

INTRODUCCIÓN

INTRODUCCIÓN

El mundo de la informática nunca hubiere existido de no ser por el desarrollo del ordenador o computador.

Que nos ha facilitado el procesamiento de datos, manipulaciones y almacenamiento de documentos, imágenes, programas.

Los equipos informáticos han abierto una nueva era en la fabricación gracias a las técnicas de automatización que son herramientas esenciales prácticamente en todos los campos de investigación y en tecnología aplicada.

En este proyecto de tesis se pretende alcanzar los conocimientos técnicos para el desarrollo de proyectos de automatización mediante lenguaje de programación Microsoft Visual Studio 2008, utilizando SQL Server 2005 con su respectivas tablas en donde se van almacenar los registros de las Ventas Diarias en la Despensa "Santa Fe" , el ingresos de los productos con sus respectivos precios, el sistema va a permitir de una manera eficaz y rápida despachar a los clientes mediante factura de venta, consultas, ingresos, modificaciones, eliminaciones de los respectivos registros

A la vez el sistema cuenta con un formulario de identificación mediante contraseña para el acceso al sistema, un manual técnico para futuras actualizaciones el cual cuenta con unos glosarios de palabras o comandos que se han utilizada en la elaboración de este proyecto.

Para facilitar la búsqueda sobre cualquiera de los temas o pasos a seguir sobre cualquier formulario del proyecto se ha desarrollado un índice detallado sobre la elaboración de cada uno de los pasos a seguir en la elaboración de este proyecto.

SECTOR SOLICITANTE

Despensa Santa Fe.

DEFINICIÓN DE GRUPOS DE TRABAJO

Las personas que han participado en todas las etapas de desarrollo de este proyecto y merecen nuestro reconocimiento y gratitud por su labor solicitado:

Sr. **Ángel Quinche Ramírez.** PROPIETARIO DESPENSA "SANTA FE"

Egdo. Ronald Cruz. PROPONENTE

Egdo. Ángel Quinche. PROPONENTE

Ing. Nelly Esparza. DIRECTORA DE TESIS

Ing. María Gonzales LECTORA DE TESIS

CAPÍTULO I:
PLANTEAMIENTO.

1. EL PROBLEMA

1.1. PLANTEAMIENTO DEL PROBLEMA

Al realizar un análisis en la despensa "Santa Fe", observamos que en la despensa no había un correcto funcionamiento en la organización de los productos y la realización de las ventas.

Al momento de hacer un control de la mercadería existente en la despensa, había muchos problemas, lo que ocasionaba una pérdida de tiempo significativa en esta pequeña empresa.

En cuanto a las ventas, no se tenía detallado el ingreso que permitía subsistir a la empresa.

El gerente de esta despensa posee dos computadoras pero no las utiliza con fines de administración de información en su negocio.

Actualmente la despensa no cuenta con un sistema informático que permita el control del inventario y ventas de los productos que ofrece a sus clientes.

El desarrollo de Software es un proceso complejo para el que se debe elegir metodologías que permitan organizar los procesos y llegar a la culminación exitosa de proyectos.

Mediante el software que vamos a implementar nos permitirá:

- Trabajar de una forma coordinada.
- Integrar las múltiples facetas para realizar ventas de manera más fácil.
- Proporcionar una guía para ordenar el control de inventario existente.
- Optimizar recursos para beneficio de la despensa.

1.2. FORMULACIÓN DEL PROBLEMA

¿Cómo mejorar el control de mercadería y ventas en la despensa “Santa Fe”, del cantón Caluma?

1.3. DELIMITACIÓN.

La investigación se llevara a cabo en la despensa “Santa Fe”, del cantón Caluma, ubicado en el sector “San Francisco”, en las calles 23 de agosto y Dr. Vinicio Noboa de la Provincia de Bolívar.

El presente trabajo investigativo se ha planificado para realizarse en un periodo de cuatro meses.

1.3.1. ALCANCE.

En el presente trabajo de investigación se pondrá gran énfasis en el estudio de los Sistemas Informáticos, procurando dar más atención a las metodologías para el desarrollo en este campo y su utilidad para el desarrollo del mismo. Sin embargo este sistema facilitará el manejo del servicio que brinda la Despensa “Santa Fe”, contribuyendo así al mejoramiento de las operaciones diarias que tienen que realizar en las Ventas. A continuación se detalla la solución que esté encierra:

- Actualización de la Mercadería en caso de que suban los precios de los productos.
- Eliminaciones de Mercadería, si la despensa decide no vender los respectivos productos.
- Registro de Categoría nueva, si la empresa decide vender productos que corresponden a una nueva categoría.
- Eliminaciones de Categoría, en caso que dicha categoría no se utilice en los productos que se tienen en bodega.
- Registro de Cliente nuevo, se utiliza cuando un nuevo cliente realiza una compra.
- Registro de Proveedor Nuevo, se utiliza para tener información de las empresas que nos proveen de productos nuevos.

- Actualización del Proveedor, en caso de que el proveedor cambie de dirección, teléfono, etc.
- Eliminación de Proveedor, se utiliza cuando el proveedor no va a facilitar productos a la despensa.
- Registro de Empleado Nuevo, para tener información detallada de los empleados que hay en la empresa.
- Consultas Generales, para tener información general de los datos que hay en las diferentes tablas de la base de datos.
- Búsqueda de productos por Código, se utiliza para ver información de un determinado producto a partir de su código.
- Búsqueda de productos por Nombre, se utiliza para ver información de un determinado producto a partir de su nombre.
- Búsqueda de productos por Fecha de entrada, se utiliza para ver información de un determinado producto a partir de su Fecha de entrada.
- Búsqueda de productos por Categoría, se utiliza para ver información de un determinado producto a partir de su categoría.
- Búsqueda de productos por Fecha de caducidad, se utiliza para ver información de un determinado producto a partir de su Fecha de caducidad.
- Poner en oferta los productos que están por caducar, con el objetivo de que la mercadería de la despensa no se pierda, para no perder lo invertido.
- Utilización de gráficos estadísticos en ventas realizadas por mes, para tener una mejor forma de entender las ventas realizadas.
- Descuentos en ventas, cuando el precio total de la venta sea una cantidad mayor a \$50.

Generación de todos los Reportes Correspondientes, porque siempre es importante tener reportes de lo que se realiza a diario en la despensa.

1.3.2. RECURSOS.

1.3.2.1. Recursos Humanos.

Para esto se ha considerado a las personas detallada en la siguiente tabla:

CANTIDAD	PERSONAL
2	Egresados
1	Profesora Directora
1	Profesora Lectora
1	Administradora De La Despensa

1.3.2.2. Recursos De Hardware.

Para la realización del sistema se utilizara el siguiente recurso de hardware:

- ✓ Utilización de dos computadoras.
- ✓ Procesador Dual Core de 2.8 GHz.
- ✓ Memoria 2 GB.
- ✓ Disco Duro de 280 GB.
- ✓ Mainboard Intel.
- ✓ Tarjeta De Red Power Net 10/100/1000 Mbps.
- ✓ Monitor 17".
- ✓ Mouse Óptico.
- ✓ Teclado.
- ✓ Ups.
- ✓ DVD Writer.
- ✓ Impresora Lexmark. X3550.

1.3.2.3. Recursos De Software.

Los Software utilizados para la realización del Sistema serán:

- ✓ Sistema Operativo Windows XP.
- ✓ Software de Base de Datos SQL SERVER 2005.
- ✓ Software de Programación Visual Studio 2008.

- ✓ Software de Microsoft Office 2007.
- ✓ Software Windows Server 2008.

1.3.2.4. Materiales A Utilizar.

- ✓ Materiales A Utilizar.
- ✓ Hojas De Papel.
- ✓ Cartuchos De Tinta.
- ✓ Carpetas.
- ✓ Útiles De Oficina.
- ✓ 5 CD's en blanco para copiar información.
- ✓ 1 pen drive de 4 GB.

1.3.2.5. Otros Materiales.

- ✓ Internet.
- ✓ Quemada de CD 's.
- ✓ Bibliografía adicional o copias.
- ✓ Información del director de tesis.
- ✓ Información administrativa de La Despensa.

1.4. OBJETIVOS.

1.4.1. Objetivo General.

- Desarrollar un sistema Informático para mejorar la gestión de ventas e inventario de mercaderías de la despensa "Santa Fe" del cantón Caluma.

1.4.2. Objetivos Específicos.

- Fundamentar la realización del inventario y realización de ventas en la despensa "Santa Fe" del cantón Caluma.
- Recolectar la información necesaria para poder realizar la automatización.
- Implementar un software para mejorar la gestión de ventas e inventarios en la despensa para optimizar tiempo y recursos.

1.5. JUSTIFICACIÓN DE LA INVESTIGACIÓN.

La Despensa "Santa Fe", requiere de carácter urgente implementar soluciones positivas al problema ya antes mencionado para ejercer controles adecuados y ayudar al personal a desarrollar su trabajo de modo más seguro y eficiente.

Actualmente, en cualquier actividad, la independencia de las instituciones y empresas adquieren una importancia estratégica. Por lo tanto se debe aprovechar los avances de la tecnología para dar una respuesta eficiente y eficaz a los problemas que se presenten en el manejo de la empresa.

Los sistemas de información sirven para lograr ventajas en el terreno de los negocios, y éstas representan un diferencial o valor agregado con respecto a los competidores.

La perspectiva estratégica considera a los sistemas de información como una herramienta para mejorar la estructura competitiva del negocio, por lo que tienen su área de influencia en el medio ambiente de la organización, a través de nuevos servicios a clientes, nuevos negocios y oportunidades de inversión.

Siendo necesario y posible de ejecutar gracias al avance tecnológico experimentado en los últimos años y a la reducción de los costos del equipamiento que sirve de soporte para el desarrollo y la implantación de sistemas de información; se puede aprovechar adecuadamente estas posibilidades para dar a la entidad una ventaja estratégica y convertir los sistemas de información en un factor crítico de éxito para el desarrollo de sus actividades.

La aportación será muy significativa, puesto que a través de la aplicación de este sistema se brindaran diferentes beneficios para la Despensa "Santa Fe" a la cual está destinada su funcionamiento.

CAPÍTULO II:
MARCO TEÓRICO

2. MARCO TEÓRICO.

2.1. ANTECEDENTES DE LA INVESTIGACIÓN.

Realizadas las investigaciones en la Despensa "Santa Fe" Del Cantón Caluma Provincia Bolívar durante un año, se puede afirmar que no existe un sistema informático que lleve el registro confiable del manejo de las ventas diarias y cumpla con las necesidades de los clientes que llegan a comprar en dicha despensa, por lo cual realizaremos una recolección de datos detallada con el personal administrativo de la Despensa, para poder generarlo sin ningún tipo de problemas al momento de ser entregado el producto; ya que éste sistema permitirá llevar un mejor control de las ventas, brindándole así mayor seguridad al momento de vender los productos a los clientes y mayor control a los administrativos de la Despensa "Santa Fe".

El cual podrá mostrar un servicio rápido y eficaz mediante los resultados que se obtengan cuando ingrese código del producto para realizar una búsqueda rápida para el cliente de dicho producto, y temas que están almacenados en la base de datos de la Despensa "Santa Fe" Del Cantón Caluma, por medio del diseño e implementación de este software.

Los beneficios directos de esta investigación serán para los administradores de la Despensa por cada producto que se encuentra en stock.

2.2. FUNDAMENTACIÓN CIENTÍFICA.

2.2.1. DESPENSA SANTA FE.

2.2.1.1. Historia.

El 12 de agosto del 2005 se constituye en el cantón Caluma provincia de Bolívar la Despensa Santa Fe, el mismo que tiene su ubicación en las calles 23 de Agosto y Doc. Vinicio Novoa, siendo su propietario el Señor Ángel Claudio Quinche Ramírez, el cual tenía este proyecto en mente ya que su familia está vinculado al área comercial.

Al transcurrir los días las ventas aumentaban por lo cual se vio obligado a contratar una ayudante de tal forma que los dos atendían en la Despensa.

En la actualidad dos personas presta servicio en la Despensa, más la administradora de manera que los recursos humanos inmersos en este negocio son tres.

2.2.1.2. Objetivos Del Negocio.

- Mantener el nivel de variedad en todas las líneas que ofrece el negocio e implementar nuevas secciones de ventas.
- Brindar excelente atención a los clientes.
- Vender de todo a menor precio que la competencia.

2.2.1.3. Misión.

La apertura de una nueva Despensa Santa Fe en puntos estratégicos del Cantón Caluma y Provincia previo a un estudio técnico de mercado.

2.2.1.4. Visión.

Alcanzar el primer lugar en línea en la Despensa Santa Fe del Cantón Caluma y Provincia, los mismos que ofrezcan toda clase de producto en ventas para satisfacer las necesidades del cliente.

2.2.2. DESPENSA EN EL ECUADOR.

2.2.2.1. Estudio Del Mercado.

El **estudio de mercado** consiste en una iniciativa empresarial con el fin de hacerse una idea sobre la viabilidad comercial de una actividad económica. El estudio de mercado consta de 3 grandes análisis:

1. Análisis del consumidor

Estudia el comportamiento de los consumidores para detectar sus necesidades de consumo y la forma de satisfacerlas, averiguar sus hábitos de compra (lugares, momentos, preferencias...), etc. Su objetivo final es aportar datos que permitan mejorar las técnicas de mercado para la venta de un producto o de una serie de productos que cubran la demanda no satisfecha de los consumidores.

2. Análisis de la competencia

Estudia el conjunto de empresas con las que se comparte el mercado del mismo producto.

Para realizar un estudio de la competencia es necesario establecer quienes son los competidores, cuántos son y sus respectivas ventajas competitivas. El plan de negocios podría incluir una plantilla con los competidores más importantes y el análisis de algunos puntos como: marca, descripción del producto o servicio, precios, estructura, procesos, recursos humanos, costes, tecnología, imagen, proveedores, entre otros.

3. Estrategia

Concepto breve pero imprescindible que marcará el rumbo de la empresa. Basándose en los objetivos, recursos y estudios del mercado y de la competencia debe definirse una estrategia que sea la más adecuada para la nueva empresa. Toda empresa deberá optar por dos estrategias posibles:

1. **Liderazgo en costos.-** Consiste en mantenerse competitivo a través de aventajar a la competencia en materia de costos.
2. **Diferenciación.-** Consiste en crear un valor agregado sobre el producto ofrecido para que este sea percibido en el mercado como único: diseño, imagen, atención a clientes, entrega a domicilio.

2.2.3. SISTEMA DE INFORMACIÓN.

2.2.3.1. Introducción.

Un Sistema de Información es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio. En un sentido amplio, un sistema de información no necesariamente incluye equipo electrónico (hardware). Sin embargo en la práctica se utiliza como sinónimo de "sistema de información computarizado".

Los elementos que interactúan entre sí son: el equipo computacional, el recurso humano, los datos o información fuente, programas ejecutados por las computadoras, las telecomunicaciones y los procedimientos de políticas y reglas de operación.

2.2.3.2. Actividades Básicas Del Sistema De Información.

Son Cuatro actividades Básicas:

1. **Entrada de información:** Proceso en el cual el sistema toma los datos que requiere para procesar la información, por medio de estaciones de trabajo, teclado, cintas magnéticas, código de barras, etc.
2. **Almacenamiento de información:** Es una de las actividades más importantes que tiene una computadora, ya que a través de esta propiedad el sistema puede recordar la información guardar en la sesión o proceso anterior.
3. **Procesamiento de la información:** Esta característica de los sistemas permite la transformación de los datos fuente en

4. información que puede ser utilizada para la toma de decisiones, lo que hace posible, entre otras cosas, que un tomador de decisiones genere una proyección financiera a partir de los datos que contiene un estado de resultados o un balance general en un año base.
5. **Salida de información:** Es la capacidad de un SI para sacar la información procesada o bien datos de entrada al exterior. Las unidades típicas de salida son las impresoras, graficadores, cintas magnéticas, la voz, etc.

2.2.4. LENGUAJE DE PROGRAMACIÓN.

2.2.4.1. Introducción.

Un lenguaje de programación es un idioma artificial diseñado para expresar computaciones que pueden ser llevadas a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana. Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Al proceso por el cual se escribe, se prueba, se depura, se compila y se mantiene el código fuente de un programa informático se le llama programación. La programación consiste en desarrollar programas para procesar información y tiene como objetivo el tratamiento de la información correctamente, con lo que se espera que un programa proporcione el resultado correcto, es decir, que cada aplicación funcione según lo esperado en términos de programación.

Otro objetivo fundamental de la programación es que sean de códigos claros y legibles, así otro programador pueda ser capaz de entender la codificación, modificarla y ampliarla. Así mismo lo más importante es la corrección, ya que un código claro y legible facilita el mantenimiento de la aplicación o sistema.

También la palabra programación se define como el proceso de creación de un programa de computadora, mediante la aplicación de procedimientos lógicos, a través de los siguientes pasos:

- El desarrollo lógico del programa para resolver un problema en particular.
- Escritura de la lógica del programa empleando un lenguaje de programación específico (codificación del programa).
- Ensamblaje o compilación del programa hasta convertirlo en lenguaje de máquina.
- Prueba y depuración del programa.
- Desarrollo de la documentación.

2.2.5. MICROSOFT VISUAL STUDIO 2008.

2.2.5.1. Introducción.

Es un lenguaje orientado a objetos y eventos que soporta **encapsulación, herencia y 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, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión net 2002). Así se pueden crear aplicaciones que se intercomunican entre estaciones de trabajo, páginas web y dispositivos móviles.

2.2.5.2. Entorno de Desarrollo

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).

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

Si la solución tiene un único proyecto, abrir el archivo de proyecto **.vbproj** o el archivo de solución **.sln** tiene el mismo resultado, pero si la solución es multiproyecto entonces deberá abrir el archivo de solución.

2.2.5.3. .NET Framework

Visual Studio .NET tiene una nueva herramienta que comparte con Visual Basic, Visual C++, Visual C#, etc.

Llamada .NET Framework que además es una interfaz subyacente que forma parte del propio sistema operativo Windows.

La estructura de .NET Framework es por Clases mismas que puede incorporar a sus proyectos a través de la instrucción Imports, por ejemplo una de sus Clases es System.Math la cual soporta los siguientes métodos.

Método	Descripción
Abs(n)	Calcula el valor absoluto de n
Atan(n)	Calcula el arcotangente de n en radianes
Cos(n)	Calcula el coseno del ángulo n expresado en radianes
Exp(n)	Calcula el constante de e elevada a n
Sign(n)	Regresa -1 si n es menor que cero, 0 si n es cero y +1 si n es mayor a cero
Sin(n)	Calcula el seno del ángulo n expresado en radianes
Sqr(n)	Calcula la raíz cuadrada de n.
Tan(n)	Calcula la tangente del ángulo n expresado en radianes

2.2.5.3.1. Framework (.Net 3.5).

El nuevo framework (.Net 3.5) está diseñado para aprovechar las ventajas que ofrece el nuevo sistema operativo "Windows Vista" a través de sus subsistemas "Windows Communication Foundation" (WCF) y "Windows Presentation Foundation" (WPF). El primero tiene como objetivo la construcción de aplicaciones orientadas a servicios mientras que el último apunta a la creación de interfaces de usuario más dinámicas que las conocidas hasta el momento.

2.2.5.4. NET

La biblioteca de clases .NET es una biblioteca de clases incluida en el Microsoft .NET Framework y está diseñada para ser la base sobre las cuales las aplicaciones .NET son construidas.

La biblioteca **.NET** contiene un número considerable de clases con código reutilizable, para ayudar a controlar esta complejidad **.NET** utiliza namespaces, colecciones de clases relacionadas.

2.2.5.5. Nuevo Proyecto

De clic en el botón [**New Project**] o File/New/Project, como tipo de proyecto seleccione Visual Basic Project, como plantilla seleccione **Windows Application**, por último indique la ubicación donde desea almacenar su proyecto. Al dar clic Visual Studio configura el entorno de

desarrollo y crea un directorio con el mismo nombre que específico para la aplicación.

2.2.5.5.1. Aplicación Windows Ejecutable

Las aplicaciones Windows creadas con Visual Basic .NET tienen la extensión **.exe** mismas que podrán ser ejecutadas en cualquier equipo que tenga instalado Microsoft Windows. Visual Basic .NET instala de manera automática los archivos de soporte, incluyendo las bibliotecas de vínculos dinámicos y archivos de .NET framework.

Visual Studio puede crear dos tipos de archivos ejecutables:

- 1) **Debug** - Compilación de depuración, se utiliza cuando se prueba y depura un programa (Default).
- 2) **Release** - Versión de edición, se utiliza cuando se termina el programa siendo esta una versión optimizada de menor tamaño

Para crear un archivo ejecutable, de clic en Build/Configuration Manager para decidir sobre el tipo de archivo ejecutable, además de otras opciones como la plataforma para la cual desea crear la aplicación.

2.2.5.6. Programación Orientada A Eventos

Visual Basic .NET soporta la Programación Orientada a Eventos en la cual las aplicaciones reconocen y responden a eventos.

2.2.5.6.1. Evento

Un **Evento** es una acción o acontecimiento reconocido por algunos objetos para los cuales es necesario escribir el código para responder a dicho evento. Los eventos pueden ocurrir como resultado de una acción del usuario (onClick), por invocación a través de código o disparados por el sistema (Timer Tick Event).

2.2.5.6.2. Manejador De Evento

Un Manejador de Eventos contiene código que responde a eventos particulares. Un desarrollador diseña cuidadosamente sus aplicaciones.

determinando los controles disponibles para el usuario y los eventos apropiados asociados a estos controles, entonces, el desarrollador escribe el código para integrar los eventos consistentes con el diseño de la aplicación.

2.2.5.6.3. Procedimiento

Un **procedimiento** es un conjunto de sentencias que realizan una acción lógica. Existen tres tipos de procedimientos en Visual Basic .NET:

1. Event procedures/Event handler, procedimiento que contiene código que es ejecutado en respuesta a un evento. Cuando el evento es disparado el código dentro del manejador de eventos es ejecutado.

Visual Basic .NET para los manejadores de eventos utiliza una convención estándar la cual combina el nombre del objeto seguido de un guión bajo y el nombre del evento.

```
Private|Public Sub objeto_Evento(parámetros) handles Objeto.Evento
```

sentencias

```
End Sub
```

2.2.5.7. Windows Form

Los formularios son el elemento básico del interfaz de usuario (IU) en aplicaciones creadas para Microsoft Windows®. Proporcionan un marco de trabajo que puede utilizarse por toda la aplicación para crear un aspecto coherente. Los formularios de aplicaciones basadas en Windows se utilizan para presentar información al usuario y aceptar la introducción de datos por parte del mismo.

Los formularios exponen propiedades que definen su apariencia, métodos que definen su comportamiento, y eventos que definen su interacción con el usuario. Estableciendo las propiedades y escribiendo código para responder a sus eventos, el formulario se personaliza para.

satisfacer los requerimientos de las aplicaciones. El formulario es un control derivado de la clase Form, que a su vez deriva de la clase Control. El marco de trabajo también permite heredar de formularios existentes para añadir más funcionalidades o modificar el comportamiento existente. Cuando se añade un formulario a un proyecto, se puede escoger si hereda de la clase Form proporcionada por el .NET Framework, o de un formulario creado con anterioridad.

2.2.5.8. Convenciones Para El Nombre De Los Controles

Es recomendable utilizar convenciones para el nombre de los controles, es decir, que al momento de dar lectura al código sea fácil de entender y comprender, por lo que el establecimiento de convenciones ayuda a identificar qué control se empleó, por ejemplo, si emplea un control botón emplee siempre como prefijo **btn** después complételo con un nombre descriptivo acorde a la funcionalidad que este tendrá.

Control	Prefijo
Button	Btn
Label	Lbl
PictureBox	Pic
Timer	Tmr
Text Box	Txt
List Box	Lst
Combo Box	Cbo
Check Box	Chk
Radio Button	Rad

Control Button (antes CommandButton)

El control **CommandButton** en Visual Basic .NET recibe el nombre de **Button**, la propiedad **Caption** ahora recibe el nombre de Text, ocurre lo mismo para el caso del control Label.

Control Label

El control **Label** ahora su propiedad **Caption** recibe el nombre de Text y por ejemplo la antes propiedad **Alignment** ahora recibe el nombre de TextAlign la cual tiene más opciones de alineación.

Control TextBox

El control **TextBox** tiene una capacidad de almacenamiento de 32 Kbytes de texto.

Control ListBox

El control **ListBox** hace visible una lista de items, donde el usuario puede seleccionar items en la lista utilizando los clics del mouse.

Control CheckBox

Un control **CheckBox** indica cuando un valor particular esta encendido o apagado, verdadero o falso, sí o no, puede ser utilizado también para seleccionar múltiples items de una lista de opciones.

Control RadioButton

Un control **RadioButton** permite al usuario seleccionar sólo un ítem de una lista de opciones.

Control DateTimePicker

El control **DateTimePicker** muestra por defecto la fecha actual que es posible modificar a través de su propiedad value.

Control LinkLabel

A través de este control es posible abrir el navegador por default y acceder a un URL específico:

```
LinkLabel1.LinkVisited = True
```

```
System.Diagnostics.Process.Start (LinkLabel1.Text)
```

Otros Controles

Control MainMenu

El Control MainMenu agrega menús a un programa, se agregar un menú simplemente seleccione el control MainMenu el cual de manera.

automática se ajusta al tamaño del formulario, este menú es de fácil manipulación.

Una vez agregado el control MainMenu también se agrega un panel debajo del formulario la cual recibe el nombre de Bandeja de Componentes a través del cual será posible definir y modificar.

Ahora bien lo que se muestra en la parte superior del formulario es una representación visual del menú misma que recibe el nombre de diseñador de menú, pero el objeto menú principal se muestra en la parte inferior dentro de la bandeja de componentes.

Si desea eliminar una opción del menú basta con seleccionarla y pulsar la tecla [**Supr**] o [Del].

Para agregar código a cada opción del menú basta con dar doble clic sobre la opción deseada.

Funciones Intrínsecas

- **Val**, regresa el número contenido en un argumento, se detiene en el primer carácter no numérico.
- **Cdbl**, regresa un valor double si el argumento puede ser convertido a double
- **IsNumeric**, regresa True si el argumento puede ser convertido a double
- **Rnd**, regresa un valor random entre 0 y 1
- **Abs**, regresa el valor absoluto de un número (es necesario importar System.Math)
- **Int**, regresa la parte entera de un número
- **FormatCurrency**, regresa un string formateado a moneda y redondeado a dos decimales
- **Formar (expr, str)**, convierte una expresión a un formato string específico.

Operadores

Visual Basic .NET dispone de los siguientes operadores matemáticos:

Operador	Descripción
+	Suma
-	Resta
*	Multiplicación
/	División
\	División entera (parte entera de la división)
Mod	Residuo (resto de la división entera)
^	Exponenciación (elevar a una potencia)
&	Concatenación de Cadenas

Operadores Aritméticos

Los operadores aritméticos requieren operadores numéricos (^, *, /, \, Mod, +, -) y producen resultados numéricos.

Operadores String

El operador de concatenación (&) String requiere operadores String y producen resultados String.

Operadores Comparativos

Los operadores Comparativos requieren operadores Comparativos (>, <, >=, <=, =, <>) y producen un resultado lógico (True or False).

Operadores Lógicos

Los operadores Lógicos requieren operadores Lógicos (NOT, AND, OR, XOR) y producen un resultado lógico.

Operadores Abreviados

Visual Basic .NET incorpora nuevos operadores abreviados que facilitan la escritura de código, por ejemplo:

```
x = x + 1 'Antes escribirá
```

```
x += 1 'Ahora puede escribir
```

Operadores Abreviados
+=
-=
*=
/=
\=
^
&=

Precedencia de Operadores

A continuación se muestra el orden de precedencia/prioridad/jerarquía de los operadores en Visual Basic .NET

Precedencia de Operadores
()
^
-
*/
\
Mod
+-

2.2.5.9. Programación Orientada a Objetos.

En la programación orientada a objetos, utilizamos la abstracción y la encapsulación para crear clases bien diseñadas.

2.2.5.9.1. Definición

Una clase es una plantilla o una estructura preliminar de un objeto. Esta estructura preliminar define atributos para almacenar datos y define operaciones para manipular esos datos. Una clase también define un conjunto de restricciones para permitir o denegar el acceso a sus atributos y operaciones.

Uso de la abstracción

Para crear una clase bien diseñada, utilizaremos la abstracción. Al implementar la abstracción, definiremos un concepto utilizando un mínimo conjunto de funcionalidades pensadas cuidadosamente que proporcione el comportamiento fundamental de la clase de un modo fácil de utilizar. Por desgracia, no es fácil crear buenas abstracciones de software. Normalmente, encontrar buenas abstracciones requiere un profundo conocimiento del problema que ha de resolver la clase y su contexto, una gran claridad de ideas y mucha experiencia.

Clases

Ejemplo de abstracción

El formulario Visual Basic .NET con el que hemos estado trabajando es un buen ejemplo de abstracción. Las propiedades esenciales de un formulario, como el título y color de fondo, se han abstraído en la clase Form. Algunas operaciones esenciales que se han abstraído son abrir, cerrar y minimizar.

Uso de la encapsulación

La abstracción se garantiza mediante la encapsulación. La encapsulación es el empaquetamiento de atributos y funcionalidades para crear un objeto que esencialmente es una "caja negra" (cuya estructura interna permanece privada). Empaquetamos los detalles de la abstracción y proporcionamos acceso sólo a los elementos que necesitan estar accesibles. Otros objetos pueden acceder a los servicios de un objeto encapsulado únicamente mediante mensajes que pasan a través de una interfaz claramente definida.

Ejemplo de encapsulación

Un ejemplo de encapsulación es un terminal de autoservicio (automatic teller machine, ATM). La interfaz de la ATM es simple para el cliente ATM, y el funcionamiento interno está oculto. Del mismo modo, una clase **BankAccount** encapsularía los métodos, campos y propiedades.

que describen una cuenta bancaria. Sin la encapsulación, deberíamos declarar procedimientos y variables distintos para almacenar y gestionar información de la cuenta bancaria, y sería difícil trabajar con más de una cuenta bancaria a la vez. La encapsulación permite a los usuarios utilizar los datos y procedimientos de la clase **BankAccount** como una unidad, sin conocer el código concreto encapsulado en la clase.

Objetos

Podemos utilizar el Examinador de objetos para examinar los elementos de programación de un componente (espacios de nombres, clases, módulos, estructuras, etc.) y los miembros de esos elementos (propiedades, métodos, eventos, variables, etc.). Los componentes que examinemos pueden ser proyectos de nuestra solución, componentes referenciados en esos proyectos o componentes externos.

Uso del Examinador de objetos

Para abrir el Examinador de objetos, pulse F2 o, en el menú Ver, seleccione **Examinador de objetos**.

El Examinador de objetos está formado por tres paneles:

- El panel Objetos muestra todos los objetos contenedores del ámbito de búsqueda en una vista en árbol. Cuando expandimos un elemento haciendo doble clic en él o haciendo clic en el signo más (+) junto a su nombre, aparecen los elementos definidos en él.
- El panel Miembros muestra los miembros de un elemento cuando seleccionamos el elemento en el panel Objetos.
- El panel Descripción muestra información detallada sobre el elemento o miembro actualmente seleccionado.

2.2.5.10. Características De Visual Studio 2008.

Visual Studio 2008 nos provee una serie de herramientas para desarrollo, así como características innovadoras para la creación de aplicaciones en una variedad de plataformas.

- Visual Studio 2008 incluye realces como un diseñador visual para desarrollo rápido con el .NET Framework 3.5, esto nos ayuda mucho a los que desarrollan en web por que se incluyen las características de Microsoft Expresión Web, que la verdad en mi punto de vista esta excelente, Visual Studio 2008 provee a desarrolladores con todas las herramientas y el framework el poder crear aplicaciones web con el soporte de AJAX.
- Visual Studio 2008 nos provee un nuevo lenguaje de consultas integrado para el manejo de la información, el cual se llama Microsoft Lenguaje Intégrate Query (LINQ), que es lo nos hará la vida más fácil para programadores individuales para poder construir soluciones que analicen y actúen sobre la información.
- Visual Studio 2008 también nos provee a desarrolladores la habilidad de poder escoger entre múltiples versiones del Framework con el mismo entorno de desarrollo, así nosotros podemos desarrollar en la versión que queramos ya sea en .NET Framework 2.0, 3.0 o 3.5, entiendo así que soporta un gran variedad de proyectos en la versión X en el mismo entorno de desarrollo.

2.2.5.11. Ventas Y Desventajas De Visual Basic 2008

2.2.5.11.1. Ventajas.

- Posee una curva de aprendizaje muy rápida.
- Integra el diseño e implementación de formularios de Windows.
- Permite usar con suma facilidad la plataforma de los sistemas Windows.
- Acostumbra a los desarrolladores a programar con eficiencia.

2.2.5.11.2. Desventajas.

Sólo permite el uso de funciones de librerías dinámicas (DLL). Algunas funciones están indocumentadas (Sin embargo esto ocurre en muchos lenguajes).

- Es un lenguaje basado en objetos y no orientado a objetos.
- No maneja muy bien los apuntadores de memoria.
- No soporta tratamiento de procesos como parte del lenguaje.
- No avisa de ciertos errores o advertencias (se puede configurar el compilador para generar ejecutables sin los controladores de desbordamiento de enteros o las comprobaciones de límites en matrices entre otros, dejando así más de la mano del programador la tarea de controlar dichos errores)

2.2.5.12. Requisitos Del Sistema.

- Sistema operativo compatible: Windows Server 2003; Windows Server 2008; Windows Vista; Windows XP
- Requisitos mínimos: CPU a 1,6 GHz, 384 MB de RAM, pantalla de 1024 x 768 y disco duro de 5400 rpm
- Requisitos recomendados: CPU a 2,2 GHz o superior, 1024 MB o más de RAM, pantalla de 1280 x 1024 y disco duro de 7200 rpm o superior
- En Windows Vista: CPU a 2,4 GHz, 768 MB de RAM.
- Mantenga la conexión a Internet durante la instalación del Service Pack hasta que aparezca un mensaje para indicar que la instalación se completó correctamente.

2.2.6. MICROSOFT SQL SERVER 2005.

2.2.6.1. Introducción.

Este artículo trata los desafíos que enfrentan las empresas que confían en tecnologías de integración de datos para proporcionar información con sentido y confiable a fin de mantener una ventaja competitiva en el mundo de los negocios actual. Describe la manera en la que SQL Server 2005 puede ayudar a los departamentos de tecnología de la información a cumplir con los requisitos de integración de datos de sus empresas. Se incluyen escenarios reales.

La siguiente versión de Microsoft® SQL Server™ se ha diseñado para ayudar a las empresas a enfrentarse a estos retos. Microsoft SQL Server 2005 es el software de última generación para el análisis y la administración de datos. Aporta un mayor grado de seguridad, escalabilidad y disponibilidad a los datos de la empresa y a las aplicaciones de análisis, al mismo tiempo que simplifica su creación, implementación y administración.

Basado en las características de solidez de SQL Server 2000, SQL Server 2005 constituye una solución de análisis y administración de datos integrados que ayudará a las organizaciones de cualquier tamaño a:

- Crear, implementar y administrar aplicaciones empresariales que resulten más seguras, escalables y confiables.
- Maximizar la productividad de la tecnología de la información al reducir la complejidad de los procesos de creación, implementación y administración de aplicaciones de bases de datos.
- Compartir datos en varias plataformas, aplicaciones y dispositivos para facilitar la conexión de sistemas internos y externos.

2.2.6.2. Herramientas de la Plataforma de Datos SQL Server.

SQL Server constituye una completa solución de datos de extremo a extremo que aporta a los usuarios de su organización una plataforma segura, confiable y productiva para las aplicaciones de datos de empresa e inteligencia empresarial (BI). SQL Server 2005 ofrece herramientas conocidas y de gran eficacia para los profesionales de TI, así como para aquellos que trabajan con la información. Estas herramientas reducen la complejidad que supone el proceso de crear, implementar, administrar y utilizar datos empresariales y aplicaciones analíticas en distintas plataformas que abarcan desde dispositivos móviles hasta sistemas de datos de empresas. Gracias a un extenso conjunto de características, interoperabilidad con los sistemas existentes

y automatización de las tareas rutinarias, SQL Server 2005 aporta una completa solución de datos para las empresas de todos los tamaños. En la figura 1 se muestra el diseño de una plataforma de datos SQL Server 2005.



Figura 1: Plataforma de datos SQL Server 2005.

La plataforma de datos SQL Server incluye las siguientes herramientas:

- **Relational Database (Base de datos relacional):** motor de base de datos relacional seguro, confiable, escalable y de alta disponibilidad con mejoras en el rendimiento y compatibilidad con datos (XML) estructurados y sin estructurar.
- **Replication Services (Servicios de duplicación):** duplicación de datos para aplicaciones de procesamiento de datos distribuidos o móviles, alta disponibilidad de los sistemas, concurrencia escalable con almacenes de datos secundarios para soluciones de creación de informes empresariales e integración con sistemas heterogéneos, incluidas las bases de datos Oracle existentes.
- **Notification Services (Servicios de notificación):** funciones avanzadas de notificación para el desarrollo e implementación de aplicaciones escalables que pueden enviar actualizaciones adecuadas y personalizadas de la información a una gran variedad de dispositivos conectados y móviles.

- **Integration Services (Servicios de integración):** funciones de extracción, transformación y carga para el almacenamiento de datos e integración de los datos en toda la empresa.
- **Analysis Services (Servicios de análisis):** funciones de procesamiento analítico en línea (OLAP) para el análisis rápido y sofisticado de conjuntos de datos complejos y de gran tamaño mediante el almacenamiento multidimensional.
- **Reporting Services (Servicios de creación de informes):** una completa solución para crear, administrar y entregar tanto los tradicionales informes en papel como los basados en Web interactivos.
- **Management Tools (Herramientas de administración):** SQL Server incluye herramientas de administración integradas para los procesos de ajuste y administración avanzados de bases de datos además de una estrecha integración con herramientas como Microsoft Operations Manager (MOM) y Microsoft Systems Management Server (SMS). Los protocolos de acceso a datos estándar reducen de forma considerable el tiempo que se tarda en integrar datos en SQL Server con sistemas existentes. Además, se ha integrado en SQL Server la compatibilidad con servicios Web para garantizar la interoperabilidad con las demás aplicaciones y plataformas.
- **Herramientas de desarrollo:** SQL Server ofrece herramientas de desarrollo integradas para el motor de base de datos, extracción de datos, transformación y carga (ETL), modelos de minería, OLAP y creación de informes que están totalmente integradas en Microsoft Visual Studio® para proporcionar funciones de desarrollo de aplicaciones de extremo a extremo. Cada subsistema principal de SQL Server se suministra con su propio modelo de objetos y conjunto de API para ampliar.

- sistema de datos en cualquier dirección que sea exclusiva de su empresa.

2.2.6.3. Administración de Datos Empresariales.

En el mundo conectado en el que vivimos hoy en día, los datos y los sistemas que administran dichos datos deben mantenerse seguros pero a la vez disponibles para los usuarios. Con SQL Server 2005, todos los usuarios y profesionales de TI de su organización se beneficiarán de la disminución del tiempo de inactividad de las aplicaciones, del aumento de la escalabilidad y rendimiento, así como de controles de seguridad exhaustivos a la vez que flexibles. SQL Server 2005 incluye también numerosas funciones nuevas y mejoradas para ayudar al personal de TI a ser más productivo. Asimismo, introduce mejoras esenciales para la administración de datos empresariales en las siguientes áreas:

- Facilidad de uso
- Disponibilidad
- Escalabilidad
- Seguridad

Facilidad de uso

Con SQL Server 2005, la implementación, administración y optimización de las aplicaciones analíticas y de datos empresariales resultan más simples y sencillas. Al ser una plataforma de administración de datos empresariales, proporciona una única consola de administración que permite que los administradores de datos de cualquier área de la organización puedan controlar, administrar y ajustar todas las bases de datos y servicios relacionados de la empresa. Ofrece una infraestructura de administración extensible que se puede programar fácilmente con SMO (SQL Management Objects), lo que permite a los usuarios personalizar y ampliar su entorno de administración y a los proveedores de software independientes (ISV) crear herramientas y funcionalidades adicionales para extender aún más las funciones ya incluidas.

Disponibilidad

Las inversiones en tecnologías de alta disponibilidad, funciones adicionales de copia de seguridad y restauración, así como mejoras en la duplicación permitirán a las empresas crear e implementar aplicaciones de alta disponibilidad. Las características innovadoras de alta disponibilidad como la creación de reflejos de bases de datos, clúster de conmutación por error, instantáneas de bases de datos y operaciones en línea mejoradas reducirán al mínimo el tiempo de inactividad y ayudarán a garantizar que los sistemas esenciales de las empresas permanecerán siempre accesibles. En esta sección repasaremos de forma detallada estas mejoras.

Creación de reflejos de bases de datos

Esta característica permite el flujo continuo del registro de transacciones desde un servidor de origen hasta un único servidor de destino. En el caso de un error del sistema principal, las aplicaciones pueden volver a conectarse de inmediato a la base de datos del servidor secundario. La segunda instancia detecta el error del servidor principal en segundos y acepta de inmediato las conexiones a la base de datos. La creación de reflejos de bases de datos funciona en el hardware de servidores estándar y no exige almacenamiento o controladores especiales.

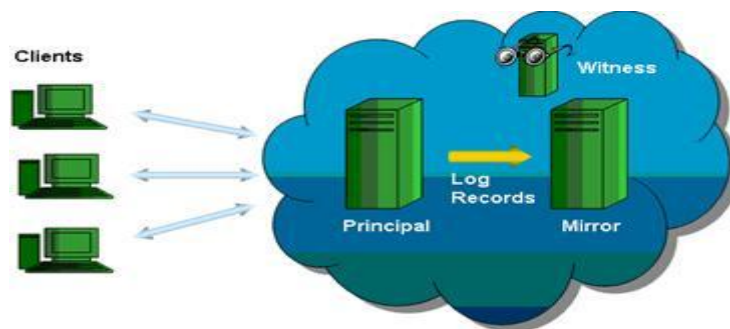


Figura 2: Configuración básica de la creación de un reflejo de una base de datos

Escalabilidad

Los avances en la escalabilidad como la creación de particiones de tablas, el aislamiento de instantáneas y la compatibilidad con 64 bits permitirá crear e implementar las aplicaciones más exigentes mediante SQL Server 2005. La creación de particiones en tablas e índices de gran tamaño mejora de forma considerable el rendimiento de las consultas en bases de datos muy extensas.

Seguridad

SQL Server 2005 realiza mejoras importantes en el modelo de seguridad de la plataforma de base de datos, con la intención de ofrecer un control más preciso y flexible que permita una seguridad mayor de los datos. Se ha realizado una considerable inversión en una serie de características a fin de proporcionar un alto nivel de seguridad para los datos de su empresa que incluyen:

- Aplicación de directivas para las contraseñas de inicio de sesión de SQL Server en el espacio de la autenticación.
- Incorporación de mayor granularidad en términos de especificación de permisos en varios ámbitos en el espacio de la autorización.
- Capacidad de separación de propietarios y esquemas en el espacio de la administración de seguridad.

2.2.6.4. Productividad del Desarrollador

SQL Server 2005 incluye un gran número de nuevas tecnologías que aportan un aumento considerable en la productividad del desarrollador. Desde compatibilidad con .NET Framework hasta la estrecha integración con Visual Studio®, estas características ofrecen a los desarrolladores la capacidad de crear de forma más sencilla aplicaciones de bases de datos sólidas y seguras a un bajo costo. SQL Server 2005 permite a los desarrolladores aprovechar sus habilidades existentes en una variedad de lenguajes de desarrollo a la vez que presenta un entorno de

desarrollo de extremo a extremo para la base de datos. Las capacidades nativas de XML también permitirán a los desarrolladores crear nuevas clases de aplicaciones conectadas en cualquier plataforma o dispositivo.

Las mejoras para la productividad de los desarrolladores incluyen:

- Compatibilidad ampliada con lenguajes
- Herramientas de desarrollo mejoradas
- Extensibilidad
- Acceso a datos mejorado
- Servicios XML y Web
- Marco de aplicación

Compatibilidad ampliada con lenguajes

Se aloja en el motor de la base de datos, los desarrolladores pueden elegir entre una variedad de lenguajes conocidos para desarrollar aplicaciones de base de datos, incluidos Transact-SQL, Microsoft Visual Basic® .NET y Microsoft Visual C#® .NET. Además, el alojamiento de CLR ofrecerá a los desarrolladores una mayor flexibilidad a través del uso de funciones y tipos definidos por el usuario. CLR también brinda oportunidades para utilizar código de terceros en el desarrollo rápido de aplicaciones de bases de datos.

Herramientas de desarrollo mejoradas

Los desarrolladores podrán utilizar una herramienta de desarrollo para Transact-SQL, XML, Multidimensional Expressions (MDX) y XML para Analysis (XML/A). La integración con el entorno de desarrollo de Visual Studio ofrecerá un desarrollo y depuración más eficientes de las aplicaciones de línea empresarial e inteligencia empresarial (BI).

Extensibilidad

Los tipos definidos por el usuario en SQL Server 2005 no constituyen un mecanismo de extensibilidad relacional. Son un modo de ampliar el sistema de tipos escalar de la base de datos. El sistema de tipos escalar incluye los tipos de columna que se suministran en SQL Server (tipos

como **int**, **nvarchar**, **uniqueidentifier**, etc.). Con los tipos definidos por el usuario, se puede establecer un tipo propio que se utilice para las definiciones de columnas, por ejemplo. Cree un tipo definido por el usuario si su tipo es realmente un valor atómico que se puede modelar como una columna.

Acceso a datos y servicios Web mejorados

En SQL Server 2005, puede desarrollar servicios Web XML en el nivel de la base de datos, convirtiendo SQL Server en un detector HTTP. De este modo, se proporciona un nuevo tipo de capacidad de acceso a datos para aplicaciones centralizada en los servicios Web. En SQL Server 2005 puede utilizar HTTP para tener acceso directamente a SQL Server, sin utilizar un detector de nivel medio como Servicios de Internet Information Server (IIS) de Microsoft. SQL Server muestra una interfaz de servicio Web para permitir la ejecución de instrucciones SQL y la invocación de funciones y procedimientos. Los resultados de las consultas se devuelven en un formato XML y puede aprovechar la infraestructura de los servicios Web de Visual Studio.

ADO.NET 2.0/ADOMD.NET

Hay muchas más novedades en la siguiente versión de ADO.NET. Desde nueva compatibilidad para notificaciones de cambio de consultas hasta conjuntos MARS (Multiple Active Result Sets), ADO.NET evoluciona en el acceso a los conjuntos de datos y la manipulación para conseguir una mayor escalabilidad y flexibilidad.

Notificación de consultas

SQL Server 2005 introduce un soporte de notificaciones para las consultas de SQL Server. Puede utilizar este soporte para enviar un comando a SQL Server y solicitar que se genere una notificación si se vuelve a ejecutar el mismo comando y se producen resultados distintos a los obtenidos inicialmente. Esto se consigue mediante un objeto de dependencia que detecta cuando se modifican los datos subyacentes.

Los comandos que se envían al servidor a través de cualquier API de cliente como ADO.NET, OLE DB, ODBC (Open Database Connectivity), Microsoft ActiveX® Data Objects (ADO) o SOAP pueden incluir una etiqueta que requiera una notificación. En cada instrucción que se ejecuta como parte de la solicitud, el servidor crea una suscripción de notificación que se activa una vez por cada instrucción que se incluye en la solicitud. Las notificaciones se entregan a través de la cola de SQL Service Broker que las aplicaciones pueden sondear y emplean los servicios de activación o las instrucciones de bloqueo que se devuelven cada vez que hay disponibles notificaciones. Las notificaciones de consulta son útiles para permitir el almacenamiento en caché de los resultados de las aplicaciones como los sitios Web destinados a bases de datos. En la figura 4 se muestra el proceso de notificación de consultas.

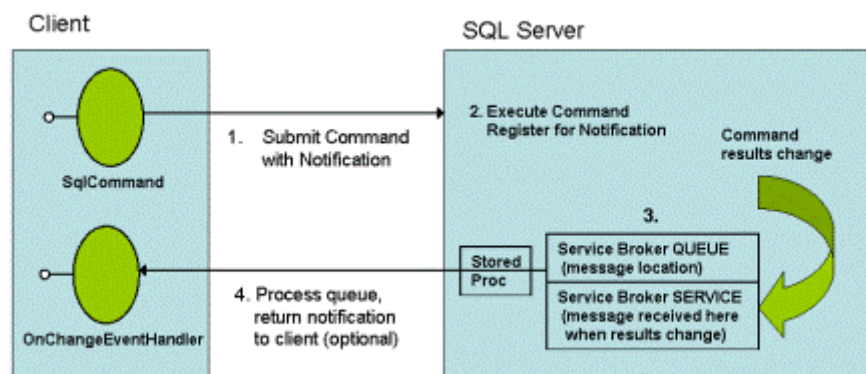


Figura 3: Notificación De Consultas.

Compatibilidad con XML

Avances como el tipo de datos nativos XML y XQuery ayudan a las organizaciones a conectar sin problemas sistemas externos e internos. SQL Server 2005 admitirá tanto los datos XML como los relacionales de forma nativa, por lo que las empresas pueden almacenar, administrar y analizar los datos con el formato que mejor se adapte a sus

necesidades. La compatibilidad con los estándares abiertos existentes y nuevos como el protocolo de transferencia de hipertexto (HTTP), XML, SOAP (Simple Object Access Protocol), XQuery y lenguajes de definición de esquemas XML (XSD) también facilitarán la comunicación con los sistemas empresariales ampliados.

Tipo de datos XML

XML puede modelar datos complejos; no se limita a los tipos escalares que admite SQL Server. De este modo, un tipo de datos integrado basado en cadenas como char o varchar no es suficiente para utilizar de forma total y efectiva la eficacia y las numerosas ventajas de XML. Por ejemplo, si XML se almacena como cadena, puede insertar o seleccionar un documento entero, o incluso recuperar bytes contiguos, pero no se puede consultar el contenido del propio documento. Al incluir el tipo de datos XML, SQL Server 2005 permite consultar partes de un documento XML, comprobar que el documento cumple con el esquema XML e incluso modificar el contenido del documento XML en su sitio. También integra datos relacionales tradicionales con datos de documentos XML semiestructurados o sin estructurar en modos que no son posibles con SQL Server 2000. En SQL Server 2005, los datos XML se almacenan como grandes objetos binarios (BLOB) en una representación interna que permite volver a realizar un eficiente análisis y realizar alguna compresión.

Una colección de esquemas XML se puede asociar con una columna de tipo XML. De este modo, se cuenta con la validación para las restricciones, inserciones y actualizaciones y la escritura de valores dentro de los datos XML almacenados, así como con optimizaciones para el almacenamiento y el procesamiento de consultas. SQL Server 2005 también ofrece varias instancias de DDL para administrar los esquemas en el servidor.

XQuery

El lenguaje de consultas XML, o XQuery, es un lenguaje inteligente y seguro que se optimiza para consultar todos los tipos de datos XML. Con XQuery se pueden ejecutar consultas con variables y columnas de tipos de datos XML a través de los últimos métodos relacionados. Al igual que con gran cantidad de estándares XML, World Wide Web Consortium (W3C) supervisa el desarrollo de XQuery. XQuery evolucionó de un lenguaje de consultas denominado Quilt, que a su vez estaba basado en una variedad de otros lenguajes de consultas como XML Path Language (XPath) versión 1.0, XQL y SQL. También contiene XPath 2.0 como subconjunto. Por tanto, si tiene experiencia en el uso de XPath 1.0, puede aprovechar sus habilidades y no tener que aprender un lenguaje de consulta totalmente nuevo. Hay, sin embargo, mejoras significativas que van más allá de XPath 1.0, como la escritura, funciones especiales y capacidad para una mejor iteración, clasificación de resultados y construcción.

Compatibilidad con servicios Web

En SQL Server 2005, puede desarrollar servicios Web XML en el nivel de la base de datos, convirtiendo SQL Server en un detector HTTP. De este modo, se proporciona un nuevo tipo de capacidad de acceso a datos para aplicaciones centralizadas en los servicios Web. En SQL Server 2005 puede utilizar HTTP para tener acceso directamente a SQL Server, sin utilizar un detector de nivel medio como Servicios de Internet Information Server (IIS) de Microsoft. SQL Server muestra una interfaz de servicio Web que permite la ejecución de instrucciones SQL y la invocación de funciones y procedimientos. Los resultados de las consultas se devuelven en un formato XML y se puede sacar provecho de la infraestructura de los servicios Web de Visual Studio.

XML para Analysis Services (XML/A)

XML para Analysis Services (XML/A) es el protocolo nativo basado en estándares para la comunicación con el servidor de Analysis Services. Se habilitan nuevos tipos de aplicaciones fáciles de desarrollar: aplicaciones que integran la analítica con operaciones en tiempo real. Con XML/A como protocolo nativo, los clientes de Analysis Services se pueden configurar para que tengan una superficie cero y cada servidor sea un servicio Web automáticamente. Hay disponible una capa Win32 de superficie ligera para la compatibilidad con versiones anteriores de herramientas que funcionan con Analysis Services 2000 en OLE DB para OLAP, ADOMD y ADOMD.NET. Muchos usuarios continuarán utilizando el modelo de objetos de ADOMD.NET para crear aplicaciones personalizadas en Analysis Services.

Marco de aplicación

SQL Server 2005 introduce un nuevo marco de aplicación de SQL Server que incluye: Service Broker, Notification Services, SQL Server Mobile y SQL Server Express. Service Broker es un marco de aplicación distribuida que presenta un sistema de mensajería asincrónica segura en la base de datos y a nivel de bases de datos.

2.2.6.5. Actualización a SQL Server 2005

A continuación, se muestran algunas sugerencias para actualizar a SQL Server 2005

- Actualizar a SQL Server 2005 desde SQL Server 7.0 o SQL Server 2000.
- Ejecutar el asesor de actualizaciones antes de actualizar para determinar si se esperan cambios de algún producto que afecten a aplicaciones existentes.
- En la instalación se pueden actualizar a Database Engine, Analysis Services y Reporting Services.

- Integration Services de SQL Server, el sustituto de DTS, se instala totalmente con DTS. Puede ejecutar paquetes DTS mediante los componentes en tiempo de dimensión de DTS.
- Notification Services de SQL Server 2005 se instala de lado a lado con Notification Services 2.0. Debe migrar instancias de Notification Services a SQL Server 2005 al actualizar el motor de base de datos.

2.2.7 MICROSOFT WINDOWS SERVER 2008.

2.2.7.1 Introducción.

Microsoft Windows Server 2008 está diseñado para ofrecer a las organizaciones la plataforma más productiva para virtualización de cargas de trabajo, creación de aplicaciones eficaces y protección de redes. Ofrece una plataforma segura y de fácil administración, para el desarrollo y alojamiento confiable de aplicaciones y servicios web. Del grupo de trabajo al centro de datos, Windows Server 2008 incluye nuevas funciones de gran valor y eficacia y mejoras impactantes en el sistema operativo base.

2.2.7.2 Características más frescas en Windows Server 2008

Usted no encontrará ninguna característica asesino en esta versión. Pero eso no quiere decir que no hay nada para emocionarse. Hay una gran cantidad que hay de nuevo, y dependiendo de la configuración de su organización, es casi seguro que encontrará algunas o todas las que sea muy valioso

Cualquier clasificación está destinada a ser subjetiva, y teniendo esto en mente, he aquí lo que creo que son las 10 nuevas características más interesantes de Windows Server 2008.

2.2.7.2.1 Virtualización

A pesar de que no estarán disponibles con el lanzamiento inicial de Server 2008, Microsoft Hyper-V basada en hipervisor de virtualización de la tecnología que promete ser una atracción estrella de Server 2008 para muchas organizaciones.

Aunque un 75 por ciento de las grandes empresas han empezado a utilizar la virtualización, sólo un 10 por ciento de los servidores se están ejecutando a cabo las máquinas virtuales . Esto significa que el mercado está todavía inmadura. Para los talleres de Windows Server 2008 utilizando la virtualización será una manera relativamente de bajo costo y bajo riesgo

2.2.7.2.2 Server Core

Muchos administradores de servidores, especialmente aquellos acostumbrados a trabajar en un entorno Linux, instintivamente disgusta tener que instalar un grande, repleta de funciones del sistema operativo para ejecutar un servidor especializado en particular. Server 2008 ofrece una instalación Server Core, que proporciona la instalación mínima necesaria para llevar a cabo una función de servidor específico, como un DHCP , DNS o servidor de impresión . Desde un punto de vista de la seguridad, esto es atractivo.

2.2.7.2.3 IIS

IIS 7, el servidor Web incluido con Server 2008, es una gran mejora respecto a la versión anterior. "Hay cambios significativos en términos de seguridad y la aplicación general que hacen de esta versión muy atractivo", dijo Barb Goldworm, presidente y analista jefe de Boulder, Colorado-basado Consulting Focus.

2.2.7.2.4 Basada en roles instalación

basada en roles instalación es una versión menos extrema de Server Core. A pesar de que se incluyó en el 2003, es mucho más completo en esta versión. El concepto es que en lugar de configurar una instalación completa del servidor para una función determinada por la desinstalación de los componentes innecesarios (y la instalación necesaria extras), sólo tiene que especificar la función del servidor es jugar, y Windows instalará lo necesario - no hay nada más. Esto hace que sea fácil para que cualquiera pueda aprovisionar un servidor en particular, sin aumentar la superficie de ataque mediante la inclusión de componentes no deseados que no harán otra cosa que presente un riesgo de seguridad.

2.2.7.2.5 Sólo lectura controladores de dominio (RODC)

Esto los convierte en una plataforma de lanzamiento ideal para los ataques a los servidores corporativos principales. RODC proporciona una manera de hacer que una base de datos de Active Directory de sólo lectura. Por lo tanto, cualquier daño realizado a la sucursal no pueden propagar su camino de regreso a envenenar el sistema de Active Directory como un todo. También reduce el tráfico en la WAN enlaces.

2.2.7.2.6 Mejora de los servicios de terminal

Servicios de Terminal Server se ha reforzado en Server 2008 en un número de maneras. TS RemoteApp permite a los usuarios remotos acceder a una aplicación centralizada (en lugar de un escritorio completo) que parece estar en ejecución en el disco duro del equipo local

2.2.7.2.7 Network Access Protection

Sistema de Microsoft para asegurar que los clientes se conectan a Server 2008 están parcheadas, ejecutando un servidor de seguridad

y en cumplimiento de las políticas corporativas de seguridad - y que los que no lo son pueden ser remediados - es útil. Sin embargo, la funcionalidad similar ha sido y sigue siendo disponible a terceros.

2.2.7.2.8 Bitlocker

Sistema de cifrado de disco puede ser una medida de seguridad razonable para los servidores ubicados en oficinas remotas o en cualquier lugar donde la seguridad física del servidor es sub-óptima. Cifrado BitLocker protege los datos si el servidor se elimina físicamente o arrancar desde medios extraíbles en un sistema operativo diferente que de otro modo podrían dar un intruso obtenga acceso a los datos que están protegidos en un entorno Windows.

2.2.7.2.9 Windows PowerShell

Nuevo de Microsoft (más o menos) de línea de comandos y lenguaje de scripting shell ha demostrado ser popular entre algunos administradores de servidores, especialmente aquellos acostumbrados a trabajar en entornos Linux. Incluido en Server 2008, PowerShell puede hacer que algunos puestos de trabajo más rápido y más fácil de realizar que va a través de la interfaz gráfica de usuario.

2.2.7.2.10 Mayor seguridad

Ya hemos mencionado varias características de seguridad incorporadas en Server 2008, como la capacidad de reducir la superficie de ataque mediante la ejecución de instalaciones mínimas y características específicas como BitLocker y PAN. Muchos otros pequeños detalles hacen Server 2008 más segura que sus predecesoras

CAPÍTULO III:
MARCO METODOLÓGICO.

3. MARCO METODOLÓGICO.

3.1. TIPO DE INVESTIGACIÓN.

En esta investigación se realizará un estudio experimental y correlacional, ya que se ha de manipular la variable independiente (causa) y se verán los resultados en la variable dependiente. Además es un diseño tecnológico ya que utiliza muchas herramientas conjuntas en el campo de la Ingeniería del Software conjuntamente con la Ingeniería de Sistemas.

Esto específicamente a lo que se ha desarrollado en la investigación y a lo que vamos a demostrar luego con el planteamiento de la hipótesis.

Experimental, porque esta investigación va más allá de simples conceptos y metodologías a utilizar, está dirigida a ver qué problemas existen, es decir este estudio se centra en explicar por qué ocurre la demora en la atención a los clientes y como se mejorará esta demanda.

Correlacional, ya que el estudio se centra en manipular la variable independiente que es Desarrollo del Sistema Gestión De Ventas E Inventarios De La Despensa "Santa Fe" para afectar la variable dependiente es decir mejorar la atención a los clientes.

Investigación De Campo:

Emplea básicamente la información obtenida a través de las técnicas de la observación, entrevista y cuestionario.

Las técnicas de investigación de campo utilizan sus propios procedimientos e instrumentos para la recolección de datos, junto a los mecanismos específicos de control y valides de la información que se obtenga en la Despensa "Santa Fe".

Se utilizarán en esta investigación los siguientes métodos:

Método Deductivo: Porque analizaremos primero los contenidos teóricos para luego desglosarlos y pasarlos a la práctica para finalmente llegar a conclusiones y recomendaciones adecuadas para diseñar nuestro sistema.

Método Inductivo: A partir del análisis de un caso o de casos particulares y observaciones de la realidad se extraen conclusiones de carácter general. Comienza con la recolección de datos, se categoriza las variables observadas, se prueban las hipótesis, se puede realizar generalizaciones para elaborar una teoría sobre la despensa.

Método Bibliográfico: Realizaremos una Investigación Bibliográfica sobre lo que conlleva el Sistema a desarrollarse como son: Bases de Datos, Lenguajes de Programación, Plataformas de Soporte y manejo de Sistemas Operativos. Se agrupará todas las referencias de libros utilizados en la presente tesis organizando de alguna manera ya sea alfabética o sistemáticamente.

3.2. HIPÓTESIS.

3.2.1. Hipótesis General.

- Con la implementación del Sistema de Venta e Inventario en la Despensa "Santa Fe" Del Cantón Caluma Provincia Bolívar se mejorará el control y se elevará la satisfacción del cliente.

3.2.2. Hipótesis Específica.

- La utilización de un Sistema informático de control de Ventas E Inventario, permitirá tener un mejor control de la mercadería en la Despensa Santa Fe.
- La implementación de un Sistema Informatico de control E Inventario, elevara la satisfacción de los clientes.
- Con el nuevo sistema de control de Ventas E Inventario en La Despensa "Santa Fe", permitirá contar con la información precisa y satisfactoria para la correcta manipulación de datos existentes en la Despensa serán más rápidos y exactos.

3.3. VARIABLES.

3.3.1. Variable Independiente.

Sistema informático desarrollado en Microsoft Visual Studio 2008 y SQL Server 2005.

3.3.2. Variable Dependiente.

Optimizar el manejo de recursos de información de la Despensa "Santa Fe" Del Cantón Caluma Provincia Bolívar.

3.3.3. Operacionalización Conceptual

VARIABLES	DEFINICIÓN
Variable Independiente Sistema Informático	Un sistema informático se basa en un conjunto de hardware, software y de un soporte humano con el fin de controlar, procesar y optimizar datos obteniendo resultados confiables, rápidos, y exactos.
Variable Dependiente Control de los procesos	Actividad que permite detectar errores en los procesos para satisfacer las necesidades establecidas.

Tabla II.1. Operacionalización Conceptual de las variables

3.4. POBLACIÓN Y MUESTRA.

3.4.1. Población.

La Despensa "Santa Fe", está localizado en el Cantón Caluma Provincia De Bolívar, barrio Sanfrancisco cuenta con una matriz principal.

En el presente trabajo de investigación, la población la componen los Clientes, Empleados, Administradora de la Despensa "Santa Fe",

aquellos que están involucrados en este proceso de control y ventas de los productos.

Así tenemos:

PROYECTO	Nº
Clientes	500
Empleados	2
Administradora	1
TOTAL	503

Tabla II.2: Población De La Despensa.

3.4.2. Muestra.

Para la recolección de la información se toma una muestra representativa de la población de clientes que realicen sus compras en el local. Mediante el sistema de muestreo aleatorio simple, utilizando la siguiente formula:

3.4.2.1. Cálculo del tamaño de la muestra cuando la población es finita

Si se conoce el tamaño de la población, es decir, la población es finita como en este caso de nuestra investigación y deseamos saber cuántos del total tendremos que estudiar, la fórmula es:

$$n = \frac{N * z^2 * p * q}{E^2 * (N - 1) + z^2 * p * q}$$

Donde:

- Z Es el valor de la variable aleatoria estandarizada para $\alpha/2$, correspondiente a un nivel de confianza o seguridad $(1 - \alpha)$.
- E Es la precisión que deseamos para nuestro estudio o Error máximo de estimación.
- p Es una idea del valor aproximado de la proporción poblacional. Esta idea se puede obtener revisando la literatura, por estudio

pilotos previos. En caso de no tener dicha información utilizaremos el valor $p = 0.5$ (50%)

q Es el valor del complemento de p , se calcula con: $q = 1 - p$.

N Tamaño de población finita

¿A cuántas personas tenemos que estudiar de una población de 400 clientes para conocer la atención a los mismos?

$$n = \frac{N z^2 p q}{E^2(N - 1) + z^2 p q} = \frac{(503)(1.96)^2(.05)(0.95)}{(.03)^2(503 - 1) + (1.96)^2(.05)(0.95)} = 145.05$$

Utilización de la fórmula para el calcula de la muestra

La muestra deberá ser de 145 clientes.

Esta fórmula se utilizará para calcular el tamaño de la muestra para nuestras distintas poblaciones, en las que deberemos encuestar a estas personas. En la siguiente tabla se muestra el cálculo de la muestra para cada una de las poblaciones existentes en el sector donde se encuentra la despensa. Tomando en cuenta la fórmula:

$$n = \frac{N z^2 p q}{E^2(N - 1) + z^2 p q}$$

Personal	N	Z	P	Q	e	N	Personas a encuestar
Clientes	500	1,96	0,05	0,95	0,04	93,01685432	93
Empleados	2	1,96	0,05	0,95	0,04	0,995634998	1
Administradora	1	1,96	0,05	0,95	0,04	100	1

Tabla II.3: Utilización de la Fórmula para el cálculo de la muestra

Desarrollo de la formula tomando encuesta los valores de la tabla 1

Personal	Numerador			Denominador					n
	Nz ²	p.q	respuesta	E ²	(N-1)	z ²	p.q	respuesta	
Cientes	1920,8	0,0475	91,238	0,0016	499	3,8416	0,0475	0,980876	93,01685432
Empleados	7,6832	0,0475	0,364952	0,0016	1	3,8416	0,0475	0,366552	0,995634998
Administradora	3,8416	0,0475	0,182476	0,0016	0	3,8416	0,0475	0,182476	100

Tabla II.4: Cálculo del tamaño de la muestra

3.4.3. Instrumento De Recolección De Datos.

Los instrumentos a utilizar serán: Encuesta, entrevista y observación, la encuesta estará basada en el cuestionario preestablecido; para la aplicación de la encuesta se realizara un adiestrado a los encuestadores. La entrevista se la realizará al personal que trabaja en la despensa, como son: el administrador y los empleados.

La observación se la realizará en la despensa para tener información que todavía con las técnicas anteriores no se pudo reunir.

3.4.3.1. Aplicación Del Instrumento De Recolección.

Se aplicará el método inductivo deductivo para resolver un problema en particular y extenderlo a lo general.

Para la recolección de la información se tuvo que realizar una encuesta a todas las personas que tengan que ver con el proceso de realización de las ventas y gestión de la mercadería, donde nos dirán que está fallando o si todo marcha bien en la despensa.

3.4.3.2. Validación De Los Instrumentos.

Para que la información obtenida haya sido satisfactoria en cuanto a la realidad, se tuvo que acudir a una persona con conocimientos en el tema para poder aplicar los instrumentos en este proceso, es decir,

tuvimos la ayuda de una persona con conocimientos en estadísticas para poder guiarnos y medir las variables.

La utilización masiva de las encuestas en procesos de toma de decisiones como lo es el caso de la investigación, se tuvo que realizar, ya que se necesitaba de datos cuantitativos que ayuden a resolver este problema.

Por esta razón se realizó una encuesta en la que cada pregunta tiene sus posibles opciones a escoger para poder dar un resultado efectivo a la variable en estudio.

3.4.3.3. Clasificación De La Información Y Tabulación.

La información obtenida se registró en gráficos, donde se colocaron todos los datos necesarios correspondientes a los clientes y empleados de la despensa "Santa Fe", los resultados obtenidos en la encuesta están presentados a continuación:

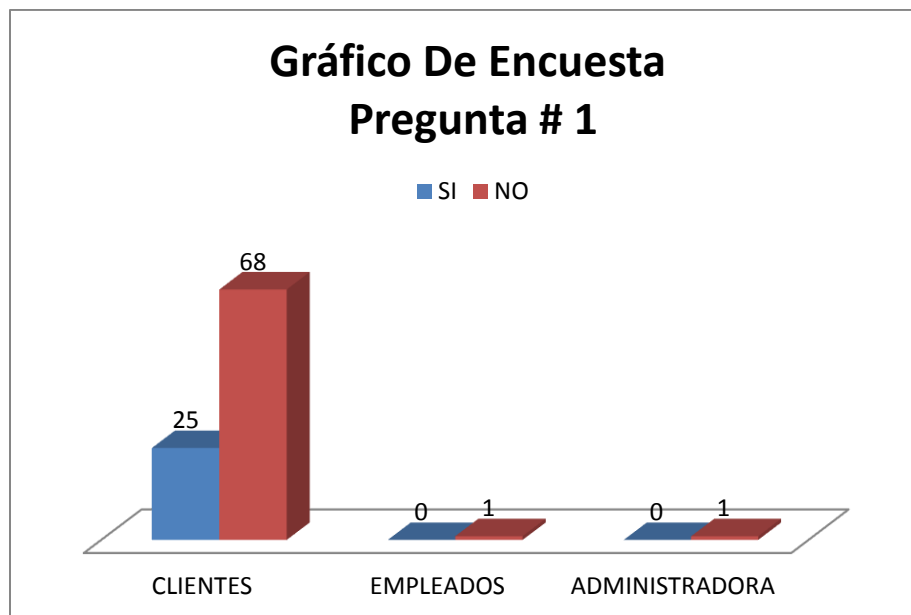
3.4.3.4. Gráficos Estadísticos.

Utilizando el Sistema Manual:

1.-) ¿Usted se siente satisfecho con la atención que se brinda al cliente en la Despensa Santa Fe del cantón Caluma Provincia Bolívar?

Calificación	Clientes	Empleados	Administradora
SI	25	0	0
NO	68	1	1
TOTALES	93	1	1

Tabla II.5: De Encuestados De La Pregunta # 1



Fuente: Encuesta a Clientes De La Despensa “Santa Fe”.

Calificación	Cientes	%	Empleados	%	Administradora	%	% General
SI	25	26,88172043	0	0	0	0	8,960573477
NO	68	73,11827957	1	100	1	100	91,03942652
TOTALES	93	100	1	100	1	100	100

Tabla II.6: Tabulación de Datos

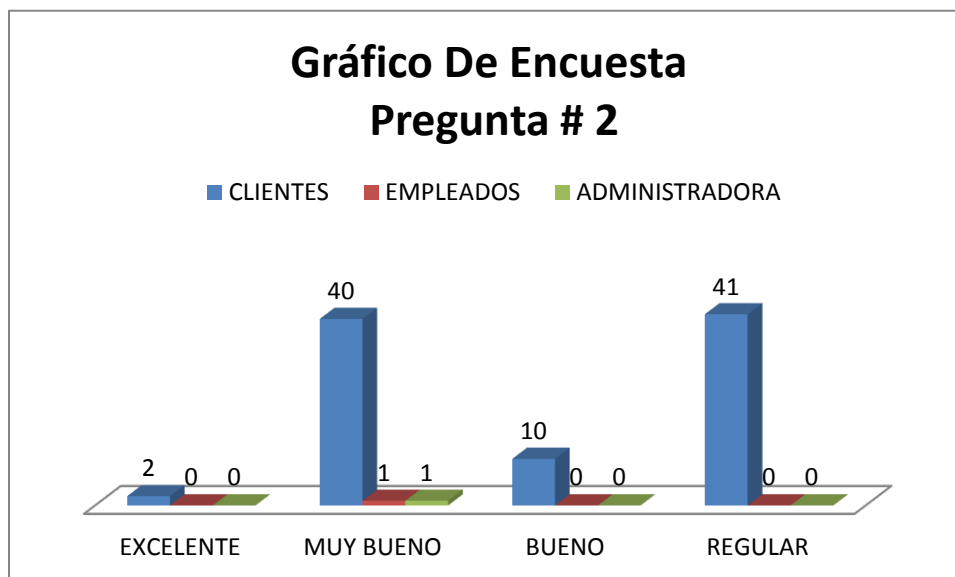
Interpretación:

Por lo tanto el 91% de los encuestados dijeron que no se encuentran satisfechos con la atención que se les brinda a los clientes.

2.-) ¿Cómo califica usted la calidad de los productos que ofrece la Despensa Santa Fe?

Calificación	Cientes	Empleados	Administradora
EXCELENTE	2	0	0
MUY BUENO	40	1	1
BUENO	10	0	0
REGULAR	41	0	0
TOTALES	93	1	1

Tabla II.7: De Encuestados De La Pregunta # 2



Fuente: Encuesta a Clientes De La Despensa “Santa Fe”.

Calificación	Clientes	%	Empleados	%	Administradora	%	% General
EXCELENTE	2	2,15053763	0	0	0	0	0,71684588
MUY BUENO	40	43,0107527	1	100	1	100	81,0035842
BUENO	10	10,7526882	0	0	0	0	3,58422939
REGULAR	41	44,0860215	0	0	0	0	14,6953405
TOTALES	93	100	1	100	1	100	100

Tabla II.8: Tabulación de Datos

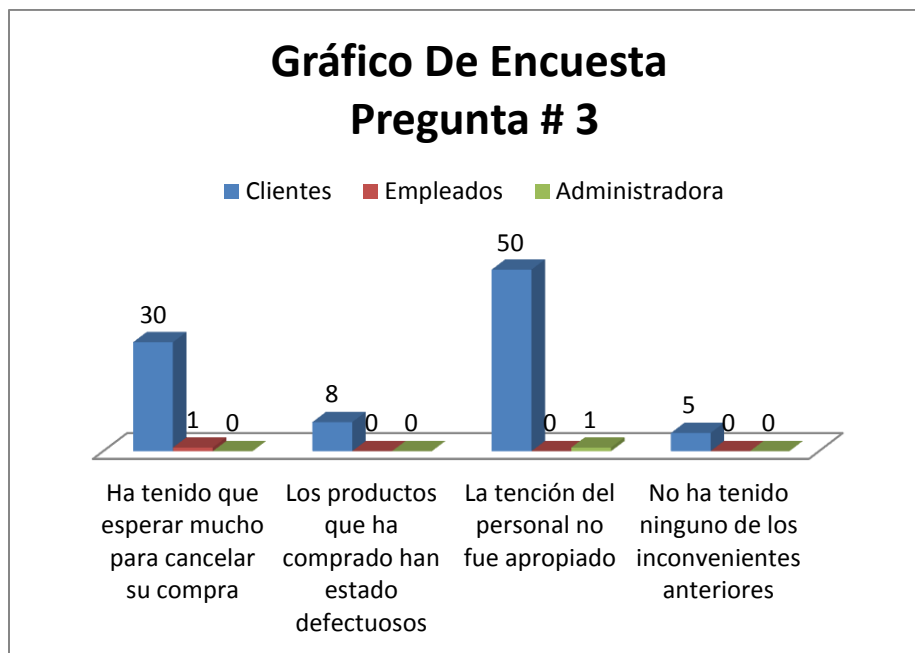
Interpretación:

Por lo tanto el 81% de los encuestados contestaron que los productos son Muy Bueno en la Despensa “Santa Fe”.

3.) ¿Cuándo ha realizado sus compras en la Despensa Santa Fe le ha sucedido alguno de estos casos?

Calificación	Clientes	Empleados	Administradora
HA TENIDO QUE ESPERAR MUCHO PARA CANCELAR SU COMPRA	30	1	0
LOS PRODUCTOS QUE HA COMPRADO HAN ESTADO DEFECTUOSOS	8	0	0
LA TENCIÓN DEL PERSONAL NO FUE APROPIADO	50	0	1
NO HA TENIDO NINGUNO DE LOS INCONVENIENTES ANTERIORES	5	0	0
TOTALES	93	1	1

Tabla II.9: De Encuestados De La Pregunta # 3



Fuente: Encuesta a Clientes De La Despensa “Santa Fe”.

Calificación	Clientes	%	Empleados	%	Administradora	%	% General
Ha tenido que esperar mucho para cancelar su compra	30	32,2580645	1	100	0	0	44,0860215
Los productos que ha comprado han estado defectuosos	8	8,60215054	0	0	0	0	2,86738351
La tención del personal no fue apropiado	50	53,7634409	0	0	1	100	51,2544803
No ha tenido ninguno de los inconvenientes anteriores	5	5,37634409	0	0	0	0	1,7921147
Totales	93	100	1	100	1	100	100

Tabla II.10: Tabulación de Datos

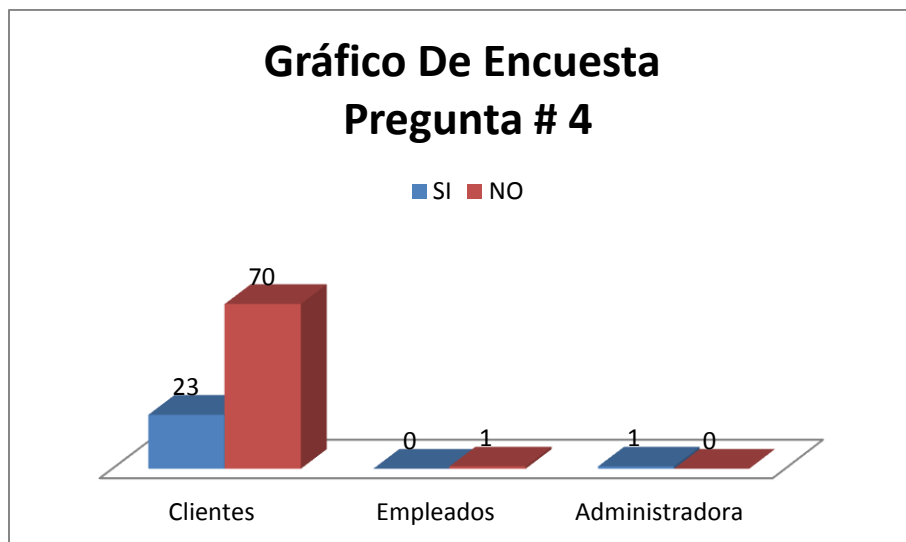
Interpretación:

Por lo tanto el 51% de los encuestados cuando ha realizado sus compras en la Despensa “Santa Fe” la atención no asido la apropiada.

4.-) ¿Al momento de escoger los productos que se encuentran en las perchas, ¿Encuentra información necesaria de los productos con sus respectivos precios y fecha de caducidad?

Calificación	Clientes	Empleados	Administradora
SI	23	0	1
NO	70	1	0
TOTALES	93	1	1

Tabla II.11: De Encuestados De La Pregunta # 4



Fuente: Encuesta a Clientes De La Despensa “Santa Fe”.

Calificación	Clientes	%	Empleados	%	Administradora	%	% General
SI	23	24,7311828	0	0	1	100	41,57706093
NO	70	75,2688172	1	100	0	0	58,42293907
TOTALES	93	100	1	100	1	100	100

Tabla II.12: Tabulación de Datos

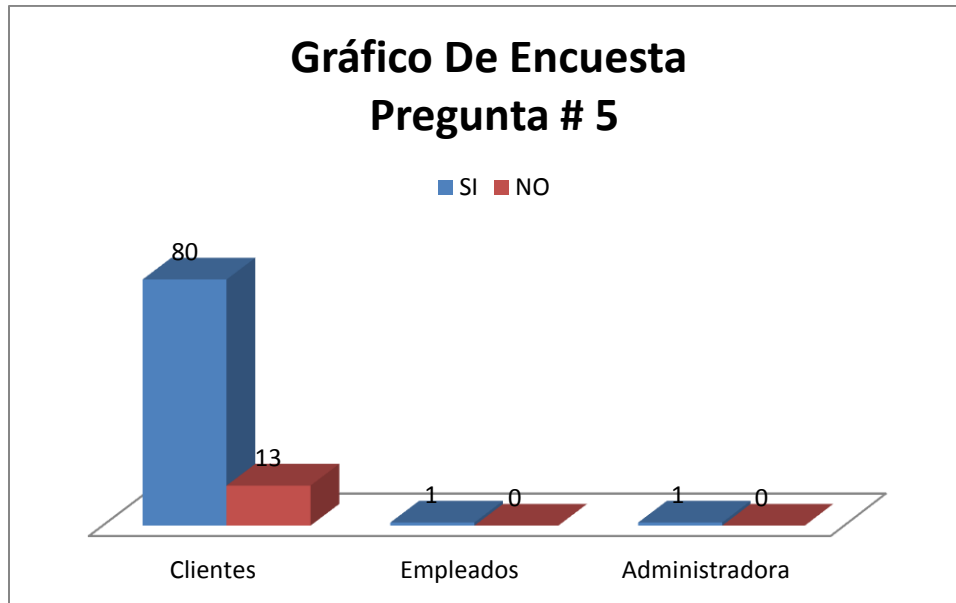
Interpretación:

Por lo tanto el 58% de los encuestados no encontraban el precio respectivo del producto en las perchas, de la Despensa “Santa Fe”.

5.-) ¿Piensa usted que se debería implantar un sistema de información en la Despensa Santa Fe para agilizar el proceso de venta de productos?

Calificación	Clientes	Empleados	Administradora
SI	80	1	1
NO	13	0	0
TOTALES	93	1	1

Tabla II.13: De Encuestados De La Pregunta # 5



Fuente: Encuesta a Clientes De La Despensa “Santa Fe”.

Calificación	Cientes	%	Empleados	%	Administradora	%	% General
SI	80	86,0215054	1	100	1	100	95,3405018
NO	13	13,9784946	0	0	0	0	4,65949821
TOTALES	93	100	1	100	1	100	100

Tabla II.14: Tabulación de Datos

Interpretación:

Por lo tanto el 95% de los encuestados pensaban que si se debería implementar un sistema informático para mejorar las Gestión de ventas e inventarios en la “Despensa Santa Fe”.

3.5. Guía de entrevista y datos que fueron proporcionados por el experto

3.5.1. Guía de Entrevista.

Los datos que a continuación le pediremos, serán esenciales para valorar la pertinencia de la tecnología a implantar en la Despensa “Santa Fe” del Cantón Caluma Provincia De Bolívar.

Datos de Identificación.

Nombre: _____ Estudios profesionales: _____

E-mail: _____ Nombre de la empresa: _____

Teléfono: _____ Fecha de Entrevista: _____

Las preguntas realizadas en la encuesta se redactan a continuación:

1.-) ¿Usted Se Siente Satisfecho Con La Atención Que Se Brinda Al Cliente En La Despensa Santa Fe Del Cantón Caluma Provincia Bolívar?

Sí No

2.-) ¿Cómo Califica Usted La Calidad De Los Productos Que Ofrece La Despensa Santa Fe?

- a) Excelente
- b) Muy Bueno
- c) Bueno
- d) Regular

3.) ¿Cuándo Ha Realizado Sus Compras En La Despensa Santa Fe Le Ha sucedido Alguno De Estos Casos?

- a) Ha tenido que esperar mucho para cancelar su compra.
- b) Los productos que ha comprado han estado defectuosos.
- c) La atención del personal no fue apropiado.
- d) No ha tenido ninguno de los inconvenientes anteriores.

4.-) ¿Al momento de escoger los productos que se encuentran en las perchas, ¿Encuentra información necesaria de los productos con sus respectivos precios y fecha de caducidad?

Sí No

5.-) ¿Piensa Usted Que Se Debería Implantar Sistema Información En La Despensa Santa Fe Para Agilizar El Proceso De Venta De Productos?

Sí No

3.6. CONCLUSIÓN

Al concluir esta investigación de campo hemos podido obtener importantes resultados, una de las más relevantes es que con la ayuda de la tecnología de la información y comunicaciones actuales y la buena predisposición de los trabajadores de la Despensa, estaban de acuerdo con la implementación de un sistema informático para potenciar las ventas y alcanzar el éxito en los negocios es necesario contar con un sistema informático acorde a los avances tecnológicos actuales, que satisfaga los requerimientos de la Despensa "Santa Fe", Ubicado en el cantón caluma Provincia de Bolívar. Podrá ser considerada eficiente; además podemos también concluir que:

La Despensa "Santa Fe", no dispone de un Sistema de Gestión De ventas E inventarios.

3.7. RECOMENDACIONES

- El sistema que hemos propuesto creo que ayudará en gran manera, porque haría nuestra labor más eficiente y más rápida y no tendríamos que pasar tantos días en el proceso de los cálculos finales para realizar los reportes.
- Capacitar al personal que va a utilizar el sistema, mediante cursos dictados por los ingenieros a cargo del desarrollo del software.
- Establecer dos niveles de accesos (Administrador y Usuario) es último es de acceso limitado para manejar los procesos asignados por el administradora del sistema.
- Mantener la privacidad las claves de las cuentas de usuarios para disminuir el riesgo de ataques externos al sistema.

CAPÍTULO IV:
MARCO APLICATIVO.

4. MARCO APLICACIÓN.

4.1. Sistema Informático De Gestión De Ventas E Inventarios.

En el Desarrollo del Sistema Informático De Gestión De Ventas E Inventarios De La Despensa "Santa Fe" se realizara lo siguientes.

4.2. ANÁLISIS

Dentro de la Despensa "Santa Fe", se tiene una secretaria o cajera, la cual realiza el cobro de las ventas realizadas en el día.

La secretaria es la encargada de llevar el correspondiente registro de la facturación diaria de todos los productos vendidos.

En el caso de realizar algún cambio de producto la secretaria tiene que estar revisando todas las facturas emitidas anteriormente conllevando esto una demora en cuestión a los tiempos.

Cuando la propietaria desee obtener un inventario total de su mercadería la secretaria puede tardar semanas en realizar esta petición ocasionando un malestar en las mismas.

Los responsables de la Despensa "Santa Fe", y encargados de su funcionamiento es la propietaria, secretaria y vendedores.

La Despensa "Santa Fe" a través de la automatización de los servicios académicos, brindará un servicio rápido y eficiente mediante el mejoramiento en el manejo de sus procesos.

Para la realización del análisis se ha de utilizar los siguientes modelos o diagramas:

- ✓ Diagrama Funcional.
- ✓ Entidad – Relación.
- ✓ Flujo de Datos.
- ✓ Diccionario de Datos.

4.3. DIAGRAMA FUNCIONAL.

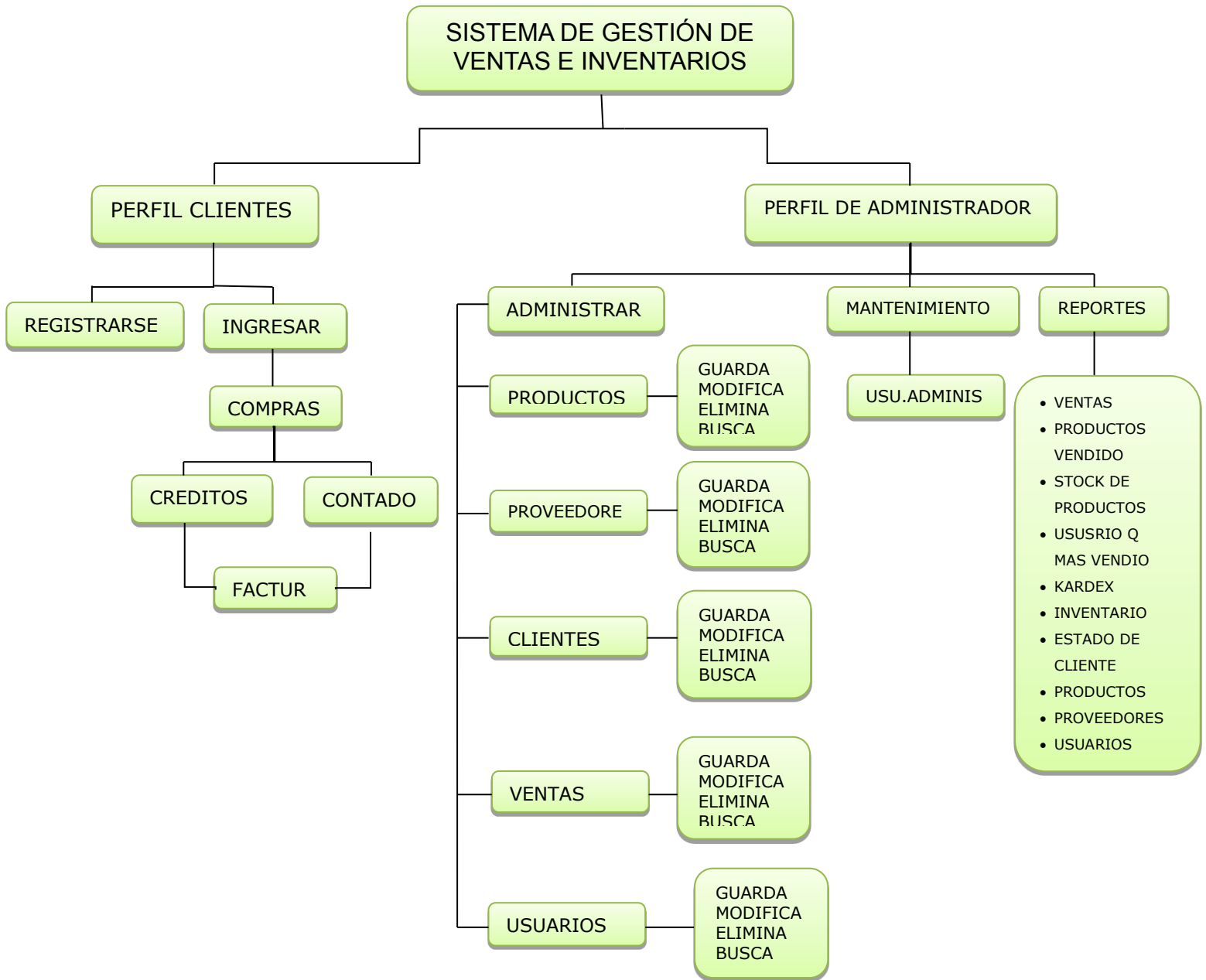


Figura IV.1. Del Diagrama Funcional # 1.

4.4. Diagrama Entidad - Relación (DER).

En este diagrama se representa las relaciones entre los objetos de datos. Dentro del mismo especificaremos los objetos de datos que entran y salen del Sistema de Gestión de Ventas E Inventarios, los atributos que definen las propiedades y las relaciones entre objetos.

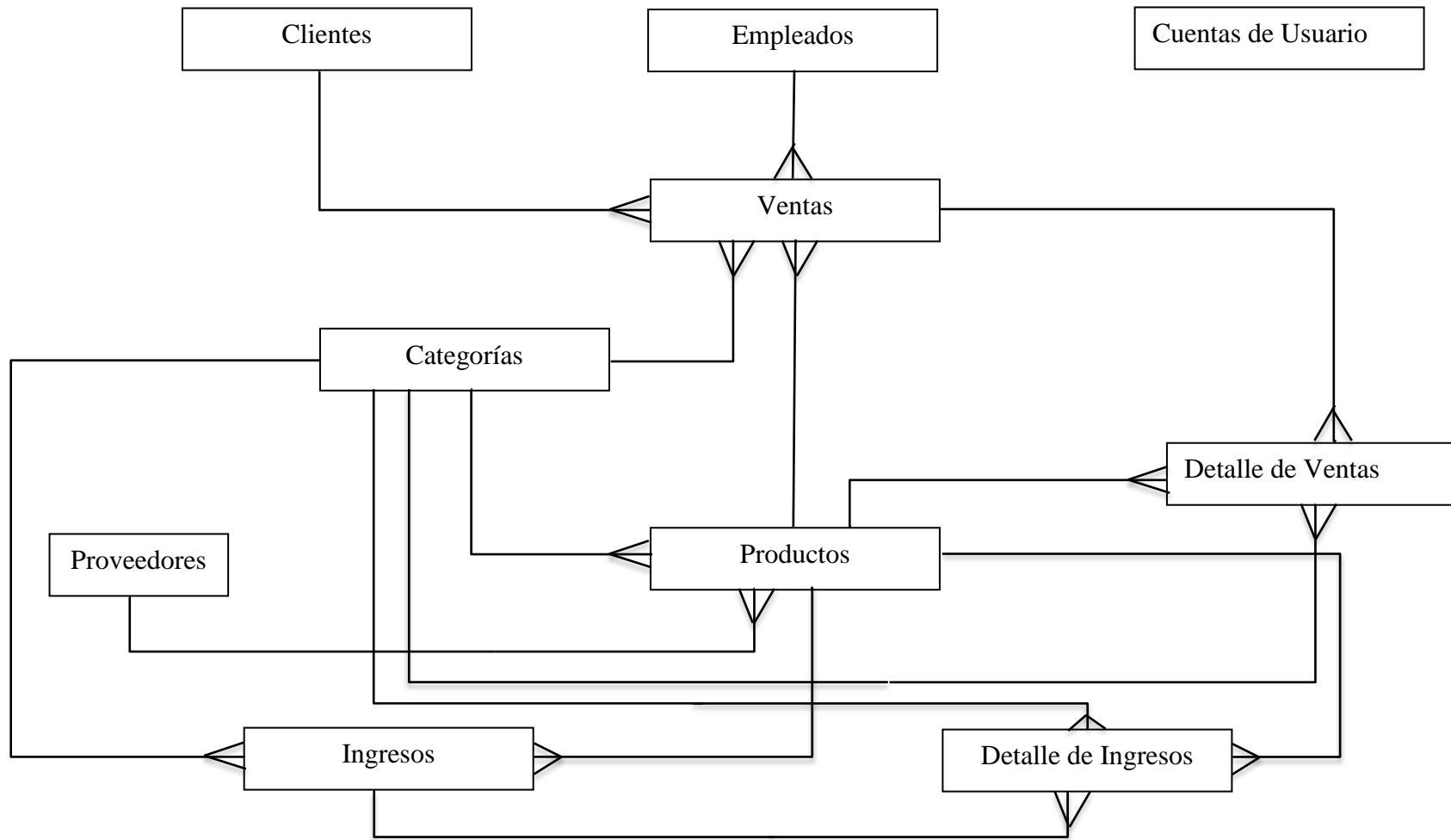
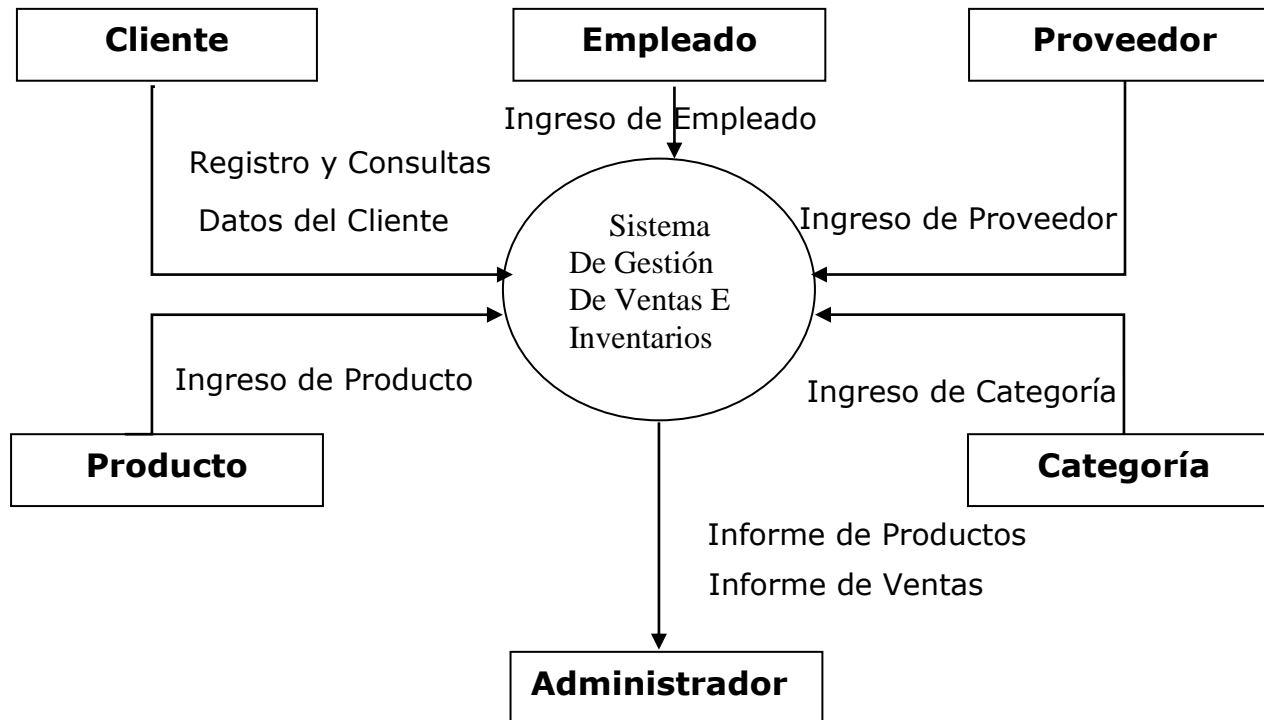


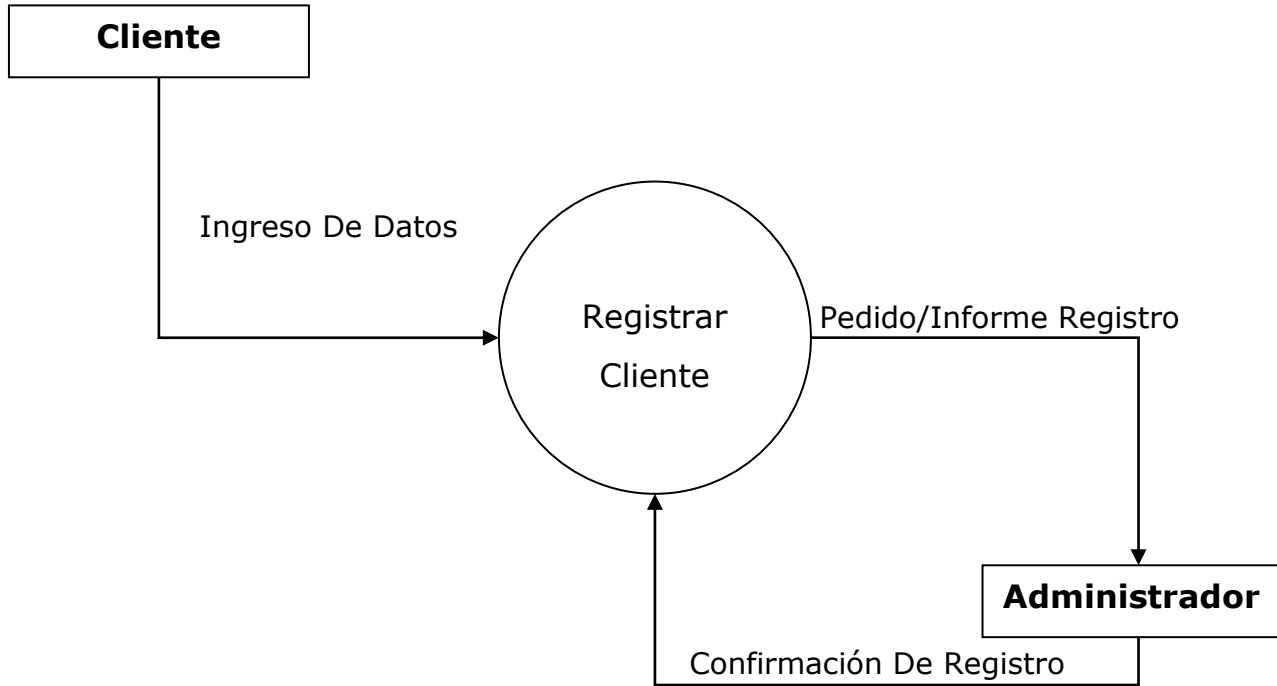
Figura IV. 4.4 Diagrama Entidad-Relación

4.5. DIAGRAMA DE FLUJO DE DATOS (DFDS)

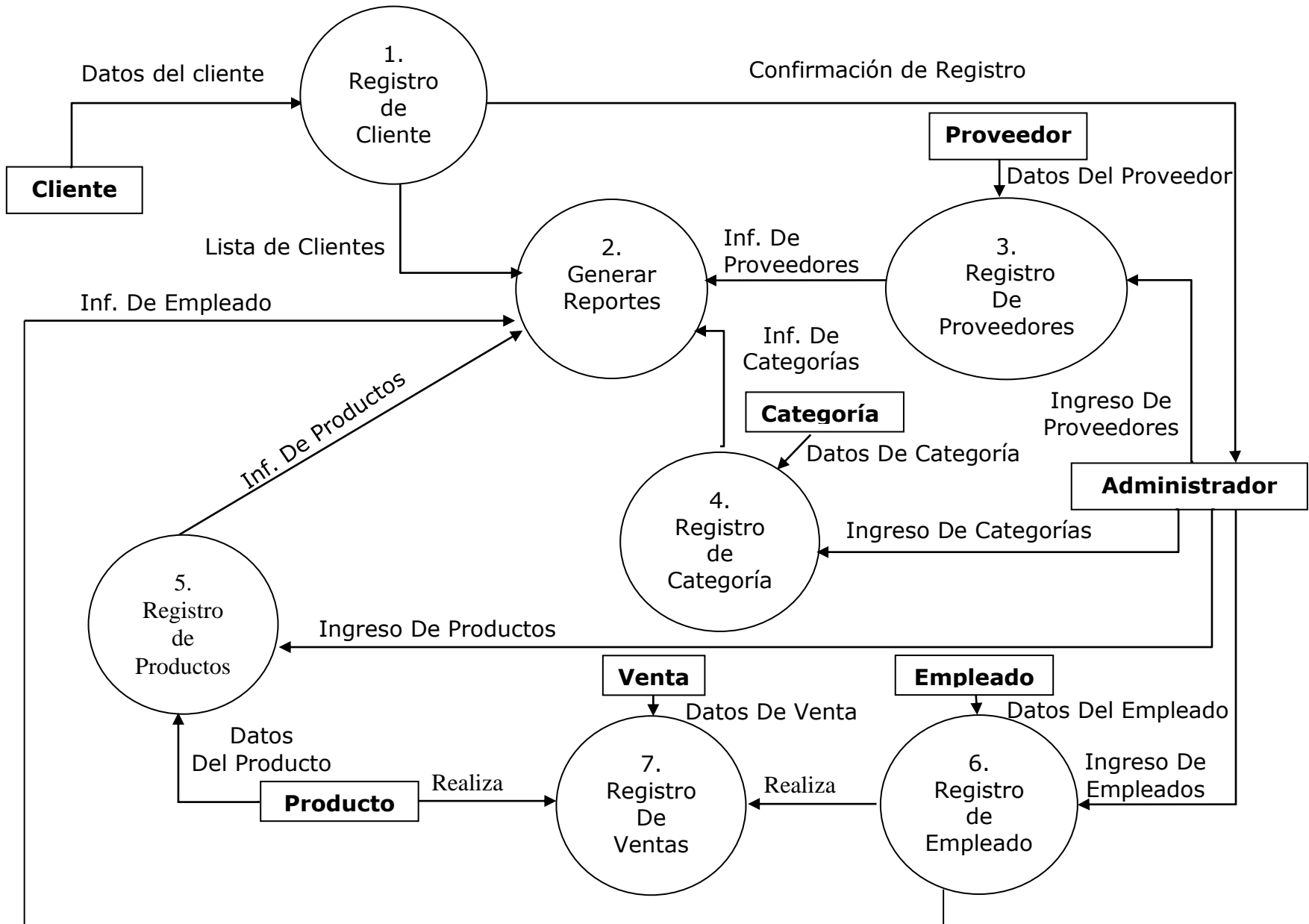
4.6. DIAGRAMA DE CONTEXTO GENERAL DEL SISTEMA



4.6.1. DIAGRAMA DE CONTEXTO NIVEL DE USUARIO

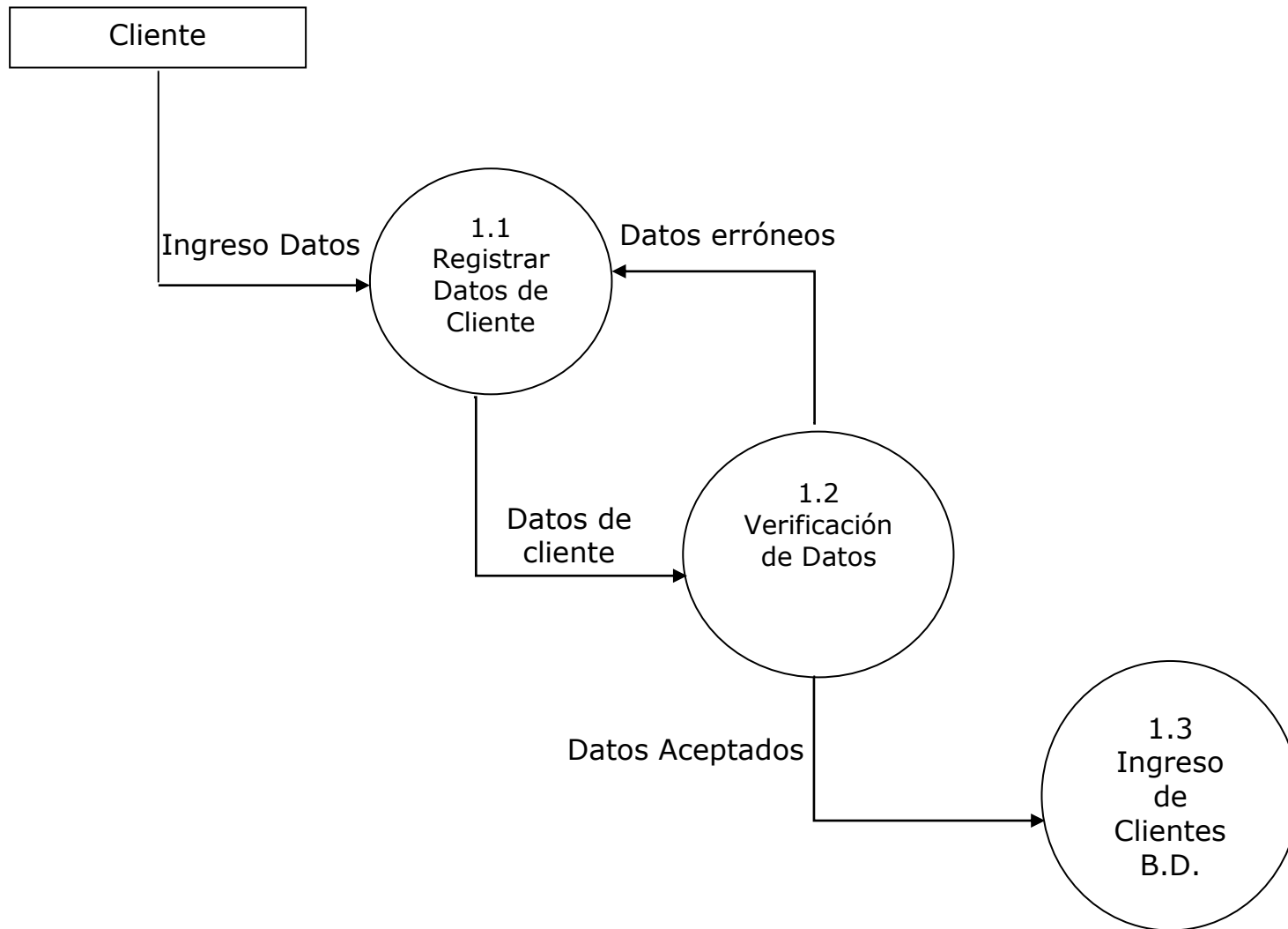


4.6.2. DIAGRAMA DE FLUJO NIVEL 0

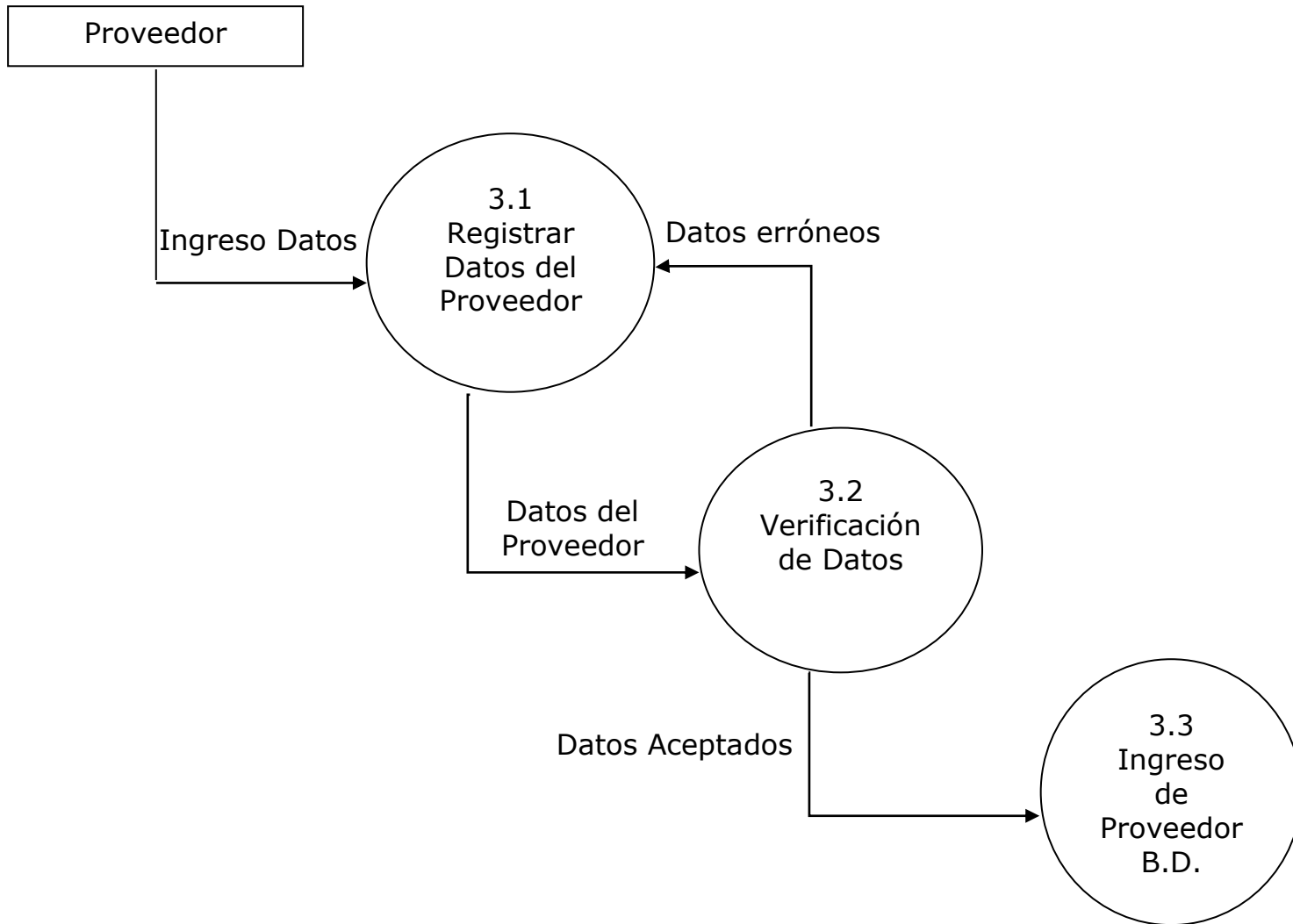


4.6.3. DIAGRAMA DE FLUJO NIVEL 1

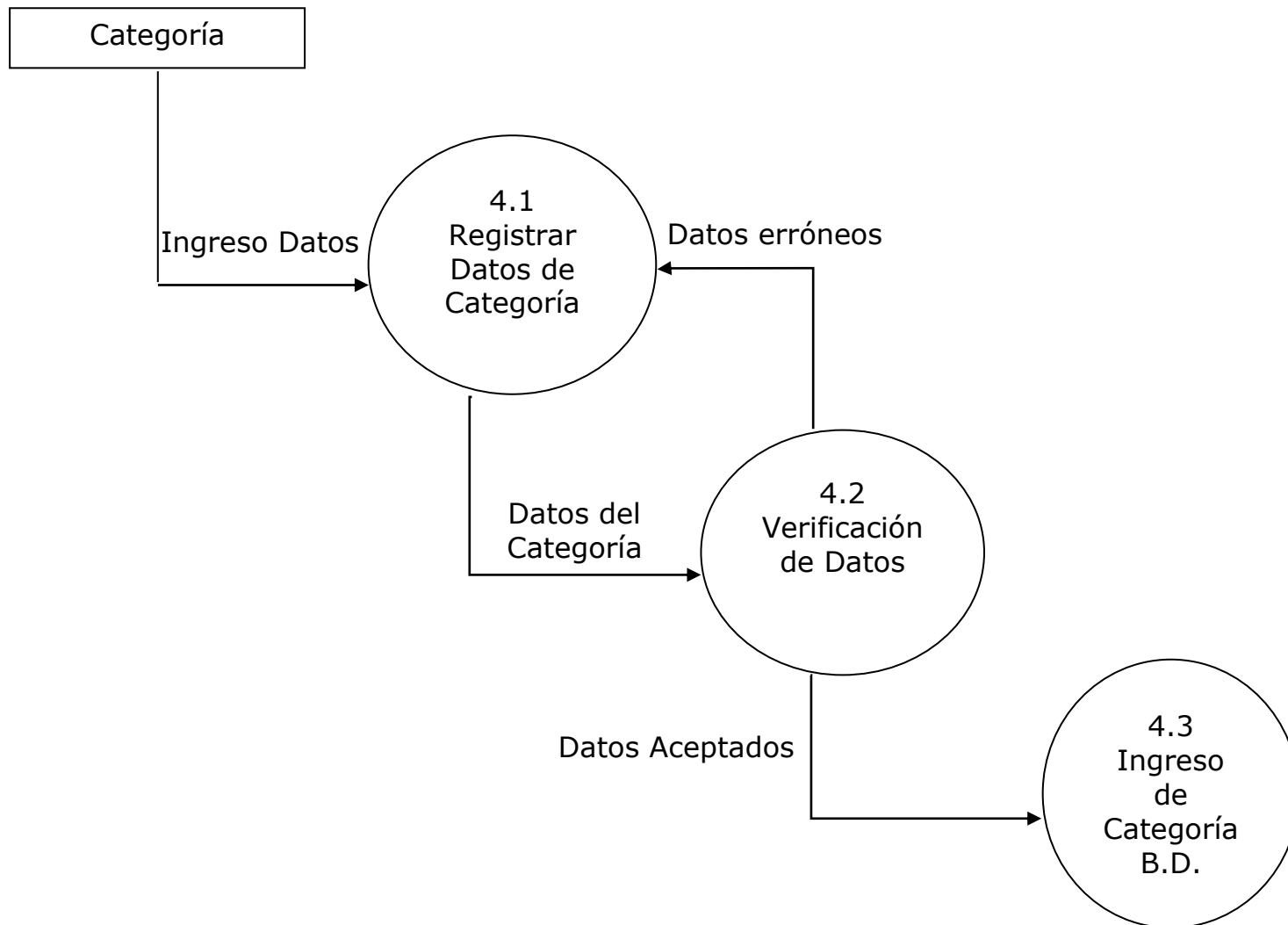
4.6.3.1. Registro De Cliente



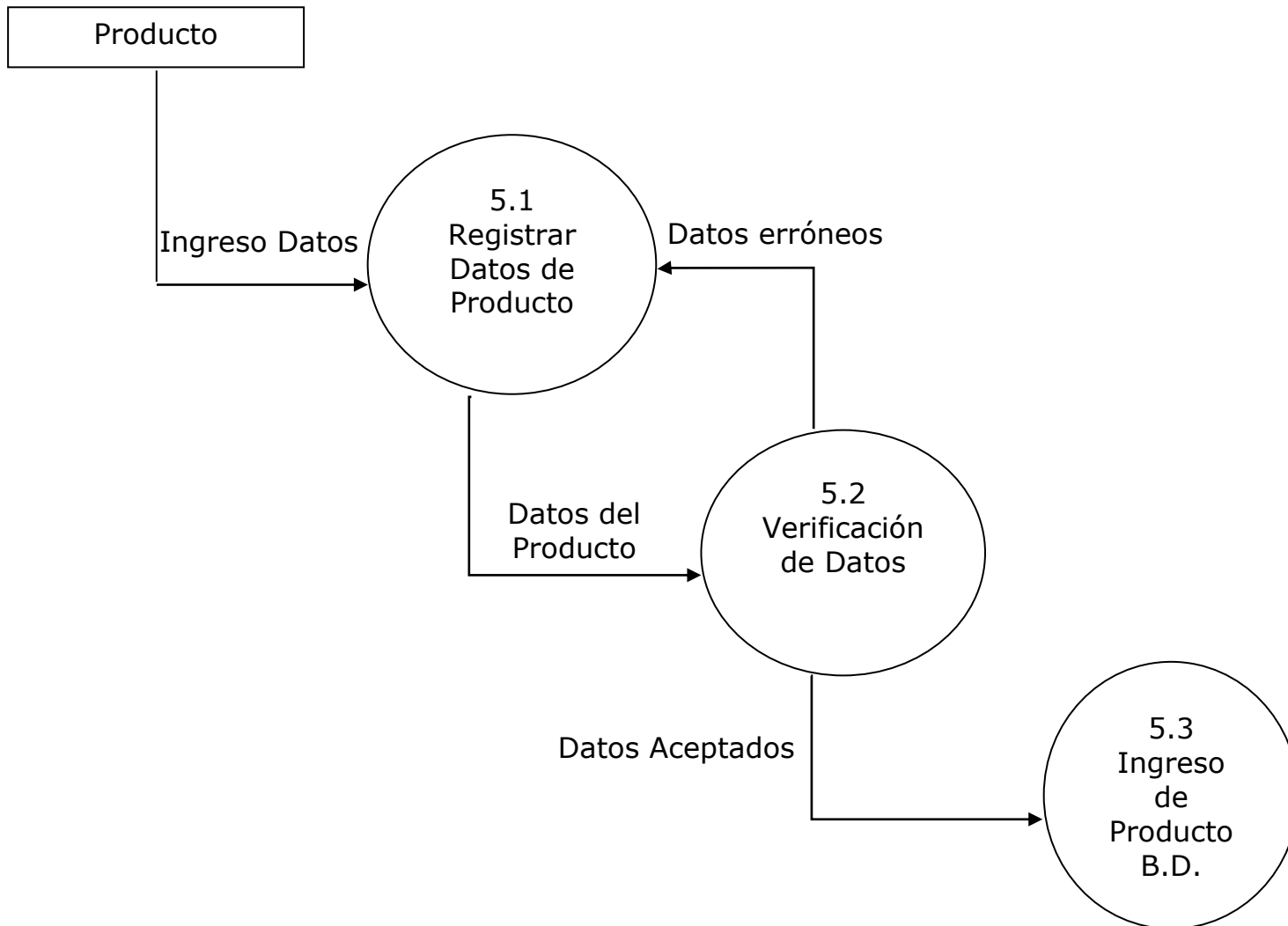
4.6.3.2. Registro De Proveedor



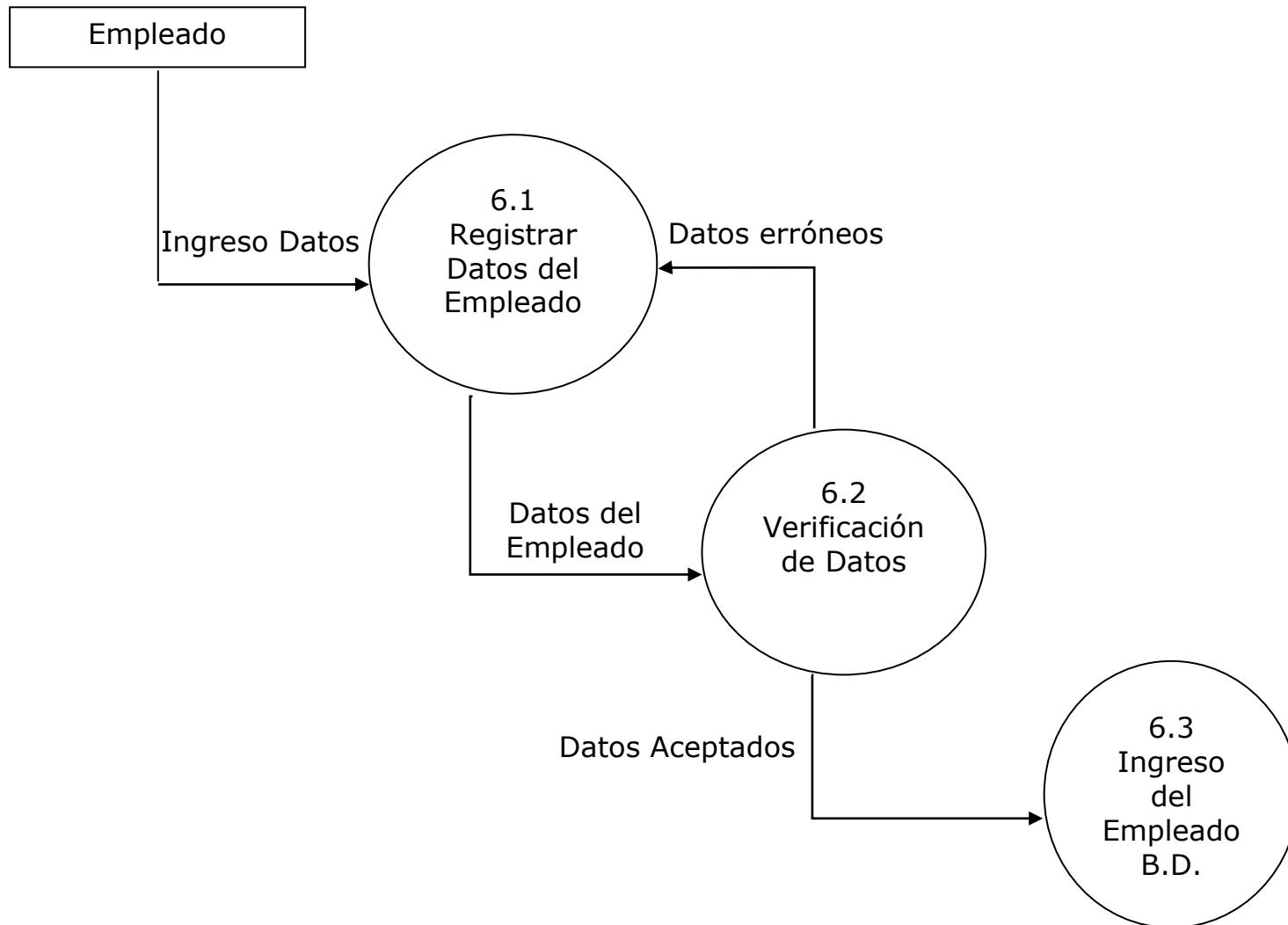
4.6.3.3. Registro De Categoría



4.6.3.4. Registro De Producto

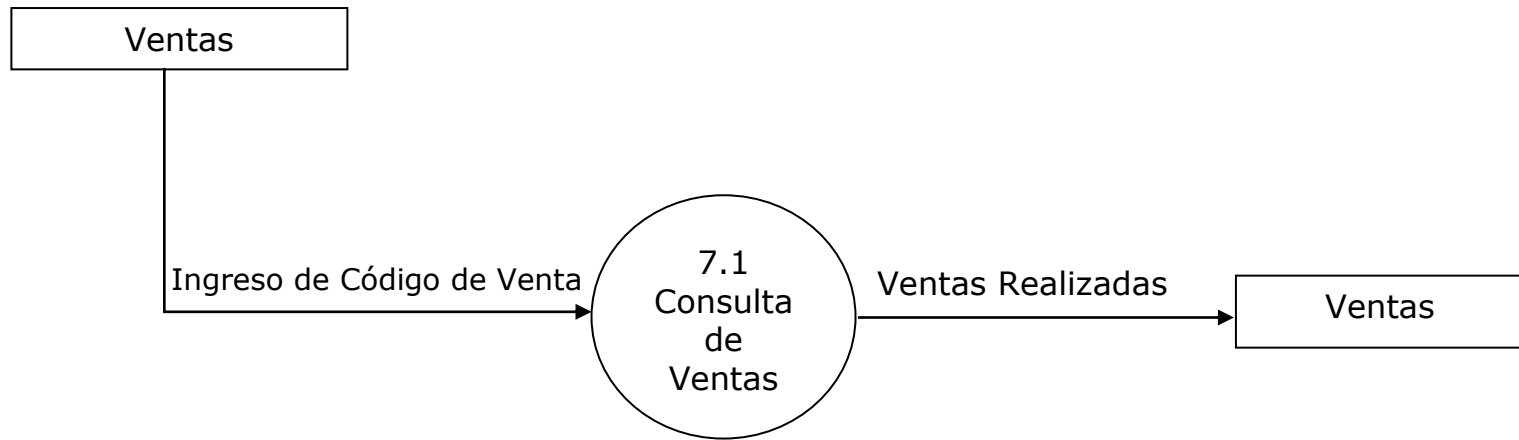


4.6.3.5. Registro De Empleado



4.6.4. DIAGRAMA DE FLUJO NIVEL 2

Consulta de Ventas



4.7. DESCRIPCIÓN DE LOS PROCESOS DEL SISTEMA

Registro del Cliente

Nombre del Proceso	1. Registrar Cliente
Descripción:	Realiza el proceso de registrar a los clientes
Flujo de Datos Internos	Ingreso de Datos Confirmación de Registro
Flujo de Datos Externos	Pedido e Informe de Registro

Verificación de Datos

Nombre del Proceso	1.1 Verificación de Datos del Cliente
Descripción:	Verifica si el nuevo cliente está bien registrado o no
Flujo de Datos Internos	Datos del cliente
Flujo de Datos Externos	Datos Aceptados.

Registro del Proveedor

Nombre del Proceso	1. Registrar Proveedor
Descripción:	Realiza el proceso de registrar a los proveedores
Flujo de Datos Internos	Ingreso de Datos Confirmación de Registro
Flujo de Datos Externos	Pedido e Informe de Registro

Verificación de Datos

Nombre del Proceso	1.1 Verificación de Datos del Proveedor
Descripción:	Verifica si el nuevo proveedor está bien registrado o no
Flujo de Datos Internos	Datos del proveedor
Flujo de Datos Externos	Datos Aceptados.

Registro de Categoría

Nombre del Proceso	1. Registrar Categoría
Descripción:	Realiza el proceso de registrar las categorías
Flujo de Datos Internos	Ingreso de Datos Confirmación de Registro
Flujo de Datos Externos	Pedido e Informe de Registro

Verificación de Datos

Nombre del Proceso	1.1 Verificación de Datos de la Categoría
Descripción:	Verifica si la nueva categoría está bien registrado o no
Flujo de Datos Internos	Datos de la Categoría
Flujo de Datos Externos	Datos Aceptados.

Registro de Producto

Nombre del Proceso	1. Registrar Los Productos
Descripción:	Realiza el proceso de registrar los productos
Flujo de Datos Internos	Ingreso de Datos Confirmación de Registro
Flujo de Datos Externos	Pedido e Informe de Registro

Verificación de Datos

Nombre del Proceso	1.1 Verificación de Datos del Producto
Descripción:	Verifica si el nuevo producto está bien registrado o no
Flujo de Datos Internos	Datos del producto
Flujo de Datos Externos	Datos Aceptados.

Registro de Empleado

Nombre del Proceso	1. Registrar al Empleado
Descripción:	Realiza el proceso de registrar a los Empleados
Flujo de Datos Internos	Ingreso de Datos Confirmación de Registro
Flujo de Datos Externos	Pedido e Informe de Registro

Verificación de Datos

Nombre del Proceso	1.1 Verificación de Datos del Empleado
Descripción:	Verifica si el nuevo empleado está bien registrado o no
Flujo de Datos Internos	Datos del empleado
Flujo de Datos Externos	Datos Aceptados.

Ingreso de Cuenta de Usuario

Nombre del Proceso	1. Ingresar Cuenta de Usuario
Descripción:	Ingresar Cuenta de Usuario del personal que accederá al sistema
Flujo de Datos Internos	Ingreso de Datos Confirmación de Registro
Flujo de Datos Externos	Pedido e Informe de Registro

Verificación de Datos

Nombre del Proceso	1.1 Verificación de Datos de Usuario
Descripción:	Verifica si la cuenta de usuario está bien registrado o no
Flujo de Datos Internos	Datos del Usuario del Sistema
Flujo de Datos Externos	Datos Aceptados.

Generar Reportes

Nombre del Proceso	1. Generar Reportes
Descripción:	Emite Reportes de los módulos del sistema a partir de los datos registrados
Flujo de Datos Internos	Datos del Cliente. Datos del Empleado. Datos del Producto. Datos de la Categoría Datos del Proveedor.
Flujo de Datos Externos	Reporte de Cliente. Reporte de Empleado. Reporte de Producto. Reporte de Categoría Reporte de Proveedor. Reporte Ventas

Generar Consultas específicas de los módulos del sistema

Nombre del Proceso	7. Genera Consulta
Descripción:	Genera consultas específicas de todos los módulos del sistema
Flujo de Datos Internos	Consulta específica del Cliente, Empleado, Producto, Categorías, Proveedor, Ventas Búsqueda de consulta específica
Flujo de Datos Externos	Respuesta de consulta específica

4.8. Diccionario De Datos (DD)

El diccionario de datos complementa el DFD específicamente cada uno de sus componentes, de esta manera el DD es un conjunto ordenado de definiciones de todos los elementos del DFD, en él se describen todos los términos utilizados con el fin de clarificar la terminología empleada en el DFD, es decir el diccionario de datos es un almacén de los datos.

En el Diccionario de Datos del proyecto de sistemas se describe ampliamente los términos utilizados en el DFD.

El siguiente grafico es el esquema general de la base de datos Despensa utilizada como almacén de datos para el sistema Gestión De Ventas E Inventario que vamos a implementar.

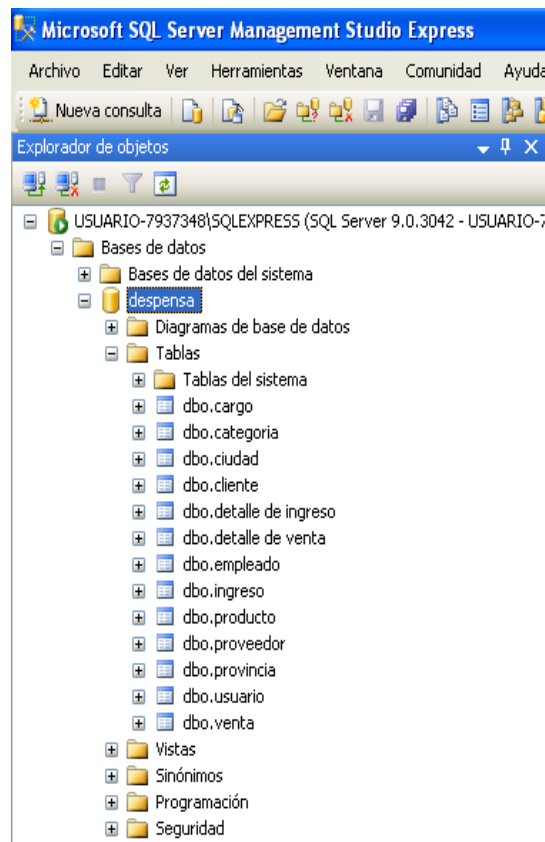


Tabla IV.1: La Base De Datos General De Las Tablas.

Esta base de Datos consta de 13 tablas, 1 Diagramas de base de datos, así como índices e información pre registrada, de manera que el sistema funcione a la perfección. Definido esto, pasamos a la explicación de lo que realiza cada una de las tablas.

Tabla Cargo

Descripción

Esta tabla permite almacenar temporalmente el cargo que tendrá el empleado en la despensa "Santa Fe"; y sus campos son:

Cod_cargo: Es de tipo entero (Int), y su longitud es 4, sirve para almacenar el código del cargo.

Cargo: Es de tipo carácter (Varchar) y su longitud es 50, sirve para almacenar el cargo.

Salario: Es de tipo decimal, su longitud es 18 dígitos en los cuales se incluye 2 decimales, permite almacenar el valor del salario que gana el empleado en la Despensa "Santa Fe".

Cargo	
	Cod_cargo Int
	Cargo Varchar(50)
	Salario Decimal(18,2)

Tabla IV. 2: Cargo.

Tabla Categoría

Descripción

Esta tabla permite almacenar temporalmente la categoría existente en la despensa "Santa Fe"; y sus campos son:

Cod_categoria: Es de tipo entero (Int), sirve para almacenar el código de la categoría.

Nombre: Es de tipo carácter (Varchar) y su longitud es 50, sirve para almacenar el nombre de la categoría.

Imagen: Es de tipo carácter (Varchar) y su longitud es 150, sirve para almacenar la dirección de la imagen.


Categoría		
	Cod_categoria	Int
	Nombre	Varchar(50)
	Imagen	Varchar(150)

Tabla IV. 3: Categoría.

Tabla Ciudad

Descripción

Esta tabla permite almacenar temporalmente la ciudad; y sus campos son:

Cod_ciudad: Es de tipo entero (Int), sirve para almacenar el código de la categoría.

Cod_provincia: Es de tipo entero (Int), sirve para almacenar el código de la provincia.

Nombre: Es de tipo carácter (Varchar) y su longitud es 50, sirve para almacenar el nombre de la ciudad.


Ciudad		
	Cod_ciudad	Int
	Cod_provincia	Int
	Nombre	Varchar(50)

Tabla IV. 4: Ciudad.

Tabla Cliente

Descripción

Esta tabla permite almacenar temporalmente los datos del cliente; y sus campos son:

Cod_cliente: Es de tipo entero (Int), sirve para almacenar el código del cliente.

Nombre: Es de tipo carácter (Varchar) y su longitud es 50, sirve para almacenar el nombre del cliente.

Cod_provincia: Es de tipo entero (Int), sirve para almacenar el código de la provincia.

Cod_ciudad: Es de tipo entero (Int), sirve para almacenar el código de la ciudad.

Teléfono: Es de tipo carácter (Varchar) y su longitud es 50, sirve para almacenar el teléfono.


Cliente	
 Cod_cliente	Int
Nombre	Varchar(50)
Cod_provincia	Int
Cod_ciudad	Int
Teléfono	Varchar(50)

Tabla IV. 5: Cliente.

Tabla Detalle De Ingreso

Descripción

Esta tabla permite almacenar temporalmente el detalle de ingreso que existente en la despensa "Santa Fe"; y sus campos son:

Código: Es de tipo entero (Int), sirve para almacenar el código del Detalle de Ingreso.

Cod_ingreso: Es de tipo entero (Int), sirve para almacenar el código de ingreso.

Cod_categoría: Es de tipo entero (Int), sirve para almacenar el código de la categoría.

Cod_producto: Es de tipo entero (Int), sirve para almacenar el código del producto.

Cod_cantidad: Es de tipo entero (Int), sirve para almacenar la cantidad de detalle de ingreso.

Detalle De Ingreso		
	Código	Int
	Cod_ingreso	Int
	Cod_categoría	Int
	Cod_producto	Int
	Cod_cantidad	Int

Tabla IV. 6: Detalle De Ingreso.

Tabla Detalle De Venta

Descripción

Esta tabla permite almacenar temporalmente el detalle de venta que existente en la despensa "Santa Fe"; y sus campos son:

Código: Es de tipo entero (Int), sirve para almacenar el código del Detalle de venta.

Cod_venta: Es de tipo entero (Int), sirve para almacenar el código de venta.

Cod_categoría: Es de tipo entero (Int), sirve para almacenar el código de la categoría.

Cod_producto: Es de tipo entero (Int), sirve para almacenar el código del producto.

Cod_cantidad: Es de tipo entero (Int), sirve para almacenar la cantidad de detalle de venta.


Detalle De Venta		
	Código	Int
	Cod_venta	Int
	Cod_categoria	Int
	Cod_producto	Int
	Cod_cantidad	Int

Tabla IV. 8: Detalle De Venta.

Tabla Empleado

Descripción

Esta tabla permite almacenar temporalmente los datos del Empleado que está trabajando en la despensa "Santa Fe"; y sus campos son:

Cedula: Es de tipo carácter (Varchar) y su longitud es 11, sirve para almacenar el número de cedula del empleado.

Nombre: Es de tipo carácter (Varchar) y su longitud es 50, sirve para almacenar el nombre del empleado.

Cod_cargo: Es de tipo entero (Int), sirve para almacenar el código de cargo que ocupa el empleado en la Despensa "Santa Fe".

Fecha_nac: Es de tipo fecha (datetime), se incluye la hora, que nos permite almacenar la fecha de nacimiento del empleado para trabajar en la Despensa "Santa Fe".

Cod_provincia: Es de tipo entero (Int), sirve para almacenar el código de la provincia actual del empleado.

Cod_ciudad: Es de tipo entero (Int), sirve para almacenar el código de la ciudad actual donde vive el empleado.

Dirección: Es de tipo carácter (Varchar) y su longitud es 50, sirve para almacenar la dirección donde vive el empleado.

Teléfono: Es de tipo carácter (Varchar) y su longitud es 15, sirve para almacenar el número del teléfono del empleado.

Foto: Es de tipo carácter (Varchar) y su longitud es 200, sirve para almacenar la dirección de la foto del empleado que trabajara en la Despensa "Santa Fe".

Empleado		
	Cedula	Varchar(11)
	Nombre	Varchar(50)
	Cod_cargo	Int
	Fecha_nac	Datetime
	Cod_provincia	Int
	Cod_ciudad	Int
	Dirección	Varchar(50)
	Teléfono	Varchar(15)
	Foto	Varchar(200)

Tabla IV. 9: Empleado.

Tabla Ingreso

Descripción

Esta tabla permite almacenar temporalmente los productos que se Ingresan a la despensa "Santa Fe"; y sus campos son:

Cod_ingreso: Es de tipo entero (Int), sirve para almacenar el código de ingreso que compra la Despensa "Santa Fe" por cada producto.

Proveedor: Es de tipo entero (Int), sirve para almacenar el número de proveedores.

Fecha: Es de tipo fecha (datetime), se incluye la hora, y nos permite almacenar la fecha que ingresamos los productos en la Despensa.

Subtotal: Es de tipo decimal, su longitud es 18 dígitos en los cuales se incluye 2 decimales, y nos permite almacenar el subtotal de los productos en la Despensa "Santa Fe".

IVA: Es de tipo decimal, su longitud es 18 dígitos en los cuales se incluye 2 decimales, permite almacenar el IVA de los productos que se encuentra en la Despensa "Santa Fe".

Total: Es de tipo decimal, su longitud es 18 dígitos en los cuales se incluye 2 decimales, permite almacenar el total de los productos que se encuentra en la Despensa "Santa Fe".


Ingreso	
 Cod_ingreso	Int
Proveedor	Int
Fecha	Datetime
Subtotal	Decimal(18,2)
IVA	Decimal(18,2)
Total	Decimal(18,2)

Tabla IV. 10: Ingreso

Tabla Producto

Descripción

Esta tabla permite almacenar temporalmente el nombre de cada producto que ingresa a la despensa "Santa Fe"; y sus campos son:

Cod_producto: Es de tipo entero (Int), sirve para almacenar el código del producto.

Nombre: Es de tipo carácter (Varchar) y su longitud es 50, sirve para almacenar el nombre del producto.

Cod_proveedor: Es de tipo entero (Int), sirve para almacenar el código del proveedor.

Cod_categoria: Es de tipo entero (Int), sirve para almacenar el código de la categoría.

Cantidad: Es de tipo entero (Int), sirve para almacenar la cantidad de productos en existencia en la Despensa "Santa Fe".

Precio: Es de tipo decimal, su longitud es 18 dígitos en los cuales se incluye 2 decimales, permite almacenar el precio del producto.

Descuento: Es de tipo entero (Int), sirve para almacenar un descuento al cliente por cada producto que compra.

Fecha_elab: Es de tipo fecha (datetime), se incluye la hora, y nos permite almacenar la fecha de elaboración del producto.

Fecha_cad: Es de tipo fecha (datetime), se incluye la hora, y nos permite almacenar la fecha de caducidad del producto.


Producto	
 Cod_producto	Int
Nombre	Varchar(50)
Cod_proveedor	Int
Cod_categoria	Int
Cantidad	Int
Precio	Decimal(18,2)
Descuento	Int
Fecha_elab	Datetime
Fecha_cad	Datetime

Tabla IV. 11: Producto.

Tabla Proveedor

Esta tabla permite almacenar temporalmente los datos del proveedor; y sus campos son:

Cod_proveedor: Es de tipo entero (Int), sirve para almacenar el código del proveedor.

Campania: Es de tipo carácter (Varchar) y su longitud es 50, sirve para almacenar el nombre de la compañía.

Ruc: Es de tipo carácter (Varchar) y su longitud es 15, sirve para almacenar el ruc del proveedor.

Nombre: Es de tipo carácter (Varchar) y su longitud es 50, sirve para almacenar el nombre del proveedor.

Teléfono: Es de tipo carácter (Varchar) y su longitud es 15, sirve para almacenar el número del teléfono.

Cod_provincia: Es de tipo entero (Int), sirve para almacenar el código de la provincia.

Cod_ciudad: Es de tipo entero (Int), sirve para almacenar el código de la ciudad.

Dirección: Es de tipo carácter (Varchar) y su longitud es 50, sirve para almacenar la dirección del proveedor.


Proveedor	
	Cod_proveedor Int
	Campania Varchar(50)
	Ruc Varchar(15)
	Nombre Varchar(50)
	Teléfono Varchar(15)
	Cod_provincia Int
	Cod_ciudad Int
	Dirección Varchar(50)

Tabla IV. 12: Proveedor.

Tabla Provincia

Descripción

Esta tabla permite almacenar temporalmente de que provincia es el cliente que compra en la despensa "Santa Fe"; y sus campos son:

Cod_provincia: Es de tipo entero (Int), sirve para almacenar el código de la provincia.

Nombre: Es de tipo carácter (Varchar) y su longitud es 50, sirve para almacenar el nombre de la provincia.


Provincia		
	Cod_provincia	Int
	Nombre	Varchar(50)

Tabla IV. 13: Provincia.

Tabla Venta

Descripción

Esta tabla permite almacenar temporalmente las ventas diarias que produce la despensa "Santa Fe"; y sus campos son:

Cod_venta: Es de tipo entero (Int), sirve para almacenar el código de la venta.

Empleado: Es de tipo carácter (Varchar) y su longitud es 50, sirve para almacenar el nombre del empleado.

Cod_cliente: Es de tipo entero (Int), sirve para almacenar el código del cliente.

Fecha hora: Es de tipo fecha (datetime), se incluye la hora, y nos permite almacenar la fecha y la hora que se vende el producto.

Subtotal: Es de tipo decimal, su longitud es 18 dígitos en los cuales se incluye 2 decimales, permite almacenar el subtotal del producto.

Descuento: Es de tipo entero (Int), sirve para almacenar un descuento en las ventas de los productos.

IVA: Es de tipo decimal, su longitud es 18 dígitos en los cuales se incluye 2 decimales, permite almacenar el IVA que se cobra por los productos que se encuentran en la despensa.


Venta	
 Cod_venta	Int
Empleado	Varchar(11)
Cod_cliente	Int
Fecha hora	Datetime
Subtotal	Decimal(18,2)
Descuento	Int
IVA	Decimal(18,2)

Tabla IV. 14: Venta

Tabla Usuario

Descripción

Esta tabla permite almacenar temporalmente el nombre del Usuario, Del Sistema de Gestión de Ventas e Inventario de la despensa "Santa Fe"; y sus campos son:

Usuario: Es de tipo carácter (Varchar) y su longitud es 50, sirve para almacenar el nombre del Usuario.

Password: Es de tipo carácter (Varchar) y su longitud es 50, sirve para almacenar la contraseña del Usuario.

Tipo: Es de tipo carácter (Varchar) y su longitud es 50, sirve para almacenar el tipo de Usuario que ingresara al sistema.

Usuario	
⌘	Usuario Varchar(50)
	Password Varchar(50)
	Tipo Varchar(50)

Tabla IV. 15: Usuario.

4.9. SCRIPT DE BASE DE DATOS.

-- Base de datos: ` Despensa `

Tabla: Cargo

```

USE [despensa]
GO
/***** Objeto: Table [dbo].[cargo]
Fecha de la secuencia de comandos: 06/02/2012 09:36:47 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[cargo](
    [cod_cargo] [int] NOT NULL,
    [cargo] [varchar](50) NULL,
    [salario] [decimal](18, 2) NULL,
    CONSTRAINT [PK_cargo] PRIMARY KEY CLUSTERED
(
    [cod_cargo] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF

```


Tabla: Categoría

```
USE [despensa]
GO
/***** Objeto: Table [dbo].[categoria]
Fecha de la secuencia de comandos: 06/02/2012 09:38:56 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[categoria](
    [cod_categoria] [int] NOT NULL,
    [nombre_categoria] [varchar](50) NULL,
    [descripcion] [varchar](70) NULL,
    [imagen] [varchar](200) NULL,
    CONSTRAINT [PK_categoria] PRIMARY KEY CLUSTERED
(
    [cod_categoria] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
```

Tabla: Ciudad

```
USE [despensa]
GO
/***** Objeto: Table [dbo].[ciudad]
Fecha de la secuencia de comandos: 06/02/2012 09:40:41 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[ciudad](
    [cod_ciudad] [int] NOT NULL,
    [cod_provincia] [int] NOT NULL,
    [nombre_ciudad] [varchar](50) NOT NULL,
```

```

CONSTRAINT [PK_ciudad] PRIMARY KEY CLUSTERED
(
    [cod_ciudad] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
ALTER TABLE [dbo].[ciudad] WITH CHECK ADD CONSTRAINT
[FK_ciudad_provincia] FOREIGN KEY([cod_provincia])
REFERENCES [dbo].[provincia] ([cod_provincia])
GO
ALTER TABLE [dbo].[ciudad] CHECK CONSTRAINT
[FK_ciudad_provincia]

```

Tabla: Detalle De Ingreso

```

USE [despensa]
GO
/***** Objeto: Table [dbo].[detalle de ingreso] Fecha de la
secuencia de comandos: 10/27/2011 08:46:38 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[detalle de ingreso](
[codigo] [int] NOT NULL,
[cod_ingreso] [int] NOT NULL,
[cod_categoria] [int] NOT NULL,
[cod_producto] [int] NOT NULL,
[cod_cantidad] [int] NOT NULL,
CONSTRAINT [PK_detalle de ingreso] PRIMARY KEY CLUSTERED
(
[codigo] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

```

```

ALTER TABLE [dbo].[detalle de ingreso] WITH CHECK ADD
CONSTRAINT [FK_detalle de ingreso_categoria] FOREIGN
KEY([cod_categoria])
REFERENCES [dbo].[categoria] ([cod_categoria])

GO
ALTER TABLE [dbo].[detalle de ingreso] CHECK CONSTRAINT
[FK_detalle de ingreso_categoria]
GO
ALTER TABLE [dbo].[detalle de ingreso] WITH CHECK ADD
CONSTRAINT [FK_detalle de ingreso_ingreso] FOREIGN
KEY([cod_ingreso])
REFERENCES [dbo].[ingreso] ([cod_ingreso])
GO
ALTER TABLE [dbo].[detalle de ingreso] CHECK CONSTRAINT
[FK_detalle de ingreso_ingreso]
GO
ALTER TABLE [dbo].[detalle de ingreso] WITH CHECK ADD
CONSTRAINT [FK_detalle de ingreso_producto] FOREIGN
KEY([cod_producto])
REFERENCES [dbo].[producto] ([cod_producto])
GO
ALTER TABLE [dbo].[detalle de ingreso] CHECK CONSTRAINT
[FK_detalle de ingreso_producto]

```

Tabla: Detalle De Venta

```

USE [despensa]
GO
/***** Objeto: Table [dbo].[detalle de venta]
Fecha de la secuencia de comandos: 06/02/2012 09:43:19 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[detalle de venta](
    [cod_venta] [int] NOT NULL,
    [cantidad1] [int] NULL,
    [producto] [int] NULL,
    [cod_categoria] [int] NULL,
    [fecha_elaboracion] [datetime] NULL,
    [fecha_caducida] [datetime] NULL,
    [precio] [decimal](18, 2) NULL,

```

```

) ON [PRIMARY]

GO
ALTER TABLE [dbo].[detalle de venta] WITH CHECK ADD CONSTRAINT
[FK_detalle de venta_categoria] FOREIGN KEY([cod_categoria])
REFERENCES [dbo].[categoria] ([cod_categoria])
GO
ALTER TABLE [dbo].[detalle de venta] CHECK CONSTRAINT [FK_detalle
de venta_categoria]
GO
ALTER TABLE [dbo].[detalle de venta] WITH CHECK ADD CONSTRAINT
[FK_detalle de venta_producto] FOREIGN KEY([producto])
REFERENCES [dbo].[producto] ([cod_producto])
GO
ALTER TABLE [dbo].[detalle de venta] CHECK CONSTRAINT [FK_detalle
de venta_producto]
GO
ALTER TABLE [dbo].[detalle de venta] WITH CHECK ADD CONSTRAINT
[FK_detalle de venta_venta] FOREIGN KEY([cod_venta])
REFERENCES [dbo].[venta] ([cod_venta])
GO
ALTER TABLE [dbo].[detalle de venta] CHECK CONSTRAINT [FK_detalle
de venta_venta]

```

Tabla: Empleado

```

USE [despensa]
GO
/***** Objeto: Table [dbo].[empleado]
Fecha de la secuencia de comandos: 06/02/2012 09:46:05 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[empleado](
    [cedula] [varchar](20) NOT NULL,
    [nombre_empleado] [varchar](50) NULL,
    [apellido] [varchar](50) NULL,
    [cod_cargo] [int] NULL,
    [fecha_nac] [datetime] NULL,
    [cod_provincia] [int] NULL,
    [cod_ciudad] [int] NULL,

```

```

[direccion] [varchar](50) NULL,
    [telefono] [varchar](15) NULL,
    [foto] [varchar](200) NULL,
    CONSTRAINT [PK_empleado] PRIMARY KEY CLUSTERED
(
    [cedula] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

```

GO

SET ANSI_PADDING OFF

GO

```

ALTER TABLE [dbo].[empleado] WITH CHECK ADD CONSTRAINT
[FK_empleado_cargo] FOREIGN KEY([cod_cargo])
REFERENCES [dbo].[cargo] ([cod_cargo])

```

GO

```

ALTER TABLE [dbo].[empleado] CHECK CONSTRAINT
[FK_empleado_cargo]

```

GO

```

ALTER TABLE [dbo].[empleado] WITH CHECK ADD CONSTRAINT
[FK_empleado_ciudad] FOREIGN KEY([cod_ciudad])
REFERENCES [dbo].[ciudad] ([cod_ciudad])

```

GO

```

ALTER TABLE [dbo].[empleado] CHECK CONSTRAINT
[FK_empleado_ciudad]

```

GO

```

ALTER TABLE [dbo].[empleado] WITH CHECK ADD CONSTRAINT
[FK_empleado_provincia] FOREIGN KEY([cod_provincia])
REFERENCES [dbo].[provincia] ([cod_provincia])

```

GO

```

ALTER TABLE [dbo].[empleado] CHECK CONSTRAINT
[FK_empleado_provincia]

```

Tabla: Ingreso

```

USE [despensa]

```

GO

```

/***** Objeto: Table [dbo].[ingreso]

```

```

Fecha de la secuencia de comandos: 06/02/2012 09:47:18 *****/

```

```

SET ANSI_NULLS ON

```

GO

```

SET QUOTED_IDENTIFIER ON

```

```

GO
CREATE TABLE [dbo].[ingreso](
    [cod_ingreso] [int] NOT NULL,
    [fecha] [datetime] NULL,
    [cod_proveedor] [int] NULL,
    [cod_categoria] [int] NULL,
    [cod_producto] [int] NULL,
    [inventario_anterior] [int] NULL,
    [resurtido] [int] NULL,
    [inventario] [int] NULL,
    [precio_venta] [money] NULL,
    [precio] [money] NULL,
    [valor_iva] [money] NULL,
    [total] [money] NULL,
    [precio_u] [money] NULL,
    CONSTRAINT [PK_ingreso] PRIMARY KEY CLUSTERED
(
    [cod_ingreso] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

```

```

GO
ALTER TABLE [dbo].[ingreso] WITH CHECK ADD CONSTRAINT
[FK_ingreso_categoria] FOREIGN KEY([cod_categoria])
REFERENCES [dbo].[categoria] ([cod_categoria])
GO
ALTER TABLE [dbo].[ingreso] CHECK CONSTRAINT
[FK_ingreso_categoria]
GO
ALTER TABLE [dbo].[ingreso] WITH CHECK ADD CONSTRAINT
[FK_ingreso_producto] FOREIGN KEY([cod_producto])
REFERENCES [dbo].[producto] ([cod_producto])
GO
ALTER TABLE [dbo].[ingreso] CHECK CONSTRAINT
[FK_ingreso_producto]
GO
ALTER TABLE [dbo].[ingreso] WITH CHECK ADD CONSTRAINT
[FK_ingreso_proveedor] FOREIGN KEY([cod_proveedor])
REFERENCES [dbo].[proveedor] ([cod_proveedor])
GO
ALTER TABLE [dbo].[ingreso] CHECK CONSTRAINT
[FK_ingreso_proveedor]

```

```

GO
ALTER TABLE [dbo].[ingreso] WITH CHECK ADD CONSTRAINT
[FK_ingreso_proveedor2] FOREIGN KEY([cod_proveedor])
REFERENCES [dbo].[proveedor] ([cod_proveedor])
GO
ALTER TABLE [dbo].[ingreso] CHECK CONSTRAINT
[FK_ingreso_proveedor2]

```

Tabla: Producto

```

USE [despensa]
GO
/***** Objeto: Table [dbo].[producto]
Fecha de la secuencia de comandos: 06/02/2012 09:48:22 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[producto](
    [cod_producto] [int] NOT NULL,
    [nombre_producto] [varchar](50) NULL,
    [cod_proveedor] [int] NULL,
    [cod_categoria] [int] NULL,
    [cantidad] [int] NULL,
    [precio] [money] NULL,
    [descuento] [int] NULL,
    [precio_venta] [money] NULL,
    [fecha_elab] [datetime] NULL,
    [fecha_cad] [datetime] NULL,
    CONSTRAINT [PK_producto] PRIMARY KEY CLUSTERED
(
    [cod_producto] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
ALTER TABLE [dbo].[producto] WITH CHECK ADD CONSTRAINT
[FK_producto_categoria] FOREIGN KEY([cod_categoria])

```

```

REFERENCES [dbo].[categoria] ([cod_categoria])
GO
ALTER TABLE [dbo].[producto] CHECK CONSTRAINT
[FK_producto_categoria]
GO
ALTER TABLE [dbo].[producto] WITH CHECK ADD CONSTRAINT
[FK_producto_proveedor] FOREIGN KEY([cod_proveedor])
REFERENCES [dbo].[proveedor] ([cod_proveedor])
GO
ALTER TABLE [dbo].[producto] CHECK CONSTRAINT
[FK_producto_proveedor]

```

Tabla: Proveedor

```

USE [despensa]
GO
/***** Objeto: Table [dbo].[proveedor]
Fecha de la secuencia de comandos: 06/02/2012 09:49:26 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[proveedor](
    [cod_proveedor] [int] NOT NULL,
    [compania] [varchar](50) NULL,
    [ruc] [varchar](15) NULL,
    [nombre_proveedor] [varchar](50) NULL,
    [telefono] [varchar](15) NULL,
    [cod_provincia] [int] NULL,
    [cod_ciudad] [int] NULL,
    [direccion] [varchar](50) NULL,
    [foto] [varchar](200) NULL,
    CONSTRAINT [PK_proveedor1] PRIMARY KEY CLUSTERED
(
    [cod_proveedor] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF

```



```

GO
ALTER TABLE [dbo].[proveedor] WITH CHECK ADD CONSTRAINT
[FK_proveedor_ciudad] FOREIGN KEY([cod_ciudad])
REFERENCES [dbo].[ciudad] ([cod_ciudad])
GO
ALTER TABLE [dbo].[proveedor] CHECK CONSTRAINT
[FK_proveedor_ciudad]
GO
ALTER TABLE [dbo].[proveedor] WITH CHECK ADD CONSTRAINT
[FK_proveedor1_provincia] FOREIGN KEY([cod_provincia])
REFERENCES [dbo].[provincia] ([cod_provincia])
GO
ALTER TABLE [dbo].[proveedor] CHECK CONSTRAINT
[FK_proveedor1_provincia]

```

Tabla: Usuario

```

USE [despensa]
GO
/***** Objeto: Table [dbo].[Usuario1]
Fecha de la secuencia de comandos: 06/02/2012 09:50:26 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Usuario1](
    [Nu_usuario] [char](1) NOT NULL,
    [Nivel_usuario] [char](1) NULL,
    [Usuario] [varchar](20) NULL,
    [Password] [varchar](20) NULL,
    [Activo] [char](1) NULL,
    CONSTRAINT [PK_Usuario1] PRIMARY KEY CLUSTERED
(
    [Nu_usuario] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO

```

```
ALTER TABLE [dbo].[Usuario1] WITH CHECK ADD CONSTRAINT
[FK_Usuario1_Privilegios] FOREIGN KEY([Nivel_usuario])
REFERENCES [dbo].[Privilegios] ([Nivel])
GO
ALTER TABLE [dbo].[Usuario1] CHECK CONSTRAINT
[FK_Usuario1_Privilegios]
```

Tabla: Venta

```
USE [despensa]
GO
/***** Objeto: Table [dbo].[venta]
Fecha de la secuencia de comandos: 06/02/2012 09:52:14 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[venta](
    [cod_venta] [int] NOT NULL,
    [empleado] [varchar](20) NULL,
    [cod_cliente] [int] NULL,
    [fecha] [datetime] NULL,
    [subtotal] [decimal](18, 2) NULL,
    [descuento] [int] NULL,
    [neto] [decimal](18, 2) NULL,
    [iva] [decimal](18, 2) NULL,
    [total] [decimal](18, 2) NULL,
    CONSTRAINT [PK_venta] PRIMARY KEY CLUSTERED
(
    [cod_venta] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
ALTER TABLE [dbo].[venta] WITH CHECK ADD CONSTRAINT
[FK_venta_cliente] FOREIGN KEY([cod_cliente])
REFERENCES [dbo].[cliente] ([cod_cliente])
GO
ALTER TABLE [dbo].[venta] CHECK CONSTRAINT [FK_venta_cliente]
```

4.10. DISEÑO

El diseño se basa en cómo se está realizando el sistema y además la transformación de los requisitos de los datos a una forma que entienda el analista o programador y lo haremos en base a los siguientes pasos:

Diseño de Datos

Diseño Procedimental

Diseño de la Interfaz

4.10.1. Diseño De Datos

Es definir las reglas del negocio que se complementará con la fase de análisis que nos servirá para modelar la Base de Datos.

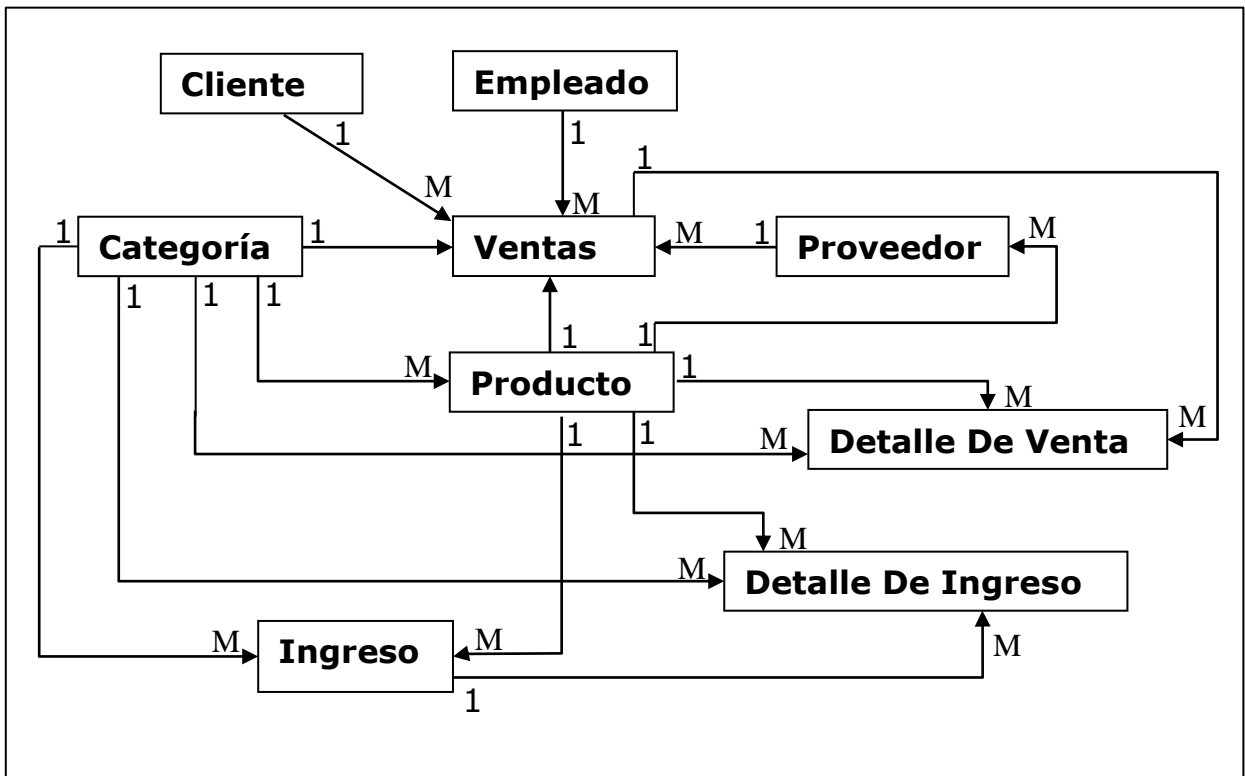


Grafico IV. 1: Entidades Y Relaciones Del Sistema.

1. Diseño Interno

Se deben definir las entidades, atributos y el tipo de datos para reservar el espacio en disco para cada uno de los atributos definidos en las entidades modeladas.

Adicionalmente, se determina la cardinalidad entre las entidades.

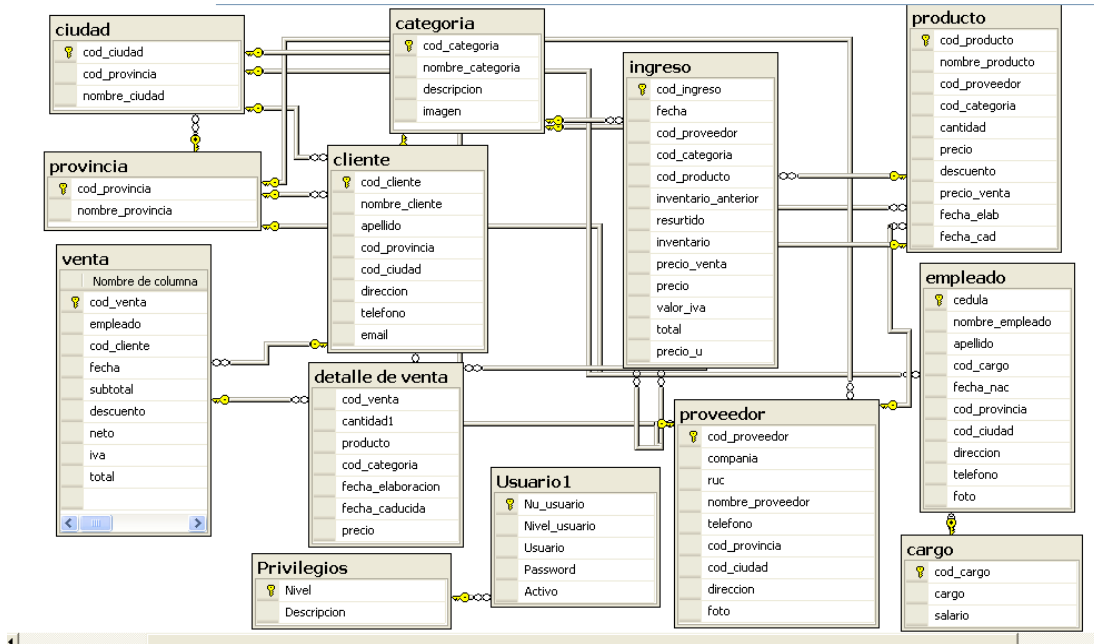


Grafico IV. 2 : Cardinalidad de las entidades

2. Integridad Referencial

Nombre de la tabla

Muestra el nombre de la tabla seleccionada en el diagrama. Si se selecciona más de una tabla en el diagrama, sólo es visible el nombre de la primera tabla.

Relación seleccionada

Muestra el nombre de la relación seleccionada en el diagrama. Si hay más de una relación seleccionada en el diagrama, sólo será visible el nombre de la primera de ellas. Expanda la lista para ver o modificar las propiedades de una relación diferente.

Cada entrada de la lista desplegable va precedida de un icono. Un icono de clave indica que la tabla participa en la relación como tabla a la que se hace referencia. Un icono de infinito indica que la tabla participa como tabla que hace referencia. La tabla que hace referencia tiene la restricción de clave externa de la relación.

Nombre de relación

Muestra el nombre de la relación seleccionada. Para cambiar el nombre de la relación, escriba otro diferente en este cuadro.

Tabla de clave principal

Muestra el nombre de la tabla con la clave principal de la relación, seguido de las columnas que forman la clave principal. Además son las tablas o entidades padres quienes contienen este tipo de atributos.

Tabla de clave externa

Muestra el nombre de la tabla con la clave externa de la relación, seguido de las columnas que forman la clave externa. Son las tablas o entidades hijas quienes contienen este tipo de atributos heredados.

4.10.2. Diseño Procedimental

Dentro del diseño procedimental vamos a realizar los diferentes códigos con su correspondiente módulo de conexión en donde ha sido creado:

```
Imports System.Data
```

```
Imports System.Data.SqlClient
```

```
Module conexionDespensa
```

```
Public conexion As New SqlConnection("Data Source=ANGEL-  
EDA5B947B\SQLEXPRESS;Initial Catalog=despensa;Integrated  
Security=True")
```

```
End Module
```

```
Imports System.Data
```

```
Imports System.Data.SqlClient
```

```
Public Class Usuario
```

```
    Dim coman As New SqlCommand  
    Dim com As SqlDataAdapter  
    Dim ds As New DataSet
```

```
Private Sub ElButton1_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles ElButton1.Click
```

```
    Dim Usuario As String  
    Dim Password As String  
    Usuario = ElEntryBox1.Text  
    Password = ElEntryBox2.Text  
    Dim Activo As String  
    conexion.Open()
```

```
Try
```

```
coman.CommandText = "select*from Usuario1 where Usuario='" +  
ElEntryBox1.Text + "' and Password='" + ElEntryBox2.Text + "'" +  
com = New SqlDataAdapter(coman.CommandText, conexion)  
ds.Clear()  
com.Fill(ds, "Usuario1")  
If ds.Tables("Usuario1").Rows.Count() = 0 Then
```

```
    If opcion = 2 Then
```

```
        MessageBox.Show("A Perdido la Oportunidad de Ingresar", "Sistema",  
        MessageBoxButtons.OK, MessageBoxIcon.Stop)
```

```
        Me.Close()
```

```
    End If
```

```
    conexion.Close()
```

```
    MessageBox.Show("Usuario o Contraseña Incorrecta", "Sistema",  
    MessageBoxButtons.OK, MessageBoxIcon.Error)
```

```
        Me.ElEntryBox1.Text = ""
```

```
        Me.ElEntryBox2.Text = ""
```

```
        Me.ElEntryBox1.Focus()
```

```
        opcion += 1
```

```
Else
```

```
    Activo = ds.Tables("Usuario1").Rows(0)(4)
```

```
    If Activo <> 1 Then
```

```
        MessageBox.Show("Usuario Inactivo", "Sistema", MessageBoxButtons.OK,  
        MessageBoxIcon.Information)
```

```
Else
```

```
    nivel = ds.Tables("Usuario1").Rows(0)(0)
```

```
    usuariov = ds.Tables("Usuario1").Rows(0)(2)
```

```
    MessageBox.Show("Bienvenido al sistema", "Sistema",  
    MessageBoxButtons.OK, MessageBoxIcon.Information)
```

```
    Timer1.Enabled = True
```

```
End If
```

```
End If
```

```
Catch ex As Exception
```

```
        MsgBox(ex.ToString)
    End Try
End Sub
```

```
Private Sub ElButton2_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ElButton2.Click
```

```
    Dim M
    M = MsgBox("¿Esta seguro de salir?", vbYesNo, "Sistema")
    If M = vbYes Then
        End
    End If
    Me.borrar()
```

```
End Sub
Private Sub borrar()
    ElEntryBox1.Text = ""
    ElEntryBox2.Text = ""
    ElEntryBox1.Focus()
End Sub
```

```
Dim i As Integer
```

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick
```

```
    Try
        i = i + 10
        ElProgress1.Value = i
        Me.Label3.Text = i
    Catch ex As Exception
        Timer1.Enabled = False
        Dim f1 As New Usuario
        Me.Hide()
        MenuDespensa.Show()
        i = 0
        'MenuD.Show()
    End Try
```

```
End Sub
```

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

```
    Timer1.Enabled = False
    ElEntryBox1.ValidationStyle.MaxLength = 13
    ElEntryBox2.ValidationStyle.MaxLength = 10
```

```
End Sub
```

```
Private Sub ElEntryBox1_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles ElEntryBox1.KeyPress
    ErrorProvider1.Clear()
```

```

If ((e.KeyChar >= "a") And (e.KeyChar <= "z")) Or ((e.KeyChar >= "A") And
(e.KeyChar <= "Z")) Or (Asc(e.KeyChar)) = 32 Or (Asc(e.KeyChar)) = 8 Or
(Asc(e.KeyChar)) = 13 Then
    e.Handled = False
Else
    e.Handled = True
    ErrorProvider1.SetError(EIEntryBox1, "Ingreso letras")
End If
If (Asc(e.KeyChar)) = 13 Then
    EIEntryBox2.Focus()
End If
End Sub
Private Sub EIEntryBox2_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles EIEntryBox2.KeyPress
    ErrorProvider1.Clear()
    If ((e.KeyChar >= "0") And (e.KeyChar <= "9")) Or (Asc(e.KeyChar)) =
8 Or (Asc(e.KeyChar)) = 13 Then
        e.Handled = False
    Else
        e.Handled = True
        ErrorProvider1.SetError(EIEntryBox2, "Ingreso numeros")
    End If
    If (Asc(e.KeyChar)) = 13 Then
        EIButton1.Focus()
    End If
End Sub
End Class

```

Fig IV.1: Procedimiento para ingresar al Sistema.

4.10.3. Diseño De La Interfaz

La interfaz que se realizó para este sistema está basado en los entornos Windows, además de que se utiliza mucha variabilidad en cuanto a la información, es decir el usuario ya no tiene que estar escribiendo demasiado. Así, tenemos una interfaz clara, legible y con ayuda para que pueda desenvolverse de una manera fácil y rápida a las personas que utilicen el sistema:

4.11. Implementación.

Instalación de visual Basic 2008

- Primero – insertamos el cd Visual Studio 2008.

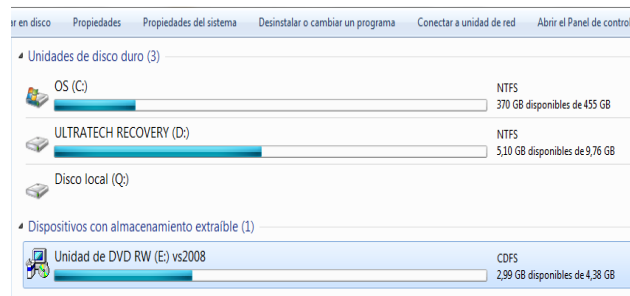


Figura 1: Continuación, el CD de Instalar Visual Studio 2008

- Damos clic derechos sobre seleccionamos instalar

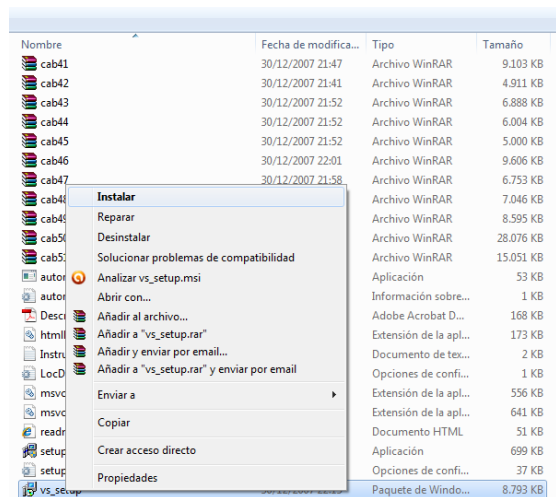


Figura 2: Damos clic en vb_setud para la Instalar Visual Studio 2008

La ejecución automática, muestra el siguiente formulario de Windows. Haga clic en el botón Léame Ver para leer la información.



Figura 3: Continuación, haga clic en Instalar Visual Studio 2008 enlace para iniciar la instalación.

- El asistente de instalacion

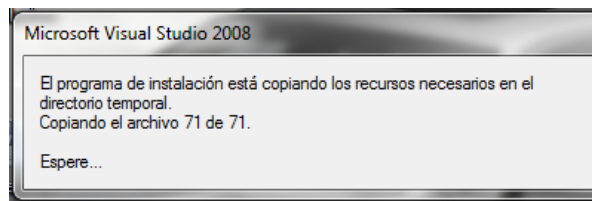


Figura 4: Comenzara a copiar los archivos necesarios en una carpeta temporal. Solo tiene que espera.

En la pagina de bienvenida del sistema de configuracion se puede activar las casilla de verificacion para evita su experiencia de instalacion de Mirosoft, si quieres.En este caso, broma dejela sim marcar.



Figura 5: Solo tiene que esperar a que el asistente para cargar los componentes de instalacio.

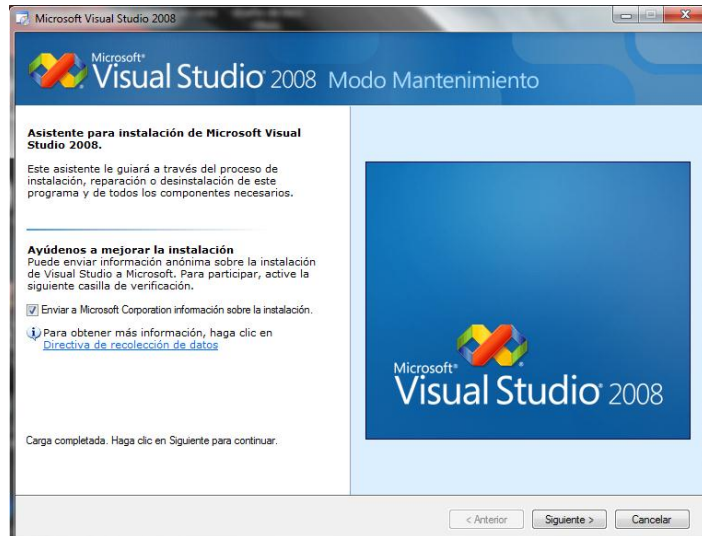


Figura 6: Haga clic en el boton siguiente para ir al siguiente paso.

El asistente de configuración de lista desplegable todos los componentes necesarios se debe instalar. Los componentes ya instalados también serán mencionados. Tenga en cuenta que Visual Studio 2008 necesita .NET Framework versión 3.5. Clave del producto y aceptar los términos de la licencia.

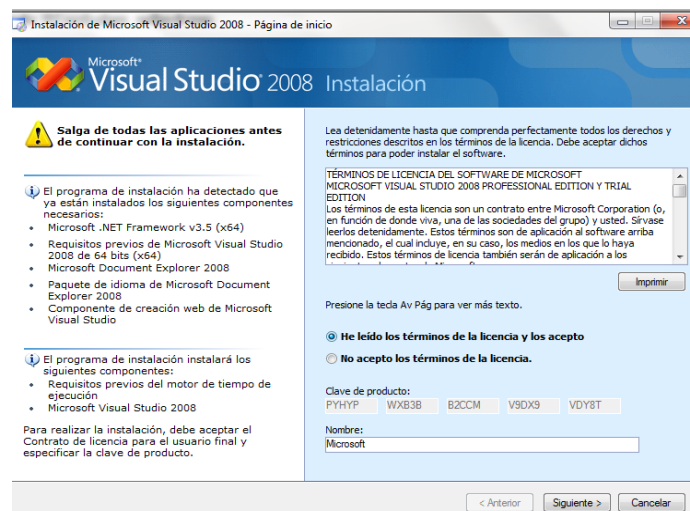


Figura 7: A continuación, haga clic en el botón Siguiente

- En el tipo de instalación, como de costumbre, tenemos tres opciones: por defecto. Completa o personalizada. En este caso, seleccionamos el tipo de instalación completa y aceptar la ruta de instalación por defecto dado. Puede cambiar la ruta de instalación y el espacio necesario para cada tipo de instalación también se muestra cuando se selecciona.

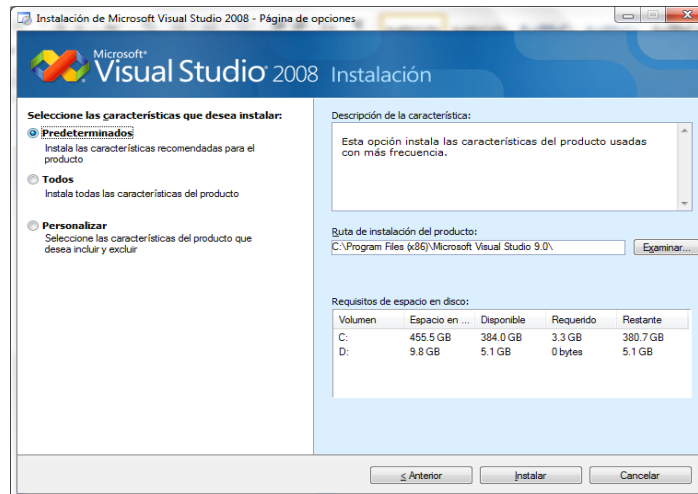


Figura 8: En este caso, seleccionamos el tipo de instalación completa y aceptar la ruta de instalación por defecto dado.

- Instalacion completar requerimiento alrededor de 4,3 GB de espacio.

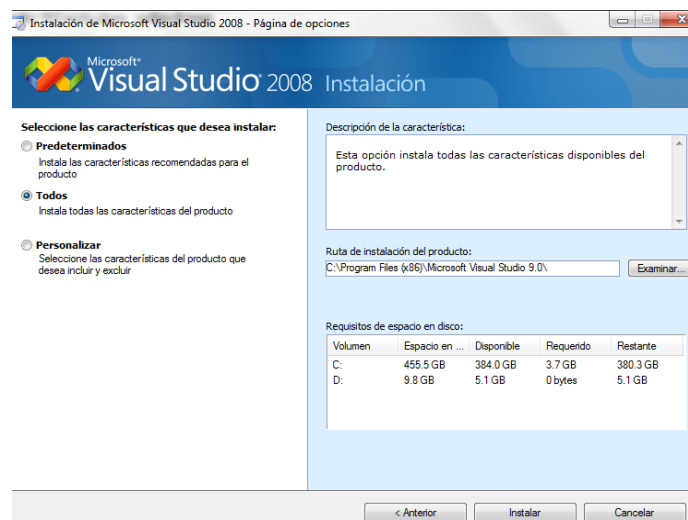


Figura 9: En este caso, seleccione la completa y haga clic en el boton

- Solo esperar y ver el paso a paso, Visual Studio 2008 componentes que se instalan.

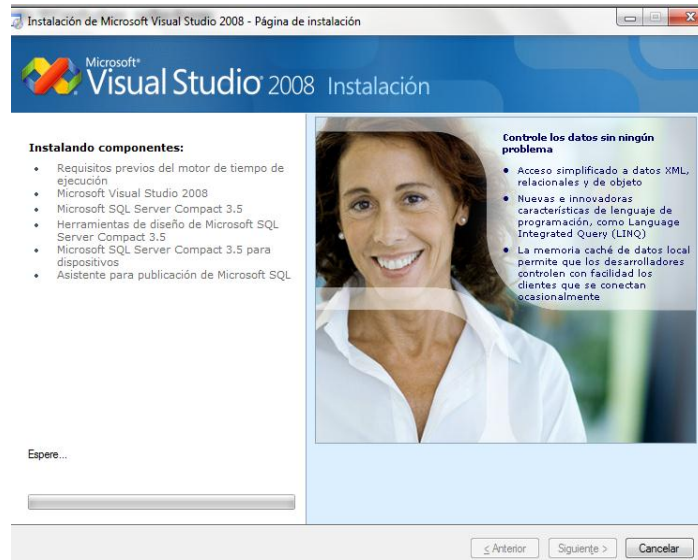


Figura 10: Comienza la instalación.

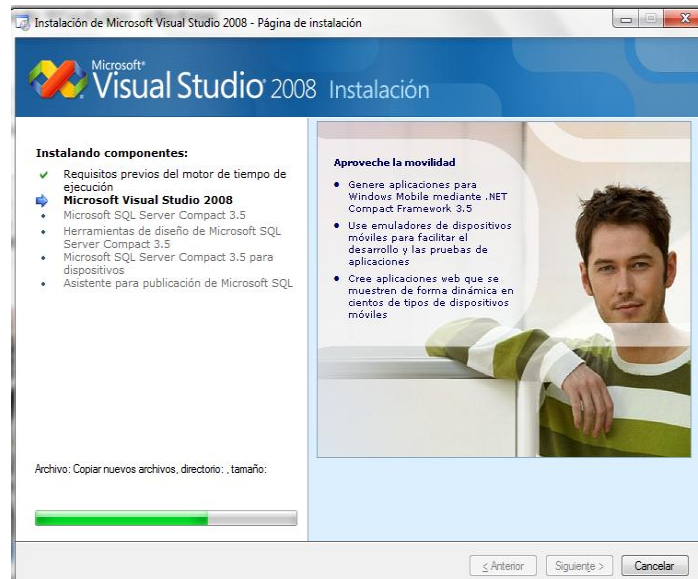


Figura 11: Pantalla de instalación de todos los componentes.

Visual Studio 2008 Standard Edition-Instalación y uso:Parte 1 Parte 2.

➤ Pasos a seguir para instalar y usar el Visual Studio 2008 Standard Edition Parte 2

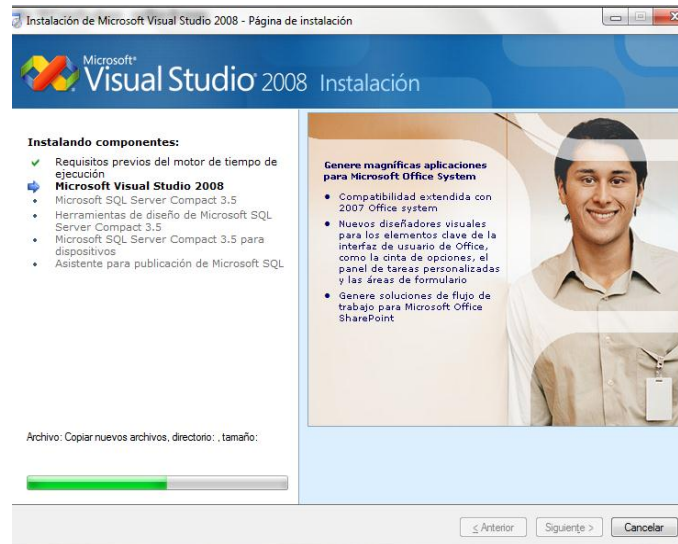


Figura 12: Paso a paso, Visual Studio 2008 componentes que se instalan.

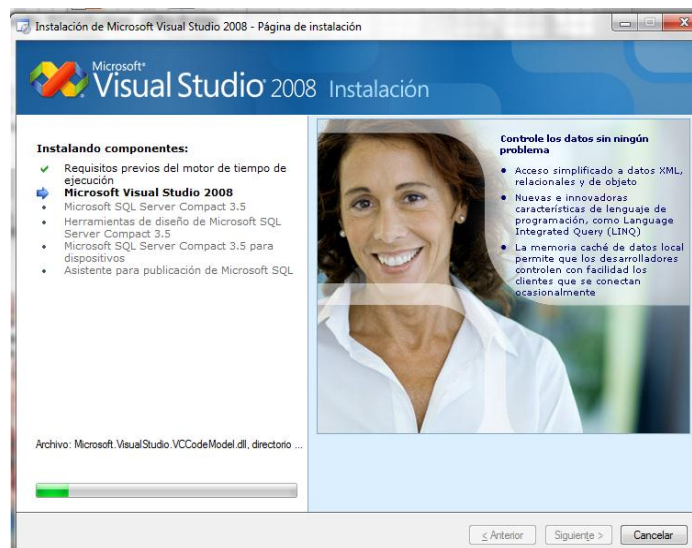


Figura 13: Paso a paso, Visual Studio 2008 componentes.

Cualquier de los componentes que se han instalado se marcara con el signo rojo en vez de la marca verde para el éxito. Después de la instalación se realiza correctamente. Puede instalarse la documentación (biblioteca MSDN) siguiendo las instrucciones de mención en la figura anterior.



Figura 14: En este caso que acabamos de salir del sistema de instalacion haciendo clic en el boton Finalizar. Visual Studio 2008.



Figura 15: Haga clic en Reiniciar ahora para reiniciar la maquina.

INSTALACIÓN SQL SERVER 2005

- Primero – insertamos el cd sql server 2005 Express

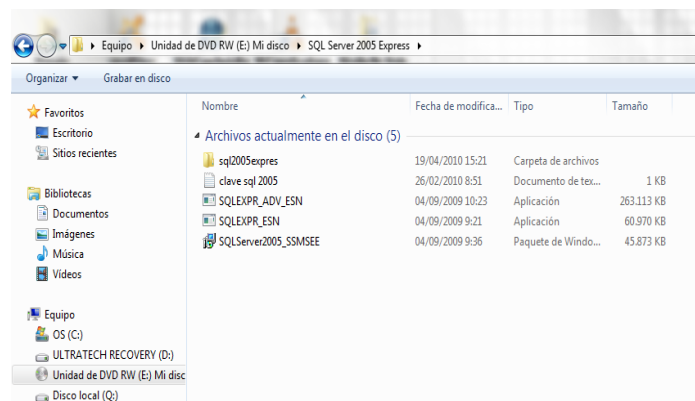


Figura 16: Continuación, el CD de Instalar De SQL Server 2005.

- Damos clic derechos sobre SQLserver2005_SSMSEE seleccionamos instalar

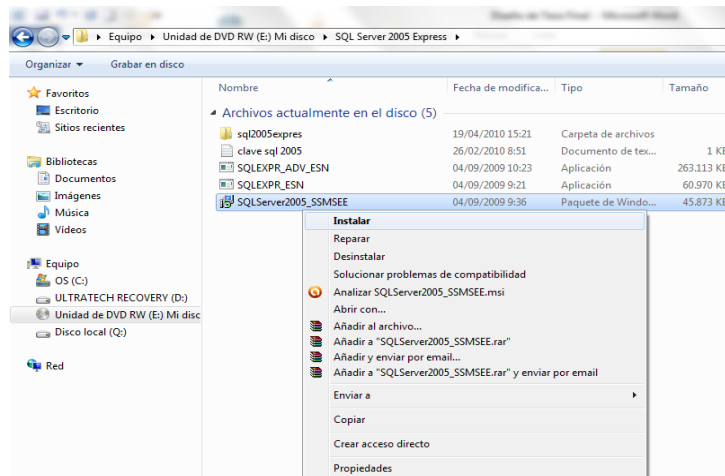


Figura 17: Damos clic en SQLServer2005_SSMSEE para la Instalar.

- clicar sobre el archivo setup e iniciar la instalación

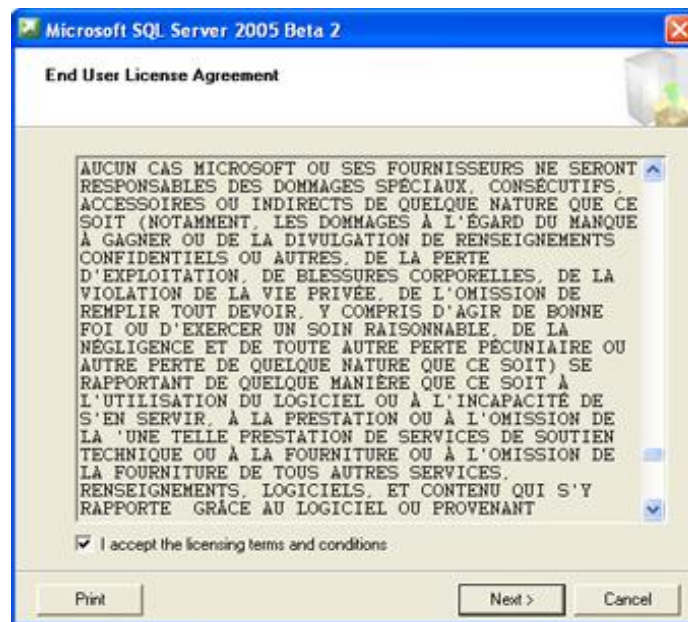


Figura 18: Aceptar el contrato.

Previamente, SQL Server instalará algunos componentes necesarios (Microsoft SQL Native Client y Microsoft SQL Server 2005 Beta 2 Setup Support Files).

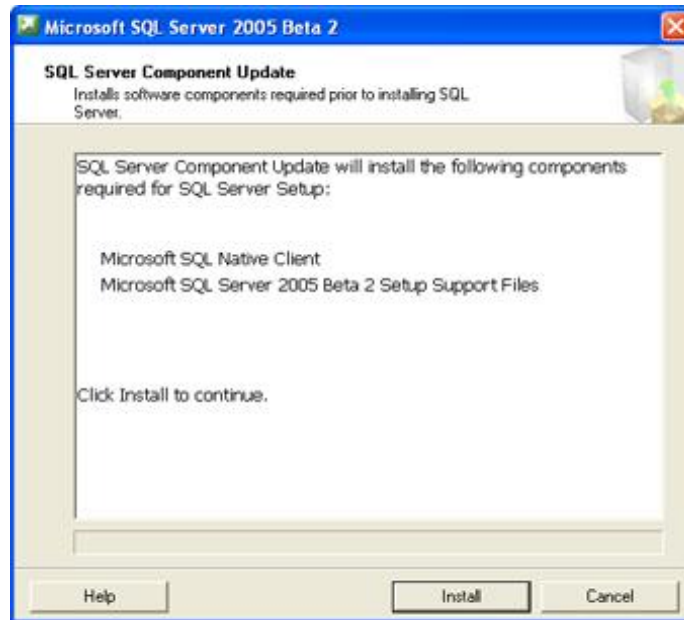


Figura 19: SQL Server instalará algunos componentes necesarios



Figura 20: Pulsaremos en "Finish" para iniciar el proceso de instalación de SQL Server.

- El asistente de instalación hará un chequeo previo de posibles problemas con algún componente de la instalación.



Figura 21: Si no hay problemas pulsaremos "Continue"

- Nos aparece una ventana de bienvenida al sistema de instalación de SQL Server 2005.

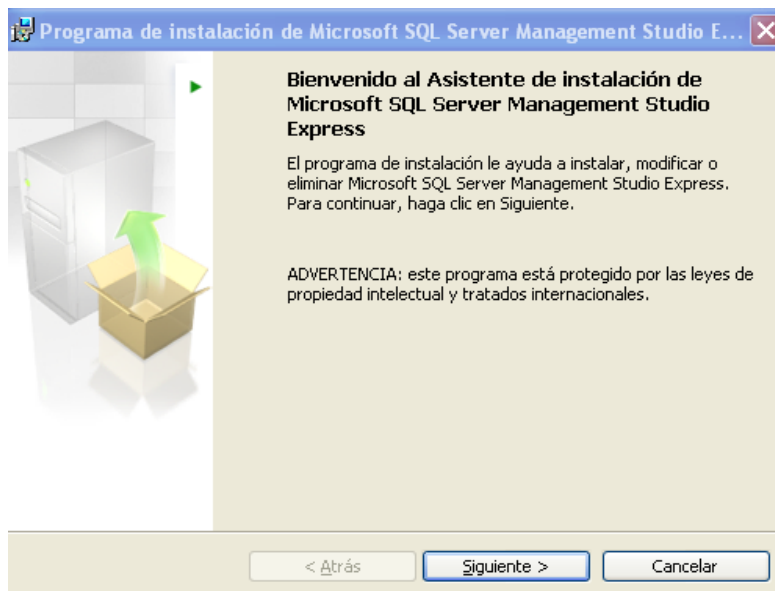


Figura 22: Damos clic en el boton Siguiente.

- En esta ventana observamos el contrato de licencia de sql server 2005 damos aceptar los términos del contrato de licencia

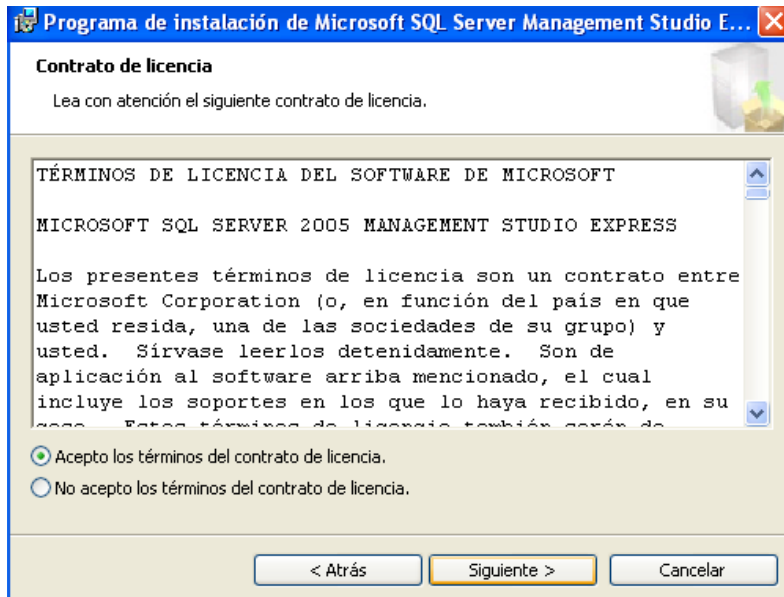


Figura 23: Damos clic en el boton Siguiente.

- Debemos Ingresar datos de la maquina para poder instalar sql server 2005.

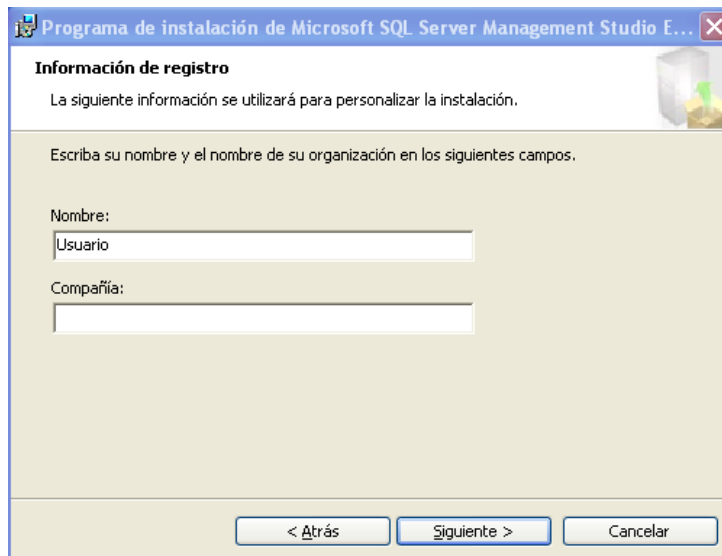


Figura 24: Damos clic en el boton Siguiente.

- Si no hay problemas con la instalación, seleccionaremos La Descripción de la características de SQL Server 2005.

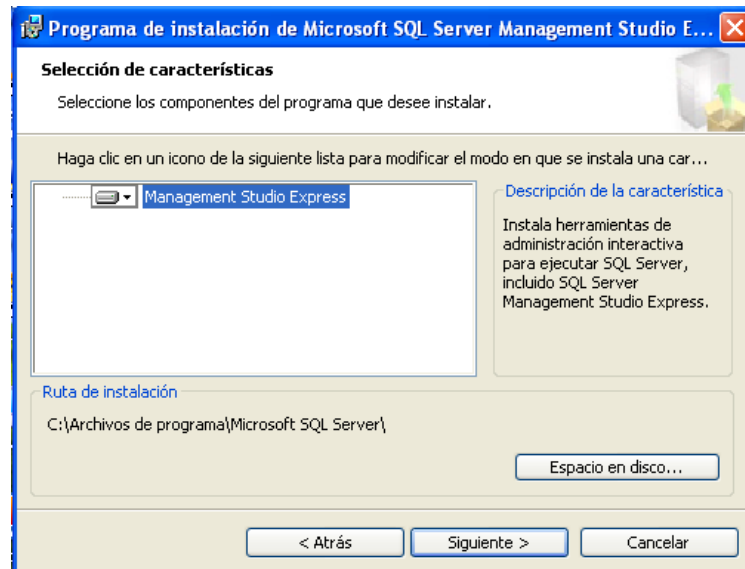


Figura 25: Haga clic en el boton Siguiente.

- Se esta preparando para instalar el programa SQL Server 2005

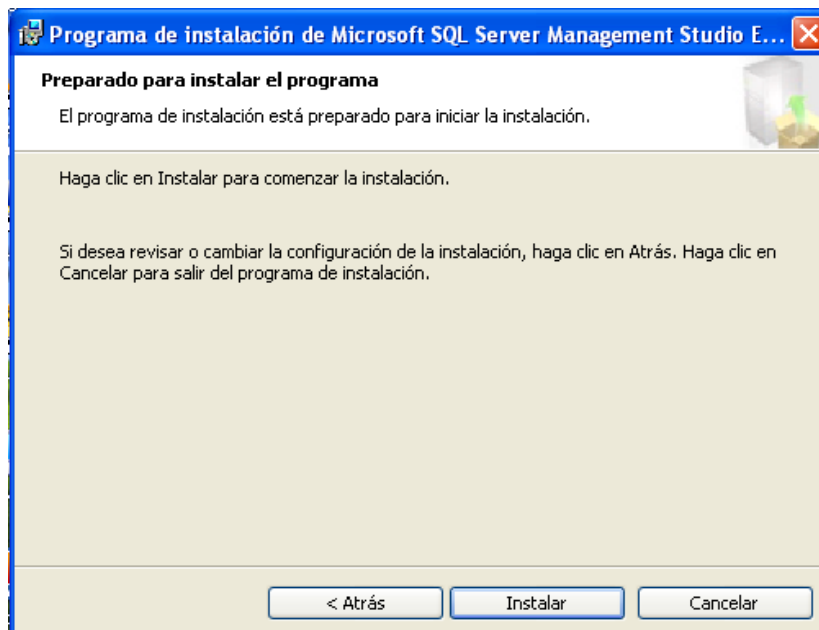


Figura 26: Haga clic en el boton Instalar.

Se está instalando las características del programa que hemos seleccionado

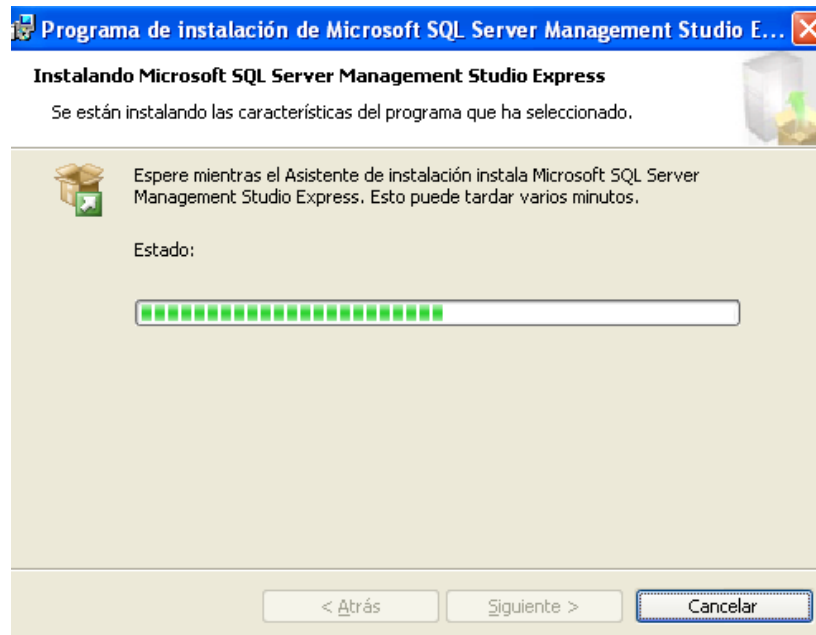


Figura 27: Esperamos que se instale todas las características.

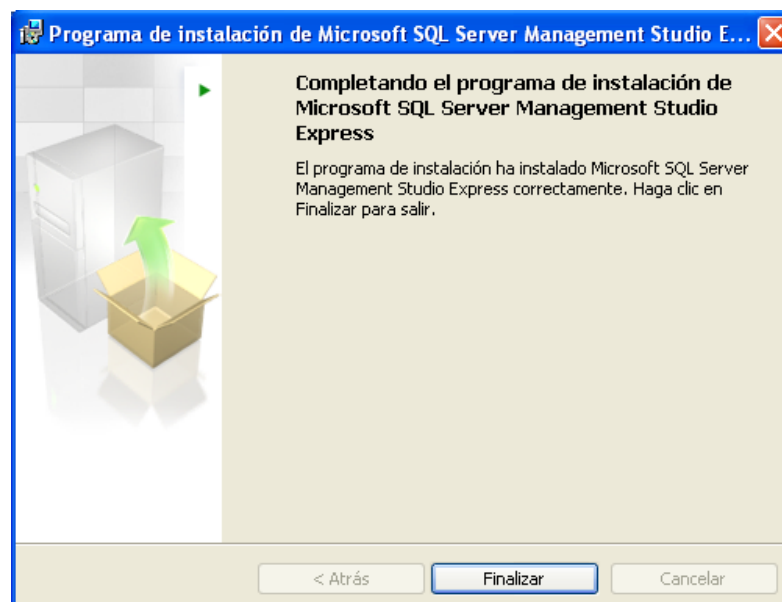


Figura 28: Damos clic en el boton Finalizar.

4.12. PRUEBA.

Dentro de las pruebas realizadas se tiene que decir que para realizar esto se contó con la ayuda de la propietaria, vendedores y clientes quienes nos dieron toda la información posible.

Cuando ya se terminó con la implementación del sistema, se tuvo que dar una capacitación de cómo se tenía que manipular dicho sistema a las personas del local, en las que no se encontró dificultad alguna.

4.13. MANTENIMIENTO.

Con la intervención de los investigadores de este trabajo, la propietaria de la despensa y los empleados se instaló el sistema y al mismo tiempo se dijo que se dará un mantenimiento cada 3 meses por un periodo de un año por cualquier motivo, ya sea por fallas en el código, por fallas de instalación o configuración del mismo; estando de acuerdo en lo manifestado por las autoridades quienes dijeron que la capacitación es lo más importante en este tipo de situaciones ya que los empleados de la Despensa no tienen conocimiento de cómo manejar Sistemas De Gestión De Venta E Inventario

Entonces se quedó, dar las capacitaciones debidas primero a los empleados y propietaria del local quienes serán las encargadas de manejar el sistema.

CAPÍTULO V:
MARCO PROPOSITIVO

5. MARCO PROPOSITIVO.

5.1. TEMA

Desarrollar un Sistema de Gestión de Ventas e Inventario en la Despensa "Santa Fe" Ubicado en el Cantón Caluma Provincia De Bolívar Utilizando Microsoft Visual Studio 2008 y SQL Server 2005 Con El Propósito De Mejorar La Atención A Los Clientes.

5.2. INTRODUCCIÓN

La realización de este trabajo de investigación propone buscar elementos tecnológicos y científicos que permitan favorecer las ventas y control de productos para de esta manera mejorar y fortalecer el crecimiento del mismo.

En este proyecto de Gestión De ventas, se plantea la solución ante este problema como es la falta de tecnología al momento, que el cliente desea comprar un producto que se encuentra en stock.

Los resultados que se obtendrán incidirán directamente al momento que el cliente compra su producto en la despensa "Santa Fe", ya que siempre estará supervisado por el administrador del local.

5.3. OBJETIVO DE LA PROPUESTA

5.3.1.1. OBJETIVO GENERAL

Desarrollar un Sistema de Gestion de Ventas e Inventario para hacer más eficiente las Ventas con el proposito de mejorar la atencion a los clientes de la Despensa "Santa Fe"

5.3.2. OBJETIVO ESPECÍFICOS

Realizar una investigación que permita formular mejoras en las Ventas de los producción de la Despensa "Santa Fe".

- Establecer estrategias de desarrollo de software que permitan incrementar el control de Inventarios en la Despensa "Santa Fe".

5.4. DESCRIPCIÓN DE LA PROPUESTA

Esta propuesta busca mejorar el proceso de Ventas E Inventrios, el mismo que ha sido desarrollado con la vision de implantar un sistema que ayude a sus necesidades que son la de tener al dia los movimientos ocurridos en la Despensa "Santa Fe" para que los clientes puedan cosultar y visualizar resultados de mejora y com mayor rapidez la optimización del funcionamiento de la aplicación del sistema.

5.4.1. METODOLOGÍA DE DESARROLLO

Una vez analizado los ciclos de vida del software hemos concluido que el modelo apropiado para la realización de nuestra tesis es en cascada.

Basándonos en la Ingeniería de software el desarrollo en cascada, es el enfoque metodológico que ordena rigurosamente las etapas del ciclo de vida del software, de forma tal que el inicio de cada etapa debe esperar a la finalización de la inmediatamente anterior.

Un ejemplo de una metodología de desarrollo en cascada es:

1. Análisis de requisitos.
2. Diseño del Sistema.
3. Codificación.
4. Pruebas.
5. Implantación.
6. Mantenimiento.

De esta forma, cualquier error de diseño detectado en la etapa de prueba conduce necesariamente al rediseño y nueva programación del código afectado, aumentando los costes del desarrollo.

5.4.2. ANÁLISIS PREVIO

Basado en la metodología de desarrollo seleccionada para elaborar sistemas, se realizó primero el respectivo análisis previo como requisito fundamental en el desarrollo del software, este análisis consta de los siguientes pasos:

- Se mantuvo varias sesiones de trabajo con el administrador de la Despensa "Santa Fe" de la Provincia de Bolívar.
- Por un tiempo prudente se analizó toda la información referente a la Despensa "Santa Fe" en la Provincia de Bolívar, para tener un mejor panorama sobre los procesos involucrados al sistema.
- Los avances sobre el desarrollo del sistema tuvimos que ir mostrándolos periódicamente para poder evaluar su rendimiento y corregir los errores posibles.

CONCLUSIONES

1. El desarrollo de este Sistema mejorara la calidad de información así como la calidad de servicio en la Despensa "Santa Fe".
2. Facilitara el almacenamiento de los datos de manera segura y rapida ya que se ha creado una clave de acceso al sistema por tanto no habrá perdida de información ni manipulación de datos por personas no autorizadas.
3. Satisface los requerimientos de los usuarios y a su vez garantiza una Exhibición clara y correcta de precios que serán detalladas en la Venta.
4. Para la utilizar este sistema informático no existe ningún obstáculo, ya que el sistema se ha hecho en base a los requerimientos del Administradora de la Despensa "Santa Fe" y además está diseñado como ellos manejan hoy en día manualmente.
5. Tiene como alto nivel de seguridad que no permita la alteración de datos por individuos externos.

RECOMENDACIONES

1. Siempre que se quiera construir o desarrollar un sistema tiene que estar asesorado por los especialistas en el tema caso contrario no podrá tener un resultado óptimo.
2. Seguir los lineamientos del ciclo de vida para el desarrollo de software hace la tarea más sencilla en la construcción de este tipo de sistemas.
3. Capacitación adecuada y oportuna al personal que utilizará la aplicación.
4. Dar mantenimiento continuo cada 3 meses al sistema.
5. Respaldo semanal a la base de datos

BIBLIOGRAFÍA.

- a) Cristian Sánchez Flores. Desarrollando Aplicaciones Visual C#.Net 2008. 1ra Edición. Lima, Pero, Maro. 431.
- b) Suarez F. 2003. Aplicaciones en Visual Basic 6.0. 1ra Edición. Lima, Perú, Ritisa Graff. 595.
- c) Desarrollo de aplicaciones accesibles:
- d) Visual Basic - Accessibility in Visual Basic - Ch16 Tutorials
- e) Microsoft Developer Network (msdn)
- f) http://es.wikipedia.org/wiki/Visual_Studio_2008#Visual_Studio_2008
- g) <http://www.es.scribd.com/doc/916042/Visual-Studio-2008>
- h) <http://www.vacationinnicaragua.com/microsoft/visual-basic/visual-basic-2008.pdf>
- i) <http://antares.sip.ucm.es/~luis/accesibilidadWEB08-09/aplicaciones.pdf>
- j) http://www.brainbell.com/tutors/Visual_Basic/ch16.htm
- k) <http://msdn.microsoft.com/>
- l) <http://translate.google.com.ec/translate?hl=es&langpair=en|es&u=http://www.serverwatch.com/trends/article.php/3716276/10-Cooler-Features-in-Windows-Server-2008.htm>

Aneiros

ANEXO 1

Las preguntas realizadas en la encuesta se redactan a continuación:

1.-) ¿Usted Se Siente Satisfecho Con La Atención Que Se Brinda Al Cliente En La Despensa Santa Fe Del Cantón Caluma Provincia Bolívar?

Sí No

2.-) ¿Cómo Califica Usted La Calidad De Los Productos Que Ofrece La Despensa Santa Fe?

e) Excelente
f) Muy Bueno
g) Bueno
h) Regular

3.) ¿Cuándo Ha Realizado Sus Compras En La Despensa Santa Fe Le Ha sucedido Alguno De Estos Casos?

e) Ha tenido que esperar mucho para cancelar su compra.
f) Los productos que ha comprado han estado defectuosos.
g) La atención del personal no fue apropiado.
h) No ha tenido ninguno de los inconvenientes anteriores.

4.-) ¿Al momento de escoger los productos que se encuentran en las perchas, ¿Encuentra información necesaria de los productos con sus respectivos precios y fecha de caducidad?

Sí No

5.-) ¿Piensa Usted Que Se Debería Implantar Sistema Información En La Despensa Santa Fe Para Agilizar El Proceso De Venta De Productos?

Sí No