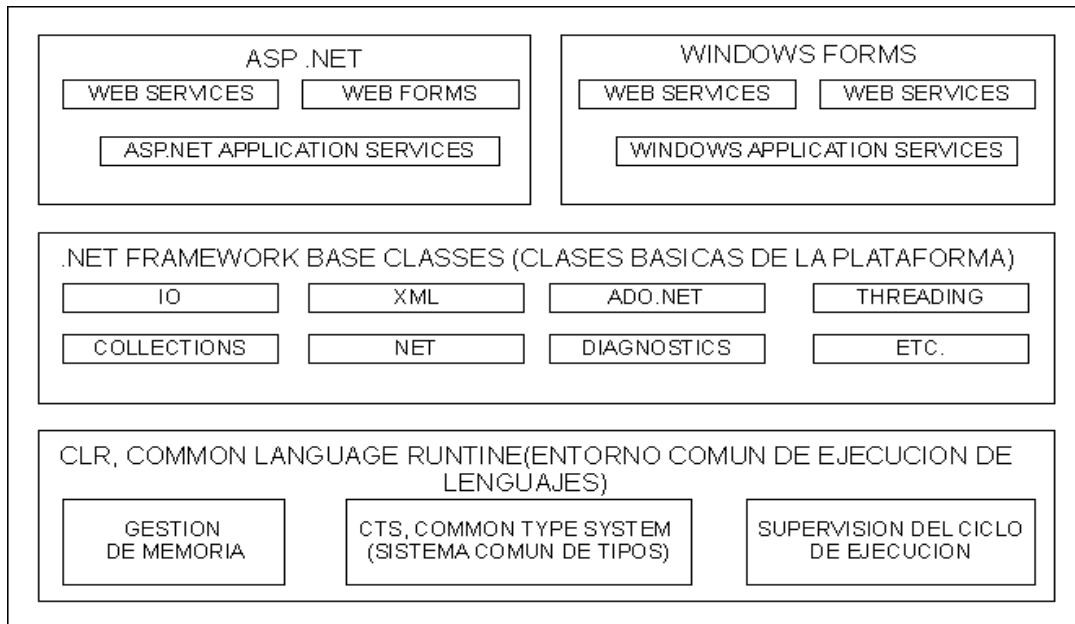


Figura 3.1 Elementos del entorno .Net Framework

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

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

Finalmente, en el nivel superior, encontramos las clases que permiten el diseño del interfaz de usuario de nuestras aplicaciones. Si necesitamos desarrollar aplicaciones para Internet, utilizaremos ASP.NET, que nos provee de todo lo necesario para crear aplicaciones para la Red: web forms, web services, etc.

3.1.2 EL CLR

¹⁰ Steven Holzner "Microsoft Visual Studio .net", 800 E. 96th St., Indianapolis, Indiana, 46240 USA, Sams p. 60

El Entorno de Ejecución Común de Lenguajes o CLR (Common Language Runtime), representa el alma de .NET Framework y es el encargado de la ejecución del código de las aplicaciones. A continuación se enumeran algunas de las características de este componente de la plataforma:

- Proporciona un desarrollo de aplicaciones más sencillo y rápido gracias a que gran parte de las funcionalidades que tradicionalmente debía de crear el programador, vienen implementadas en el entorno de ejecución.
- Administra el código en tiempo de ejecución, en todo lo referente a su carga, disposición en memoria, recuperación de memoria no utilizada a través de un recolector de memoria, etc.
- Implementa características de gestión a bajo nivel (administración de memoria, por ejemplo), que en ciertos lenguajes, eran labor del programador.
- Proporciona un sistema común de tipos para todos los lenguajes del entorno.
- Gestiona la seguridad del código que es ejecutado.
- Dispone de un diseño abierto a lenguajes y herramientas de desarrollo creadas por terceros fabricantes.
- Facilita enormemente la distribución e instalación de aplicaciones, ya que en teoría, es posible instalar una aplicación simplemente copiando los ficheros que la componen en uno de los directorios del equipo en el que se vaya a ejecutar, eliminando los temibles conflictos de versiones entre librerías, problema conocido también con el nombre de Infierno de las DLL o DLL Hell.

El la Fig. 3.2 se muestra el esquema en la que se encuentra formado el CLR ¹¹.

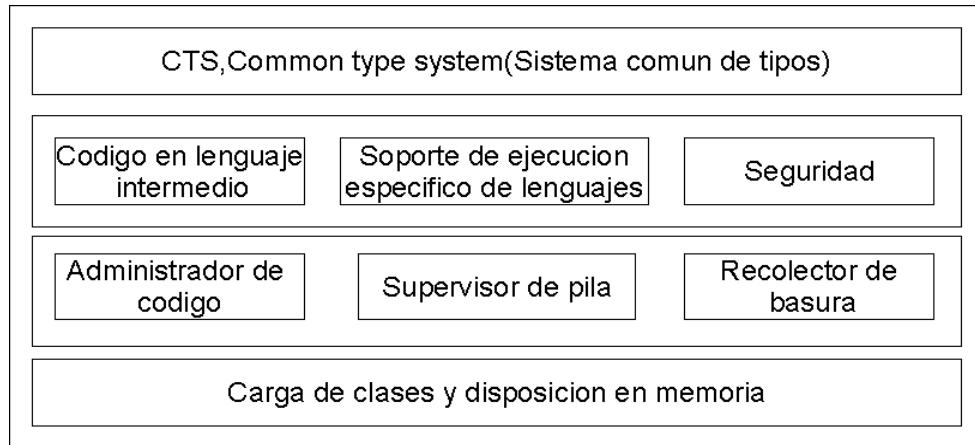


Figura 3.2 Muestra de un esquema de la organización interna del CLR

3.1.2 ENSAMBLADOS

Un ensamblado o assembly, consiste en un conjunto de tipos y recursos, reunidos para formar la unidad más elemental de código que puede ejecutar el entorno de .NET Framework.

De igual forma que los edificios se crean a base de la unión de un conjunto de materiales, dentro de la tecnología .NET, los ensamblados se presentan como los bloques de construcción software, que se unen o ensamblan para crear aplicaciones. Una aplicación desarrollada para .NET Framework debe estar compuesta por uno o varios ensamblados, ver Fig. 3.3 ¹².

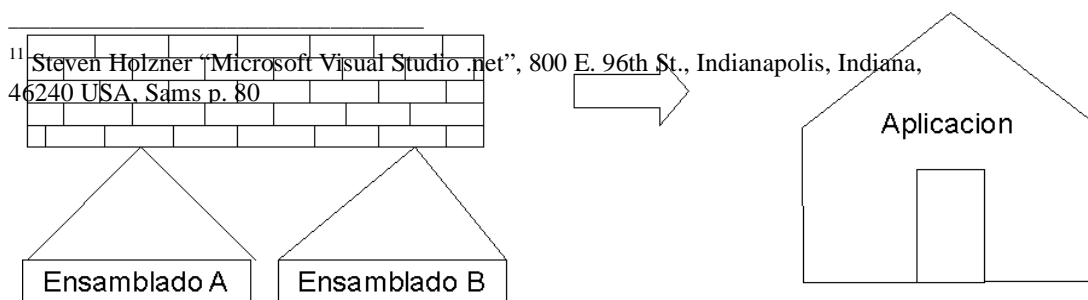


Figura 3.3 los ensamblados forman bloques de construcción de aplicaciones

Podemos establecer una analogía entre un ensamblado y una DLL, ya que ambos contienen clases, que se exponen a otras aplicaciones. Por dicho motivo, a un ensamblado también se le da el nombre de DLL lógica; el término DLL se emplea porque tiene un comportamiento similar al de las DLL's tradicionales, y el término lógica porque un ensamblado es un concepto abstracto, ya que se trata de una lista de ficheros que se referencian en tiempo de ejecución, pero que no se compilan para producir un fichero físico, a diferencia de lo que ocurre con las DLL's tradicionales.

Sin embargo, un ensamblado extiende sus funcionalidades a un horizonte mucho más amplio, ya que puede contener otros elementos aparte de clases, como son recursos, imágenes, etc.

Por otro lado, simplifican los tradicionales problemas de instalación y control de versiones sobre los programas, uno de los objetivos de la tecnología .NET, en la que en teoría, para instalar una aplicación, sólo sería necesario copiar los ficheros que la componen en un directorio de la máquina que la vaya a ejecutar.

Cuando creamos un nuevo proyecto en VB.NET desde Visual Studio .NET, dicho proyecto es ya un ensamblado, creado de forma implícita.

3.2 VISUAL STUDIO .NET

¹² Steven Holzner "Microsoft Visual Studio .net", 800 E. 96th St., Indianapolis, Indiana, 46240 USA, Sams p. 85

3.2.1 INTRODUCCIÓN

Visual Studio .NET es un conjunto completo de herramientas de desarrollo para la construcción de aplicaciones Web ASP, servicios Web XML, aplicaciones para escritorio y aplicaciones móviles. Visual Basic .NET, Visual C++ .NET, Visual C# .NET y Visual J# .NET utilizan el mismo entorno de desarrollo integrado (IDE), que les permite compartir herramientas y facilita la creación de soluciones en varios lenguajes.

Asimismo, dichos lenguajes aprovechan las funciones de .NET Framework, que ofrece acceso a tecnologías clave para simplificar el desarrollo de aplicaciones Web ASP y servicios Web XML.

3.2.2 ENTORNO INTELLISENSE

Cuando trabajamos con Visual Studio .Net, Trabajamos con un entorno intellisense, pero ¿Que es un entorno intellisense? Un entorno intellisense es aquel que permite a partir de un objeto determinado, expandir todas sus referencias, clases, objetos y propiedades que pueden ser utilizados junto al objeto que se esta trabajando.

Un ejemplo de intellisense en acción es el que se muestra en la fig. 3.4 ¹³.

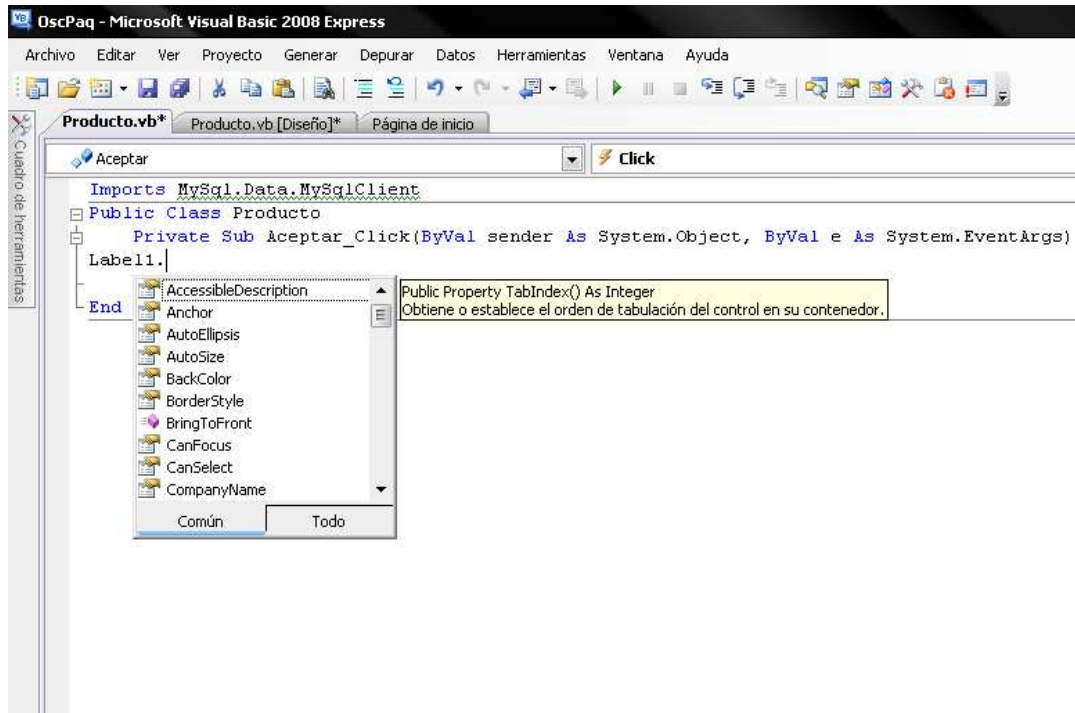


Figura 3.4 Ejemplo de IntelliSense en acción

Para habilitar o deshabilitar la herramienta intellisense, bastara con acudir al menú Herramientas > Opciones y dentro de este, a la propiedad Basic > General.

3.2.3 PROGRAMACIÓN ORIENTADA A OBJETOS

La Programación Orientada a Objetos (POO u OOP según sus siglas en inglés) es un paradigma de programación que usa objetos y sus interacciones para diseñar aplicaciones y programas de computadora. Está basado en varias técnicas, incluyendo herencia, modularidad, polimorfismo, y encapsulamiento. Su uso se popularizó a principios de la década de 1990. Actualmente son muchos los lenguajes de programación que soportan la orientación a objetos.

Los objetos son entidades que combinan estado, comportamiento e identidad:

El estado está compuesto de datos, serán uno o varios atributos a los que se habrán asignado unos valores concretos (datos).

¹³ Figura realizada por el autor

El comportamiento está definido por los procedimientos o métodos con que puede operar dicho objeto, es decir, qué operaciones se pueden realizar con él.

La identidad es una propiedad de un objeto que lo diferencia del resto, dicho con otras palabras, es su identificador.

La programación orientada a objetos expresa un programa como un conjunto de estos objetos, que colaboran entre ellos para realizar tareas. Esto permite hacer los programas y módulos más fáciles de escribir, mantener y reutilizar.

Los métodos (comportamiento) y atributos (estado) están estrechamente relacionados por la propiedad de conjunto. Esta propiedad destaca que una clase requiere de métodos para poder tratar los atributos con los que cuenta.

Esto difiere de la programación estructurada tradicional, en la que los datos y los procedimientos están separados y sin relación, ya que lo único que se busca es el procesamiento de unos datos de entrada para obtener otros de salida. La programación estructurada anima al programador a pensar sobre todo en términos de procedimientos o funciones, y en segundo lugar en las estructuras de datos que esos procedimientos manejan.

En la programación estructurada sólo se escriben funciones que procesan datos. Los programadores que emplean éste nuevo paradigma, en cambio, primero definen objetos para luego enviarles mensajes solicitándoles que realicen sus métodos por sí mismos.

3.2.3.1 CONCEPTOS FUNDAMENTALES

La programación orientada a objetos es una nueva forma de programar que trata de encontrar una solución a estos problemas. Introduce nuevos conceptos, que superan y amplían conceptos antiguos ya conocidos. Entre ellos destacan los siguientes:

Objeto: entidad provista de un conjunto de propiedades o atributos (datos) y de comportamiento o funcionalidad (métodos). Corresponden a los objetos reales del

mundo que nos rodea, o a objetos internos del sistema (del programa).

Clase: definiciones de las propiedades y comportamiento de un tipo de objeto concreto. La instanciación es la lectura de estas definiciones y la creación de un objeto a partir de ellas.

Método: algoritmo asociado a un objeto (o a una clase de objetos), cuya ejecución se desencadena tras la recepción de un "mensaje". Desde el punto de vista del comportamiento, es lo que el objeto puede hacer. Un método puede producir un cambio en las propiedades del objeto, o la generación de un "evento" con un nuevo mensaje para otro objeto del sistema.

Evento: un suceso en el sistema (tal como una interacción del usuario con la máquina, o un mensaje enviado por un objeto). El sistema maneja el evento enviando el mensaje adecuado al objeto pertinente. También se puede definir como evento, a la reacción que puede desencadenar un objeto, es decir la acción que genera.

Mensaje: una comunicación dirigida a un objeto, que le ordena que ejecute uno de sus métodos con ciertos parámetros asociados al evento que lo generó.

Propiedad o atributo: contenedor de un tipo de datos asociados a un objeto (o a una clase de objetos), que hace los datos visibles desde fuera del objeto y esto se define como sus características predeterminadas, y cuyo valor puede ser alterado por la ejecución de algún método.

3.2.3.2 CARACTERÍSTICAS DE LA POO

Hay un cierto desacuerdo sobre exactamente qué características de un método de programación o lenguaje le definen como "orientado a objetos", pero hay un consenso general en que las características siguientes son las más importantes:

Abstracción: Esta característica permite concentrarse en lo que hace, pero no en como lo hace. Mediante esta, es posible usar un objeto conociendo solamente las características que nos interesan del mismo, y manteniendo oculta la forma en la

cual lo hace. Lógicamente, esto nos brinda el gran beneficio de la facilidad.

Encapsulación: Brinda la posibilidad de esconder la implementación, y proveer el acceso a través de una apariencia estándar llamada interfaz. Una interfaz es una colección de métodos y propiedades, aunque en algunos lenguajes se incluye también la definición de eventos de la misma. De esta forma, la clase puede ofrecer una o varias de éstas para interactuar con sus funcionalidades, sin necesidad de exponer directamente su implementación. Esta característica mejora la integración entre aplicaciones, ya que se emplea una forma estándar de utilización.

Agregación: Hace posible que una propiedad de una clase puede contener otras clases. En general las capacidades de encapsulación y agregación trabajan en conjunto, ya que varias clases podrían estar incluidas dentro de una misma estructura.

Herencia: Una clase puede heredar las funcionalidades de otra ya existente, con el fin de ganar sus cualidades, y posteriormente adicionar, eliminar y hasta sobrescribir los miembros originales. En algunos lenguajes se permite que una clase pueda heredar las características de varias de ellas, y a esto se le denomina herencia.

Polimorfismo: El término significa (un objeto y muchas formas), y hace posible que un método de una clase (función o procedimiento) pueda tener diferentes comportamientos teniendo en cuenta el o los tipos de dato utilizados en su invocación. Una clase llamada Operaciones que contenga un método Suma y que acepte dos argumentos podría ejecutar implementaciones diferentes si es invocada con tipos de datos distintos.

3.3 INSTALACIÓN DE VISUAL STUDIO .NET

.NET Framework SDK

Se trata del kit de desarrollo de software para .NET Framework (Software Development Kit o SDK), que contiene la propia plataforma .NET y un conjunto de herramientas independientes, algunas funcionan en modo comando (en una ventana MS-DOS) y otras en modo gráfico. Los elementos imprescindibles para poder desarrollar aplicaciones para .NET están contenidos en este conjunto de herramientas.

Visual Studio .NET 2008

Es la nueva versión de la familia de herramientas de desarrollo de software de Microsoft, naturalmente orientadas hacia su nuevo entorno de programación: .NET Framework. Si bien es posible la escritura de programas empleando sólo el SDK de .NET Framework, este último, al estar compuesto de herramientas independientes, constituye un medio más incómodo de trabajo.

Requisitos hardware

Mínimo Recomendado

Procesador Pentium III – 733 MHz – Pentium IV – 1.6 GHz

Memoria 256 MB 512 MB

Espacio en disco duro 3 GB

Sistema operativo

VS.NET puede ser instalado en un equipo con uno los siguientes sistemas operativos:

Windows Vista

Windows Xp Service pack 2.0

Windows 2000 (se requiere tener instalado el Service Pack 2).

Windows NT 4.0. (se requiere tener instalado el Service Pack 5).

Para aprovechar todo el potencial de desarrollo de la plataforma, es recomendable usar como sistema operativo Windows XP, ya que ciertos aspectos del entorno (las características avanzadas de gestión gráfica por ejemplo) no están disponibles si instalamos .NET en otro sistema con menos prestaciones a continuación en la Fig. 3.5 se muestra el proceso de instalación de Visual Studio 2008 ¹⁴.



Figura 3.5 Menú de Instalación de VS 2008

¹⁴ Figura realizada por el autor

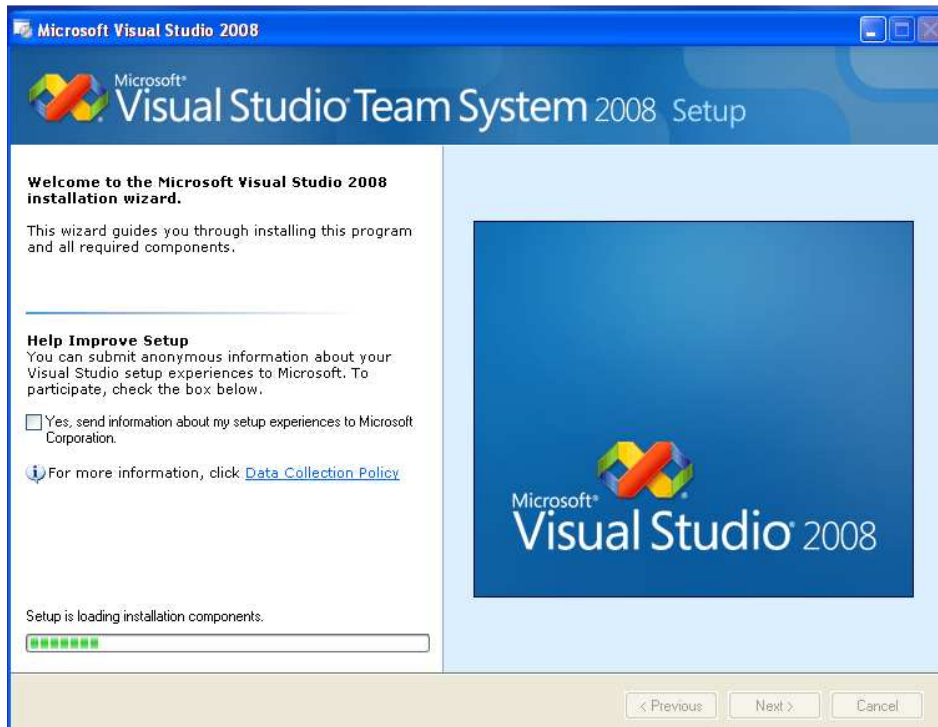


Fig. 3.6 Bienvenida de VS 2008 y preparación para la instalación

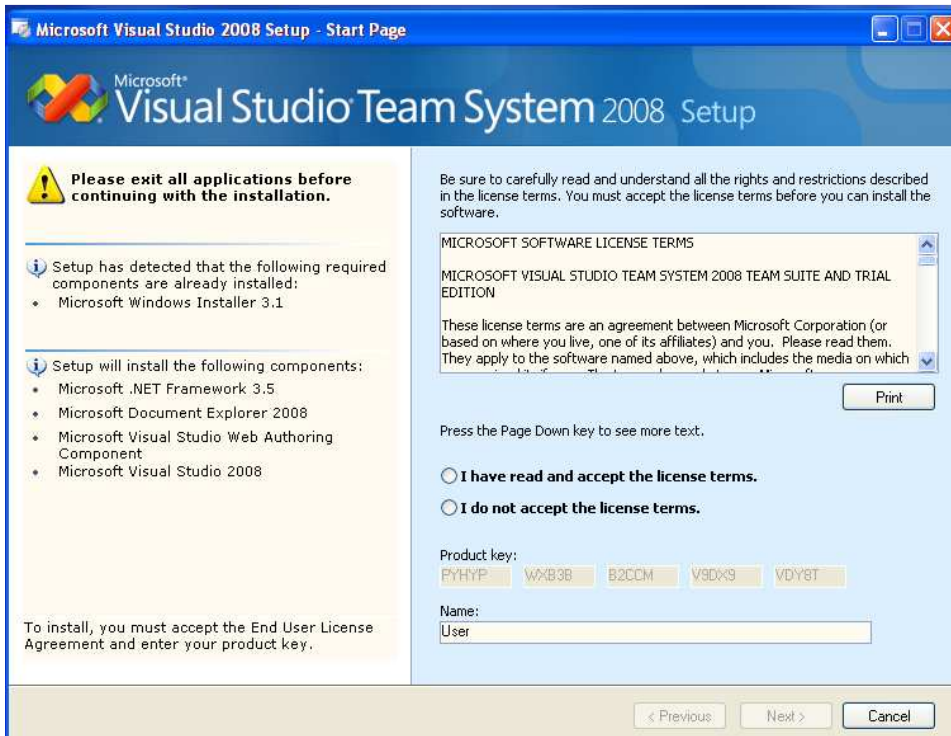


Fig. 3.7 Licencia del programa

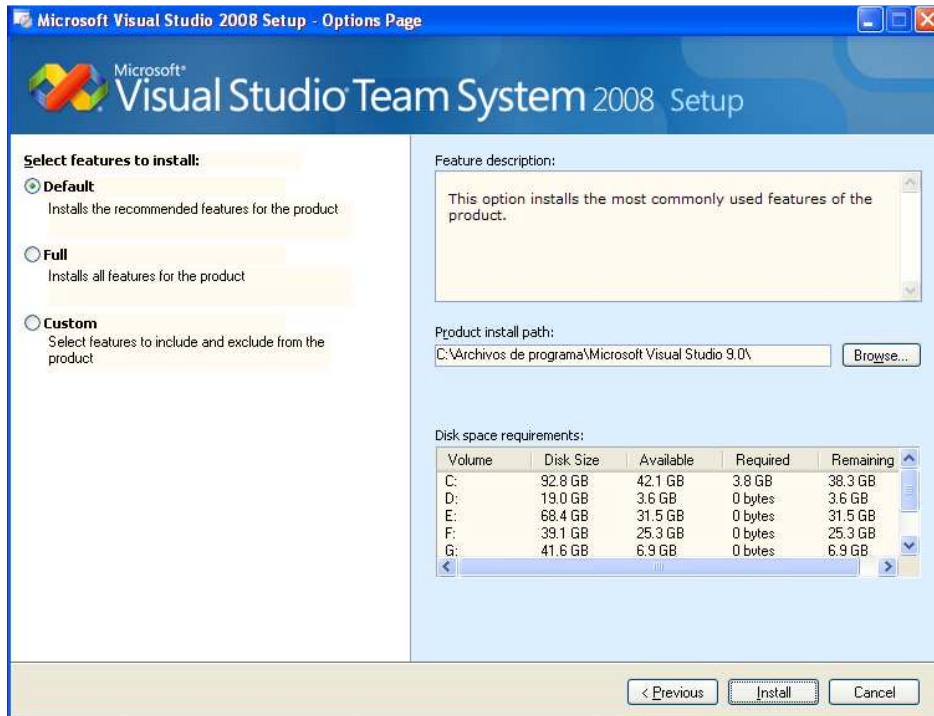


Fig. 3.8 Componentes a instalar y tamaño de disco

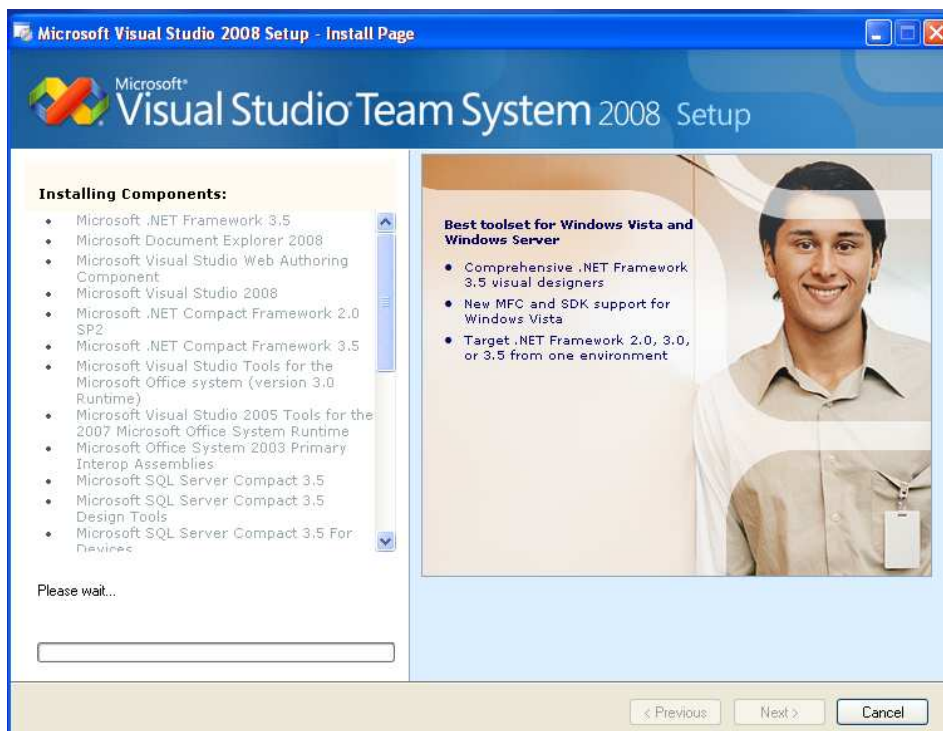


Fig. 3.9 Proceso de instalación

3.4 VISUAL BASIC

3.4.1 HISTORIA

Visual Basic es un lenguaje de programación desarrollado por Alan Cooper para Microsoft. El lenguaje de programación es un dialecto de BASIC, con importantes añadidos. Su primera versión fue presentada en 1991 con la intención de simplificar la programación utilizando un ambiente de desarrollo completamente gráfico que facilitara la creación de interfaces gráficas y en cierta medida también la programación misma. En 2001 Microsoft propone abandonar el desarrollo basado en la API Win32 y pasar a trabajar sobre un framework o marco común de librerías independiente de la versión del sistema operativo.

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

3.4.3 MÓDULOS Y CLASES

En Visual Basic .NET podemos crear clases de dos formas distintas, usando la instrucción `Module` o usando la instrucción `Class`, en ambos casos, a continuación de esas instrucciones, se indicará el nombre que tendrá ese elemento.

Tanto los módulos como las clases, deben estar declarados dentro de un espacio

de nombres. Dentro de una clase podemos definir otras clases, pero no podemos definir módulos dentro de otros módulos. La diferencia entre un módulo y una clase, es que un módulo define todos sus miembros como compartidos (Shared).

Cuando queremos crear un objeto basado en una clase, debemos usar New para crear una nueva instancia en la memoria, cada nuevo objeto creado con New será independiente de los otros que estén basados en esa clase. Por otro lado, para usar los elementos contenidos en un módulo, no necesitamos crear una nueva instancia, los usamos directamente y asunto arreglado; esto es así porque esos elementos están compartidos son estáticos, es decir, siempre existen en la memoria y por tanto no es necesario crear un nuevo objeto. Al estar siempre disponible, sólo existe una copia en la memoria.

3.4.4 PROCEDIMIENTOS

En toda aplicación de consola existe un procedimiento de tipo Sub llamado Main que es el que se utiliza como punto de entrada del ejecutable.

Los métodos de una clase pueden ser de dos tipos: Sub o Function.

Los procedimientos Sub son como las instrucciones o palabras clave de Visual Basic: realizan una tarea.

Los procedimientos Function además de realizar una tarea, devuelven un valor, el cual suele ser el resultado de la tarea que realizan. Debido a que las funciones devuelven un valor, esos valores se pueden usar para asignarlos a una variable además de poder usarlos en cualquier expresión. Para que quede clara la diferencia entre estos dos tipos de procedimientos, en el siguiente ejemplo se mostrara tanto un procedimiento de tipo Sub y otro de tipo Function:

```
Module Module1
    Sub Main()
        MostrarS()
    End Sub
End Module
```



```

    Dim s As String = MostrarF()
    Console.WriteLine(s)
    Console.ReadLine()
End Sub

Sub MostrarS()
    Console.WriteLine("Este es el procedimiento MostrarS")
End Sub

Function MostrarF() As String
    Return "Esta es la función MostrarF"
End Function
End Module

```

La salida producida por este código será la siguiente:

Este es el procedimiento MostrarS

Esta es la función MostrarF

Los procedimientos de tipo Function empiezan con la instrucción Function seguido del nombre de la función y el tipo de dato que devolverá la función, ya que, debido a que las funciones siempre devuelven un valor, lo lógico es que podamos indicar el tipo que devolverá. El final de la función viene indicado por End Function.

Parámetros por valor y parámetros por referencia

Normalmente, cuando se pasa un parámetro a un procedimiento, éste se suele pasar o indicar lo que se llama por valor, es decir, el parámetro será una copia del valor indicado, en el caso de Saludar("Gerardo") la constante "Gerardo" se copiará en la variable nombre. Cualquier cambio que se realice dentro del procedimiento a la variable nombre no afectará al parámetro. Seguramente dirás que sería imposible cambiar el contenido de una constante, que es al fin y al cabo lo que se le pasa al procedimiento pero ¿que ocurre si ese parámetro es una variable en

lugar de una constante? Por ejemplo, si tenemos el siguiente procedimiento:

```
Sub Saludar2(ByVal nombre As String)
    nombre = "Hola " & nombre
    Console.WriteLine(nombre)
End Sub
```

Al que llamamos con este otro código:

```
Sub PruebaSaludar2()
    Dim elNombre As String
    elNombre = "Gerado"
    Saludar2(elNombre)
    '
    ' ¿qué valor mostraría esta línea?
    Console.WriteLine(elNombre)
End Sub
```

¿Qué valor tendrá la variable elNombre después de llamar al procedimiento Saludar2?

La respuesta es: el que tenía antes de llamar al procedimiento.

Porque se ha pasado por valor (ByVal) y por tanto, lo que se ha pasado al procedimiento es una copia del contenido de la variable elNombre, con lo cual, cualquier cambio que se realice en la variable nombre sólo afectará a la copia, no al original. Pero si queremos que el procedimiento pueda modificar el valor recibido como parámetro, tendremos que indicarle al Visual Basic .NET de que lo pase por referencia, para ello habrá que usar la instrucción ByRef en lugar de ByVal.

ejemplo:

```
Sub Saludar3(ByRef nombre As String)
    nombre = "Hola " & nombre
    Console.WriteLine(nombre)
End Sub
```

```
Sub PruebaSaludar3()
    Dim elNombre As String
    elNombre = "Gerardo"
    Saludar3(elNombre)
    '
    ' ¿qué valor mostraría esta línea?
    Console.WriteLine(elNombre)
End Sub
```

En esta ocasión la variable `elNombre` contendrá "Hola Gerardo", al pasar la variable por referencia (`ByRef`), el VB lo que ha hecho es asignar a la variable `nombre` del procedimiento la misma dirección de memoria que tiene la variable `elNombre`, de forma que cualquier cambio realizado en `nombre` afectará a `elNombre`.

Las variables indicadas con `ByVal` se pasan por valor, es decir se hace una copia del contenido de la variable o constante y es esa copia la que se pasa al procedimiento. Por otro lado, los parámetros indicados con `ByRef` se pasan por referencia, es decir se pasa al procedimiento una referencia a la posición de memoria en la que está el contenido de la variable en cuestión, por tanto cualquier cambio efectuado a la variable dentro del procedimiento afectará a la variable indicada al llamar al procedimiento.

Por supuesto, todo esto es aplicable tanto a los procedimientos de tipo Sub como a los de tipo Function. En el caso de las funciones, el utilizar parámetros ByRef nos permiten devolver más de un valor: el que devuelve la función más los que se puedan devolver en los parámetros declarados con ByRef.

3.4.4 APLICACIONES MDI

Una aplicación de tipo o estilo MDI (Multiple Document Interface), Interfaz de Documento Múltiple, se compone de un formulario principal, también denominado formulario MDI, que actuará como contenedor de otros formularios (documentos) abiertos durante el transcurso del programa, denominados formularios hijos o secundarios MDI.

Como ejemplos de este tipo de aplicación tenemos PowerPoint o Access.

A diferencia de lo que ocurría en versiones anteriores de VB, un formulario MDI admite los mismos controles que un formulario normal, aunque dada su orientación de formulario contenedor, se recomienda limitar los controles en un MDI a los estrictamente necesarios. El menú es el ejemplo más identificativo de control idóneo para un formulario MDI, ya que a través de sus opciones, podremos abrir los formularios hijos de la aplicación.

Una vez creado el nuevo proyecto, cambiaremos el nombre del formulario por defecto a frmPrincipal. Para conseguir que este formulario tenga el comportamiento de un contenedor MDI, debemos asignar el valor True a su propiedad IsMdiContainer. También debemos establecer a este formulario como inicial en las propiedades del proyecto.

Ahora pasaremos a la creación de los formularios hijos del MDI. El primero, Reporte, permite la introducción de datos del equipo y del cliente, cuyo contenido podremos en una base de datos en Mysql. La Figura 3.10 muestra este

formulario¹⁵.

The image shows a Windows MDI child form titled "Reporte". It contains several sections for data entry:

- Recepcionista:** Fields for "Recepcionista", "Folio", "Fecha", and "Hora".
- Datos del Reporte:** Fields for "Cliente" (with a dropdown), "Nombre", "Domicilio", "C.P.", "Colonia", "Telefono", "Ciudad", "Estado", and "Contacto".
- Datos del Equipo:** Fields for "Tipo de Equipo" (dropdown), "Marca" (dropdown), "Modelo", and "No. de Serie".
- Observaciones:** A text box for "Observaciones" and a section for "Falla" with a text box.
- Datos del Servicio:** Fields for "Tipo de Servicio" (dropdown) and "Observaciones del Servicio" (text box).

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

Figura 3.10 Formulario hijo de MDI .

Para conseguir que dichos formularios se comporten como hijos del MDI, debemos asignar a su propiedad `MdiParent`, la instancia actual del formulario en ejecución, es decir, `Me`.

```
Private Sub ReportesToolStripMenuItem_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles ReportesToolStripMenuItem.Click
```

```
Dim reporte As Reporte
```

```
    reporte = New Reporte
```

```
    reporte.MdiParent = Me
```

```
    reporte.Show()
```

```
End Sub
```

¹⁵ Figura realizada por el autor

3.4.5 MYSQL CONNECTOR/NET

3.4.5.1 INTRODUCCIÓN

El Connector/Net de MySQL permite a los desarrolladores crear fácilmente aplicaciones .NET que requieren seguridad, alto rendimiento y conectividad con MySQL. Este controlador implementa las interfaces requeridas de ADO.NET y se integra dentro de las herramientas de ADO.NET. Los desarrolladores pueden construir aplicaciones para MySQL usando el Connector/Net y su mejor opción en lenguajes .NET, tal como Visual Basic, C++ o C#. El Connector/Net es un controlador ADO.NET completamente manejable escrito al 100% en C#.

El Connector/Net incluye soporte completo para :

- Características de MySQL 5.0 (procedimientos almacenados, Etc.)
- Características de MySQL 4.1 (Sentencias preparadas del lado del servidor, Unicode y Acceso a memoria compartida, etc.)
- Soporte de paquetes grandes para enviar y recibir filas y datos de más de 2 gigabytes de tamaño.
- El protocolo de compresión que permite la compresión de flujos de datos entre el cliente y el servidor.
- Soporte para conectar usando sockets TCP/IP o memoria compartida en Windows.
- Soporte de conectividad usando sockets TCP/IP o sockets Unix.
- Soporte para Mono, el framework .NET de código abierto desarrollado por Novell.

- Completamente manejable, no utiliza las librerías cliente de MySQL.

3.4.5.2 DESCARGANDO E INSTALANDO

El Connector/Net funciona en cualquier plataforma que soporte el framework .NET, principalmente en versiones recientes de Microsoft Windows

El Connector/Net puede ser instalado a través de un paquete instalador de Windows (.msi), o bien, a través de un sencillo archivo ZIP.

El connector/net esta disponible para su descarga en:

<http://dev.mysql.com/downloads/connector/net/>

Por simplicidad, se ha optado por descargar el archivo ZIP sin el instalador MSI, concretamente se va a utilizar el archivo mysql-connector-net-5.0.3-noinstall.zip que al momento contiene la versión mas reciente del Connector/Net.

Usando un programa ZIP, desempaquetamos este archivo en el directorio de nuestra elección.

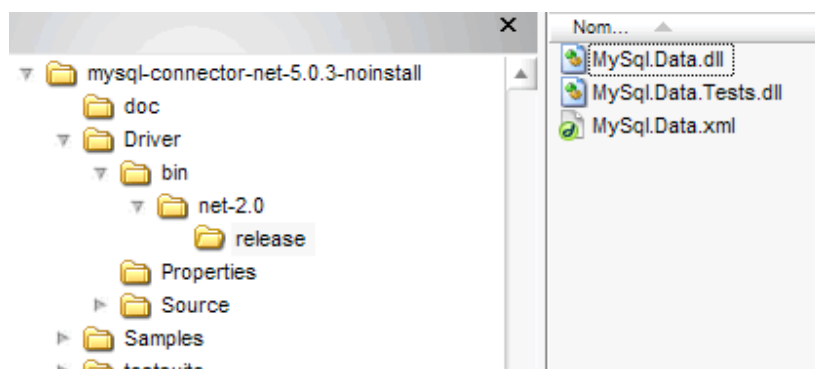


Figura 3.11 Descompresión del archivo connector

Para poder usar el Connector/Net simplemente copiamos el archivo MySql.Data.dll ubicado en la ruta Driver\bin\net-2.0\release al directorio donde ubicaremos

nuestros programas en .

3.4.5.3 ARQUITECTURA DEL CONNECTOR/NET

El Connector/Net contiene varias clases que son usadas para conectar a la base de datos, ejecutar consultas y sentencias y manejar resultados de consultas.

Las siguientes son las principales clases del Connector/Net:

Clase	Descripción
MySqlCommand	Representa una sentencia SQL a ejecutar sobre una base de datos MySQL.
MySqlCommandBuilder	Genera automáticamente comandos de tabla sencilla usados para aplicar los cambios hechos a un dataset con la base de datos asociada.
MySqlConnection	Representa una conexión a un servidor de base de datos.
MySqlDataAdapter	Representa un conjunto de comandos y una conexión que son usados para llenar un dataset y actualizar una base de datos.
MySqlDataReader	Provee un medio de lectura de filas de datos únicamente hacia adelante.
MySqlException	La excepción que es lanzada cuando MySQL devuelve un error.
MySqlHelper	Clase de ayuda que hace más fácil trabajar con el proveedor.
MySqlTransaction	Representa una transacción SQL que se hará en una base de datos.

Las clases que provee el Connector/NET están contenidas en la biblioteca

MySql.Data.dll (el archivo que copiamos previamente), así que al crear una aplicación que haga uso del Connector/Net este archivo debe ser referenciado. Las clases usadas por el servidor MySQL están definidas en el espacio de nombres MySql.Data.MySqlClient, esto es algo muy importante que hay que tener presente.

3.4.5.4 MYSQLCONNECTION

Para crear las conexiones se utiliza la clase MySqlConnection que esta definida en el espacio de nombres MySql.Data.MySqlClient. Antes de usar esta clase primero debe incluir dicho espacio de nombres su archivo a continuación veremos la forma de establecer una conexión permanente a un origen de datos.

' Procedimiento de Arranque

Sub Main()

'Los parámetros de la sobrecarga con más parámetros son:

'1.Nombre del usuario a la base de datos=sae_user

'2.Password del usuario a la base de datos=amiami

'3.Nombre de la base de datos=sae_compupaq

'4.Nombre del servidor o direccion IP=192.168.1.2 o compupaq.homelinux.net

'5. Puerto por el que se accede al servidor. Típicamente 3306

Cn = New MySqlConnection()

Cn.ConnectionString = "User Id=user;Password=amiami;Initial

Catalog=sae_compupaq;Data Source=bioenlace.webhop.info"

Cn.Open()

MDI.Showdialog()

Catch ErrorCn As MySqlException

MessageBox.Show("Error al conectar: " & ErrorCn.Message)

End Try
End Sub

3.4.5.5 MYSQLCOMMAND

Cualquier acción que se ejecuta en el servidor o en una base de datos es realizada por un objeto llamado comando.

Para soportar las diversas acciones que se pueden ejecutar en un servidor MySQL, el connector/NET provee la clase MySqlCommand. En concreto, un objeto MySqlCommand representa una sentencia SQL que será ejecutada en contra de un servidor MySQL. Para usar esta clase, tenemos que crear un objeto del tipo MySqlCommand usando cualquiera de los constructores disponibles. La acción que será ejecutada es creada como una sentencia en una cadena. Esta acción es representada por la propiedad MySqlCommand.CommandText que es del tipo string.

Ejecución del comando

Después de establecer una conexión y especificar el comando adecuado, lo podemos ejecutar. Para hacer esto la clase MySqlCommand esta equipada con diversos métodos que están relacionados con el tipo de consulta a ejecutar. Dos de estos métodos son ExecuteNonQuery() y ExecuteReader().

Ninguno de estos métodos toma algún argumento. El objeto comando que lo invoca debe tener una sentencia que sea valida y previamente preparada. El método ExecuteNonQuery() ejecuta una sentencia SQL y regresa el numero de filas afectadas. El método ExecuteReader() manda la consulta y construye un objeto MySqlDataReader. Con estas características, debe ser obvio que el primer método será usado comúnmente con sentencias INSERT, DELETE, UPDATE, etc., y el segundo método con sentencias SELECT.

CÓDIGO:

```
Imports MySql.Data
Imports MySql.Data.MySqlClient
Module PGenerales
    Public useractivado, idclientecli rep As String
    Dim dolar As Decimal
    Public Cn As New MySqlConnection
    Public MDI As New MDI
    Dim mysqlcommSQL As New MySqlCommand
    Dim mysqldatareader As MySql.Data.MySqlClient.MySqlDbType
    Dim comando As MySql.Data.MySqlClient.MySqlCommand
    Dim Ds As New DataSet
    ' Enumeración de tipos de datos que se pueden validar
    Enum TipoNumero
        Entero = 0
        Real = 1
    End Enum
    ' Procedimiento de Arranque
    Sub Main()
        Cn = New MySqlConnection()
        Cn.ConnectionString = "User Id=user;Password=amiami;Initial
Catalog=sae_compupaq;Data Source=bioenlace.webhop.info"
        Try
            Dim frm As New Login
            If frm.ShowDialog() = DialogResult.OK Then
                useractivado = frm.txtusuario.Text
                dolar = frm.TextBox1.Text
            Else
                End
            End If
            Cn.Open()
        End Try
    End Sub
End Module
```

```
Application.Run(MDI)
```

```
Catch ErrorCn As MySqlException  
    MessageBox.Show("Error al conectar: " & ErrorCn.Message)  
End Try  
End Sub
```

```
Public Sub MOSTRAR(ByVal comandos As MySqlCommand, ByVal ubicacion  
As Object)  
    Dim dr As MySqlDataReader  
    comando = comandos  
    dr = comando.ExecuteReader  
    While dr.Read  
        ubicacion.Text = dr.Item(0).ToString  
    End While  
    dr.Close()  
End Sub
```

```
MOSTRAR(New MySqlCommand("select nombre FROM cliente where  
id=" & idclienteclirep & "", Cn), Me.NOMBRE)
```

3.4.6 DATAGRIDVIEW

Para que al ejecutar una aplicación, un control datagrid se llene automáticamente (encabezados y datos) con el conjunto de registro de un control de datos, es suficiente asignar a su propiedad datasource el nombre del control de datos, esto permite a los usuarios examinar y modificar tablas completas de bases de datos o de resultados de consultas.

El control datagrid depende de 2 objetos: el objeto recordset del control de datos y la colección columns del propio control de datagridview. Esto es porque en realidad un datagridview es una colección de objetos column, de los cuales cada uno tiene un número variable de filas.

Cada columna corresponde a un campo de la tabla de la base de datos, mientras que cada fila corresponde a un registro.

Data set

El DataSet de ADO.NET es una representación de datos residente en memoria que proporciona un modelo de programación relacional coherente independientemente del origen de datos que contiene. Un DataSet representa un conjunto completo de datos, incluyendo las tablas que contienen, ordenan y restringen los datos, así como las relaciones entre las tablas.

Hay varias maneras de trabajar con un DataSet, que se pueden aplicar de forma independiente o conjuntamente. Puede:

Crear mediante programación una DataTable, DataRelation y una Constraint en un DataSet y rellenar las tablas con datos.

Llenar el DataSet con tablas de datos de un origen de datos relacional existente mediante DataAdapter.

Cargar y hacer persistente el contenido de DataSet mediante XML. Para obtener más información, vea Utilizar XML en un DataSet.

'DATAGRID VIEW CÓDIGO

```
Private Sub llenar()  
    Try  
        Dim connStr As String 'Cadena de caracteres para la conexión  
        Dim adapter As New MySqlConnection 'Adaptador de datos de MySQL  
para poder conectar un dataset  
        Dim sqlstr As String 'Comando SQL para correr en el comando del  
adaptador o comando para leer  
  
        objAlignRightCellStyle.Alignment =  
DataGridViewContentAlignment.MiddleRight  
        ' Declare and set the alternating rows style...  
objAlternatingCellStyle.BackColor = Color.LightYellow  
DataGridView1.AlternatingRowsDefaultCellStyle = objAlternatingCellStyle  
        ' Declare and set the style for currency cells ...  
  
        connStr="UserId=user;Password=amiami;Initial  
Catalog=sae_compupaq;Data Source=bioenlace.webhop.info"  
        sqlstr = "select * from cliente"  
  
        adapter.SelectCommand = New MySqlCommand(sqlstr, Cn)  
myDataSet.Clear()  
adapter.Fill(myDataSet, "Columnas")  
DataView1.Table = myDataSet.Tables("Columnas")  
DataGridView1.DataSource = DataView1  
Dim objConn As New MySqlConnection(connStr)  
Dim Comando As New MySqlConnection("Select  
Id,Nombre,rfc,ciudad,estado From cliente", objConn)  
objConn.Open()  
Comando.Fill(myDataSet, "tabla")  
DataGridView1.DataSource = myDataSet.Tables("tabla")
```

```

objCurrencyCellStyle.Format = "c"
objCurrencyCellStyle.Alignment =
DataGridViewContentAlignment.MiddleRight
' Change column names and styles using the column index
DataGridView1.Columns(0).HeaderText = "ID CLIENTE"
DataGridView1.Columns(1).HeaderText = "NOMBRE"
DataGridView1.Columns(2).HeaderText = "RFC"
DataGridView1.Columns(3).HeaderText = "CIUDAD"
DataGridView1.Columns(4).HeaderText = "ESTADO"

DataGridView1.Columns(0).Width = 60
DataGridView1.Columns(1).Width = 265
DataGridView1.Columns(2).Width = 70
DataGridView1.Columns(3).Width = 70
DataGridView1.Columns(4).Width = 65
Catch ErrorCn As Exception
    MessageBox.Show("Error: " & ErrorCn.Message)
End Try

End Sub

```

3.5 MYSQL

3.5.1 INTRODUCCIÓN

El software MySQL proporciona un servidor de base de datos SQL (Structured Query Language) muy rápido, multi-threaded, multi usuario y robusto. El servidor MySQL está diseñado para entornos de producción críticos, con alta carga de trabajo así como para integrarse en software para ser distribuido. MySQL es una marca registrada de MySQL AB.

El software MySQL tiene una doble licencia. Los usuarios pueden elegir entre usar el software MySQL como un producto Open Source bajo los términos de la licencia GNU General Public License o pueden adquirir una licencia comercial estándar de MySQL AB.

3.5.2 PANORÁMICA DEL SISTEMA DE GESTIÓN DE BASE DE DATOS MYSQL

MySQL, el sistema de gestión de bases de datos SQL Open Source más popular, lo desarrolla, distribuye y soporta MySQL AB. MySQL AB es una compañía comercial, fundada por los desarrolladores de MySQL. Es una compañía Open Source de segunda generación que une los valores y metodología Open Source con un exitoso modelo de negocio.

El sitio web MySQL (<http://www.mysql.com/>) proporciona la última información sobre MySQL y MySQL AB.

- MySQL es un sistema de gestión de bases de datos.
- Una base de datos es una colección estructurada de datos.
- MySQL es un sistema de gestión de bases de datos relacionales.

- Una base de datos relacional almacena datos en tablas separadas en lugar de poner todos los datos en un gran almacén. Esto añade velocidad y flexibilidad.
- MySQL software es Open Source.

Open Source significa que es posible para cualquiera usar y modificar el software. Cualquiera puede bajar el software MySQL desde internet y usarlo sin pagar nada. Si lo desea, puede estudiar el código fuente y cambiarlo para adaptarlo a sus necesidades. El software MySQL usa la licencia GPL (GNU General Public License). El servidor de base de datos MySQL es muy rápido, fiable y fácil de usar. MySQL Server trabaja en entornos cliente/servidor o incrustados.

El software de bases de datos MySQL es un sistema cliente/servidor que consiste en un servidor SQL multi-threaded que trabaja con diferentes programas y bibliotecas cliente, herramientas administrativas y un amplio abanico de interfaces de programación para aplicaciones (APIs).

3.5.3 PRINCIPALES CARACTERÍSTICAS DE MYSQL

La siguiente lista describe algunas de las características más importantes del software de base de datos MySQL..

- Interioridades y portabilidad.
- Escrito en C y en C++ .
- Probado con un amplio rango de compiladores diferentes.
- APIs disponibles para C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, y Tcl.
- Uso completo de multi-threaded mediante threads del kernel. Pueden usarse fácilmente multiple CPUs si están disponibles.

- Proporciona sistemas de almacenamiento transaccionales y no transaccionales.
- Usa tablas en disco B-tree (MyISAM) muy rápidas con compresión de índice.
- Relativamente sencillo de añadir otro sistema de almacenamiento. Esto es útil si desea añadir una interfaz SQL para una base de datos propia.
- Un sistema de reserva de memoria muy rápido basado en threads.
- Joins muy rápidos usando un multi-join de un paso optimizado.
- Las funciones SQL están implementadas usando una librería altamente optimizada y deben ser tan rápidas como sea posible. Normalmente no hay reserva de memoria tras toda la inicialización para consultas.
- El servidor está disponible como un programa separado para usar en un entorno de red cliente/servidor.
- Seguridad.
 - Un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite verificación basada en el host. Las contraseñas son seguras porque todo el tráfico de contraseñas está encriptado cuando se conecta con un servidor.
- Escalabilidad y límites.
- Soporte a grandes bases de datos que contiene 5 millones de registros.
- Conectividad

- Los clientes pueden conectar con el servidor MySQL usando sockets TCP/IP en cualquier plataforma. En sistemas Windows de la familia NT (NT,2000,XP, o 2003), los clientes pueden usar named pipes para la conexión. En sistemas Unix, los clientes pueden conectar usando ficheros socket Unix.

En MySQL 5.0, los servidores Windows soportan conexiones con memoria compartida si se inicializan con la opción `--shared-memory`. Los clientes pueden conectar a través de memoria compartida usando la opción `--protocol=memory`.

La interfaz para el conector ODBC (MyODBC) proporciona a MySQL soporte para programas clientes que usen conexiones ODBC (Open Database Connectivity). Por ejemplo, puede usar MS Access para conectar al servidor MySQL. Los clientes pueden ejecutarse en Windows o Unix.

La interfaz para el conector J MySQL proporciona soporte para clientes Java que usen conexiones JDBC. Estos clientes pueden ejecutarse en Windows o Unix.

Localización El servidor puede proporcionar mensajes de error a los clientes en muchos idiomas.

Clientes y herramientas

MySQL server tiene soporte para comandos SQL para chequear, optimizar, y reparar tablas. Estos comandos están disponibles a través de la línea de comandos y el cliente `mysqlcheck`.

3.5.4 INSTALAR MYSQL SERVER EN WINDOWS

El instalador para la versión Windows de MySQL 5.0, en conjunto con un asistente de configuración dotado de interfaz gráfica, instala automáticamente MySQL, crea un fichero de opciones, inicia el servidor, y otorga seguridad a las cuentas de usuario por defecto. A continuación se muestra los pasos de para instalar Mysql.



Figura 3.12 Tipo de instalación

Hay disponibles tres tipos de instalación: **típica**, **completa**, y **personalizada** Fig. 3.12.

La instalación **típica** instala el servidor MySQL, el cliente de línea de comandos **mysql**, y las utilidades de línea de comandos. Los clientes y utilidades incluyen **mysqldump**, **myisamchk**, y otras herramientas que ayudan a administrar el servidor MySQL.

La instalación **completa** instala todos los componentes incluidos en el paquete. El paquete completo incluye componentes como el servidor incrustado (embedded), el conjunto de pruebas de rendimiento (benchmarks), scripts de mantenimiento, y documentación.

La instalación **personalizada** otorga un control completo sobre los paquetes que se desea instalar y el directorio de instalación que se utilizará.

Si se escoge la instalación **típica** o la **completa**, al hacer click sobre el botón Siguiente se avanza a la pantalla de confirmación para verificar las opciones y comenzar la instalación. Si se escoge la instalación **personalizada**, al hacer click sobre el botón Siguiente, se avanza al cuadro de diálogo de instalación personalizada, descrito en Fig. 3.13.



Figura 3.13 Instalación personalizada

Si se desea cambiar el directorio de instalación o los componentes que se instalarán, se deberá elegir el tipo de instalación personalizada.

Todos los componentes disponibles se encuentran en un diagrama de árbol en el lado izquierdo del cuadro de diálogo de instalación personalizada. Los componentes que no serán instalados tienen un icono X rojo; los componentes que se instalarán tienen un icono gris. Para indicar si un componente se instalará o no, debe hacerse click en su icono y elegir una opción de la lista desplegable que aparece. Se puede cambiar el directorio de instalación por defecto haciendo click en el botón Cambiar... a la derecha del directorio de instalación que se

muestra.

Después de elegir los componentes a instalar y el directorio de instalación, hacer click en el botón Siguiente hará avanzar al cuadro de diálogo de confirmación Fig. 3.14.



Figura 3.14 Ventana de confirmación de instalación

Una vez que se elige un tipo de instalación y los componentes a instalar, se avanza al cuadro de diálogo de confirmación. Se muestran el tipo y el directorio de instalación se para ser confirmados.

Para instalar MySQL una vez que se está conforme con la configuración, debe hacerse click en el botón Instalar. Para cambiar la configuración, debe hacerse click en el botón Retroceder. Para abandonar el asistente de instalación sin terminar la instalación de MySQL, debe hacerse click en el botón Cancel.

Una vez que la instalación está completa, se proporciona la opción de registrarse en el sitio web de MySQL. El registro otorga acceso para publicar mensajes en los foros de Mysql, en forums.mysql.com, junto con la posibilidad de informar errores en bugs.mysql.com y suscribirse al boletín electrónico. La pantalla final del

instalador brinda un resumen de la instalación y la opción de ejecutar el asistente de configuración MySQL, que se utiliza para crear un fichero de configuración, instalar el servicio MySQL, y establecer la seguridad.

En la siguiente ventana podemos registrarnos en MySQL.com o si ya estamos registrados introducir email de registro y contraseña Fig. 3.15. También podemos cancelar el registro. En nuestro caso, puesto que ya estamos registrados marcaremos "Login to MySQL.com e introduciremos email y contraseña".

Para registrarse desde aquí marcaremos en "Create a new free MySQL.com account" y iremos rellenando los datos que nos pide:



Figura 3.15 Ventana de registro de mysql

Si queremos configurar MySQL en este momento dejaremos marcada la opción "Configure the MySQL Server now" y pulsaremos en "Finish" (Ver Fig. 3.16).



Figura 3.16 Ventana de confirmación de configuración de Mysql

El asistente de configuración MySQL automatiza el proceso de configurar el servidor bajo Windows. Crea un fichero my.ini personalizado, realizando una serie de preguntas y aplicando las respuestas a una plantilla para generar un fichero my.ini apropiado para la instalación en curso. El asistente de configuración MySQL se incluye con la versión 5.0 de MySQL, y por el momento sólo está disponible para usuarios de Windows Fig. 3.17.



Figura 3.17 Configuración de Mysql

El asistente de configuración generalmente se ejecuta a continuación del asistente de instalación, ni bien éste finaliza. También puede iniciarse haciendo click en la entrada MySQL Server Instance Config Wizard de la sección MySQL del menú Inicio.



Figura 3.18 Tipo de configuración

Hay disponibles dos tipos de configuración: Configuración detallada (Detailed Configuration) y Configuración estándar (Standard Configuration) (Ver Fig. 3.18).

La Configuración estándar está orientada a usuarios nuevos que deseen comenzar rápidamente con MySQL sin tener que tomar varias decisiones relativas a la configuración del servidor.

La Configuración detallada está dirigida a usuarios avanzados que deseen un control más preciso sobre la configuración del servidor.

Si se trata de un usuario nuevo de MySQL y necesita un servidor configurado para un ordenador de desarrollo con un único usuario, la Configuración estándar debería cubrir sus necesidades. Al elegir la Configuración estándar el asistente de configuración MySQL establece todas las opciones de configuración

automáticamente, a excepción de Opciones de servicio (Service options) y Opciones de seguridad (Security options) .

La Configuración estándar establece opciones que pueden ser incompatibles con sistemas donde existen instalaciones de MySQL previas. Si se posee una instalación de MySQL anterior además de la que se está configurando, se recomienda optar por la Configuración detallada (Detailed configuration).



Figura 3.19 Configurar tipo de servidor

Hay tres tipos de servidor (Ver Fig. 3.19) distintos para elegir, y el tipo que se escoja afectará a las decisiones que el asistente de configuración MySQL tomará en relación al uso de memoria, disco y procesador.

Developer machine (Ordenador de desarrollo): Esta opción se aplica a ordenadores de escritorio donde MySQL está orientado a un uso personal solamente. Se asume que se estarán ejecutando varias otras aplicaciones, por lo que el servidor MySQL se configura para utilizar una cantidad mínima de recursos del sistema.

Server machine (Servidor): Esta opción se aplica a servidores donde MySQL se

ejecuta junto con otras aplicaciones de servidor como son FTP, correo electrónico, y servidores web. MySQL se configura para utilizar una cantidad moderada de recursos del sistema.

Dedicated MySQL Server Machine (Servidor MySQL dedicado): Esta opción se aplica a ordenadores donde solamente se ejecuta el servidor MySQL. Se asume que no hay otras aplicaciones ejecutándose. El servidor MySQL se configura para utilizar todos los recursos disponibles en el sistema.



Figura 3.20 Uso de la base de datos

El cuadro de diálogo Uso de la base de datos (Database usage) Fig. 3.20 permite indicar los gestores de tablas que se planea utilizar al crear tablas de MySQL. La opción que se escoja determinará si el motor de almacenamiento InnoDB estará disponible y qué porcentaje de los recursos de servidor estarán disponibles para InnoDB.

Base de datos polifuncional (Multifunctional database) : Esta opción habilita tanto el motor de almacenamiento InnoDB como MyISAM y reparte los recursos uniformemente entre ambos. Se recomienda para usuarios que emplearán los dos motores de almacenamiento en forma habitual.

Base de datos transaccional exclusiva (Transactional database only) : Esta opción habilita tanto el motor de almacenamiento InnoDB como MyISAM, pero destina más recursos del servidor al motor InnoDB. Se recomienda para usuarios que emplearán InnoDB casi exclusivamente, y harán un uso mínimo de MyISAM.

Base de datos no-transaccional exclusiva (Non-transactional database only) : Esta opción deshabilita completamente el motor de almacenamiento InnoDB y destina todos los recursos del servidor al motor MyISAM. Recomendado para usuarios que no utilizarán InnoDB.



Figura 3.21 Ubicación de los ficheros InnoDB

Algunos usuarios pueden querer ubicar los ficheros InnoDB en una ubicación diferente al directorio de datos del servidor MySQL. Esto puede ser deseable si el sistema tiene disponible un dispositivo de almacenamiento con mayor capacidad o mayor rendimiento, como un sistema RAID.

Para modificar la ubicación por defecto de los ficheros InnoDB (Ver Fig. 3.21), debe elegirse una nueva unidad de disco en la lista desplegable de letras de unidades y elegir una nueva ruta en la lista desplegable de rutas. Haciendo click

en el botón ... podrá crearse una ruta personalizada.

Si se está modificando la configuración de un servidor preexistente, debe hacerse click en el botón Modify antes de cambiar la ruta. En dicho caso habrá que desplazar manualmente los ficheros InnoDB existentes hacia la nueva ubicación antes de iniciar el servidor.



Figura 3.22 Conexiones simultaneas

Es importante establecer un límite para las conexiones simultáneas que se podrán establecer con el servidor MySQL, para evitar que éste se quede sin recursos. El cuadro de diálogo Conexiones simultáneas (Concurrent connections) Fig. 3.22 permite indicar el uso que se planea darle al servidor, y establecer en consecuencia el límite de conexiones simultáneas. También es posible introducir manualmente el límite.

Soporte de decisiones (Decision support (DSS)/OLAP) : Debe escogerse esta opción si el servidor no necesitará una gran cantidad de conexiones simultáneas. El número máximo de conexiones se establece en 100, asumiéndose un promedio de 20 conexiones simultáneas.

Proceso de transacciones en línea (Online transaction processing (OLTP)) : Debe escogerse esta opción si el servidor necesitará un gran número de conexiones simultáneas. El número máximo de conexiones se establece en 500.

Configuración manual (Manual setting): Debe escogerse esta opción para establecer manualmente el número máximo de conexiones simultáneas que admitirá el servidor. El número deseado puede elegirse de una lista desplegable o teclearse si no figura en ella.



Figura 3.23 Opciones de red

El cuadro de diálogo Opciones de red (Networking options) Fig. 3.23 permite activar o desactivar el protocolo TCP/IP y modificar el número de puerto por el que se accederá al servidor MySQL.

El protocolo TCP/IP está activado por defecto. Para desactivarlo debe quitarse la marca de la casilla al lado de la opción Activar TCP/IP (Enable TCP/IP networking) Por defecto se utiliza el puerto 3306 para acceder a MySQL. Para modificar este valor, el número deseado puede elegirse de una lista desplegable o teclearse si no figura en la lista. Si el puerto indicado ya se encuentra en uso, se solicitará la confirmación de la elección.



Figura 3.24 Selección de caracteres

El servidor MySQL soporta múltiples conjuntos de caracteres, y es posible establecer uno por defecto, que se aplicará a todas las tablas, columnas y bases de datos, a menos que se sustituya (Ver Fig. 3.24). Debe emplearse el cuadro de diálogo Character set para cambiar en el servidor el conjunto de caracteres por defecto.

Juego de caracteres estándar (Standard character set) : Esta opción establecerá a Latin1 como el juego de caracteres por defecto en el servidor. Latin1 se usa para el Inglés y muchos idiomas de Europa Occidental.

Soporte multilingüe mejorado (Best support for multilingualism) : Esta opción establece a UTF8 como el conjunto de caracteres por defecto en el servidor. UTF8 puede almacenar caracteres de muchos idiomas diferentes en un único juego.

Selección manual del conjunto de caracteres por defecto / colación (Manual selected default character set / collation) : Esta opción se emplea cuando se desea elegir manualmente el juego de caracteres por defecto del servidor, a través de una lista desplegable.



Figura 3.25 Asistente de configuración de Mysql

En plataformas basadas en Windows NT, el servidor MySQL puede instalarse como un servicio (Ver Fig. 3.25). De ese modo, se iniciará automáticamente durante el inicio del sistema, e incluso será reiniciado automáticamente por Windows en caso de producirse un fallo en el servicio.

El asistente de configuración MySQL instala por defecto el servidor MySQL como un servicio, utilizando el nombre de servicio MySQL. Si se desea evitar la instalación del servicio, debe vaciarse la casilla al lado de la opción Instalar como servicio Windows (Install as Windows service) . Se puede modificar el nombre del servicio eligiendo un nuevo nombre o tecleándolo en la lista desplegable provista. Para instalar el servidor MySQL como un servicio pero que no se ejecute al iniciarse Windows, debe vaciarse la casilla al lado de la opción Ejecutar el servidor MySQL automáticamente (Launch the MySQL server automatically).



Figura 3.26 Seguridad de Mysql

Se recomienda fuertemente que se establezca una contraseña para el usuario root del servidor MySQL (Ver Fig. 3.26). El asistente de configuración MySQL la solicita por defecto. Si no se desea establecer una contraseña, debe vaciarse la casilla al lado de la opción Modificar configuración de seguridad (Modify security settings) . Para establecer la contraseña del usuario root, se debe introducir tanto en el cuadro de texto Nueva contraseña de root (New root password) como en Confirmar (Confirm). Si se está reconfigurando un servidor existente, también será necesario introducir la contraseña en vigencia dentro del cuadro de texto Contraseña de root actual (Current root password).

Para evitar que el usuario root inicie sesión desde cualquier punto de la red, debe marcarse la casilla al lado de la opción Root sólo puede conectarse en modo local (Root may only connect from localhost). Esto fortalece la seguridad de la cuenta de root.

Para crear una cuenta de usuario anónimo, debe marcarse la casilla al lado de la opción Crear una cuenta de anónimo (Create An Anonymous Account) . No se recomienda crear un usuario anónimo porque puede disminuir la seguridad del servidor y ocasionar dificultades de inicio de sesión y de permisos.

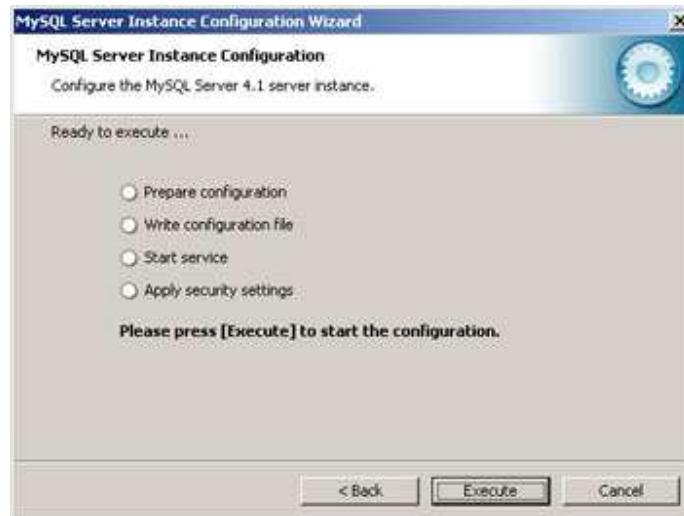


Figura 3.27 Ventana de confirmación

El último cuadro de diálogo Fig. 3.27 del asistente de configuración MySQL es el de Confirmación (Confirmation dialog). Para concretar el proceso de configuración, debe hacerse click en el botón Ejecutar (Execute). Para volver a un cuadro de diálogo anterior, debe hacerse click en el botón Atrás (Back). Para abandonar el asistente de configuración sin cambiar la configuración del servidor, debe hacerse click en el botón Cancelar (Cancel).

Después de hacer click en el botón Ejecutar (Execute), el asistente de configuración MySQL llevará a cabo una serie de tareas cuyo avance se mostrará en la pantalla a medida que cada etapa termine (Ver Fig. 3.28).



Figura 3.28 Configurar tipo de servidor

Después de que el asistente de configuración MySQL haya completado sus tareas, se mostrará un resumen. Haciendo click en el botón Terminar (Finish) se abandonará el asistente.

CAPÍTULO IV

Implementación del sistema

4.1 IMPLEMENTACIÓN

En este capítulo se muestran las ventanas terminadas del sistema administrativo remoto. Estas ventanas se implementaron en el lenguaje de programación Visual Basic y usando Mysql 5.0 como manejador de base de datos.

El manejador de base de datos como toda la información se encuentra montada en Centos 5.0 que es una distribución de Linux por default en el momento de la instalación del sistema operativo se instalo la última versión del mysql que es la 5.0.

A continuación se presentan las ventajas de utilización de Linux:

- En Linux pueden correr varios procesos a la vez de forma ininterrumpida como un servidor de red al tiempo que un procesador de textos, una animación, copia de archivos o revisar el correo electrónico.
- Seguridad porque es un sistema operacional diseñado con la idea de Cliente - Servidor con permisos de acceso y ejecución a cada usuario. Esto quiere decir que varios usuarios pueden utilizar una misma máquina al tiempo sin interferir en cada proceso.
- Linux es software libre, casi gratuito. Linux es popular entre programadores y desarrolladores e implica un espíritu de colaboración.
- Linux integra una implementación completa de los diferentes protocolos y estándares de red, con los que se puede conectar fácilmente a Internet y acceder a todo tipo de información disponible.
- Su filosofía y sus programas están dictados por el movimiento 'Open Source' que ha venido crecido en los últimos años y ha adquirido el suficiente fortaleza para

hacer frente a los gigantes de la industria del software.

- Linux puede ser utilizado como una estación personal pero también como un potente servidor de red.
- Linux incorpora una gama de sistemas de interfaz gráfica (ventanas) de igual o mejor calidad que otras ofrecidas en muchos paquetes comerciales.
- El paquete incluye el código fuente, lo que permite modificarlo de acuerdo a las necesidades del usuario.
- Utiliza varios formatos de archivo que son compatibles con casi todos los sistemas operacionales utilizados en la actualidad.

INSTALACIÓN DE LOS COMPONENTES DEL SISTEMA

Antes de iniciar con el programa necesitaremos instalar los elementos que componen al sistema en este caso son 2 el .Net framework 2.0 y Mysql Connector 5.0.7. Los 2 programas vienen con el sistema ya junto en caso de que no este framework se puede descargar desde www.microsoft.com/downloads es la versión gratuita de Microsoft en el caso de ser un sistema operativo mas avanzado del xp con en el caso del vista no es necesario instalarlo ya que viene integrado.

Instalación de .Net Framework

Algunos de los últimos programas se están desarrollando bajo la plataforma .Net, estas aplicaciones necesitan .Net Framework para poder ejecutarse. Con esta aplicación tendrás todas las librerías y rutinarias necesarias para poder instalar y utilizar los programas.

Instalación de Mysql connector .net 5.0.7

Con este programa la aplicación tendrá la posibilidad de conectarse remotamente al sistema, y ofrecer un mejor rendimiento a la hora de realizar consultas o cualquier movimiento.

Ya con los complementos instalados lo único que deberemos es correr la aplicación y abrirá lo que será nuestra ventana con su menú (Ver Fig. 4.1) ²⁸.

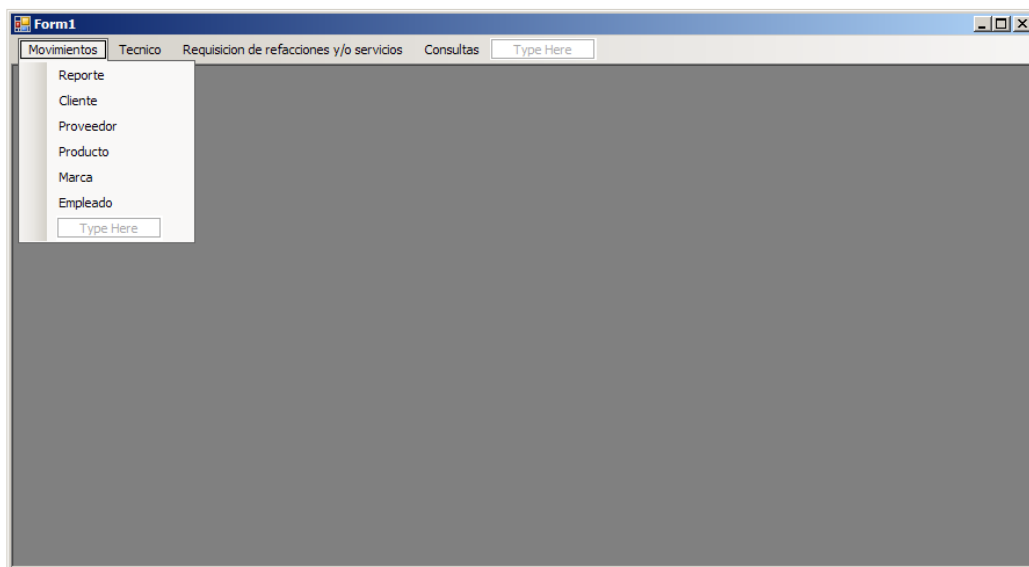


Figura 4.1 Ventana principal del sistema

¹⁶ Figura realizada por el autor

4.2 REPORTE

Aquí se captura toda la información del cliente y del equipo a reparar necesaria para el reporte.

La forma de acceder a este módulo es desde el menú principal en la parte de movimientos¹⁷.

Reporte

Recepcionista: JUAN GERARDO CARDENAS TRIGUEROS

Folio: 229

Fecha: 2008-4-19

Hora: 02:00:48 p.m.

Datos del Reporte

Cliente: [] []

Nombre: []

Domicilio: []

C.P.: []

Colonia: []

Telefono: []

Ciudad: []

Estado: []

Contacto: []

Datos del Equipo

Tipo de Equipo: ADAPTADOR DE CORRIE

Marca: 3COM

Modelo: []

No. de Serie: []

Observaciones: []

Falla: []

Datos del Servicio

Tipo de Servicio: GARANTIA

Observaciones del Servicio: []

Aceptar Cancelar

Figura 4.2 Ventana de alta de reportes

Como se puede en esta ventana se muestran todo los datos que corresponden a los datos del cliente, del equipo, servicio y los datos del día y hora que se dio de alta el reporte.

Para introducir los datos del cliente solo es necesario introducir el número del cliente y automáticamente se llenaran todos los campos de la parte de datos del cliente o en caso de no saber el id del cliente se hace clic en el botón de llamar ventana alta de cliente o se presiona F11 y aparecerá la siguiente ventana¹⁸.

¹⁷ Figura realizada por el autor

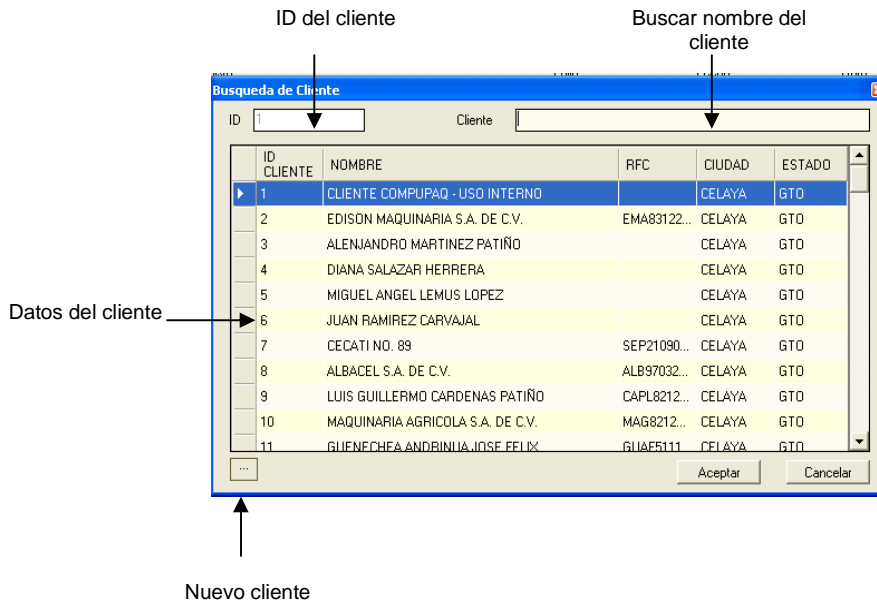


Figura 4.2 Consulta de clientes

En este módulo se hace la búsqueda de cliente para buscar a un cliente solo se escribe el nombre del cliente en el campo de buscar nombre del cliente y automáticamente se irá filtrando la información hasta que aparezca el nombre deseado en caso de aparecer el nombre se selecciona y hace clic en aceptar se cerrara el módulo y llenar los campos en el reporte en caso contrario se hace clic en nuevo cliente o presionando F2 y aparece el módulo de altas de cliente.

¹⁸ Figura realizada por el autor

Datos generales del cliente

Cliente

Datos Generales | Datos Financieros | Contactos

ID: 105

Nombre: []

Domicilio: []

Colonia: [] Ciudad: []

Estado: AGUASCALIENTES País: MEXICO

Email: [] C.P.: []

Teléfono: [] Fax: []

RFC: [] CURP: []

Facturable

Aceptar Cancelar

Figura 4.3 Ventana de clientes

En este módulo se da de alta los datos generales del usuario. En el checkbox facturable en caso de estar seleccionado se introduce el RFC del cliente en caso contrario no permitirá insertar información ¹⁹.

¹⁹ Figura realizada por el autor

The image shows a software window titled "Cliente" with a blue title bar and a close button. It has three tabs: "Datos Generales", "Datos Financieros", and "Contactos". The "Datos Financieros" tab is selected. The window contains the following fields:

Saldo	\$	<input type="text"/>	Días de Crédito	<input type="text"/>
Límite de Crédito	\$	<input type="text"/>	Revisión	\$ <input type="text"/>
Acumulado	\$	<input type="text"/>	Pago	\$ <input type="text"/>

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

Figura 4.4 Ventana de clientes datos financieros

En la pestaña de de datos financieros se le introducen esta información solo para clientes con tiempo recibiendo servicios del negocio ²⁰.

²⁰ Figura realizada por el autor

The image shows a software window titled "Cliente" with a blue border and a close button in the top right corner. The window has three tabs: "Datos Generales", "Datos Financieros", and "Contactos", with "Contactos" being the active tab. The window is divided into two main sections: "Contacto de Compras" and "Contacto de Pagos". Each section contains three input fields: "Nombre" (Name), "Teléfono" (Phone) and "Extensión" (Extension) (Phone Extension), and "Email". At the bottom right of the window, there are two buttons: "Aceptar" (Accept) and "Cancelar" (Cancel).

Figura 4.5 Ventana de clientes contactos

En la pestaña de contactos se introducen los datos en caso de que el servicio se le este ofreciendo a un negocio o empresa en el que hay varias personas encargadas de compras y ventas; esos datos se introducen en sus respectivos campos.

Ya realizado lo anterior en el módulo de reportes se insertan los campos respectivos y se le da clic en aceptar y preguntar si se desea guardar los datos y que se impriman, al imprimir el reporte se muestra a continuación el reporte ya impreso como se le dará al usuario este incluye todos los datos y las políticas del soporte técnico en el que se le pide que firme de que está de acuerdo y se le da una copia del reporte (Ver fig. 4.6)²¹.

FOLIO: 230 **FECHA:** 2008-4-20 **HORA:** 03:26:23 p.m.

ELABORO: JUAN GERARDO CARDENAS TRIGUEROS

DATOS DEL CLIENTE

CLIENTE: 1	NOMBRE: CLIENTE COMPUPAQ - USO INTERNO
DOMICILIO: HERMENEGILDO GALEANA NO. 346	COLONIA: CENTRO
CIUDAD: CELAYA	ESTADO: GTO CP: 38000
TELEFONO: 461-616-04-29	CONTACTO:

DATOS DEL EQUIPO

TIPO DE EQUIPO: ADAPTADOR DE CORRIEN	MARCA: 3COM	MODELO:
NO. SERIE:		
OBSERVACIONES:		
FALLA:		

DATOS DEL SERVICIO

TIPO DE SERVICIO: GARANTIA
OBSERVACIONES:

POLÍTICAS DEL DEPTO. DE SOPORTE TÉCNICO:

- El diagnóstico de cada servicio tendrá un costo determinado por el tabulador correspondiente.
- El costo de la reparación será determinado por el depto. De soporte y cotizado al cliente por el depto. De ventas. En caso de que por cualquier motivo la reparación no se efectúe, se hará el cargo correspondiente de la revisión. En caso de que la reparación proceda, la revisión será gratuita.
- Una vez notificado el diagnóstico y/o reparación al cliente, deberá recoger su equipo en un plazo no mayor a 30 días naturales. A partir de ese momento se hará un cargo de \$15.00 m.n IVA incluido por cada semana de almacenaje.
- Si el cliente no atiende a la recolección de su equipo, pasados 6 meses después de la fecha de notificación, CompuPAQ podrá disponer del equipo para recuperar el costo de diagnóstico y/o reparación, perdiendo el cliente cualquier derecho de reclamar su equipo total o parcialmente.
- Para servicios urgentes se hará un cargo adicional determinado por el departamento de soporte técnico.
- Para servicios a domicilio se hará un cargo adicional determinado por el departamento de soporte técnico.
- Todo servicio tendrá una garantía de 30 días naturales en mano de obra.
- La garantía de las refacciones será la ofrecida por el fabricante.
- El departamento de soporte técnico determinará si es necesario un anticipo y cual sería su porcentaje.
- El departamento de soporte técnico determinará el tiempo requerido para el diagnóstico y la reparación.
- Para tramitar una garantía será necesario presentar el comprobante de compra.

FIRMA DE CONFORMIDAD DEL CLIENTE

Hermenegildo Galeana #346 - Zona Centro - C.P 38000 - Celaya, Gto - Tel/Fax 01(461) 616-04-29 y 018007010631

Figura 4.6 Formato del reporte impreso

²¹ Figura realizada por el autor

4.3 ASIGNACIÓN DE REPORTE

En este módulo la función es el de asignar un equipo a un técnico específico para ingresar a este módulo es ir al menú movimiento y dar clic la opción asignar técnico y aparecerá la siguiente ventana ²².



Figura 4.7 Ventana de asignación de clientes

El botón imprimir realiza una impresión del datagridview listado del cliente.

En este módulo se muestran todos los reportes a los que no se le han asignado un técnico. Se selecciona el id o solo se escribe en la caja ID y se le da clic en aceptar y se mostrara la siguiente ventana.

²² Figura realizada por el autor

The image shows a software dialog box titled "Tecnico". It contains the following fields and values:

- Folio: 163
- Cliente: CLIENTE COMPUPAQ - USO INTERNO
- Fecha: 10/12/2007
- Hora: 12:08:23 P.M.
- Tipo de Equipo: MONITOR CRT
- Marca: HEWLETT PACKARD
- Tecnico (dropdown): LUIS ENRIQUE

Buttons: Aceptar, Cancelar

Figura 4.8 Ventana de selección de técnico

En el módulo técnico aparece el folio y esta filtrando la información mostrando el cliente la fecha y el equipo, en el combobox aparecen los técnico que se le pueden asociar a este folio. Solo se selecciona y se da clic en aceptar, la siguiente vez que quiera asignar el folio a un técnico el folio ya no volverá a parecer ya que tiene asignado un técnico ²³.

²³ Figura realizada por el autor

4.3 PRODUCTOS

En este módulo se da de alta un producto el que se da detalladamente las características del producto entre ellas está la descripción del producto, las características de este como el modelo, la ubicación en el almacén, unidad de acuerdo al peso y una fotografía del producto el stock es donde se ubica en el o los almacenes y los proveedores que lo pueden surtir ²⁴.

Figura 4.9 Ventana de productos

Para consultar los productos existentes solo es necesario dar clic en el botón de sku y aparecerá la siguiente pantalla.

²⁴ Figura realizada por el autor



Figura 4.10 Ventana de búsqueda de productos

En caso de que un proveedor no exista se presiona F11 y aparece el módulo de proveedor²⁵.

The screenshot shows a window titled 'Proveedor' with three tabs: 'Datos Generales', 'Datos Financieros', and 'Contactos'. The 'Datos Generales' tab is active, showing the following fields:

- ID:
- Nombre:
- Domicilio:
- Colonia: Ciudad:
- Estado: AGUASCALIENTES (dropdown) País: MEXICO (dropdown)
- Email: Fax:
- Teléfono: CURP:
- RFC: C.P.:

'Aceptar' and 'Cancelar' buttons are at the bottom right.

Figura 4.11 Ventana de proveedor

En este módulo se dan de alta los datos generales del proveedor, el módulo de proveedor es similar al módulo de cliente en el que contiene la pestaña de datos financieros en el que muestra los días de crédito, el saldo y acumulado, la lista de contactos tiene los nombre y los teléfono con la que se puede obtener información

²⁵ Figura realizada por el autor

de la distribuidora y poder obtener soporte o resolver dudas; nota: el módulo proveedor se encuentra en el menú principal movimientos

The screenshot shows a window titled 'Proveedor' with three tabs: 'Datos Generales', 'Datos Financieros', and 'Contactos'. The 'Datos Financieros' tab is active. It contains four input fields: 'Saldo' with a '\$' symbol, 'Crédito' with a '\$' symbol, 'Días de Crédito', and 'Acumulado' with a '\$' symbol. At the bottom right, there are two buttons: 'Aceptar' and 'Cancelar'.

Figura 4.12 Ventana de proveedor datos financieros

The screenshot shows the same 'Proveedor' window, but with the 'Contactos' tab active. It is divided into two sections: 'Contacto de Ventas' and 'Contacto de Pagos'. Each section has three input fields: 'Nombre', 'Teléfono', and 'Extensión'. Below the 'Contacto de Pagos' section is an 'Email' input field. At the bottom right, there are two buttons: 'Aceptar' and 'Cancelar'.

Figura 4.13 Ventana de proveedor contactos

4.4 REQUISICIÓN DE PRODUCTOS O SERVICIOS

La requisición es el medio por el cual da a conocer al cliente la necesidad para el mantenimiento de un equipo y la solución de este esto es con respecto a los datos del reporte. Y con respecto al producto son las refacciones o material que se necesita para la reparación de un equipo de cómputo.

Para entra a este módulo se entra al menú principal en la parte de requisición.

Al entrar nos mostrara una ventana en la que seleccionaremos el folio y a continuación damos clic en aceptar.



Figura 4.14 Ventana de selección de requisición



Figura 4.15 Ventana de requisición de refacciones y/o servicios

Aquí se muestra los datos del equipo mostrando el id del cliente y su nombre y la

información del equipo en el campo solución se pone el diagnostico que se le hizo y cuál fue su acción correctiva, en el caso de que el equipo necesite algún producto o refacción se da clic en el botón requisición.

Sku	Cantidad	Linea	Marca	Precio_Contado	Descripcion_pos	Importe
HDwDD0...	3	HD	WDD	541,2	D/DURO WESTERN DIGITAL 80GB SATA 7200RPM	1623,6
HDwDD0...	2	AC	3CM	484,74	D/DURO WSSTER DIGITAL 80GB IDE 7200RPM 2	969,48
*						0

SUB TOTAL \$ 2.591,00
IVA \$ 388,65
TOTAL \$ 2.979,65

Figura 4.11 Ventana de alta de requisición

En esta parte se irán agregando los productos necesarios para su reparación, solo falta dar doble clic en la parte dentro del cuadro blanco y aparecerá la siguiente ventana.

SKU	DESCRIPCION POS	LINEA	MARCA	NUMERO DE PARTE
		AC	3CM	
567842		AC	3CM	
BTSER001	BATERIA RECARGABLE SERCOM 12V / 7.0AH	BT	SER	
HDwDD001	D/DURO WSSTER DIGITAL 80GB IDE 7200RPM 2	AC	3CM	WD800BB
HDwDD002	D/DURO WESTERN DIGITAL 80GB SATA 7200RPM	HD	WDD	WD800JD

Figura 4.12 Ventana de búsqueda de productos

En esta parte la búsqueda se realiza mediante el campo sku o descripción solo falta teclear en cada caja y se irá filtrando la información hasta que aparezca el producto requerido. En caso de no encontrarse se presiona F11 y aparecerá la ventana de alta de producto, seleccionamos el producto y damos clic en el botón aceptar.

Ya realizado lo anterior la información aparecerá con los datos necesario y con su precio solo hay que teclear la cantidad que se necesita y realizara los cálculos automáticamente y dará el total, se puede imprimir esta información en papel o en formato pdf. Presionamos aceptar, y guardamos la información y los datos quedan registrados.

CONCLUSIONES

Las herramientas de computación deben ser capaces de facilitar las tareas del analista y diseñador de software. La mayoría de las empresas requieren de buena calidad en tiempos muy cortos, es por eso que no pueden esperar mucho tiempo para el desarrollo de una aplicación

Visual Studio. Net ofrece la capacidad de simplificar el desarrollo de sistemas debido a que es uno de los mejores lenguajes orientados a objetos. Solamente debe arrastrar el control y pegarlo en el formulario, después implantar el código asociado con este.

Con esto también los sistemas administradores de bases de datos han ayudado en gran medida en el desarrollo de sistemas ya que son potentes y flexibles un ejemplo de ellos es MySQL que proporciona un servidor de base de datos SQL muy rápido, multi-threaded, multi usuario y robusto. El servidor MySQL está diseñado para entornos de producción críticos, con alta carga de trabajo así como para integrarse en software para ser distribuido y utiliza la tecnología cliente – servidor en la que no es necesario instalar un software para conectarse a la base de datos con solo especificar la dirección ip o dns y el programa hace lo suyo. Y debido al Internet esto es posible ya que aunque este en lugares remotos nos podremos comunicar hacia la base de datos remota.

Como conclusión, cuando el analista diseñador de sistemas inicia un proyecto, es importante seguir una metodología de análisis y diseño, para seleccionar la más adecuada, debe experimentar con diferentes soluciones para escoger la que mejor se adapte a las necesidades ya que un sistema de información da varias ventajas internas (control de datos, ahorro de mano de obras, y una respuesta más rápida, y prevención ante situaciones inesperadas, disponibilidad inmediata de la información) y externas (como una mejor competitividad, ofrecer un servicio de calidad, dar una mejor presentación con el cliente y los proveedores).

Actualmente los sistemas de información junto con la tecnología son de gran importancia hasta para los negocios que van empezando para no quedar fuera con las competencias. Otras de las ayudas son las telecomunicaciones ya que rompen barreras y obtiene acceso a la información de una manera más rápida.

BIBLIOGRAFÍA

- Boehm, B., "A Spiral Model For Software Development And Enhancement",
- Jerry Fitzgerald, Stallings Warren. "Fundamentos de Análisis de Sistemas". México, Continental 1989.
- Senn James A., "Análisis y Diseño de Sistemas de Información", México, Mc Graw Hill, 1990.
- Victor Gerez, "Desarrollo y Administración de Programas de Computadora", México, Continental 1984.
- Steven Holzner "Microsoft Visual Basic .net ",800 E. 96th St., Indianapolis, Indiana, 46240 USA,Sams
- Jeff Ferguson,Brian Patterson, Jason Beres, "La Biblia de VB ", Editorial Anaya
- Internet www.mysql-hispano.org
- Internet: <http://www.elguruprogramador.com>
- Internet <http://www.solomanuales.org>
- Internet: <http://www.manuales.com>