



UNIVERSIDAD LATINA S.C.

LICENCIATURA EN INFORMÁTICA

**“IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO PARA EL
CONTROL DE INVENTARIO EN UNA EMPRESA DISTRIBUIDORA DE
GRANOS”**

T E S I S

**QUE PARA OBTENER EL TÍTULO DE
LICENCIADO EN INFORMÁTICA**

P R E S E N T A

APOLINAR PUNTOS DOMÍNGUEZ

ASESOR ACADÉMICO: LI. CLAUDIA AGUIRRE SALAZAR

MÉXICO, D.F.

ABRIL DE 2010



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

A MI PADRE

Agradezco a la vida y a Dios por tener este padre, que siempre me ha apoyado en todo momento a pesar de mis fracasos en la vida demostrándome su amor en cada momento. Gracias a su apoyo, consejos, esfuerzos, sacrificios he llegado hasta donde me encuentro ahorita. Gracias por todo.

A MI HERMANO

Gracias hermano por tu apoyo incondicional, a pesar de que soy el hermano mayor, tú me has puesto el ejemplo de cómo si se puede llegar a ser profesional y eso me motivó a terminar esta etapa de mi vida profesional.

A MI FAMILIA

Gracias a sus cuidados, consejos y amor he podido terminar ésta etapa de mi vida.

A MI ASESOR ACADÉMICO

Agradezco su apoyo, su paciencia y dedicación en la revisión de mi tesis, y gracias por compartir el conocimiento dentro del aula de clases.

ÍNDICE

INTRODUCCIÓN	1
JUSTIFICACIÓN	2
UNIDAD I MARCO TEÓRICO.....	3
1.1 Programa	4
1.2 Lenguaje de programación.....	4
1.2.1 Tipo de lenguajes de programación.....	4
1.3 Sistema.....	6
1.3.1 Elementos Del Sistema.....	7
1.3.2 Categorías De Los Sistemas.....	8
1.4 ¿Qué es Software?.....	14
1.4.1 Tipos de Software.....	15
1.5 Visual Basic. Net.....	18
1.6 Análisis Y Diseño De Sistemas.....	20
1.6.1 Análisis De Sistemas.....	20
1.6.2 Diseño De Sistemas.....	24
1.6.3 El Analista De Sistemas Moderno.....	24
1.7 Metodología ciclo de vida del desarrollo de sistemas.....	28
1.7.1 Identificación de problemas, oportunidades y objetivos.....	28
1.7.2 Determinación de los requerimientos de información.....	30
1.7.3 Análisis de las necesidades del sistema.....	32
1.7.4 Diseño del sistema recomendado.....	33
1.7.5 Desarrollo y documentación del software.....	35
1.7.6 Pruebas y mantenimiento del sistema.....	36
1.7.7 Implementación y evaluación del sistema.....	37
1.7.8 La importancia del mantenimiento.....	38
1.8 Evolución Del Desarrollo De Sistemas.....	40
1.8.1 Cualidades del software.....	41
1.8.2 Especificación del software.....	42
1.8.3 Verificación del software.....	43
1.8.4 Integración del sistema.....	43
1.8.5 Mantenimiento del software.....	45
1.9 Base de Datos.....	46
1.9.1 Modelo De Datos.....	47
1.9.2 Modelo Entidad-Relación.....	47
1.9.3 MySQL.....	54
UNIDAD II METODOLOGÍA DE APLICACIÓN.....	57
2.1 Definición del problema y objetivos.....	58
2.1.1 Objetivo General.....	58
2.1.2 Objetivo Específicos.....	58
2.2 Determinación de los requerimientos de información.....	58
2.3 Análisis de las necesidades del sistema.....	60
2.4 Diseño del sistema.....	62
2.4.1 Descripción de Tablas.....	63
2.5 Desarrollo y documentación del software.....	67
2.6 Pruebas y mantenimiento del sistema.....	81
2.7 Implementación y evaluación del sistema.....	82
CONCLUSIONES.....	83
BIBLIOGRAFÍA.....	85
ANEXO 1.....	86
ANEXO 2.....	87
ANEXO 3.....	88

ANEXO 4.....	116
ANEXO 5.....	117

INTRODUCCIÓN

Hoy en día el mundo está rodeado de tecnología que está evolucionando rápidamente, las computadoras han invadido la mayoría de las labores de los seres humanos, los sistemas informáticos son de gran importancia ya que gracias a éstos se han facilitado muchas de las tareas que ordinariamente nos tomaría gran tiempo realizarlas.

Los sistemas de información no son objetos estáticos, existen en un ambiente sujeto a cambios constantes. A medida que el ambiente cambia o se comprende mejor el sistema debe adaptarse a esos cambios o ir perdiendo utilidad, hasta acabar siendo desechado.

El sistema informático para el control de inventario en una empresa distribuidora de granos ha sido desarrollado bajo este marco conceptual y es por ello que se entregará con toda la documentación necesaria para operación.

Visto operacionalmente, permite carga de datos a través de tablas en MYSQL. También contempla las operaciones básicas de mantenimiento de estos catálogos dato por dato (Alta, Baja y Modificación). El sistema está programado con la interfaz gráfica de Visual Basic.Net para que sea más atractivo por el usuario y fácil de usar.

UNIDAD I MARCO TEÓRICO

1.1 Programa.

La programación de computadoras es el proceso de planificar una secuencia de instrucciones que ha de seguir una computadora. Un programa es la secuencia de instrucciones que indica las acciones que ha de ejecutar la computadora.

1.2 Lenguaje de programación.

Al igual que los lenguajes humanos, tales como el inglés o el español, los lenguajes de programación poseen una estructura (gramática o sintaxis) y un significado (semántica). Los lenguajes de computadoras tienen menos combinaciones aceptables que los lenguajes naturales, sin embargo, éstas combinaciones deben ser utilizadas correctamente; ello contrasta con los lenguajes naturales que se pueden utilizar aunque no siga las reglas gramaticales. Un lenguaje de programación es un conjunto de reglas, símbolos y palabras especiales que permiten construir un programa.

1.2.1 Tipo de lenguajes de programación.

Los tipos de lenguajes de programación los podemos clasificar en dos grandes grupos. Los lenguajes de programación de bajo nivel y los de alto nivel. El tipo de lenguaje de programación de bajo nivel depende totalmente de la máquina, en

este caso de la computadora u ordenador, éstos solo entienden el lenguaje binario o el código máquina, que consiste en ceros y unos. Es decir, que para realizar cualquier acción, solo utilizan este tipo de lenguaje de programación.

Los lenguajes de programación más cercanos a la arquitectura del hardware pertenecen al tipo de lenguajes de programación de bajo nivel. Mientras que los lenguajes de programación que se encuentran más próximos a los usuarios y programadores pertenecen al tipo de lenguajes de programación de alto nivel.

El tipo de lenguaje de programación de bajo nivel es totalmente dependiente de la computadora u ordenador, es decir que no podemos utilizarlo en cualquier otra. Este tipo de lenguaje de programación está prácticamente diseñado a la medida del hardware y aprovecha las características de este. Dentro de este tipo de lenguajes de programación podemos citar al lenguaje máquina y al lenguaje ensamblador.

Dentro del tipo de lenguajes de programación de alto nivel tenemos a todos aquellos lenguajes de programación que son más afines al lenguaje natural que al lenguaje máquina. Estos lenguajes de programación son completamente independientes de la arquitectura del hardware de la computadora u ordenador. Por lo que en general, un programa escrito con un lenguaje de programación de alto nivel lo podemos utilizar en cualquier otra computadora.

1.3 Sistema.

La palabra sistema es posiblemente el término más usado y abusado del léxico técnico. Hablamos de sistemas políticos, sistemas educativos, sistemas de fabricación, sistemas bancarios, etc. La palabra no nos dice gran cosa. Usamos el adjetivo para describir el sistema y para entender el contexto en que se emplea.

El diccionario Webster define sistema como un conjunto o disposición de cosas relacionadas de manera que forman una unidad o conjunto de hechos, principios, reglas, etc., clasificadas y dispuestas de manera ordenada mostrando un plan lógico de unión de las partes, un método o plan de clasificación o disposición, una manera establecida de hacer algo¹.

Un sistema basado en computadora es un conjunto o arreglo de elementos que están organizados para realizar un objetivo predefinido procesando información.

Un sistema informático como todo sistema es el conjunto de partes interrelacionadas, hardware, software y de Recurso Humano.

Un sistema informático es un conjunto de partes que funcionan relacionándose entre sí con un objetivo preciso. Sus partes son: hardware, software y las personas que lo usan.

¹ Roger S. Pressman, Ingeniería del software un enfoque practico, Mc Graw Hill, Cuarta Edición 1998.

1.3.1 Elementos Del Sistema.

Software: Programas de computadora, estructuras de datos y su documentación que sirven para hacer efectivo el método lógico, procedimiento o control requerido².

Hardware: Dispositivos electrónicos que proporcionan capacidad de calculo y dispositivos electromecánicos que proporcionan una función externa.

Personas: Usuarios y operadores del hardware y software.

Base de datos: Una gran colección de información organizada a la que se accede por medio del software.

Documentación: Manuales, formularios y otra información descriptiva que retrata el empleo y operaciones del sistema.

Procedimientos: Los pasos que definen el empleo específico de cada elemento del sistema o el contexto procedimental en que reside el sistema.

Los elementos se combinan de varias maneras para transformar la información. Una característica complicada de los sistemas basados en computadoras es que

² Roger S. Pressman, Ingeniería del software, Mc Graw Hill, Sexta Edición 2005.

los elementos que componen un sistema pueden también representar un macro elemento de un sistema aun más grande.

El macro elemento es un sistema basado en computadora que es parte de un sistema más grande basado en computadoras.

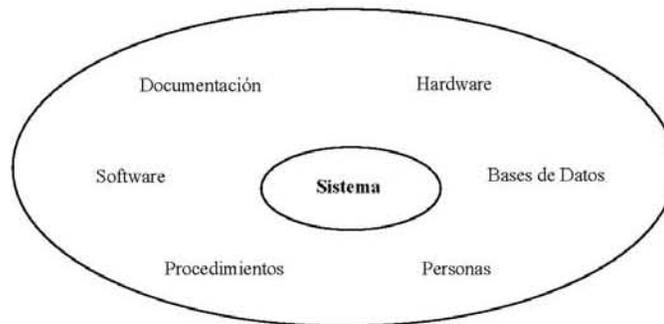


Figura 1. Elementos del sistema

1.3.2 Categorías De Los Sistemas.

1.3.2.1 Sistema de procesamiento de transacciones.

Los sistemas de procesamiento de transacciones (TPS, Transaction Processing Systems) son sistemas de información computarizados desarrollados para procesar gran cantidad de datos para transacciones rutinarias de los negocios, tales como nomina e inventario. Los TPS eliminan el tedio de las transacciones operacionales necesarias y reducen el tiempo que alguna vez se requirió para ejecutarlas manualmente, aunque la gente todavía debe alimentar datos a los sistemas computarizados.

Los sistemas de procesamiento de transacciones son sistemas que traspasan fronteras y que permiten que la organización interactúe con ambientes externos. Debido a que los administradores consultan los datos generados por el TPS para información al minuto acerca de lo que está pasando en sus compañías, es esencial para las operaciones diarias que estos sistemas funcionen lentamente y sin interrupción.

1.3.2.2 Sistemas de automatización de oficina y sistemas de manejo de conocimientos.

Al nivel de conocimiento de la organización hay dos clases de sistemas. Los sistemas de automatización de oficinas (OAS, Office Automation Systems) que dan soporte a los trabajadores de datos, quienes, por lo general, no crean un nuevo conocimiento sino que usan la información para analizarla y transformar datos, o para manejarla en alguna forma y luego compartirla o diseminarla formalmente por toda la organización y algunas veces más allá de ella. Los aspectos familiares de los OAS incluyen procesamientos de palabra, hojas de cálculo, editor de publicaciones, calendarización electrónica y comunicación mediante correo de voz, correo electrónico y videoconferencias.

Los sistemas de manejo de conocimiento (KWS, Knowledge Work Systems) dan soporte a los trabajadores profesionales, tales como científicos, ingenieros y

doctores, les ayudan a crear un nuevo conocimiento que contribuya a la organización o a toda la sociedad.

1.3.2.3 Sistemas de información gerencial.

Los sistemas de información gerencial (MIS, Management Information Systems) no reemplazan a los sistemas de procesamiento de transacciones, sino que todos los MIS incluyen procesamiento de transacciones. Los MIS son sistemas de información computarizada que trabajan debido a la interacción resuelta entre personas y computadoras. Requieren que las personas, el software (programas de computadora) y el hardware (computadoras, impresoras, etc.) trabajen al unísono. Los sistemas de información dan soporte a un espectro más amplio de tareas organizacionales que los sistemas de procesamiento de transacciones, incluyendo el análisis de decisiones y la toma de decisiones.

Para poder ligar la información, los usuarios de un sistema de información gerencial comparten una base de datos común. La base de datos guarda modelos que ayudan a los usuarios a interpretar y aplicar esos mismos datos. Los sistemas de información gerencial producen información que es usada en la toma de decisiones. Un sistema de información gerencial también puede llegar a unificar algunas de las funciones de información computarizada, aunque no exista como una estructura singular en ningún lugar del negocio.

1.3.2.4 Sistemas de apoyo a decisiones.

Una clase de más alto nivel en los sistemas de información computarizada son los sistemas de apoyo a decisiones (DSS, Decisión Support Systems). El DSS es similar al sistema de información gerencial tradicional en que ambos dependen de una base de datos como fuente. Un sistema de apoyo a decisiones se aparta del sistema de información gerencial tradicional en que enfatiza el apoyo a la toma de decisiones en todas sus fases, aunque la decisión actual todavía es del dominio del tomador de decisiones. Los sistemas de apoyo a decisiones están más hechos a la medida de la persona o grupo que los usa que los sistemas de información gerencial tradicionales.

1.3.2.5 Sistemas expertos e inteligencia artificial.

La inteligencia artificial (AI, Artificial Intelligence) puede ser considerada la meta de los sistemas expertos. El empuje general a la AI ha sido desarrollar máquinas que se comporten de forma inteligente. Dos caminos de la investigación de la AI son la comprensión del lenguaje natural y el análisis de la habilidad para razonar un problema y llegar a conclusiones lógicas. Los sistemas expertos usan los enfoques del razonamiento de la AI para resolver los problemas que les plantean los usuarios de negocios (y otros).

Los sistemas expertos son un caso muy especial de un sistema de información, cuyo uso ha sido factible para los negocios a partir de la reciente y amplia

disponibilidad del hardware y software tal como las microcomputadoras y sistemas expertos. Un sistema experto (también llamado un sistema basado en conocimiento) captura en forma efectiva y usa el conocimiento de un experto para resolver un problema en particular experimentado en una organización. Observe que a diferencia del DSS, que deja la decisión final al tomador de decisiones, un sistema experto selecciona la mejor solución a un problema o a una clase específica de problemas.

Los componentes básicos de un sistema expertos son la base de conocimiento, una máquina de inferencia que conecta al usuario con el sistema, procesando consultas por medio de lenguajes tales como SQL (Structured Query Language, lenguaje de consultas estructurado), y la interfaz de usuario. Personas llamadas ingenieros de conocimiento capturan la experiencia de los expertos, construyen un sistema de computadora donde incluyen el conocimiento del experto y luego lo implementan. Es totalmente posible que la construcción e implementación de sistemas expertos sea el trabajo futuro de muchos analistas de sistemas.

1.3.2.6 Sistemas de apoyo a decisiones de grupo.

Cuando los grupos necesitan trabajar juntos para tomar decisiones de grupo puede plantear una solución. Los sistemas de apoyo a decisiones de grupo (GDSS, Group Decisión Support System) son usados en cuartos especiales, equipados en varias configuraciones diferentes, que permiten que los miembros

del grupo interactúen con apoyo electrónico, frecuentemente en forma de software especializado y con una persona que da facilidades al grupo. Los sistemas para decisiones de grupo están orientados para reunir a un grupo, a fin de que resuelvan un problema con la ayuda de varios apoyos como votaciones, cuestionarios, aportación de ideas y creación de escenarios.

El software GDSS puede ser diseñado para minimizar el comportamiento negativo típico de un grupo, tal como la falta de participación debido al miedo a represiones por expresar un punto de vista no popular o conflictivo, dominación por miembros del grupo con voz dominante y la toma de decisiones de "pensamiento en grupo". Algunas veces los GDSS son tratados bajo el término más general trabajo colaborativo apoyado por computadoras (CSCW, Computer-Supported Collaborative Work Systems), que puede incluir el apoyo de software llamado "*groupware*" para la colaboración en equipo por medio de computadoras en red.

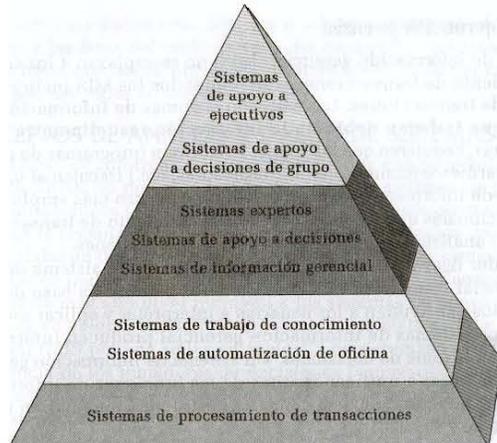


Figura 2. Categoría de los sistemas

1.4 ¿Qué es Software?

- Es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación.

- El software es una producción inmaterial del cerebro humano y tal vez una de las estructuras más complicadas que la humanidad conoce. De hecho, los expertos en computación aún no entienden del todo cómo funciona, su comportamiento, sus paradojas y sus límites. Básicamente, el software es un plan de funcionamiento para un tipo especial de máquina, una máquina virtual o abstracta. Una vez escrito mediante algún lenguaje de programación, el software

se hace funcionar en computadoras, que temporalmente se convierten en esa máquina para la que el programa sirve de plan. El software permite poner en relación al ser humano y a la máquina y también a las máquinas entre sí. Sin ese conjunto de instrucciones programadas, las computadoras serían objetos inertes, como cajas de zapatos, sin capacidad siquiera para mostrar algo en la pantalla.

- El software es un elemento lógico, en lugar de físico, de un sistema.

1.4.1 Tipos de Software

1.4.1.1 Software de sistemas.

El software de sistema es una colección de programas escritos para servir a otros programas. Algunos programas de sistemas (como los compiladores, editores y utilerías para la administración de archivos) procesan estructuras de información compleja pero determinada. Otras aplicaciones de sistemas (por ejemplo, componentes del sistema operativo, controladores, software de red, procesadores para telecomunicaciones) procesan datos indeterminados. En cada caso, el área de software de sistema se caracteriza por una interacción muy intensa con el hardware de la computadora; utilización por múltiples usuarios; operación concurrente que requiere la gestión de itinerarios, de compartición de recursos, y de procesos sofisticados; estructuras de datos complejas y múltiples interfaces externas.

1.4.1.2 Software de aplicación.

El Software de aplicación consiste en programas independientes que resuelven una necesidad de negocios específica. Las aplicaciones en esta área procesan datos empresariales o técnicos de forma que facilitan las operaciones de negocios o la toma de decisiones técnicas o de gestión. Además del procesamiento de datos convencionales, el software de aplicación se utiliza para controlar las funciones de negocios en tiempo real (por ejemplo, el procesamiento de transacciones en los puntos de venta y el control de procesos de manufactura en tiempo real.)

1.4.1.3 Software científico y de ingeniería.

El software científico y de ingeniería, que se caracterizaba por algoritmos "devoradores de números", abarca desde la astronomía hasta la vulcanología, desde el análisis de tensión automotriz hasta la dinámica orbital de los transbordadores espaciales, y desde la biología molecular hasta la manufactura automatizada. Sin embargo, las aplicaciones modernas dentro del área científica y de ingeniería se alejan en la actualidad de los algoritmos numéricos convencionales. El diseño asistido por computadora, la simulación de sistemas y otras aplicaciones interactivas han comenzado a tomar características de software en tiempo real e incluso de software de sistemas.

1.4.1.4 Software de línea de productos.

El software de línea de productos, diseñados para proporcionar una capacidad específica y la utilización de muchos clientes diferentes, se pueden enfocar en un nicho de mercado limitado (como en los productos para el control de inventarios) o dirigirse hacia los mercados masivos (por ejemplo, aplicaciones de procesadores de palabras, hojas de cálculo, gráficas por computadora, multimedia, entretenimiento, manejo de base de datos, administración de personal y finanzas en los negocios.)

1.4.1.5 Aplicaciones basadas en Web.

Las "Web Apps" engloban en espectro amplio de aplicaciones. En su forma más simple, las WebApps son apenas un poco más que un conjunto de archivos de hipertexto ligados que presentan información mediante texto y algunas gráficas. Sin embargo, a medida que el comercio electrónico y las aplicaciones B2B() adquieren mayor importancia, las WebApps evolucionan hacia ambientes computacionales sofisticados que no solo proporcionan características, funciones de cómputo y contenidos independientes al usuario final, sino que están integrados con bases de datos corporativas y aplicaciones de negocios.

1.4.1.6 Software de inteligencia artificial.

Este software utiliza algoritmos no numéricos en la resolución de problemas complejos que es imposible abordar por medio de un análisis directo. Las aplicaciones dentro de esta área incluyen la robótica, los sistemas expertos, el reconocimiento de patrones (imagen y voz), las redes neuronales artificiales, la comprobación de teoremas y los juegos de computadora.

1.5 Visual Basic.Net

Microsoft Visual Basic.Net es una importante mejora del popular sistema de desarrollo denominado Visual Basic, un producto que es utilizado por una numerosa cantidad de programadores en todo el mundo.

Es un lenguaje orientado a objetos y eventos que soporta la encapsulación, la herencia y el polimorfismo. Es una mejora a Visual Basic formando parte de Visual Studio y compartiendo el entorno de desarrollo con Microsoft Visual C++ .NET, Microsoft Visual C# .NET, etc.

El Entorno de Desarrollo recibe el nombre de Entorno de Desarrollo de Microsoft Visual Studio .NET. Este entorno es personalizable y contiene todas las herramientas necesarias para construir programas para Microsoft Windows.

El Entorno de Desarrollo contiene múltiples ventanas y múltiples funcionalidades y es por consecuencia llamado un entorno de desarrollo integrado (integrated development environment IDE). La ventana central es la ventana de diseño (Designer Window), la cual contiene el formulario a desarrollar.

La caja de herramientas (ToolBox) se localiza de lado izquierdo. En el extremo derecho tenemos la ventana de explorador de soluciones (Solution Explorer).

La ventana de propiedades (Properties window) contiene tres partes:

- La parte superior contiene un combo box que muestra el nombre y la clase del objeto seleccionado.
- La parte media contiene la lista de propiedades del objeto seleccionado, de lado derecho contiene un conjunto de cajas para ver y editar el valor de la propiedad seleccionada.
- La parte inferior es un cuadro descriptivo que proporciona una breve descripción de la propiedad seleccionada.

Es necesario tener instalado el Visual Studio .NET, al ejecutarlo se presenta una página de inicio, en caso de no presentarse entonces de clic en Help/Show Start Page. En esta página será posible establecer su perfil, por ejemplo identificarse como Desarrollador Visual Studio o más específico como Desarrollador Visual

Basic con lo cual Visual Studio configura de inmediato el entorno de desarrollo para programar en Visual Basic.

Para iniciar un nuevo proyecto, de clic en la opción Projects y clic en el botón [New Project], esta acción abre una ventana donde se indicará el archivo a abrir, los proyectos Visual Basic .NET tiene la extensión .vbproj. Una vez que abre el proyecto si la página de inicio estaba visible continuará así y en el Explorador de Soluciones (Solution Explorer) se cargan los archivos correspondientes al proyecto.

En Visual Basic .NET existen dos archivos:

- Un archivo de proyecto .vbproj, el cual contiene información específica para una determinada tarea de programación.
- Un archivo de solución .sln, el cual contiene información relacionada con uno o más proyectos. Este tipo de archivo puede administrar varios proyectos relacionados entre sí y son similares a los archivos de grupos de proyecto (.vbg) en Visual Basic 6

1.6 Análisis Y Diseño De Sistemas.

1.6.1 Análisis De Sistemas.

-Proceso de clasificación e interpretación de los hechos, diagnóstico de problemas y empleo de la información para recomendar mejoras al sistema³.

-Actividad que permite la fragmentación de un todo complejo, en sus partes constituyentes para realizar un estudio individual.

-Proceso de aplicar distintas técnicas y principios con el propósito de definir un dispositivo, proceso o sistema con los suficientes detalles como para permitir su realización física.

-Proceso de planificar, reemplazar o complementar un sistema organizacional. Es el cómo lo vamos a hacer.

1.6.1.1 Objetivos Del Análisis De Sistemas.

Identificación de necesidades. El primer paso del proceso de análisis del sistema afecta la identificación de la necesidad. El analista (especialista de sistemas) se reúne con el cliente y el usuario final (si es otro distinto que el cliente). El cliente puede ser el representante de una compañía, el departamento de mercadotecnia de la compañía del analista (cuando se está definiendo un producto), u otro departamento técnico (cuando hay que desarrollar un sistema

³ Kenneth E. Kendall, Análisis y Diseño de Sistemas, PRENTICE HALL, Tercera Edición 1997

interno). Como en la ingeniería de la información, la intención es entender los objetivos del producto y definir las metas necesarias para alcanzar esos objetivos.

Una vez que se han identificado las metas globales, el analista pasa a la evaluación de la información suplementaria: ¿Existe la tecnología para construir el sistema? ¿Qué recursos especiales de desarrollo y fabricación serán necesarios? ¿Qué límites se han puesto al presupuesto y a la planificación temporal? Si el nuevo sistema es un producto a desarrollar para venderlo a muchos clientes, se plantea además las siguientes cuestiones: ¿Cuál es el mercado potencial del producto? ¿Cómo es comparativamente este producto con los de la competencia? ¿Qué posición ocupa este producto en la línea general de producción de la compañía?

La información reunida durante el paso de identificación de necesidades es especificada de un documento conceptual del sistema. El documento conceptual original es preparado a veces por el cliente antes de reuniones con el analista. Invariablemente, la comunicación cliente-analista origina modificaciones en el documento.

Estudio de la viabilidad. Todos los proyectos son posibles: ¡si se tiene infinitos recursos y tiempo! Desgraciadamente, el desarrollo de un sistema o producto basado en computadora es muy probable que este plagado de escaseces de recursos y de fechas de entrega difíciles (o totalmente no realistas). Es necesario

y prudente evaluar la viabilidad de un proyecto cuanto antes. Se pueden evitar meses o años de esfuerzo, miles o millones de dólares y un bochorno profesional indeseable si se reconoce un sistema mal concebido en la pronta fase de definición.

La viabilidad y el analista de riesgo están relacionadas de muchas maneras. Si el riesgo del proyecto es alto, la viabilidad de producir software de calidad se reduce. Durante la ingeniería de producto, sin embargo, concentramos nuestra atención en cuatro áreas principales de interés:

- Viabilidad económica. Una evaluación del costo de desarrollo sopesando con los ingresos netos o beneficios obtenidos del sistema o producto desarrollado.

- Viabilidad técnica. Un estudio de función, rendimiento y restricciones que puedan afectar a la consecución de un sistema aceptable.

- Viabilidad operativa. Se refiere a que se cuente con capacidad del recurso humano disponible para realizar los procesos.

- Viabilidad legal. Determinar cualquier infracción, violación o responsabilidad legal en que se podría incurrir por el desarrollo del sistema.

- Alternativas. Una evaluación de los enfoques alternativos al desarrollo del sistema o producto.

No es necesario un estudio de viabilidad para sistemas en que la justificación económica es obvia, el riesgo técnico es bajo, se separan pocos problemas legales y no existe ninguna alternativa razonable. Sin embargo, si falla alguna de las condiciones anteriores, se deberían hacer un estudio del área en cuestión.

1.6.2 Diseño De Sistemas.

1. Proceso de aplicar distintas técnicas y principios con el propósito de definir un dispositivo, proceso o sistema con los suficientes detalles como para permitir su realización física.
2. Proceso de planificar, reemplazar o complementar un sistema organizacional. Es el cómo lo vamos a hacer.

1.6.3 El Analista De Sistemas Moderno

Los sistemas de información, a través de su papel central en la economía de la información, están llevando a cabo los cambios en cuatro aspectos fundamentales: 1) las personas trabajan de manera más inteligente, 2) un cambio global en el concepto industria, 3) tanto las ideas como la información están tomando mayor importancia que el dinero, y 4) las personas que trabajan con la información dominan la fuerza de trabajo.

El analista de sistemas es una persona de métodos que puede empezar con un problema complejo, desmenuzarlo lógicamente e identificar las soluciones razonables. El analista puede estudiar un sistema que está fallando y producir mejores alternativas. O bien, si se le dan varios objetivos, el analista puede idear medios sistemáticos para alcanzarlos. El analista de sistemas ve una situación de sistemas en términos de su alcance, sus objetivos y el marco de trabajo de organización.

Por ejemplo, el alcance de un proyecto de sistemas es el rango del proyecto mismo. En otras palabras, el rango es el área o la amplitud que el estudio abarcará. Los objetivos de un proyecto de sistemas son cualquier cosa que el analista está tratando de conseguir con el nuevo diseño de sistemas. El analista también ve un sistema en términos de la información con la que el sistema opera. Es muy importante que un analista de sistemas entienda completamente la información que se usa en el sistema actual y la información que se puede usar en el nuevo sistema propuesto.

Otro elemento muy importante que el analista debe considerar al ver una situación de sistemas, es dónde están involucradas las personas en la operación del sistema y qué es lo que hacen. El ingrediente básico de un sistema es la gente. Si el analista ignora a la gente, el sistema no será tan eficiente como podría serlo. El equipo empleado en un sistema, las formas y los reportes usados en éste y las responsabilidades asignadas a cada departamento involucrado en el sistema, son

todos ellos elementos importantes para el analista de sistemas. Un analista que ignore cualquiera de los elementos anteriores, podría hacer peligrar la posibilidad de un buen diseño de sistemas. La responsabilidad primaria de los analistas de sistemas, por consiguiente, no es sólo desarrollar sistemas que satisfagan los objetivos y las metas de toda la compañía, sino también satisfacer los objetivos y las metas de los departamentos individuales que están involucrados con el sistema específico que se está diseñando. Una buena solución de sistemas satisface todos sus objetivos. Una solución excelente de sistemas satisface todos sus objetivos y además, puede ajustarse plenamente dentro del marco de trabajo de la organización y tomar ventaja de los recursos humanos existentes.

Las responsabilidades de los analistas, sin embargo, así como su denominación dentro de una empresa, cambian de una organización a otra. Las funciones más comunes asignadas a los analistas de sistemas son:

-Análisis de Sistemas. En este caso la única responsabilidad del analista es conducir estudios de sistemas para detectar hechos relevantes relacionados con la actividad de la empresa. La función más importante en este caso es reunir información y determinar los requerimientos. Los analistas no son responsables del diseño de sistemas (Analista de información).

-Análisis y Diseño de Sistemas. Además de llevar a cabo el estudio completo de los sistemas, el analista tiene la responsabilidad adicional de diseñar el nuevo

sistema. Los que se responsabilizan tanto del análisis como del diseño trabajan en menos proyectos que los analistas de información pero invierten más tiempo en ellos. (Diseñadores de sistemas, diseñadores de aplicaciones).

-Análisis, Diseño y Programación de Sistemas. El analista conduce la investigación de sistemas, desarrolla las especificaciones de diseño y escribe el software necesario para implantar el diseño. (Analista programador).

De lo anterior no se debe concluir que el papel de algunos analistas es superior o inferior al de otros ya que es el tamaño de la organización el que, con bastante frecuencia, dicta la naturaleza del trabajo del analista.

Resumiendo, podemos decir, que el **Analista de Sistemas** es el profesional que tiene la responsabilidad de cumplir las siguientes tareas:

1. Guía las actividades del grupo de desarrollo a lo largo de las diversas fases.
2. Entiende los problemas desde el punto de vista del usuario.
3. Identifica y define los problemas del cliente.
4. Desarrolla soluciones cuyo costo/beneficio sea conveniente.
5. Identifica, evalúa y recomienda alternativas para la implementación del proyecto.
6. Identifica todas las áreas que requieren de investigación adicional, realiza dicha investigación y documenta sus resultados.

7. Verifica que el equipo de trabajo mantenga los objetivos iniciales del proyecto.
8. Identifica las necesidades del usuario y las comunica al grupo de desarrollo de forma que se aclare la definición del problema mediante la verificación de especificaciones.
9. Encauza al equipo hacia los estándares propios de la organización y hacia el procedimiento del ciclo de vida del proyecto.

1.7 Metodología ciclo de vida del desarrollo de sistemas.

El ciclo de vida del desarrollo de sistemas (SDLC por sus siglas en inglés.) El SDLC es un enfoque por fases del análisis y diseño que sostiene que los sistemas son desarrollados de mejor manera mediante el uso de un ciclo específicos de actividades del analista y del usuario.

Los analistas no están de acuerdo con que tantas fases exactas hay en el ciclo de vida del desarrollo de sistemas, pero, por lo general alaban su enfoque organizado.

1.7.1 Identificación de problemas, oportunidades y objetivos.

En la primera fase del ciclo de vida del desarrollo de sistemas el análisis tiene que ver con la identificación de problemas, oportunidades y objetivos. Esta etapa es crítica para el éxito del resto de proyecto, debido a que nadie quiere desperdiciar el tiempo subsecuente resolviendo el problema equivocado.

La primera fase requiere que el analista observe honestamente lo que esta sucediendo en un negocio. Luego, junto con los demás miembros de la organización, el analista hace resaltar los problemas. Frecuentemente éstos ya han sido vistos por los demás, y son la razón por la cual el analista fue llamado inicialmente.

Las oportunidades son situaciones que el analista considera que pueden ser mejoradas por medio del uso de sistemas de información computarizados. El aprovechar las oportunidades puede permitir que el negocio gane un avance competitivo o pongan un estándar de la industria.

La identificación de los objetivos es también un componente importante de la primera fase. En primer lugar, el analista debe descubrir lo que esta tratando de hacer el negocio. Luego será capaz de ver si algún aspecto de la aplicación de sistemas de información puede ayudar para que el negocio alcance sus objetivos atacando problemas específicos u oportunidades.

Las personas involucradas en la primera fase son los usuarios, analistas y administradores de sistemas que coordinan el proyecto. Las actividades de esta fase consiste en entrevistas a los administradores de los usuarios, sumarización del conocimiento obtenido, estimación del alcance del proyecto y documentación de los resultados. La salida de esta fase es un estudio de factibilidad que contiene una definición del problema y la sumarización de los objetivos. Luego los administradores deben tomar una decisión para ver si continúan con el proyecto

propuesto. Si el grupo de usuarios no tiene los suficientes fondos en su presupuesto y desea atacar problemas que no están relacionados, o los problemas no requieren un sistema de cómputo, puede ser recomendada una solución manual y el proyecto de sistemas ya no continúa.

1.7.2 Determinación de los requerimientos de información.

La siguiente fase a la que entra el analista es la de la determinación de los requerimientos de información para los usuarios particulares involucrados. Entre las herramientas utilizadas para definir los requerimientos de información en el negocio se encuentran: muestreo e investigación de los datos relevantes, entrevista, cuestionarios, el comportamiento de los tomadores de decisiones y su ambiente de oficina y hasta la elaboración de prototipos.

En ésta fase el analista está esforzándose por comprender que información necesitan los usuarios para realizar su trabajo. Se puede ver que varios de los métodos para determinar los requerimientos de información involucran la interacción directa con los usuarios. Esta fase sirve para formar la imagen que el analista tiene de la organización y sus objetivos.

Algunas veces solamente se completan las dos primeras fases del ciclo de vida del desarrollo de sistemas. Este tipo de estudio puede tener diferentes propósitos, y es realizado típicamente por un especialista llamado analista de información (IA). Las personas involucradas en esta fase son los analistas y los usuarios, típicamente los administradores de las operaciones y los trabajadores de las operaciones. El analista de sistemas necesita saber los detalles de las funciones actuales del sistema: quién (las personas que están involucradas), qué (la actividad del negocio), dónde (el ambiente donde se lleva a cabo el trabajo), cuándo (en qué momento) y cómo (de que manera se desarrollan los procedimientos actuales) del negocio bajo estudio. El analista debe preguntar porqué el negocio usa el sistema actual. Puede haber muy buenas razones para desarrollar el negocio usando los métodos actuales, y deben ser considerados cuando se diseña cualquier sistema nuevo.

Sin embargo, si la razón de las operaciones actuales es que “siempre se han hecho así”, el analista puede desear la mejora de los procedimientos. La reingeniería de procesos del negocio puede ayudar a enmarcar un enfoque para volver a pensar en el negocio en forma creativa. Al término de esta fase, el analista debe comprender el por qué de las funciones del negocio y tener información completa sobre las personas, objetivos, datos y procedimientos involucrados.

1.7.3 Análisis de las necesidades del sistema.

La siguiente fase que realiza el analista de sistemas involucra el análisis de las necesidades del sistema. Nuevamente, herramientas y técnicas especiales ayudan para que el analista haga las determinaciones de los requerimientos. Una herramienta de éstas es el uso de diagramas de flujo de datos para diagramar la entrada, proceso y salida de las funciones del negocio en forma gráfica estructurada. A partir de los diagramas de flujo de datos se desarrolla un diccionario de datos, que lista todos los conceptos de datos usados en el sistema, así como sus especificaciones, si son alfanuméricos y que tanto espacio ocupan cuando se imprimen.

Durante esta fase el analista de sistemas también analiza las decisiones estructuradas que se hacen. Las decisiones estructuradas son aquellas para las que pueden ser determinadas las condiciones como alternativas de condición, acciones y reglas de acción. Hay tres métodos principales para el análisis de decisiones estructurales: lenguaje estructurado, tablas de decisión y árboles de decisión.

No todas las decisiones de la organización son estructuradas, pero todavía es importante que el analista de sistemas las comprenda. Las decisiones semiestructuradas (decisiones tomadas bajo riesgo) son sustentadas frecuentemente por los sistemas de apoyo a decisiones. Cuando se analizan

decisiones semiestructuradas, el analista examina las decisiones con base en el grado de complejidad del problema y la cantidad de criterios considerados cuando se toma la decisión.

El analista de las decisiones de criterios múltiples (decisiones en las que deben ser balanceados muchos factores) también parte de esta fase. Se dispone de muchas técnicas para el análisis de decisiones de criterios múltiples, incluyendo el proceso de compromiso y el uso de métodos ponderados.

En este punto del ciclo de vida del desarrollo de sistemas, el analista prepara una propuesta de sistema que resume lo que ha sido encontrado, proporciona análisis de costo/beneficio de las alternativas y hace recomendaciones sobre lo que debe ser hecho (en caso de haberlo). Si alguna de las recomendaciones es aceptable para la administración, el análisis continúa sobre ese curso. Cada problema de sistemas es único y nunca hay una sola solución correcta. La manera en que se formula una solución o recomendación depende de la capacidad y entrenamiento profesional individual de cada analista.

1.7.4 Diseño del sistema recomendado.

En esta fase del ciclo de vida del desarrollo de sistemas, el analista usa la información recolectada anteriormente para realizar el diseño lógico del sistema de

información. El analista diseña procedimientos precisos para la captura de datos, a fin de que los datos que van a entrar al sistema de información sean correctos. Además, el analista también proporciona entrada efectiva para el sistema de información mediante el uso de técnicas para el buen diseño de formas y pantallas.

Parte del diseño lógico del sistema de información es diseñar la interfaz de usuario. La interfaz conecta al usuario con el sistema y es, por lo tanto, extremadamente importante. Ejemplo de interfaces de usuario incluyen un teclado para introducir preguntas y respuestas, menús en pantalla para elegir comandos del usuario y un ratón para seleccionar opciones.

La fase de diseño también incluye el diseño de archivos o bases de datos que guardaran la mayor parte de los datos necesarios para los tomadores de decisiones de la organización. Una base de datos bien organizada es la base para todos los sistemas de información. En esta fase, el analista también trabaja con los usuarios para diseñar la salida (ya sea en pantalla o impresa) que satisfaga sus necesidades de información.

Por ultimo el analista debe diseñar procedimientos de control y respaldo para proteger al sistema y a los datos y producir paquetes de especificaciones de programa para los programadores. Cada paquete debe contener diseños de entrada y salida, especificaciones de archivos y detalles de procesamiento, y

también puede incluir árboles o tablas de decisión, diagramas de flujo de datos, un diagrama de flujo del sistema y los nombres y funciones de cualesquier de las rutinas de código que ya hayan sido escritas.

1.7.5 Desarrollo y documentación del software.

En la quinta fase del ciclo de vida del desarrollo de sistema el analista trabaja con los programadores para desarrollar cualquier software original que se necesite. Algunas de las técnicas estructuradas para el diseño y documentación de software incluyen diagramas estructurados, el método HIPO, diagramas de flujo, diagramas Nassi-Schneiderman y Warnier-Orr y pseudocódigo. El analista de sistemas usa uno o más de estos dispositivos para comunicar al programador lo que necesita ser programado.

Durante esta fase, el análisis también trabaja con los usuarios para desarrollar documentación efectiva para el software, incluyendo manuales de procedimientos. La documentación le dice al usuario la manera de usar el software y también que hacer si se suceden problemas con el software.

Los programadores tienen un papel principal en esta fase conforme diseñan, codifican y eliminan errores de sintaxis de los programas de computadora. Si el programa va a ser ejecutado en un ambiente de macrocomputadora, se debe crear el lenguaje de control de trabajos (JCL). Para asegurar la calidad, un

programador puede realizar ya sea un diseño o un ensayo del código, explicado las partes complejas del programa a un equipo de otros programadores.

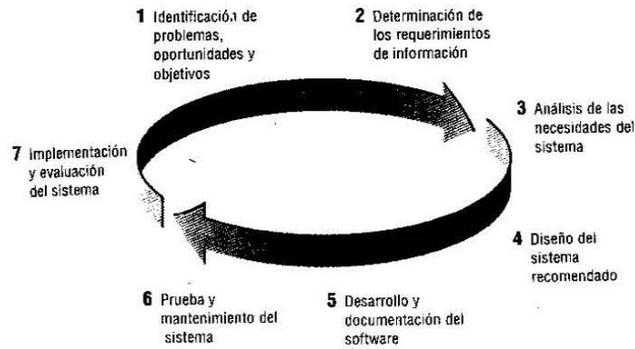


Figura 3. Ciclo de vida del desarrollo de sistemas.⁴

1.7.6 Pruebas y mantenimiento del sistema.

Antes de que pueda ser usado, el sistema de información debe ser probado. Es mucho menos costoso encontrar problemas antes de que el sistema sea entregado a los usuarios. Algunas de las pruebas son realizadas por los programadores solos, y otras por los analistas de sistemas junto con los programadores. Primero se ejecuta una serie de pruebas para que destaquen los

⁴ Kenneth E. Kendall, Análisis y Diseño de Sistemas, PRENTICE HALL, Tercera Edición 1997

problemas con datos de ejemplo y eventualmente con datos reales del sistema actual.

El mantenimiento del sistema y de su documentación comienza en esta fase y es efectuado rutinariamente a lo largo de la vida del sistema de información. Mucho del trabajo rutinario del programador consiste en el mantenimiento, ya que los negocios gastan gran cantidad de dinero en dicho mantenimiento. Muchos de los procedimientos sistemáticos que emplea el analista a lo largo del ciclo de vida del desarrollo del sistema pueden ayudar a asegurar que el mantenimiento se mantenga al mínimo.

1.7.7 Implementación y evaluación del sistema.

En ésta fase el analista ayuda a implementar el sistema de información. Esto incluye el entrenamiento de los usuarios para que manejen el sistema. Algún entrenamiento es hecho por los proveedores, pero la supervisión del entrenamiento es responsabilidad del analista de sistema. Adicionalmente, el analista necesita un plan para una conversión suave del sistema antiguo al nuevo. Este proceso incluye la conversión de archivos de formatos antiguos a nuevos o la construcción de una base de datos, la instalación de equipo y la puesta del nuevo sistema en producción.

La evaluación se muestra como parte de esta fase final de ciclo de vida del desarrollo del sistema, principalmente para efectos de de discusión. De hecho, la evaluación se realiza durante cada fase. Un criterio principal que debe ser satisfecho es si los usuarios pretendidos ya están usando el sistema.

Deben hacerse notar que a veces los sistemas trabajan en forma cíclica.

Cuando un analista termina una fase del desarrollo de sistema y pasa a la siguiente, el descubrimiento de un problema puede obligar a que el analista regrese a la fase anterior y modifique el trabajo que allá hizo. Por ejemplo, durante la fase de prueba el programador puede descubrir que el programa no trabaja correctamente, ya sea debido a que no se escribió código para apoyar determinadas partes del diseño del sistema o aquel diseño fue incompleto. En cualquier caso deben ser modificados los programas, y el analista puede tener que cambiar algunos de los materiales del diseño del sistema. A su vez, esto puede necesitar que el analista se reúna con el usuario y vuelva a investigar como funciona una actividad específica del negocio.

1.7.8 La importancia del mantenimiento.

Después que el sistema esta instalado se le debe dar mantenimiento, esto significa que los programas de computadora deben ser modificados y mantenidos actualizados.

El mantenimiento se realiza por dos razones. La primera de éstas es para corregir errores de software. Sin importar que tan completamente se pruebe el sistema, se deslizan errores en los programas de computadora. Los errores del software comercial para microcomputadoras son a veces documentados como anomalías conocidas, y son corregidos cuando son lanzadas nuevas versiones del software o versiones intermedias. En el software personalizado los errores deben ser corregidos conforme son detectados.

La otra razón para realizar el mantenimiento del sistema es para mejorar las capacidades del software en respuesta a las necesidades organizacionales cambiantes y, por lo general, involucran algunas de las siguientes tres situaciones:

- Los usuarios frecuentemente solicitan características adicionales después de que se familiarizan con el sistema de cómputo y sus capacidades. Estas características solicitadas pueden ser tan simples como el desplegado de totales adicionales en un reporte o tan complicadas como el desarrollo de nuevo software.
- El negocio cambia a través del tiempo. Se debe modificar el software para abarcar cambios tales como nuevos requerimientos de reportes gubernamentales o corporativos, la necesidad de producir nueva información para clientes.

- El hardware y software están cambiando a un ritmo acelerado. Un sistema que usa tecnología antigua puede ser modificado para usar las capacidades de una tecnología nueva. Un ejemplo de tal cambio es el reemplazo de una terminal de macrocomputadora con una estación de trabajo de microcomputadoras, o una microcomputadora con una computadora de escritorio.

Resumiendo, el mantenimiento es un proceso continuo a largo de ciclo de vida de un sistema de información. Después de que es instalado el sistema de información, el mantenimiento por lo general toma la forma de corrección de errores de programa no detectados previamente. Una vez que son corregidos, el sistema alcanza un estado estable proporcionando servicios confiables a sus usuarios. El mantenimiento durante este periodo puede consistir en la eliminación de unos cuantos errores no detectados anteriormente y la actualización del sistema con unas cuantas mejoras menores. Sin embargo, conforme pasa el tiempo y cambia el negocio y la tecnología, los esfuerzos de mantenimiento se incrementan gradualmente.

1.8 Evolución Del Desarrollo De Sistemas

Hace veinte años, menos del 1 por ciento del público podía describir inteligentemente lo que significa Software de Computadoras. Para poder

comprender lo que es el software, es importante examinar las características que lo hacen diferente de otras cosas que los hombres pueden construir. Cuando se construye el hardware, el proceso creativo humano (análisis, diseño, construcción, prueba) se traduce finalmente en una forma física.

El software es un elemento lógico del sistema. Por tanto, el software tiene unas características considerablemente distintas a las del hardware:

1.8.1 Cualidades del software.

Las características o cualidades del software se dividen en internas y externas. Las internas son aquellas perceptibles para los desarrolladores del sistema las externas son aquellas importantes y visibles para el usuario.

1.8.1.1 Cualidades Internas del Software

- Correctez: habilidad para desarrollar las tareas de manera exacta, como fueron definidas en los requerimientos y especificaciones.
- Robustez: habilidad para funcionar en situaciones anormales.
- Extendibilidad: habilidad para adaptarse a cambios en las especificaciones.
- Reusabilidad: habilidad para ser rehusado completamente o en partes.
- Compatibilidad: habilidad para usarse en conjunción con otro software ya existente.

1.8.1.2 Cualidades Externas del Software

- Eficiencia: que el software haga uso eficiente de los recursos del hardware.
- Portabilidad: que pueda implementarse en diversas plataformas.
- Verificabilidad: que detecte fácilmente errores.
- Integración: que sea seguro.
- Facilidad de uso: que el usuario pueda aprender a manejarlo fácilmente.

1.8.2 Especificación del software.

Al igual que con las características de un sistema de información, el software también debe ser diseñado con cuidado. El diseño de sistemas incluye la formulación de las especificaciones de software.

Estas especificaciones establecen las funciones de entrada, salida y procesamiento así como los algoritmos necesarios para efectuarlas. Los módulos de software junto con las rutinas, se enfocan sobre lo que cada función realiza; asimismo, se especifican los procedimientos necesarios para llevar a cabo dichas funciones. La selección de lenguajes de programación, paquetes de software y utilerías se efectúa durante el proceso de diseño lógico y las recomendaciones se incluyen como parte de las especificaciones del software.

1.8.3 Verificación del software.

La verificación tiene la intención de hallar errores. Se lleva a cabo ejecutando un programa en un ambiente simulado.

Cuando los sistemas comerciales se desarrollan con la intención explícita de distribuirlos a través de terceros para su venta, o comercializarlos por medio de oficinas de la propia compañía, primero deben pasar por la verificación, a veces llamada prueba alfa. La retroalimentación de la fase de validación generalmente produce cambios en el software para resolver los errores y fallas que se descubren. Se elige un conjunto de instalaciones usuarias que ponen a trabajar el sistema en un ambiente real. Estas instalaciones de *prueba beta* usan el sistema en las actividades cotidianas; procesan transacciones en directo y producen salidas normales del sistema. El sistema está a prueba en toda la extensión de la palabra, excepto que los usuarios están advertidos de que están usando un sistema que puede fallar. Sin embargo, las transacciones que se están procesando y las personas que usan el sistema son reales.

1.8.4 Integración del sistema.

La estructura del programa define la jerarquía de control, independientemente de las decisiones y secuencias de procesamiento. El procedimiento de software se enfoca sobre los detalles de procesamiento de cada módulo individualmente. El

procesamiento debe dar una especificación precisa del procesamiento, incluyendo secuencia de sucesos, puntos de decisiones exactos, operaciones repetitivas e incluso organización/estructura de datos.

Por supuesto, existe una relación entre estructura y procedimiento. El procesamiento indicado por cada módulo debe incluir una referencia a todos los módulos subordinados del módulo que se describe.

La importancia del concepto de modularidad en el software se ha tenido en cuenta desde hace casi cuatro décadas. La arquitectura implica modularidad; esto es, el software se divide en elementos con nombres y direcciones separadas, llamados módulos, que se integran para satisfacer los requerimientos del problema.

Esto conduce a una conclusión de "divide y vencerás" –es más fácil resolver un problema complejo cuando se le parte en trozos manejables. Es importante observar que un sistema puede ser diseñado modularmente, incluso si su implementación debe ser "monolítica".

Hay situaciones (por ejemplo, software de tiempo real, software de microprocesadores) en las cuales una velocidad relativamente mínima y sobrecarga de memoria introducida por los subprogramas es inaceptable. En tales situaciones, el software puede y debe ser diseñado considerando la modularidad como la filosofía principal. El código puede desarrollarse en línea. Aunque el código fuente del programa pueda no parecer modular a primera vista,

la filosofía ha sido mantenida y el programa tendrá las ventajas de un sistema modular.

1.8.5 Mantenimiento del software.

La corrección de errores sobre la marcha es un aspecto de mantenimiento. Lientz y Swanson [1980] encontraron que en esto consistían más del 21 por ciento de los esfuerzos de mantenimiento global en las organizaciones estadounidenses de proceso de datos. El mantenimiento también involucra la modificación de un sistema para reflejar cambios en el hardware, modificaciones para apresurar ciertos aspectos operacionales o modificaciones para reflejar cambios en los requerimientos del usuario final del sistema.

El mantenimiento de software es un problema primordial en muchas organizaciones, el trabajo que se hace en la mayoría de las organizaciones de desarrollo de sistemas se asocia con la revisión, modificación, conversión, mejoramiento o corrección de errores en algún programa de computadora que otra persona escribió hace mucho.

El problema es peor aún. Si fuera simplemente un caso de que el software fuese malo, se podría considerar desecharlo o reemplazarlo. Pero muchas organizaciones jamás han capitalizado su software (los costos se determinan cada

año), y sus políticas de administración y de negocios hacen que se vuelva prohibitivamente caro reemplazar los sistemas antiguos. Y existe un problema aún más fundamental: en la mayoría de las organizaciones, no existe una descripción coherente de lo que se supone que los sistemas deben hacer. La documentación existente suele ser obsoleta y confusa. En cualquier caso, proporciona, cuando más, alguna idea de cómo funciona el software, pero no de cuál es su propósito fundamental, o qué política de negocios se supone debe realizar.

Por eso, aunque se pueda argumentar que el mantenimiento es enteramente función de la implantación (es decir, algo que preocupa a los programadores), es imposible mantener un sistema si no existe un modelo preciso y actualizado de sus requerimientos. Esto, a fin de cuentas, es labor del analista.

1.9 Base de Datos

- Una base de datos, en su definición más sencilla, es una colección de archivos relacionados⁵.

- Es el conjunto de datos que pertenece al mismo contexto almacenados sistemáticamente para su posterior uso.

⁵ Abraham Silberschatz, Henry F. Korth, S. Sudarshan, Fundamentos de Base de datos, Mc Graw Hill, Cuarta Edición 2002

- Es el registro agrupando la información asociada a un elemento de un conjunto y esta compuesta por campos.
- Es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulan ese conjunto (entre si) de datos.
- Es una colección de información organizada de forma que un programa de la computadora pueda seleccionar rápidamente los fragmentos de datos que necesite. Una base de datos es un sistema de archivos electrónico.

1.9.1 Modelo De Datos

Un modelo de datos es una colección de herramientas conceptuales para la descripción de datos, relaciones entre datos, semántica de los datos y restricciones de consistencia.

1.9.2 Modelo Entidad-Relación

El modelo de datos entidad-relación (E-R) esta basado en una percepción del mundo real consistente en objetos básicos llamadas entidades y de relaciones

entre estos objetos. Se desarrolló para facilitar el diseño de base de datos permitiendo la especificación de un esquema de la empresa que representa la estructura lógica completa de una base de datos. El modelo de datos E-R es uno de los diferentes modelos de datos semánticos; el aspecto semántico del modelo yace en la presentación del significado de los datos. El modelo E-R es extremadamente útil para hacer corresponder los significados e interacciones de las empresas del mundo real con un esquema conceptual. Debido a esta utilidad, muchas herramientas de diseño de base de datos se basan en los conceptos del modelo E-R.

Hay tres nociones básicas que emplean el modelo de datos E-R conjuntos de entidades, conjuntos de relaciones y atributos.

1.9.2.1 Conjunto de entidades.

Una entidad es una cosa u objeto en el mundo real que es distinguible de todos los demás objetos. Por ejemplo, cada persona en un desarrollo es una entidad. Una entidad tiene un conjunto de propiedades, y los valores para algún conjunto de propiedades pueden identificar una entidad de forma unívoca.

Un conjunto de entidades es un conjunto de entidades del mismo tipo que comparten las mismas propiedades, o atributos.

Una entidad se representa mediante un conjunto de atributos. Los atributos describen propiedades que posee cada miembro de un conjunto de entidades. La asignación de un atributo para un conjunto de entidades expresa que la base de datos almacena información similar concerniente a cada entidad del conjunto de entidades; sin embargo, cada entidad puede tener su propio valor para cada atributo.

Un atributo, como se usa en el modelo E-R, se puede caracterizar por los siguientes tipos de atributo:

- Atributos simples y compuestos. Los atributos simples no están divididos en subpartes por ejemplo id-cliente, nombre del cliente, calle del cliente, los atributos compuestos, en cambio, se pueden dividir en subpartes (es decir, en otros atributos). Por ejemplo, nombre-cliente podría estar estructurado como un atributo compuesto consistente en nombre, primer-apellido y segundo-apellido. Usar atributos compuestos en un esquema de diseño es una buena elección si el usuario desea referirse a un atributo completo en algunas ocasiones y, en otras, a algún componente del atributo.
- Atributos monovalorados y multivalorados. Los atributos que se han especificado en los ejemplos tienen todos un valor sólo para una entidad concreta. Por ejemplo, el atributo número-préstamo para una entidad préstamo específico, referencia a un único número de préstamo. Tales

atributos se llaman monovalorados. Puede haber ocasiones en las que un atributo tiene un conjunto de valores para una entidad específica. Considere un conjunto de entidades empleado con el atributo número-teléfono. Cualquier empleado particular puede tener cero, uno o más números de teléfono. Este tipo de atributo se llama multivalorado. En ellos, se pueden colocar apropiadamente límites inferior y superior en el número de valores en el atributo multivalorado.

- Atributos derivados. El valor para este tipo de atributo se puede derivar de los valores de otros atributos o entidades relacionados. Por ejemplo, sea el conjunto de entidades cliente que tiene un atributo préstamos que presentan cuantos préstamos tiene un cliente en le banco. Ese atributo se puede derivar contando el número de entidades préstamo asociado con ese cliente.

Un atributo toma un valor de nulo cuando una entidad no tiene un valor para un atributo. El valor nulo también puede indicar no aplicable, es decir, que el valor no existe para la entidad. Por ejemplo, una persona puede no tener segundo nombre de pila. Nulo puede también designar que el valor de un atributo es desconocido.

1.9.2.2 Conjuntos de relaciones.

Una relación es una asociación entre diferentes entidades. Por ejemplo, se puede definir una relación que asocie al cliente López con el préstamo P-15. Esta relación específica que López es un cliente con le préstamo numero P-15.

Un conjunto de relaciones es un conjunto de relaciones del mismo tipo.

1.9.2.3 La correspondencia de cardinalidades.

La correspondencia de cardinalidades, o razón de cardinalidad, expresar el número de entidades a las que otra entidad puede estar asociada vía un conjunto de relaciones.

La correspondencia de cardinalidades es la más útil describiendo conjuntos relacionales binarias, aunque ocasionalmente contribuye a la descripción de conjuntos de entidades.

Para un conjunto de relaciones binarias R entre los conjuntos de entidades A y B , la correspondencia de cardinalidades debe ser una de las siguientes:

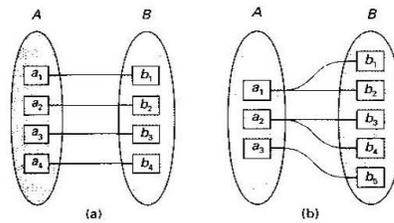


Figura 4. Correspondencia de cardinalidades (a) uno a uno. (b) uno a varios.

- Uno a uno. Una entidad en A se asocia con a lo sumo una entidad en B, y una entidad en B se asocia con a lo sumo una entidad en A (véase la Figura 4a).
- Uno a varios. Una entidad en A se asocia con cualquier número de entidades en B (ninguna o varias). Una entidad en B, sin embargo, se puede asociar con a lo sumo una entidad en A (véase la Figura 4b).
- Varios a uno. Una entidad en A se asocia con a lo sumo una entidad en B. Una entidad en B, sin embargo, se puede asociar con cualquier número de entidades (ninguna o varias) en A (véase la Figura 5a).
- Varios a Varios. Una entidad en A se asocia con cualquier número de entidades (ninguna o varias) en B, y una entidad en B se asocia con cualquier número de entidades (ninguna o varias) en A (véase la Figura 5b).

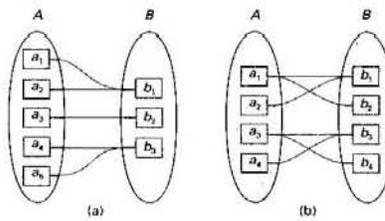


Figura 5. Correspondencia de cardinalidades(a) varios a uno. (b) varios a varios.

1.9.2.4 Diagrama Entidad-Relación.

La estructura lógica general de una base de datos se puede expresar gráficamente mediante un diagrama E-R. Los diagramas son simples y claros, cualidades que pueden ser responsables del amplio uso del modelo E-R. Tal diagrama consta de los siguientes componentes principales:

- Rectángulos, que presenta conjuntos de entidades.
- Elipses, que representa atributos.
- Rombo, que representan relaciones.
- Líneas, que unen atributos a conjuntos de entidades y conjuntos de entidades a conjuntos de relaciones.
- Elipses dobles, que representa atributos multivalorados.
- Elipses discontinuas, que denotan atributos derivados.
- Líneas dobles, que indican participación total de una entidad en un conjunto de relaciones.



Figura 6. Diagrama E-R con atributos compuestos, multivalorados y derivados.

1.9.3 MySQL

MySQL es la base de datos de código abierto más popular del mundo. Código abierto significa que todo el mundo puede acceder al código fuente, es decir, al código de programación de MySQL. Todo el mundo puede contribuir para incluir elementos, arreglar problemas, realizar mejoras o sugerir optimizaciones. MySQL ha pasado de ser una pequeña base de datos a una completa herramienta y ha conseguido superar a una gran cantidad de base de datos comerciales (lo que ha asustado a la mayor parte de los proveedores comerciales de base de datos). Por lo tanto, su rápido desarrollo se debe a la contribución de mucha gente al proyecto, así como a la dedicación del equipo de MySQL.

A diferencia de los proyectos propietarios, en los que el código fuente es desarrollado por un número reducido de personas y se protege atentamente, los proyectos de código abierto no excluyen a nadie interesado en aportar ideas, si disponen de los conocimientos necesarios. En el año 2000, cuando MySQL contaba con solo cuatro años de existencia, Michael Monty Widenius, el fundador de MySQL, predijo grandes avances para MySQL durante la primera convención sobre bases de datos de código abierto. En aquel entonces, muchos proveedores de base de datos de burlaron de sus palabras. Hoy en día ya han desaparecido varios.

La versión 3 de MySQL logró hacerse con el dominio de la gama baja del mercado de Internet. Con el lanzamiento de la versión 4, este producto se dirige ahora a una base de clientes mucho más amplia. MySQL hace su entrada en el mercado de las bases de datos en un momento en el que Apache es el producto de código abierto dominante en el mercado de servidores Web y en el que la presencia de varios sistemas operativos de código abierto (como Linux y FreeBSD) es cada día más notable en el mercado de servidores.

MySQL es un sistema de administración de base de datos relacional (RDBMS). Se trata de un programa capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización, desde pequeños establecimientos comerciales o grandes empresas y organismos administrativos. MySQL compite con sistemas RDBMS propietarios conocidos, como Oracle, SQL Server y DB2.

MySQL incluye todos los elementos necesarios para instalar el programa, preparar diferentes niveles de acceso a usuario, administrar el sistema, proteger y hacer volcados de datos. Puede desarrollar sus propias aplicaciones de base de datos en la mayor parte de los lenguajes de programación utilizando en la actualidad y ejecutarlo en casi todos los sistemas operativos, incluyendo algunos de los que probablemente no ha oído nunca hablar. MySQL utiliza el lenguaje de consulta estructurado (SQL). Se trata del lenguaje utilizado por todas las bases de

relaciones. Este lenguaje permite crear base de datos, así como agregar, manipular y recuperar datos en función de criterios específicos.

JUSTIFICACIÓN

En la actualidad las empresas requieren de nuevas alternativas para la realización de sus tareas y la mejora de sus procesos, para poder así alcanzar sus objetivos y ser más competitivas, por tal motivo la realización del sistema informático para el control de inventario de una empresa distribuidora de granos ayudara en la mejora de los procesos.

Con el uso de éste sistema se reducirá el tiempo de ejecución de los procesos, se evitara la pérdida de información, se ahorra tiempo respecto a la búsqueda de proveedores y clientes, así como saber cuanto se tiene de producto en el almacén, y llevar un control de las ventas.

UNIDAD II METODOLOGÍA DE APLICACIÓN

2.1 Definición del problema y objetivos.

La empresa no cuenta con un sistema que administre las entradas y salidas del inventario, todo este control se lleva bajo papel y todo se escribe a mano lo cual se comete muchos errores de escritura y no es muy entendible la letra por lo cual esta información no es muy confiable y con lleva a cometer muchos errores, pérdidas económicas y de tiempo para el negocio así como al propietario.

2.1.1 Objetivo General.

Crear un sistema que controle las entradas y salidas del inventario, así como el control de proveedores, clientes y ventas, reduciendo tiempos en la búsqueda de información.

2.1.2 Objetivo Específicos.

- Automatizar los procesos de registro de inventario, clientes y proveedores.
- Controlar las entradas y salidas del inventario.
- Tener actualizado la cartera de clientes y proveedores.
- Reducir tiempos en búsqueda de información.
- Llevar el control de las ventas.

2.2 Determinación de los requerimientos de información.

Cuestionario de aplicación al dueño de la empresa:

1.- ¿Qué entiende usted como sistema de información?

R= Una automatización de la información.

2.- ¿Con cuantas áreas cuenta su empresa?

R= Almacén, compras, ventas.

3.- ¿Con que sistemas operativos cuenta actualmente?

R= Windows xp

4.- ¿Por qué cree que es necesario adquirir o implementar un sistema de información?

R= Para el control de mis inventarios.

5.- ¿Cuánto estaría dispuesto a pagar por un sistema de información?

R= \$15,000.00

6.- ¿Cuáles son sus actividades principales dentro de la empresa?

R= Compra de la materia prima, supervisar la entrega de la compra, pagos a proveedores, venta a clientes, supervisar la entrega de la venta a los clientes.

7.- ¿Cómo se llevan a cabo las entradas del Inventario?

R= Se registra a mano en un cuaderno la compra de la materia prima.

8.- ¿Cómo se llevan a cabo las salidas del Inventario?

R= Se registra a mano en un cuaderno la venta y se le resta al inventario.

9.- ¿Qué áreas necesitan control específico?

R= Almacén y Ventas

2.3 Análisis de las necesidades del sistema.

El proceso que se sigue en la compra de producto, almacenado y venta del actual es el siguiente:

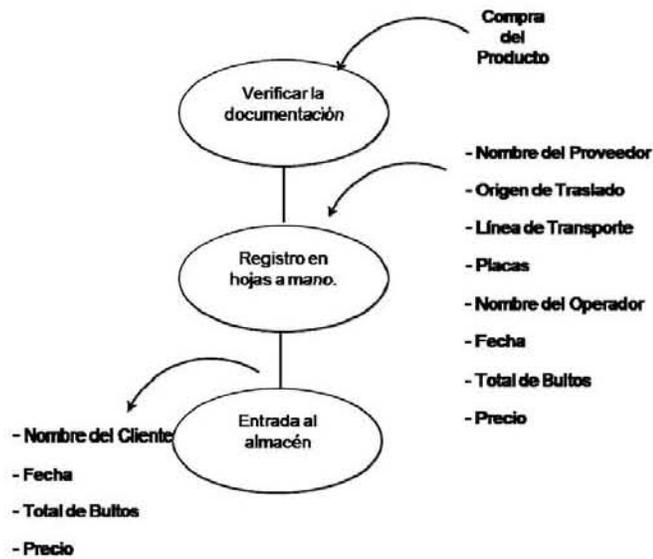


Figura 7. Diagrama de Proceso Actual.

Como podemos observar el proceso actual que se lleva para la entrada y salida del inventario es a mano y puede haber muchísimos errores lo adecuado sería hacer un sistema que automatice todos estos procesos, la propuesta que hago es diseñar una base de datos en MYSQL y una interfaz gráfica en Visual Basic.Net

Estudio De Factibilidad

En éste estudio se determina lo necesario para poder llevar a cabo el sistema informático para el control de inventario en una empresa distribuidora de granos, como son los aspectos técnicos y humanos.

Requerimientos de Equipo:

Hardware:

Computadora

- Procesador Intel Pentium IV a 2.0 GHz
- Memoria RAM de 512 Mb
- Disco Duro de 80 Gb
- CD RW
- Mouse
- Teclado
- Monitor de 15"

Software:

Windows Vista o XP

MYSQL

Microsoft Visual Basic.Net

Requerimientos humanos:

Un solo elemento para realiza el sistema.

Costos Aproximados

Software.....	\$15,000.00
Papelería.....	\$100.00
Total.....	\$15,100.00

2.4 Diseño del sistema.

Con la información recabada se plantea una solución al diseño de la base de datos la cual tendrá por nombre SICI (Sistema informático para el control de inventario) y contendrá 9 tablas las cuales son: acceso, almacén, clientes, proveedores, compras, producto, producto_compras, ventas, producto_ventas, que a continuación se muestra con sus respectivos campos y atributos.

2.4.1 Descripción de Tablas.

Acceso		
Campo	Tipo	Descripción
Usuario	Varchar(30)	Contiene el nombre y apellidos del usuario
NickName	Varchar(30)	Contiene el nickName del usuario
Contraseña	Varchar(30)	Contiene la contraseña del usuario
Permiso	int(2)	Contiene el permiso del usuario

Tabla 1. Acceso

Almacen		
Campo	Tipo	Descripción
Cve_producto	Varchar(4)	Contiene la clave del producto
Bultos	int(6)	Contiene el total de bultos

Tabla 2. Almacén

Clientes		
Campo	Tipo	Descripción
Cve_cliente	int(7)	Contiene la clave del cliente
RFC	Varchar(13)	Contiene el RFC del cliente
Razon_social	Varchar(60)	Contiene la razón social del cliente
Apaterno	Varchar(15)	Contiene el nombre paterno del cliente
Amaterno	Varchar(15)	Contiene el nombre materno del cliente
Nombre	Varchar(20)	Contiene el nombre del cliente
Calle	Varchar(20)	Contiene la calle donde vive el cliente
Numero	Varchar(6)	Contiene el numero donde vive el cliente
Colonia	Varchar(20)	Contiene la colonia donde vive el cliente
CP	int(5)	Contiene el código postal donde vive el cliente
Delegacion	Varchar(15)	Contiene la delegación donde vive el cliente
Estado	Varchar(15)	Contiene el estado donde vive el cliente
Telefono	Varchar(12)	Contiene el teléfono del cliente
Celular	Varchar(12)	Contiene el celular del cliente
Fax	Varchar(12)	Contiene el fax del cliente

Tabla 3. Clientes

Proveedores		
Campo	Tipo	Descripción
Cve_proveedor	int(7)	Contiene la clave del proveedor
RFC	Varchar(13)	Contiene el RFC del proveedor
Razon_social	Varchar(60)	Contiene la razón social del proveedor
Apaterno	Varchar(15)	Contiene el nombre paterno del proveedor
Amaterno	Varchar(15)	Contiene el nombre materno del proveedor
Nombre	Varchar(20)	Contiene el nombre del proveedor
Calle	Varchar(20)	Contiene la calle donde vive el proveedor
Numero	Varchar(6)	Contiene el numero donde vive el proveedor
Colonia	Varchar(20)	Contiene la colonia donde vive el proveedor
CP	int(5)	Contiene el código postal donde vive el proveedor
Delegacion	Varchar(15)	Contiene la delegación donde vive el proveedor
Estado	Varchar(15)	Contiene el estado donde vive el proveedor
Telefono	Varchar(12)	Contiene el teléfono del proveedor
Celular	Varchar(12)	Contiene el celular del proveedor
Fax	Varchar(12)	Contiene el fax del proveedor

Tabla 4. Proveedores

Compras		
Campo	Tipo	Descripción
Cve_compra	int(7)	Contiene la clave de compra
Cve_proveedor	int(7)	Contiene la clave del proveedor
Origen_del_producto	Varchar(25)	Contiene la procedencia del producto
LineaT	Varchar(15)	Contiene la línea de transporte
Placas	Varchar(8)	Contiene las placas del transporte
Operador	Varchar(30)	Contiene el nombre del operador
Fecha	date	Contiene la fecha de compra
Total	decimal(10,0)	Contiene el total de la compra

Tabla 5. Compras

Producto		
Campo	Tipo	Descripción
Cve_producto	Varchar(4)	Contiene la clave del producto
Descripción	Varchar(25)	Contiene la descripción del producto
Precio	decimal(10,0)	Contiene el precio del producto

Tabla 6. Producto

Producto_compras		
Campo	Tipo	Descripción
Cve_producto	Varchar(4)	Contiene la clave del producto
Cve_compra	int(7)	Contiene la clave de compra
Bultos	int(6)	Contiene la cantidad de bultos
Precio	decimal(10,0)	Contiene el precio del bulto
Subtotal	decimal(10,0)	Contiene le precio subtotal de los bultos

Tabla 7. Producto-Compras

Ventas		
Campo	Tipo	Descripción
Cve_venta	int(7)	Contiene la clave de la venta
Cve_cliente	int(7)	Contiene la clave del cliente
Total	decimal(10,0)	Contiene la venta total
Fecha	date	Contiene la fecha de venta

Tabla 8. Ventas

Producto_ventas		
Campo	Tipo	Descripción
Cve_producto	int(7)	Contiene la clave del producto
Cve_venta	int(7)	Contiene la clave de venta
Bultos	int(6)	Contiene la cantidad de bultos
Subtotal	decimal(10,0)	Contiene le precio subtotal de los bultos

Tabla 9. Producto-Ventas

Diagrama Entidad-Relación.

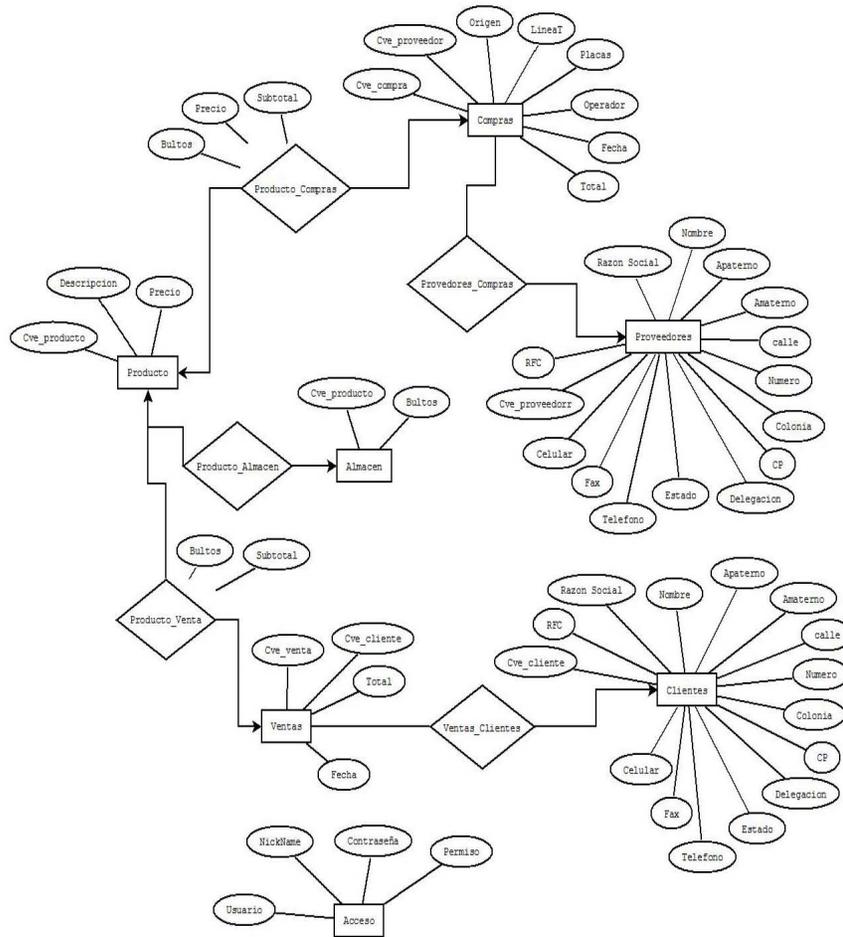


Figura 8. Diagrama Entidad-Relación

En el modelado de la base de datos se mostrara las relaciones que se tienen entre las tablas.

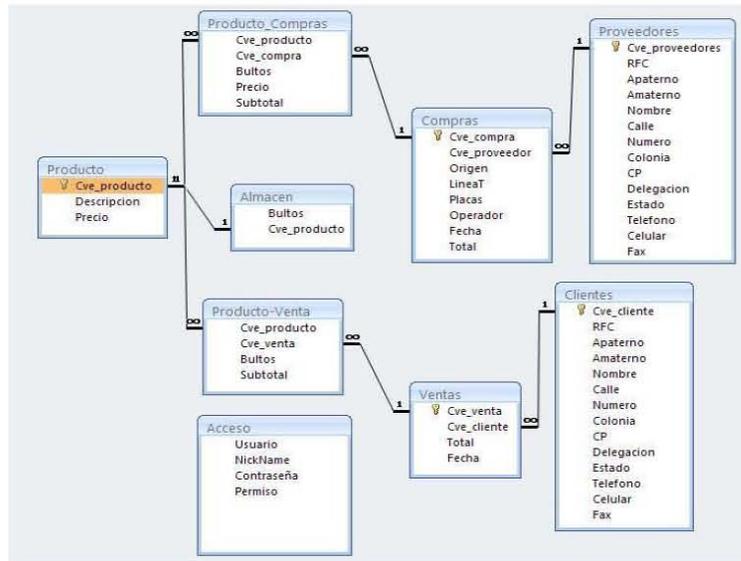


Figura 9. Modelado de la Base de Datos

2.5 Desarrollo y documentación del software.

En el desarrollo se mostrará las ventanas del sistema y su código correspondiente a cada una de ellas.

La ventana principal (Figura 10) que contiene ocho submenús pero solo dos están habilitados el de Sesiones y el de Salir, lo cual obliga a solo entrar a alguna de estas dos opciones. En el menú de Sesiones se despliega iniciar Sesión y Cerrar



Figura 11. Ventana de Inicio de Sesión

En la ventana de alta de clientes (Figura 12) que se encuentra localizada en el menú clientes, se van a capturar todos los datos personales de los clientes (código en anexo 3).

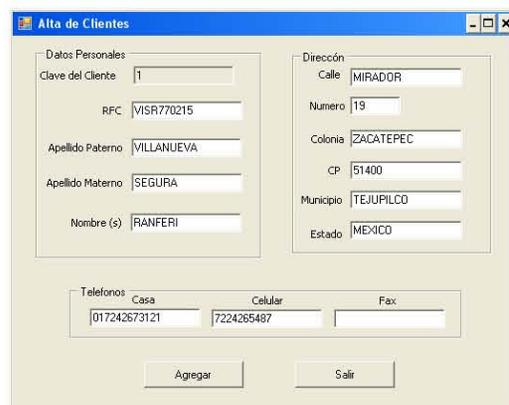


Figura 12. Ventana de Alta de Clientes

En la ventana de Baja de Clientes (Figura 13) que se encuentra localizada en el menú Clientes, se introduce la clave del cliente y si se encuentra la clave se da de baja (código en anexo 3).



Figura 13. Ventana de Baja de Clientes

En la ventana de Modificación de Clientes (Figura 14) que se encuentra localizada en el menú Clientes, se introduce la clave del cliente y si se encuentra la clave se despliegan los datos del cliente, se hacen los cambios y después se guardan los cambios dando clic en el botón modificar (código en anexo 3).

Introduce la Clave del Cliente: 4

Datos Personales

RFC: JACI800814

Apellido Paterno: JAIMES

Apellido Materno: CARBAJAL

Nombre (s): ISAIAS

Dirección

Calle: EL RINCON

Numero: 42

Colonia: ZACATEPEC

CP: 51400

Municipio: TEJUPLICO

Estado: ESTADO DE MEXIC

Telefonos

Casa: 017242675214

Celular: 7224125468

Fax:

Clave de Cliente	RFC	Apellido Paterno	Apellido Materno	Nombre	Calle	Numero	Colonia	Codigo Postal
1	VISR780212	VILLANUEVA	SEGURA	RANFERI	MIRADOR	48	ZACATEPEC	51400
2	VIPJ690911	VILLA	PEREZ	JOSE	ALLENDE	13	MEXICO 68	51400
3	SAAH701124	SANTIN	AVILES	HUGO	PRESA	36	EL LLAND	51400
4	JACI800814	JAIMES	CARBAJAL	ISAIAS	EL RINCON	42	ZACATEPEC	51400
5	GOPR740619	GOMEZ	PALACIOS	ROLANDO	AZUCENA	24	EL SOL	51400
*								

Figura 14. Ventana de Modificación de Clientes

En la ventana de Consulta General de Clientes (Figura 15) que se encuentra localizada en el menú Clientes, se visualizan todos los clientes (código en anexo 3).

	Clave de Cliente	RFC	Apellido Paterno	Apellido Materno	Nombre	Calle	Numero	Colonia	Codigo Postal
▶	1	VISR780212	VILLANUEVA	SEGURA	RANFERI	MIRADOR	48	ZACATEPEC	51400
	2	VIPJ690911	VILLA	PEREZ	JOSE	ALLENDE	13	MEXICO 68	51400
	3	SAAH701124	SANTIN	AVILES	HUGO	PRESA	36	EL LLAND	51400
	4	JACI800814	JAIME3	CARBAJAL	ISAIAS	EL RINCON	42	ZACATEPEC	51400
	5	GOPR740619	GOMEZ	PALACIOS	ROLANDO	AZUCENA	24	EL SOL	51400
*									

Figura 15. Ventana de Consultas General de Clientes

En la ventana de Alta de Proveedores (Figura 16) que se encuentra localizada en el menú proveedores, se va a capturar todos los datos personales de los proveedores (código en anexo 3).

Alta de Proveedores

Datos Personales

Clave del Proveedor 2

RFC

Apellido Paterno

Apellido Materno

Nombre (s)

Dirección

Calle

Numero

Colonia

CP

Municipio

Estado

Telefonos

Casa Celular Fax

Agregar Salir

Figura 16. Ventana de Alta de Proveedores

En la ventana de Baja de Proveedores (Figura 17) que se encuentra localizada en el menú Proveedores, se introduce la clave del Proveedor y si se encuentra la clave se da de baja(código en anexo 3).

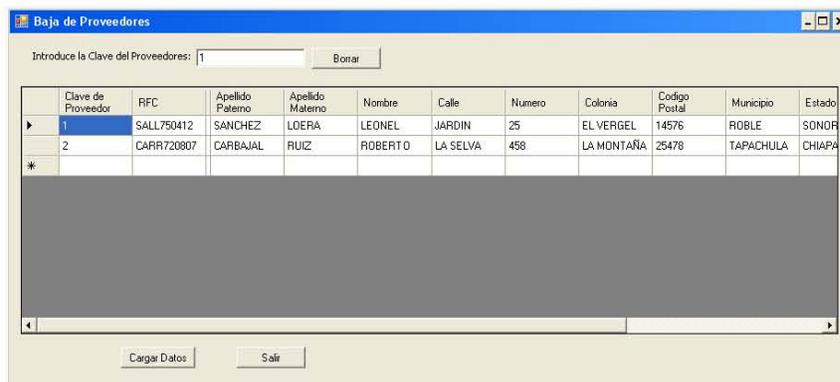


Figura 17. Ventana de Baja de Proveedores

En la ventana de Modificación de Proveedores (Figura 18) que se encuentra localizada en el menú Proveedores, se introduce la clave del proveedor y si se encuentra la clave se despliegan los datos del proveedor, se hacen los cambios y después se guardan los cambios dando clic en el botón modificar (código en anexo 3).

Introduce la Clave del Proveedor:

Datos Personales

RFC:

Apellido Paterno:

Apellido Materno:

Nombre (s):

Dirección

Calle:

Numero:

Colonia:

CP:

Municipio:

Estado:

Telefonos

Casa:

Celular:

Fax:

Clave de Proveedor	RFC	Apellido Paterno	Apellido Materno	Nombre	Calle	Numero	Colonia	Código Postal
▶ 1	SALL750412	SANCHEZ	LOERA	LEONEL	JARDIN	25	EL VERGEL	14576
2	CARR720807	CARBAJAL	RUIZ	ROBERTO	LA SELVA	458	LA MONTAÑA	25478
*								

Figura 18. Ventana de Modificación de Proveedores

En la ventana de Consulta General de Proveedores (Figura 19) que se encuentra localizada en el menú Proveedores, se visualizan todos los proveedores (código en anexo 3).

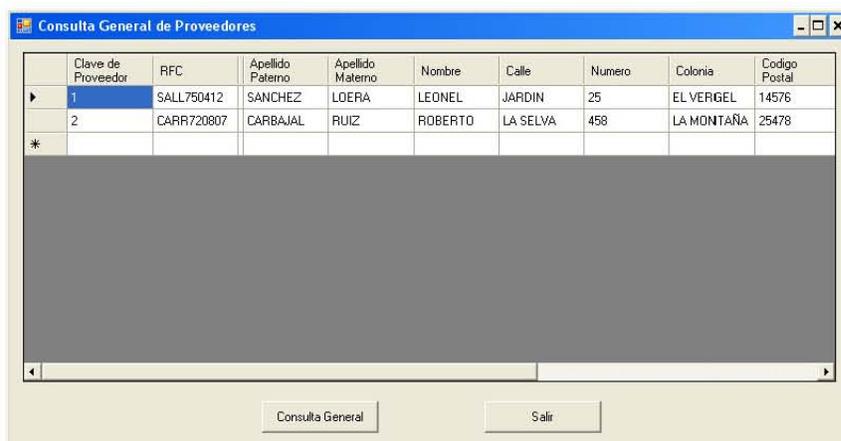


Figura 19. Ventana de Consulta General de Proveedores

En la ventana de Alta de Producto (Figura 20) que se encuentra localizada en el menú inventario, se va a capturar los datos del producto (código en anexo 3).



Figura 20. Ventana de Alta de Productos

En la ventana de Baja de Producto (Figura 21) que se encuentra localizada en el menú inventario, se introduce la clave del Producto y si se encuentra la clave se da de baja (código en anexo 3).



Figura 21. Ventana de Baja de Productos

En la ventana de Modificación de Productos (Figura 22) que se encuentra localizada en el menú inventario, se introduce la clave del Producto y si se encuentra la clave se despliegan los datos del Producto, se hacen los cambios y después se guardan los cambios dando clic en el botón modificar (código en anexo 3).

Introduce la Clave del Producto:

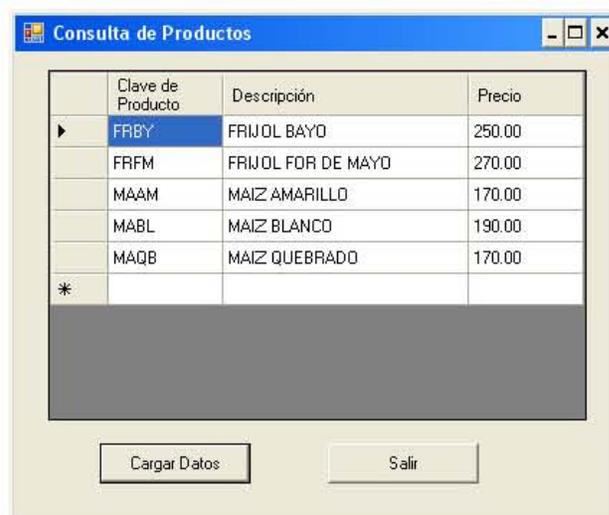
Descripción

Precio

Clave de Producto	Descripción	Precio
FRBY	FRIJOL BAYO	250.00
FRFM	FRIJOL FOR DE MAYO	270.00
MAAM	MAIZ AMARILLO	170.00

Figura 22. Ventana de Modificación de Productos

En la ventana de Consulta de Producto (Figura 23) la cual se encuentra localizada en el menú inventario, se visualizan todos los productos (código en anexo 3).



	Clave de Producto	Descripción	Precio
▶	FRBY	FRIJOL BAYO	250.00
	FRFM	FRIJOL FOR DE MAYO	270.00
	MAAM	MAIZ AMARILLO	170.00
	MABL	MAIZ BLANCO	190.00
	MAQB	MAIZ QUEBRADO	170.00
*			

Cargar Datos Salir

Figura 23. Ventana de Consulta de Productos

En la ventana de Compras de Productos (Figura 24) que se encuentra localizada en el menú inventario, se guardan las compras de productos que se realiza a los proveedores (código en anexo 3).

The screenshot shows a software window titled "Compras de Producto". It is divided into two main sections: "Datos del Proveedor" and "Productos".

Datos del Proveedor:

- Clave del Compra: 2
- Clave del Proveedor: 1
- Nombre de Proveedor: SANCHEZ LOERA LEONEL
- Origen de Traslado: SONORA
- Línea de Transporte: PARTICULAR
- Placas: LS782514
- Nombre del Operador: HERNANDEZ GONZALEZ PEDRO
- Fecha: 07/01/2010

Productos:

- Clave del Producto: MABL
- Descripción: MAIZ BLANCO
- Bultos: 600
- Precio \$: 140
- Total \$: 84000

At the bottom of the window, there are two buttons: "Guardar Comprar" and "Salir".

Figura 24. Ventana de Compras de Productos

En la ventana de Ventas de Productos (Figura 25) que se encuentra localizada en el menú ventas, se guardan las ventas realizadas a los clientes (código en anexo 3).

The screenshot shows a software window titled "Ventas de Producto". It is divided into two main sections: "Datos del Cliente" and "Productos".

Datos del Cliente:

- Clave de Venta: 1
- Clave del Cliente: 4
- Nombre del Cliente: JAIMES CARBAJAL ISAIAS
- Fecha: 2/02/2010

Productos:

- Clave del Producto: MAAM
- Descripción: MAIZ AMARILLO
- Bultos: 20
- Precio \$: 170.00
- Total \$: 5000

At the bottom of the window, there are two buttons: "Guardar Venta" and "Salir".

Figura 25. Ventana de Ventas de Productos

2.6 Pruebas y mantenimiento del sistema.

Antes de que fuera el sistema implementado se le hicieron las pruebas al sistema informático para el control de inventario en una empresa distribuidora de granos, esto en primera instancia en el código para asegurar los procesos y vinculación con las tablas de la base de datos y verificando que la inserción de los datos y transformación de ésta, arrojará la salida de información correcta, por otra parte se revisó el diseño de la ventanas que fueran entendibles para los usuarios finales y de fácil manejo, el mantenimiento del sistema aun no lo requiere ya que está

funcionando perfectamente y si el usuario requiriera aumentar módulos entonces se le haría el mantenimiento correspondiente.

2.7 Implementación y evaluación del sistema.

Para la implementación del sistema respecto al software se requirió del programa de instalación de MySQL, también se necesitó un conector de MySQL. Respecto al Hardware se instaló el sistema en un equipo con las siguientes características: Procesador de 1.7 GHZ, Disco Duro de 80 GB, Memoria RAM de 1GB.

Ya que se tiene el Software y Hardware se instala el sistema SICI (sistema Informático Para el Control de Inventario).

Este sistema se encuentra en un CD después seguir los pasos correspondientes para la instalación, también se tiene la base de datos SICI.sql la cual de debe de cargar al sistema, desde MSDOS se cargar en la siguiente ruta:

```
c:\wamp\mysql\bin\
```

Después de cargar la ruta, teclear la siguiente sintaxis (esto es para cargar la base de datos):

```
c:\wamp\mysql\bin\mysqldump u- pwd- sici<e:sici.sql
```

donde "e": es la ubicación del cd.

CONCLUSIONES

Las empresas productivas deben responder a los cambios en el entorno con rapidez para su crecimiento y es importante que utilicen las herramientas que ofrecen las tecnologías de información.

Por ello es que me puse en la tarea de desarrollar un sistema informático para el control de inventario de una empresa distribuidora de granos.

Con el uso de éste sistema se reduce el tiempo de ejecución de los procesos, se evita la pérdida de información, se ahorra tiempo respecto a la búsqueda de proveedores y clientes, así como saber cuánto se tiene de producto en el almacén.

Con éste sistema se logró tener la información actualizada, y que esto repercute en la toma de decisiones del dueño de la empresa con respecto al rumbo su negocio.

En este sistema puse todos mis conocimientos adquiridos en la universidad, lo más importante es que este sistema está funcionando actualmente en la empresa y es cuando realmente vale invertir tiempo, conocimientos y esfuerzo para realización de un proyecto, esto me satisface y me da la pauta para seguir trabajando en otros proyectos que ayuden al manejo de la información.

Cabe mencionar que hubo una encuesta final (Anexo 1) de aprobación por parte del usuario propietario del sistema el cual avala que el funcionamiento es correcto de acuerdo a lo propuesto desde el inicio del proyecto.

Terminando esta etapa de conclusiones, desde un punto de vista propio, creo que no se aprende todo en la universidad como a veces se piensa, sino ya estando en el ambiente laboral la realidad cambia y uno tiene que tomar la iniciativa y actitud de aprender y querer hacer las cosas y así poder ser un cada día un mejor profesionalista.

BIBLIOGRAFÍA

LIBROS

Roger S. Pressman, Ingeniería del software, Mc Graw Hill, Sexta Edición año 2005, pág. 5, 8-10

Kenneth E. Kendall, Análisis y Diseño de Sistemas, PRENTICE HALL, Tercera Edición año 1997, pág. 1-13

Roger S. Pressman, Ingeniería del software un enfoque practico, Mc Graw Hill, Cuarta Edición año 1998, pág. 156-158, 170-174

Michael Halvorson, Visual Basic.NET, Mc Graw Hill, Primera Edición año 2002, pág. 5-100

Abraham Silberschatz, Henry F. Korth, S. Sudarshan, Fundamentos de Base de datos, Mc Graw Hill, Cuarta Edición año 2002, pág. 1-100

Lan Gilfillan, La Biblia de MySQL, Anaya Multimedia, Primera Edición año 2001, pág. 39-41

ANEXO 1

Encuesta de evaluación final

1. ¿El sistema cumple con las expectativas esperadas?
Si No
2. ¿El sistema es entendible?
Si No
3. ¿El sistema es amigable?
Si No
4. ¿Le resulta útil y de fácil manejo el sistema?
Si No
5. ¿El diseño del sistema le pareció adecuado?
Si No
6. ¿Los resultados hasta el momento han sido satisfactorios?
Si No
7. ¿Tiene usted algún comentario adicional que hacer al mismo?
Si No
Cual _____

ANEXO 2

ÍNDICE DE FIGURAS

Figura 1. Elementos del sistema	8
Figura 2. Categoría de los sistemas	14
Figura 3. Ciclo de vida del desarrollo de sistemas	36
Figura 4. Correspondencia de cardinalidades (a) uno a uno. (b) uno a varios.	51
Figura 5. Correspondencia de cardinalidades(a) varios a uno. (b) varios a varios.	52
Figura 6. Diagrama E-R con atributos compuestos, multivalorados y derivados.	53
Figura 7. Diagrama de Proceso Actual.....	60
Figura 8. Diagrama Entidad-Relación.....	66
Figura 9. Modelado de la Base de Datos	67
Figura 10. Ventana Principal.....	68
Figura 11. Ventana de Inicio de Sesión.....	69
Figura 12. Ventana de Alta de Clientes.....	69
Figura 13. Ventana de Baja de Clientes.....	70
Figura 14. Ventana de Modificación de Clientes.....	71
Figura 15. Ventana de Consultas General de Clientes.....	72
Figura 16. Ventana de Alta de Proveedores.....	73
Figura 17. Ventana de Baja de Proveedores.....	74
Figura 18. Ventana de Modificación de Proveedores.....	75
Figura 19. Ventana de Consulta General de Proveedores.....	76
Figura 20. Ventana de Alta de Productos	76
Figura 21. Ventana de Baja de Productos	77
Figura 22. Ventana de Modificación de Productos	78
Figura 23. Ventana de Consulta de Productos.....	79
Figura 24. Ventana de Compras de Productos.....	80
Figura 25. Ventana de Ventas de Productos.....	81

ÍNDICE DE TABLAS

Tabla 1. Acceso.....	63
Tabla 2. Almacén.....	63
Tabla 3. Clientes	63
Tabla 4. Proveedores	64
Tabla 5. Compras.....	64
Tabla 6. Producto.....	65
Tabla 7. Producto-Compras	65
Tabla 8. Ventas	65
Tabla 9. Producto-Ventas.....	65

ANEXO 3

Código de ventana principal.

```
Public Class MenuP
    Private Sub AltasToolStripMenuItemem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
        AltasToolStripMenuItemem.Click
    End Sub

    Private Sub ASercaDeToolStripMenuItemem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
        ASercaDeToolStripMenuItemem.Click
    End Sub

    Private Sub PagoToolStripMenuItemem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
        PagoToolStripMenuItemem.Click
    End Sub

    Private Sub SalirToolStripMenuItemem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
        SalirToolStripMenuItemem.Click
    End
    End Sub

    Private Sub CerrarSesiónToolStripMenuItemem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
        Handles CerrarSesiónToolStripMenuItemem.Click
        ToolStripStatusLabel1.Text = "No hay sesión inicializada"
        estado()
    End Sub

    Private Sub IniciarSesiónToolStripMenuItemem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
        Handles IniciarSesiónToolStripMenuItemem.Click
        valor = 0
        Enabled = False
        Sesion.ShowDialog()
    End Sub

    Private Sub InventarioToolStripMenuItemem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
        InventarioToolStripMenuItemem.Click
    End Sub

    Private Sub Menu_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        estado()
    End Sub

    Private Sub VentasToolStripMenuItemem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
        VentasToolStripMenuItemem.Click
    End Sub

    Private Sub ClientesToolStripMenuItemem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
        ClientesToolStripMenuItemem.Click
    End Sub

    Private Sub ProveedoresToolStripMenuItemem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
        Handles ProveedoresToolStripMenuItemem.Click
    End Sub
End Class
```

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

```
End Sub
```

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

```
End Sub
```

```
Private Sub estado()
```

```
valor = 0  
InventarioToolStripMenuItem.Enabled = False  
VentasToolStripMenuItem.Enabled = False  
ClientesToolStripMenuItem.Enabled = False  
ProveedoresToolStripMenuItem.Enabled = False  
HerramientasToolStripMenuItem.Enabled = False  
AyudaToolStripMenuItem.Enabled = False
```

```
End Sub
```

```
Private Sub MenuP_EnabledChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.EnabledChanged
```

```
If valor = 1 Then  
InventarioToolStripMenuItem.Enabled = True  
VentasToolStripMenuItem.Enabled = True  
ClientesToolStripMenuItem.Enabled = True  
ProveedoresToolStripMenuItem.Enabled = True  
HerramientasToolStripMenuItem.Enabled = True  
AyudaToolStripMenuItem.Enabled = True  
ToolStripStatusLabel1.Text = nomusu
```

```
Else
```

```
If valor = 2 Then  
InventarioToolStripMenuItem.Enabled = True  
VentasToolStripMenuItem.Enabled = True  
AltasToolStripMenuItem.Enabled = False  
BajasToolStripMenuItem.Enabled = False  
ModificacionesToolStripMenuItem.Enabled = False
```

```
ClientesToolStripMenuItem.Enabled = True  
AltasToolStripMenuItem1.Enabled = False  
BajasToolStripMenuItem1.Enabled = False  
ModificacionesToolStripMenuItem1.Enabled = False  
ProveedoresToolStripMenuItem.Enabled = True  
AltasToolStripMenuItem2.Enabled = False  
BajasToolStripMenuItem2.Enabled = False  
ModificacionesToolStripMenuItem2.Enabled = False  
AyudaToolStripMenuItem.Enabled = True  
ToolStripStatusLabel1.Text = nomusu
```

```
Else  
estado()  
End If
```

```
End If
```

```
End Sub
```

```
Private Sub StatusStrip1_ItemClicked(ByVal sender As System.Object, ByVal e As System.Windows.Forms.ToolStripItemClickedEventArgs) Handles StatusStrip1.ItemClicked
```

```
End Sub
```

```
Private Sub ToolStripStatusLabel1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripStatusLabel1.Click
```

```
End Sub

Private Sub AltaDeUsuariosToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
AltaDeUsuariosToolStripMenuItem.Click
    AUsuarios.Show()
End Sub

Private Sub AltasToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
AltasToolStripMenuItem1.Click
    AClientes.Show()
End Sub

Private Sub BajasToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
BajasToolStripMenuItem.Click
End Sub

Private Sub ModificacionesToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles ModificacionesToolStripMenuItem.Click
End Sub

Private Sub BajasToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
BajasToolStripMenuItem1.Click
    BClientes.Show()
End Sub

Private Sub ModificacionesToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles ModificacionesToolStripMenuItem1.Click
    MClientes.Show()
End Sub

Private Sub AltasToolStripMenuItem2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
AltasToolStripMenuItem2.Click
    AProveedores.Show()
End Sub

Private Sub SesionesToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
SesionesToolStripMenuItem.Click
End Sub

Private Sub ConsultasToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles ConsultasToolStripMenuItem1.Click
    CClientes.Show()
End Sub

Private Sub BajasToolStripMenuItem2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
BajasToolStripMenuItem2.Click
    BProveedores.Show()
End Sub

Private Sub ModificacionesToolStripMenuItem2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles ModificacionesToolStripMenuItem2.Click
    MProveedores.Show()
End Sub
```

```

Private Sub ConsultasToolStripMenuItem2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles ConsultasToolStripMenuItem2.Click
    CProveedores.Show()
End Sub

Private Sub BajaDeUsuariosToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles BajaDeUsuariosToolStripMenuItem.Click
    BUsuarios.Show()
End Sub

Private Sub ModificacionesDeUsuariosToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ModificacionesDeUsuariosToolStripMenuItem.Click
    MUsuarios.Show()
End Sub

Private Sub AltasToolStripMenuItem3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
AltasToolStripMenuItem3.Click
    AProductos.Show()
End Sub

Private Sub BajasToolStripMenuItem3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
BajasToolStripMenuItem3.Click
    BProductos.Show()
End Sub

Private Sub ModificacionesToolStripMenuItem3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles ModificacionesToolStripMenuItem3.Click
    MProductos.Show()
End Sub

Private Sub ConsultasToolStripMenuItem3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles ConsultasToolStripMenuItem3.Click
    CProductos.Show()
End Sub

Private Sub ConsultasToolStripMenuItem4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles ConsultasToolStripMenuItem4.Click
    CAImacen.Show()
End Sub

Private Sub AltasToolStripMenuItem4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
AltasToolStripMenuItem4.Click
    ACompras.Show()
End Sub
End Class

```

Código de ventana de inicio de Sesión.

```

Public Class Sesion
    Dim bo As Boolean
    Dim cont As Integer
    Dim pun As Integer
    Dim dig As Integer

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Close()
        MenuP.Enabled = True
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

```

```

Dim existe As Boolean
Dim us As String
us = ""
Dim perm As String
perm = ""
existe = False
Dim re As Integer
Dim cad As String = "Server=localhost;Database=sici;Uid=root;Pwd= ;Port=3306"
Dim comando As New MySqlCommand
Dim cnn As New MySqlConnection(cad)
cnn.Open()
comando.Connection = cnn
comando.CommandText = "select *from acceso where NickName= " + TextBox1.Text + " and Contraseña=" +
TextBox2.Text + ""
Dim ad As New MySqlDataAdapter(comando.CommandText, cnn)
Dim tabla As New DataSet()
ad.Fill(tabla, "acceso")

While re < tabla.Tables("acceso").Rows.Count
    existe = True
    us = tabla.Tables(re).Rows(re)(0).ToString()
    perm = tabla.Tables(re).Rows(re)(3).ToString()
    re = re + 1
End While
cnn.Close()
nomusu = us
If existe = False Then
    MsgBox("El Usuario o la Contraseña son erroneos...")
    TextBox1.BackColor = (Color.White)
    TextBox1.Clear()
    TextBox2.BackColor = (Color.White)
    TextBox2.Clear()
Else
    MsgBox("Bienvenido al Sistema: " + us)
    If perm = "1" Then
        valor = 1
        MenuP.Enabled = True
        Hide()
    Else
        valor = 2
        MenuP.Enabled = True
        Hide()
    End If
End If

End If

End Sub

Private Sub Sesion_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    TextBox1.Text = ""
    TextBox2.Text = ""
    bo = False
    cont = 0
    pun = 0
    dig = 5
End Sub

End Class

```

Código de ventana de Alta de Clientes.

```

Imports MySql.Data
Imports MySql.Data.MySqlClient
Public Class AClientes

```

```

Private Sub AClientes_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    Dim cad As String = "Server=localhost;Database=sici;Uid=root;Pwd= ;Port=3306"
    Dim comando As New MySqlCommand
    Dim cnn As New MySqlConnection(cad)
    Dim a As Integer
    Dim re As Integer

    cnn.Open()
    comando.Connection = cnn
    Dim ad As New MySqlDataAdapter("Select *from clientes1", cnn)
    Dim tabla As New DataSet()
    ad.Fill(tabla, "clientes1")

    While re < tabla.Tables("clientes1").Rows.Count

        a = tabla.Tables(re).Rows(re)(0).ToString()
        re = re + 1
    End While
    TextBox1.Text = a + 1
    cnn.Close()

    TextBox2.Text = ""
    TextBox3.Text = ""
    TextBox4.Text = ""
    TextBox5.Text = ""
    TextBox6.Text = ""
    TextBox7.Text = ""
    TextBox8.Text = ""
    TextBox9.Text = ""
    TextBox10.Text = ""
    TextBox11.Text = ""
    TextBox12.Text = ""
    TextBox13.Text = ""
    TextBox14.Text = ""

End Sub

Private Sub Label3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim cad As String = "Server=localhost;Database=sici;Uid=root;Pwd= ;Port=3306"
    Dim comando As New MySqlCommand
    Dim cnn As New MySqlConnection(cad)

    cnn.Open()
    comando.Connection = cnn
    comando.CommandText = "insert into clientes1 values(" + TextBox1.Text + "," + TextBox2.Text + "," + TextBox3.Text
+ "," + TextBox4.Text + "," + TextBox5.Text + "," + TextBox6.Text + "," + TextBox7.Text + "," + TextBox8.Text + "," +
    TextBox9.Text + "," + TextBox10.Text + "," + TextBox11.Text + "," + TextBox12.Text + "," + TextBox13.Text + "," +
    TextBox14.Text + ");"
    comando.ExecuteNonQuery()
    MsgBox("Se ha agregado el Cliente")
    cnn.Close()
    MenuP.Show()
    TextBox1.Text = ""
    TextBox2.Text = ""
    TextBox3.Text = ""
    TextBox4.Text = ""
    TextBox5.Text = ""
    TextBox6.Text = ""
    TextBox7.Text = ""
    TextBox8.Text = ""
    TextBox9.Text = ""
    TextBox10.Text = ""

```

```
TextBox11.Text = ""
TextBox12.Text = ""
TextBox13.Text = ""
TextBox14.Text = ""

End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
Close()
End Sub
End Class
```

Código de ventana de Bajas de Clientes.

```
Imports MySql.Data
Imports MySql.Data.MySqlClient
Public Class BClientes

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
Dim cad As String = "Server=localhost;Database=sic1;Uid=root;Pwd=;Port=3306"
Dim comando As New MySqlCommand
Dim cnn As New MySqlConnection(cad)
Dim tabla As New DataTable

cnn.Open()
comando.Connection = cnn
Dim ad As New MySqlCommandAdapter("Select *from clientes1", cnn)
Dim Cmd As New MySqlCommandBuilder(ad)
ad.Fill(tabla)
DataGridView1.DataSource = tabla
DataGridView1.Columns(0).Width = 80
DataGridView1.Columns(0).HeaderText = "Clave de Cliente"
DataGridView1.Columns(1).Width = 80
DataGridView1.Columns(1).HeaderText = "RFC"
DataGridView1.Columns(2).Width = 80
DataGridView1.Columns(2).HeaderText = "Apellido Paterno"
DataGridView1.Columns(3).Width = 80
DataGridView1.Columns(3).HeaderText = "Apellido Materno"
DataGridView1.Columns(4).Width = 80
DataGridView1.Columns(4).HeaderText = "Nombre"
DataGridView1.Columns(5).Width = 80
DataGridView1.Columns(5).HeaderText = "Calle"
DataGridView1.Columns(6).Width = 80
DataGridView1.Columns(6).HeaderText = "Numero"
DataGridView1.Columns(7).Width = 80
DataGridView1.Columns(7).HeaderText = "Colonia"
DataGridView1.Columns(8).HeaderText = "Codigo Postal"
DataGridView1.Columns(9).Width = 80
DataGridView1.Columns(9).HeaderText = "Municipio"
DataGridView1.Columns(10).Width = 80
DataGridView1.Columns(10).HeaderText = "Estado"
DataGridView1.Columns(11).Width = 80
DataGridView1.Columns(11).HeaderText = "Telefono"
DataGridView1.Columns(12).Width = 80
DataGridView1.Columns(12).HeaderText = "Celular"
DataGridView1.Columns(13).Width = 80
DataGridView1.Columns(13).HeaderText = "Fax"
cnn.Close()
End Sub
```

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
    Close()
End Sub
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim cad As String = "Server=localhost;Database=sici;Uid=root;Pwd= ;Port=3306"
    Dim comando As New MySqlCommand
    Dim cnn As New MySqlConnection(cad)
    cnn.Open()
    comando.Connection = cnn
    comando.CommandText = "Delete from clientes1 where Cve_cliente= " + TextBox1.Text + ""
    comando.ExecuteNonQuery()
    MsgBox("Se ha eliminado el Cliente")
    cnn.Close()
End Sub
```

```
Private Sub BCientes_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

    End Sub
End Class
```

Código de ventana de Modificación de Clientes

```
Imports MySql.Data
Imports MySql.Data.MySqlClient
Public Class MClientes
```

```
    Private Sub MClientes_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        ocultar()
    End Sub
```

```
    Private Sub limpiar()
        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox3.Text = ""
        TextBox4.Text = ""
        TextBox5.Text = ""
        TextBox6.Text = ""
        TextBox7.Text = ""
        TextBox8.Text = ""
        TextBox9.Text = ""
        TextBox10.Text = ""
        TextBox11.Text = ""
        TextBox12.Text = ""
        TextBox13.Text = ""
        TextBox14.Text = ""
    End Sub
```

```
    Private Sub mostrar()
        TextBox2.Visible = True
        TextBox3.Visible = True
        TextBox4.Visible = True
        TextBox5.Visible = True
        TextBox6.Visible = True
        TextBox7.Visible = True
        TextBox8.Visible = True
        TextBox9.Visible = True
        TextBox10.Visible = True
        TextBox11.Visible = True
        TextBox12.Visible = True
        TextBox13.Visible = True
        TextBox14.Visible = True
    End Sub
```

```

Label2.Visible = True
Label3.Visible = True
Label4.Visible = True
Label5.Visible = True
Label6.Visible = True
Label7.Visible = True
Label8.Visible = True
Label9.Visible = True
Label10.Visible = True
Label11.Visible = True
Label12.Visible = True
Label13.Visible = True
Label14.Visible = True
Button4.Visible = True
End Sub

Private Sub ocultar()
    TextBox2.Visible = False
    TextBox3.Visible = False
    TextBox4.Visible = False
    TextBox5.Visible = False
    TextBox6.Visible = False
    TextBox7.Visible = False
    TextBox8.Visible = False
    TextBox9.Visible = False
    TextBox10.Visible = False
    TextBox11.Visible = False
    TextBox12.Visible = False
    TextBox13.Visible = False
    TextBox14.Visible = False
    Label2.Visible = False
    Label3.Visible = False
    Label4.Visible = False
    Label5.Visible = False
    Label6.Visible = False
    Label7.Visible = False
    Label8.Visible = False
    Label9.Visible = False
    Label10.Visible = False
    Label11.Visible = False
    Label12.Visible = False
    Label13.Visible = False
    Label14.Visible = False
    Button4.Visible = False
End Sub

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click
    Dim cad As String = "Server=localhost;Database=sici;Uid=root;Pwd= ;Port=3306"
    Dim comando As New MySqlCommand
    Dim cnn As New MySqlConnection(cad)
    cnn.Open()
    comando.Connection = cnn
    comando.CommandText = "update clientes1 set RFC="" + TextBox2.Text + "", Apaterno="" + TextBox3.Text + "",
Amaterno="" + TextBox4.Text + "", Nombre="" + TextBox5.Text + "", Calle="" + TextBox6.Text + "", Numero="" +
    TextBox7.Text + "", Colonia="" + TextBox8.Text + "", CP="" + TextBox9.Text + "", Delegacion="" + TextBox10.Text + "",
    Estado="" + TextBox11.Text + "", Telefono="" + TextBox12.Text + "", Celular="" + TextBox13.Text + "", Fax="" +
    TextBox14.Text + "" where Cve_cliente="" + TextBox1.Text + """
    comando.ExecuteNonQuery()
    MsgBox("Se a modificado el Cliente")
    cnn.Close()
    ocultar()
    limpiar()
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    Dim cad As String = "Server=localhost;Database=sici;Uid=root;Pwd= ;Port=3306"
    Dim comando As New MySqlCommand
    Dim cnn As New MySqlConnection(cad)

```

```

Dim tabla As New DataTable

cnn.Open()
comando.Connection = cnn
Dim ad As New MySqlDataAdapter("Select *from clientes1", cnn)
Dim Cmd As New MySqlCommandBuilder(ad)
ad.Fill(tabla)
DataGridView1.DataSource = tabla
DataGridView1.Columns(0).Width = 80
DataGridView1.Columns(0).HeaderText = "Clave de Cliente"
DataGridView1.Columns(1).Width = 80
DataGridView1.Columns(1).HeaderText = "RFC"
DataGridView1.Columns(2).Width = 80
DataGridView1.Columns(2).HeaderText = "Apellido Paterno"
DataGridView1.Columns(3).Width = 80
DataGridView1.Columns(3).HeaderText = "Apellido Materno"
DataGridView1.Columns(4).Width = 80
DataGridView1.Columns(4).HeaderText = "Nombre"
DataGridView1.Columns(5).Width = 80
DataGridView1.Columns(5).HeaderText = "Calle"
DataGridView1.Columns(6).Width = 80
DataGridView1.Columns(6).HeaderText = "Numero"
DataGridView1.Columns(7).Width = 80
DataGridView1.Columns(7).HeaderText = "Colonia"
DataGridView1.Columns(8).Width = 80
DataGridView1.Columns(8).HeaderText = "Codigo Postal"
DataGridView1.Columns(9).Width = 80
DataGridView1.Columns(9).HeaderText = "Municipio"
DataGridView1.Columns(10).Width = 80
DataGridView1.Columns(10).HeaderText = "Estado"
DataGridView1.Columns(11).Width = 80
DataGridView1.Columns(11).HeaderText = "Telefono"
DataGridView1.Columns(12).Width = 80
DataGridView1.Columns(12).HeaderText = "Celular"
DataGridView1.Columns(13).Width = 80
DataGridView1.Columns(13).HeaderText = "Fax"
cnn.Close()
End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
Dim existe As Boolean
existe = False
Dim re As Integer
Dim cad As String = "Server=localhost;Database=sici;Uid=root;Pwd= ;Port=3306"
Dim comando As New MySqlCommand
Dim cnn As New MySqlConnection(cad)

cnn.Open()
comando.Connection = cnn
comando.CommandText = "select *from clientes1 where Cve_cliente=" + TextBox1.Text + ""
Dim ad As New MySqlDataAdapter(comando.CommandText, cnn)
Dim tabla As New DataSet()
ad.Fill(tabla, "clientes1")

While re < tabla.Tables("clientes1").Rows.Count
    existe = True
    TextBox2.Text = tabla.Tables(re).Rows(re)(1).ToString()
    TextBox3.Text = tabla.Tables(re).Rows(re)(2).ToString()
    TextBox4.Text = tabla.Tables(re).Rows(re)(3).ToString()
    TextBox5.Text = tabla.Tables(re).Rows(re)(4).ToString()
    TextBox6.Text = tabla.Tables(re).Rows(re)(5).ToString()
    TextBox7.Text = tabla.Tables(re).Rows(re)(6).ToString()
    TextBox8.Text = tabla.Tables(re).Rows(re)(7).ToString()
    TextBox9.Text = tabla.Tables(re).Rows(re)(8).ToString()
    TextBox10.Text = tabla.Tables(re).Rows(re)(9).ToString()
    TextBox11.Text = tabla.Tables(re).Rows(re)(10).ToString()
    TextBox12.Text = tabla.Tables(re).Rows(re)(11).ToString()

```

```

        TextBox13.Text = tabla.Tables(re).Rows(re)(12).ToString()
        TextBox14.Text = tabla.Tables(re).Rows(re)(13).ToString()
        re = re + 1
    End While
    If existe = True Then
        mostrar()

    Else
        MsgBox("La Clave del Cliente No Existe")
    End If
    cnn.Close()
End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
    Close()
End Sub
End Class

```

Código de ventana de Consultas de Clientes

```

Imports MySql.Data
Imports MySql.Data.MySqlClient
Public Class CClientes

    Private Sub DataGridView1_CellContentClick(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.DataGridViewCellEventArgs) Handles DataGridView1.CellContentClick

        End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
        Close()
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Dim cad As String = "Server=localhost;Database=sici;Uid=root;Pwd= ;Port=3306"
        Dim comando As New MySqlCommand
        Dim cnn As New MySqlConnection(cad)
        Dim tabla As New DataTable

        cnn.Open()
        comando.Connection = cnn
        Dim ad As New MySqlDataAdapter("Select *from clientes1", cnn)
        Dim Cmd As New MySqlCommandBuilder(ad)
        ad.Fill(tabla)
        DataGridView1.DataSource = tabla
        DataGridView1.Columns(0).Width = 80
        DataGridView1.Columns(0).HeaderText = "Clave de Cliente"
        DataGridView1.Columns(1).Width = 80
        DataGridView1.Columns(1).HeaderText = "RFC"
        DataGridView1.Columns(2).Width = 80
        DataGridView1.Columns(2).HeaderText = "Apellido Paterno"
        DataGridView1.Columns(3).Width = 80
        DataGridView1.Columns(3).HeaderText = "Apellido Materno"
        DataGridView1.Columns(4).Width = 80
        DataGridView1.Columns(4).HeaderText = "Nombre"
        DataGridView1.Columns(5).Width = 80
        DataGridView1.Columns(5).HeaderText = "Calle"
        DataGridView1.Columns(6).Width = 80
        DataGridView1.Columns(6).HeaderText = "Numero"
        DataGridView1.Columns(7).Width = 80
        DataGridView1.Columns(7).HeaderText = "Colonia"
        DataGridView1.Columns(8).Width = 80
    End Sub
End Class

```

```

DataGridView1.Columns(8).HeaderText = "Codigo Postal"
DataGridView1.Columns(9).Width = 80
DataGridView1.Columns(9).HeaderText = "Municipio"
DataGridView1.Columns(10).Width = 80
DataGridView1.Columns(10).HeaderText = "Estado"
DataGridView1.Columns(11).Width = 80
DataGridView1.Columns(11).HeaderText = "Telefono"
DataGridView1.Columns(12).Width = 80
DataGridView1.Columns(12).HeaderText = "Celular"
DataGridView1.Columns(13).Width = 80
DataGridView1.Columns(13).HeaderText = "Fax"
cnn.Close()
End Sub

Private Sub CCientes_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

End Sub
End Class

```

Código de ventana de Altas de Proveedores

```

Imports MySql.Data
Imports MySql.Data.MySqlClient
Public Class AProveedores

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
Close()
End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
Dim cad As String = "Server=localhost;Database=sici;Uid=root;Pwd=.;Port=3306"
Dim comando As New MySqlCommand
Dim cnn As New MySqlConnection(cad)

cnn.Open()
comando.Connection = cnn
comando.CommandText = "insert into proveedores1 values(" + TextBox1.Text + "," + TextBox2.Text + "," +
TextBox3.Text + "," + TextBox4.Text + "," + TextBox5.Text + "," + TextBox6.Text + "," + TextBox7.Text + "," +
TextBox8.Text + "," + TextBox9.Text + "," + TextBox10.Text + "," + TextBox11.Text + "," + TextBox12.Text + "," +
TextBox13.Text + "," + TextBox14.Text + ");"
comando.ExecuteNonQuery()
MsgBox("Se ha agregado el Proveedor")
cnn.Close()
TextBox1.Text = ""
TextBox2.Text = ""
TextBox3.Text = ""
TextBox4.Text = ""
TextBox5.Text = ""
TextBox6.Text = ""
TextBox7.Text = ""
TextBox8.Text = ""
TextBox9.Text = ""
TextBox10.Text = ""
TextBox11.Text = ""
TextBox12.Text = ""
TextBox13.Text = ""
TextBox14.Text = ""
MenuP.Show()
End Sub

```

```

Private Sub Aproveedores_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    Dim cad As String = "Server=localhost;Database=sici;Uid=root;Pwd= ;Port=3306"
    Dim comando As New MySqlCommand
    Dim cnn As New MySqlConnection(cad)
    Dim a As Integer
    Dim re As Integer

    cnn.Open()
    comando.Connection = cnn
    Dim ad As New MySqlDataAdapter("Select *from proveedores1", cnn)
    Dim tabla As New DataSet()
    ad.Fill(tabla, "proveedores1")

    While re < tabla.Tables("proveedores1").Rows.Count

        a = tabla.Tables(re).Rows(re)(0).ToString()
        re = re + 1
    End While

    TextBox1.Text = a + 1
    cnn.Close()

    TextBox2.Text = ""
    TextBox3.Text = ""
    TextBox4.Text = ""
    TextBox5.Text = ""
    TextBox6.Text = ""
    TextBox7.Text = ""
    TextBox8.Text = ""
    TextBox9.Text = ""
    TextBox10.Text = ""
    TextBox11.Text = ""
    TextBox12.Text = ""
    TextBox13.Text = ""
    TextBox14.Text = ""

End Sub
End Class

```

Código de ventana de Bajas de Proveedores

```

Imports MySql.Data
Imports MySql.Data.MySqlClient
Public Class BProveedores

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
        Dim cad As String = "Server=localhost;Database=sici;Uid=root;Pwd= ;Port=3306"
        Dim comando As New MySqlCommand
        Dim cnn As New MySqlConnection(cad)
        Dim tabla As New DataTable

        cnn.Open()
        comando.Connection = cnn
        Dim ad As New MySqlDataAdapter("Select *from proveedores1", cnn)
        Dim Cmd As New MySqlCommandBuilder(ad)
        ad.Fill(tabla)
        DataGridView1.DataSource = tabla
        DataGridView1.Columns(0).Width = 80
        DataGridView1.Columns(0).HeaderText = "Clave de Proveedor"
        DataGridView1.Columns(1).Width = 80
        DataGridView1.Columns(1).HeaderText = "RFC"
        DataGridView1.Columns(2).Width = 80
        DataGridView1.Columns(2).HeaderText = "Apellido Paterno"
        DataGridView1.Columns(3).Width = 80
        DataGridView1.Columns(3).HeaderText = "Apellido Materno"
        DataGridView1.Columns(4).Width = 80
    End Sub
End Class

```

```

DataGridView1.Columns(4).HeaderText = "Nombre"
DataGridView1.Columns(5).Width = 80
DataGridView1.Columns(5).HeaderText = "Calle"
DataGridView1.Columns(6).Width = 80
DataGridView1.Columns(6).HeaderText = "Numero"
DataGridView1.Columns(7).Width = 80
DataGridView1.Columns(7).HeaderText = "Colonia"
DataGridView1.Columns(8).Width = 80
DataGridView1.Columns(8).HeaderText = "Codigo Postal"
DataGridView1.Columns(9).Width = 80
DataGridView1.Columns(9).HeaderText = "Municipio"
DataGridView1.Columns(10).Width = 80
DataGridView1.Columns(10).HeaderText = "Estado"
DataGridView1.Columns(11).Width = 80
DataGridView1.Columns(11).HeaderText = "Telefono"
DataGridView1.Columns(12).Width = 80
DataGridView1.Columns(12).HeaderText = "Celular"
DataGridView1.Columns(13).Width = 80
DataGridView1.Columns(13).HeaderText = "Fax"
cnn.Close()
End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
Dim cad As String = "Server=localhost;Database=sici;Uid=root;Pwd= ;Port=3306"
Dim comando As New MySqlCommand
Dim cnn As New MySqlConnection(cad)

cnn.Open()
comando.Connection = cnn
comando.CommandText = "Delete from proveedores1 where Cve_proveedor= " + TextBox1.Text + ""
comando.ExecuteNonQuery()
MsgBox("Se ha eliminado el Proveedor")
cnn.Close()
End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
Close()
End Sub

Private Sub BProveedores_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

End Sub
End Class

```

Código de ventana de Modificación de Proveedores

```

Imports MySql.Data
Imports MySql.Data.MySqlClient
Public Class MProveedores
Private Sub limpiar()
    TextBox1.Text = ""
    TextBox2.Text = ""
    TextBox3.Text = ""
    TextBox4.Text = ""
    TextBox5.Text = ""
    TextBox6.Text = ""
    TextBox7.Text = ""
    TextBox8.Text = ""
    TextBox9.Text = ""
    TextBox10.Text = ""
    TextBox11.Text = ""
    TextBox12.Text = ""
    TextBox13.Text = ""
    TextBox14.Text = ""

```

```
End Sub
Private Sub mostrar()
    TextBox2.Visible = True
    TextBox3.Visible = True
    TextBox4.Visible = True
    TextBox5.Visible = True
    TextBox6.Visible = True
    TextBox7.Visible = True
    TextBox8.Visible = True
    TextBox9.Visible = True
    TextBox10.Visible = True
    TextBox11.Visible = True
    TextBox12.Visible = True
    TextBox13.Visible = True
    TextBox14.Visible = True
    Label2.Visible = True
    Label3.Visible = True
    Label4.Visible = True
    Label5.Visible = True
    Label6.Visible = True
    Label7.Visible = True
    Label8.Visible = True
    Label9.Visible = True
    Label10.Visible = True
    Label11.Visible = True
    Label12.Visible = True
    Label13.Visible = True
    Label14.Visible = True
    Button4.Visible = True
End Sub
Private Sub ocultar()
    TextBox2.Visible = False
    TextBox3.Visible = False
    TextBox4.Visible = False
    TextBox5.Visible = False
    TextBox6.Visible = False
    TextBox7.Visible = False
    TextBox8.Visible = False
    TextBox9.Visible = False
    TextBox10.Visible = False
    TextBox11.Visible = False
    TextBox12.Visible = False
    TextBox13.Visible = False
    TextBox14.Visible = False
    Label2.Visible = False
    Label3.Visible = False
    Label4.Visible = False
    Label5.Visible = False
    Label6.Visible = False
    Label7.Visible = False
    Label8.Visible = False
    Label9.Visible = False
    Label10.Visible = False
    Label11.Visible = False
    Label12.Visible = False
    Label13.Visible = False
    Label14.Visible = False
    Button4.Visible = False
End Sub
Private Sub MProvedores_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    ocultar()
End Sub
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim existe As Boolean
    existe = False
    Dim re As Integer
```

```

Dim cad As String = "Server=localhost;Database=sici;Uid=root;Pwd= ;Port=3306"
Dim comando As New MySqlCommand
Dim cnn As New MySqlConnection(cad)

cnn.Open()
comando.Connection = cnn
comando.CommandText = "select *from proveedores1 where Cve_proveedor=" + TextBox1.Text + ""
Dim ad As New MySqlDataAdapter(comando.CommandText, cnn)
Dim tabla As New DataSet()
ad.Fill(tabla, "proveedores1")

While re < tabla.Tables("proveedores1").Rows.Count
    existe = True
    TextBox2.Text = tabla.Tables(re).Rows(re)(1).ToString()
    TextBox3.Text = tabla.Tables(re).Rows(re)(2).ToString()
    TextBox4.Text = tabla.Tables(re).Rows(re)(3).ToString()
    TextBox5.Text = tabla.Tables(re).Rows(re)(4).ToString()
    TextBox6.Text = tabla.Tables(re).Rows(re)(5).ToString()
    TextBox7.Text = tabla.Tables(re).Rows(re)(6).ToString()
    TextBox8.Text = tabla.Tables(re).Rows(re)(7).ToString()
    TextBox9.Text = tabla.Tables(re).Rows(re)(8).ToString()
    TextBox10.Text = tabla.Tables(re).Rows(re)(9).ToString()
    TextBox11.Text = tabla.Tables(re).Rows(re)(10).ToString()
    TextBox12.Text = tabla.Tables(re).Rows(re)(11).ToString()
    TextBox13.Text = tabla.Tables(re).Rows(re)(12).ToString()
    TextBox14.Text = tabla.Tables(re).Rows(re)(13).ToString()
    re = re + 1
End While
If existe = True Then
    mostrar()
Else
    MsgBox("La Clave del Proveedores No Existe")
End If
cnn.Close()
End Sub

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click
Dim cad As String = "Server=localhost;Database=sici;Uid=root;Pwd= ;Port=3306"
Dim comando As New MySqlCommand
Dim cnn As New MySqlConnection(cad)
cnn.Open()
comando.Connection = cnn
comando.CommandText = "update proveedores1 set RFC=" + TextBox2.Text + ", Apaterno=" + TextBox3.Text + ",
Amatemo=" + TextBox4.Text + ", Nombre=" + TextBox5.Text + ", Calle=" + TextBox6.Text + ", Numero=" +
TextBox7.Text + ", Colonia=" + TextBox8.Text + ", CP=" + TextBox9.Text + ", Delegacion=" + TextBox10.Text + ",
Estado=" + TextBox11.Text + ", Telefono=" + TextBox12.Text + ", Celular=" + TextBox13.Text + ", Fax=" +
TextBox14.Text + " where Cve_proveedores=" + TextBox1.Text + ""
comando.ExecuteNonQuery()
MsgBox("Se a modificado el Proveedores")
cnn.Close()
ocultar()
limpiar()
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
Dim cad As String = "Server=localhost;Database=sici;Uid=root;Pwd= ;Port=3306"
Dim comando As New MySqlCommand
Dim cnn As New MySqlConnection(cad)
Dim tabla As New DataTable

cnn.Open()
comando.Connection = cnn
Dim ad As New MySqlDataAdapter("Select *from proveedores1", cnn)
Dim Cmd As New MySqlCommandBuilder(ad)
ad.Fill(tabla)
DataGridView1.DataSource = tabla

```

```

DataGridView1.Columns(0).Width = 80
DataGridView1.Columns(0).HeaderText = "Clave de Proveedor"
DataGridView1.Columns(1).Width = 80
DataGridView1.Columns(1).HeaderText = "RFC"
DataGridView1.Columns(2).Width = 80
DataGridView1.Columns(2).HeaderText = "Apellido Paterno"
DataGridView1.Columns(3).Width = 80
DataGridView1.Columns(3).HeaderText = "Apellido Materno"
DataGridView1.Columns(4).Width = 80
DataGridView1.Columns(4).HeaderText = "Nombre"
DataGridView1.Columns(5).Width = 80
DataGridView1.Columns(5).HeaderText = "Calle"
DataGridView1.Columns(6).Width = 80
DataGridView1.Columns(6).HeaderText = "Numero"
DataGridView1.Columns(7).Width = 80
DataGridView1.Columns(7).HeaderText = "Colonia"
DataGridView1.Columns(8).Width = 80
DataGridView1.Columns(8).HeaderText = "Codigo Postal"
DataGridView1.Columns(9).Width = 80
DataGridView1.Columns(9).HeaderText = "Municipio"
DataGridView1.Columns(10).Width = 80
DataGridView1.Columns(10).HeaderText = "Estado"
DataGridView1.Columns(11).Width = 80
DataGridView1.Columns(11).HeaderText = "Telefono"
DataGridView1.Columns(12).Width = 80
DataGridView1.Columns(12).HeaderText = "Celular"
DataGridView1.Columns(13).Width = 80
DataGridView1.Columns(13).HeaderText = "Fax"
cnn.Close()
End Sub
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
    Close()
End Sub
End Class

```

Código de ventana de Consulta de Proveedores

```

Imports MySql.Data
Imports MySql.Data.MySqlClient
Public Class CProveedores
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Dim cad As String = "Server=localhost;Database=sici;Uid=root;Pwd=;Port=3306"
        Dim comando As New MySqlCommand
        Dim cnn As New MySqlConnection(cad)
        Dim tabla As New DataTable

        cnn.Open()
        comando.Connection = cnn
        Dim ad As New MySqlDataAdapter("Select *from proveedores1", cnn)
        Dim Cmd As New MySqlCommandBuilder(ad)
        ad.Fill(tabla)
        DataGridView1.DataSource = tabla
        DataGridView1.Columns(0).Width = 80
        DataGridView1.Columns(0).HeaderText = "Clave de Proveedor"
        DataGridView1.Columns(1).Width = 80
        DataGridView1.Columns(1).HeaderText = "RFC"
        DataGridView1.Columns(2).Width = 80
        DataGridView1.Columns(2).HeaderText = "Apellido Paterno"
        DataGridView1.Columns(3).Width = 80
        DataGridView1.Columns(3).HeaderText = "Apellido Materno"
        DataGridView1.Columns(4).Width = 80
        DataGridView1.Columns(4).HeaderText = "Nombre"
    End Sub
End Class

```

```

DataGridView1.Columns(5).Width = 80
DataGridView1.Columns(5).HeaderText = "Calle"
DataGridView1.Columns(6).Width = 80
DataGridView1.Columns(6).HeaderText = "Numero"
DataGridView1.Columns(7).Width = 80
DataGridView1.Columns(7).HeaderText = "Colonia"
DataGridView1.Columns(8).Width = 80
DataGridView1.Columns(8).HeaderText = "Codigo Postal"
DataGridView1.Columns(9).Width = 80
DataGridView1.Columns(9).HeaderText = "Municipio"
DataGridView1.Columns(10).Width = 80
DataGridView1.Columns(10).HeaderText = "Estado"
DataGridView1.Columns(11).Width = 80
DataGridView1.Columns(11).HeaderText = "Telefono"
DataGridView1.Columns(12).Width = 80
DataGridView1.Columns(12).HeaderText = "Celular"
DataGridView1.Columns(13).Width = 80
DataGridView1.Columns(13).HeaderText = "Fax"
cnn.Close()
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    Close()
End Sub

Private Sub CProveedores_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

End Sub
End Class

```

Código de ventana de Alta de Productos

```

Imports MySql.Data
Imports MySql.Data.MySqlClient
Public Class AProductos

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim cad As String = "Server=localhost;Database=sici;Uid=root;Pwd= ;Port=3306"
    Dim comando As New MySqlCommand
    Dim cnn As New MySqlConnection(cad)

    cnn.Open()
    comando.Connection = cnn
    comando.CommandText = "insert into producto values('" + TextBox1.Text + "','" + TextBox2.Text + "','" + TextBox3.Text
+ "','"
comando.ExecuteNonQuery()
    MsgBox("Se ha agregado el Producto")
    cnn.Close()
    MenuP.Show()
    TextBox1.Text = ""
    TextBox2.Text = ""
    TextBox3.Text = ""

End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    Close()
End Sub
End Class

```

Código de ventana de Baja de Productos

```
Imports Mysql.Data
Imports Mysql.Data.MySqlClient
Public Class BProductos

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
Dim cad As String = "Server=localhost;Database=sic;Uid=root;Pwd='';Port=3306"
Dim comando As New MySqlCommand
Dim cnn As New MySqlConnection(cad)

cnn.Open()
comando.CommandText = "Delete from producto where Cve_producto= " + TextBox1.Text + ""
comando.ExecuteNonQuery()
MsgBox("Se ha eliminado el Producto")
cnn.Close()

End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
Close()

End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
Dim cad As String = "Server=localhost;Database=sic;Uid=root;Pwd='';Port=3306"
Dim comando As New MySqlCommand
Dim cnn As New MySqlConnection(cad)

cnn.Open()
comando.CommandText = "Delete from producto where Cve_producto= " + TextBox1.Text + ""
comando.ExecuteNonQuery()
MsgBox("Se ha eliminado el Producto")
cnn.Close()

End Sub
```

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
Close()

End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
Dim cad As String = "Server=localhost;Database=sic;Uid=root;Pwd='';Port=3306"
Dim comando As New MySqlCommand
Dim cnn As New MySqlConnection(cad)

cnn.Open()
comando.CommandText = "Select *from producto ", cnn)
Dim cmd As New MySqlCommandBuilder(cmd)
ad.Fill(tabla)
DataGridView1.DataSource = tabla
DataGridView1.Columns(0).Width = 80
DataGridView1.Columns(0).HeaderText = "Clave de Producto"
DataGridView1.Columns(1).Width = 170
DataGridView1.Columns(1).HeaderText = "Descripcion"
DataGridView1.Columns(2).Width = 80
DataGridView1.Columns(2).HeaderText = "Precio"

cnn.Close()

End Sub
End Class
```

Código de ventana de Modificación de Productos

```
Imports Mysql.Data
Imports Mysql.Data.MySqlClient
Public Class MProductos

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
Dim cad As String = "Server=localhost;Database=sic;Uid=root;Pwd='';Port=3306"
Dim comando As New MySqlCommand
Dim cnn As New MySqlConnection(cad)
Dim tabla As New DataTable

cnn.Open()
comando.CommandText = "cnn
```

```

Dim ad As New MySqlDataAdapter("Select *from producto", cnn)
Dim Cmd As New MySqlCommandBuilder(ad)
ad.Fill(tabla)
DataGridView1.DataSource = tabla
DataGridView1.Columns(0).Width = 80
DataGridView1.Columns(0).HeaderText = "Clave de Producto"
DataGridView1.Columns(1).Width = 170
DataGridView1.Columns(1).HeaderText = "Descripción"
DataGridView1.Columns(2).Width = 80
DataGridView1.Columns(2).HeaderText = "Precio"

cnn.Close()

End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
Close()
End Sub

Private Sub MProductos_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
TextBox2.Visible = False
TextBox3.Visible = False
Label2.Visible = False
Label5.Visible = False
Button4.Visible = False

End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
Dim existe As Boolean
existe = False
Dim re As Integer
Dim cad As String = "Server=localhost;Database=sici;Uid=root;Pwd= ;Port=3306"
Dim comando As New MySqlCommand
Dim cnn As New MySqlConnection(cad)

cnn.Open()
comando.Connection = cnn
comando.CommandText = "select *from producto where Cve_producto= " + TextBox1.Text + ""
Dim ad As New MySqlDataAdapter(comando.CommandText, cnn)
Dim tabla As New DataSet()
ad.Fill(tabla, "producto")

While re < tabla.Tables("producto").Rows.Count
existe = True
TextBox2.Text = tabla.Tables(re).Rows(re)(1).ToString()
TextBox3.Text = tabla.Tables(re).Rows(re)(2).ToString()

re = re + 1
End While
If existe = True Then
TextBox2.Visible = True
TextBox3.Visible = True
Label2.Visible = True
Label5.Visible = True
Button4.Visible = True

Else
MsgBox("La Clave del Producto No Existe")
End If
cnn.Close()
End Sub

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click

```

```

Dim cad As String = "Server=localhost;Database=sici;Uid=root;Pwd= ;Port=3306"
Dim comando As New MySqlCommand
Dim cnn As New MySqlConnection(cad)
cnn.Open()
comando.Connection = cnn
comando.CommandText = "update producto set Descripcion="" + TextBox2.Text + ", Precio="" + TextBox3.Text + "
where Cve_producto="" + TextBox1.Text + ""
comando.ExecuteNonQuery()
MsgBox("Se a modificado el Producto")
cnn.Close()
TextBox2.Visible = False
TextBox3.Visible = False
Label2.Visible = False
Label5.Visible = False
Button4.Visible = False
TextBox1.Text = ""
TextBox2.Text = ""
TextBox3.Text = ""

End Sub
End Class

```

Código de ventana de Consulta de Productos

```

Imports MySql.Data
Imports MySql.Data.MySqlClient
Public Class CProductos

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
Close()
End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
Dim cad As String = "Server=localhost;Database=sici;Uid=root;Pwd= ;Port=3306"
Dim comando As New MySqlCommand
Dim cnn As New MySqlConnection(cad)
Dim tabla As New DataTable

cnn.Open()
comando.Connection = cnn
Dim ad As New MySqlDataAdapter("Select *from producto", cnn)
Dim Cmd As New MySqlCommandBuilder(ad)
ad.Fill(tabla)
DataGridView1.DataSource = tabla
DataGridView1.Columns(0).Width = 80
DataGridView1.Columns(0).HeaderText = "Clave de Producto"
DataGridView1.Columns(1).Width = 170
DataGridView1.Columns(1).HeaderText = "Descripción"
DataGridView1.Columns(2).Width = 80
DataGridView1.Columns(2).HeaderText = "Precio"

cnn.Close()
End Sub

Private Sub BProductos_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

End Sub
End Class

```

Código de ventana de Compras de Productos

```
Imports MySql.Data
Imports MySql.Data.MySqlClient
Public Class ACompras

Private Sub ACientes_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
Dim cad As String = "Server=localhost;Database=sici;Uid=root;Pwd= ;Port=3306"
Dim comando As New MySqlCommand
Dim cnn As New MySqlConnection(cad)
Dim a As Integer
Dim re As Integer
Dim b As Integer
Dim ra As Integer
Dim ri As Integer
Dim c As String
cnn.Open()
comando.Connection = cnn
Dim ad As New MySqlDataAdapter("Select * from Proveedores1", cnn)
Dim tabla As New DataSet()
ad.Fill(tabla, "Proveedores1")

While re < tabla.Tables("Proveedores1").Rows.Count
a = tabla.Tables("Proveedores1").Rows(re)(0).ToString()
ComboBox1.Items.Add(a)
re = re + 1
End While

Dim ad1 As New MySqlDataAdapter("Select * from compras", cnn)
Dim tabla1 As New DataSet()
ad1.Fill(tabla1, "compras")

While ra < tabla1.Tables("compras").Rows.Count

b = tabla1.Tables("compras").Rows(ra)(0).ToString()
ra = ra + 1
End While
TextBox1.Text = b + 1
TextBox2.Text = ""
TextBox3.Text = ""
TextBox4.Text = ""
TextBox5.Text = ""
TextBox6.Text = ""
TextBox7.Text = ""
TextBox8.Text = ""
TextBox9.Text = ""
TextBox10.Text = ""
TextBox11.Text = ""

Dim ad2 As New MySqlDataAdapter("Select * from producto", cnn)
Dim tabla2 As New DataSet()
ad2.Fill(tabla2, "producto")

While ri < tabla2.Tables("producto").Rows.Count

c = tabla2.Tables("producto").Rows(ri)(0).ToString()
ComboBox2.Items.Add(c)

ri = ri + 1
End While

cnn.Close()
TextBox7.Text = Date.Today

End Sub
```

```

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    Close()
End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
    Dim cad As String = "Server=localhost;Database=sici;Uid=root;Pwd= ;Port=3306"
    Dim comando As New MySqlCommand
    Dim cnn As New MySqlConnection(cad)
    Dim fecha As String
    Dim fecha2 As Date = DateTime.Now.Date.ToString
    Dim tot As Integer
    Dim re As Integer
    cnn.Open()
    comando.Connection = cnn
    comando.CommandText = "insert into producto_compras values('" + ComboBox2.Text + "','" + TextBox1.Text + "','" +
    TextBox9.Text + "','" + TextBox10.Text + "','" + TextBox11.Text + ');"
    comando.ExecuteNonQuery()
    cnn.Close()
    cnn.Open()
    fecha = fecha2.ToString.Substring(8, 2)
    fecha = fecha + fecha2.ToString.Substring(3, 2)
    fecha = fecha + fecha2.ToString.Substring(0, 2)

    comando.Connection = cnn
    comando.CommandText = "insert into compras values('" + TextBox1.Text + "','" + ComboBox1.Text + "','" +
    TextBox3.Text + "','" + TextBox4.Text + "','" + TextBox5.Text + "','" + TextBox6.Text + "','" + fecha & "','" + TextBox11.Text +
    ');"
    comando.ExecuteNonQuery()

    comando.Connection = cnn
    Dim ad As New MySqlDataAdapter("Select * from almacén", cnn)
    Dim tabla As New DataSet()
    ad.Fill(tabla, "almacén")

    While re < tabla.Tables("almacén").Rows.Count

        If ComboBox2.SelectedItem = tabla.Tables("almacén").Rows(re)(0).ToString() Then
            tot = tabla.Tables("almacén").Rows(re)(1).ToString()
            TextBox9.Text = tot + TextBox9.Text
            comando.Connection = cnn
            comando.CommandText = "update almacén set Bullos=" + TextBox9.Text + " where Cve_producto=" +
            ComboBox2.Text + ""
            comando.ExecuteNonQuery()
        End If

        re = re + 1
    End While

    MsgBox("Se ha agregado la Compra")
    cnn.Close()
    MenuP.Show()
    TextBox1.Text = ""
    TextBox2.Text = ""
    TextBox3.Text = ""
    TextBox4.Text = ""
    TextBox5.Text = ""
    TextBox6.Text = ""
    TextBox7.Text = ""
    TextBox8.Text = ""
    TextBox9.Text = ""
    TextBox10.Text = ""
    TextBox11.Text = ""

    Close()
End Sub

```

```

Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
ComboBox1.SelectedIndexChanged
    Dim cad As String = "Server=localhost;Database=sici;Uid=root;Pwd= ;Port=3306"
    Dim comando As New MySqlCommand
    Dim cnn As New MySqlConnection(cad)
    Dim re As Integer
    cnn.Open()
    comando.Connection = cnn
    Dim ad As New MySqlDataAdapter("Select * from proveedores1", cnn)
    Dim tabla As New DataSet()
    ad.Fill(tabla, "proveedores1")

    While re < tabla.Tables("proveedores1").Rows.Count

        If ComboBox1.SelectedItem = tabla.Tables("proveedores1").Rows(re)(0).ToString() Then

            TextBox2.Text = tabla.Tables("proveedores1").Rows(re)(3).ToString() + " " +
tabla.Tables("proveedores1").Rows(re)(4).ToString() + " " + tabla.Tables("proveedores1").Rows(re)(5).ToString()

            End If
            re = re + 1
        End While

        cnn.Close()
    End Sub

Private Sub ComboBox2_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
ComboBox2.SelectedIndexChanged
    Dim cad As String = "Server=localhost;Database=sici;Uid=root;Pwd= ;Port=3306"
    Dim comando As New MySqlCommand
    Dim cnn As New MySqlConnection(cad)
    Dim re As Integer
    cnn.Open()
    comando.Connection = cnn
    Dim ad As New MySqlDataAdapter("Select * from producto", cnn)
    Dim tabla As New DataSet()
    ad.Fill(tabla, "producto")

    While re < tabla.Tables("producto").Rows.Count

        If ComboBox2.SelectedItem = tabla.Tables("producto").Rows(re)(0).ToString() Then

            TextBox8.Text = tabla.Tables("producto").Rows(re)(1).ToString()

            End If

            re = re + 1
        End While

        cnn.Close()
    End Sub

Private Sub TextBox10_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
TextBox10.TextChanged

    If TextBox10.Text <> "0" And TextBox10.Text.Length <> 0 Then
        TextBox11.Text = TextBox9.Text * TextBox10.Text
    Else
        TextBox11.Text = ""
    End If
End Sub

End Class

```

Código de ventana de Ventas de Productos

```
Imports MySql.Data
Imports MySql.Data.MySqlClient
Public Class Ventas

Private Sub ACientes_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
Dim cad As String = "Server=localhost;Database=sici;Uid=root;Pwd= ;Port=3306"
Dim comando As New MySqlCommand
Dim cnn As New MySqlConnection(cad)
Dim a As Integer
Dim re As Integer
Dim b As Integer
Dim ra As Integer
Dim ri As Integer
Dim c As String
cnn.Open()
comando.Connection = cnn
Dim ad As New MySqlDataAdapter("Select * from clientes1", cnn)
Dim tabla As New DataSet()
ad.Fill(tabla, "clientes1")

While re < tabla.Tables("clientes1").Rows.Count

a = tabla.Tables("clientes1").Rows(re)(0).ToString()
ComboBox1.Items.Add(a)

re = re + 1
End While
Dim ad1 As New MySqlDataAdapter("Select * from ventas", cnn)
Dim tabla1 As New DataSet()
ad1.Fill(tabla1, "ventas")

While ra < tabla1.Tables("ventas").Rows.Count

b = tabla1.Tables("ventas").Rows(ra)(0).ToString()
ra = ra + 1
End While
TextBox1.Text = b + 1

TextBox2.Text = ""
TextBox7.Text = ""
TextBox8.Text = ""
TextBox9.Text = ""
TextBox10.Text = ""
TextBox11.Text = ""

Dim ad2 As New MySqlDataAdapter("Select * from producto", cnn)
Dim tabla2 As New DataSet()
ad2.Fill(tabla2, "producto")

While ri < tabla2.Tables("producto").Rows.Count

c = tabla2.Tables("producto").Rows(ri)(0).ToString()
ComboBox2.Items.Add(c)

ri = ri + 1
End While

cnn.Close()

```

```

    TextBox7.Text = Date.Today

End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    Close()
End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
    Dim cad As String = "Server=localhost;Database=sici;Uid=root;Pwd=;Port=3306"
    Dim comando As New MySqlCommand
    Dim cnn As New MySqlConnection(cad)
    Dim fecha As String
    Dim fecha2 As Date = DateTime.Now.Date.ToString
    Dim tot As Integer
    Dim re As Integer

    cnn.Open()
    Dim ad As New MySqlDataAdapter("Select * from almacén", cnn)
    Dim tabla As New DataSet()
    ad.Fill(tabla, "almacén")

    While re < tabla.Tables("almacén").Rows.Count

        If tabla.Tables("almacén").Rows(re)(1).ToString() >= TextBox9.Text Then
            comando.Connection = cnn
            comando.CommandText = "insert into producto_venta values(" + ComboBox2.Text + "," + TextBox1.Text + "," +
            TextBox9.Text + "," + TextBox11.Text + ");"
            comando.ExecuteNonQuery()

            fecha = fecha2.ToString.Substring(8, 2)
            fecha = fecha + fecha2.ToString.Substring(3, 2)
            fecha = fecha + fecha2.ToString.Substring(0, 2)

            comando.Connection = cnn
            comando.CommandText = "insert into ventas values(" + TextBox1.Text + "," + ComboBox1.Text + "," +
            TextBox11.Text + " & " & fecha & ");"
            comando.ExecuteNonQuery()

            comando.Connection = cnn
            Dim ad1 As New MySqlDataAdapter("Select * from almacén", cnn)
            Dim tabla1 As New DataSet()
            ad1.Fill(tabla1, "almacén")

            While re < tabla.Tables("almacén").Rows.Count

                If ComboBox2.SelectedItem = tabla1.Tables("almacén").Rows(re)(0).ToString() Then

                    tot = tabla1.Tables("almacén").Rows(re)(1).ToString()
                    TextBox9.Text = tot - TextBox9.Text

                    comando.Connection = cnn
                    comando.CommandText = "update almacén set Bultos=" + TextBox9.Text + " where Cve_producto=" +
                    ComboBox2.Text + ""
                    comando.ExecuteNonQuery()
                End If

                re = re + 1
            End While

            MsgBox("Se ha agregado la Venta")
        End If
    End While
End Sub

```

```

Else
MsgBox("No hay Producto...")
End If

re = re + 1
End While
cnn.Close()

MenuP.Show()
TextBox1.Text = ""
TextBox2.Text = ""
TextBox7.Text = ""
TextBox8.Text = ""
TextBox9.Text = ""
TextBox10.Text = ""
TextBox11.Text = ""

Close()
End Sub

Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
ComboBox1.SelectedIndexChanged
Dim cad As String = "Server=localhost;Database=sici;Uid=root;Pwd= ;Port=3306"
Dim comando As New MySqlCommand
Dim cnn As New MySqlConnection(cad)
Dim re As Integer
cnn.Open()
comando.Connection = cnn
Dim ad As New MySqlDataAdapter("Select * from clientes1", cnn)
Dim tabla As New DataSet()
ad.Fill(tabla, "clientes1")

While re < tabla.Tables("clientes1").Rows.Count

If ComboBox1.SelectedItem = tabla.Tables("clientes1").Rows(re)(0).ToString() Then
If tabla.Tables("clientes1").Rows(re)(2).ToString().Length = 0 Then
TextBox2.Text = tabla.Tables("clientes1").Rows(re)(3).ToString() + " " +
tabla.Tables("clientes1").Rows(re)(4).ToString() + " " + tabla.Tables("clientes1").Rows(re)(5).ToString()

Else
TextBox2.Text = tabla.Tables("clientes1").Rows(re)(2).ToString()
End If
End If

re = re + 1
End While
cnn.Close()
End Sub

Private Sub ComboBox2_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
ComboBox2.SelectedIndexChanged
Dim cad As String = "Server=localhost;Database=sici;Uid=root;Pwd= ;Port=3306"
Dim comando As New MySqlCommand
Dim cnn As New MySqlConnection(cad)
Dim re As Integer
cnn.Open()
comando.Connection = cnn
Dim ad As New MySqlDataAdapter("Select * from producto", cnn)
Dim tabla As New DataSet()
ad.Fill(tabla, "producto")

```

```
While re < tabla.Tables("producto").Rows.Count
    If ComboBox2.SelectedItem = tabla.Tables("producto").Rows(re)(0).ToString() Then
        TextBox8.Text = tabla.Tables("producto").Rows(re)(1).ToString()
        TextBox10.Text = tabla.Tables("producto").Rows(re)(2).ToString()

    End If
    re = re + 1
End While

conn.Close()
End Sub

Private Sub TextBox9_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
TextBox9.TextChanged
    If TextBox9.Text <> "0" And TextBox9.Text.Length <> 0 Then
        TextBox11.Text = TextBox9.Text * TextBox10.Text
    Else
        TextBox11.Text = ""
    End If
End Sub

End Class
```

ANEXO 4 Diagrama de Gantt.

A	T	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Recopilación de información		■	■																												
Analisis			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
* Sistema Actual			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
* Sistema Ideal																															
Diseño																															
Codificación																															
Pruebas																															
Instalación																															

A	T	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	
Recopilación de información																																
Analisis																																
* Sistema Actual																																
* Sistema Ideal																																
Diseño																																
Codificación																																
Pruebas																																
Instalación																																

A= Actividades
T= Tiempo (días)

ANEXO 5

Glosario de términos.

B2B. Es la transmisión de información referente a transacciones comerciales electrónicamente.

Computadora. Es una colección de circuitos integrados y otros componentes relacionados que puede ejecutar con exactitud, rapidez y de acuerdo a lo indicado por un usuario o automáticamente por otro programa.

Encapsulación. Es el proceso de ocultar todos los detalles de un objeto que no contribuyen a sus características esenciales. Esto significa que aquello que está en interior de la clase está oculto, solo las interfaces externas pueden ver otros objetos.

GB. Es una unidad de medida informática cuyo símbolo es el GB, y puede equivalerse a 2^{30} bytes o a 10^9 bytes, según el uso.

GHZ. Es un múltiplo de la unidad de medida de frecuencia hercio (Hz) y equivale a 10^9 (1.000.000.000).

MB. Es una unidad de medida de cantidad de datos informáticos. Es un múltiplo del octeto, que equivale a 10^6 (1.000.000 octetos).

Memoria RAM. Es la memoria desde donde el procesador recibe las instrucciones y guarda los resultados.

Polimorfismo. Se refiere a la posibilidad de definir clases diferentes que tienen métodos o atributos denominados de forma idéntica, pero que se comportan de manera distinta.

RDBMS. Es un Sistema Administrador de Bases de Datos Relacionales. RDBMS viene del acrónimo en inglés *Relational Data Base Management System*.

SQL. Es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en éstas.

WEB. World Wide Web, el sistema de documentos (o páginas web) interconectados por enlaces de hipertexto, disponibles en Internet.