

AGRADECIMIENTO

Mi gratitud, principalmente está dirigida a Dios por haberme dado la existencia y permitido llegar al final de la carrera.

A los docentes que me han acompañado durante este largo camino, brindándome siempre su orientación con profesionalismo ético en la adquisición de conocimientos.

Igualmente a nuestro director y lector de tesis quien nos ha orientado en todo momento en la realización de este proyecto.

Stefany Gabriela Cruz Pizco

AGRADECIMIENTO

Agradezco a Dios por haberme guiado e iluminado mi camino hasta ahora, a cada uno de los que son parte de mi familia mi PADRE, a mi MADRE, a mis hermanos por siempre haberme dado su fuerza y apoyo incondicional que me han ayudado y llevado hasta donde estoy ahora. Y a mí director y lector de tesis quién nos ayudó en todo momento.

Juana Maribel Mullo Ochoa

DEDICATORIA

A Dios por ofrecerme la oportunidad y la dicha de la vida, al brindarme los medios necesarios para continuar mis estudios, y siendo un apoyo incondicional para lograrlo ya que sin él no hubiera podido.

A mis padres que contribuyeron incondicionalmente para lograr mis metas y objetivos propuestos y brindarme con su ejemplo a ser perseverante y darme la fuerza que me impulsó a conseguirlo.

Stefany Gabriela Cruz Pizco

DEDICATORIA

Dedico este presente trabajo a mis padres hermanos, por que han estado conmigo a cada paso que doy, fue lo que me hizo ir hasta el final va por ustedes. Por qué admiro su fortaleza y por lo que han hecho de mí, siendo mi apoyo en todo momento, y en cada triunfo de mi vida.

Juana Maribel Mullo Ochoa

CERTIFICACION DE AUTORIA

Yo, Stefany Gabriela Cruz Pizco; soy responsable del contenido, ideas expuestas en esta Tesis de Grado, que me corresponde exclusivamente; y el patrimonio intelectual de la misma a la Escuela de Sistemas de la Facultad de Administración, Finanzas e Informática de la Universidad Técnica de Babahoyo”

Stefany Gabriela Cruz Pizco

CERTIFICACION DE AUTORIA

Yo, Juana Maribel Mullo Ochoa; soy responsable del contenido, ideas expuestas en esta Tesis de Grado, que me corresponde exclusivamente; y el patrimonio intelectual de la misma a la Escuela de Sistemas de la Facultad de Administración, Finanzas e Informática de la Universidad Técnica de Babahoyo”

Juana Maribel Mullo Ochoa

FIRMAS DE RESPONSABILIDAD

DECANO

Lcdo. Teodoro Flores Carpio _____

DIRECTOR DE ESCUELA

Lcdo. Washington Ávila Ortega _____

DIRECTOR DE TESIS

Ing. Fernando Mejía Peñafiel _____

LECTOR DE TESIS

Ing. Raúl Ramos Morocho _____

MIEMBROS DEL TRIBUNAL

Nombres	Nota	Firma
Ing. _____	_____	_____
Ing. _____	_____	_____
Ing. _____	_____	_____

ÍNDICE GENERAL

Portada	
Agradecimientos	
Dedicatorias	
Certificación De Autoría	
Firmas de Responsabilidad	

CAPITULO I

1. Problema	15
1.1. Planteamiento del Problema	15
1.2. Formulación del Problema	16
1.3. Delimitación	16
2. Objetivos	16
2.1. Objetivo General	16
2.2. Objetivo Especifico	16
3. Justificación	17
3.1. Justificación	17

CAPITULO II

2.1. Fundamentación Teórica	18
2.2. Comisariato El Popular	18
2.2.1. Historia	18
2.2.2. Características	19
2.2.3. Misión	19
2.2.4. Visión	19

2.2.5. Ubicación	19
2.2.6. Personal	19
2.3. Metodologías de Desarrollo del Software	20
2.3.1. Metodología Orientada a Objetos	20
2.3.2. Metodología Estructurada	22
2.3.3. Estructura Secuencial	24
2.3.4. Estructura Selectiva o de Selección	25
2.3.5. Estructura Iterativa	26
2.4. Ciclo de Vida del Desarrollo del Software	26
2.5. Modelos de Ciclo de Vida del Software	28
2.5.1. Ciclo de Vida de Cascada	28
2.5.2. Ciclo de Vida en V	29
2.5.3. Ciclo de Vida Sashimi	29
2.5.4. Ciclo de Vida en Cascada con Sub proyectos	30
2.5.5. Ciclo de Vida en Cascada Incremental	30
2.5.6. Ciclo de Vida en Cascada con Reducción de Riesgos	31
2.5.7. Ciclo de Vida en Espiral	31
2.6. Visual .Net	32
2.6.1. Introducción de Visual .NET	33
2.6.2. .NET Framework	33
2.6.3. Requisitos Hardware	34
2.6.4. Sistema Operativo	34
2.6.5. Recomendaciones Previas	35
2.6.6. Diseño del Formulario	35
2.6.7. Controles Funciones y Eventos	35

2.6.8. Ámbito de las Variables	37
2.6.9. Cuadros de Diálogo	41
2.6.10. Tipos de Datos	43
2.6.11. Operadores	44
2.6.12. Sentencias de Control	46
2.6.13. Ventana de Comando	48
2.6.14. Ventajas y Desventajas	49
2.7. Bases de Datos	50
2.7.1. Tipos de Bases de Datos	50
2.7.2. SQL Server 2005	51
2.7.3. Especificaciones	52
2.7.4. Operaciones con Conjuntos	52
2.7.5. Tipos de Datos	54
2.7.6. Variables	56
2.7.7. Estructuras de Control	56
2.7.8. Sentencias en SQL Server	59
2.7.9. Clausulas en SQL Server	61
2.8. Hipótesis y Variables	62
2.8.1. Hipótesis	62
2.8.2. Variables	62

CAPITULO III

3.1. Marco Metodológico	63
3.2. Modalidad de la Investigación	63
3.3. Tipo de Investigación	64
3.4. Población y muestra de la Investigación	64
3.5. Métodos Técnicas e Instrumentos de la Investigación	65
3.6. Interpretación de Resultados	67
3.7. Conclusiones	77
3.8. Recomendaciones	77

CAPITULO IV

4. Desarrollo Técnico de la Investigación	78
4.1. Introducción	78
4.2. Objetivo de la Propuesta	78
4.2.2. Objetivos General	78
4.2.1. Objetivo Específicos	78
4.3. Metodología de Desarrollo Utilizada	79
4.4. Análisis Previo	79
4.4.1. Listado de Requisitos y funciones que tendrá el Software	79
4.5. Diseño	80
4.5.1. Base de Datos	81
4.5.1.1. Modelo Conceptual	81
4.5.1.2. Modelo Físico – Modelo Entidad Relación	82
4.5.1.3. Diccionario de Datos	82
4.5.1.4. Script de Base de Datos	85

4.6. Diagrama de Caso de Uso	86
4.7. Diagrama de Secuencia	88
4.8. Diagrama de Actividad	89
4.9. Diagrama de Despliegue	90
4.10. Diseño de Interfaces	91
4.11. Diseño de salidas	95
4.12. Desarrollo	98
4.12.1. Pruebas	99
4.12.2. Implementación del sistema	100
4.12.2.1. Requerimientos de hardware	107
4.12.2.2. Requerimientos de software	107
4.12.2.3. Proceso de instalación	107
4.12.2.4. Seguridades	107
4.13. Conclusiones y Recomendaciones para una eficiente implementación del software	109
Bibliografía	110
Linografías	111
Anexos	112

ÍNDICE DE GRAFICAS Y TABLAS

Fig. 2.5.1.1: Ciclo de vida en cascada	28
Fig. 2.5.2.1: Ciclo de vida en V	29
Fig. 2.5.3.1: Ciclo de vida Sashimi	30
Fig. 2.5.5.1: Cascada incremental	31
Fig. 2.5.7.1: Ciclo de vida en espiral	32
Tabla 2.6.8.1: Ámbito de las Variables	38
Tabla 2.6.11.1: Operadores	44
Tabla 2.6.11.2: Operadores Abreviados	45
Tabla 2.6.11.3: Precedencia de Operadores	45
Tabla 2.6.11.4: Operadores Lógicos	46
Fig. 3.5.1: Grafico de la Primera Pregunta	67
Fig. 3.5.2: Grafico de la Segunda Pregunta	68
Fig. 3.5.3: Grafico de la Tercera Pregunta	69
Fig. 3.5.4: Grafico de la Cuarta Pregunta	70
Fig. 3.5.5: Grafico de la Quinta Pregunta	71
Fig. 3.5.6: Grafico de la Sexta Pregunta	72
Fig. 3.5.7: Grafico de la Séptima Pregunta	73
Fig. 3.5.8: Grafico de la Octava Pregunta	74
Fig. 3.5.9: Grafico de la Novena Pregunta	75
Fig. 3.5.10: Grafico de la Décima Pregunta	76

Tabla. 4.5.1.3.1: Tabla Proveedor	82
Tabla. 4.5.1.3.2: Tabla Compras	83
Tabla. 4.5.1.3.3: Tabla Producto	83
Tabla. 4.5.1.3.4: Tabla Salida	84
Tabla. 4.5.1.3.5: Tabla Compras Ingresos	84
Tabla. 4.5.1.3.6: Tabla Local	84
Fig. 4.10.1: Ventana de Presentación	91
Fig. 4.10.2: Ventana Inicio de Sesión	92
Fig. 4.10.3: Ventana Menú de Usuario	92
Fig. 4.10.4: Opciones del Menú / Ingresos	93
Fig. 4.10.5: Opciones del Menú /Consultas	93
Fig. 4.10.6: Opciones del Menú / Administrador	94
Fig. 4.10.7: Opciones del Menú / Salir	94
Fig. 4.11.1: Ejemplo de Consulta / Sucursal	95
Fig. 4.11.2: Ejemplo de Reportes / Sucursal	95
Fig. 4.11.3: Ejemplo de Consulta / Proveedor	96
Fig. 4.11.4: Ejemplo de Reportes / Proveedor	96
Fig. 4.11.5: Ejemplo de Consulta / Productos	97
Fig. 4.11.6: Ejemplo de Reportes / Productos	97

1. PROBLEMA

1.1. PLANTEAMIENTO DEL PROBLEMA

En la actualidad el comisariato El “Popular” no posee un sistema Administrador de Inventario lo que significa que los procesos de ingresos de mercadería y los reportes de bodega son realizados de forma manual.

Debido a que la demanda de clientes va en aumento, se ha hecho muy difícil, manejar de manera confiable la información del negocio, teniendo como resultado mucha insatisfacción en los mismos, falta de información y de procedimientos adecuados para llevar el control de mercadería.

Los problemas existenciales en el comisariato El “Popular”, han surgido debido a que los inventarios de los productos siempre se ha manejado de una manera lenta, y siempre ha sido la secretaria quien lleva la labor de desempeñar dicho trabajo, existe pérdida de tiempo al momento de consultar los datos de todos sus productos debido que no cuenta con las herramientas necesarias que facilite dicha tarea.

El comisariato no posee un archivo general de todos sus productos, pero ya que no cuenta con la utilización de un sistema actual se ha detectado pérdida de información de los registros de control de inventario e informes de administración.

Todas estas dificultades mencionadas anteriormente, provocaban la imposibilidad de realizar análisis sobre la rotación de productos para comprar las cantidades correctas como para su disposición física en bodega.

Ante esta necesidad de mejorar los procesos de transacciones de la empresa, para volverlo eficiente y competente ante otras empresas, se considera pertinente elaborar una solución informática que garantice la seguridad de los servicios y la confiabilidad de los datos.

1.2. FORMULACIÓN DEL PROBLEMA

¿Cómo mejorar los procesos de información de datos a través de un Sistema Administrador de Bodega para el Control de Inventario en el Comisariato El Popular?

1.3. DELIMITACIÓN

El desarrollo de este sistema se realizará en el comisariato El “Popular” ubicado en las calles 27 de Mayo y Venezuela del cantón Montalvo.

Este sistema se llevará a cabo en aproximadamente 6 meses.

Objeto de Estudio: Ingeniería en Sistemas.

Campo de acción: Los Sistemas Informáticos.

2. OBJETIVOS

2.1. Objetivo General

Desarrollar un sistema informático para mejorar los procesos de administración de bodega para el control de inventario del comisariato el “Popular” del Cantón Montalvo.

2.2. Objetivos Específicos

- Controlar los ingresos y salidas de los productos de bodega.
- Recocer, manejar y adquirir los conceptos técnicos para una adecuada administración de bodega y control de inventario.
- Agilizar todos los procesos, permitiendo manejar rápidamente la información.

3. JUSTIFICACIÓN

Este proyecto permitirá disponer de información cierta y oportuna para obtener de forma rápida, segura y detallada que en un momento dado requiere el administrador de la empresa reduciendo el tiempo en cada uno de los procesos que se realice, la administración de bodega y el control de inventario será eficiente ya que se podrá tener actualizado el stock de los productos, permitirá fijar un stock mínimo y se podrá anticipar debidamente los productos que están por terminarse para pedir a los proveedores respectivos.

El proyecto es de fundamental importancia por tener entre sus objetivos mejorar el funcionamiento de bodega donde se almacena la mercadería, en el cual se pretende hacer una propuesta viable, necesitando de una herramienta donde se pueda llevar un control de todos los proveedores, como también la rotación de productos, por esta razón se desarrollara una aplicación informática, para lograr una observación absoluta de los productos que más se necesitan ya sea semanal o periódicamente, al contar con un registro confiable y seguro.

Esta aplicación permitirá tener un control sobre los productos que estén por caducarse, logrando de esta forma impedir la caducidad de los mismos.

Mediante la creación de este sistema podremos efectuar con mayor claridad y eficiencia una labor de inventario, conjuntamente con los complementos que se aplicarán como: SQL server y Visual.NET.

2.1 ANTECEDENTES DE LA INVESTIGACIÓN

En esta investigación no le precede proyecto similar, al momento de investigar en diferentes archivos de esta empresa, este trabajo surge debido a un estudio efectuado del personal interno del comisariato el popular.

Después de haber hecho un análisis exhaustivo se notó que se debe tener control estricto en los archivos correspondientes, a los niveles de stock de los diversos tipos de recursos que utiliza la empresa, es prácticamente fundamental para garantizar la fluidez del sistema.

Por consiguiente a la empresa le interesa minimizar el tamaño de los inventarios y optimizar su gestión en cuanto a los procesos de ingresos, búsqueda y elaboración de informes de manera rápida ordenada a través de este sistema, de tal manera los datos estarán respaldados y organizados sin importar el aumento o disminución del mismo este sistema ofrece facilidades para manipular de manera fácil y rápida para el usuario.

Tomando en cuenta la necesidad de la empresa se desarrollará este sistema, el cual mostrará toda su potencialidad a través de los resultados que se obtenga al mejorar el desempeño administrativo en cuanto al proceso del control de inventarios, los beneficios directos de esta investigación para los propietarios de la empresa, los cuales podrán observar la rapidez al momento de dar la información, además lograrán promover el desempeño de los usuarios.

2.2. COMISARIATO EL POPULAR

2.2.1. HISTORIA

En el año 1989 el Sr. Julio Manzano Carrasco, tuvo la idea de abrir una microempresa en la Ciudad de Montalvo en las calles 27 de Mayo y Venezuela, como vendedor y propietario con el nombre “Hnos. Manzanos” con el apoyo de su esposa la Sr. Ana Paula Carrasco Bozada como auxiliar de venta, e hijos.

A través del tiempo la microempresa fue progresando y en la cual sus demandas en sus ventas fue aumentando y los gastos fueron muchos más grandes, por lo que el Sr. Julio Manzano, hizo por primera vez un préstamo en una empresa Bancaria, para ampliar su negocio.

Al comienzo del año 1996 la microempresa fue progresando y obteniendo muchas ganancias y en la actualidad podemos conocer la evolución de un gran comisariato “El popular” que así tiene el nombre actualmente en el que se ha ido modernizando y también seguirá por el mismo sendero, tratando siempre de ser el comisariato más importante de la ciudad.

2.2.2. CARACTERÍSTICAS

Esta empresa se caracteriza por ser comercial, competitiva y eficiente ante los otros comisariatos, brindando así confiabilidad y seguridad a todos sus clientes.

2.2.3. MISIÓN

La misión de esta empresa es la de proporcionar bienestar a nuestros clientes, suministrándoles soluciones a sus necesidades y servicios de conveniencias a precios accesibles, en un ambiente cálido, amplio, cómodo y moderno, contribuyendo de esta manera a mejorar su calidad de vida.

2.2.4. VISIÓN

Nuestra visión, es ser el comisariato más efectivo, más innovador de la ciudad, que cuenta con un personal capacitados para ofrecer una buena atención a los clientes, proporcionando productos diversos en calidad, marcas y presentaciones accesible a todos los niveles de la sociedad, logrando la distinción tanto de clientes como de proveedores y competidores.

2.2.5. UBICACIÓN

La empresa se encuentra ubicada en el centro de la ciudad y es apropiado para la venta de los productos de compra habitual y de compra impulsiva, ya que se orienta a la comodidad del consumidor. Proporciona una oferta variada tanto en productos como en precios.

2.2.6. PERSONAL

Lo más importante en una empresa es el personal competitivo, partiendo de esta premisa el desarrollo del personal no es una exigencia por cumplir de parte de todo empleador sino un requisito indispensable para el triunfo de toda organización.

Por esta razón en esta empresa el cliente no es lo primero, sino el empleado. Siempre y cuando el empleado este conforme con las labores que realiza, de esta manera hace propia la visión, misión y los objetivos de la empresa,

comprometiéndose con el éxito de ésta. Este razonamiento se amplía al nivel estratégico de la empresa porque el comportamiento y las decisiones de éstos, influyen sobre el estado de ánimo y el conocimiento del resto del personal.

2.3. METODOLOGÍAS DE DESARROLLO DEL SOFTWARE

Metodología de desarrollo del software es un modo sistemático de realizar gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito.

Existen muchas definiciones sobre lo que es una metodología de desarrollo del Software, la mayoría de ellas coincide en que dividen un proyecto en fases y las tareas a realizar en cada una, especificando cuales son las entradas y las salidas que producen.

Una metodología de desarrollo de software se refiere a un que es usado para estructurar, planear y controlar el proceso de desarrollo en sistemas de información a lo largo del tiempo, una gran cantidad de métodos han sido desarrollados diferenciándose por su fortaleza y debilidad.

2.3.1. METODOLOGÍA ORIENTADA A OBJETOS

La metodología orientada a objetos es una de las metodologías de análisis y diseño más maduras y eficientes que existen en la actualidad. La gran virtud que aporta esta metodología es su carácter directo, que le permite ser de dominio público y, en consecuencia sobrevivir con enorme vitalidad. Las fases que conforman a la metodología orientada a objetos son: Análisis, Diseño del Sistema, Diseño de Objetos, Implementación.

Análisis

El analista construye un modelo del dominio del problema, mostrando sus propiedades más importantes. El modelo de análisis es una abstracción resumida y precisa de lo que debe hacer el sistema deseado y no de la forma en que se hará. Los elementos del modelo deben ser conceptos del dominio de aplicación y no conceptos informáticos tales como estructuras de datos. Un

buen modelo debe poder ser entendido y criticado por expertos en el dominio del problema que no tengan conocimientos informáticos.

Diseño del sistema

El diseñador del sistema toma decisiones de alto nivel sobre la arquitectura del mismo. Durante esta fase el sistema se organiza en subsistemas basándose tanto en la estructura del análisis como en la arquitectura propuesta y se selecciona una estrategia para afrontar el problema.

Diseño de objetos

El creador de objetos construye un modelo de diseño basándose en el modelo de análisis, pero incorporando detalles de implementación. El diseño de objetos se centra en las estructuras de datos y algoritmos que son necesarios para implementar cada clase. La metodología orientada a objetos, describe la forma en que el diseño puede ser implementado en distintos lenguajes orientados y no orientados a objetos.¹

Implementación

Durante la fase de implementación es importante tener en cuenta los principios de la ingeniería del software de forma que la correspondencia con el diseño sea directa y el sistema implementado sea flexible y extensible. Las clases de objetos y relaciones desarrolladas durante el análisis de objetos se traducen finalmente a una implementación concreta.

La metodología orientada a objetos emplea tres clases de modelos para describir el sistema:

➤ **Modelo de Objetos.**- Se refiere al arreglo estática de los objetos del sistema, identidad, relaciones con otros objetos, atributos y operaciones. El modelo de objetos proporciona el entorno esencial en el cual se pueden situar

¹ <http://www.monografias.com/trabajos13/metomt/metomt.shtml>/Víctor Manuel Chávez Gaona

el modelo dinámico y el modelo funcional. El objetivo es capturar aquellos conceptos del mundo real que sean importantes para la aplicación.

➤ **Modelo Dinámico.**- Representa los aspectos de un sistema que tratan de la temporización y secuencia de operaciones como sucesos que marcan los cambios, secuencias de hechos, estados que definen el contexto para los acontecimientos y la organización de sucesos y estados.

➤ **Modelo Funcional.**- Describe las transformaciones de valores de datos tales como funciones, correspondencias, restricciones y dependencias funcionales que ocurren dentro del sistema. Captura lo que hace el sistema, independientemente de cuándo se haga o de la forma en que se haga. Se representa mediante diagramas de flujo de datos.²

2.3.2. METODOLOGÍA ESTRUCTURADA

Tiene como objetivo emplear las metodologías de análisis y diseño estructurado para su uso, incrementando la productividad en el desarrollo e implantación de los sistemas de información. Crea los modelos de forma descendente, son las orientadas a procesos, a datos y las mixtas. Intentan aplicar formas ingenieriles para solucionar problemas técnicos al obtener un sistema de información, proponen la creación de modelos, flujos y estructuras aproximándose al problema.

Está orientada a procesos, es decirse centra en especificar y descomponer la funcionalidad del sistema. Se utilizan varias herramientas como los diagramas de flujo de datos que representan la forma en la que los datos se mueven y se transforman, esto incluye a los procesos, flujos de datos y almacenes de datos.

Los procesos individuales se pueden a su vez descomponer en otros diagramas de flujos de datos de nivel superior.

²<http://es.scribd.com/doc/73772826/7/Modelos-de-AOO-y-DOO>

Especificaciones de Procesos.- Es lo que se escribe para uno de los procesos definidos en el diagrama de flujo de datos cuando no se puede descomponer más.

Diccionario de Datos.- Son los nombres de todos, los tipos de datos y almacenes de datos junto con sus definiciones.

Diagramas de Transición de Estados.- Modelan procesos que dependen del tiempo Diagramas entidad – relación. Los elementos del modelo entrada y salida se corresponden con almacenes de datos en el diagrama de flujo de datos.

Ventajas

- Los programas son más fáciles de entender, ya que pueden ser leído de forma secuencial, sin necesidad de hacer seguimiento a saltos de línea GOTO dentro de los bloques de código para entender la lógica.
- La estructura del programa es clara puesto que las instrucciones están más ligadas o relacionadas entre sí.
- Reducción de los costos de mantenimiento de los programas.

Programas más sencillos y más rápidos, ya que es más fácil su optimización.

Desventajas

El principal inconveniente es que se obtiene un único bloque de programa, que cuando se hace demasiado grande puede resultar problemático su manejo, esto se resuelve empleando la programación modular, definiendo módulos interdependientes programados y compilados por separado. Un método un poco más sofisticado es la programación por capas.

2.3.2.1. METODOLOGÍA ESTRUCTURADA MIXTA

Esta metodología estructura mixta estudian los sistemas desde varios puntos de vista. La metodología orientada a objetos el modelado del sistema se concibe

como conjunto de objetos que interactúan entre sí y se busca el enfoque unificador de los objetos.

Requisito.- Condiciones que debe tener un sistema para satisfacer un contrato, una norma o especificación y condición que un usuario necesita para resolver un problema.

Especificación.- Es un documento que define los requisitos, el diseño y el comportamiento u otras características de un sistema.

Análisis da Viabilidad.- Los aspectos a tener en cuenta son: económicos que determinan si el proyecto va a ser rentable, técnico estudia si la funcionalidad y los factores marcados del proyecto son factibles, legales se discute si los requisitos atentan contra la ley y operativo es el que determina si el sistema se puede implantar de forma efectiva en la empresa.³

2.3.3. ESTRUCTURA SECUENCIAL

Una estructura secuencial es la más sencilla ya que es aquella en que la instrucción sigue a otra en secuencia y también se la conoce como estructura lineal. Se compone de instrucciones que deben ejecutarse en forma consecutiva de tal modo que la salida de una es la entrada de otra y así sucesivamente hasta el final del proceso, siguiendo una línea ya que tiene una entrada y una salida.

Solamente los problemas muy sencillos pueden resolverse haciendo uso únicamente de esta estructura. Normalmente, la estructura secuencial hace parte de soluciones a problemas complejos en las que se la utiliza mezclada con estructuras iterativas repetir varias veces un conjunto de instrucciones y condicionales tomar decisiones.

Una estructura de control secuencial puede contener cualquiera de las siguientes instrucciones:

- Declaración variable
- Asignación de valores

³<http://www.xuletas.es/ficha/metodologias-estructuradas/>

- Entrada de datos
- Procesamiento de datos o llamados también operaciones
- Reporte de resultados

Un problema se puede dividir en acciones elementales o instrucciones, usando un número limitado de estructura de control básico y su combinación que pueden servir para resolver un problema con el fin de aumentar la eficacia de la programación y el mantenimiento de los programas, haciendo que los mismos sean más fiables, adaptables, manejables, fácilmente comprensibles y transportables.

Por ejemplo:

1. inicio
2. leer n
3. proceso sumar $n=n+1$
4. mostrar n
5. fin

Para escribir un algoritmo con estructura secuencial, es necesario determinar el método que se va a utilizar. Los más comunes son los flujogramas conocidos también como diagrama de flujo.

2.3.4. ESTRUCTURA SELECTIVA O DE SELECCIÓN

Como su mismo nombre lo indica, este tipo de estructuras realizan una selección de las acciones o sentencias a ejecutar, es decir dependiendo de si se cumple o no una determinada condición, si se ejecutan o no, un en determinado grupo de sentencias. Es importante mencionar que la condición que rige el control del flujo puede ser tan elaborada como la naturaleza del problema lo requiera, pero se deberá tener la garantía de que el valor final de la condición podrá ser evaluado como un valor booleano, esto es el elementos de la programación estructurada; algoritmos, pesado código y diagramas de flujo

como verdadero o falso, ya que de lo contrario la condición tendrá ambigüedad. Las estructuras de selección pueden ser simples, dobles o múltiples. Los flujos que se siguen, dependiendo de la evaluación de la condición, pueden visualizarse más fácilmente al observar sus representaciones en diagrama de flujo.⁴

2.3.5. ESTRUCTURA ITERATIVA

La estructura iterativa o de repetición permite ejecutar una o varias instrucciones, un número determinado de veces o indefinidamente, mientras se cumpla una condición. Esta estructura ayuda a simplificar los algoritmos, ahorrando tiempo valioso a quien resuelve problemas con ayuda de la computadora. En programación existen al menos dos tipos de estructuras repetitivas, las cuales a su vez tienen variantes en los diferentes lenguajes de programación.

La característica común es que ambos tipos permiten ejecutar una o varias instrucciones:

- Un número determinado de veces
- Mientras se cumpla una condición

El primer tipo de estructura iterativa o repetitiva, permite ejecutar una o varias instrucciones en un número determinado de veces. Esto consigue resolver una buena cantidad de problemas cuya solución requiera de estas estructuras en forma combinada.⁵

2.4. CICLO DE VIDA DEL DESARROLLO DEL SOFTWARE

El ciclo de vida de desarrollo de software requieren de un tiempo y esfuerzo considerable para su proceso y deben permanecer en uso por un periodo mucho mayor.

⁴<http://www.di.uniovi.es/~dani/asignaturas/apuntes-leccion4.PDF>

⁵http://www.atc.us.es/asignaturas/fi/curso_de_c/Estructuras_iterativas.html

Durante este tiempo de desarrollo y uso, desde que se detecta la necesidad de construir un sistema de software hasta que este es retirado, se identifican varias etapas que en conjunto se denominan el ciclo de vida del software y en cada caso, en función de cuales sean las características del proyecto, se configurará el ciclo de vida de forma diferente. Usualmente se consideran las etapas como especificación y análisis de requisitos, diseño del sistema, implementación del software, aplicación y pruebas, entrega y mantenimiento.

Un aspecto esencial dentro de las tareas del desarrollo del software es la documentación de todos los elementos y especificaciones en cada fase. Dado que esta tarea siempre estará influida por la fase del desarrollo en curso, se explicará de forma distribuida a lo largo de las diferentes fases como un apartado especial para recalcar su importancia en el conjunto del desarrollo del software.

Tal como ya hemos mencionado, las etapas principales a realizar en cualquier ciclo de vida son:

Análisis de requisitos.-Construye un modelo de los requisitos.

Diseño.- A partir del modelo de análisis se deducen las estructuras de datos, la estructura en la que descompone el sistema y la interfaz de usuario.

Codificación.- Construye el sistema. La salida de esta fase es código ejecutable.

Pruebas.- Se comprueba que se cumplen criterios de corrección y calidad.

Mantenimiento.- En esta fase, que tiene lugar después de la entrega se asegura que el sistema siga funcionando y adaptándose a nuevos requisitos.

Las formas de organizar y estructurar la secuencia de ejecución de las tareas en las diferentes fases de cada uno de los métodos pueden dar lugar a un tipo de ciclo de vida diferente.⁶

⁶ <http://es.wikipedia.org/wiki/Software>

2.5. MODELOS DE CICLO DE VIDA DEL SOFTWARE

2.5.1. CICLO DE VIDA EN CASCADA

Este modelo permite la posibilidad de hacer iteraciones, es decir, durante las modificaciones que se hacen en el mantenimiento se puede ver por ejemplo la necesidad de cambiar algo en el diseño, lo cual significa que se harán los cambios necesarios en la codificación y se tendrán que realizar de nuevo las pruebas, es decir si se tiene que volver a una de las etapas anteriores al mantenimiento hay que recorrer de nuevo el resto de las etapas.

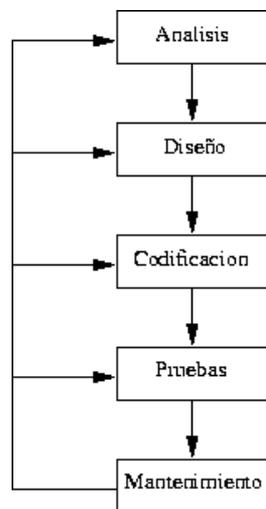


Fig. 2.5.1.1: Ciclo de vida en cascada

Ventajas

- La planificación es sencilla.
- La calidad del producto resultante es alta.
- Permite trabajar con personal poco calificado.

Desventajas

- Si se han cometido errores en una fase es difícil volver atrás.
- No se tienen indicadores fiables del progreso del trabajo.
- Es comparativamente más lento que los demás y el valor es superior.

2.5.2. CICLO DE VIDA EN V

Este ciclo tiene las mismas fases que el anterior pero se imagina el nivel de abstracción de cada una de las fases además de utilizarse como entrada para la siguiente, sirve para validar o verificar otras fases posteriores. Su estructura está representada en la siguiente figura.

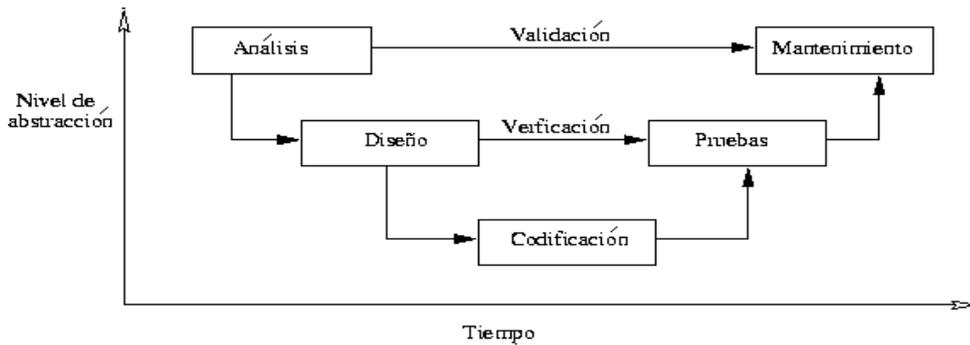


Fig. 2.5.2.1: Ciclo de vida en V

2.5.3. CICLO DE VIDA TIPO SASHIMI

Según el modelo en cascada una fase solo puede empezar cuando ha terminado la anterior. En este caso sin embargo, se permite un solapamiento entre fases. Por ejemplo, sin tener terminado del todo el diseño se comienza a implementar.

Una ventaja de este modelo es que no necesita generar tanta documentación como el ciclo de vida en cascada debido a la continuidad del mismo personal entre fases. Los problemas planteados son:

- Es aún más difícil controlar el progreso del proyecto debido a que los finales de fase ya no son un punto de referencia claro.
- Al hacer cosas en paralelo si hay problemas de comunicación pueden surgir inconsistencias.

El diseño arquitectónico es el de alto nivel, el detallado el de bajo nivel. En la figura se ha representado la estructura del ciclo de vida Sashimi.

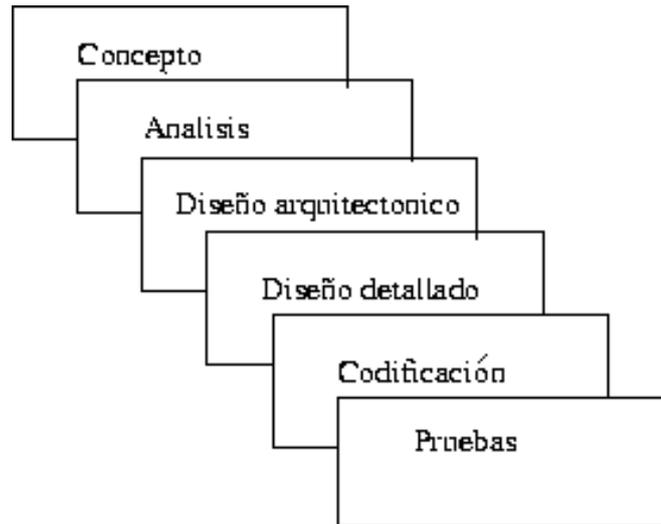


Fig. 2.5.3.1: Ciclo de vida Sashimi

2.5.4. CICLO DE VIDA EN CASCADA CON SUBPROYECTOS

Si una vez que se ha llegado al diseño arquitectónico, se comprueba que el sistema se divide en varios subsistemas independientes entre sí, sería razonable suponer que a partir de ese punto cada uno se puede desarrollar por separado y en consecuencia en paralelo con los demás. Cada uno tendrá seguramente fechas de terminación distintas. Una vez que han terminado todos se integran y se prueba el sistema en su conjunto.

La ventaja es que se puede tener a más gente trabajando en paralelo de forma eficiente.⁷

2.5.5. CICLO DE VIDA EN CASCADA INCREMENTAL

En este ciclo se va implementando el sistema y añadiendo pequeñas funcionalidades. Cada uno de los pequeños incrementos es parecido a lo que ocurre dentro de la fase de mantenimiento.

La ventaja de este método es que no es necesario tener todos los requisitos en un principio y el inconveniente es que los errores en la detección de requisitos se encuentran tarde.

⁷[http://ciclodevidasoftware.wikispaces.com/CICLO+DE+VIDA+INCREMENTAL/Manuel Arias](http://ciclodevidasoftware.wikispaces.com/CICLO+DE+VIDA+INCREMENTAL/Manuel+Arias)

Hay dos partes en el ciclo de vida, similares al anterior. Por un lado está el análisis y el diseño global y por otra parte están los pequeños incrementos, con las fases de diseño detallado, codificación y mantenimiento. En la siguiente figura 2.2.3.7.5.1 se puede ver su estructura.⁸

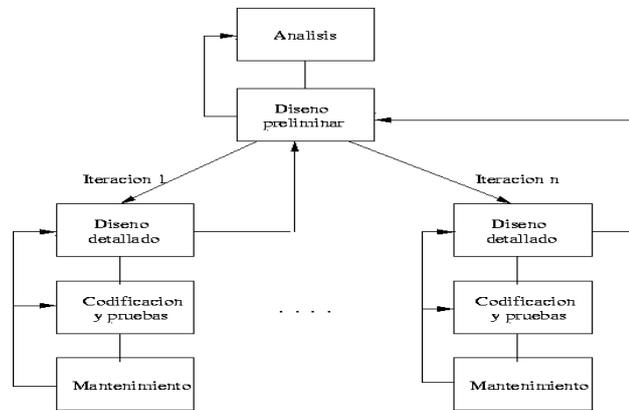


Fig. 2.5.5.1: Cascada incremental

2.5.6. CICLO DE VIDA EN CASCADA CON REDUCCIÓN DE RIESGOS

Uno de los problemas del ciclo de vida en cascada es que si se entienden mal los requisitos esto sólo se descubrirá cuando se entregue el producto. Para evitar este problema se puede hacer un desarrollo iterativo durante las fases de análisis y diseño global. Esto consistiría en:

1. Preguntar al usuario.
2. Hacer el diseño global que se desprende del punto 1.
3. Hacer un prototipo de interfaz de usuario, entrevistas con los usuarios y volver con ello al punto uno para identificar más requisitos o corregir malentendidos.

2.5.7. CICLO DE VIDA EN ESPIRAL

Consiste en una serie de ciclos que se repiten. Cada uno tiene las mismas fases y cuando termina da un producto ampliado con respecto al ciclo anterior. En

⁸ <http://www.ia.uned.es/ia/asignaturas/adms/GuiaDidADMS/node10.html>/José R. Álvarez

este sentido es parecido al modelo incremental, la diferencia importante es que tiene en cuenta el concepto de riesgo. Un riesgo puede ser muchas cosas, requisitos no comprendidos, mal diseño, errores en la implementación.⁹

Una representación típica de este modelo se muestra en la siguiente figura.

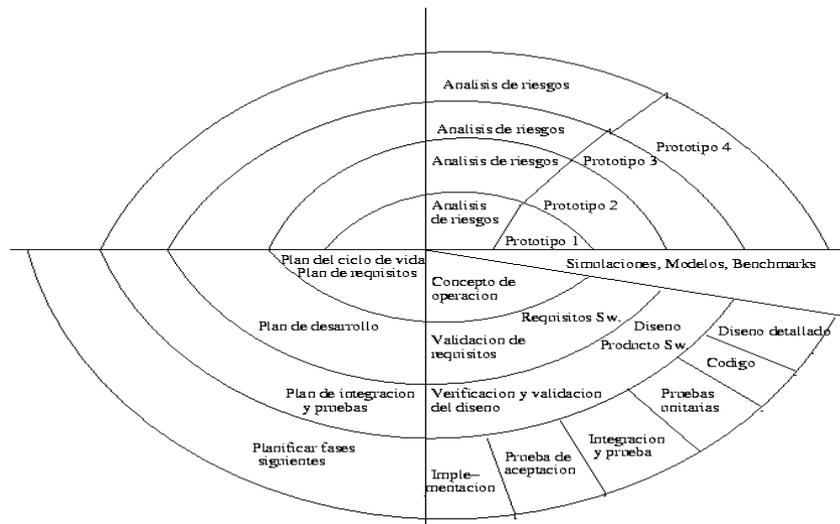


Fig. 2.5.7.1 Ciclo de vida en espiral

2.6. VISUAL .NET

Es la nueva versión de la familia de herramientas de desarrollo de software de Microsoft, naturalmente orientadas hacia su nuevo entorno de programación .NET Framework.

Si bien es posible la escritura de programas empleando en .NET Framework, este al estar compuesto de herramientas independientes, constituye un medio más incómodo de trabajo. Programación con Visual Basic .NET. Al tratarse de un entorno de desarrollo integrado como compiladores, editores y ayuda, facilitando en gran medida la creación de programas. Por este motivo, todas las explicaciones y ejemplos desarrollados a lo largo de este texto se harán basándose en este entorno de programación.

⁹<http://www.slideshare.net/guest37183b/modelo-de-ciclo-de-vida-en-esprial>

2.6.1. INTRODUCCIÓN DE VISUAL .NET

Con la aparición de .NET Framework, y de Visual Basic .NET, como una de las herramientas estrella para el desarrollo sobre esta nueva plataforma de trabajo, estamos asistiendo a una evolución o revolución sin precedentes en el mundo de la informática, que sitúa a este clásico de la programación en una posición difícil de igualar y menos aún de superar.

Visual Basic .NET (VB.NET a partir de ahora), como cada nueva versión de las que han aparecido en el mercado de este producto, incorpora, como es natural, un buen conjunto de novedades. Sin embargo, la inclusión de Visual Basic en el entorno de .NET, añade también un resumen de drásticos cambios para los programadores de versiones anteriores, derivados en su conjunto, de la necesidad de afrontar con garantías de éxito el desarrollo de la nueva generación de aplicaciones para Internet, objetivo perseguido por todas las herramientas de desarrollo actuales.

Tales cambios, como decimos, son necesarios para la plena integración de Visual Basic con el resto de lenguajes del entorno de .NET; un alto porcentaje, suponen la mejora sobre ciertas características del lenguaje y la eliminación de aspectos obsoletos, arrastrados por una compatibilidad, que en ocasiones como la actual, es necesario dejar atrás; en otros casos, se trata de adaptar nuestras costumbres a nuevos modos y hábitos de programar.¹⁰

2.6.2. .NET FRAMEWORK

.NET Framework constituye la plataforma y elemento principal sobre el que se asienta Microsoft.NET. De cara al programador, es la pieza fundamental de todo este nuevo modelo de trabajo, ya que proporciona las herramientas y servicios que necesitará en su labor habitual de desarrollo. .NET Framework permite el desarrollo de aplicaciones a través del uso de un conjunto de herramientas y servicios que proporciona, y que pueden agruparse en tres bloques principales: el Entorno de Ejecución Común; la jerarquía de clases básicas de la plataforma o .NET Framework Base Classes; y el motor de

¹⁰<http://es.scribd.com/doc/28268501/Programacion-en-ASP>

generación de interfaz de usuario, que permite crear interfaces para la web o para el tradicional entorno Windows, así como servicios para ambos entornos operativos. Finalmente, en el nivel superior, encontramos las clases que permiten el diseño del interfaz de usuario de nuestras aplicaciones. Si necesitamos desarrollar aplicaciones para Internet, utilizaremos ASP.NET, que nos provee de todo lo necesario para crear aplicaciones para la Red. Y no piense el programador tradicional de Windows, que todo en .NET Framework es programación para Internet. La plataforma no se ha olvidado de este colectivo de programadores, que necesitan desarrollar programas para este sistema operativo, y pone a su disposición los denominados WindowsForms, la nueva generación de formularios, con características avanzadas y muy superiores a las del motor de generación de formularios de VB6.¹¹

2.6.3. REQUISITOS HARDWARE

Características mínimas y recomendadas que debe tener el equipo en el que instalemos Visual Studio .NET.

Procesador mínimo Pentium IV de 733 MHz pero es recomendado Pentium Dual Core de 3.0 GHz, con una memoria de 2GB como recomendado 4GB. Con un Disco Duro como mínimo de 320GB.

2.6.4. SISTEMA OPERATIVO

En Visual Studio.NET puede ser instalado en un equipo con uno de los siguientes sistemas operativos:

- Windows Server 2003 Service Pack 2;
- Windows Server 2008; Windows Server 2008 R2;
- Windows Vista Service Pack 1;
- Windows XP Service Pack 3
- Windows XP SP3

¹¹ <http://www.pcmassas.com/viewtopic.php?f=88&t=4002/Markust>

- Windows Server 2003 SP2
- Windows 7

Para aprovechar todo el potencial de desarrollo de la plataforma, es recomendable usar como sistema operativo Windows 2000, ya que ciertos aspectos del entorno no están disponibles si instalamos .NET en otro sistema con menos prestaciones.

2.6.5. RECOMENDACIONES PREVIAS

Es recomendable realizar la instalación sobre un equipo limpio, es decir, un equipo con el software mínimo para poder realizar pruebas con .NET Framework, o con otro tipo de aplicaciones sobre las que estemos seguros de que no se van a producir conflictos con el entorno. En este sentido, una buena práctica consiste en crear en nuestro disco duro una partición que utilizaremos para el trabajo cotidiano, y otra partición en la que instalaremos Visual Studio.NET.¹²

2.6.6. DISEÑO DEL FORMULARIO

Realizaremos un nuevo proyecto al que daremos el nombre de Practica1. En el formulario del proyecto, Form1, añadiremos los controles que permitirán al usuario escribir un texto, grabar dicha programación con Visual Basic .NET.

2.6.7. CONTROLES FUNCIONES Y EVENTOS

Windows es un sistema basado en eventos. Esto quiere decir que cualquier cosa que ocurra dentro de un formulario, bien sobre el propio formulario, o a cualquiera de los controles que lo componen, se detecta a través de un suceso o evento: pasar el ratón sobre un control, hacer clic, escribir en un control de texto, cerrar el formulario, redimensionarlo, entre otros. Cualquier acción de este tipo provoca el evento correspondiente. Los botones tienen eventos, así como los demás controles, y aunque no lo percibamos, se están produciendo constantemente, lo que sucede, es que una vez que dibujamos un control en un

¹²<http://www.slideshare.net/Rokr02/primeros-pasos-375966>

formulario, dicho control inicialmente no está programado para responder a los eventos, por lo que debemos escribir el código para el evento correspondiente, en respuesta a la acción del usuario.

Procedimiento para el Evento Clic de un Button

Se trata de un procedimiento cuyo nombre, `btnSalir_Clic`, compuesto del nombre del control y del evento, lo proporciona automáticamente el IDE. Recibe dos parámetros `sender` y con información adicional proporcionada por el entorno. Pero lo verdaderamente importante está al final de la declaración: “Handles `btnSalir_Clic`”. La palabra clave `Handles`, seguida del nombre de un control, un punto y el nombre de un evento de ese control, le indica al entorno de .NET Framework que debe ejecutar este procedimiento cuando se produzca el evento para ese control.¹³

Programación Orientada A Eventos

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

Eventos.-Un evento es una acción 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 TickEvent`.

Manejador De Eventos.- 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 y asociados a estos controles, entonces el desarrollador escribe el código para integrar los eventos consistentes con el diseño de la aplicación.¹⁴

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

¹³<http://es.scribd.com/doc/100531540/106/Controles-y-eventos>

¹⁴<http://www.taringa.net/posts/apuntes-y-monografias/4618296/Visual-Studio--Ejercicios.html>

1. Eventprocedures.- 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.

Cada manejador de eventos provee dos parámetros, el primer parámetro llamado sender provee una referencia al objeto que dispara el evento, el segundo parámetro es un objeto cuyo tipo de dato depende del evento que es manejado. Ambos parámetros son pasados por valor.

2. Sub procedures.- contiene código que el desarrollador crea para realizar una acción lógica.

3. Functionprocedures.- Contiene código que el desarrollador crea para realizar una acción lógica y regresa un valor, el valor que una función envía de regreso al programa que lo invoca es llamado valor de regreso. Para regresar un valor se utiliza la sentencia Return.¹⁵

2.6.8. ÁMBITO DE LAS VARIABLES

El ámbito de una variable es la región de código en la cual la variable se referencia directamente. Existen dos tipos de ámbitos de las variables:

1. Local.- Es una variable declarada dentro de un procedimiento y se destruye cuando el procedimiento termina de ejecutarse.

2. Modular.- Es una variable declarada a nivel módulo fuera de cualquier procedimiento y son declaradas en la parte superior del Editor de Código arriba del primer procedimiento, este espacio es llamado Sección de Declaraciones Generales Convencionales para el nombre de los controles.

Es recomendable utilizar convenciones para el nombre de los controles, es decir, que al momento de dar lectura

¹⁵<http://www.programatium.com/vbnet.htm>

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 utiliza un control botón, es recomendado utilizar como prefijo btn después complételo con un nombre descriptivo acorde a la funcionalidad que este tendrá.¹⁶

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

2.6.8.1. Ámbito de las Variables

Control Button.-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 Datetimepicker.- El control Datetimepicker muestra por defecto la fecha actual que es posible modificar a través de su propiedad value.

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 ítems de una lista de opciones.

¹⁶ <http://es.scribd.com/doc/61199544/12/Ambito-de-las-variables> / Danny Farfan

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

Evento KeyPress.- El evento Keypress es utilizado para capturar la tecla digitada por el usuario. Este evento distingue entre letras mayúsculas y minúsculas. El segundo argumento del este evento expone dos propiedades: Handled y KeyChar, la propiedad KeyChar es el caracter correspondiente a la tecla digitada. La propiedad Handled es un valor booleano, si se establece este valor como True entonces indicamos al formulario que no procese el evento.

Control Linklabel.- A través de este control es posible abrir el navegador por default y acceder a un URL específico. Donde LinkLabel1.Text es el valor que tiene asignado la propiedad Text por ejemplo file:///C:/pagina.html si quiere probar sin tener acceso a internet o si quiere probar con acceso a internet seria <http://www.elSitioWeb.com>.

Si requiere especificar el navegador con el cual desea abrir el URL basta con señalarlo con `System.Diagnostics.Process.Start ("firefox.exe", LinkLabel2.Text)`.

Control Timer.- A través del control Timer es posible ejecutar una instrucción en un intervalo de tiempo específico, este Timer se activa estableciendo su propiedad Interval con el valor True y se ejecutará hasta que por medio de la acción del usuario lo detenga o se desactive el temporizador.

Cuando agrega un Timer este no es visible en el formulario pero si en la parte inferior de la pantalla (Bandeja de Componentes), justo debajo del formulario.

En Visual Basic .NET el nombre del control cambia su nombre por `Timer_Tick`.

Controles Anteriores Y Otros Controles.- Es posible utilizar antiguos controles ActiveX y utilizarlos con algunas limitaciones, teniendo como restricción técnica que deberán estar en un contenedor.

Funciones Intrínsecas

- Val, regresa el número contenido en un argumento, se detiene en el primer carácter no numérico.
- CDbI, 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.
- Format (expr,str), convierte una expresión a un formato string específico.¹⁷

Funciones De Tiempo Y Hora.- En Visual Basic .NET existen funciones a través de las cuales es posible manipular el tiempo y la hora:

- TimeString, regresa la hora actual del sistema.
- DateString, regresa la fecha actual del sistema.
- Now, regresa un valor codificado que representa la hora y fecha actual del sistema.
- Hour (hora), regresa el número de hora actual del sistema.
- Minute (hora), regresa el número de minuto actual del sistema.
- Second (hora), regresa el número del segundo actual del sistema.

¹⁷ http://www.sc.ehu.es/sbweb/webcentro/automatica/web_8051/Contenido/cursoc51/Capitulo%2010/funciones_intrInsecas.htm

- Day (fecha), regresa el número del día actual del sistema.
- Month (fecha), regresa el número de mes actual del sistema.
- Year (fecha), regresa el año actual del sistema.
- Weekday (fecha), regresa el número que representa el día de la semana. (1 = Domingo, 2 = Lunes,...).¹⁸

2.6.9. CUADROS DE DIÁLOGO

Visual Basic .NET cuenta con siete tipos de cuadro de diálogo que facilitan la construcción de interfaz de usuario:

1. OpenFileDialog, obtiene unidad, nombre de directorio y nombre de un archivo existente.
2. SaveFileDialog, obtiene unidad, nombre de directorio y nombre para un archivo nuevo.
3. FontDialog, para que el usuario seleccione una configuración para la fuente.
4. ColorDialog, para que el usuario seleccione un color de una paleta.
5. PrintDialog, para que el usuario defina opciones de impresión.
6. PrintPreviewDialog, muestra al usuario una vista previa de impresión.
7. PageSetupDialog, para que el usuario controle las opciones de configuración de página.

Al momento de diseñar la interfaz de usuario los controles de cuadro de diálogo no aparecen en la pantalla pero si se muestran visibles al pie de la misma, pudiendo acceder a sus propiedades para configurar el control.¹⁹

Si requiere hacer visible el cuadro de diálogo en modo ejecución tendrá que emplear el método ShowDialog ().

¹⁸<http://www.buenastareas.com/materias/funciones-en-visual-basic/0>

¹⁹<http://www.monografias.com/trabajos2/guiavb/guiavb2.shtml>

Dim as

Visual Basic .NET no permite el uso del tipo de dato Variant y todas las variables deberán ser declaradas por las instrucciones Dim nombreVariable as tipo Dato.

Dim es la abreviatura de Dimensión, que sirve para reservar espacio para la variable. La sentencia Dim especifica el nombre de la variable y su tipo de dato asociado.

Las variables deben tener un nombre único llamado Identificador, los identificadores se forman siguiendo las siguientes reglas:

- Comenzar con una letra o guión bajo
- Deben contener letras, dígitos y guiones bajos
- No pueden ser palabras reservadas²⁰

Recomendaciones para las variables:

- El nombre de las variables deberían comenzar con una letra.
- El nombre de las variables deberían distinguir cada nueva palabra escribiendo la primera letra en mayúscula.
- El nombre de las variables deberían ser lo suficientemente largo para ser significativo y lo suficientemente corto para ser manejable.

Si no se define un tipo de dato para la variable, entonces asume el tipo de dato por default que es Object.

Visual Basic .NET incorpora la instrucción Option Strict si desea que los tipos de datos en las operaciones sean del mismo tipo y evitar la conversión y pérdida de datos.²¹

²⁰ http://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=140:tipos-de-variables-en-visual-basic-integer-string-variant-boolean-etc-ejemplos-cu00308

²¹ <http://www.eggheadcafe.com/Ángel Esteban Núñez>

2.6.10. TIPOS DE DATOS

Los tipos de datos numéricos en Visual Basic .NET se agrupan en dos categorías: enteros y de punto flotante.

- Los tipos de datos numéricos a su vez se agrupan en, Byte, Short, Integer y Long.
- Los tipos de datos de punto flotante son Single y Double.
- Los tipos de datos Char almacenan un sólo carácter en formato Unicode.
- Los tipos de datos String almacenan una secuencia de caracteres Unicode.

Unicode es un sistema internacional de codificación de 16-bit que cubre valores para más de 45.000 caracteres. Un carácter Unicode es almacenado como un valor numérico sin signo de 16-bit, de 0 a 65.535. Los primeros 128 (0-127) caracteres Unicode corresponden al conjunto de caracteres ASCII. Del 32 al 127 corresponden a caracteres alfanuméricos y símbolos para un teclado US. Del 128 al 255 representan caracteres especiales.

- El tipo de dato Boolean es un valor sin signo el cual es interpretado como falso o verdadero.
- El tipo de dato Date almacena fechas y tiempo, el rango de valores posible es del 12:00:00 AM Enero 1 al 31 de Diciembre del 9999 11:59:59 PM. Los valores para las fechas deben ser encerrados entre el signo # y con el formato m/d/aaaa, por ejemplo #12/7/1971#.
- El tipo de dato Object es el tipo de dato universal en Visual Basic .NET y es también el tipo de dato por default para las variables que son declaradas sin especificar su tipo de dato.²²

²²<http://teleformacion.edu.aytolacoruna.es/PASCAL/document/tipos.htm>

Tipos de Datos Definidos por el Usuario

Structure

Visual Basic permite al desarrollador crear sus propios tipos de datos, esto es posible.NET a través del empleo de la instrucción Structure, ejemplo:

```
Structure Persona
```

```
    Dim sNombre As String
```

```
    Dim nEdad As Integer
```

```
    Dim dFechaNac As Date
```

```
End Structure
```

2.6.11. OPERADORES

Visual Basic .NET dispone de los siguientes operadores matemáticos:

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

Tabla 2.6.11.1: Operadores

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 se escribía

`x += 1` 'Ahora puede escribir

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

Tabla 2.6.11.2: 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
+ -

Tabla 2.6.11.3: Precedencia de Operadores

2.6.12. SENTENCIAS DE CONTROL

La sentencia For se utiliza cuando se conoce de antemano el número de iteraciones a ejecutarse. Do se utiliza cuando no se conoce de antemano el número de iteraciones a ejecutarse. Si al menos debe ejecutarse 1 iteración es necesario colocar la expresión condicional al final del ciclo. Evite ciclos infinitos, para ello asegúrese de que estos ciclos tengan una expresión condicional de salida.

La única sentencia de control que tiene un cambio relevante sintácticamente es While ya que antes se escribía While - Wend ahora es necesario escribir While - End While.

Visual Basic .NET incorpora 2 nuevos operadores lógicos para utilizarlos en las sentencias de control, a continuación se mostrará la tabla de operadores lógicos anteriores y nuevos:

TR>

Operadores Lógicos	Descripción
And	Las 2 expresiones deben ser verdaderas
Or	Alguna de las 2 expresiones es verdadera
Not	Negación del resultado de la expresión
Xor	Si 1 y sólo 1 de las expresiones es verdadera
AndAlso	Si la primer y segunda condición son verdaderas
OrElse	Si la primer o segunda condición es verdadera

Tabla 2.6.11.4: Operadores Lógicos

Los nuevos operadores lógicos AndAlso y OrElse reciben el nombre de sistema de corto-circuito.²³

La sentencia de control Do - Loop

Es la sentencia general iterativa, la cual permite ejecutar repetitivamente un grupo de sentencias hasta que una condición sea cumpla. Existen cuatro versiones de la sentencia Do - Loop:

Do While condición

Sentencias

Loop

Do Until condición

Sentencias

Loop

Do

Sentencias

Loop While condición

Do

Sentencias

Loop Until condición²⁴

Función IsNumeric

La función IsNumeric () regresa un valor booleano el cual indica cuando una expresión puede ser evaluada como un número.

Esta función es básica de una aplicación para validar la entrada correcta de datos de tipo numérico.

Tendrá que hacer su función de validación tan robusta como la requiera, por ejemplo una validación previa en una entrada de datos, seria pensar que el usuario intento dar un número pero al escribir, sin desearlo digito un espacio, para la perspectiva del usuario novato, quizá vea que efectivamente escribió un número y no le toma importancia a los espacios en blanco, por lo tanto lo

²³<http://es.scribd.com/doc/54966596/Manual-de-Visual>

²⁴[https://www.itescam.edu.mx/Juan Germán Gama Aguilar](https://www.itescam.edu.mx/Juan_Germán_Gama_Aguilar)

primero que tendríamos que realizar es la eliminación de los espacios en blanco, pero no sólo los que probablemente estén al inicio, sino los que también estén en la parte intermedia y los que estén en la parte final, para la eliminación de los espacios en blanco al inicio y al final de la entrada del usuario podríamos utilizar una sola función Trim.

Trim, LTrim, RTrim

Visual Basic .NET provee mecanismos para la eliminación de espacios en blanco contenidos en un string a través de las siguientes funciones:

- LTrim, regresa un string que contiene una copia de un string específico al cual se eliminaron los espacios en blanco contenidos al inicio del string.
- RTrim, regresa un string que contiene una copia de un string específico al cual se eliminaron los espacios en blanco contenidos al final del string.
- Trim, regresa un string que contiene una copia de un string específico al cual se eliminaron los espacios en blanco contenidos al inicio y al final del string.

2.6.13. VENTANA DE COMANDOS

A través del depurador de Visual Basic .NET es posible abrir una ventana que nos permita conocer el valor que tienen las variables en cierto momento o para requerir de una ayuda auxiliar.

La Ventana de Comandos nos permite escribir por ejemplo un objeto y al escribir la notación de punto (.) aparecen las propiedades y métodos de este objeto., por ejemplo escriba:

>File.

Para abrir o activar esta ventana de clic en DebugWindows/Inmediate, esta ventana es capaz de soportar dos modos:

1. En Modo Inmediato (Inmediate)
2. En Modo Comando (Command)

Para cambiar de un modo a otro simplemente escriba:

1. >cmd, si está en modo Inmediato y se desea pasar a modo Comando
2. >immed, si está en modo Comando y se desea pasar a modo Inmediato.²⁵

Ventana de Comandos - Inmediata

La Ventana de Comandos - Inmediata, nos permite evaluar o conocer el valor de una variable por ejemplo si su programa utiliza una variable llamada sNombre y quiere conocer qué valor tiene.²⁶

2.6.14. VENTAJAS Y DESVENTAJAS

Ventajas

- Integra el diseño e implementación de formularios de Windows.
- Permite usar con facilidad la plataforma de los sistemas Windows, dado que tiene acceso prácticamente total a la API de Windows, incluidas librerías actuales.
- Tiene integrada una herramienta de depuración de errores, con la cual es posible ejecutar el código:
- Es uno de los lenguajes de uso más extendido, por lo que resulta fácil encontrar información, documentación y fuentes para los proyectos.
- Permite desarrollar grandes y complejas aplicaciones, también provee un entorno adecuado para realizar pequeños prototipos rápidos.

Desventajas

Las desventajas que tiene Visual Basic anteriores a VB.NET son variadas, entre ellas se citan:

- No puede manejar directamente los porque no hay una codificación más necesaria y el pensamiento para manejar un puntero.²⁷

²⁵<http://issuu.com/Joel Martin Chuco Marrufojmchm19@hotmail.com>

²⁶<http://www.programatium.com/vbnet.htm>

2.7. BASES DE DATOS

Una base de datos también es abreviada con la sigla BD en un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. En la actualidad, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital (electrónico), que ofrece un amplio rango de soluciones al problema de almacenar datos.

Existen programas denominados Sistemas Gestores de Bases de Datos, abreviado SGBD, que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de estos SGBD, así como su utilización y administración, se estudian dentro del ámbito de la informática.

Las aplicaciones más usuales son para la gestión de empresas e instituciones públicas. También son ampliamente utilizadas en entornos científicos con el objetivo de almacenar la información experimental.

2.7.1. TIPOS DE BASES DE DATOS

Las bases de datos pueden clasificarse de varias maneras, de acuerdo al contexto que se esté manejando, la utilidad de las mismas o las necesidades que satisfagan.

Según La Variabilidad De Los Datos Almacenados

➤ Bases de datos estáticas

Son bases de datos de sólo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones y tomar decisiones.

²⁷Las desventajas que tiene Visual Basic anteriores a VB.NET

➤ **Bases de datos dinámicas**

Éstas son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización, borrado y adición de datos, además de las operaciones fundamentales de consulta. Un ejemplo de esto puede ser la base de datos utilizada en un sistema de información de un supermercado, una farmacia, un videoclub o una empresa, esta es la base de dato que normalmente se utiliza.

Según El Contenido

➤ **Bases de datos bibliográficas**

Sólo contienen un representante de la fuente primaria, que permite localizarla. Un registro típico de una base de datos bibliográfica contiene información sobre el autor, fecha de publicación, editorial, título, edición, de una determinada publicación, etc. Puede contener un resumen o extracto de la publicación original, pero nunca el texto completo, porque si no, estaríamos en presencia de una base de datos a texto completo.²⁸

2.7.2. SQL SERVER 2005

SQL Server 2005 es una plataforma global de base de datos que ofrece administración de datos empresariales con herramientas integradas de inteligencia empresarial. El motor de la base de datos SQL Server 2005 ofrece almacenamiento más seguro y confiable tanto para datos relacionales como estructurados, lo que permite crear y administrar aplicaciones de datos altamente disponibles y con mayor rendimiento para utilizar en su negocio.

El motor de datos SQL Server 2005 constituye el núcleo de esta solución de administración de datos empresariales. Así mismo, combina lo mejor en análisis, información, integración y notificación. Esto permite que su negocio cree y despliegue soluciones rentables que ayuden a su equipo a incorporar datos en cada rincón del negocio a través de tableros de comando, escritorios digitales, servicios Web y dispositivos móviles.

²⁸http://es.wikipedia.org/wiki/Base_de_datos

SQL Server 2005 ofrece soluciones innovadoras que le ayudan a obtener más valor de sus datos.

2.7.3. ESPECIFICACIONES

- Soporte de transacciones.
- Escalabilidad, estabilidad y seguridad.
- Soporta procedimientos almacenados.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo acceden a la información.
- Además permite administrar información de otros servidores de datos.²⁹

2.7.4. OPERACIONES CON CONJUNTOS

SQL Server 2005 permite tres tipos de operaciones con conjuntos:

- Union
- Except
- Intersect

Para utilizar operaciones de conjuntos debemos cumplir una serie de normas.

- Las consultas a unir deben tener el mismo número campos, y además los campos deben ser del mismo tipo.
- Sólo puede haber una única clausula ORDER BY al final de la sentencia SELECT.

UNION.- Esta disponible en todas las versiones de SQL Server y este devuelve la suma de dos o más conjuntos de resultados. El conjunto obtenido como resultado de UNION tiene la misma estructura que los conjuntos originales.

²⁹<http://www.monografias.com/trabajos73/microsoft-sql-server/microsoft-sql-server2.shtml>/Diego Martin Ramos Ortega

El siguiente ejemplo muestra el uso de UNION.

```
SELECT      Nombre,      Apellido1      ,
Apellido2,NifCif,FxNacimiento

FROM EMPLEADOS

UNION

SELECT      Nombre,      Apellido1      ,
Apellido2,NifCif,FxNacimiento

FROM CLIENTES
```

Cuando realizamos una consulta con UNION internamente se realiza una operación DISTINCT sobre el conjunto de resultados final. Si queremos obtener todos los valores debemos utiliza UNION ALL.

```
SELECT      Nombre,      Apellido1      ,
Apellido2,NifCif,FxNacimiento

FROM EMPLEADOS

UNIONALL

SELECT      Nombre,      Apellido1      ,
Apellido2,NifCif,FxNacimiento

FROM CLIENTES
```

EXCEPT.- Es nuevo en SQL Server 2005 y devuelve la diferencia o resta de dos o más conjuntos de resultados. El conjunto obtenido como resultado de EXCEPT tiene la misma estructura que los conjuntos originales.

El siguiente ejemplo muestra el uso de EXCEPT

```
SELECT      Nombre,      Apellido1      ,
```

```
Apellido2,NifCif,FxNacimiento
```

```
FROM EMPLEADOS
```

```
EXCEPT
```

```
SELECT Nombre, Apellido1 ,  
Apellido2,NifCif,FxNacimiento
```

```
FROM CLIENTES
```

El uso de EXCEPT, como norma general, es mucho más rápido que utilizar condiciones NOT IN o EXISTS en la cláusula WHERE.

INTERSECT.- También es nuevo en SQL Server 2005.Devuelve la intersección entre dos o más conjuntos de resultados en uno. El conjunto obtenido como resultado de INTERSECT tiene la misma estructura que los conjuntos originales.³⁰

El siguiente ejemplo muestra el uso de INTERSECT

```
SELECT Nombre, Apellido1 ,  
Apellido2,NifCif,FxNacimiento
```

```
FROM EMPLEADOS
```

```
INTERSECT
```

```
SELECT Nombre, Apellido1 ,  
Apellido2,NifCif,FxNacimiento
```

```
FROM CLIENTES
```

2.7.5. TIPOS DE DATOS

SQL Server 2005 tiene varios tipos de datos que los especificamos a continuación.

³⁰<http://www.devjoker.com/html/Operaciones-con-conjuntos.html>

- **BINARY** Tiene una longitud 1 byte que es para consultas sobre tabla adjunta de productos de bases de datos que definen un tipo de datos Binario.
- **BIT** Posee un 1 byte Valores Si/No ó True/False
- **BYTE** Este tipo de dato posee 1 byte que es un valor entero entre 0 y 255
- **COUNTER** Tiene 4 bytes que es un número incrementado automáticamente (de tipo Long)
- **CURRENCY** Tiene 8 bytes que es un entero escalable entre 922.337.203.685.477,5808 y 922.337.203.685.477,5807
- **DATETIME** Posee 8 bytes cuyo valor de fecha y hora tiene entre los años 100 y 9999.
- **SINGLE** Tiene 4 bytes posee un valor en punto flotante de precisión simple con un rango de -3.402823×10^{38} a $-1.401298 \times 10^{-45}$ para valores negativos, 1.401298×10^{-45} a 3.402823×10^{38} para valores positivos, y 0
- **DOUBLE** Tiene 8 bytes con un valor en punto flotante de doble precisión con un rango de $-1.79769313486232 \times 10^{308}$ a $-4.94065645841247 \times 10^{-324}$ para valores negativos, $4.94065645841247 \times 10^{-324}$ a $1.79769313486232 \times 10^{308}$ para valores positivos, y 0
- **SHORT** Tiene 2 bytes cuyo entero corto es -32,768 y 32,767.
- **LONG** posee 4 bytes que consiste en un entero largo entre -2,147,483,648 y 2,147,483,647.
- **LONGTEXT** Tiene 1 byte por carácter de cero a un máximo de 1.2 gigabytes.
- **LONGBINARY** Según se necesite se utiliza de cero 1 gigabyte. Utilizado para objetos OLE.

- **TEXT** Utiliza 1 byte por carácter De cero a 255 caracteres.³¹

2.7.6. VARIABLES

Una variable es un valor identificado por un nombre o identificador sobre el que podemos realizar modificaciones, los identificadores de variables deben comenzar por el carácter @, es decir; el nombre de una variable debe comenzar por @. Para declarar variables en SQL debemos utilizar la palabra clave declare, seguido del identificador y tipo de datos de la variable.

```
sql = "INSERT INTO usuario(nombres,usuario,clave,confclave)
VALUES(@nom,@nom,@nom,@nom)"
```

```
DECLARE @nombre VARCHAR(100),
@apellido1 VARCHAR(100),
@apellido2 VARCHAR(100)
```

2.7.7. ESTRUCTURAS DE CONTROL

Estructura condicional IF

La estructura condicional IF nos permite evaluar una expresión booleana de resultado SI o NO, y ejecutar las operaciones contenidas en el bloque formado por BEGIN END.

```
IF (<expresion>)
BEGIN
...
END
ELSEIF (<expresion>)
BEGIN
...
END
ELSE
```

³¹<http://www.utm.mx/~rruiz/cursos/PE/notas.pdf/ErnestMart>

```
BEGIN  
...  
END
```

Estructura condicional CASE

La estructura condicional CASE permite evaluar una expresión y devolver un valor u otro.

La sintaxis general de case es:

```
CASE<expresion>  
  
    WHEN<valor_expresion>THEN<valor_devuelto>  
  
    WHEN<valor_expresion>THEN<valor_devuelto>  
    ELSE<valor_devuelto>-- Valor por defecto  
END
```

Bucle WHILE

El bucle WHILE se repite mientras la expresión se evalúe como verdadero. Es el único tipo de bucle del que dispone Transact SQL.

```
WHILE<expresion>  
BEGIN  
...  
END
```

Podemos pasar a la siguiente iteración del bucle utilizando **CONTINUE**.

```
DECLARE @contador int  
SET @contador = 0  
WHILE (@contador < 100)  
BEGIN  
    SET @contador = @contador + 1  
IF (@contador % 2 = 0)  
        CONTINUE
```

```

        PRINT 'Iteracion del bucle ' +
cast(@contadorASvarchar)
    END

```

El bucle se dejará de repetir con la instrucción **BREAK**.

```

    DECLARE @contadorint
    SET @contador = 0
    WHILE (1 = 1)
    BEGIN
        SET @contador = @contador + 1
        IF (@contador % 50 = 0)
            BREAK
        PRINT 'Iteracion del bucle ' +
cast(@contadorASvarchar)
    END

```

También podemos utilizar el bucle **WHILE** conjuntamente con subconsulta.

```

    DECLARE@coReciboint
    WHILE EXISTS (SELECT *
        FROM RECIBOS
        WHERE PENDIENTE = 'S')-- Ojo,
la subconsulta se ejecuta
-- una vez por cada
iteracion
-- del bucle!
    BEGIN
        SET @coRecibo = (SELECTTOP 1 CO_RECIBO
            FROM RECIBOS WHERE PENDIENTE = 'S')
        UPDATE RECIBOS
        SET PENDIENTE = 'N'
        WHERE CO_RECIBO = @coRecibo
    END

```

Estructura GOTO

La sentencia goto nos permite desviar el flujo de ejecución hacia una etiqueta. Fue muy utilizada en versiones anteriores de SQL Server conjuntamente con la variable de sistema @@ERROR para el control de errores.

Actualmente, se desaconseja el uso GOTO, recomendándose el uso de TRY - CATCH para la gestión de errores.

```
DECLARE @divisor int,
        @dividendo int,
        @resultado int

SET @dividendo = 100
SET @divisor = 0
SET @resultado =
@dividendo/@divisor
    IF @@ERROR > 0
        GOTO error
PRINT 'No hay error'
RETURN
error:
    PRINT 'Se ha producido una división por cero'
```

2.7.8. SENTENCIAS EN SQL SERVER

Hasta ahora hemos visto cómo se almacenan los datos en una base de datos y como consultar esos datos almacenados, pero no hemos visto como almacenar dichos datos.

INSERT.- Para almacenar información en una base de datos debemos insertar filas en las tablas. Para ellos SQL pone a nuestra disposición la sentencia **INSERT**.

Utilizado para cargar lotes de datos en la base de datos en una única operación.

Inserción De Filas

El proceso de inserción de filas consiste en añadir a una tabla una o más filas y en cada fila todos o parte de sus campos.

Podemos distinguir dos formas de insertar filas:

- Inserción individual de filas.
- Inserción múltiple de filas.

La sintaxis de la sentencia **INSERT** es diferente según cual sea nuestro propósito. Sólo podremos omitir un campo al efectuar una inserción cuando este acepte valores nulos.

Para realizar la inserción individual de filas SQL posee la instrucción **INSERT INTO**. La inserción individual de filas es la que más comúnmente utilizaremos. Su sintaxis es la siguiente:

```
INSERT INTO<nombre_tabla>  
  
[ (<campo1>[, <campo2>, ...] ) ]  
values  
(<valor1>,<valor2>,...);
```

DELETE.- Se lo utiliza para eliminar algún registro de una tabla en una base de datos como SQL Server 2005.

SELECT.- Es utilizado para consultar registros de la base de datos que satisfagan un criterio determinado.

UPDATE.- El Update en SQL Server es utilizado para modificar los valores de los campos y registros especificados.

TRANSACCIONES.- Una transacción es un conjunto de operaciones que van a ser tratadas como una única unidad.

CONSULTAS.- Es un lenguaje de bases de datos normalizado, utilizado por los diferentes motores de bases de datos para realizar determinadas operaciones sobre los datos o sobre la estructura de los mismos.³²

2.7.9. CLAUSULAS EN SQL SERVER

Las cláusulas en SQL Server 2005 son muy importante ya que se las puede utilizar en las instrucciones SELECT en las que se estén recuperando datos de tablas o vistas.

FROM.- Use la cláusula FROM para lo siguiente:

➤ Enumerar las tablas y vistas que contienen las columnas a las que se hace referencia en la lista de selección y en la cláusula WHERE. Los nombres de las tablas y vistas se pueden sustituir por alias mediante la cláusula AS.

La evaluación de las diferentes cláusulas en tiempo de ejecución se efectúa en el siguiente orden:

- WHERE filtra las filas
- GROUP BY crea una tabla de grupo nueva
- HAVING filtra los grupos
- ORDER BY clasifica la salida

GROUP BY.- SQL nos permite agrupar las filas resultado de una consulta en conjuntos y aplicar funciones sobre esos conjuntos de filas.

La sintaxis es la siguiente:

SELECT { * | {columna,}+ }

FROM {tabla,}+

WHERE condición

³²<http://personal.lobocom.es/claudio/sql001.htm>

GROUP BY {columna ,}+

HAVING condición

ORDER BY {expresión Columna [ASC | DESC],}+;

En la cláusula GROUP BY se colocan las columnas por las que vamos a agrupar. Y en la cláusula HAVING se especifica la condición que han de cumplir los grupos para pasar al resultado.

HAVING.- Se coloca generalmente cerca del fin de la instrucción SQL, y la instrucción SQL con la cláusula HAVING. Puede o no incluir la cláusula GROUP BY.

Sintaxis para HAVING es:

```
SELECT "nombre1_columna", SUM ("nombre2_columna")  
FROM "nombre_tabla"  
GROUP BY "nombre1_columna"  
HAVING (condición de función aritmética)
```

2.8. HIPÓTESIS Y VARIABLES

2.8.1. HIPÓTESIS GENERAL

Con el desarrollo de un sistema de gestión facilitaremos la administración de bodega para el control de inventario en el comisariato el Popular.

2.8.2. HIPÓTESIS ESPECÍFICA

- Con el nuevo sistema, los procesos de ingresos y consultas de productos se realizarán de forma rápida.
- Los procesos de reportes serán más rápidos y exactos.

2.8.3. VARIABLES

Variable Independiente: Control de Inventario

Variable Dependiente: Sistema de Gestión

3. MARCO METODOLOGICO

3.1 MODALIDAD DE LA INVESTIGACIÓN

Una investigación es un proceso sistemático, organizado y objetivo teniendo como finalidad aumentar el conocimiento de información sobre algo desconocido para este trabajo aplicaremos dos modalidades diferentes.

La investigación más apropiada se inicia con la búsqueda de lo más eficaz para nuestro propósito, lo que implica que sea competente, es decir, que corresponda desde el punto de vista de la negociación que se produce entre: a) las posibilidades del investigador, b) los requerimientos de la investigación, y c) las necesidades a las cuales se destina la investigación, es para la toma de decisiones, información para un proyecto o programa futuro.

Modalidad Cuantitativa

Esta modalidad fue aplicada por que nos permite examinar los datos de manera científica, habitualmente con ayuda de herramientas del campo de la estadística el objeto de estudio.

Para que exista metodología cuantitativa se requiere que haya claridad entre los elementos de investigación que conforman el problema, y que sea posible definirlo, limitarlos y saber exactamente dónde se inicia el problema, en cuál dirección va y qué tipo de incidencia existe entre sus elementos en este caso ha sido representativo ya que ha permitido la evaluación del proceso.

Modalidad Cualitativa

Se utiliza la modalidad cualitativa porque estudia la realidad es decir, tal como sucede, pretendiendo aclarar el objeto de estudio, implica la utilización y recopilación de una gran variedad de materiales que describen la rutina y las situaciones problemáticas y los significados en la vida de las personas.

Considerando esta explicación esta modalidad fue empleada por que permite evaluar aspectos de comportamiento humano es decir se investigó los procesos operativos de la empresa y su influencia en el aspecto administrativo.

3.2. TIPO DE INVESTIGACIÓN

El estudio de investigación se realizara de forma bibliográfica, utilizando encuesta al personal que trabaja en la empresa.

Se puede mencionar dos tipos de investigación.

Bibliográfica.- Este tipo de investigación ha permitido fundamentar científicamente la variable de la presente y sirve para elaborar el marco teórico.

De campo.- Con este tipo de investigación se ha podido valorar el problema, ratificarlo y luego nos orienta a una propuesta de solución.

3.3. POBLACION Y MUESTRA DE LA INVESTIGACIÓN

Formula

N = 30 Tamaño de Población

n = ? Tamaño de la muestra

e²= 0.5 Error admisible

$$n = \frac{N}{(e)^2 \frac{(N-1)}{30} + 1}$$

$$n = \frac{30}{(0.5)^2 \frac{(30-1)+1}{30}}$$

$$n = \frac{30}{(0.0025) \frac{(29)+1}{30}}$$

$$n = \frac{30}{0.0725 + 1}$$

$$n = \frac{1.0725}{27.972}$$

3.4. MÉTODOS, TÉCNICAS E INSTRUMENTOS DE LA INVESTIGACIÓN

Encuesta realizada dentro del personal de la Empresa.

Objetivo: Desarrollar un sistema informático para mejorar los procesos de administración de bodega y control de inventario del comisariato el “Popular”.

1. ¿Está usted de acuerdo con la actual forma en que se realiza el ingreso de datos de los productos para el control de inventarios en el comisariato?

Acuerdo ()

Desacuerdo ()

2. ¿Encuentra alguna dificultad con la forma en que se lleva el registro de información de mercadería que posee la empresa?

Si ()

No ()

3. ¿Cree usted que hubo la necesidad de implementar un sistema administrador de Bodega que permita controlar el manejo de datos de inventarios?

Si ()

No ()

4. ¿En el tiempo que lleva trabajando dentro de la empresa ha notado pérdida de Información?

Si ()

No ()

Desconoce ()

5. ¿Qué opina sobre el actual sistema automatizado para el control de información de los productos de manera rápida y ordenada en el comisariato?

Excelente ()

Bueno ()

Malo ()

6. ¿Considera usted que se maneja un gran volumen de documentación respecto al abastecimiento de mercadería para el comisariato?

Si ()

No ()

7. ¿Considera usted que debido al manejo automatizado de la información las operaciones realizadas en el comisariato se torna de manera rápida?

Si ()

No ()

Tal vez ()

8. ¿Es dificultoso buscar información referente a la mercadería que existe en la empresa para saber el stock de productos?

Si ()

No ()

9. ¿Cree usted que el área administrativa conoce el valor a cancelar de los costos que generan la mercadería?

Si ()

No ()

10. ¿Sería necesario una capacitación informática en el transcurso del manejo del sistema implementado?

Acuerdo ()

Desacuerdo ()

3.5 INTERPRETACION DE RESULTADOS

Encuesta realizada dentro del personal de la Empresa.

1 ¿Está usted de acuerdo con la actual forma en que se realiza el ingreso de datos de los productos para el control de inventarios en el comisariato?

Alternativas	Frecuencia	Porcentaje
De Acuerdo	25	93 %
En desacuerdo	2	7 %
TOTAL	27	100 %

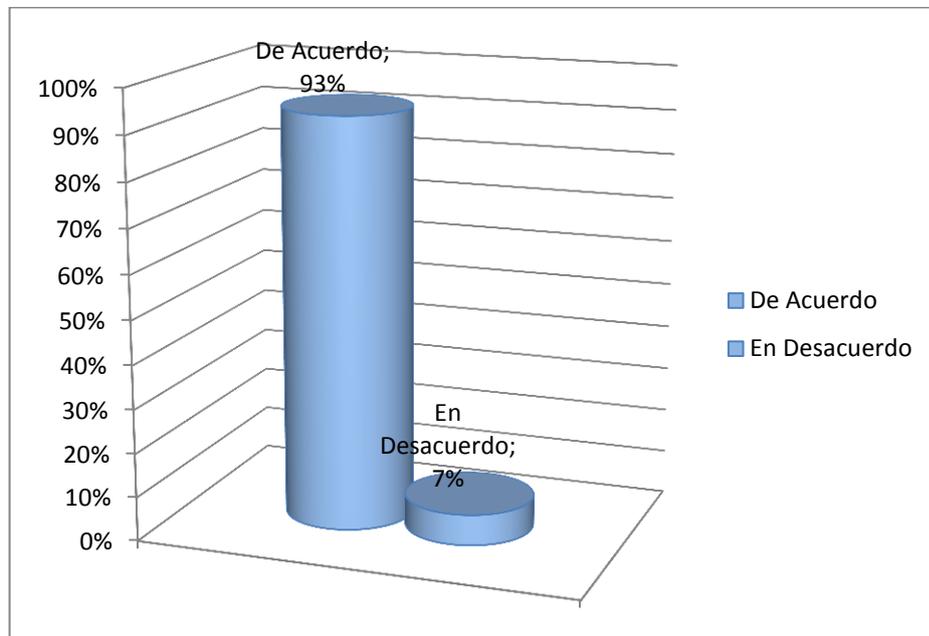


Fig. 3.5.1: Grafico de la Primera Pregunta

INTERPRETACION:

Del número de personas encuestadas, dieron a conocer que estaban de acuerdo, porque la empresa durante toda su trayectoria que lleva debería ya de poseer un sistema automatizado que le permita una gestión adecuada del stock de productos. Para que de esta manera se realice de manera rápida el ingreso de datos que existe en bodega.

2 ¿Encuentra alguna dificultad con la forma en que se lleva el registro de información de mercadería que posee la empresa?

Alternativa	Frecuencia	Porcentaje
Si	6	22%
No	21	78%
TOTAL	27	100%

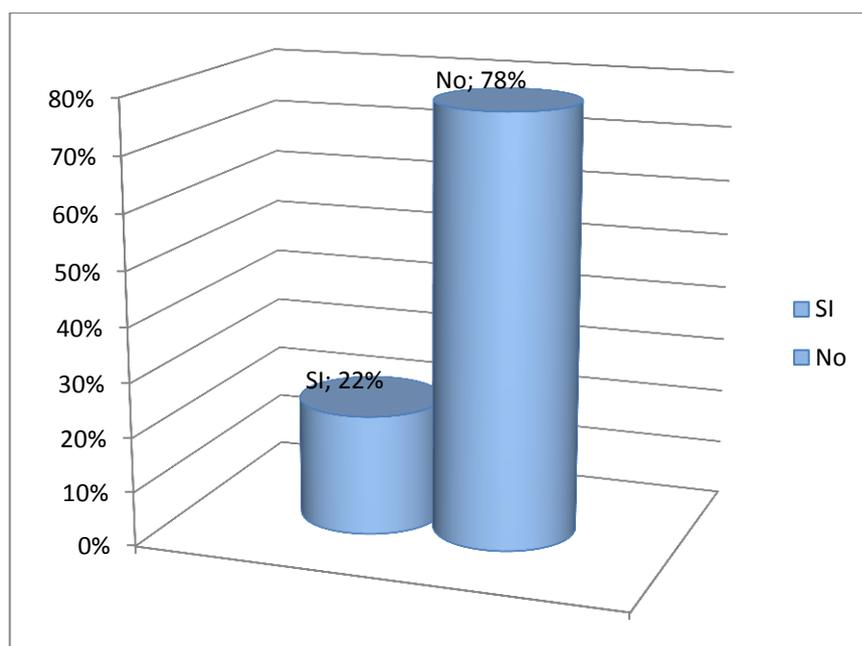


Fig. 3.5.2: Grafico de la Segunda Pregunta

INTERPRETACION:

La mayor parte de las personas encuestada declararon que no tienen problemas en mantener la información del stock al día y actualizada. Además de ello se les facilita en el instante de verificar la rotación que tiene el producto, ya que la información de entradas y salidas es diaria.

3 ¿Cree usted que hubo la necesidad de implementar un sistema administrador de Bodega que permita controlar el manejo de datos de inventarios?

Alternativas	Frecuencia	Porcentaje
Si	23	85%
No	4	15%
TOTAL	27	100 %

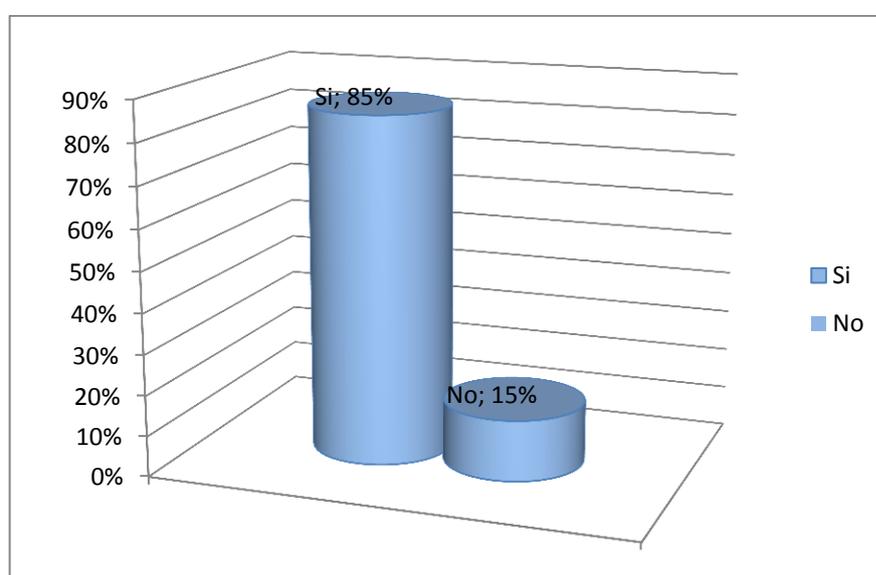


Fig. 3.5.3: Grafico de la Tercera Pregunta

INTERPRETACION:

La mayoría de las persona afirman que fue conveniente implementar el sistema, para obtener datos exactos de su inventario, operaciones de almacenamiento y distribución de mercaderías, que el departamento tanto de contabilidad y administrativo sepa con que suministros puede contar, conociendo el valor o capital inmovilizado que posee en sus bodegas, facilitando así el manejo del control de inventario.

4 ¿En el tiempo que lleva trabajando dentro de la empresa ha notado pérdida de Información?

Alternativas	Frecuencia	Porcentaje
Si	3	11%
No	20	74%
Desconoce	4	15%
TOTAL	27	100 %

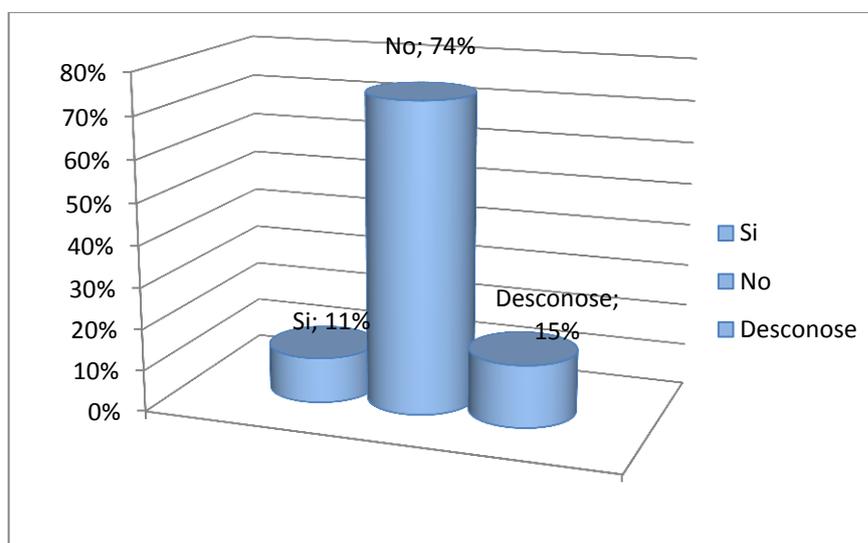


Fig. 3.5.4: Grafico de la Cuarta Pregunta

INTERPRETACION:

El 74% de las personas manifestaron, que al contar con un sistema automatizado se ha reducido la perdida de información, esto se debe por lo general, que se registra todos los días en el sistema las transacciones realizadas por bodega, por lo que hay puntualidad en el registro de consumos, ingresos y transferencia, generando normalmente que entreguen reportes para validar las transferencias recibidas, enviadas de los consumos.

5 ¿Qué opina sobre el actual sistema computarizado para el control de información de los productos de manera rápida y ordenada en el comisariato?

Alternativas	Frecuencia	Porcentaje
Excelente	25	93%
Bueno	2	7%
Malo	0	0%
TOTAL	27	100%

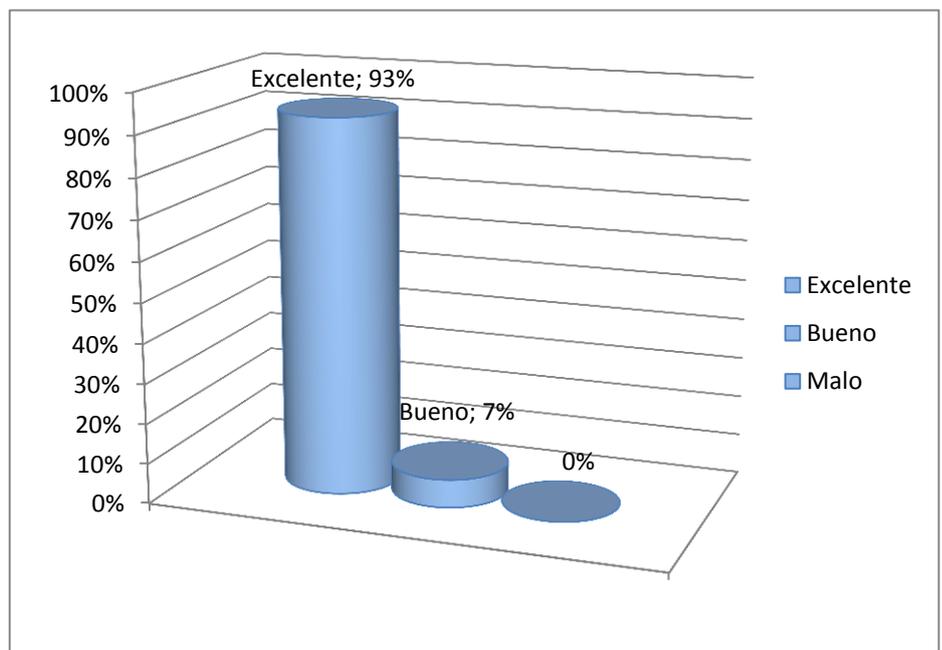


Fig. 3.5.5: Grafico de la Quinta Pregunta

INTERPRETACION:

En su mayoría manifestaron estar de acuerdo con la implementación del sistema que permite el control de información de los productos de manera rápida y ordenada, efectuando así el menor esfuerzo, tiempo y costo, maximizando la eficiencia y eficacia administrativa y contable.

6 ¿Considera usted que se maneja un gran volumen de documentación respecto al abastecimiento de mercadería para el comisariato?

Alternativas	Frecuencia	Porcentaje
Si	22	82%
No	5	18%
TOTAL	27	100%

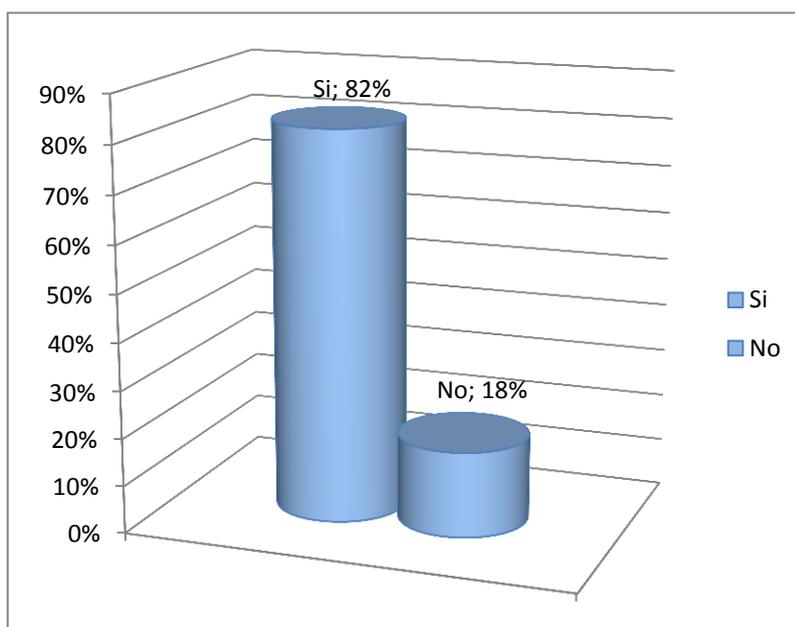


Fig. 3.5.6: Grafico de la Sexta Pregunta

INTERPRETACION:

Algunas personas opinaron que debido a una gran cantidad de productos existente en bodega, ellos deben de llevar la información en sus respectivos registros de manera detallada, dando lugar a una gran recopilación de datos en lo referente a mercadería.

7 ¿Considera usted que debido al manejo automatizado de la información las operaciones realizadas en el comisariato se torna de manera rápida?

Alternativas	Frecuencia	Porcentaje
Si	21	78%
No	3	11%
Tal vez	3	11%
TOTAL	27	100%

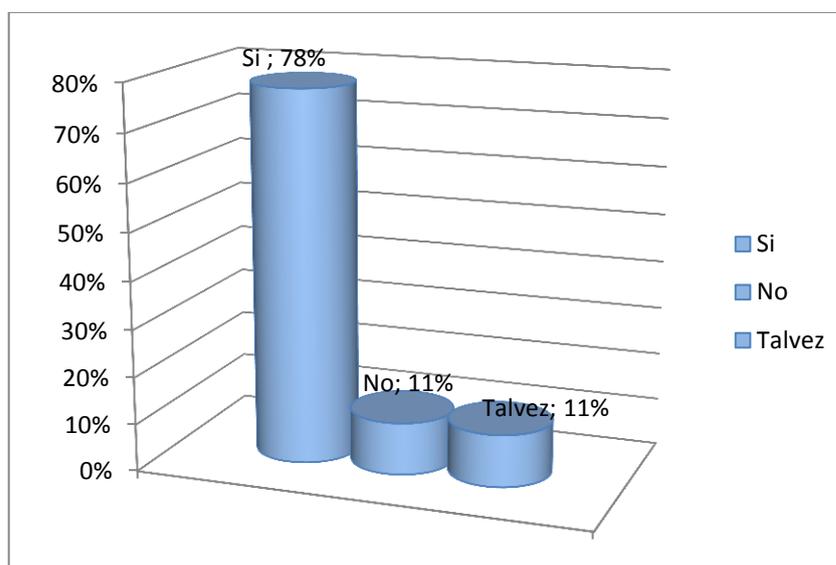


Fig. 3.5.7: Grafico de la Séptima Pregunta

INTERPRETACION:

El 23% determinaron que el número de clientes que visitan el comisariato es verdaderamente considerable, por tal motivo al momento de verificar si el producto solicitado aún está disponible en bodega, la persona encargada debe verificar la existencia del mismo en el sistema, dando lugar a una búsqueda rápida ahorrando tiempo, por todo lo mencionado anteriormente, se da la eficiencia de los procesos ya que cuenta con un sistema que agilice dicha tarea.

8 ¿Es dificultoso buscar información referente a la mercadería que existe en la empresa para saber el stock de productos?

Alternativas	Frecuencia	Porcentaje
Si	3	11%
No	24	89%
TOTAL	27	100%

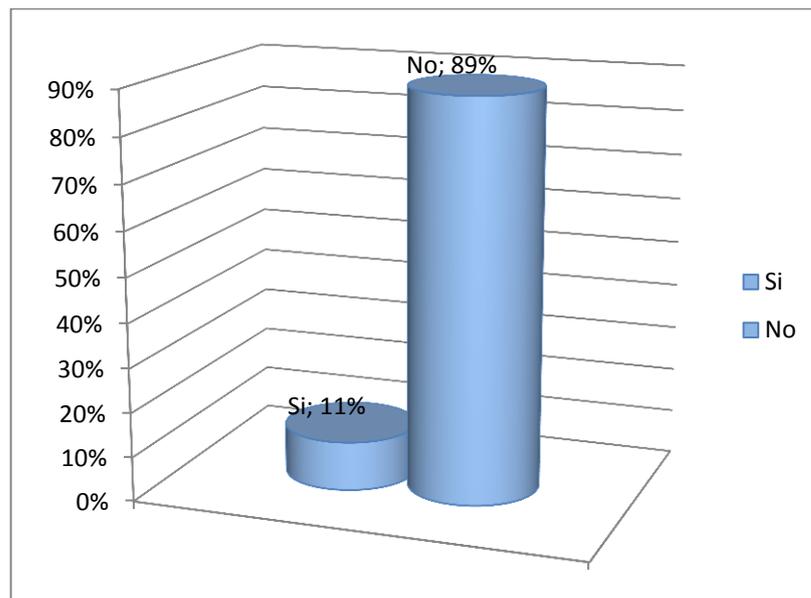


Fig. 3.5.8: Grafico de la Octava Pregunta

INTERPRETACION:

Se mencionó que al contar con un sistema que facilite este trabajo, en el momento que se necesita saber acerca de algún producto, la búsqueda de información sobre el stock existente se lleva a cabo mediante las consultas que el sistema nos proporciona se ha hecho fácil, tener los respectivos datos de cada uno de los productos, teniendo como resultado mucha satisfacción en los mismos ya que no existe dificultad alguna en el momento de solicitar información.

9 ¿Cree usted que el área administrativa conoce el valor a cancelar de los costos que generan la mercadería?

Alternativas	Frecuencia	Porcentaje
Si	20	74%
No	7	26%
TOTAL	27	100%

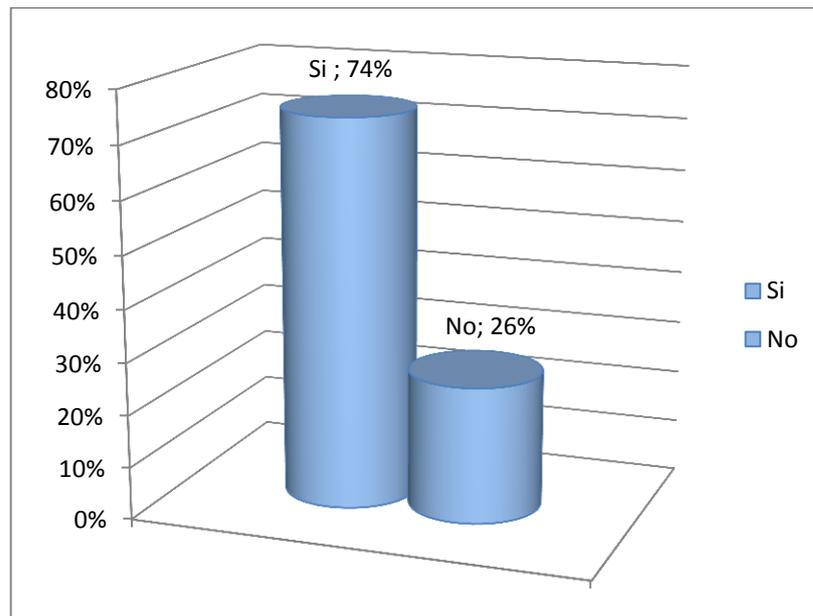


Fig. 3.5.9: Grafico de la Novena Pregunta

INTERPRETACION:

De la encuesta realizada manifestaron que tienen conocimiento sobre el valor real, en lo referente a pérdidas o ganancias. Con el desarrollo del sistema se obtiene valores correctos, convirtiéndose en una ayuda al estar conscientes que los datos obtenidos son confiables.

10 ¿Sería necesario una capacitación informática en el transcurso del manejo del sistema implementado?

Alternativas	Frecuencia	Porcentaje
Acuerdo	25	93%
Desacuerdo	2	7%
TOTAL	27	100%

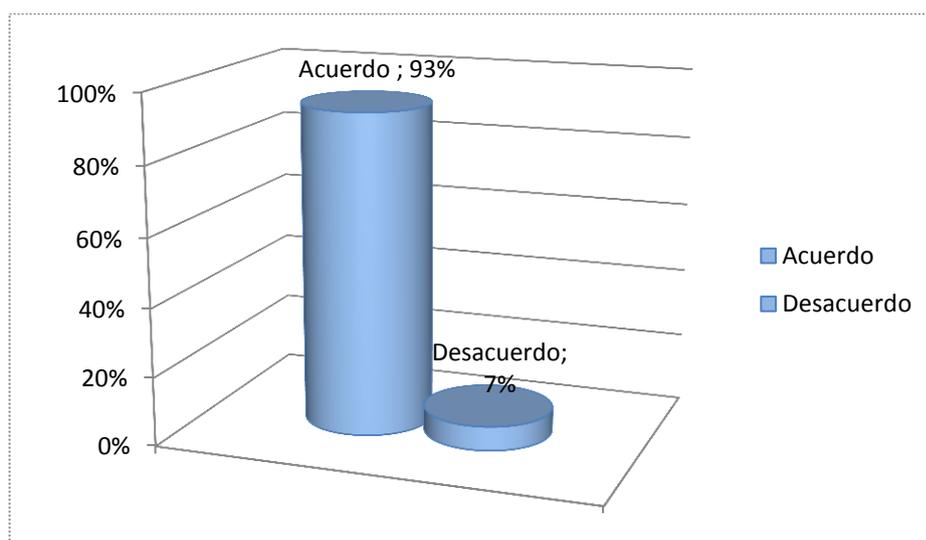


Fig. 3.5.10: Grafico de la Décima Pregunta

INTERPRETACION:

El 27% de los encuestados están de acuerdo que es necesaria recibir una capacitación, para conocer las herramientas de gestión que permitan mejorar el manejo de la bodega y obtener beneficios relacionados con el adecuado almacenamiento, en especial para todos los encargados de bodega, que actualmente tienen bajo su responsabilidad la protección y operación de la bodega esto permitirá una eficiente administración, almacenamiento y control del total de los productos, materias, bienes e insumos de una empresa.

3.7. CONCLUSIONES

Luego del desarrollo de la encuesta realizada al personal de la empresa del comisariato El Popular se concluye que:

El personal que labora en la empresa considera conveniente contar con un sistema que permita controlar el inventario del comisariato el popular brindando así la solución a una necesidad existente en el comisariato.

El dueño de la empresa manifiesta que es factible contar con un sistema administrador de bodega para controlar el inventario del comisariato el popular ofreciendo la respectiva seguridad que garantiza la información, tanto para el acceso del administrador de sistema y el ingreso de la información en la base de datos.

3.8. RECOMENDACIONES

- El administrador de bodega deberá anular la clave de ingreso al sistema e ingresar una nueva, en caso de que el empleado encargado deje de trabajar en la empresa.
- El usuario que tenga acceso al sistema tendrá que cerciorarse de no dejar abierto el sistema en ningún momento, ya que podrían ingresar valores erróneos.

4. DESARROLLO TECNICO DE LA INVESTIGACION

4.1. Introducción

Las empresas continuamente enfrentan el problema de cómo llevar a cabo el proceso de información y asignar sus recursos de la manera más eficiente, para obtener los mejores beneficios, ya que uno de los grandes retos de las empresas es la tecnología.

De manera general, coincide en que nos encontremos en medio de un proceso de avances tecnológicos que se desarrollara aún por algunos años de tal forma que cambiara la manera en que obtenemos y manejamos la información almacenada sistemáticamente. Los problemas que se han observado internamente en la empresas pérdida de tiempo al momento de consultar los datos de todos sus productos debido que no cuenta con las herramientas necesarias que facilite dicha tarea, perdida de información de los registros de control de inventario.

Por estos antecedentes mencionados, se considera pertinente elaborar una solución informática que garantice la seguridad de los servicios y la confiabilidad de los datos que cuente con los elementos requeridos para proveer información adecuada exacta.

De esta forma el sistema de control de inventario ayudara a superar las dificultades que se presentan actualmente en los procesos manuales brindando al administrador seguridad, confiabilidad de la información.

4.2. OBJETIVO DE LA PROPUESTA

4.2.1. OBJETIVO GENERAL

✓ Desarrollar un sistema administrador de inventario que permita cubrir todas las necesidades del departamento de bodega.

4.2.2. OBJETIVOS ESPECIFICOS

✓ Recopilar la información suficiente y necesaria relacionada a los requerimientos del departamento de bodega.

- ✓ Proporcionar información confiable al control de la gestión administrativa.
- ✓ Elaborar una interfaz amigable.

4.3. METODOLOGIA DE DESARROLLO UTILIZADA

La metodología a desarrollar surge ante la necesidad de utilizar una serie de procedimientos, técnicas, y herramientas.

Existen ocasiones en la que se debe elegir una metodología de desarrollo de Software cuando se trata de elaborar sistemas o aplicaciones de diferentes tipos, ya que debemos considerar una metodología que se adopte plenamente a los requerimientos que el usuario tiene por ello nos hemos inclinado por una metodología en espiral.

4.4. ANALISIS PREVIO

Teniendo como referencia los resultados que se ha obtenido en las pruebas y estudios podemos sugerir que la sistematización de consulta tiene grandes ventajas y mayor seguridad.

El comisariato la popular no cuenta en la actualidad con un sistema capaz de controlar información asociados a los productos por lo que ha sido importante diseñar un sistema que permita mejorar la eficiencia en el proceso de control de inventario de los productos.

4.4.1. LISTADO DE REQUERIMIENTOS Y FUNCIONES QUE TENDRA EL SOFTWARE

Requerimientos

Seguridad: La información solo podrá ser manipulada por el personal autorizado que cumpla con todos los requerimientos de seguridad evadiendo que los datos sean modificados por personas no autorizadas.

Confiabilidad: Con la implementación del sistema conseguiremos resultados existentes y confiables ya que todos los datos que se obtiene cuenta con la seguridad del administrador.

Efectividad: La información se obtendrá de manera rápida y confiable sin la necesidad de tener un mayor esfuerzo posible.

Interfaz de usuario: El diseño de la aplicación es sencillo por lo que podrá ser manipulado fácilmente.

Funciones

- ✓ Ingresos
- ✓ Consultas
- ✓ Restricción de Acceso
- ✓ Impresiones

4.5. DISEÑO

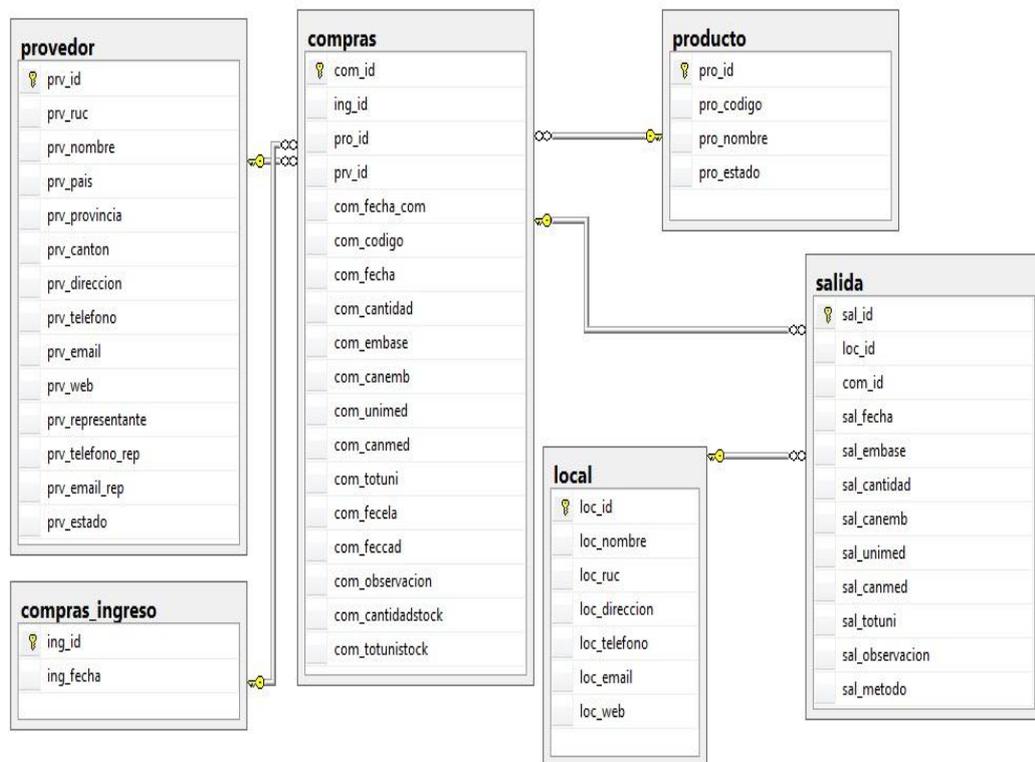
Se basa en cómo está ejecutado el sistema y además la alternativa de los requisitos de los datos de una forma que entienda el analista o programador y lo aremos basados en los siguientes cuatros pasos.

- 1.- Diseño de Datos
- 2.- Diseño Estructurado
- 3.- Diseño Procedimental
- 4.- Diseño de Interfaz

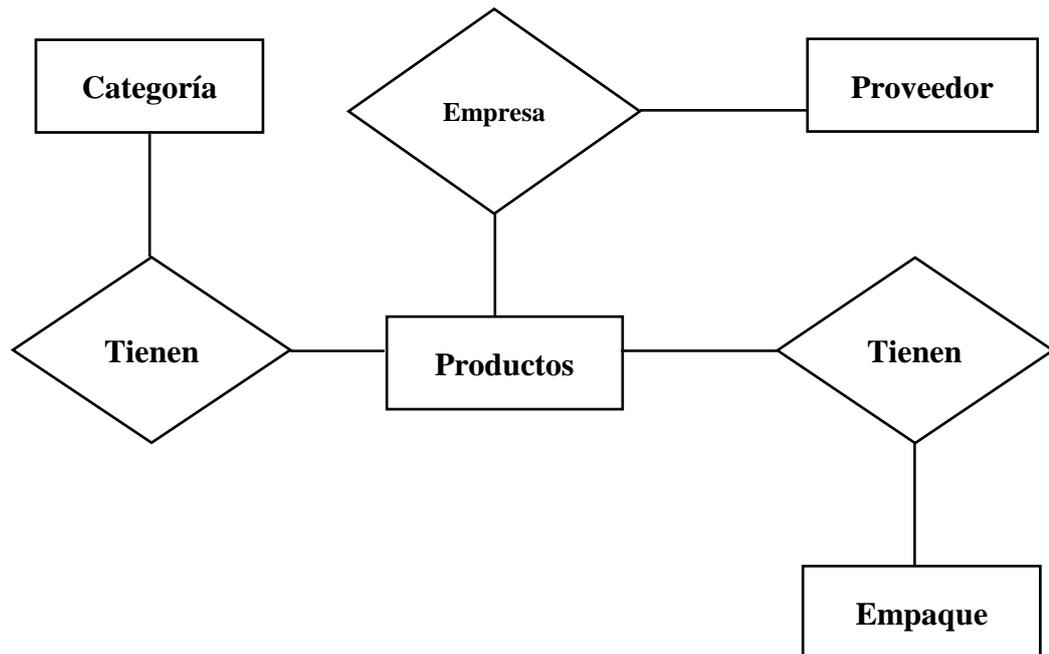
4.5.1. BASE DE DATOS

4.5.1.1. MODELO CONCEPTUAL

La misma notación de modelado de objetos que hemos utilizado para describir la estructura de la pila en programas que se están ejecutando es decir, qué objetos hay y cómo se hallan relacionados por campos puede emplearse de un modo más abstracto para describir el estado espacial de un sistema o del entorno en el que éste opera. Denominaremos a estos modelos "modelos conceptuales", aunque el libro de texto se refiere a ellos como "modelos de datos".



4.5.1.2. MODELO FISICO – MODEO ENTIDAD RELACION



4.5.1.3. DICCIONARIO DE DATOS

A continuación se muestra el diccionario de datos de la base utilizada para el sistema.

Tabla Proveedor: Con sus respectivos campos.

dbo.proveedor: Tabla(abcxyz-pc\.....DF)			
	Nombre de columna	Tipo de datos	Permitir val...
▶	prv_id	numeric(18, 0)	<input type="checkbox"/>
	prv_ruc	nvarchar(13)	<input checked="" type="checkbox"/>
	prv_nombre	nvarchar(50)	<input checked="" type="checkbox"/>
	prv_pais	nvarchar(50)	<input checked="" type="checkbox"/>
	prv_provincia	nvarchar(50)	<input checked="" type="checkbox"/>
	prv_canton	nvarchar(50)	<input checked="" type="checkbox"/>
	prv_direccion	nvarchar(150)	<input checked="" type="checkbox"/>
	prv_telefono	nvarchar(50)	<input checked="" type="checkbox"/>
	prv_email	nvarchar(50)	<input checked="" type="checkbox"/>
	prv_web	nvarchar(50)	<input checked="" type="checkbox"/>
	prv_representante	nvarchar(50)	<input checked="" type="checkbox"/>
	prv_telefono_rep	nvarchar(50)	<input checked="" type="checkbox"/>
	prv_email_rep	nvarchar(50)	<input checked="" type="checkbox"/>
	prv_estado	bit	<input checked="" type="checkbox"/>

Tabla. 4.5.1.3.1: Tabla Proveedor

Tabla Compras: Con sus respectivos campos.

dbo.compras: Tabla(abcxyz-pc\...\...DF)		dbo.merinventario: Diagrama	
	Nombre de columna	Tipo de datos	Permitir val...
▶	com_id	numeric(18, 0)	<input type="checkbox"/>
	ing_id	numeric(18, 0)	<input type="checkbox"/>
	pro_id	numeric(18, 0)	<input type="checkbox"/>
	prv_id	numeric(18, 0)	<input type="checkbox"/>
	com_fecha_com	datetime	<input checked="" type="checkbox"/>
	com_codigo	nvarchar(50)	<input checked="" type="checkbox"/>
	com_fecha	datetime	<input checked="" type="checkbox"/>
	com_cantidad	numeric(18, 2)	<input checked="" type="checkbox"/>
	com_embase	nvarchar(50)	<input checked="" type="checkbox"/>
	com_canemb	numeric(18, 2)	<input checked="" type="checkbox"/>
	com_unimed	nvarchar(50)	<input checked="" type="checkbox"/>
	com_canmed	numeric(18, 2)	<input checked="" type="checkbox"/>
	com_totuni	numeric(18, 2)	<input checked="" type="checkbox"/>
	com_fecela	datetime	<input checked="" type="checkbox"/>
	com_feccad	datetime	<input checked="" type="checkbox"/>
	com_observacion	text	<input checked="" type="checkbox"/>
	com_cantidadstock	numeric(18, 2)	<input checked="" type="checkbox"/>
	com_totunistock	numeric(18, 2)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Tabla. 4.5.1.3.2: Tabla Compras

Tabla Productos: Con sus respectivos campos.

dbo.producto: Tabla(abcxyz-pc\.....DF)			
	Nombre de columna	Tipo de datos	Permitir val...
▶	pro_id	numeric(18, 0)	<input type="checkbox"/>
	pro_codigo	nvarchar(20)	<input checked="" type="checkbox"/>
	pro_nombre	nvarchar(100)	<input checked="" type="checkbox"/>
	pro_estado	bit	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Tabla. 4.5.1.3.3: Tabla Producto

Tabla Salida: Con sus respectivos campos.

dbo.salida: Tabla(abcxyz-pc\...\I...MDF)		
Nombre de columna	Tipo de datos	Permitir val...
sal_id	numeric(18, 0)	<input type="checkbox"/>
loc_id	numeric(18, 0)	<input type="checkbox"/>
com_id	numeric(18, 0)	<input type="checkbox"/>
sal_fecha	datetime	<input checked="" type="checkbox"/>
sal_embase	nvarchar(50)	<input checked="" type="checkbox"/>
sal_cantidad	numeric(18, 2)	<input checked="" type="checkbox"/>
sal_canemb	numeric(18, 2)	<input checked="" type="checkbox"/>
sal_unimed	nvarchar(50)	<input checked="" type="checkbox"/>
sal_canmed	numeric(18, 2)	<input checked="" type="checkbox"/>
sal_totuni	numeric(18, 2)	<input checked="" type="checkbox"/>
sal_observacion	text	<input checked="" type="checkbox"/>
sal_metodo	nvarchar(50)	<input checked="" type="checkbox"/>

Tabla. 4.5.1.3.4: Tabla Salida

Tabla Compras Ingreso: Con sus respectivos campos.

dbo.compras_ingreso: Tabla(abcxy...F)		
Nombre de columna	Tipo de datos	Permitir val...
ing_id	numeric(18, 0)	<input type="checkbox"/>
ing_fecha	datetime	<input checked="" type="checkbox"/>

Tabla. 4.5.1.3.5: Tabla Compras Ingresos

Tabla Compras Ingreso: Con sus respectivos campos.

dbo.local: Tabla(abcxyz-pc\...\IN...DF)		
Nombre de columna	Tipo de datos	Permitir val...
loc_id	numeric(18, 0)	<input type="checkbox"/>
loc_nombre	nvarchar(150)	<input checked="" type="checkbox"/>
loc_ruc	nvarchar(13)	<input checked="" type="checkbox"/>
loc_direccion	nvarchar(150)	<input checked="" type="checkbox"/>
loc_telefono	nvarchar(50)	<input checked="" type="checkbox"/>
loc_email	nvarchar(50)	<input checked="" type="checkbox"/>
loc_web	nvarchar(50)	<input checked="" type="checkbox"/>

Tabla. 4.5.1.3.6: Tabla Local

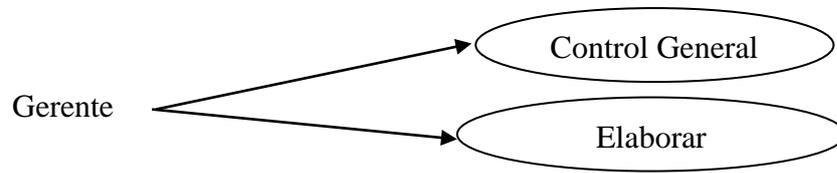
4.5.1.4. SCRIPT DE BASE DE DATOS

```
USE [BODEGA]
GO
/***** Objeto: Table [dbo].[producto]      Fecha de la secuencia de
comandos: 08/28/2012 10:56:54 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[producto] (
    [pro_id] [numeric](18, 0) NOT NULL,
    [pro_codigo] [nvarchar](20) NULL,
    [pro_nombre] [nvarchar](100) NULL,
    [pro_estado] [bit] NULL,
    CONSTRAINT [PK_producto] PRIMARY KEY CLUSTERED
(
    [pro_id] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

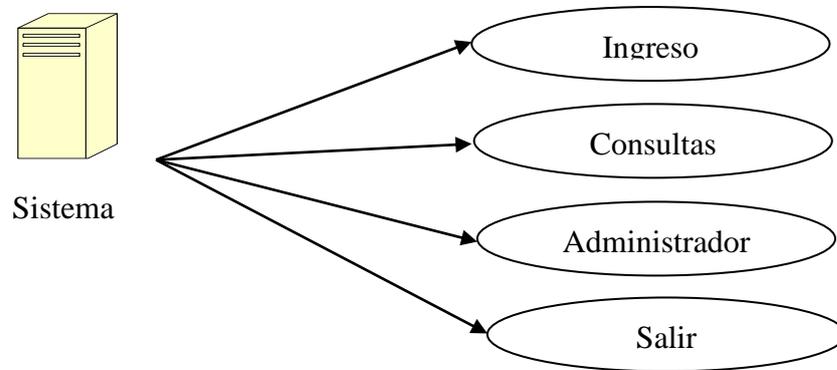
GO
/***** Objeto: Table [dbo].[proveedor]      Fecha de la secuencia de
comandos: 08/28/2012 10:57:00 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[proveedor] (
    [prv_id] [numeric](18, 0) NOT NULL,
    [prv_ruc] [nvarchar](13) NULL,
    [prv_nombre] [nvarchar](50) NULL,
    [prv_pais] [nvarchar](50) NULL,
    [prv_provincia] [nvarchar](50) NULL,
    [prv_canton] [nvarchar](50) NULL,
    [prv_direccion] [nvarchar](50) NULL,
    [prv_telefono] [nvarchar](50) NULL,
    [prv_email] [nvarchar](50) NULL,
    [prv_web] [nvarchar](50) NULL,
    [prv_representante] [nvarchar](50) NULL,
    [prv_telefono_rep] [nvarchar](50) NULL,
    [prv_email_rep] [nvarchar](50) NULL,
    [prv_estado] [bit] NULL,
    CONSTRAINT [PK_proveedor] PRIMARY KEY CLUSTERED
(
    [prv_id] 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
```

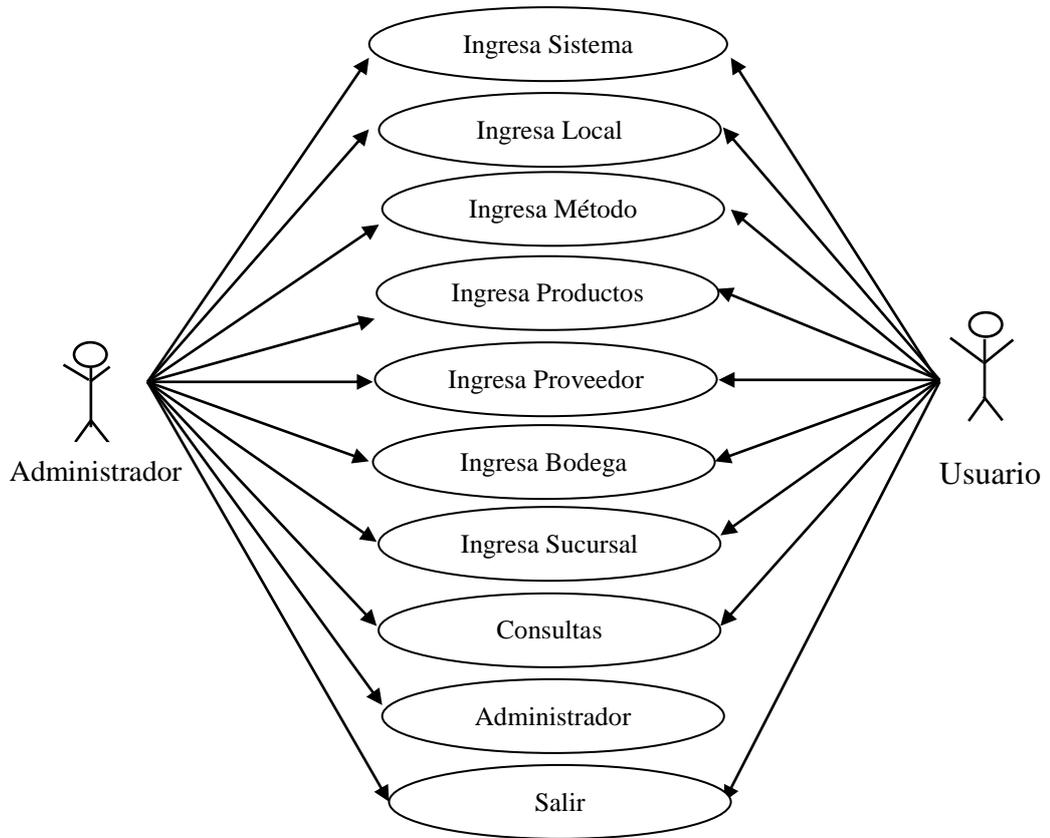
4.6. DIAGRAMAS DE CASOS DE USO



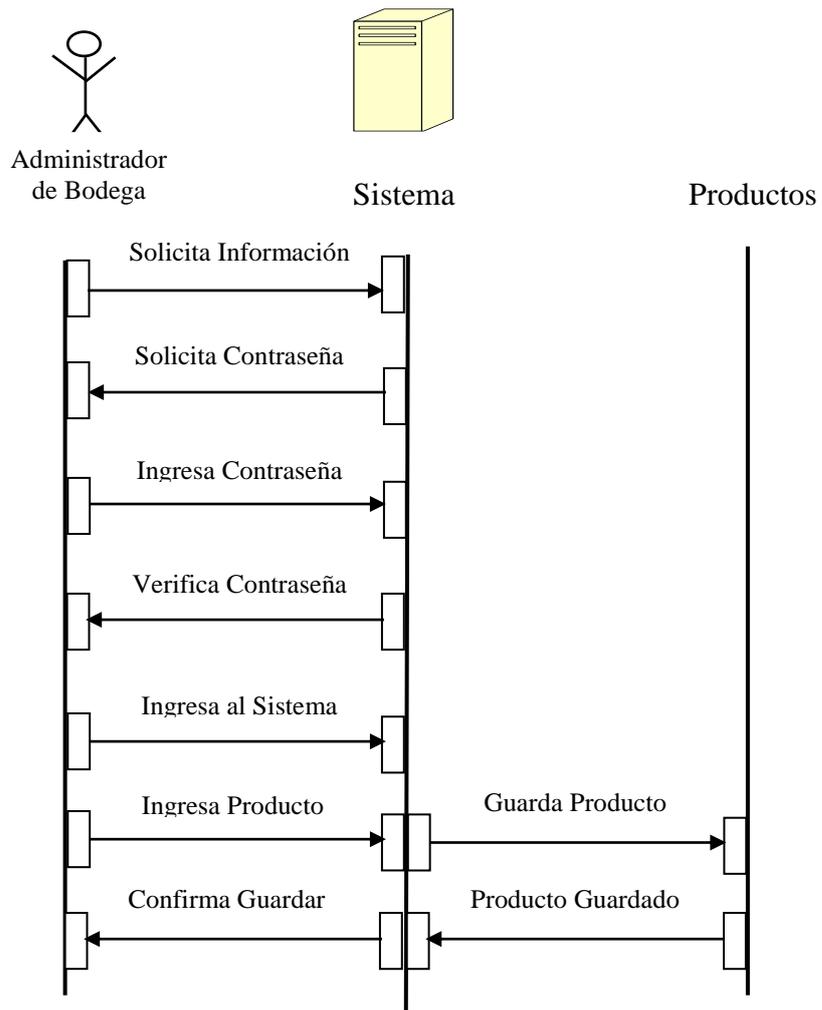
Sistema Administrador de Bodega y Control de Inventario



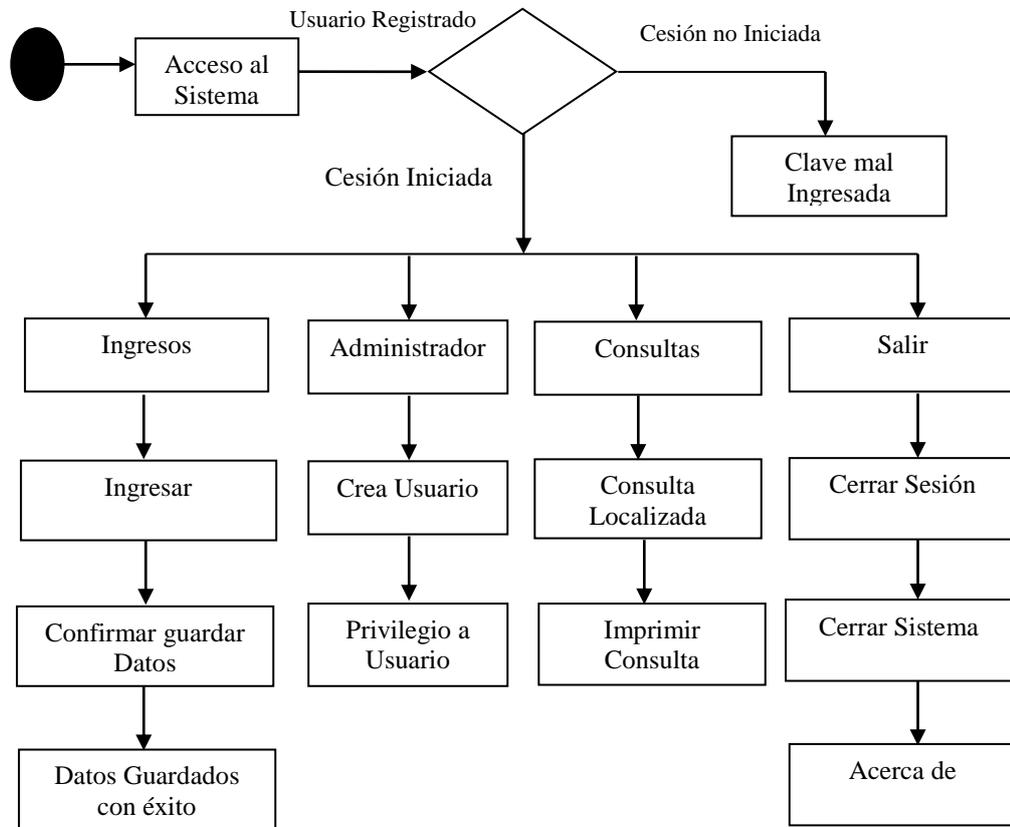
Acceso al Sistema



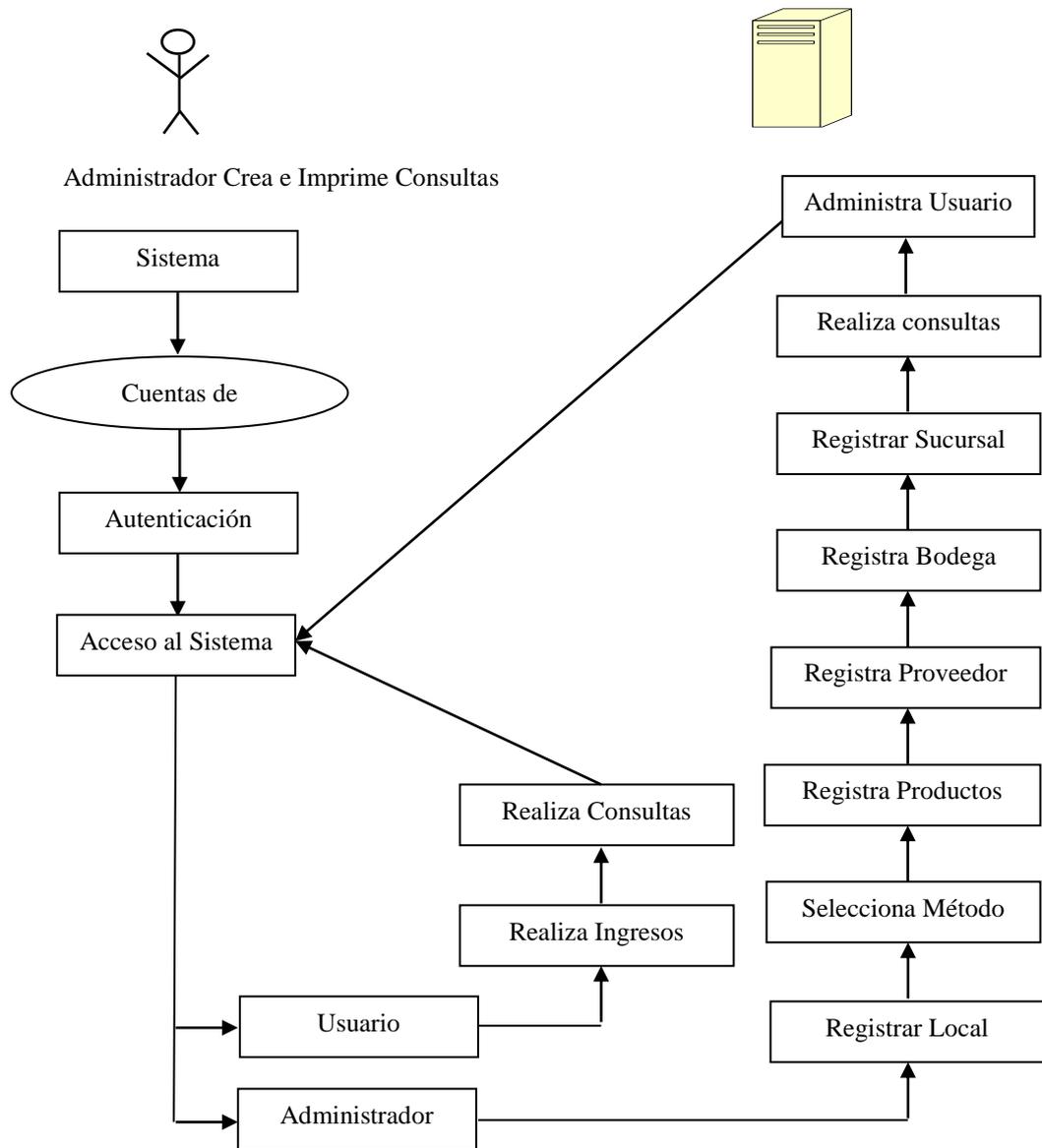
4.7. DIAGRAMAS DE SECUENCIA



4.8. DIAGRAMAS DE ACTIVIDAD



4.9. DIAGRAMAS DE DESPLIEGUE



4.10. DISEÑO DE INTERFACES

Entre los aspectos más principales a tener en cuenta es la interfaz de usuario. La misma que tiene que ser explícita y estar confeccionada con colores que no sean escandalosos ni ocupen la atención del usuario.

Los formularios de ingreso de datos están diseñados de una manera sencilla con la cantidad de elementos que se requiere al fin de hacer más fácil para el usuario el trabajo con los mismos.

El diseño de muestra interfaz es la siguiente:

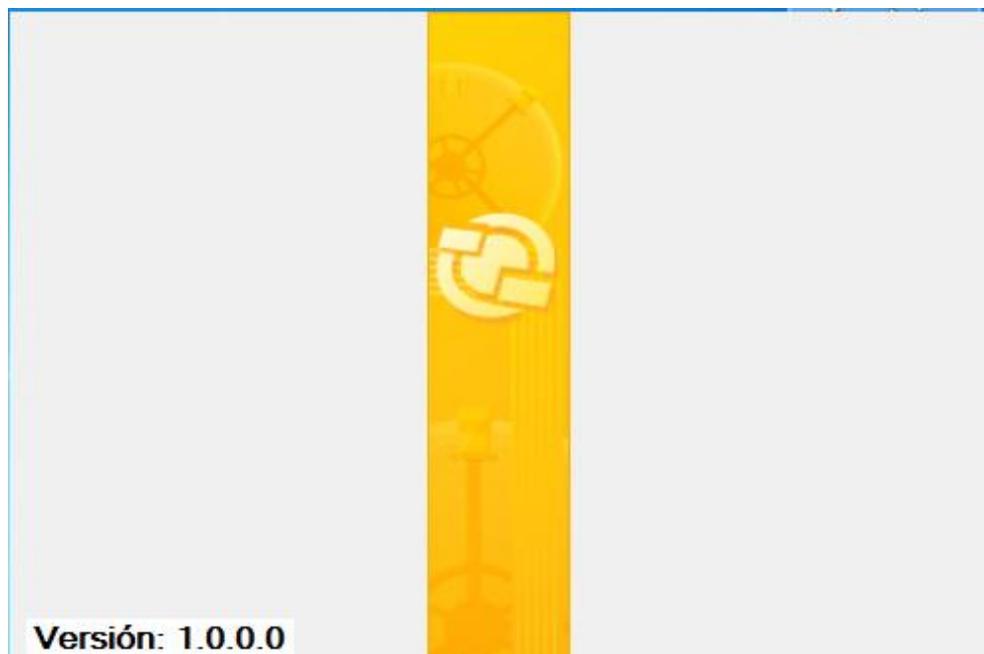


Fig. 4.10.1: Ventana de Presentación

La siguiente pantalla nos permitirá escribir el usuario y la contraseña.

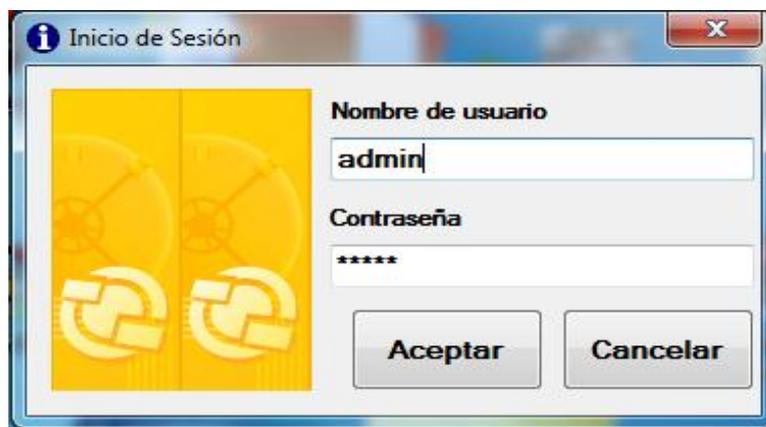


Fig. 4.10.2: Ventana Inicio de Sesión

Luego ingresamos al menú.



Fig. 4.10.3: Ventana Menú de Usuario

Estas son las opciones de los diferentes menús.



Fig. 4.10.4: Opciones del Menú / Ingresos

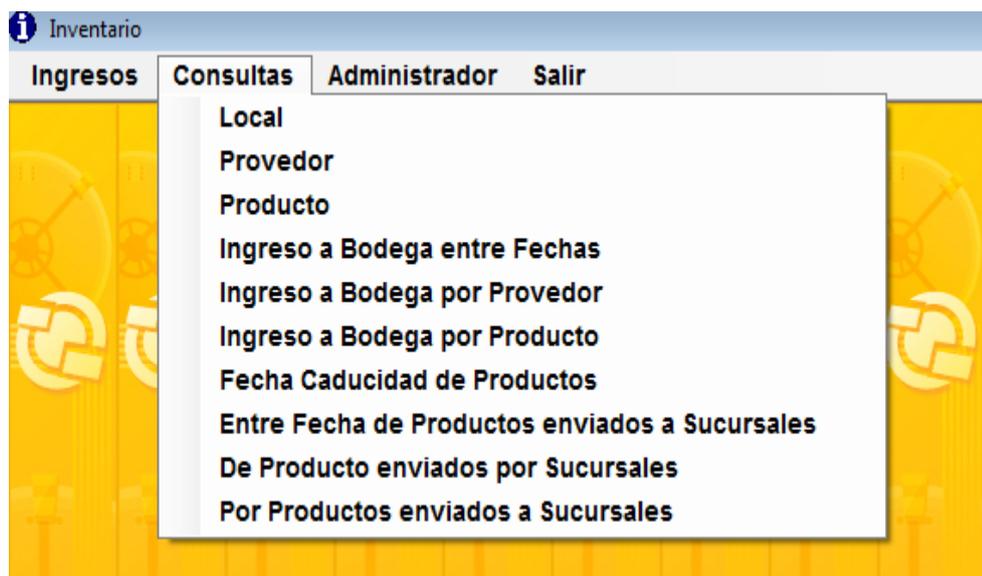


Fig. 4.10.5: Opciones del Menú /Consultas



Fig. 4.10.6: Opciones del Menú / Administrador

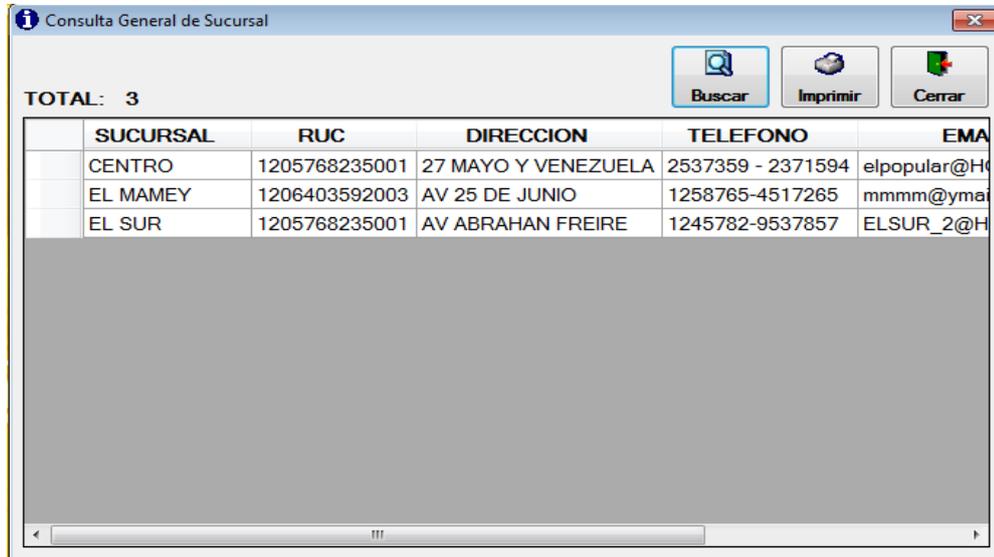


Fig. 4.10.7: Opciones del Menú / Salir

4.11. DISEÑO DE SALIDAS

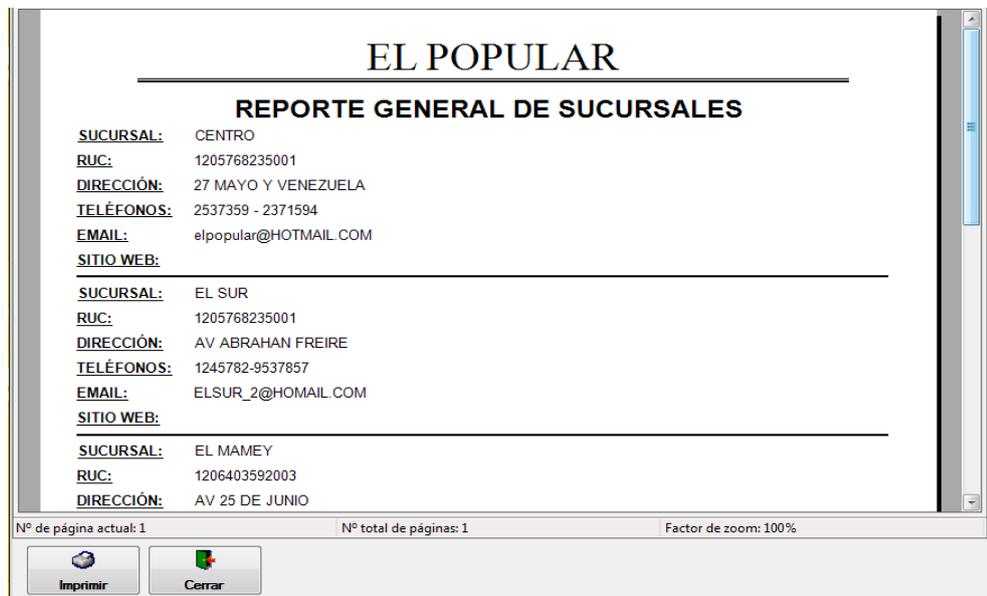
Permite visualizar toda la información de los ingresos realizados mediante consultas.

Aquí tenemos una consulta con sus respectivos datos.



SUCURSAL	RUC	DIRECCION	TELEFONO	EMAIL
CENTRO	1205768235001	27 MAYO Y VENEZUELA	2537359 - 2371594	elpopular@H
EL MAMEY	1206403592003	AV 25 DE JUNIO	1258765-4517265	mmmm@yma
EL SUR	1205768235001	AV ABRAHAN FREIRE	1245782-9537857	ELSUR_2@H

Fig. 4.11.1: Ejemplo de Consulta / Sucursal



EL POPULAR

REPORTE GENERAL DE SUCURSALES

SUCURSAL: CENTRO
RUC: 1205768235001
DIRECCIÓN: 27 MAYO Y VENEZUELA
TELÉFONOS: 2537359 - 2371594
EMAIL: elpopular@HOTMAIL.COM
SITIO WEB:

SUCURSAL: EL SUR
RUC: 1205768235001
DIRECCIÓN: AV ABRAHAN FREIRE
TELÉFONOS: 1245782-9537857
EMAIL: ELSUR_2@HOMAIL.COM
SITIO WEB:

SUCURSAL: EL MAMEY
RUC: 1206403592003
DIRECCIÓN: AV 25 DE JUNIO

Nº de página actual: 1 Nº total de páginas: 1 Factor de zoom: 100%

Imprimir **Cerrar**

Fig. 4.11.2: Ejemplo de Reportes / Sucursal

Consulta General de Proveedor

TOTAL: 3

Buscar Imprimir Cerrar

RUC	NOMBRE	PAIS	PROVINCIA	CANTON	DIRECCION
1204330078003	EL RANCHITO	ECUADOR	EL ORO		
1204330078001	EL VENDE MAS				
1206403592002	NESTLE	ECUADOR	GUAYAS		

Fig. 4.11.3: Ejemplo de Consulta / Proveedor

EL POPULAR

REPORTE GENERAL DE PROVEDOR

RUC: 1204330078001
 PROVEDOR: EL VENDE MAS
 PAIS:
 PROVINCIA:
 CANTÓN:
 DIRECCIÓN:
 TELEFONO:
 EMAIL:
 WEB:
 REPRESENTANTE:
 TELÉFONO:
 EMAIL:
 ACTIVO: Sí

RUC: 1204330078003
 PROVEDOR: EL RANCHITO

Nº de página actual: 1 Nº total de páginas: 1+ Factor de zoom: 100%

Imprimir Cerrar

Fig. 4.11.4: Ejemplo de Reportes / Proveedor

Consulta General de Productos

TOTAL: 4

Buscar Imprimir Cerrar

	CODIGO	NOMBRE	ACTIVO
	ACE11	ACEITE GIRASOL	<input checked="" type="checkbox"/>
	ARR4	ARROZ	<input checked="" type="checkbox"/>
	MAN8	MANTEQUILLA	<input checked="" type="checkbox"/>
	TOM1	TOMATES	<input checked="" type="checkbox"/>

Fig. 4.11.5: Ejemplo de Consulta / Productos

EL POPULAR

REPORTE GENERAL DE PRODUCTOS

CÓDIGO	NOMBRE	ACTIVO
	ARROZ	Si
ACE13	ACEITE GIRAZOL	Si
AZU14	AZUCAR	Si
CAF15	CAFE	Si
LEC17	LECHE	Si
YOG18	YOGURT	Si
PAP19	PAPEL HIGIENICO FLOR	Si
SAL20	SALSA DE TOMATE MAGUI	Si
MAY21	MAYONESA MAGUI	Si
CHO23	CHOCOLATE PALITO	Si
MAN24	MANICHOS	Si
CHI25	CHICLES TUMIX	Si

Nº de página actual: 1 Nº total de páginas: 1 Factor de zoom: 100%

Imprimir Cerrar

Fig. 4.11.6: Ejemplo de Reportes / Productos

4.12. DESARROLLO

Mostrar información:

- Etiquetas
- Grupos de objetos

Introducción de datos:

- Cajas de texto

Órdenes:

- Botones para activar la acción a realizar por el usuario

Opciones:

- Para seleccionar una opción entre varias
- Activar una opción en concreto

Selección de datos:

- Cajas desplegadas
- Listas de datos

Especializados:

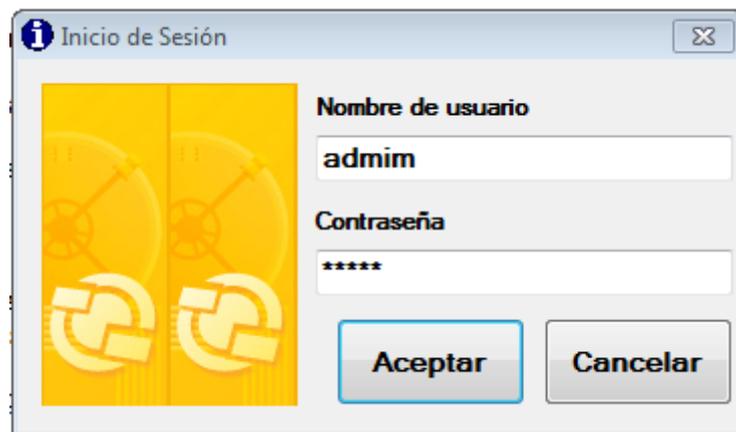
- Impresión
- Menús y barras

4.12.1. PRUEBAS

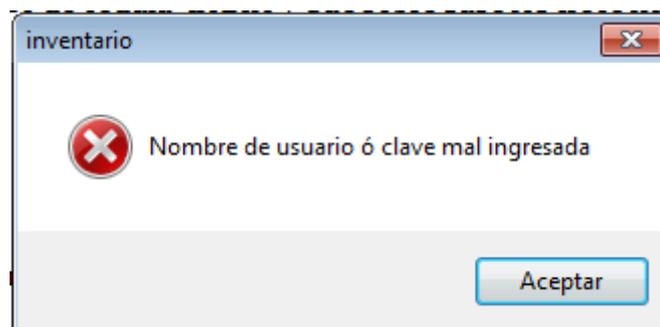
El objetivo de las pruebas es presentar información sobre la calidad del producto a las personas responsables de este. Las pruebas realizadas se efectuaron en conjunto con el gerente y la persona que va a manejar el sistema.

En esta sección aparecen tres casos de prueba y los resultados de cada uno sobre las diferentes funciones del programa. Obsérvese que:

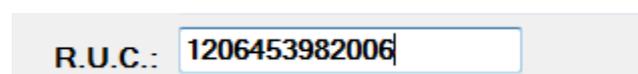
Prueba1: Ingreso de Contraseña



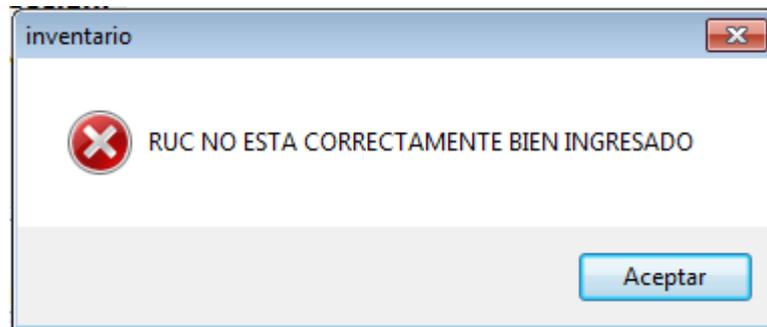
Presionando el botón **Aceptar**, verifica que la contraseña y el nombre de usuario ingresados coincidan con los datos guardados en el sistema.



Prueba2: Ingreso del Ruc



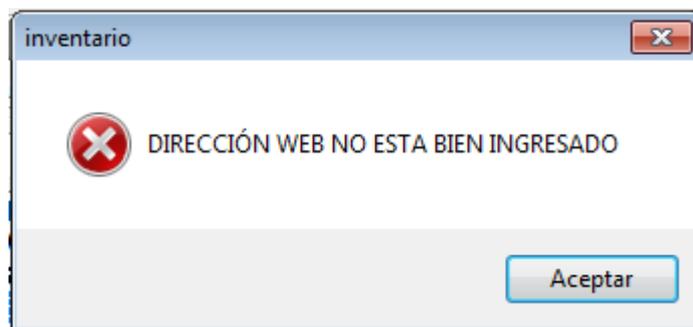
Solo se admite el ingreso de un ruc real y que no ha sido ingresado anteriormente.



Prueba3: Ingreso de la Página Web



Esta validado para que el usuario ingrese los datos de forma correcta

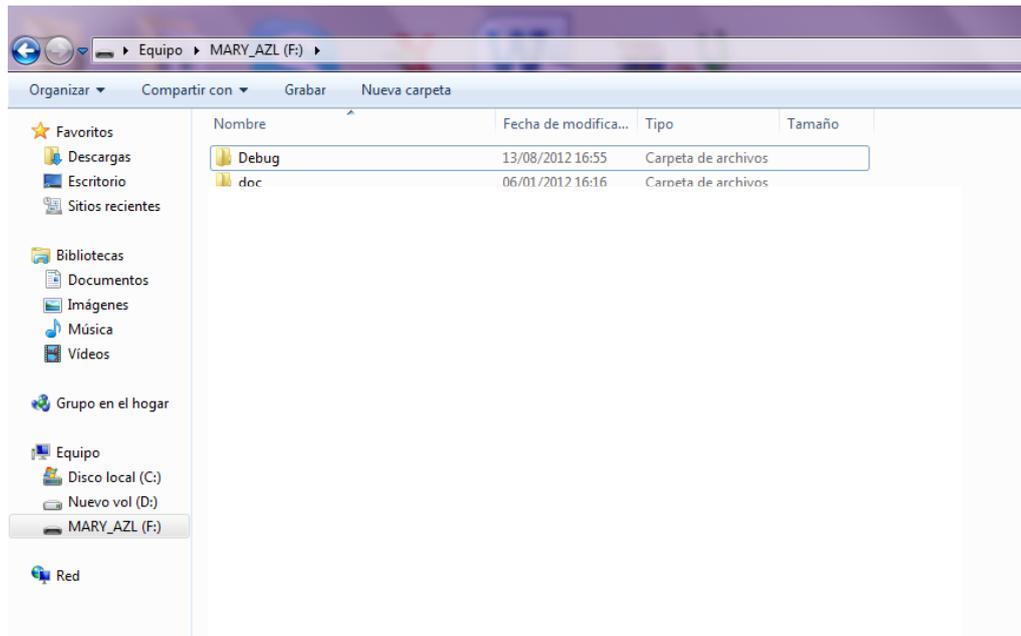


Todas las pruebas realizadas fueron exitosas

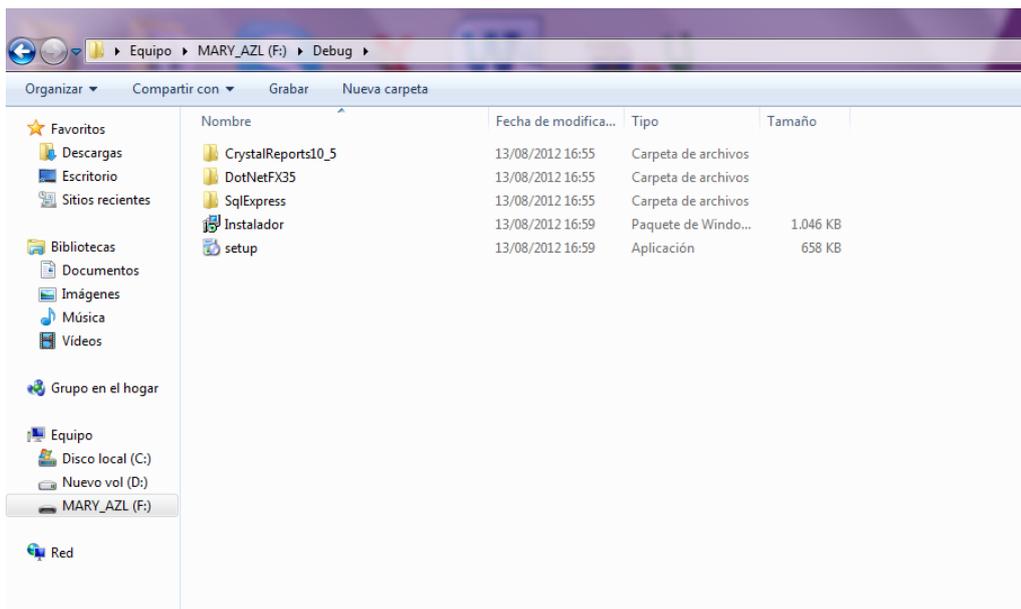
4.12.2. IMPLEMENTACION DEL SISTEMA

Una vez obtenido los objetivos de prueba de manera satisfactoria, es posible implementar el sistema. En la presente sección, se explicara, describirán y mostraran la manera de ejecutar el sistema.

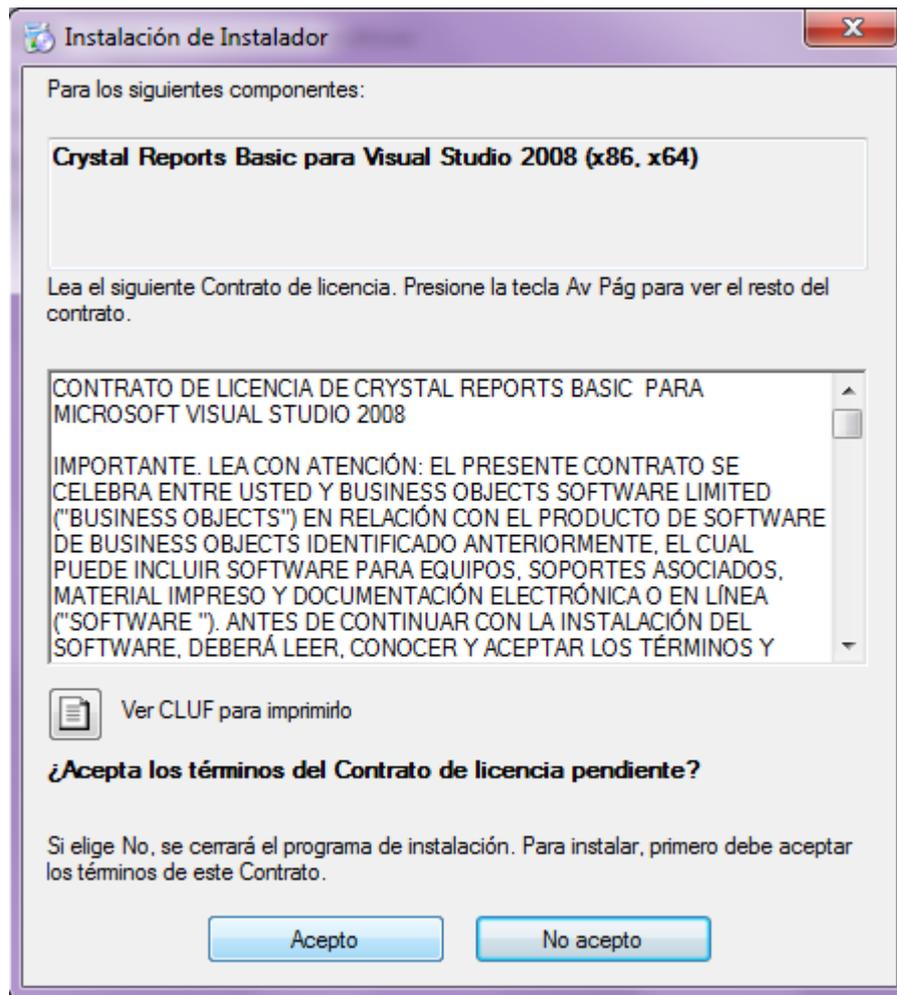
En la siguiente pantalla vamos a ejecutar el programa, damos clic en la carpeta Debug.



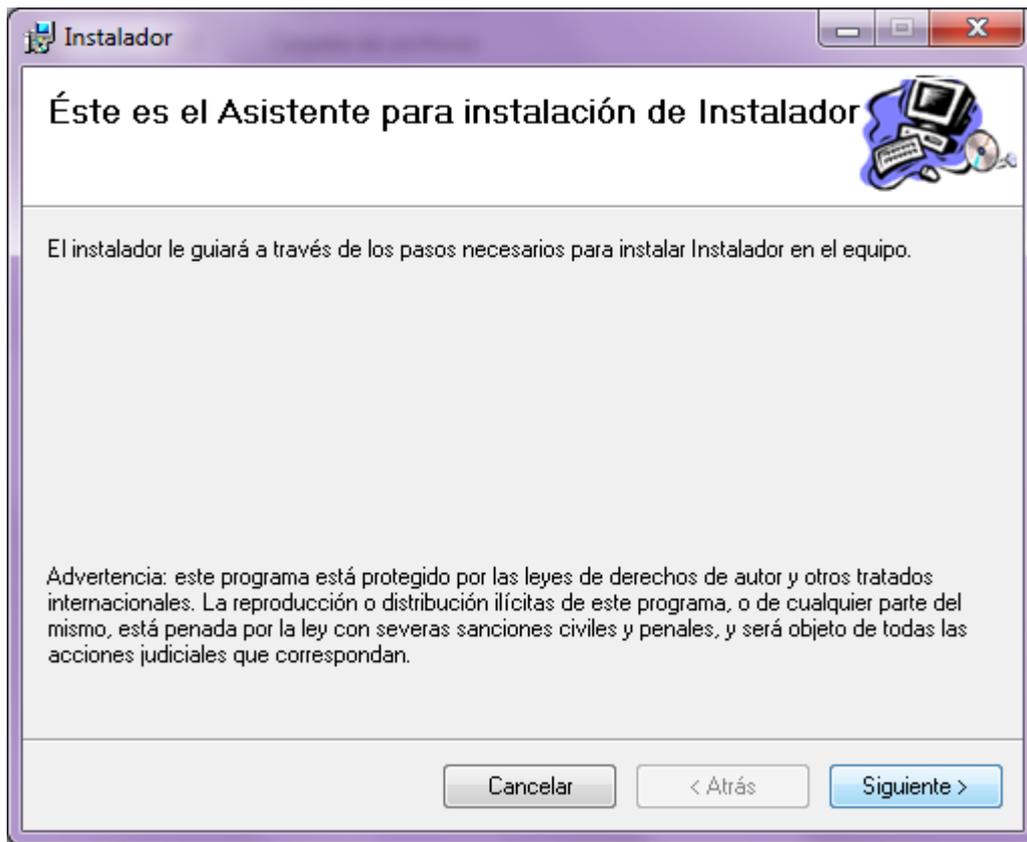
Se nos mostrara carpetas donde instalaremos cada una de ellas. Una vez realizada dicha acción se procede a instalar el setup.



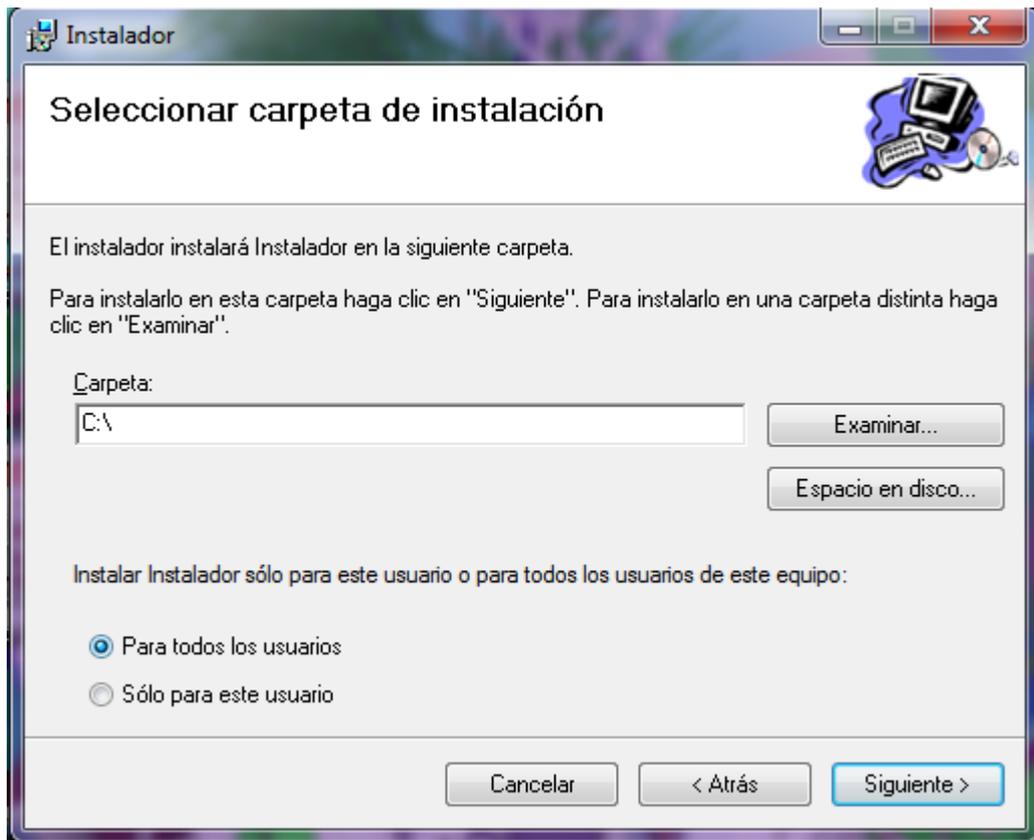
Se muestra la siguiente pantalla al presionar doble clic sobre setup. Donde seleccionamos el botón **Acepto**.



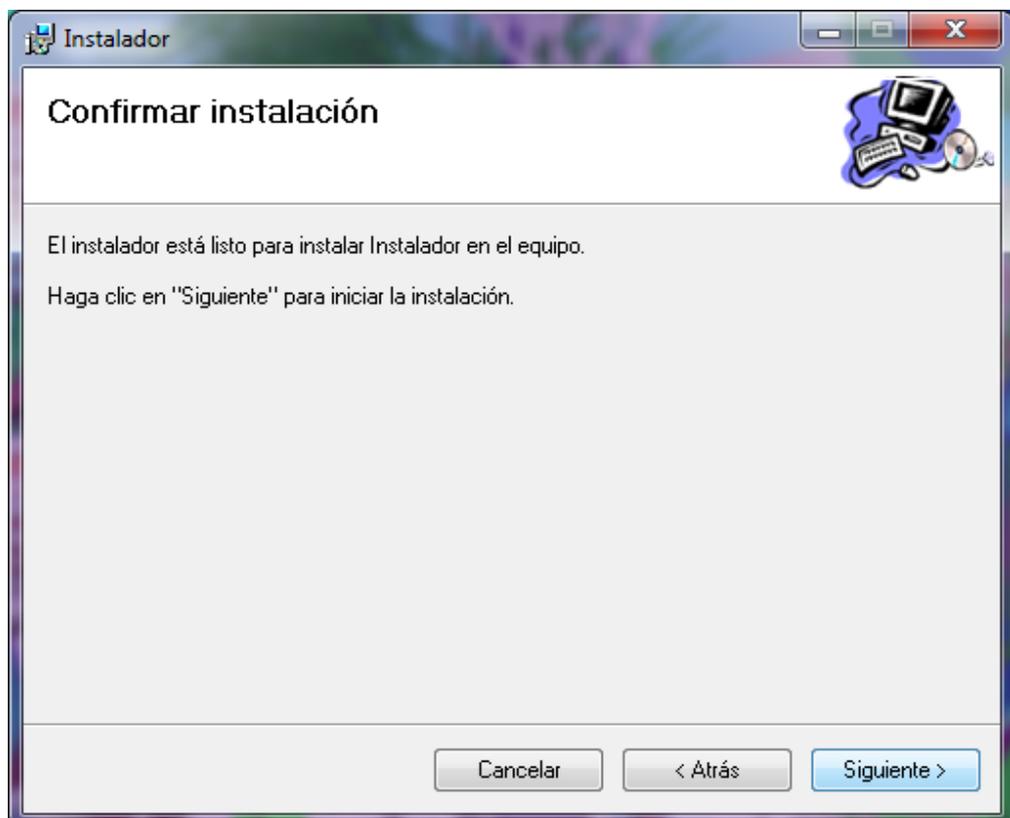
En esta seleccionamos el botón siguiente para proceder con la instalación.



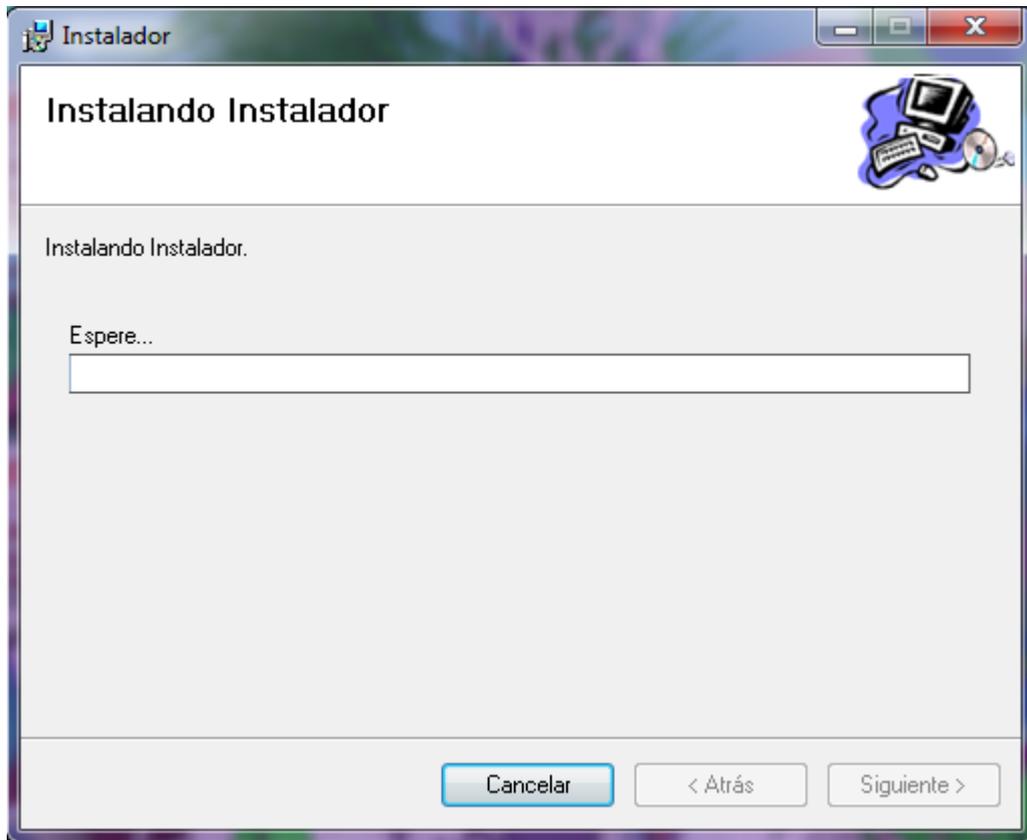
Seleccionamos la ubicación donde vamos a instalar el sistema escogemos la opción Para todos los usuario. Y damos clic en el botón siguiente.



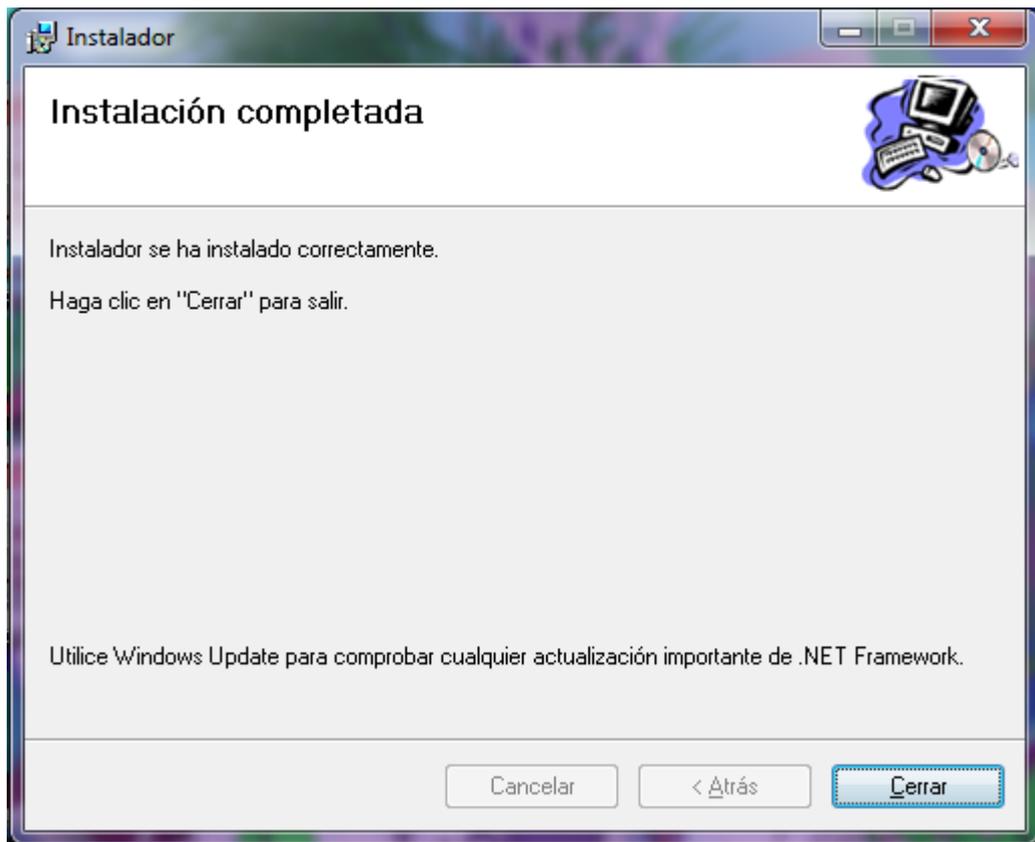
En esta ventana se confirma la instalación del sistema, damos clic en el botón siguiente.



Se presenta esta ventana donde esperamos unos segundos el proceso de instalación.



Una vez completado el proceso se muestra la siguiente ventana confirmando que la instalación ha sido completada. Y damos clic en cerrar.



Se muestra en el escritorio el icono del sistema.



4.12.2.1. REQUERIMIENTOS DE HARDWARE

Hardware como mínimo

- Disco Duro de 320 Gb
- Memoria de 512Mb
- Procesador AMD

4.12.2.2. REQUERIMIENTOS DE SOFTWARE

- Plataforma Visual Studio 2008 y sus componentes de,
- Base de Datos SQL Server 2005
- Windows 7

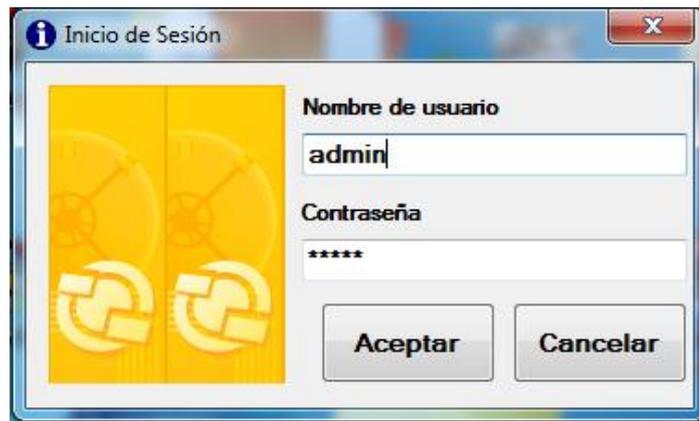
4.12.2.3. PROCESO DE INSTALACION

- CrystalReports10_5
- DotNetFX35
- SqlExpress
- setup

4.12.2.4. SEGURIDADES

El sistema cuenta con restricción de usuario, donde el administrador tiene acceso a todos los procesos realizados dentro del sistema permitiendo la utilización de cada uno de los menús. El usuario común solo tiene acceso a los módulos asignados por el administrador.

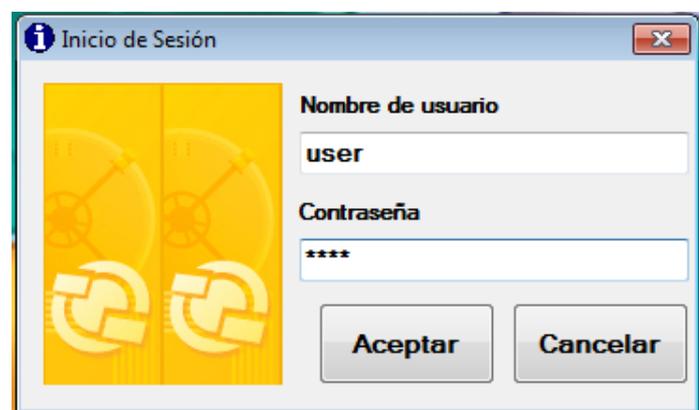
Inicio de sesión ejecutado por el administrador:



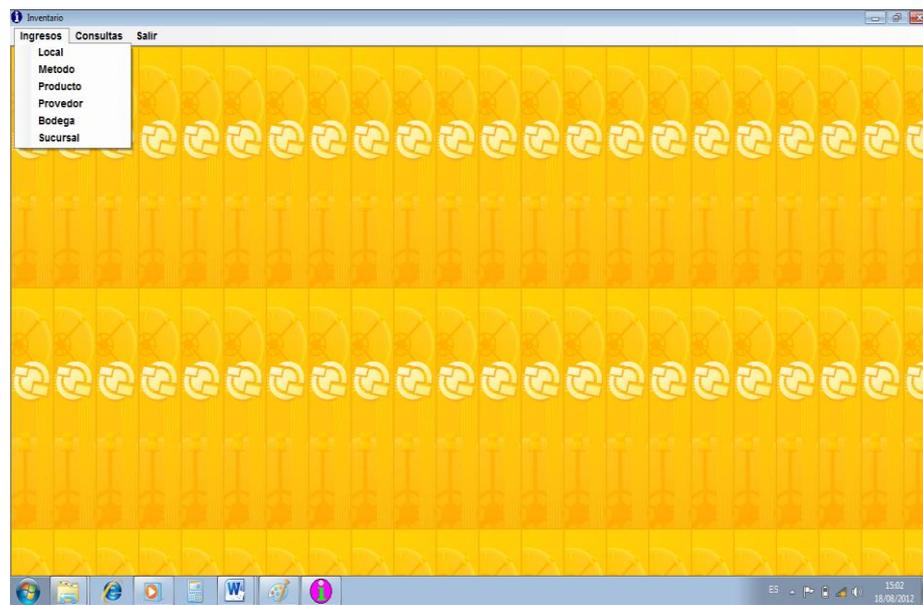
Esta ventana muestra el acceso a todas las opciones por parte del administrador.



Inicio de sesión ejecutado por del usuario:



Esta ventana muestra el acceso a las opciones por parte del usuario.



4.13. CONCLUSIONES Y RECOMENDACIONES PARA UNA EFICIENTE IMPLEMENTACION

4.13.1. CONCLUSIONES

Este sistema es de gran importancia para el departamento de bodega, podemos decir que el sistema cumple con lo necesario para poder realizar, ingresos, consultas, modificaciones, y de tal manera disponer de una completa información que permita administrar datos de los productos que se encuentran en bodega. Además de contener todas las herramientas importantes, es de fácil manipulación y posee una base de datos con gran espacio para almacenar la información necesaria.

4.13.2. RECOMENDACIONES

- Al personal encargado de utilizar el sistema tenga un completo aprendizaje para un óptimo manejo de la aplicación.
- El administrador de bodega deberá anular la clave de ingreso al sistema e ingresar una nueva, en caso de que el empleado encargado deje de trabajar en la empresa.

BIBLIOGRAFÍAS

Autor: Lic. Víctor Manuel Chávez Gaona

Tema: Metodología Orientada

Autor: Dr. Manuel Arias

Tema: Ciclo de vida del Software

Autor: José R. Álvarez

Tema: Ciclo de vida en Espiral

Autor: Ms. Danny Farfan

Tema: Visual Studio

Autor: Lic. Ángel Esteban Núñez

Tema: Visual Studio 2008

LINKOGRAFÍAS

- <http://www.monografias.com/trabajos13/metomt/metomt.shtml>/Víctor Manuel Chávez Gaona
- <http://es.scribd.com/doc/73772826/7/Modelos-de-AOO-y-DOO>
- <http://www.xuletas.es/ficha/metodologias-estructuradas/>
- <http://www.di.uniovi.es/~dani/asignaturas/apuntes-leccion4.PDF>
- http://www.atc.us.es/asignaturas/fi/cursode_c/Estructuras_iterativas.html
- <http://es.wikipedia.org/wiki/Software>
- <http://ciclodevidasoftware.wikispaces.com/CICLO+DE+VIDA+INCREMENTAL>/Manuel Arias
- <http://www.ia.uned.es/ia/asignaturas/adms/GuiaDidADMS/node10.html/> José R. Álvarez
- <http://www.slideshare.net/guest37183b/modelo-de-ciclo-de-vida-en-espiral>
- <http://es.scribd.com/doc/28268501/Programacion-en-ASP>
- <http://www.pcmasm.com/viewtopic.php?f=88&t=4002/Markust>
- <http://www.slideshare.net/Rokr02/primeros-pasos-375966>
- <http://es.scribd.com/doc/100531540/106/Controles-y-eventos>
- http://www.taringa.net/posts/apuntes-y-monografias/4618296/Visual-Studio-_Ejercicios.html
- <http://www.programatium.com/vbnet.htm>
- [http://es.scribd.com/doc/61199544/12/Ambito-de-las-variables/Danny Farfan](http://es.scribd.com/doc/61199544/12/Ambito-de-las-variables/DannyFarfan)

- http://www.sc.ehu.es/sbweb/webcentro/automatica/web_8051/Contenido/cursoc51/Capitulo%2010/funciones_intrInsecas.htm
- <http://www.buenastareas.com/materias/funciones-en-visual-basic/0>
- <http://www.monografias.com/trabajos2/guiavb/guiavb2.shtml>
- http://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=140:tipos-de-variables-en-visual-basic-integer-string-variant-boolean-etc-ejemplos-cu00308
- <http://www.eggheadcafe.com/Ángel Esteban Núñez>
- <http://teleformacion.edu.aytolacoruna.es/PASCAL/document/tipos.htm>
- <http://es.scribd.com/doc/54966596/Manual-de-Visual>
- <https://www.itescam.edu.mx/Juan Germán Gama Aguilar>
- <http://issuu.com/Joel Martin Chuco Marrufojmchm19@hotmail.com>
- <http://www.programatium.com/vbnet.htm>
- Las desventajas que tiene Visual Basic anteriores a VB.NET
- http://es.wikipedia.org/wiki/Base_de_datos
- <http://www.monografias.com/trabajos73/microsoft-sql-server/microsoft-sql-server2.shtml/Diego Martin Ramos Ortega>
- <http://www.devjoker.com/html/Operaciones-con-conjuntos.html>
- <http://www.utm.mx/~rruiz/cursos/PE/notas.pdf/ErnestMart>
- <http://personal.lobocom.es/claudio/sql001.htm>

ANEXOS