

Universidad de las Ciencias Informáticas

Facultad 6



**TRABAJO DE DIPLOMA PARA OPTAR POR EL  
Título de Ingeniero en Ciencias Informáticas**

**Título**

Herramienta para el análisis económico financiero a partir de la Base de datos del Sistema VERSAT Sarasola

**Autor**

José Alberto Alomá Santos

**Tutor**

Ing. Dania Mora Valdés.

**Co-tutor**

Ing. Dayris Espinosa Ronquillo.

La Habana, Junio 2011.  
“Año 53 de la Revolución”

**DEDICATORIA**

*A mis padres: Alomá y Tania*

*A mis hermanos: Karen y Yeve*

*A mis amigos (los de verdad)*

**EN ESPECIAL A MI QUERIDA ESPOSA Y AMADO HIJO**

*Anny Maylén y Aarón Daniel*

## **AGRADECIMIENTOS**

A mi papá que siempre está cuando más lo necesito, que siempre me ha exigido y ha hecho de mí lo que hoy soy. A mi madre que es el ser más dulce que existe sobre la faz de la tierra. A mis dos hermanos que han sido mi meta, a mi querida esposa Anny Maylén que siempre me ha apoyado en las buenas y en las malas y en especial a mi motor impulsor, esa pequeña personita que es mi motivo de inspiración, mi mayor orgullo y por quien quiero que mis logros hoy sean pasos a seguir en su mañana, mi hijo AARÓN DANIEL ALOMÁ ACOSTA.

A mis tutores de tesis Dania Mora Valdés y en especial Dayris Espinosa Ronquillo por su generosidad al brindarme la oportunidad de recurrir a su capacidad y experiencia en un marco de confianza, afecto y amistad. Al tribunal por la seriedad puesta durante la investigación y por la ayuda brindada. A todos mis amigos, Alex, Elvira, Aimee mi cuñada, Eliezer y esposa, Yeni, Yadriel, Wilfredo, Julio Collado mi alumno, Adrián (Jerry), en especial a Alberto Menéndez que siempre ha estado ahí, cuando más lo he necesitado y ha sido mi trampolín de salida y amigo de los de verdad. A Gregorio y a su esposa por lograr a lo largo de mi carrera llegar a ocupar un lugar importante en mi vida. A Gerardo por sus incansables viajes a la escuela. A mi tía Marja y Frank que han demostrado ser padres para mí, porque son personas tan especiales de las pocas que quedan.

Por último, agradecerle a Joshaner García que ha demostrado ser uno más de la familia, por todos los momentos que hemos compartido que han reafirmado que más que amigos somos hermanos, mil gracias por todo.

## **DECLARACIÓN DE AUTORÍA**

Declaro que soy el único autor de este trabajo y autorizo a la Empresa Comercializadora y de Servicios de Productos Universales de Las Tunas y a la Facultad 6 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

José Alberto Alomá Santos  
Autor

---

Ing. Dania Mora Valdés  
Tutor

## **DATOS DE CONTACTO**

**Nombre y apellidos:** Ing.Dania Mora Valdés.

**Correo electrónico:** [dmora@uci.cu](mailto:dmora@uci.cu)

**Categoría docente:** Instructor Recién Graduado.

**Año de graduación:** 2008.

**Profesión:** Ingeniero en Ciencias Informáticas.

**Breve descripción:** Jefa del proyecto SyGMe: Sistema para la Gestión y Publicación de Metadatos Geográficos y profesora de Seguridad Informática.

**Nombre y apellidos:** Ing.Dayris Espinosa Ronquillo.

**Correo electrónico:** [dronquillo@uci.cu](mailto:dronquillo@uci.cu)

**Categoría docente:** Instructor Recién Graduado.

**Año de graduación:** 2008.

**Profesión:** Ingeniero en Ciencias Informáticas.

**Breve descripción:** Asesora de planificación y control de la facultad 1, profesora de ISW 1 y 2.

## **RESUMEN**

Producto del vertiginoso auge de las Tecnologías de la Información y las Comunicaciones, las cuales han impuesto un nuevo orden mundial al proceso de gestión de la información, la mayoría de las empresas de la actualidad optan por alternativas automatizadas con el objetivo de seguir desarrollando sus principales procesos a la altura que supone la dinámica de la toma de decisiones de la actualidad.

Con la presente investigación se pretende desarrollar una aplicación informática de beneficio para la Empresa Comercializadora y de Servicios de Productos Universales de Las Tunas que ofrezca a través de gráficos la información del comportamiento de los principales indicadores económicos registrado en el Sistema Económico Integrado VERSAT-Sarasola.

De la explotación de la futura herramienta se derivan ventajas importantes como son: mayor eficiencia en el análisis del cierre económico-financiero con reducción de posibles errores producto de la actividad humana, agiliza el tiempo de entrega y la manipulación de los datos obtenidos en la dirección contable financiera.

ÍNDICE

<b>INTRODUCCIÓN .....</b>	<b>1</b>
<b>CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA.....</b>	<b>6</b>
1.1 INTRODUCCIÓN.....	6
1.2 CONCEPTOS ASOCIADOS AL DOMINIO DEL PROBLEMA .....	6
1.2.1 <i>Contabilidad</i> .....	6
1.2.2 <i>Estados financieros</i> .....	6
1.2.3 <i>Indicadores</i> .....	7
1.2.3.1 <i>Resultados económicos</i> .....	7
1.2.3.2 <i>Resultados financieros</i> .....	8
1.3 OBJETO DE ESTUDIO.....	9
1.3.1 <i>Descripción general</i> .....	9
1.3.2 <i>Descripción actual del dominio del problema</i> .....	10
1.3.3 <i>Situación problemática</i> .....	11
1.4 ANÁLISIS DE OTRAS SOLUCIONES EXISTENTES.....	12
1.5 CONCLUSIONES.....	14
<b>CAPÍTULO 2 TENDENCIAS Y TECNOLOGÍAS ACTUALES A DESARROLLAR ..</b>	<b>16</b>
2.1 INTRODUCCIÓN.....	16
2.2 APLICACIÓN INFORMÁTICA .....	16
2.3 APLICACIÓN DE ESCRITORIO O DESKTOP.....	16
2.4 METODOLOGÍAS DE DESARROLLO DE SOFTWARE .....	17
2.4.1 <i>Proceso Unificado de Desarrollo de Software (RUP-Ágil)</i> .....	18
2.5 EL LENGUAJE UNIFICADO DE MODELADO (UML) COMO SOPORTE DE LENGUAJE ORIENTADO A OBJETO PARA EL MODELADO DE APLICACIONES DE ESCRITORIO.....	21
2.6 HERRAMIENTAS CASE DE DESARROLLO DE SOFTWARE .....	22
2.6.1 <i>Visual Paradigm</i> .....	23
2.7 CRYSTALREPORTS .....	24
2.8 ZEDGRAPH .....	26
2.9 ARQUITECTURA EN CAPAS .....	26
2.9.1 <i>Arquitectura de tres capas</i> .....	26
2.10 SISTEMA GESTOR DE BASE DE DATOS .....	27
2.10.1 <i>SQL Server 2000</i> .....	28
2.11 LENGUAJES DE PROGRAMACIÓN .....	30
2.11.1 <i>Lenguajes de programación C#</i> .....	31
2.12 IDE DE DESARROLLO .....	33
2.12.1 <i>Visual Studio.Net</i> .....	33
2.13 CONCLUSIONES.....	34
<b>CAPÍTULO 3 PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA.....</b>	<b>35</b>
3.1 INTRODUCCIÓN.....	35
3.2 MODELO DE NEGOCIO .....	35
3.2.1 <i>Actores y Trabajadores del Negocio</i> .....	35
3.2.2 <i>Procesos del negocio</i> .....	36
3.2.3 <i>Diagrama de Casos de Uso del Negocio</i> .....	36
3.2.4 <i>Descripción textual del caso de uso del negocio</i> .....	36
3.3 REQUISITOS FUNCIONALES .....	38
3.4 REQUISITOS NO FUNCIONALES .....	38

---

3.5 DESCRIPCIÓN DEL SISTEMA PROPUESTO .....	40
3.5.1 <i>Diagrama de Casos de Uso del Sistema</i> .....	40
3.5.2 <i>Descripción textual de los casos de uso del Sistema</i> .....	40
3.6 CONCLUSIONES.....	48
<b>CAPÍTULO 4 CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA .....</b>	<b>49</b>
4.1 INTRODUCCIÓN.....	49
4.2 PRINCIPIOS DE DISEÑO .....	49
4.2.1 <i>Estándares de la interfaz de la aplicación</i> .....	49
4.3 PATRONES DE DISEÑO .....	50
4.3.1 DIAGRAMA DE CLASES DEL DISEÑO .....	51
4.4 DISEÑO DE LA BASE DE DATOS .....	52
4.4.1 <i>Diagrama de clases persistentes</i> .....	53
4.4.2 <i>Diagrama Entidad – Relación</i> .....	53
4.5 GENERALIDADES DE LA IMPLEMENTACIÓN .....	54
4.5.1 <i>Modelo de componentes</i> .....	54
4.5.2 <i>Modelo de despliegue</i> .....	55
4.5.3 <i>Descripción de los nodos</i> .....	56
4.6 PRUEBA DEL SISTEMA PROPUESTO .....	56
4.6.1 <i>Pruebas de Caja Negra</i> .....	57
4.6.2 <i>Caso de uso: Configurar Sistema</i> .....	59
4.6.3 <i>Caso de uso: Gestionar planes</i> .....	61
4.7 CONCLUSIONES.....	66
<b>CONCLUSIONES .....</b>	<b>67</b>
<b>RECOMENDACIONES .....</b>	<b>68</b>
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>69</b>
<b>BIBLIOGRAFÍA .....</b>	<b>70</b>



## **ÍNDICE DE TABLAS**

TABLA 1 ACTORES Y TRABAJADORES DEL NEGOCIO.....	36
TABLA 2 DESCRIPCIÓN TEXTUAL. CU ELABORAR ANÁLISIS ECONÓMICO-FINANCIERO.....	38
TABLA 3 DESCRIPCIÓN TEXTUAL. CU GESTIONAR PLANES.....	44
TABLA 4 DESCRIPCIÓN TEXTUAL. GENERAR INFORME .....	46
TABLA 5 DESCRIPCIÓN TEXTUAL. CONFIGURAR SISTEMA.....	47
TABLA 6 SECCIONES A PROBAR EN EL CASO DE USO CONFIGURAR SISTEMA .....	60
TABLA 7 DESCRIPCIÓN DE LAS VARIABLES PARA EL CASO DE USO CONFIGURAR SISTEMA ....	60
TABLA 8 MATRIZ DE DATOS. SC1. CONFIGURAR SISTEMA .....	61
TABLA 9 SECCIONES A PROBAR EN EL CASO DE USO GESTIONAR PLANES .....	64
TABLA 10 DESCRIPCIÓN DE LAS VARIABLES PARA EL CASO DE USO GESTIONAR PLANES .....	64
TABLA 11 MATRIZ DE DATOS. SC1. “ADICIONAR PLAN” .....	65
TABLA 12 MATRIZ DE DATOS. SC2. “EDITAR PLAN”.....	66

## **ÍNDICE DE FIGURAS**

FIGURA 1 CICLO DE VIDA DE RUP-ÁGIL .....	18
FIGURA 2 DIAGRAMA DE CASOS DE USO DEL NEGOCIO .....	36
FIGURA 3 DIAGRAMA DE CASOS DE USO DEL SISTEMA .....	40
FIGURA 4 DIAGRAMA DE CLASES DEL DISEÑO. PAQUETE DE DISEÑO.....	52
FIGURA 5 DIAGRAMA DE CLASES DEL DISEÑO CU GENERAR INFORME .....	52
FIGURA 6 DIAGRAMA DE CLASES PERSISTENTES .....	53
FIGURA 7 DIAGRAMA ENTIDAD-RELACIÓN.....	54
FIGURA 8 DIAGRAMA DE COMPONENTES.....	55
FIGURA 9 DIAGRAMA DE COMPONENTES COMPILADOS DEL SISTEMA .....	55

# **INTRODUCCIÓN**

El mundo empresarial es cada vez más complejo y dinámico, las empresas tienen que crear herramientas que le permitan adaptarse a las condiciones cambiantes en su entorno. Una de las tendencias de gestión más modernas es la gestión estratégica empresarial en su lógica de planear, implementar, controlar y mejorar los procesos que se desarrollan dentro de la empresa al hacer uso de las Tecnologías de la Información y las Comunicaciones (TIC).

Estas han impuesto un nuevo ritmo a los actuales procesos de generación, manipulación y transferencia de la información, de manera que organizaciones gubernamentales y no gubernamentales, empresas públicas y privadas, han buscado la manera de insertarse en algo que ha tendido a llamarse “informatización de la sociedad” con el objetivo de garantizar la respuesta efectiva a sus clientes y/o necesidades en el tiempo y la forma que supone este nuevo orden internacional de la información. Esta situación real presenta nuevos desafíos y retos para la administración de las organizaciones, ante la cual los directivos han experimentado continuamente con diversas respuestas y enfoques.

La Empresa Comercializadora y de Servicios de Productos Universales de Las Tunas, ubicada en Camino del Oriente s/n, zona industrial, municipio Las Tunas, fue constituida por Resolución 105/77 dictada por el entonces Comité Estatal de Abastecimiento Técnico Material (CEATM) en la fecha del 2 de junio de 1977, con subordinación al organismo nacional del Ministerio del Comercio Interior.

Su objeto social, según la Resolución 263 de fecha 5 de junio del año 2007 del Ministerio de Economía y Planificación consiste en:

1. Comercializar de forma mayorista bienes de consumo e intermedios, en pesos cubanos, según nomenclatura aprobada.
2. Prestar servicios de almacenamiento y conservación de las mercancías de la reserva especial y estatal, en pesos cubanos.
3. Brindar servicios de alquiler de almacenes y cámaras frías, en pesos cubanos.
4. Prestar servicios de transportación de cargas cumpliendo con las regulaciones establecidas por el Ministerio del Transporte, en pesos cubanos.

En el marco de la dirección estratégica la empresa ha formulado su misión bajo el concepto de comercializar de forma mayorista productos en las ramas de la química, ferretería, alimentos, metales, maderas, textil y calzado, artículos de uso personal, equipos, materiales de construcción y programa energético, todos estos en moneda nacional, con calidad, eficiencia; su objetivo es satisfacer las expectativas y exigencias de los clientes. Almacenar y conservar productos de las reservas estatales, prestar servicio de arrendamiento de espacio y almacén, comercialización de productos de los planes especiales (uniformes escolares, canastillas). Es el único canal de distribución mayorista de productos universales en moneda nacional (CUP) para el sector del comercio en el territorio nacional.

Para garantizar el cumplimiento de la misión planteada la empresa se encuentra estructurada en cuatro direcciones de regulación y control, donde las tres últimas se subordinan a la dirección general:

- Dirección general.
- Dirección adjunta.
- Dirección de capital humano.
- Dirección contable financiera.

La dirección contable financiera de la Empresa Comercializadora y de Servicios de Productos Universales de Las Tunas, se encuentra enclavada en la misma entidad, subdividiéndose a su vez en tres grupos para el registro ordenado de los hechos económicos que responde a una estructura descentralizada. Los grupos o departamentos se encuentran ubicados a razón de uno por cada Unidad Empresarial de Base y se caracterizan de esta manera:

- Contabilidad Dirección Empresa.
- Contabilidad UEB Comercializadora.
- Contabilidad UEB de Aseguramiento y Servicios.

Dentro de los principales procesos que allí se realizan se encuentran:

- Reconocer o registrar los hechos económicos y financieros que se suceden en la empresa, en su contabilidad a partir del documento primario donde estos se plasman.

- Procesar, controlar y analizar el hecho económico y financiero registrado en la contabilidad.

Dentro de la mencionada dirección se han realizado numerosos intentos para la informatización de sus procesos internos, es por ello que actualmente se explota el Sistema Económico Integrado VERSAT-Sarasola Versión 2.0.0 Actualización 6.7 (ultima en explotación) para el registro contable financiero de las operaciones que la empresa desarrolla.

El sistema tiene un grupo de características que lo hacen útil para el usuario al permitirle garantizar la rapidez y la fiabilidad en el proceso de la gestión de la información, a partir de configuraciones que se pueden hacer del proceso de contabilización de los documentos primarios y de las propias posibilidades de trabajo creadas en cada módulo o subsistema que logran una interacción usuario-sistema fácil y segura.

No obstante, en la práctica los usuarios del sistema no logran obtener un análisis más completo e integral sobre los datos procesados por causas puntuales que se pueden señalar, entre ellas: la falta de conocimientos, habilidades e interés por alcanzar estadios superiores que proporcionen al directivo, fundamentalmente, los elementos claves, precisos y oportunos para lograr una mejor toma de decisiones en el cumplimiento de su trabajo. Por otra parte, este sistema sólo posibilita la inserción de datos porque es una base de datos estática donde se almacenan los diferentes indicadores para el análisis económico-financiero, lo que no permite la realización de análisis gráficos y en tiempo real de los principales elementos asociados al balance económico-financiero, convirtiéndose la accesibilidad de los datos en un proceso muy complejo para el usuario, lo cual imposibilita la realización de diferentes informes de forma automática ;hasta el momento sólo pueden hacerlos manualmente a partir de la adquisición de los datos en la base de datos.

Por todo lo anteriormente expresado queda evidenciado que el **problema a resolver** es ¿cómo gestionar de forma asequible, gráfica y en tiempo real los resultados económicos-financieros en la Empresa Comercializadora y de Servicios de Productos Universales de Las Tunas?

Asimismo el **objeto de estudio** lo constituyen los procesos económicos-financieros de la Empresa Comercializadora y de Servicios de Productos Universales de Las Tunas y el **campo de acción** la Automatización de los procesos económicos-financieros en la Empresa Comercializadora y de Servicios de Productos Universales de Las Tunas.

Se persigue con esta investigación lograr el siguiente **objetivo general**: Desarrollar una herramienta para automatizar los procesos económicos-financieros en la Empresa Comercializadora y de Servicios de Productos Universales de Las Tunas.

Para darle cumplimiento al objetivo general, se proponen las siguientes **tareas investigativas**:

1. Identificar las herramientas y tendencias en el desarrollo de aplicaciones para mostrar los resultados económicos financieros en empresas.
2. Determinar las tecnologías sobre las que se desarrollará el sistema para el análisis económico-financiero.
3. Identificar requisitos funcionales y no funcionales para desarrollar la herramienta.
4. Diseñar la herramienta para el análisis económico-financiero.
5. Implementar la herramienta para el análisis económico-financiero.
6. Documentar la investigación con vistas a su socialización.


Se plantea la siguiente **idea a defender**:

Con el desarrollo de una herramienta fácil de utilizar basada en nuevas tecnologías, se contribuirá a perfeccionar los procesos de gestión económicos-financieros en la Empresa Comercializadora y de Servicios de Productos Universales de Las Tunas.

La **población** está constituida por la Empresa Comercializadora y de Servicios de Productos Universales de Las Tunas y la **muestra** la dirección contable financiera.

Como **métodos de investigación científica** se utilizaron:

### **Métodos Teóricos:**

 Histórico-lógico: posibilita el análisis histórico de los procesos que se realizan en la empresa, para ello se realiza un estudio detallado de los documentos que rigen su funcionamiento.

✚ Análisis-síntesis: se analiza la bibliografía, los mecanismos existentes en la empresa para realizar el análisis económico-financiero y se desarrolla una síntesis de los mismos.

✚ Modelación: se modelan un conjunto de diagramas, basados en la metodología utilizada, los que facilitan la implementación de la herramienta, debido a que organizan las estructuras lógicas del sistema mostrando las principales clases y relaciones a implementar.

## **Métodos Empíricos:**

✚ Entrevista: Se realiza al director de la dirección contable financiera con el fin de precisar la idea a defender, así como para la validación de la propuesta que se presenta.

✚ Observación: se realizaron visitas a la Empresa Comercializadora y de Servicios de Productos Universales de Las Tunas, específicamente a la dirección contable financiera para observar los procesos manuales y con la base de datos VERSAT Sarasola que se llevan a cabo en la actualidad.

## **Estructuración del contenido**

La investigación consta de 4 capítulos, en el **Capítulo 1** se aborda en detalle todo lo relacionado con la fundamentación teórica de la investigación y se especifican algunos conceptos asociados en la misma. Se trata de forma detallada el objeto de estudio y la situación problemática para mejor entendimiento del propósito de la investigación.

En el **Capítulo 2** se hace referencia a las Tecnologías de la Información y las Comunicaciones (TIC), además de tratar las tendencias y tecnologías actuales para el desarrollo de herramientas desktop, lo que propicia fundamentar la elección del Lenguaje de programación que se escoja así como de la Metodología de Desarrollo, el Lenguaje de Modelado y la Herramienta CASE.

El **Capítulo 3** comprende las características del sistema, realizándose la presentación de la solución propuesta.

En el **Capítulo 4** es donde se efectúa la construcción de dicha solución, la cual incluye el diagrama de clases, los diagramas de diseño, al igual que el diseño de la base de datos, la implementación y la prueba del sistema propuesto.

## **CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA**

### **1.1 Introducción**

En el presente capítulo se abordan los temas referidos al aseguramiento teórico de la investigación, se analizan y clasifican los principales conceptos asociados al dominio o marco del problema que permitan su mejor entendimiento, se describe la situación actual de la organización, los principales procesos que allí se llevan a cabo y luego se profundiza en la situación problemática. Finalmente se analizan otras soluciones existentes que pudieran contribuir con una posible variante de solución del problema.

### **1.2 Conceptos asociados al dominio del problema**

#### **1.2.1 Contabilidad**

La Contabilidad es la Ciencia que proporciona información de hechos económicos, financieros y sociales suscitados en una empresa; con el apoyo de técnicas para registrar, clasificar y resumir de manera significativa y en términos de dinero, “transacciones y eventos”, de forma continua, ordenada y sistemática, de tal manera que se obtenga información oportuna y veraz sobre la marcha o desenvolvimiento de la empresa u organización con relación a sus metas y objetivos trazados, con el objeto de conocer el movimiento de las riquezas y sus resultados. (Baldiviezo, 2008)

#### **1.2.2 Estados financieros**

Son aquellos documentos que muestran la situación económica de una empresa y su capacidad de pago en una fecha determinada, o bien el resultado de operaciones obtenidas en un período o ejercicios pasados en situaciones normales o especiales.

Su objetivo es suministrar información acerca de la situación financiera, desempeño y cambios en la posición financiera de una empresa. La utilidad se manifiesta en la información que brinda a disímiles de usuarios. Sus principales características radican en su comprensibilidad, relevancia, fiabilidad y comparación.

Los estados financieros se confeccionan y presentan mensualmente cubriendo las necesidades comunes de información de los usuarios. Forman parte del proceso de información financiera de la empresa.



Tipos de Estados financieros:

1. Estado de situación.
2. Estado de resultado.
3. Estado de cambio en la posición financiera.
4. Estado de flujo de efectivo.
5. Estados complementarios.

### **1.2.3 Indicadores**

Herramientas para clasificar y definir de forma más precisa, objetivos e impactos. Medidas verificables de cambio o resultado, diseñadas para contar con un estándar con el cual evalúa, estima o demuestra el progreso con respecto a metas establecidas, estas facilitan el reparto de insumos, produce productos y alcanza objetivos. (ONU)

#### **1.2.3.1 Resultados económicos**

Los resultados económicos de una empresa tienen que ver con el cumplimiento de sus planes de circulación mercantil, servicios y producción a lo que se destine el objeto social de la misma a partir de la eficiencia y eficacia con que los alcanza, trata siempre de cubrir sus gastos con sus ingresos para obtener un mínimo de utilidades.

#### **Cumplimiento del Plan de ventas**

Expresa como se comporta la circulación mercantil real (ventas mayoristas) alcanzadas en el período analizado al compararse con el plan de la economía previsto para igual fecha. En la empresa objeto de este trabajo, las ventas mayoristas son las ventas de mercancías que se realizan a los clientes, constituidos en su mayoría por las empresas minoristas del sector del comercio en el territorio nacional. Los planes de ventas se aprueban por el Ministerio de Economía y Planificación a propuesta de la empresa para el período de un año.

#### **Cumplimiento del plan de ventas por estructura**

Expresa como se comporta la circulación mercantil real (ventas mayoristas) alcanzada en el período analizado desglosada por la estructura planificada, es decir, por conceptos o surtidos de ventas previamente definidos y planificados y su comparación con el plan,

lo que permite conocer el peso específico que tiene cada surtido en el cumplimiento o no del plan total de la economía.

## **Comportamiento de los inventarios de mercancías para la venta**

Las mercancías para la venta son el principal inventario de bienes con que cuenta la empresa para desarrollar su actividad de comercio mayorista como único canal de distribución de productos universales en pesos CUP (moneda nacional) que tiene el país. Expresa como están compuestos los inventarios de la empresa en una fecha determinada, refleja el peso específico por sus valores que tienen por surtidos, dentro del total de inventarios de mercancías para la venta en la fecha tomada para el análisis.

## **Comportamiento de las utilidades antes de impuestos**

Las utilidades de la empresa surgen de restar a los ingresos del período sus gastos, si el resultado es positivo tendrá utilidades, si es negativo entonces habrá pérdidas. Las utilidades en el sector empresarial cubano gravan un impuesto sobre utilidades (35%) por eso se dice “utilidades antes de impuestos”, es decir, las utilidades alcanzadas antes de aplicar el gravamen del 35%. Su resultado real en la fecha que se analice, siempre dentro de un año fiscal, se compara con lo planificado para acreditar el grado de cumplimiento en la fecha prevista de este indicador.

### **1.2.3.2 Resultados financieros**

Los resultados financieros tienen que ver con la posibilidad de la empresa de cubrir sus obligaciones de pago a corto, mediano y largo plazo, expresan la situación de la empresa desde el punto de vista de su dinero o posibilidades financieras.

### **Liquidez general**

Mide la solvencia (capacidad de pagar) a corto plazo de la empresa e indica el grado en que los derechos de sus acreedores a corto plazo se encuentran cubiertos por activos (bienes) que se esperan se conviertan en efectivo en un período más o menos igual que el vencimiento de sus obligaciones. Es el resultado de dividir el total del activo circulante entre el total del pasivo circulante, ambos indicadores se muestran en el estado de situación de la empresa y su resultado se expresa en pesos.

## **Liquidez inmediata**

Este índice proporciona una imagen más precisa de la situación de la empresa para hacer frente a sus obligaciones a corto plazo e indica la habilidad de la misma para liquidar sus obligaciones más perentorias. Es el resultado de dividir el total del activo circulante excluyendo los inventarios (por ser los menos líquidos) entre el total del pasivo circulante.

### **1.3 Objeto de estudio**

#### **1.3.1 Descripción general**

Actualmente todas las empresas en el mundo realizan los procesos económicos-financieros de sus entidades guiadas por sus objetivos y con las especificaciones de su entorno laboral.

Sin duda, el incremento de las aportaciones externas de recursos constituye una oportunidad para el cumplimiento de la misión de cada empresa, siempre que las organizaciones establezcan relaciones coherentes con sus valores y que éstas no impliquen la adopción de comportamientos que comprometan su sostenibilidad a largo plazo. Pero supone también afrontar una serie de problemas desde el punto de vista de la gestión económico-financiera (incertidumbre, inestabilidad, riesgo de dependencia de un único financiador, tensiones de tesorería), que es necesario gestionar adecuadamente.

Por ello, es necesario que las organizaciones refuercen sus dinámicas de planificación, organización y control económico-financiero, que desarrollen formas de pensar, de gestionar sus recursos orientados a garantizar su sostenibilidad económica, social y medioambiental como una condición necesaria para cumplir con su misión. Esto se convierte en paso de avance para la Empresa Comercializadora y de Servicios de Productos Universales de Las Tunas, donde el proceso de gestión de información contable-financiera consistirá en una herramienta conectada directamente a la ya existente base de datos VERSAT Sarasola. Luego esta BD permitirá el cumplimiento de las funciones de la dirección contable financiera, dirigido a la fiscalización, control y planificación de los indicadores contables-financiero. Este proceso aporta además los

elementos necesarios para una correcta toma de decisiones a partir de los informes emitidos por la herramienta.

### **1.3.2 Descripción actual del dominio del problema**

La Empresa Comercializadora y de Servicios de Productos Universales de Las Tunas, específicamente la dirección contable financiera realiza numerosa operaciones contables distribuidas de la forma siguiente:

Contabilidad–Dirección-Empresa, registra lo inherente a las actividades del ciclo de tesorería, nóminas, cobros y pagos, activos fijos así como las transacciones del patrimonio de la entidad y las inversiones o reparaciones ejecutadas. Controla además lo relacionado con el movimiento del fondo fijo para pagos menores aprobados y su custodia, posee la responsabilidad de realizar la planificación de los indicadores económicos financieros y su estadística a nivel de empresa.

Contabilidad UEB<sup>1</sup> Comercializadora, es una unidad que tiene como actividad fundamental la comercialización de forma mayorista de bienes de consumo e intermedios en moneda nacional, presta servicios especializados referente al alquiler de almacenes en el marco de las aprobaciones existentes, también comercializa de forma mayorista a distribución del alcohol, el programa de sarcófagos y otros programas aprobados por la dirección central del Ministerio de Comercio Interior (MINCIN). Contablemente registra operaciones del ciclo de nóminas, cobros y pagos, ingresos y activos fijos así como lo relacionado con el ciclo de inventarios.

Contabilidad UEB de Aseguramiento y Servicios, es una unidad dedica a las actividades de servicio y apoyo de la gestión empresarial; atiende las operaciones relacionadas con los insumos de materiales auxiliares, víveres u otros necesarios para asegurar el funcionamiento del comedor obrero y cafetería enclavados en la entidad. De igual forma asegura el funcionamiento del plan vacacional de los trabajadores (alojamiento y transporte) y la actividad de autoconsumo. Realiza operaciones de los ciclos de inventarios, ingresos no fundamentales asociados a la atención al hombre, nóminas, pagos, activos fijos y el ciclo de gastos.

---

<sup>1</sup> Unidad Empresarial de Base.

Estas áreas de trabajo proporcionan un conjunto de información económica-contable que hacen posible una mayor planeación, ejecución y control de todas las actividades que la empresa desarrolla, que puedan favorecerla o no. Todos estos procesos son monitoreados por el director de la dirección contable financiera, responsable de manejar y evaluar dicha información, de igual modo todos los informes generados de forma independiente por las áreas de trabajo son aprobados y revisados por el director para luego ser entregados y discutidos ante el consejo de dirección de la empresa.

### **1.3.3 Situación problemática**

En la Empresa Comercializadora y de Servicios de Productos Universales de Las Tunas hasta el momento no existe ningún sistema automatizado que gestione los procesos contables-financieros o estado mensual y anual de los mismos. El control sobre los indicadores contables-financieros se obtiene de forma manual. Estos indicadores ya están registrados en la BD VERSAT Sarasola pero por su importancia se hace necesario comenzar a realizarlo de forma diferente a través de una herramienta que gestione y sirva de interfaz para el usuario y los datos ya registrados.

La dirección contable financiera encargada de planificar y mantener el control sobre los indicadores, cada mes al finalizar este proceso, debe entregar al director de la empresa un informe con los diferentes indicadores donde se hace un análisis a profundidad de las variaciones ocurridas en estos indicadores durante el período evaluado, lo que posibilita la toma de decisiones para próximos períodos. Estos informes están soportados en diferentes formatos como word o excel, lo que hace más dificultoso el acceso a la información.

Por el interés que existe en la empresa en cuanto a la gestión de los datos sobre dichos indicadores se acordó una nueva estrategia para llevar a cabo este proceso. Por lo que no sólo se trabajaría en realizar de forma automática un balance de los mismos, sino todo lo relacionado con el control y análisis de las posibles fluctuaciones o variaciones que ocurren que pudieran afectar las ventas y los resultados finales de la empresa.

Con el desarrollo de esta nueva estrategia, se crea un sistema que facilita la entrega de estos indicadores en el tiempo acordado, ganando en oportunidad, calidad y confiabilidad de la información procesada por el sistema VERSAT Sarasola.

## 1.4 Análisis de otras soluciones existentes

En el mundo existen varias empresas que se dedican a la producción de software para la automatización de la planificación, el control y análisis de la gestión económica.

Después de un análisis de como se lleva este proceso en otros países como España, Argentina y Brasil. Se toman algunos elementos que se pueden adaptar y hacerlos parte del proceso objeto de esta investigación. Hay que tener en cuenta que las empresas relacionadas en estos procesos son privadas, por lo que el control de estos datos no se puede llevar de igual forma porque las características son diferentes y el entorno empresarial es bajo producto al sistema.

Entre las soluciones existentes encontradas se puede mencionar **ERP<sup>2</sup> New Pyme**, el cual es un sistema de planificación de recursos empresariales de última generación basados en los actuales estándares en cuanto a ingeniería del software se refiere, enfocado a las pequeñas y medianas empresas para cubrir sus necesidades administrativas, operativas y productivas. Esta herramienta que integra en forma rápida y unificada todas las áreas de la empresa que abarca: contabilidad general, cartera, proveedores, tesorería, inventarios, facturación, puntos de venta POS, gráficos dinámicos y medios magnéticos.

El usuario puede crear rápidamente cualquier gráfico bajo la Tecnología OLAP<sup>3</sup> visualiza información y la composición del activo, el pasivo o el patrimonio. Un reporte OLAP para que sea mejor entendido, es comparable a una tabla dinámica en Excel, donde usted puede agregar o quitar campos, comparar períodos y rotar la presentación de datos. Por ejemplo, usted podrá graficar los gastos de su empresa contra la utilidad de los cuatro años anteriores o de los meses corridos del presente año y podrá pasar de un gráfico de ventas por producto a ventas por vendedor con solo rotar el cubo de información.

Esta solución dada a Cuba no satisface totalmente las necesidades de la empresa porque no es gratis, cuenta con tres versiones: industrial, empresarial y profesional, con un costo entre los 100 y 275 usd. Es un software europeo y su licencia debe ser

---

<sup>2</sup> Por sus siglas en inglés: Enterprise Resource Planning, lo cual quiere decir en español planificación de recursos empresariales.

<sup>3</sup> Acrónimo en inglés de procesamiento analítico en línea (On-Line Analytical Processing).

comprada; además de esto en Cuba el cumplimiento del plan de producción, de la circulación mercantil o de servicios con el máximo de economía y eficiencia constituye el objetivo central de toda organización económica. Este sistema depende en gran medida de la gestión administrativa para corregir los puntos débiles y sacar partida de los puntos fuertes en el proceso de dirección de las operaciones. Este proceso no es visualizado en la herramienta anteriormente mencionada, es por ello que su uso en la entidad se limita a simples reportes gráficos.

Entre los años 80 y 90 en Cuba se dan los primeros pasos en la automatización de la planificación, el control y el análisis de la gestión económica en empresas del Ministerio de la Industria Azucarera (MINAZ), donde se crea el sistema de gestión VERSAT Sarasola. Comienza a engendrarse así un paquete de tránsito que se consolida a partir de 1988 con la interrelación de los primeros subsistemas.

Este sistema es un producto cubano, adecuado a los principios del control interno y regulaciones vigentes en el país, que garantiza su continuo desarrollo ante nuevos mecanismos de organismos rectores, para ponerlo en manos del sistema empresarial cubano. Además constituye una valiosa herramienta que permite a los directivos analizar, controlar y evaluar los resultados de su negocio, en tiempo real mediante un software integrado, orientado a la gestión empresarial y presupuestaria, soportado sobre una plataforma de trabajo "Cliente-Servidor".

Entre las principales características del VERSAT Sarasola se encuentran:

- Herramienta para la planificación económica, el control y el análisis de gestión.
- Se emplea en cualquier tipo de entidad empresarial o presupuestada.
- Permite llevar el control y registro contable individual de todos los hechos económicos que se originan en las estructuras internas de las entidades, así como exponer el estado financiero y toda la información económica y contable en este universo.
- Se estructura en un grupo de subsistemas en los cuales se procesan y contabilizan los documentos primarios, donde se anotan los movimientos, los recursos materiales, laborales y financieros que se utilizan en una entidad.
- Se logra establecer un proceso de interacción usuario-sistema.

- Rapidez y fiabilidad a partir de la configuración del proceso de contabilización de los documentos primarios y de las propias posibilidades de trabajo contenidas en cada subsistema.

Aunque el sistema presenta esas características y realiza esa serie de funcionalidades, no cubre hoy en día muchas expectativas. No permite ofrecer a través de gráficos que interactuando unos con otros lleven en un lenguaje ameno, claro y sencillo, los principales resultados de una organización a sus directivos y trabajadores. No muestra en tiempo real un conjunto de indicadores que han sido afectados en un determinado estado financiero. Impide realizar de forma automática informes donde se registren las comparaciones entre los diferentes estados financieros de un año respecto a otro y registrar a las personas autorizadas para elaborar, eliminar o actualizar un estado financiero, lo que garantiza la fiabilidad de los datos e integridad de los mismos.

**BI Versat** es una solución de Inteligencia de Negocios que muestra en forma de Dashboards <sup>4</sup> las métricas de los principales procesos de la empresa. En él se aborda en forma de gráficos los indicadores claves de desempeño del proceso de negocio, su estado actual, el desarrollo histórico de este, así como comparaciones en distintos períodos de tiempo. Esta manera de visualizar la información ahorra mucho tiempo al directivo y permite que no se pierda entre tablas y números.

El sistema fue desarrollado por los programadores de la casa del software Habana (HavaSoft). A pesar de ser un sistema desarrollado en Cuba, no cuenta con un módulo que le permita al usuario realizar un informe general con todos los datos necesarios (gráficas, tablas, textos) sobre los principales indicadores para desarrollar un análisis económico-financiero, siendo un proceso necesario para la empresa en cuestión, es por ello que el sistema no satisface las necesidades de la empresa.

## 1.5 Conclusiones

En el presente capítulo se exponen todos los contenidos referidos al aseguramiento teórico de la investigación, se explican todas las clasificaciones de los principales indicadores que se registran en un estado financiero en la Empresa Comercializadora y de Servicios de Productos Universales de Las Tunas.

---

<sup>4</sup> Dashboards: Tableros o cuadros de mando.



Se detalla minuciosamente el objeto de estudio en términos de procesos, situación problemática y soluciones existentes, lo que permite la conclusión de que las aplicaciones actuales no brindan las posibilidades necesarias para llevar a cabo con excelencia los procesos de análisis concernientes a la gestión, evaluación y la seguridad de los datos e indicadores para el estado financiero en la Empresa Comercializadora y de Servicios de Productos Universales de Las Tunas.

## **CAPÍTULO 2 TENDENCIAS Y TECNOLOGÍAS ACTUALES A DESARROLLAR**

### **2.1 Introducción**

En el presente capítulo se describen todas las tecnologías que se utilizan en la construcción de la aplicación, se analizan las principales tendencias actuales en cuanto a tipos de aplicaciones, Metodologías de Desarrollo de Software, Herramientas CASE y de Programación, y Sistemas Gestores de Bases de Datos. El capítulo incluye la justificación de todos los usos de tecnologías o tendencias a utilizar, lo que evidencia claramente las principales características que hicieron posible su selección.

### **2.2 Aplicación Informática**

En un ordenador, computadora o sistema de cómputo, el software representa las instrucciones responsables para que el hardware (la máquina) realice su tarea. Como concepto general, el software puede dividirse en varias categorías basadas en el tipo de trabajo realizado: por una parte, el sistema operativo que controla los trabajos del sistema, y por otra, el software de aplicación que dirige las distintas tareas que se requiere.

Un programa informático, sinónimo de software, es el conjunto de instrucciones que ejecuta un ordenador o computadora, donde una vez ejecutadas realizarán una o varias tareas; también suele referirse al código fuente original o a la versión ejecutable. (STAIR, 2003)

Una aplicación informática se puede definir como un programa informático diseñado para facilitar al usuario la realización de un determinado tipo de trabajo. Generalmente resultan ser una solución informática para la automatización de ciertas tareas complicadas como pueden ser la contabilidad, la redacción de documentos, la gestión de un almacén y la comunicación de datos.

### **2.3 Aplicación de escritorio o Desktop**

Es la aplicación creada para ejecutarse en un ordenador de escritorio, sobre un sistema operativo de interfaz visual como Windows o Linux.

Cuando se refiere al término aplicación de escritorio o desktop se está en presencia de una aplicación informática que se ejecuta en una computadora de escritorio o en una portátil (laptop) de manera local, en contraste con las aplicaciones web.

Este tipo de aplicación informática se ejecuta en el ordenador en su propia dirección de memoria física o virtual y no requiere cargarse en otras aplicaciones como navegadores para poder funcionar. No obstante, una aplicación de escritorio puede encontrarse distribuida en varios ordenadores y comunicarse a través de diferentes mecanismos de la red, tales como TCP/IP, UDP, RPC, entre otros; además de poder hacer uso de servidores de almacenamiento de datos. (Desarrollo Web)

### **2.4 Metodologías de Desarrollo de Software**

La arquitectura de una aplicación es la vista conceptual de la estructura de ella misma. Toda aplicación contiene código de presentación, código de procesamiento de datos y código de almacenamiento de datos. La arquitectura de las aplicaciones difiere según como está distribuido este código. (CORNEJO, 2001)

Uno de los principales problemas en la actualidad en el desarrollo de software es seleccionar la metodología adecuada que posibilite obtener los resultados óptimos que se desean; o sea, cómo trabajar eficientemente y evitar el fracaso de proyectos a nivel mundial. Una metodología tiene como principal objetivo aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo.

El desarrollo de software no es sin dudas una tarea fácil. Como resultado a este problema ha surgido una alternativa: la Metodología. Las metodologías imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Se realiza a partir de desarrollar un proceso detallado con un fuerte énfasis en planificar inspirado por otras disciplinas de la ingeniería.

Actualmente se cuenta con una buena cantidad de propuestas que definen el marco de trabajo, se escucha hablar de metodologías tradicionales o robustas como Rational Unified Process (RUP), Microsoft Solutions Framework (MSF) y Métrica 3.0 o

metodologías ágiles como Extreme Programming, Scrum, Crystal Methods y Feature Driven Development, pero no se trata solo de seleccionar una de las propuestas y seguirla simplemente, es seleccionar la correcta para solucionar el problema. (FRANK BUSCHMANN, 2000)

### 2.4.1 Proceso Unificado de Desarrollo de Software (RUP-Ágil)

El RUP-Ágil es una versión simplificada de Rational Unified Process (RUP). Este describe un enfoque simple y fácil de entender para el desarrollo del software, usa técnicas y conceptos que aún se mantienen vigentes en RUP. Los enfoques aplican técnicas ágiles incluidas en el Desarrollo Dirigido por Pruebas (TDD), Desarrollo Dirigido por Modelado Ágil (AMDD), administración de cambios ágil, y refactorización de bases de datos para mejorar la productividad.

La Figura 1 representa el ciclo de vida de RUP-Ágil. Lo primero es que las disciplinas han cambiado. Primero, la disciplina de Modelado abarca las disciplinas de Modelado del Negocio, requisitos, Análisis y Diseño de RUP. El modelado es una parte importante en RUP-Ágil, pero no domina el proceso, puede seguir ágil creando modelos y documentos los cuales son apenas lo suficientemente bueno. Segundo, las disciplinas de la Administración de la Configuración y Cambios ahora es la disciplina de la Administración de la Configuración.

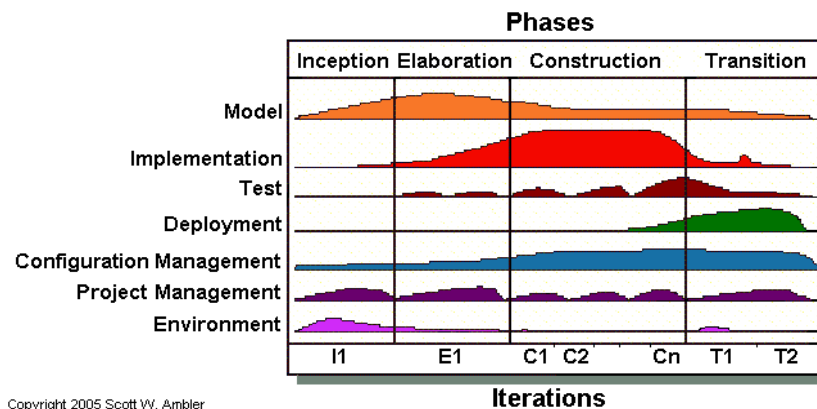


Figura 1 Ciclo de vida de RUP-Ágil.

La naturaleza serial en RUP-Ágil es capturada en cuatro fases:

- Iniciación: identifica el alcance inicial del proyecto, una arquitectura potencial de su sistema para obtener la financiación inicial del proyecto y la aceptación del involucrado.
- Elaboración: mejora la arquitectura del sistema.
- Construcción: construye el software funcional en una base regular e incremental, la cual cumpla con las necesidades de prioridad más alta de los involucrados de su proyecto.
- Transición: valida y despliega su sistema en su ambiente de producción.

Las disciplinas son ejecutadas en una manera interactiva, define las actividades y los miembros del equipo ejecutan para construir, validar y liberar el software funcional que cumpla con las necesidades de sus involucrados. Las disciplinas son:

- Modelado: entiende el negocio de la organización, el problema de dominio que se aborda en el proyecto, e identificar las soluciones viables para manejar el dominio del problema.
- Implementación: transforma su modelo (s) en código ejecutable y lleva a cabo un nivel básico de las pruebas, en particular, la unidad de prueba.
- Pruebas: ejecuta una objetiva evaluación para asegurar la calidad. Esto incluye la detección de defectos, validaciones para que el sistema funciona como fue diseñado y verifica que se cumplan los requisitos.
- Despliegue: planifica la entrega del proyecto de desarrollo y ejecuta el plan para dejar disponible el sistema al usuario final.
- Administración de la Configuración: maneja el acceso a sus productos de trabajo de proyecto. Esta no sólo incluye el rastreo de versiones del trabajo del producto en el tiempo, sino que también el control y administración de los cambio estos productos.
- Administración del Proyecto: dirige las actividades a lo largo del proyecto. Esto incluye la administración del riesgo, dirección del personal (asignación de tareas y rastreo del progreso) y coordinación con personas y sistemas fuera del alcance del proyecto para asegurar su liberación a tiempo y dentro del presupuesto.
- Entorno: soporta el resto del esfuerzo lo que asegura que el proceso apropiado, las guías (normas y directrices), y herramientas (hardware y software) estén disponibles para cuando el equipo las necesite. (Rup-Ágil)

Los equipos de RUP-Ágil típicamente liberan versiones desarrolladas al final de cada iteración en un área (s) de prueba de pre-producción y/o demostraciones. Una versión de desarrollo de una aplicación es algo que podría ser liberado en la producción si se ponen a través de su pre-producción de aseguramiento de la calidad (QA), pruebas y proceso de despliegue. La primera versión de producción comúnmente toma más tiempo de entrega que otras versiones subsecuentes; en la primera liberación de un sistema probablemente necesita colocar mucha “cañería” en su lugar y su equipo probablemente no cuente con lo suficiente para convertirlo en algo colaborativamente eficiente. La primera versión de producción puede tardar 12 meses en liberación, la segunda nueve meses para liberación y las demás liberaciones son entregadas cada 6 meses.

Después de realizar un estudio de las metodologías más usadas actualmente en el mundo, se llega a la conclusión, que la metodología a utilizar para el desarrollo de la aplicación de escritorio propuesta es, RUP-Ágil, la cual es efectiva para modelar y documentar un proyecto de software. Es una colección de principios y prácticas que pueden ser aplicados de manera simple y ligera. El uso de esta metodología facilitó el desarrollo del sistema ya que su función principal se centra en las funcionalidades del mismo dando prioridad en cada iteración a los entregables del cliente.

La metodología RUP-Ágil tiene varios principios que la diferencian sobre las metodologías tradicionales reflejados en:

- Los individuos y sus relaciones sobre las personas y los procesos: - este principio se pone de manifiesto en el énfasis de esta metodología sobre las personas, porque de ellas son de las que depende el éxito o el fracaso de un proyecto, es a las que se les debe motivar.
- Un software funcional, que trabaje sobre la documentación más completa: - lo más importante es que el software trabaje, cumpla con las necesidades de negocio, no hacer de la documentación un fin en si mismo, porque este es solo para dar soporte , no es el objetivo primario del desarrollo, existen situaciones donde incluso la documentación podría ser innecesaria, por ejemplo, una pequeña aplicación emergente, que una vez pasada la emergencia , esta aplicación desaparece, el cargar de documentación de requisitos, arquitectura y

- testeo podría considerarse de sobra, sin embargo, eso no quiere decir que no es necesaria la documentación, esta debe existir pero solo la suficiente.
- Colaboración el cliente sobre el contrato de negocio.- se trata de colaborar con el cliente el mayor tiempo, no de luchar con él sobre un contrato minucioso, esto puede ser difícil porque los clientes no están acostumbrados, ellos están acostumbrados a trabajar sobre un contrato con el que puedan defenderse si las cosas van mal.
  - Ser capaz de responder a los cambios y no obsesionarse sobre el seguimiento de un plan: es tener la capacidad de adaptación, no decir NO A LOS CAMBIOS, aceptar las sugerencias de los usuarios, sin hacer a un lado la planificación.

### **2.5 El Lenguaje Unificado de Modelado (UML) como soporte de lenguaje orientado a objeto para el modelado de aplicaciones de escritorio**

Cualquier rama de ingeniería o arquitectura ha encontrado útil desde hace mucho tiempo la representación de los diseños de forma gráfica. Desde los inicios de la informática se han utilizado distintas formas de representar los diseños de una forma más bien personal o con algún modelo gráfico. La falta de estandarización en la manera de representar gráficamente un modelo impedía que los diseños gráficos realizados se pudieran compartir fácilmente entre distintos diseñadores.

Se necesitaba por tanto, un lenguaje no sólo para comunicar las ideas a otros desarrolladores, sino también para servir de apoyo en los procesos de análisis de un problema. Con este objetivo se creó el Lenguaje Unificado de Modelado (UML: Unified Modeling Language). UML se ha convertido en ese estándar tan ansiado para representar y modelar la información con la que se trabaja en las fases de análisis y especialmente de diseño.

UML ayuda a los usuarios a entender la realidad desde un punto de vista de la tecnología y la posibilidad de que reflexione antes de invertir y gastar grandes cantidades de dinero en proyectos que no estén seguros en su desarrollo, reduce el costo y el tiempo empleado en la construcción de los módulos que construirán el software. (RUMBAUHG, 2000)

UML es ante todo un lenguaje. Un lenguaje proporciona un vocabulario y una regla para permitir una comunicación. En este caso, este lenguaje se centra en la representación gráfica de un sistema.

Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

- Visualizar: UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- Especificar: UML permite especificar cuáles son las características de un sistema antes de su construcción.
- Construir: a partir de los modelos especificados se pueden construir los sistemas diseñados.
- Documentar: los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

Algunas de las propiedades de UML como lenguaje de modelado son:

- Es un lenguaje distribuido y adecuado a las necesidades de conectividades actuales y futuras, ampliamente utilizado por la industria del software.
- Reemplaza a decenas de notaciones empleadas por otros lenguajes.
- Modela estructuras complejas.
- Las estructuras más importantes que soporta tienen su fundamento en la tecnología orientada a objetos, tales como objetos, clases, componentes y nodos.
- Comportamiento del sistema: casos de usos, diagramas de secuencia, de colaboración, que sirve para evaluar el estado de las máquinas.

Se utilizará como notación el Lenguaje Unificado de Modelado para lograr un mayor entendimiento porque se logra modelar y describir secuencialmente por pasos todos los procesos que se lleva a cabo según la problemática planteada. Sirve porque es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema de software.

### **2.6 Herramientas CASE de desarrollo de software**

Actualmente existen una gran variedad de herramientas CASE (Computer Aided Software Engineering o Ingeniería de Software Asistida por Computadora) para el



proceso de desarrollo de software, por lo que seleccionar alguna se convierte en una difícil tarea.

Se puede definir a las herramientas CASE como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software.

### 2.6.1. Visual Paradigm

Visual Paradigm es una herramienta CASE que utiliza UML como lenguaje de modelado. Está diseñada para una amplia gama de usuarios interesados en construir sistemas de software fiables, con el uso del paradigma orientado a objetos, incluyendo actividades como ingeniería de software, análisis de sistemas y análisis de negocios.

Visual Paradigm es una herramienta fácil de utilizar que emplea las últimas notaciones de UML, ingeniería inversa, generación del código, importación de Rational Rose, exportación/importación XML, generador de impresos, integración del ms Visio, IDE Integration de UML con Eclipse, NetBeans. (G. Booch, 1999)

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

Esta herramienta CASE es muy valiosa en todos los aspectos del ciclo de vida de desarrollo del software por lo cual es la más apropiada según las características del proyecto al cual está vinculado este trabajo investigativo, por lo que se llega a la conclusión que la más idónea es Visual Paradigm debido a todas las facilidades que ofrece:

- Entorno de creación de diagramas para UML 2.1.
- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.

- Capacidades de ingeniería directa (versión profesional) e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones para cada necesidad.
- Disponibilidad de integrarse en los principales IDE.
- Disponibilidad en múltiples plataformas software con licencia gratuita y comercial.

Esta última característica es un factor principal al realizar la selección, porque el Visual Paradigm es una herramienta libre y el Rational es propietario. Constituye una característica negativa porque Cuba no tiene permitido usar software de origen norteamericano. Esto forma parte de las restricciones del bloqueo económico al que esta sometido el país.

### **2.7 CrystalReports**

CrystalReports es un producto de alta tecnología para la creación e integración de reportes con datos provenientes de múltiples fuentes de datos.

Seleccionado por más de 300 socios de negocios por su alta tecnología de elaboración de reportes, CrystalReports es un líder comprobado en el diseño de reportes que cumplan los desafíos que día a día enfrentan los analistas de negocio y los desarrolladores.

Con CrystalReports se puede:

- Transformar rápidamente cualquier fuente de datos en contenido interactivo.
- Integrar estrechamente capacidades de diseño, modificación y visualización en aplicaciones .NET, Java o COM.
- Permitir a los usuarios finales acceder e interactuar con los reportes a través de portales Web, dispositivos móviles y documentos de Microsoft Office®.
- El más completo acceso a datos, CrystalReports provee más opciones de conectividad a datos que cualquier otra herramienta. Incluye más de 30 drivers para acceso a bases de datos relacionales, incluyendo Microsoft SQL Server.
- Diseño integral y opciones de formato. La tecnología flexible de diseño de CrystalReports provee control completo sobre el acceso y la presentación de los datos en los reportes. Elije entre más de 100 opciones de formato, incluyendo

- parámetros, mapas, tablas cruzadas, gráficos e hipervínculos, para incrementar el impacto de sus reportes. También se incluyen más de 160 fórmulas, funciones y operadores para un control completo de la presentación de los datos.
- **Productividad:** el nuevo Repositorio Crystal le permite almacenar elementos claves de los reportes de su organización, tales como objetos de textos, imágenes, sentencias SQL y funciones personalizadas. Gracias a este repositorio central usted puede reutilizar estos objetos en múltiples reportes. Este almacén centralizado de objetos le permite minimizar los esfuerzos de mantenimiento de sus reportes y al mismo tiempo ser más productivo en el diseño de reportes nuevos.
  - Se pueden crear plantillas corporativas para ser aplicadas a todos los reportes de una organización y así evitar dar formato una y otra vez a cada uno de los nuevos reportes que se crean.
  - Habilita la visualización e interacción con los reportes a través de una amplia variedad de dispositivos y entornos. Se utiliza los ReportParts (fracciones de reporte) o el soporte para Smart Tags en Office XP, para proveer acceso, navegación e interacción con los reportes vías portales web, documentos de Microsoft Office o dispositivos inalámbricos.
  - Los visores Web avanzados habilitan a los usuarios finales para realizar búsquedas dentro de los datos de un reporte y exportarlas posteriormente a Microsoft Excel, Word y páginas HTML con el vínculo dinámico al reporte original. Adicionalmente, el reporte completo puede ser exportado a una variedad de formatos incluyendo XML, PDF, HTML y Microsoft Excel.
  - La flexibilidad y el control que se necesita: con flexibles SDKs (Software Development Kits) para aplicaciones .NET, Java y COM y controles para que incluso el usuario final pueda hacer cambios a los reportes, CrystalReports habilita la estrecha integración tanto en aplicaciones Cliente/Servidor como en aplicaciones Web.
  - Control SQL - Edite las sentencias SQL directamente para un control ilimitado sobre la conectividad con la base de datos.
  - ReportApplication Server - Servicio de procesamiento de reportes basado en Web, para realizar integración de contenido dinámico en aplicaciones Web empresariales. (CrystalReport)

### 2.8 ZedGraph

ZedGraph es una biblioteca de clases, UserControl de Windows Forms, y ASP Web accesible para la creación de 2D línea, barra, pastel y gráficos de datos arbitrarios. Las clases proporcionan un alto grado de flexibilidad, casi todos los aspectos de la gráfica de usuario pueden ser modificados. Al mismo tiempo, el uso de las clases es sencillo por proporcionar valores por defecto para todos los atributos del gráfico. Las clases incluyen el código para la elección de la escala apropiada y rangos de tamaños de paso sobre la base de la gama de valores de datos se representan gráficamente.

### 2.9 Arquitectura en Capas

Es un estilo de programación, su objetivo primordial es la separación de la capa de presentación, capa de negocio y la capa de datos.

- La ventaja principal es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, sólo se ataca al nivel requerido sin tener que revisar entre código mezclado.
- En el diseño de sistemas informáticos actuales se suele usar las arquitecturas multilineal o programación por capas.
- Permite distribuir el trabajo de una aplicación por niveles; cada grupo de trabajo está totalmente abstraído del resto de niveles.

Todas las aplicaciones basadas en n-capas permitirán trabajar con clientes ligeros, tal como navegadores de Internet, WebTV, Teléfonos Inteligentes, PDAs (Personal Digital Assistants o Asistentes Personales Digitales) y otros dispositivos preparados para conectarse a Internet.

#### 2.9.1 Arquitectura de tres capas

En el diseño de los sistemas informáticos actuales se suelen usar las arquitecturas multinivel o programación por capas. En dichas arquitecturas a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables (que pueden ampliarse con facilidad en caso de que las necesidades aumenten). El diseño más utilizado actualmente es en tres capas.

Una arquitectura común de los sistemas de información que abarca una interfaz para el usuario y el almacenamiento persistente de datos se conoce con el nombre de arquitectura de tres capas. La calidad tan especial de la arquitectura de tres capas consiste en aislar la lógica de la aplicación y en convertirla en una capa intermedia bien definida y lógica del software. En la capa presentación se realiza relativamente poco procesamiento de la aplicación; las ventanas envían a la capa intermedia peticiones de trabajo y este se comunica con la capa de almacenamiento del extremo posterior. (Larman, 2004)

**Capa de presentación:** es la que ve el usuario (se le denomina "capa de usuario"). También es conocida como interfaz gráfica y debe tener la característica de ser "amigable" (entendible y fácil de usar) para el usuario. Esta capa se comunica únicamente con la capa de negocio.

**Capa de negocio:** se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las tareas y reglas que rigen el proceso. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él.

**Capa de datos:** es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo tipo de mecanismo de almacenamiento de datos persistente, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

### 2.10 Sistema Gestor de Base de Datos

Un Sistema Gestor de Bases de Datos (SGBD) o DBMA (DataBase Management System) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantiza la seguridad e integridad de los mismos.

Algunos ejemplos de SGBD son Oracle, DB2, PostgreSQL, MySQL y MS SQL Server.

Las características de un Sistema Gestor de Base de Datos SGBD son:

- Abstracción de la información: los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos.
- Independencia: la independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- Redundancia mínima: un buen diseño de una base de datos logrará evitar la aparición de información repetida o redundante.
- Consistencia: en aquellos casos en los que no se ha logrado esta redundancia nula, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.
- Seguridad: la información almacenada en una base de datos puede llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentra segura frente a usuarios malintencionados, que intenten leer información privilegiada; frente a ataques que deseen manipular o destruir la información; o simplemente ante las torpezas de algún usuario autorizado pero despistado.
- Integridad: se trata de adoptar las medidas necesarias para garantizar la validez de los datos almacenados. Es decir, se trata de proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada.
- Respaldo y recuperación: los SGBD deben proporcionar una forma eficiente de realizar copias de respaldo de la información almacenada en ellos y de restaurar a partir de estas copias los datos que se hayan podido perder.
- Control de la concurrencia: en la mayoría de entornos (excepto el doméstico), lo más habitual es que sean muchas las personas que acceden a una base de datos, bien para recuperar información, bien para almacenarla. (SQL-Server-2000)

### 2.10.1 SQL Server 2000

SQL Server es un gestor de bases de datos relacionales (SGDBR o RDBMS: Relational Database Management System) para trabajar con grandes volúmenes de información y

la capacidad de cumplir con los requisitos de proceso de esta para aplicaciones comerciales y sitios Web.

Es un sistema administrador para bases de datos relacionales basadas en la arquitectura cliente/servidor. Transact-SQL es el lenguaje que emplea para mandar peticiones entre el cliente y el servidor. Es un lenguaje exclusivo de SQL Server, pero basado en el lenguaje SQL estándar, utilizado por casi todos los tipos de bases de datos relacionales que existen. SQL Server otorga a los administradores una herramienta potencialmente robusta, provista de las herramientas suficientes que le permiten mantener un óptimo nivel de seguridad en la utilización de los recursos del sistema y de la base de datos que en este camino van cogidos de la mano.

La finalidad de SQL Server 2000 es analizar y administrar datos, dar mayor escalabilidad, disponibilidad y seguridad a las aplicaciones de análisis y los datos empresariales y potenciar las aplicaciones dando una mayor rentabilidad. (SQL-Server-2000)

SQL Server permite la creación de procedimientos almacenados, los cuales consisten en instrucciones que se almacenan dentro de una base de datos de SQL Server, realizados en lenguaje SQL. Son procedimientos que se guardan semicompilados en el servidor y que pueden ser invocados desde el cliente. Se ejecutan más rápido que instrucciones SQL independientes.

Luego de analizar el SGBD descrito anteriormente, se realiza la selección de SQL Server 2000 porque es el mejor SGBD para Windows, es compatible con las tecnologías, se utilizan para el desarrollo de la Aplicación de Escritorio y puede aportar muchos beneficios. Dado las características y teniendo en cuenta que todas las Bases de Datos (BD) de la Universal están sobre este gestor y los desarrolladores de la Aplicación tienen experiencia en el trabajo con este SGBD se reafirma la selección de SQL Server 2000.

Las características de SQL Server 2000 incluyen:

- Integración con Internet.

El motor de base de datos de SQL Server 2000 incluye compatibilidad integrada con XML. También cuenta con las características de escalabilidad, disponibilidad y

seguridad necesarias para operar el componente de almacenamiento de datos de los sitios Web de mayor tamaño.

- Escalabilidad y disponibilidad.

El mismo motor de base de datos se puede utilizar en un intervalo de plataformas desde equipos portátiles que ejecutan Microsoft Windows® 98 por medio de grandes servidores con varios procesadores que ejecutan Microsoft Windows 2000, Data Center. SQL Server 2000 Enterprise admite características como servidores federados, vistas indizadas y soporte para memorias grandes, que le permiten ampliarse a los niveles de rendimiento requeridos por los mayores sitios Web.

- Características de base de datos corporativas.

El motor de base de datos relacional de SQL Server 2000 admite las características necesarias para satisfacer los exigentes entornos de procesamiento de datos. El motor de base de datos protege la integridad de los datos a la vez que minimiza la carga de trabajo que supone la administración de miles de usuarios y modifica la base de datos simultáneamente. Las consultas distribuidas de SQL Server 2000 permiten hacer referencia a datos de varios orígenes como si fuesen parte de una base de datos de SQL Server 2000.

- Facilidad de instalación, distribución y utilización.

SQL Server 2000 incluye un conjunto de herramientas administrativas y de desarrollo que mejora el proceso de instalación, distribución, administración y uso de SQL Server en varios sitios.

- Almacenamiento de datos.

SQL Server 2000 incluye herramientas para extraer y analizar datos de resumen para el procesamiento analítico en línea. SQL Server incluye también herramientas para diseñar gráficamente las bases de datos y analizar los datos mediante preguntas en inglés. (Ojeda, 2001)

### 2.11 Lenguajes de programación

Es un conjunto de símbolos junto a un conjunto de reglas para combinar dichos símbolos que se usan para expresar programas. Constan de un léxico, una sintaxis y una semántica.

Léxico: conjunto de símbolos permitidos o vocabulario.



Sintaxis: reglas que indican cómo realizar las construcciones del lenguaje.

Semántica: reglas que permiten determinar el significado de cualquier construcción del lenguaje.

Tipos de lenguajes: atendiendo al número de instrucciones necesarias para realizar una tarea específica podemos clasificar los lenguajes informáticos en dos grandes bloques:

- bajo nivel
- alto nivel

Lenguaje de bajo nivel: es el tipo de lenguaje que cualquier computadora es capaz de entender. Se dice que los programas escritos en forma de ceros y unos están en lenguaje de máquina, porque esa es la versión del programa que la computadora realmente lee y sigue.

Lenguaje de alto nivel: son lenguajes de programación que se asemejan a las lenguas humanas usando palabras y frases fáciles de entender.

### 2.11.1 Lenguajes de programación C#

C# es algo más que un simple lenguaje de programación, porque reúne todas las características que son demandadas en un lenguaje de programación moderno. Está diseñado especialmente para los desarrolladores actuales porque comparte la filosofía y el entorno de Microsoft .NET junto a la orientación a objetos, con todas las virtudes y ventajas que ofrece esta plataforma para los programadores.

Ventajas del lenguaje:

- Orientación a objetos: como todo lenguaje de programación de propósito general actual, C# es un lenguaje orientado a objetos. Una diferencia de este enfoque orientado a objetos respecto al de otros lenguajes como C++ es que el de C# es más puro, en tanto que no admiten ni funciones ni variables globales sino que todo el código y datos han de definirse dentro de definiciones de tipos de datos, lo que reduce problemas por conflictos de nombres y facilita la legibilidad del código.
- Orientación a componentes: la propia sintaxis de C# incluye elementos propios del diseño de componentes que otros lenguajes tienen que simular mediante construcciones más o menos complejas. Es decir, la sintaxis de C# permite definir cómodamente propiedades (similares a campos de acceso controlado),

eventos (asociación controlada de funciones de respuesta a notificaciones) o atributos (información sobre un tipo o sus miembros).

- Seguridad de tipos: C # incluye mecanismos que permiten asegurar que los accesos a tipos de datos siempre se realicen correctamente, lo que evita que se produzcan errores difíciles de detectar por acceso a memoria no perteneciente a ningún objeto y es especialmente necesario en un entorno gestionado por un recolector de basura. A diferencia de Java, C# incluye delegados, que son similares a los punteros a funciones de C++ pero siguen un enfoque orientado a objetos, pueden almacenar referencias a varios métodos simultáneamente, y se comprueba que los métodos a los que apunten tengan parámetros y valor de retorno del tipo indicado al definirlos. Pueden definirse métodos que admitan un número indefinido de parámetros de un cierto tipo, y a diferencia lenguajes como C/C++, en C# siempre se comprueba que los valores que se les pasen en cada llamada sean de los tipos apropiados.
- Extensibilidad de operadores: para facilitar la legibilidad del código y conseguir que los nuevos tipos de datos básicos que se definan a través de las estructuras estén al mismo nivel que los básicos predefinidos en el lenguaje, al igual que C++ y a diferencia de Java, C# permite redefinir el significado de la mayoría de los operadores, incluidos los de conversión, tanto para conversiones implícitas como explícitas cuando se apliquen a diferentes tipos de objetos.
- Eficiente: en principio, en C# todo el código incluye numerosas restricciones para asegurar su seguridad y no permite el uso de punteros. Sin embargo, y a diferencia de Java, en C# es posible saltarse dichas restricciones al manipular objetos a través de punteros. Para ello basta marcar regiones de código como inseguras (modificador unsafe) y podrán usarse en ellas punteros de forma similar a como se hace en C++, lo que puede resultar vital para situaciones donde se necesite una eficiencia y velocidad procesamiento muy grandes.
- Compatible: para facilitar la migración de programadores, C# no sólo mantiene una sintaxis muy similar a C, C++ o Java que permite incluir directamente en código escrito en C# fragmentos de código escrito en estos lenguajes, sino que el CLR también ofrece, a través de los llamados Platform Invocation Services (PInvoke), la posibilidad de acceder a código nativo escrito como funciones sueltas no orientadas a objetos tales como las DLLs de la API Win32. Nótese

que la capacidad de usar punteros en código inseguro permite que se pueda acceder con facilidad a este tipo de funciones, porque éstas muchas veces esperan recibir o devuelven punteros. (SECO, 2001)

Combina los mejores elementos de múltiples lenguajes de amplia difusión como C++, Java, Visual Basic o Delphi y permite al igual que estos lenguajes programar fácilmente a bajo nivel. Es un lenguaje orientado a objetos, posee una sintaxis similar a la de C++, implementa una fuerte política de seguridad de tipos, elimina la utilización de punteros aunque mantiene una reserva en caso extremo de su uso.

### 2.12 IDE de desarrollo

Un entorno de desarrollo integrado o IDE (de la inglé Integrated Development Environment), es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica de forma amigable. Los IDE pueden ser aplicaciones por sí solos o pueden ser parte de aplicaciones existentes.

#### 2.12.1 Visual Studio.Net

Como herramientas de desarrollo de aplicaciones .NET se encuentran Visual Studio.NET y Web Matrix. Se decidió trabajar con Visual Studio.NET porque permite diseñar, desarrollar, depurar e instalar soluciones basadas en .NET. Todos los lenguajes que son incorporados a Visual Studio pueden acceder a un conjunto común de herramientas, editores y diseñadores. Se pueden crear aplicaciones con formularios Windows o formularios Web que integran datos y lógica propia de la aplicación, así como Servicios Web. Facilita la creación y personalización visual de controles de formulario Windows y Web. Brinda posibilidades para el acceso a datos y asistentes para realizar determinadas acciones. Además es una herramienta de desarrollo multilinguaje integrado (Visual Basic .NET, C++, C# y J#) lo que la hace una de las herramientas más completa alcanzadas en la actualidad.

Características de Visual Studio.Net:

- Detección de errores automáticos: es posible ahorrar muchas horas de trabajo al utilizar Visual Studio.Net para detectar los errores antes de intentar ejecutar su

- aplicación. Los errores se subrayan de la misma forma que lo hacen algunos procesadores de textos mientras se escribe.
- Herramientas de depuración: Visual Studio.Net mantiene sus herramientas de depuración legendarias que le permiten observar su código en acción y seguir la pista del contenido de las variables.

Ventajas de Visual Studio .Net:

- Arquitectura flexible y ágil de aplicaciones.
- Creada para los servicios Web XML.
- Constituye un modelo sencillo, flexible y basado en estándares para integrar, ampliar y publicar aplicaciones.
- Máxima productividad del desarrollador.
- Soporta varios lenguajes de programación.
- Pueden reutilizar las aplicaciones escritas en cualquier lenguaje de programación.
- Operaciones mejoradas.
- Aumenta el rendimiento, la escalabilidad y la fiabilidad.
- Ofrece al usuario un tiempo en actividad percibido del 100%.
- Elimina el problema de conflictos entre versiones. Las nuevas aplicaciones pueden ejecutar aplicaciones existentes. (SECO, 2001)

### 2.13 Conclusiones

En este capítulo se definen las principales tecnologías que se utilizan en el desarrollo de la aplicación, se justificó la elección de la utilización de aplicaciones de escritorio o Desktop. Para la construcción de la herramienta propuesta se decide desarrollarla en la plataforma .Net utilizando el IDE Visual Studio.Net y lenguaje de programación C#, se sigue la metodología RUP-Ágil. Los diagramas serán modelados en la herramienta CASE Visual Paradigm, los reportes que se generen en la aplicación serán diseñados por la herramienta CrystalReports, todo esto acompañado del Lenguaje de Modelado UML.

---

## **CAPÍTULO 3 PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA**

### **3.1 Introducción**

En el presente capítulo se describen los procesos de negocio, actores del negocio y trabajadores del negocio. Se expone, además, el diagrama de casos de uso del Negocio (en lo adelante DCUN) y se detallan todos los casos de uso del negocio a través de sus descripciones textuales.

En un segundo momento se plantean los requisitos funcionales y no funcionales y finalmente se presenta la solución desde la perspectiva del sistema, se definen sus actores, se exhibe el Diagrama de casos de uso del Sistema (en lo adelante DCUS) y luego sus descripciones textuales.

### **3.2 Modelo de negocio**

Un Modelo de Negocio describe los procesos de negocio que se desarrollan dentro de un empresa en términos de Casos de Uso del Negocio y actores del negocio que se corresponden con los procesos de negocio y sus clientes respectivamente, además, presenta el entorno de la organización desde la perspectiva de su uso y esquematiza cómo proporciona valor a sus usuarios. (PRESSMAN, 2005)

#### **3.2.1 Actores y Trabajadores del Negocio**

<b>Actores</b>	<b>Justificación</b>
<b>Director</b>	Se utiliza para identificar a la persona con la máxima autoridad y responsabilidad al frente de una entidad otorgada por los niveles superiores de dirección. Debe tener un amplio prestigio profesional y personal.
<b>Trabajadores</b>	<b>Justificación</b>
<b>Planificador</b>	Se utiliza para identificar al técnico encargado de actualizar la información de los Planes que intervienen en el análisis económico-financiero

	de la empresa.
<b>Especialista económico</b>	Se utiliza para identificar a los técnicos en gestión económica encargados de trabajar los indicadores recogidos dentro del análisis económico-financiero de la empresa.

Tabla 1 Actores y Trabajadores del Negocio.

### 3.2.2 Procesos del negocio

Fundamentalmente se desarrolla dentro de la Dirección Económica-Contable un proceso, que concierne a:

1. La elaboración del análisis económico-financiero de la empresa.

En cada mes y al finalizar el año se elabora análisis económico-financiero de la Empresa con la información concerniente a los datos recogidos por otra parte, es posible que en cualquier momento del año, se produzca un cambio considerable en los indicadores que conforman este análisis económico-financiero producto de las diferentes actividades económicas que realiza la empresa, momento en el cual se sugiere la re-elaboración del análisis. De cualquier forma, es el director apoyado por la Dirección Económica-Contable de la Empresa el que toma la decisión de reelaborarlo. Una vez tomada la decisión, se le comunica a la mencionada Dirección Económica-Contable la cual se encarga entonces de imprimirlo y distribuir copias a la Administración para que esta evalúe el comportamiento de los indicadores plasmados en el análisis.

### 3.2.3 Diagrama de Casos de Uso del Negocio

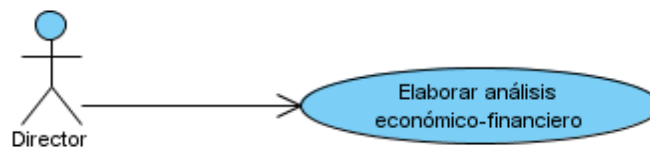


Figura 2 Diagrama de Casos de Uso del Negocio.

### 3.2.4 Descripción textual del caso de uso del negocio

**Caso de uso: Elaborar análisis económico-financiero.**

<b>Caso de Uso</b>	<b>Elaborar análisis económico-financiero.</b>	
<b>Actores</b>	Director.	
<b>Propósito</b>	Capacidad del usuario para elaborar el análisis económico-financiero.	
<b>Resumen</b>	El caso de uso se inicia en el momento en que el director de la empresa solicita elaborar el análisis económico-financiero de la empresa al finalizar cada mes.	
<b>Curso Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Negocio</b>	
1. El caso de uso inicia cuando el director de la empresa le comunica al Especialista de la Dirección Económica-Contable la decisión de elaborar el análisis económico-financiero.	2. El Especialista de la Dirección Económica-Contable le indica al Planificador que realice la entrada de los planes de los indicadores.	
	3. El Planificador introduce los planes de los principales indicadores.	
	4. La Dirección Económica-Contable revisa los planes introducidos por el Planificador. Si los planes introducidos están correctos le da confirmación al Planificador.	
	5. Se elabora el análisis económico-financiero y se entrega al Director.	
6. El caso de uso termina cuando el director de la empresa recibe la copia del análisis económico-financiero y lo revisa. Si el informe está correcto le da confirmación al especialista.		
<b>Curso Alternativo de Eventos</b>		
6. Si el informe presenta errores el director le notifica al especialista, con los errores señalados.	4. En caso de error se le informa al Planificador la corrección de los planes, una vez corregido los errores, el caso de uso regresa a la acción 4 del flujo normal de eventos.	
7. El especialista corrige los errores señalados y el caso de uso regresa a la acción 5 del flujo		

normal de eventos.	
<b>Pos-condiciones</b>	El análisis económico-financiero queda elaborado.

Tabla 2 Descripción textual. CU Elaborar análisis económico-financiero.

### 3.3 Requisitos Funcionales

Los requisitos funcionales definen las funciones que el sistema será capaz de realizar. Describen las transformaciones que el sistema realiza sobre las entradas para producir salidas, de esta forma se puede decir que los requisitos funcionales del sistema en cuestión quedan enunciados como se muestran a continuación:

- RF1: seleccionar el servidor y la base de datos a donde el sistema se va a conectar.
- RF2: permitir que a la entrada siempre se registre el usuario y la contraseña.
- RF3: generar informes que se manejan dentro de la empresa relacionados con la Dirección Económica Contable al finalizar cada mes y cada año.
- RF4: imprimir y exportar a diferentes formatos los informes generados.
- RF5: permitir que cada uno de los planes se inserten y modifiquen sus datos cada vez que sea necesario.
- RF6: construir un gráfico con cada uno de los valores de los indicadores.
- RF7: configurar los gráficos permitiendo realizar diferentes operaciones sobre él.
- RF8: permitir que cada uno de los comentarios se inserten y modifiquen sus datos cada vez que sea necesario.

### 3.4 Requisitos no funcionales

Los atributos o requisitos no funcionales representan propiedades o cualidades que el producto de software debe tener lo que hace del mismo que sea más atractivo, más usable, rápido y más confiable.

#### Requisitos de usabilidad

1. El sistema debe poder ser usado por cualquier persona que posea conocimientos básicos de computación (trabajo con ventanas, trabajo con menú, trabajo con el mouse) y de contabilidad financiera.



2. El sistema está concebido para ser usado por los trabajadores de la Empresa Comercializadora y de Servicios de Productos Universales de Las Tunas, por lo que es necesario que cuente con una interfaz entendible y amigable, de fácil acceso y manipulación y que facilite la localización de las diferentes funcionalidades presentes en ella.
3. La información deberá estar disponible en todo momento.

### **Apariencia o interfaz externa**

1. El sistema debe tener una apariencia profesional con un diseño gráfico sencillo de tonalidades de colores claros que facilite la localización de las diferentes funciones del sistema.

### **Requisitos de portabilidad y operatividad**

1. La aplicación debe ser compatible con los Sistemas Operativos Microsoft Windows 2000 NT, Microsoft Windows XP y cualquier distribución Windows.

### **Requisitos de rendimiento**

1. El sistema debe ser lo más eficiente, rápido y preciso para poder lograr un tiempo de respuesta adecuado.

### **Requisitos de software del sistema.**

1. Las computadoras que utilizarán el software deben tener instalado:
  - a) Microsoft Windows 2000 NT, Microsoft Windows XP Profesional o cualquier distribución de Windows.
  - b) Framework de .NET en su versión 2.0 o superior.

### **Requisitos de seguridad**

1. El sistema debe tener confidencialidad e integridad.
2. Cada usuario debe tener solo los permisos necesarios para realizar las operaciones que le sean permitidas.
3. Mantener la integridad de la información, es decir que no se ha transformado durante su almacenamiento o transporte.

### **Requisitos de hardware del sistema**

1. Las computadoras que utilizarán la aplicación deberán contar con un microprocesador con velocidad de procesamiento superior a un 1 GHz<sup>5</sup> y memoria RAM<sup>6</sup> de 256 MB o superior.

## 3.5 Descripción del sistema propuesto

El Modelo de Casos de Uso ayuda al cliente, a los usuarios y a los desarrolladores a llegar a un acuerdo sobre cómo utilizar el sistema. La mayoría de los sistemas tienen muchos tipos de usuarios. Cada usuario se representa mediante un actor. Los actores utilizan el sistema al interactuar con los casos de uso. (PRESSMAN, 2005)

Luego de analizado los requisitos funcionales y aplicar los diferentes patrones de casos de uso queda elaborado el diagrama de casos de uso del sistema como se muestra a continuación:

### 3.5.1 Diagrama de Casos de Uso del Sistema

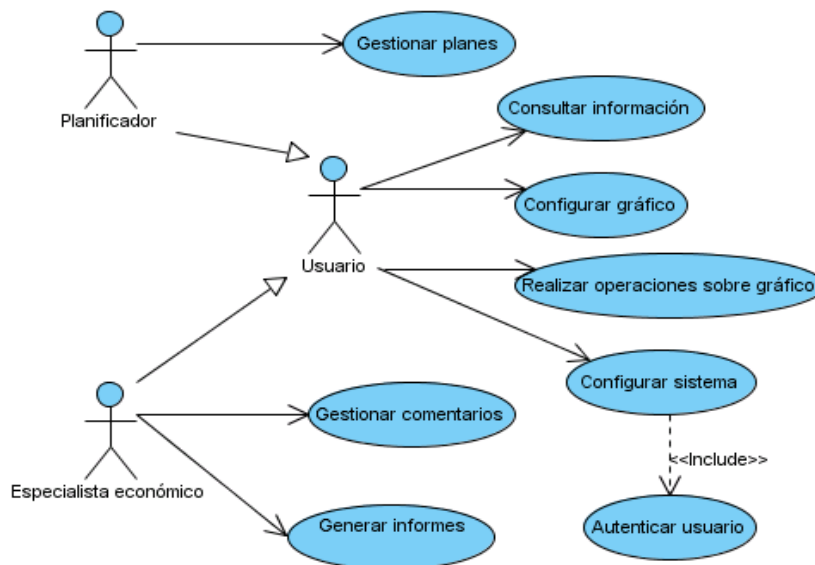


Figura 3 Diagrama de Casos de Uso del Sistema.

### 3.5.2 Descripción textual de los casos de uso del Sistema

<sup>5</sup> Gigahercio: es un múltiplo de la unidad de medida de frecuencia hercio (Hz) y equivale a 109 Hz.

<sup>6</sup> Memoria de Acceso Aleatorio.

A modo de aclaración es válido decir que durante la realización del trabajo de diploma sólo se expondrán en el documento las descripciones y realizaciones de los casos de uso arquitectónicamente significativos.

### Caso de uso: Gestionar Planes.

<b>Caso de Uso</b>	<b>Gestionar planes.</b>	
<b>Actores</b>	Planificador, Administrador.	
<b>Propósito</b>	Capacidad del usuario para gestionar los planes.	
<b>Resumen</b>	El caso de uso comienza cuando el usuario necesita gestionar los planes para confeccionar los gráficos de los indicadores. El caso de uso termina cuando el usuario completa el registro de los planes.	
<b>Referencias</b>	<b>RF5</b>	
<b>Precondiciones</b>	El usuario debe estar autenticado.	
<b>Prioridad</b>	Crítico.	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El caso de uso inicia cuando el usuario selecciona del menú principal la opción Gestión de planes.	2. El sistema muestra la pantalla Gestionar Plan con los indicadores disponibles y los datos de los planes gestionados. [Ver interfaz 1]	
3. El usuario selecciona un indicador de la lista de indicadores mostrados.		
4. De las opciones que muestra el sistema, el usuario selecciona una de ellas.[Ver interfaz 1] - Adicionar Plan. - Editar Plan.	5. En dependencia de la opción seleccionada por el usuario, el sistema ejecuta una operación: <ul style="list-style-type: none"> <li>• Si el usuario selecciona la opción Adicionar plan. [Ver sección Adicionar plan]</li> <li>• Si el usuario selecciona la opción Editar plan. [Ver sección Editar plan]</li> </ul>	

6. El caso de uso termina cuando el sistema gestiona los planes según la selección del usuario.

## Interfaz 1

The screenshot shows a software interface with a menu bar (Inicio, Indicadores, Gestión, Informes) and two main buttons: 'Gestión de planes' and 'Gestión de comentarios'. Below this is a window titled 'Gestionar Plan' which contains a dropdown menu for 'Indicadores' set to 'Ventas por Estructuras Ropa Reciclada', and buttons for 'Actualizar datos', 'Editar plan', and 'Cerrar'. A table displays monthly data for the year 2011, with a total annual plan of 45505.

Indicadores	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre
Ventas Netas	2	4	6	8	10	12	14	16	18	20	22	24
→Ropa Reciclada	8500	13000	19500	26000	32500	39000	45500	45501	45502	45503	45504	45505
→Otros Productos	1	2	3	4	5	6	7	8	9	10	11	12
Utilidades antes de Impu...	1	2	3	4	5	6	7	8	9	10	11	12
Costos	1	2	3	4	5	6	7	8	9	10	11	12
Ingresos	1	2	3	4	5	6	7	8	9	10	11	12

At the bottom of the window, it says 'Plan del Año: 45505'. The system logo 'SistemaGTR gráficos en tiempo real' is visible in the bottom right corner.

## Sección "Adicionar Plan"

1. El usuario selecciona la opción Adicionar Plan.

2. El sistema muestra la pantalla Plan por meses para Adicionar los planes. [Ver interfaz 2]

3. El sistema muestra un formulario con los datos necesarios que el usuario debe introducir.

4. El usuario introduce los datos necesarios que aparecen a continuación y da clic en el botón Aceptar.

5. El sistema muestra el mensaje. "Datos de planes guardados correctamente" y la ventana Plan por meses se cierra.

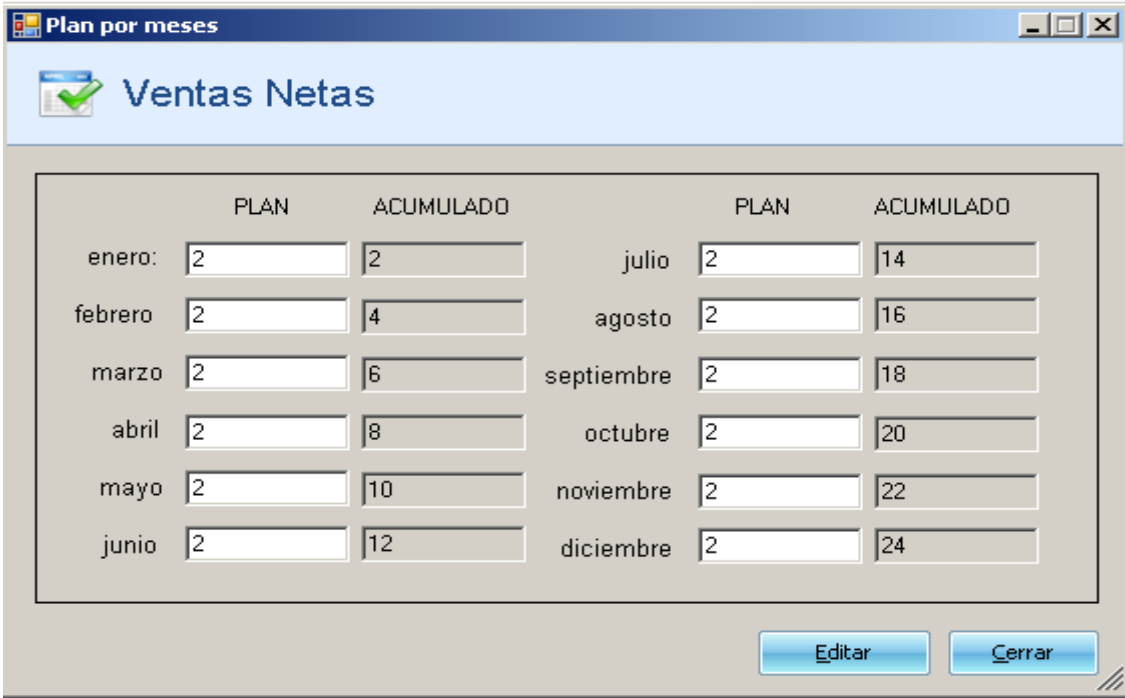
- enero
- febrero
- marzo
- ...
- diciembre

Flujos Alternos	
	2. El sistema muestra el mensaje “Seleccione un indicador”.
4. El usuario introduce mal los datos.	5. El sistema muestra el mensaje “Algunos planes se encuentra vacios”.
4. El usuario selecciona el botón Cerrar.	5. El sistema cierra la pantalla Plan por meses.

**Interfaz 2**

The screenshot shows a window titled "Plan por meses" with a sub-header "Ventas Netas" and a green checkmark icon. Below this is a table with two columns: "PLAN" and "ACUMULADO". The table lists months from January to December. Each month has two input fields, one for "PLAN" and one for "ACUMULADO", both containing the value "0". At the bottom right of the window are two buttons: "Aceptar" and "Cerrar".

Sección “Editar Plan”	
1. El usuario selecciona la opción Editar Plan.	2. El sistema muestra la pantalla Plan por meses para Editar los planes. [Ver interfaz 3]
	3. El sistema muestra un formulario con los datos necesarios que el usuario debe introducir.
4. El usuario introduce los datos necesarios que aparecen a continuación y da clic en el botón Aceptar.	5. El sistema muestra el mensaje. “Datos de planes editados correctamente” y la ventana Plan por meses se cierra.

<ul style="list-style-type: none"> <li>• enero</li> <li>• febrero</li> <li>• marzo</li> <li>• ...</li> <li>• diciembre</li> </ul>	
<b>Cursos Alternos</b>	
	2. El sistema muestra el mensaje "Seleccione un indicador".
4. El usuario introduce mal los datos.	5. El sistema muestra el mensaje "Algunos planes se encuentra vacios".
4. El usuario selecciona el botón Cerrar.	5. El sistema cierra la pantalla Plan por meses.
<b>Interfaz 3</b>	
	
<b>Pos-condiciones</b>	Se gestionan los planes.

**Tabla 3 Descripción textual CU Gestionar Planes.**

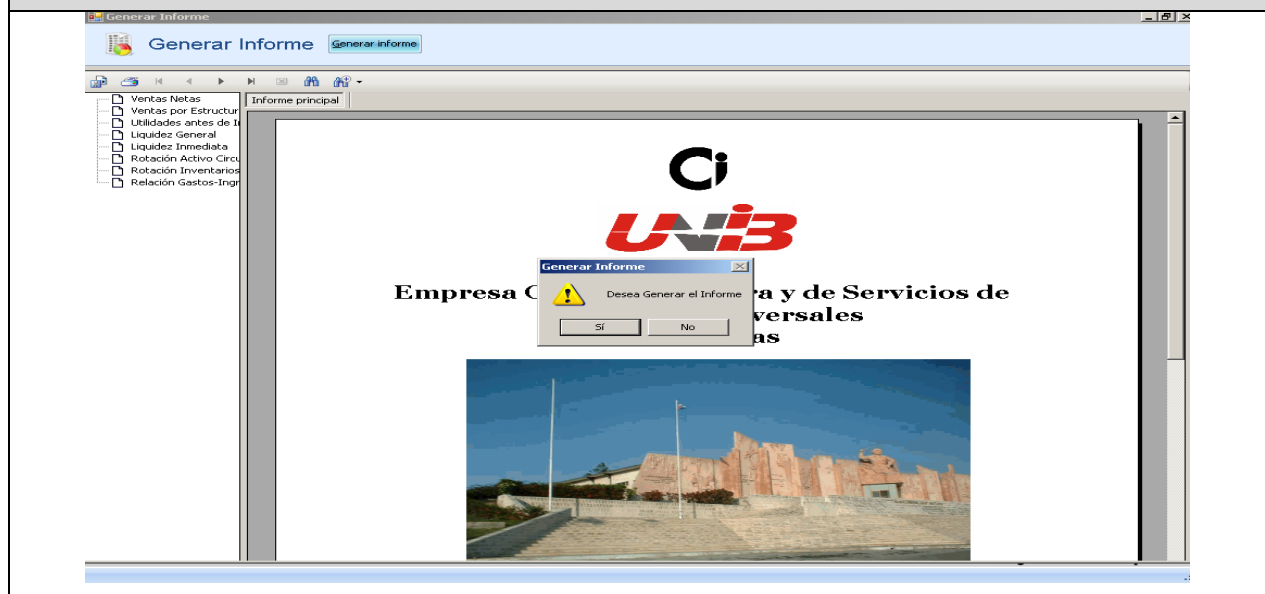
**Caso de uso: Generar Informe.**

<b>Caso de Uso</b>	<b>Generar Informe.</b>
<b>Actores</b>	Administrador.
<b>Propósito</b>	Capacidad del usuario para generar el informe del análisis económico-financiero.
<b>Resumen</b>	El caso de uso comienza cuando el usuario necesita generar el informe del análisis económico-financiero y termina cuando el sistema genera el informe.
<b>Referencias</b>	<b>RF3</b>
<b>Precondiciones</b>	El usuario debe estar autenticado.
<b>Prioridad</b>	Crítico.

**Curso Normal de Eventos**

<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El caso de uso inicia cuando el usuario selecciona del menú principal Generar Informe.	2. El sistema muestra la pantalla Generar Informe con el informe realizado hasta la fecha. [Ver Interfaz 4]
3. El usuario selecciona el botón Generar Informe.	4. El sistema muestra el mensaje “Desea Generar el Informe”.
5. El usuario selecciona el botón Sí. [Ver interfaz 4]	6. El caso de uso termina cuando el sistema muestra el mensaje “Informe creado satisfactoriamente”.

**Interfaz 4**



Flujos Alternos	
	4. El sistema muestra el mensaje “El informe de este mes esta creado”.
5. El usuario selecciona el botón No.	
<b>Pos-condiciones</b>	Se genera el informe análisis económico-financiero.

**Tabla 4 Descripción textual. Generar Informe.**

**Caso de uso: Configurar Sistema.**

Caso de Uso	Configurar Sistema.
<b>Actores</b>	Administrador, Usuario, Planificador.
<b>Propósito</b>	Capacidad del usuario para autenticarse en el sistema.
<b>Resumen</b>	El caso de uso comienza cuando el usuario necesita introducir los datos necesarios para configurar el sistema. El sistema realiza la opción seleccionada por el usuario y termina el Caso de Uso.
<b>Referencias</b>	RF1
<b>Prioridad</b>	Crítico.
Curso Normal de Eventos	
Acción del Actor	Respuesta del Sistema
	1. El sistema muestra la pantalla Configurar Sistema con los campos requeridos para su configuración. [Ver interfaz 5]
2. El usuario introduce los datos solicitados por el sistema que aparecen a continuación y da clic en el botón Conectar. <ul style="list-style-type: none"> <li>• Servidor.</li> <li>• Base de datos.</li> <li>• Usuario.</li> <li>• Password.</li> </ul>	3. El caso de uso termina cuando el usuario se conecta al sistema.





Interfaz 5	
	
Flujos Alternos	
<p>2. El usuario deja campos vacíos o introduce punto y coma en los campos de textos.</p>	<p>3. El sistema muestra un icono rojo al lado del campo incorrecto con un mensaje de error. [Ver interfaz 6]</p>
<p>2. El usuario da clic en el botón Cancelar.</p>	<p>3. El sistema cierra la pantalla Configurar Sistema.</p>
Interfaz 6	
	
<b>Pos-condiciones</b>	Se configura el sistema.

Tabla 5 Descripción textual. Configurar Sistema.

### **3.6 Conclusiones**

En el presente capítulo se describe detalladamente cómo se modeló a través del DCUN y luego se precisaron las características del sistema, los requisitos funcionales y no funcionales concluyendo que el software a desarrollar es de fácil aplicación. A partir de este punto se puede comenzar a construir el sistema que constituye la propuesta de solución.

## **CAPÍTULO 4 CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA**

### **4.1 Introducción**

El presente capítulo describe la solución en términos de diagrama de clases del diseño dividido por secciones de casos de uso para facilitar una mejor comprensión, quedan definidos los estándares de la interfaz de la aplicación, se presenta el diseño de la base de datos a través del diagrama Entidad-Relación y el diagrama de clases persistentes. Se define además, el modelo de despliegue donde se precisan los componentes que conforman la estructura física de la aplicación.

### **4.2 Principios de diseño**

Definir principios de diseño en el desarrollo de una aplicación permite que la misma se convierta en una herramienta útil y atractiva para el usuario por su estética y su utilidad. Además posibilitan también que la misma brinde adecuadamente la información que le sea útil al usuario.

Como principios de diseño del sistema se tienen:

- ❖ Brindar una interfaz sencilla y amigable, de forma tal que cualquier persona con un mínimo dominio de conocimientos informáticos trabaje con la aplicación sin presentar dificultades notables.
- ❖ Mostrar al usuario, siempre que vaya a realizar una acción notable sobre la aplicación, un mensaje de confirmación que le permita asegurarse que es correcta la opción seleccionada.
- ❖ Los mensajes que son expuestos al usuario deben ser precisos y de fácil comprensión.
- ❖ Permitir que exista contraste entre los colores de los textos y el fondo, y por otra parte, el tamaño de la fuente sea adecuado a la vista del usuario.

#### **4.2.1 Estándares de la interfaz de la aplicación**

##### **Barra de Título**

- ❖ El título debe ubicarse a la izquierda y los botones minimizar y cerrar, ubicados del lado derecho.
- ❖ No deben usarse colores fuertes que impidan la visibilidad de los elementos anteriores.

### **Barra de Menú**

- ❖ Los menús se ordenan según la importancia de las funcionalidades que brindan.
- ❖ Para una mejor claridad de la barra de menú se utilizan líneas divisorias para lograr un acceso rápido.

### **Punteros del Ratón**

- ❖ Se usan los punteros estándares del sistema operativo cuando se ejecuta la aplicación.

## **4.3 Patrones de diseño**

Un patrón de diseño es una descripción de clases y objetos comunicándose entre si adaptada para resolver un problema de diseño general en un contexto particular. Cada patrón describe un problema que ocurre una y otra vez en el entorno, para describir después el núcleo de la solución a ese problema, de tal manera que esa solución pueda ser usada más de un millón de veces sin hacerlo ni siquiera dos veces de la misma forma. (Gamma, 1995)

### **Características**

- ❖ Solucionar un problema: los patrones capturan soluciones, no sólo principios o estrategias abstractas.
- ❖ Ser un concepto probado: capturan soluciones demostradas, no teorías o especulaciones.
- ❖ La solución no es obvia: los mejores patrones generan una solución a un problema de forma indirecta.
- ❖ Describe participantes y relaciones entre ellos: describen módulos, estructuras del sistema y mecanismos complejos.

- ❖ El patrón tiene un componente humano significativo: todo software proporciona a los seres humanos confort y calidad de vida (estética y utilidad). (Gamma, 1995)

### Clasificación de patrones de diseño (GOF<sup>7</sup>):

- ❖ Creacional: creación de objetos.
- ❖ Estructural: composición de clases y objetos.
- ❖ Comportamiento: interacción entre objetos.

### Descripción de los patrones de creación.

- ❖ **Factoría Abstracta:** proporciona una interfaz para crear familias de objetos relacionados o dependientes sin especificar su clase concreta.
- ❖ **Única instancia:** garantiza que solamente se crea una instancia de la clase y provee un punto de acceso global a él.

### Descripción de los patrones estructurales.

- ❖ **Fachada:** simplifica los accesos a un conjunto de objetos relacionados proporcionando un objeto de comunicación.

### Descripción de los patrones de comportamiento.

- ❖ **Cadena de responsabilidad:** evita el acoplamiento entre quien envía una petición y el receptor de la misma.
- ❖ **Plantilla:** define un esqueleto de algoritmo y delega partes concretas de un algoritmo a las subclases.

#### 4.3.1 Diagrama de clases del diseño

Los diagramas de clases son diagramas de estructura estática que muestran las clases del sistema y las relaciones entre ellas; muestran lo que el sistema puede hacer y cómo puede ser construido. Cuando se crea un diagrama de clases, se modela una parte de los elementos y relaciones que configuran la vista de diseño del sistema.

---

<sup>7</sup> Por sus siglas en inglés: Gand Of Four, lo cual quiere decir en español La Banda de los Cuatro.

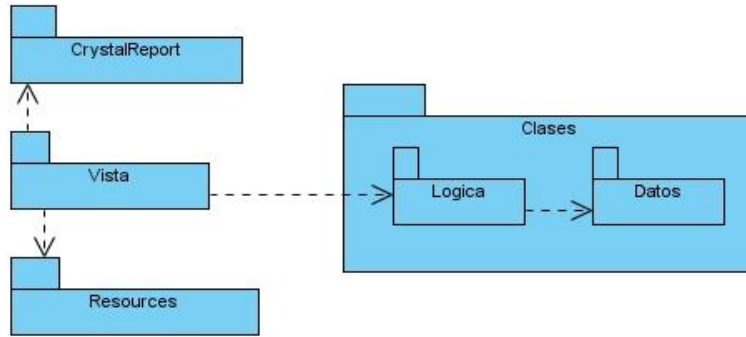


Figura 4 Diagrama de clases del diseño. Vista Global de paquetes del diseño.

A continuación se muestra el diagrama de clases del diseño para el caso de uso Generar Informe.

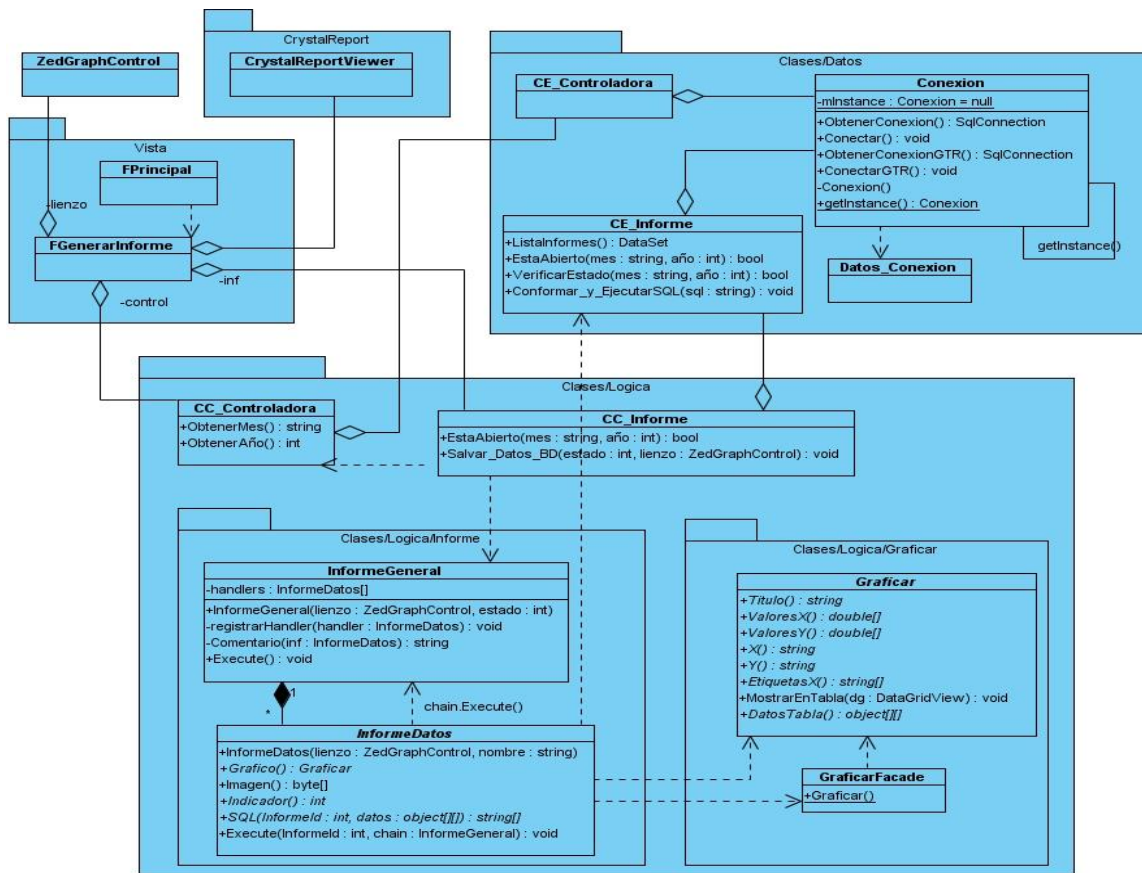


Figura 5 Diagrama de clases del diseño CU Generar Informe.

#### 4.4 Diseño de la base de datos

El primer paso para la construcción de una base de datos es definir su estructura, de forma tal que permita un adecuado mecanismo para almacenar los datos y posteriormente recuperarlos. Para lograr un buen diseño de la base de datos es necesario seguir un conjunto de pasos que comienzan con definir los diagramas correspondientes de Entidad-Relación y de Clases Persistentes.

## 4.4.1 Diagrama de clases persistentes

Todas las clases identificadas durante el desarrollo del diseño no tienen que ser necesariamente persistentes, la persistencia de una clase está dada por la capacidad de la misma para mantener su valor en el espacio y en el tiempo; contrario a las clases temporales que son utilizadas y almacenadas por la aplicación en tiempo de ejecución, dejando de existir al finalizar el programa.

En la siguiente figura se muestra el diagrama de clases persistentes correspondiente al sistema planteado.

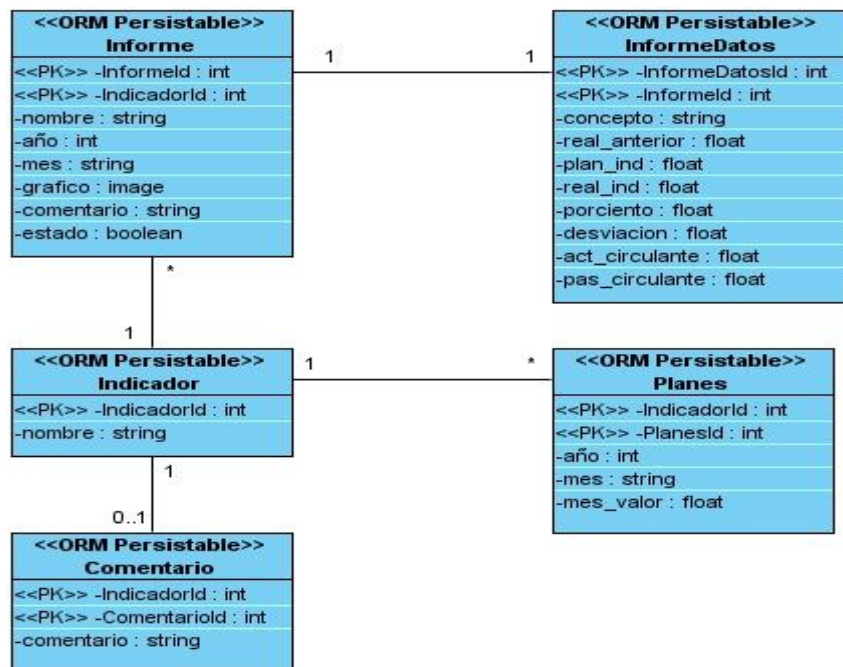


Figura 6 Diagrama de Clases Persistentes.

## 4.4.2 Diagrama Entidad – Relación

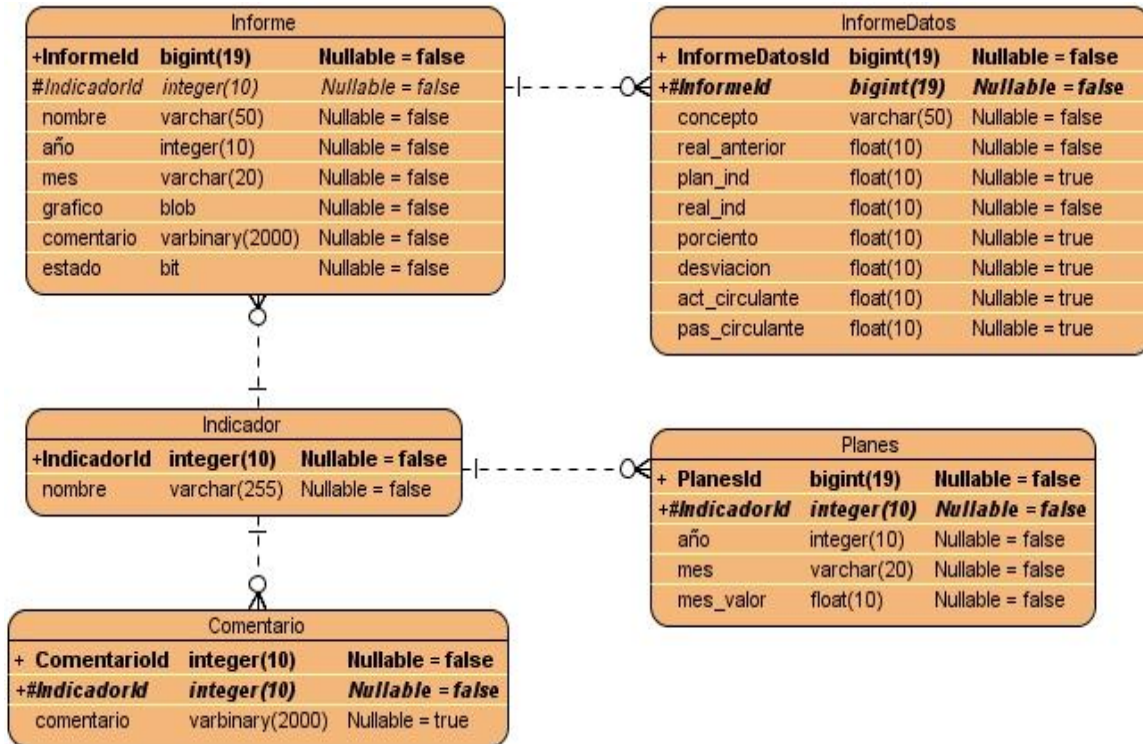


Figura 7 Diagrama Entidad-Relación.

## 4.5 Generalidades de la implementación

En la fase de construcción se implementa el sistema, se crea el código adecuado al resultado de la fase de diseño por lo que sigue los patrones y la arquitectura escogida.

El modelo de implementación permite planificar las integraciones de sistemas necesarias en cada iteración, distribuye el sistema al asignar componentes ejecutables a nodos en el diagrama de despliegue, implementa las clases y subsistemas encontrados durante el diseño y posibilita probar los componentes individualmente para después integrarlos.

### 4.5.1 Modelo de componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas pueden ser simples archivos, paquetes, bibliotecas cargadas dinámicamente.



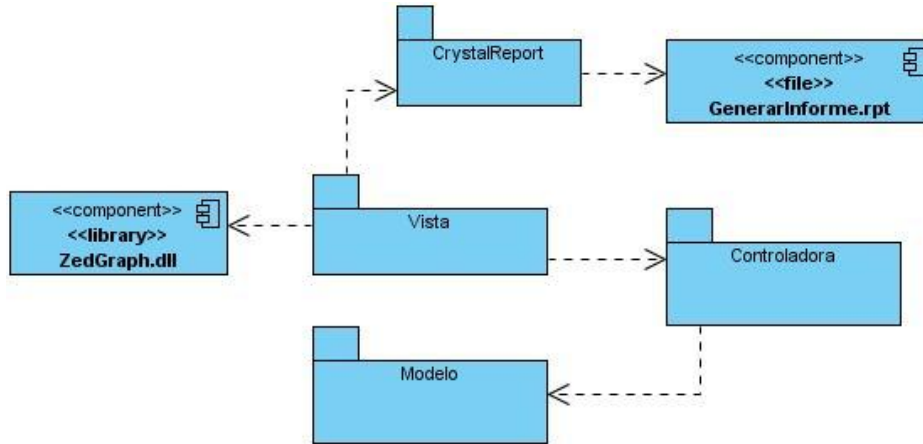


Figura 8 Diagrama de componentes.

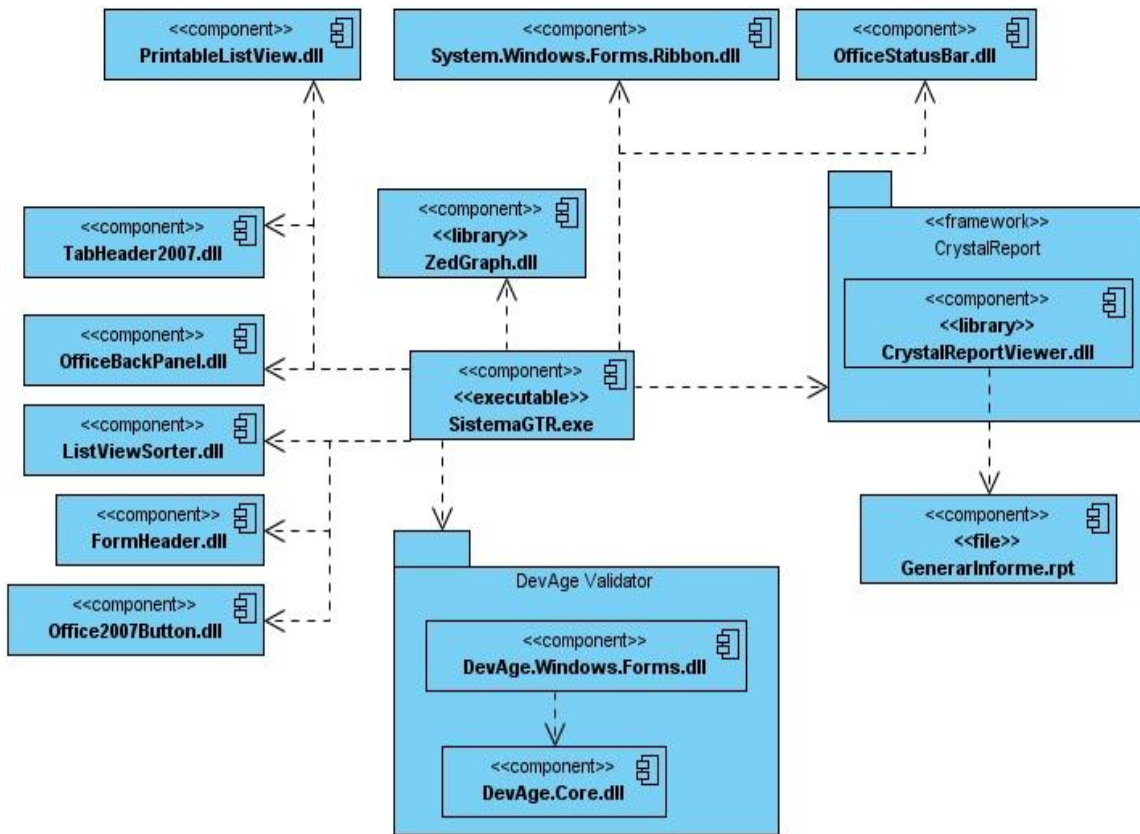


Figura 9 Diagrama de componentes compilados del Sistema.

### 4.5.2 Modelo de despliegue

Los Diagramas de Distribución muestran la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. Un nodo es

un elemento físico que existe en tiempo de ejecución y representa un recurso computacional que generalmente tiene algo de memoria y a menudo, capacidad de procesamiento. Los nodos se utilizan para modelar la topología del hardware sobre el que se ejecuta el sistema. Representa típicamente un procesador o un dispositivo sobre el que se pueden desplegar los componentes.

En la siguiente figura se muestra el modelo de despliegue para el sistema planteado.

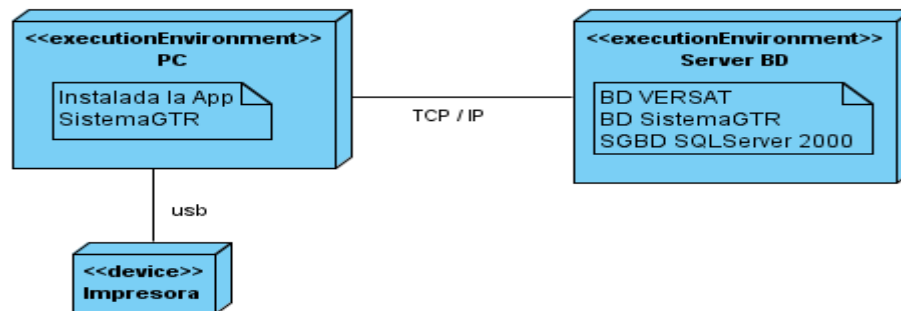


Figura 8 Modelo de Despliegue.

### 4.5.3 Descripción de los nodos

Dispositivo Impresora: este dispositivo satisface las necesidades de los clientes ante la posibilidad de imprimir reportes u otro tipo de documentos adquiridos a través de la aplicación. La conexión es a través del puerto USB.

Procesador PC Cliente: su función es la ejecución del sistema e interactuar con el mismo según sus necesidades, además de interactuar con los dispositivos de impresión.

Procesador Servidor de Base de Datos: es el encargado de almacenar toda la información generada del sistema. Estará recibiendo constantes peticiones de la PC cliente debido a que en el se encuentran alojada las BD VERSAT y SistemaGTR, con los que se comunicará mediante TCP/IP.

### 4.6 Prueba del sistema propuesto

El desarrollo del software implica una serie de actividades en las que la posibilidad de que aparezcan errores humanos es muy común.

La prueba de software es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación, estas tienen diferentes clasificaciones, algunas de ellas pueden ser:

- ❖ **Pruebas de Verificación:** se comprueba el cumplimiento de las especificaciones del diseño.
- ❖ **Pruebas de Validación:** se encargan de cuidar el cumplimiento de los requisitos del análisis.
- ❖ **Pruebas de Caja Blanca:** se conoce el código y se trata de probar la interacción interna entre los componentes del sistema.
- ❖ **Pruebas de Caja Negra:** solamente se conoce la interfaz y se trata de probar cada uno de los elementos que componen a la misma cumpliendo las precondiciones y pos condiciones para cada caso de uso.

Cualquier tipo de prueba tiene el objetivo principal de hallar errores o defectos en el software.

Para que las pruebas aplicadas a un software tengan éxito es necesario efectuar casos de pruebas con probabilidad de descubrir los errores en el sistema.

### 4.6.1 Pruebas de Caja Negra

En el caso de este sistema se propone la realización de una prueba de Caja Negra, la cual se refiere a las pruebas que se llevan a cabo sobre la interfaz del software, por lo que los casos de prueba intentan demostrar que las funciones del software son operativas, que la entrada se acepta de forma correcta y que la salida producida es adecuada.

Muchos autores consideran que estas pruebas permiten encontrar: (PRESSMAN, 2005)

- ❖ Funciones incorrectas o ausentes.
- ❖ Errores de interfaz.
- ❖ Errores en estructuras de datos o en accesos a las Bases de Datos externas.
- ❖ Errores de rendimiento.
- ❖ Errores de inicialización y terminación.

Para preparar los casos de pruebas se necesita un número de datos que ayuden a la ejecución de estos casos y que permitan que el sistema se ejecute en todas sus variantes.

Para desarrollar la prueba de caja negra existen varias técnicas, entre ellas están: (PRESSMAN, 2005)

- ❖ **Técnica de la Partición de Equivalencia:** divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del sistema.
- ❖ **Técnica del Análisis de Valores Límites:** prueba la habilidad del sistema para manejar datos que se encuentran en los límites aceptables.
- ❖ **Técnica de Grafos de Causa-Efecto:** permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

Dentro del método de Caja Negra la técnica de la Partición de Equivalencia es una de las más efectivas porque permite examinar los valores válidos e inválidos de las entradas existentes en el sistema, descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico.

La partición equivalente se dirige a la definición de casos de pruebas que descubran clases de errores, reduce así en número de clases de prueba que hay que desarrollar.

El diseño de casos de prueba para la partición equivalente se basa en una evaluación de las clases de equivalencia para una condición de entrada. Una clase de equivalencia representa un conjunto de estados válidos o inválidos para condiciones de entrada. Regularmente, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica. (PRESSMAN, 2005)

Las clases de equivalencia se pueden definir de acuerdo con las siguientes directrices:

- ❖ Si un parámetro de entrada debe estar comprendido en un cierto rango, aparecen 3 clases de equivalencia: por debajo, en y por encima del rango.

- ❖ Si una entrada requiere un valor concreto, aparecen 3 clases de equivalencia: por debajo, en y por encima del rango.
- ❖ Si una entrada requiere un valor presente en un conjunto de valores, aparecen 2 clases de equivalencia: en el conjunto o fuera de él.
- ❖ Si una entrada es booleana, hay 2 clases: si o no.

Al aplicar estas directrices se ejecutan casos de pruebas para cada elemento de datos del campo de entrada a desarrollar, donde los casos de prueba se seleccionan de forma que ejerciten el mayor número de atributos de cada clase de equivalencia a la vez.

A continuación se muestra los casos de pruebas para el caso de uso “Configurar Sistema” y “Gestionar planes” donde se utiliza la técnica de Partición de Equivalencia.

## 4.6.2 Caso de uso: Configurar Sistema

### Descripción general del caso de uso

El caso de uso comienza cuando el usuario necesita introducir los datos necesarios para configurar el sistema. El sistema realiza la opción seleccionada por el usuario y termina el caso de uso.

### Secciones a probar en el caso de uso

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
<b>SC 1:</b> “Configurar Sistema”.	<b>EC 1.1:</b> “Configurar Sistema exitosamente”.	El sistema muestra la pantalla Configurar Sistema con los campos requeridos para su configuración. El usuario introduce los datos solicitados por el sistema que son Servidor, Base de datos, Usuario y el Password, luego el usuario da clic en el botón “Conectar”.	1. Configurar Sistema. 2. Clic botón “Conectar”.
	<b>EC 1.2:</b> “Configurar Sistema	El usuario introduce mal los datos para configurar el sistema, dejando	1. Configurar Sistema.

	incorrectamente”.	campos vacíos e introduce punto y coma en los campos de textos. El sistema muestra un icono rojo al lado del campo incorrecto con un mensaje de error.	2. Clic botón “Conectar”.
	<b>EC 1.3:</b> “Cancelar la Configuración del Sistema”.	Al seleccionar el botón “Cancelar”, se debe cerrar la ventana “Configurar Sistema” y se cancela la operación.	1. Configurar Sistema. 2. Clic botón “Cancelar”.

**Tabla 6 Secciones a probar en el caso de uso Configurar Sistema.**

**Descripción de las variables**

No.	Nombre del campo	Clasificación	Valor Nulo	Descripción
1	Servidor	Campo de Texto	No	Servidor al cual se conecta el sistema.
2	Base de datos	Campo de Texto	No	Base de datos donde se encuentra instalada.
3	Usuario	Campo de Texto	No	Indica el usuario que va acceder al sistema.
4	Password	Campo de Texto	No	Indica el password del usuario que va acceder al sistema.

**Tabla 7 Descripción de las variables para el caso de uso Configurar Sistema.**

**Matriz de datos (SC 1. Configurar Sistema)**

Id del escenario	Escenario	Servidor	Base de Datos	Usuario	Password	Respuesta del sistema	Resultado de la prueba

EC 1.1	“Configurar Sistema exitosamente”.	V"localhost"	V"VERSAT"	V"sa"	V"sa"	Es configurado el sistema por el usuario.	
EC 1.2	“Configurar Sistema incorrectamente”.	V"localhost"	V"VERSAT"	V"aloma"	I"China"	El sistema muestra un mensaje de error.	
		V"localhost"	V"VERSAT"	I"Niño"	V"sa"	El sistema muestra un mensaje de error.	
		I"localhost ;"	V"VERSAT"	V"sa"	V"sa"	El sistema muestra un icono rojo al lado del campo incorrecto con un mensaje de error.	
		V"localhost"	I"VERSAT123"	V"bolo"	V"bolo123"	El sistema muestra un mensaje de error.	
EC 1.3	“Cancelar la Configuración del Sistema”.	N/A	N/A	N/A	N/A	El sistema cierra la pantalla Configurar Sistema.	

Tabla 8 Matriz de Datos. SC1. Configurar Sistema.

### 4.6.3 Caso de uso: Gestionar planes

#### Descripción general del caso de uso

El caso de uso comienza cuando el usuario necesita gestionar los planes para confeccionar los gráficos de los indicadores. El caso de uso termina cuando el usuario completa el registro de los planes.

**Condiciones de Ejecución**

El usuario debe estar autenticado.

**Secciones a probar en el caso de uso**

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
SC 1: "Adicionar Plan".	EC 1.1: "Seleccionar Indicador".	El usuario selecciona la opción "Adicionar Plan". El sistema muestra el mensaje "Seleccione un indicador".	1. Menú Principal. 2. Gestionar Plan. 3. Clic botón "Adicionar Plan".
	EC 1.2: "Adicionar Plan correctamente".	El usuario selecciona la opción "Adicionar Plan". El sistema muestra la pantalla Plan por meses para Adicionar los planes, un formulario con los datos necesarios que el usuario debe introducir. El usuario introduce los datos necesarios y da clic en el botón "Aceptar". El sistema muestra el mensaje. "Datos de planes guardados correctamente" y la ventana Plan por meses se cierra.	1. Menú Principal. 2. Gestionar Plan. 3. Clic botón "Adicionar Plan". 4. Plan por meses. 5. Clic botón "Aceptar".
	EC 1.4: "Adicionar Plan incorrectamente".	El usuario introduce mal los datos necesarios El sistema muestra el mensaje "Algunos planes se encuentra vacios".	1. Menú Principal. 2. Gestionar Plan. 3. Clic botón "Adicionar Plan". 4. Plan por meses. 5. Clic botón "Aceptar".



	EC 1.5: “Cerrar Plan por meses”.	El usuario selecciona la opción “Cerrar” para detener la operación de Adicionar planes. El sistema cierra la pantalla Plan por meses.	<ol style="list-style-type: none"> <li>1. Menú Principal.</li> <li>2. Gestionar Plan.</li> <li>3. Clic botón “Adicionar Plan”.</li> <li>4. Plan por meses.</li> <li>5. Clic botón “Cerrar”.</li> </ol>
SC 2: “Editar Plan”.	EC 2.1: “Seleccionar Indicador”.	El usuario selecciona la opción Editar Plan. El sistema muestra el mensaje “Seleccione un indicador”.	<ol style="list-style-type: none"> <li>1. Menú Principal.</li> <li>2. Gestionar Plan.</li> <li>3. Clic botón Editar Plan.</li> </ol>
	EC 2.2: “Editar Plan correctamente”.	El usuario selecciona la opción “Editar Plan”. El sistema muestra la pantalla Plan por meses para Editar los planes, un formulario con los datos necesarios. El usuario introduce los datos necesarios para editarlos y da clic en el botón “Editar”. El sistema muestra el mensaje. “Datos de planes editados correctamente” y la ventana Plan por meses se cierra.	<ol style="list-style-type: none"> <li>1. Menú Principal.</li> <li>2. Gestionar Plan.</li> <li>3. Clic botón “Editar Plan”.</li> <li>4. Plan por meses.</li> <li>5. Clic botón “Editar”.</li> </ol>
	EC 2.3: “Editar Plan incorrectamente”.	El usuario introduce mal los datos. El sistema muestra un ícono rojo al lado del campo necesario con un mensaje del error.	<ol style="list-style-type: none"> <li>1. Menú Principal.</li> <li>2. Gestionar Plan.</li> <li>3. Clic botón “Editar Plan”.</li> <li>4. Plan por meses.</li> <li>5. Clic botón “Editar”.</li> </ol>

	EC 2.4: “Cerrar Plan por meses”.	El usuario selecciona la opción “Cerrar” para detener la operación de Editar planes. El sistema cierra la pantalla Plan por meses.	<ol style="list-style-type: none"> <li>1. Menú Principal.</li> <li>2. Gestionar Plan.</li> <li>3. Clic botón “Editar Plan”.</li> <li>4. Plan por meses.</li> <li>5. Clic botón “Cerrar”.</li> </ol>
--	----------------------------------	--	---

**Tabla 9 Secciones a probar en el caso de uso Gestionar planes.**

**Descripción de las variables**

No.	Nombre del campo	Clasificación	Valor Nulo	Descripción
1	Indicador	ComboBox	No	Indicador al cual se le va adicionar o editar los planes.
2	TablaIndicadores	PrintableListView	Si	Muestra los planes de los indicadores.
3	Meses	Campo de Texto	No	Representa el plan de un mes.

**Tabla 10 Descripción de las variables para el caso de uso Gestionar planes.**

**Matriz de datos (SC 1. “Adicionar Plan”)**

ID del escenario	Escenario	Indicador	TablaIndicadores	meses	Respuesta del sistema	Resultado de la prueba
EC 1.1	Seleccionar Indicador.	I	N/A	N/A	El sistema muestra el mensaje “Seleccione un	

					indicador”.	
EC 1.2	“Adicionar Plan correctamente”.	V	N/A	V	El sistema muestra el mensaje “Datos de planes guardados correctamente” y la ventana Plan por meses se cierra.	
EC 1.3	“Adicionar Plan incorrectamente”.	V	N/A	I	El sistema muestra el mensaje “Algunos planes se encuentra vacios”.	
		I	N/A	N/A	El sistema muestra el mensaje “Seleccione un indicador”.	
EC 1.4	“Cerrar Plan por meses”.	N/A	N/A	N/A	El sistema cierra la pantalla Plan por meses.	

Tabla 11 Matriz de Datos. SC1. “Adicionar Plan”.

**Matriz de datos (SC 2. “Editar Plan”)**

ID del escenario	Escenario	Indicador	TablaIndicadores	plan	Respuesta del sistema	Resultado de la prueba
EC 2.1	Seleccionar Indicador.	I	N/A	N/A	El sistema muestra el mensaje “Seleccione un indicador”.	
EC 2.2	“Editar Plan correctamente”.	V	N/A	V	El sistema muestra el mensaje. “Datos de planes editados	

					correctamente” y la ventana Plan por meses se cierra.	
EC 2.3	“Editar Plan incorrectamente”.	V	N/A	I	El sistema muestra el mensaje “Algunos planes se encuentra vacios”.	
		I	N/A	N/A	El sistema muestra el mensaje “Seleccione un indicador”.	
EC 2.4	“Cerrar Plan por meses”.	N/A	N/A	N/A	El sistema cierra la pantalla Plan por meses.	

**Tabla 12 Matriz de Datos. SC2. “Editar Plan”.**

#### **4.7 Conclusiones**

En el presente capítulo se arribó a un sistema completamente diseñado y construido en términos de clases del diseño. Se generaron además cada uno de los artefactos y diagramas referentes al flujo de trabajo de implementación y se culmina con la modelación completa de la solución propuesta.

## **CONCLUSIONES**

Una vez concluida la investigación y luego de obtenidos los resultados, es posible resaltar las conclusiones que se enumeran a continuación:

1. Las tecnologías, herramientas, lenguajes y framework seleccionados para el desarrollo de la aplicación que centra la investigación, obedecen a criterios de selección de tecnologías libres y multiplataforma y en el caso de la selección de tecnologías con pagos de licencias, la empresa se encarga de pagar las mismas.
2. La puesta en explotación del sistema desarrollado facilita mayor rendimiento a la Empresa Comercializadora y de Servicios de Productos Universales de Las Tunas y el procesamiento de los datos resulta más seguro debido a la reducción de errores por la actividad humana, por otra parte, agiliza el tiempo de entrega y la manipulación de los datos obtenidos en la dirección contable financiera.
3. El diseño y la arquitectura de información de la aplicación se presentan de manera sencilla, fácil de usar, intuitiva y con tonalidad de colores que facilitan un entorno agradable a los trabajadores de la empresa.
4. El diseño de las pruebas de caja negra permite en un momento posterior validar los requisitos funcionales capturados y comprobar que el sistema realiza las acciones en el momento y el lugar en el que las debe realizar.

## **RECOMENDACIONES**

El autor de la presente investigación recomienda:

1. Establecer un período de capacitación para todos los usuarios nuevos que deban utilizar el sistema para el desarrollo de sus actividades.
2. Continuar la identificación de requisitos funcionales en aras de ampliar las principales funcionalidades de la herramienta desarrollada.
3. Generalizar el uso del sistema desarrollado en el resto de las empresas universales pertenecientes al Ministerio de Comercio Interior del país.

### REFERENCIAS BIBLIOGRÁFICAS

1. **Baldiviezo, Por: Lic. Adm. Janneth Mónica Thompson. 2008.**  
<http://www.promonegocios.net/contabilidad/concepto-contabilidad.html>. [En línea] 2008.
2. **(ONU), Organización de las Naciones Unidas.** *Integrated and coordinated implementation and follow-up of major. United Nations conferences and summits.* Nueva York, Estados Unidos de América : s.n.
3. **STAIR, Ralph M. 2003.** *Principles of Information Systems, Sixth Edition.* Thomson Learning, Inc., pp.132. 2003. ISBN/0-619-06489-7.
4. **Desarrollo Web.** <http://desarrolloweb.com/wiki/aplicacion-de-escritorio.html>. [En línea]
5. **CORNEJO, J. E. G. 2001.** *Arquitectura en Capas. Un camino hacia los procesos distribuidos.* 2001.
6. **FRANK BUSCHMANN, D.S., MICHAEL STAL Y HANS ROHNERT. 2000.** *PATTERN-ORIENTED SOFTWARE ARCHITECTURE: PATTERNS FOR CONCURRENT AND NETWORKED OBJECTS.* 2000.
7. **RUP-Ágil, La naturaleza serial en.** <http://cgi.una.ac.cr/AUP/html/overview.html>.
8. **RUMBAUHG, J. I. JACOBSON. 2000.** *El Lenguaje Unificado de Modelado. Manual de Referencia.* Madrid : Pearson Educación, 2000.
9. **G. Booch, J. Rumbaugh, I. Jacobson. 1999.** *El Lenguaje Unificado de Modelado.* . s.l. : Addison Wesley Iberoamericana, 1999.
10. **CrystalReport, Soluciones.**  
<http://www.crystalsolutions.com.ar/productos/crystalreports.html>.
11. **Larman, Craig. 2004.** *UML Y PATRONES Introducción al análisis y diseño orientados a objeto.* La Habana : Felix Varela, 2004.
12. **SQL-Server-2000.** <http://www.formaselect.com/curso/experto-en-sql-server-2000/presentacion.htm>.
13. **Ojeda, Francisco Charle. 2001.** *SQL SERVER 2000.* 2001.
14. **SECO, J.A.G. 2001.** *El Lenguaje de Programación C#.* 2001.
15. **Gamma. 1995.** *Design Patterns: Elements of Reusable Object-Oriented Software.* s.l. : Addison Wesley, 1995.
16. **PRESSMAN. 2005.** *R. S. INGENIERIA DEL SOFTWARE. UN ENFOQUE PRACTICO.* 2005.

## **BIBLIOGRAFÍA**

1. Ayaviri García Daniel. CONTABILIDAD BÁSICA Y DOCUMENTOS MERCANTILES. Edición 1º; Editorial "N-DAG"; Argentina. Pág 10-11.
2. Sydney, Davidson & Roman, L. Weil. BIBLIOTECA MCGRAW-HILL DE CONTABILIDAD. Tomo I; Edición 3º; Editorial MCGRAW-HILL; Pág. 1-3.
3. Terán Gandarillas Gonzalo J TEMAS DE CONTABILIDAD BASICA E INTERMEDIA.4ta Edición; Editorial Educación y Cultura; Pág 4.
4. Baldiviezo, Por: Lic. Adm. Janneth Mónica Thompson. 2008.  
<http://www.promonegocios.net/contabilidad/concepto-contabilidad.html>. [En línea] 2008.
5. Ricardo Romero Las Técnicas Modernas de Venta y sus Aplicaciones, de Allan L. Reid, Editorial Diana, Pág.54.
6. Marketing, Editora Palmir E.I.R.L., de, Pág. 35.
7. Real Academia Española: Disponible en <http://www.rae.es/>. Sección: Diccionario de la Lengua Española.
8. STAIR, Ralph M. 2003. *Principles of Information Systems, Sixth Edition*. Thomson Learning, Inc., pp.132. 2003. ISBN/0-619-06489-7.
9. [http://www.freedownloadmanager.org/es/downloads/Paradigma\\_Visual\\_para\\_UML\\_%5Bcuenta\\_de\\_Plataforma\\_de\\_Java\\_14715\\_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%5Bcuenta_de_Plataforma_de_Java_14715_p/)
10. SECO, J.A.G. 2001. *El Lenguaje de Programación C#*. 2001.
11. CLEMENTS, M. S. Y. P. A FIELD GUIDE TO BOXOLOGY: PRELIMINARY CLASSIFICATION OF ARCHITECTURAL STYLES FOR SOFTWARE SYSTEMS, 1997.
12. FILEHEAVEN. VISUAL PARADIGM FOR UML (COMMUNITY EDITION) FOR WINDOWS 3.1, 2008. [2008]. Disponible en: [ww.fileheaven.com/descargar/visual-paradigm-for-uml-communityedition-for-windows/32083.htm](http://ww.fileheaven.com/descargar/visual-paradigm-for-uml-communityedition-for-windows/32083.htm)
13. FRANK BUSCHMANN, D. S., MICHAEL STAL Y HANS ROHNERT. PATTERN-ORIENTED SOFTWARE ARCHITECTURE: PATTERNS FOR CONCURRENT AND NETWORKED OBJECTS. 2000. P.
14. GARLAN, D. SOFTWARE ARCHITECTURE: A ROADMAP. 2000. P.
15. HASTUPROGRAMA. TIPOS DE APLICACIONES, 2008. [2008]. DISPONIBLE EN:[www.haztuprograma.com/tiposAplicaciones.html](http://www.haztuprograma.com/tiposAplicaciones.html)
16. RALLO, E. H. EL LENGUAJE UNIFICADO DE MODELADO (UML). 2006. [2008]. DISPONIBLE EN: [www.disca.upv.es/enheror/pdf/ActaUML.PDF](http://www.disca.upv.es/enheror/pdf/ActaUML.PDF)



17. G. Booch, J. Rumbaugh, I. Jacobson. 1999. *El Lenguaje Unificado de Modelado*. .  
s.l. : Addison Wesley Iberoamericana, 1999.
18. PRESSMAN. 2005. R. S. *INGENIERIA DEL SOFTWARE. UN ENFOQUE  
PRACTICO*. 2005.