

**ESCUELA POLITECNICA DEL EJÉRCITO**

**SEDE LATACUNGA**



**FACULTAD DE INGENIERIA DE SISTEMAS E INFORMATICA**

**“DISEÑO DE UN SISTEMA CONTABLE PARAMETRIZABLE FINANCIERO ON  
LINE”**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TITULO DE INGENIERO EN  
SISTEMAS E INFORMATICA**

**AUTORES:**

**QUISPE QUISPE LUIS ANIBAL**

**CHICAIZA MOLINA JENNY ALEXANDRA**

**Latacunga-Ecuador**

**2007**

## **AGRADECIMIENTO**

Gracias a Dios por darme sabiduría y vida para culminar con éxito mi carrera, un infinito y merecido agradecimiento a mis padres que con amor, sacrificio, abnegación y consejos me han apoyado durante todo el transcurso de mi carrera, quienes han compartido los buenos y difíciles momentos de mi vida, dándome la fuerza y valor para seguir adelante.

A la ESPE sede Latacunga, ilustre politécnica, forjadora de profesionales líderes y propulsores del cambio; a los distinguidos Docentes por compartir día a día sus conocimientos contribuyendo en mi formación tanto profesional como humana y brindado su amistad y apoyo incondicional; a mis queridos compañeros con quienes compartí los más bellos y inolvidables momentos de vida.

Jenny

## **AGRADECIMIENTO**

Deseo expresar mis más sinceras muestras de agradecimiento:

A mi querida Esposa, mi ayuda idónea, por su amor, paciencia, comprensión y motivación, sin lo que hubiese sido imposible lograr terminar estos estudios. A mis Padres, Hermano y Hermanas por creer y confiar siempre en mí, apoyándome en

todas las decisiones que he tomado en la vida. A mis maestros por sus consejos y por compartir desinteresadamente sus amplios conocimientos y experiencia. Por ultimo agradezco a todos aquellos que también me dieron su apoyo incondicional en momentos que realmente necesitaba, de verdad gracias.

Luis

### **DEDICATORIA**

Este trabajo símbolo de una meta alcanzada, lo dedico a mis padres, y hermanos que son las personas más importantes en mi vida, quienes me dan la fuerza y valor para seguir adelante.

Jenny

### **DEDICATORIA**

A Dios por haberme iluminado todo este tiempo de esfuerzo, por darme la oportunidad de conocer a alguien que ha sido como un pilar muy importante para mi, y sobre todo por los padres tan maravillosos que me dio.

A mis padres por el respeto, responsabilidad, dedicación y tenacidad que me inculcaron para que siempre lograra mis objetivos, por haber sido mis amigos, pues me enseñaron distinguir entre el bien y el mal, entre la sinceridad y el juego, me

Luis

# CAPITULO I

## **I. INGENIERIA DE SOFTWARE Y METODOLOGÍAS DE DESARROLLO DE SISTEMAS DE INFORMACION (WEB)**

En las dos últimas décadas las notaciones de modelado y posteriormente las herramientas de desarrollo de software pretendieron ser las "balas de plata" para su éxito, sin embargo, las expectativas no fueron satisfechas. Esto se debe en gran parte a que otro importante elemento, la metodología de desarrollo, había sido postergado. De nada sirven buenas notaciones y herramientas si no se proveen directivas para su aplicación.

Así, esta década ha comenzado con un creciente interés en metodologías de desarrollo de software. Hasta hace poco, se utilizaba el esquema "tradicional" para el desarrollo de software, el mismo que ha demostrado ser efectivo y necesario en proyectos de gran tamaño (respecto a tiempo y recursos)

Ante esta situación, las metodologías ágiles aparecen, están orientadas para proyectos pequeños, constituyen una solución a medida, con una elevada simplificación, pero a pesar de ello no renuncia a las prácticas esenciales de asegurar la calidad del producto.

## **1.1 EL SOFTWARE**

### **1.1.1 DEFINICIÓN DE SOFTWARE**

“Software es la expresión de un conjunto de instrucciones mediante palabras, códigos, planes o en cualquier otra forma que al ser incorporados en un dispositivo de lectura automatizada, es capaz de hacer que un computador ejecute una tarea u obtenga un resultado. La protección del programa de ordenador comprende también la documentación técnica y los manuales de uso.”<sup>1</sup>

“Software, es el conjunto de ítems u objetos que forman una configuración que incluyen programas, documentos y datos.”<sup>2</sup>

"Software, son los programas, procedimientos, reglas y documentación posible, asociada con la computación, así como los datos pertenecientes a la operación de un sistema de computo." <sup>3</sup>

Para el autor, el software es un sistema informático compuesto por un conjunto de instrucciones que, cuándo se ejecutan produce resultados de acuerdo con los objetivos y función principal predeterminada, permitiendo utilizar y explotar las enormes capacidades de procesamiento y almacenamiento del hardware. Dicho conjunto de instrucciones está organizado en estructuras de datos que permiten la manipulación de la información.

---

<sup>1</sup> Concepto de software según el D. Legislativo 882.- (Ley sobre derechos de autor)

<sup>2</sup> Pressman, Roger S; Ingeniería del Software; Un enfoque práctico; Tercera Edición; 1992

<sup>3</sup> Glosario Estándar de Términos de Ingeniería del Software de IEEE, 1983

### **1.1.2 IMPORTANCIA DE SOFTWARE**

Hoy en día, el software forma parte en la vida cotidiana e influye sobre muchas de las actividades que se realiza, están incluidas en innumerables aplicaciones de software como editores, compiladores, sistemas operativos, entornos gráficos, programas de telecomunicaciones, procesadores de texto, hojas de cálculo, bases de datos, aplicaciones gráficas, inteligencia artificial, redes neuronales, robótica, juegos, sistemas electrónicos, entre otros; por ello el software se considera un motor económico de creciente importancia

### **1.1.3 EVOLUCION DE SOFTWARE**

Los sistemas informáticos llevan cinco décadas (desde los años 50), por lo tanto, esa es la edad del software (ya que una computadora sin software no hace nada).

Sin embargo, la importancia del software frente al hardware ha cambiado durante estos años:

- **1950 – 1965**

Son los años del desarrollo de las computadoras. Se investigaba principalmente en el hardware. Al software se le restaba importancia.

- Orientación por lotes (tarjeta resultado). No interactivo
- Distribución limitada
- Software a medida
- El software se desarrollaba sin ninguna planificación
- No existía documentación.

- **1965 – 1975**

Los sistemas informáticos comienzan a captar la atención de más gente. Por lo tanto, existen mayores posibilidades de que el software llegue a más usuarios.

- El Software cobra más importancia
- Aparecen casas desarrolladoras de software
- Sistemas distribuidos
- Sistemas en tiempo real
- Bases de datos
- Librerías de software

Pero la manera del desarrollo del software no cambio. Aparece el concepto de “Mantenimiento del Software”, que representaba grandes gastos.

- **1975- 1985**

Desarrollo de todos los dispositivos de hardware actuales: Sistemas distribuidos, redes, herramientas y entornos de desarrollo, entre otros.

- El hardware se estandariza, pero el software no.
- Sistemas distribuidos
- Incorporación de Inteligencia Artificial
- Hardware de bajo coste
- Gran consumo de software
- La gente gasta más dinero en software que en hardware para la empresa

- **1985 – actualidad**

La mayoría de la gente puede acceder a hardware más potente. Los avances se basan en nuevos conceptos de software.

- Tecnología orientada a objetos
- Sistemas expertos
- Redes neuronales
- Algoritmos genéticos
- Programación paralela

- **Futuro**

En el horizonte se divisan nubarrones:

- El arte visual será el resultado de la colaboración entre humanos y software inteligente.
- Traductores de lenguaje incluido en los teléfonos.
- Los chats serán ambientes totalmente virtuales y muy reales.
- Transacciones en Línea

#### **1.1.4 CARACTERISTICAS DE SOFTWARE**

- **El software se desarrolla, no se fabrica en un sentido clásico.-** Existen similitudes entre el proceso de desarrollo del software y la construcción del hardware. En ambos casos existen fases de análisis, diseño y desarrollo o construcción, y la buena calidad del producto final se obtiene mediante un buen diseño. Sin embargo, en la fase de producción del software pueden producirse problemas que afecten a la calidad.



- **El software no se estropea.-** El Software no es susceptible a los males del entorno pero se deteriora; el software al principio de su vida presenta muchas fallas debido a defectos de diseño o baja calidad de la fase de producción, el mismo que requerirá cambios debido a fallas; cada cambio producido al software va degradando poco a poco el funcionamiento del sistema.
- **La mayoría del software se construye a medida.-** La mayoría de software se fabrica a medida, siendo la reutilización muy baja, esto hace que el impacto de costes de software sea muy elevado.

### **1.1.5 CRISIS DE SOFTWARE**

El origen del término “Crisis del Software” se atribuye a dos conferencias organizadas por la OTAN en 1967 y 1968, donde muchos observadores determinaron que existían excesivos problemas en el desarrollo del software, que es todavía un problema no resuelto.

En la industria del software hemos tenido una “crisis” que ha estado con nosotros cerca de 30 años.

La Crisis de Software tiene las siguientes fases:

#### **Primera Fase.-** Apogeo de la crisis (1965 - 1970)

- Desarrollo inacabable de grandes programas
- Ineficiencia en el manejo de los recursos
- Errores de funcionamiento
- Costos impredecibles
- Nada es posible

### **Segunda Fase.-** Innovación conceptual (1970 - 1980)

- Fundamentos de programación
- Verificación de programas
- Metodologías de diseño

### **Tercera Fase.-** El diseño es el problema (1980 - actualidad)

- Entornos de programación
- Especificación formal
- Programación automática

La crisis en la industria del software ha permanecido durante muchos años, lo cual parece una contradicción para el término. Lo que si se podría decir es que hay un problema crónico en el desarrollo de software.

#### **1.1.6 MITOS DE SOFTWARE**

- **El costo de computadores es menos que el de dispositivos analógicos o electromecánicos.-** El hardware es barato; pero el costo de escribir y certificar software muy confiable y seguro, más el costo de la mantención sin poner en peligro la confiabilidad y la seguridad, puede ser enorme.
- **Es fácil cambiar el software.-** Los cambios son fáciles, pero hacer cambios sin introducir errores es muy difícil. Hay que verificar el software de nuevo con cada cambio. También, con cambios el software se pone frágil.

- **Los computadores proveen más confiabilidad que los dispositivos que reemplazan.-** El software no falla como dispositivos normales, pero hay poca evidencia que indica el comportamiento erróneo de software. No existen técnicas como la redundancia para aumentar sencillamente la confiabilidad del software. Y aun cuando sea posible escribir software sin errores, las condiciones ideales para desarrollar software (dinero y tiempo sin límites) nunca existen.
  
- **Mayor confiabilidad de software aumenta la seguridad.-** Se puede mejorar la confiabilidad de software eliminando errores; la mayoría de los errores de software crítico a la seguridad son debidos a errores en los requerimientos.
  
- **La prueba o verificación formal del software puede eliminar todos los errores.-** Las limitaciones de la prueba de software son bien conocidas. Básicamente hay demasiados estados en software real para probarlo completamente, para la verificación se debe chequear la consistencia entre las especificaciones y la implementación.
  
- **Los computadores disminuyen el riesgo por contraste con los sistemas mecánicos**
  - Los computadores permiten un control más fino ya que pueden revisar parámetros más frecuentemente, hacer cálculos en tiempo real, y tomar acción rápidamente.
  - Los sistemas automatizados permiten a los operadores trabajar más lejos de áreas peligrosas.
  - Con la eliminación de operadores se eliminan los errores humanos.
  - Los computadores pueden proveer mejor información a los operadores.

## **1.2 INGENIERIA DE SOFTWARE**

### **1.2.1 DEFINICIÓN DE INGENIERIA DE SOFTWARE**

A través de los años han surgido diversas definiciones de Ingeniería de Software, entre las más importantes se pueden destacar las siguientes:

“Ingeniería de Software, trata del establecimiento de los principios y métodos de ingeniería a fin de obtener software de modo rentable que sea fiable y trabaje en máquinas reales”.<sup>4</sup>

“Ingeniería del Software, es la aplicación práctica del conocimiento científico en el diseño y construcción de programas de computadora y la documentación asociada requerida para desarrollar, operar (funcionar) y mantenerlos. Se conoce también como desarrollo de software o producción de software”.<sup>5</sup>

“Ingeniería de Software, es el estudio de los principios y metodologías para desarrollo y mantenimiento de sistemas de software”.<sup>6</sup>

“Ingeniería de Software, es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación (funcionamiento) y mantenimiento del software”<sup>7</sup>

“Ingeniería de Software, es una disciplina que comprende todos los aspectos de la producción de software desde las etapas iniciales de especificación del sistema, hasta el mantenimiento de éste después de que se utiliza”.<sup>8</sup>

---

<sup>4</sup> Bauer, F.L; Software Engineering; Information Processing, 71, North Holland Publishing Co; Ámsterdam; 1972

<sup>5</sup> Bohem, B.W; Software Engineering, IEEE; Transactions on Computers,1076; Pág 1226 - 1241

<sup>6</sup> Zelkovits, M.V., Shaw, A.C. y Gannon, J.D; Principles of Software Engineering and Design; Prentice-Hall; Englewoods CLIF; 1979.

<sup>7</sup> IEEE, Standars Collection; Software Engineering; IEEE Standard 610.12-1990; IEEE; 1993

<sup>8</sup> Sommerville, 2002

Para el autor, la Ingeniería de Software es la aplicación de un enfoque sistemático, disciplinado y cuantificable hacia el desarrollo, operación y mantenimiento de sistemas software de calidad que resuelven problemas de tipo general y específico.

### **1.2.2 ELEMENTOS DE INGENIERIA DE SOFTWARE**

La ingeniería del software abarca un conjunto de tres elementos clave: métodos, herramientas y procedimientos, que facilitan el proceso de desarrollo y suministran a los implementadores bases para construir de forma productiva software de alta calidad.

- 1) Métodos o técnicas:** Indican cómo construir técnicamente el software, comprende una serie de tareas que incluyen la planificación y estimación de proyectos, el análisis de requisitos, el análisis y la especificación del sistema, el diseño de estructuras de datos, programas y procedimientos, la codificación, la documentación, las pruebas y el mantenimiento. Los métodos introducen frecuentemente una notación específica para la tarea en cuestión y una serie de criterios de calidad.
- 2) Herramientas:** Son instrumentos o sistemas que producen resultados exactos, eficientes, productivos, que refuerza la calidad del producto resultante. Proporcionan un soporte automático o semiautomático para todas las fases del desarrollo.
- 3) Procedimientos:** Son la combinación de técnicas y herramientas que en forma conjunta dan un resultado particular. Los procedimientos indicarán qué herramientas deberán utilizarse cuando se aplican determinadas técnicas. Definen la secuencia como se aplican los métodos, los controles que aseguran la calidad, las directrices y los documentos que permitan a gestores evaluar los progresos.

### **1.2.3 OBJETIVOS DE INGENIERIA DE SOFTWARE**

En la construcción y desarrollo de proyectos se aplican métodos y técnicas para resolver los problemas, la informática aporta procedimientos sobre los que se apoya la ingeniería de software, para producir software de calidad.

- Aumentar la productividad y trabajo de los ingenieros del software.
- Facilitar el control del proceso de desarrollo de software.
- Suministrar a los desarrolladores las bases para construir software de alta calidad en forma eficiente.
- Definir la disciplina que garantice la producción y el mantenimiento de productos de software desarrollados en el plazo fijado y dentro del costo estimado.

### **1.2.4 PROCESO DE DESARROLLO SOFTWARE**

El proceso de desarrollo de software contiene tres fases genéricas: definición, desarrollo, y mantenimiento, se encuentran en todo el proceso independientemente del área de aplicación, del tamaño del proyecto o de la complejidad.

#### **1.2.4.1 Fase de Definición del Software**

La fase de definición se centra sobre el qué, identifica que información debe ser proporcionada, qué función y rendimiento se desea, qué interfaces deben establecerse, qué restricciones de diseño existen y qué criterios de validación se necesitan para definir un sistema correcto. Aunque los métodos aplicados durante la fase de definición variarán dependiendo del paradigma de ingeniería del software aplicado, de alguna forma se producirán tres pasos específicos:

- **Análisis del sistema.** Define el papel de cada elemento del sistema informático, asignando finalmente al software el papel que va a desempeñar.
- **Planificación del proyecto de software.** Una vez establecido el ámbito del software, se analizan los riesgos, se asignan los recursos, se estiman los costos, se definen las tareas y se planifica el trabajo.
- **Análisis de Requerimientos.** El ámbito establecido para el software proporciona la dirección a seguir, pero antes de comenzar a trabajar es necesario disponer de una información más detallada del ámbito de información y de la función del software.

#### 1.2.4.2 Fase de Desarrollo del Software

La fase de desarrollo se centra en el cómo, se determina, cómo han de diseñarse las estructuras de datos y la arquitectura del software, cómo han de implementarse los detalles procedimentales, cómo ha de traducirse el diseño a un lenguaje de programación y cómo ha de realizarse la prueba. Los métodos aplicados durante la fase de desarrollo varían, pero de alguna forma se aplicarán tres pasos concretos.

- **Diseño de software.** El diseño traduce los requisitos de software a un conjunto de representaciones (algunas gráficas y otras tabulares o basadas en lenguajes) que describen las estructuras de bases de datos, la arquitectura, el procedimiento algorítmico y las características de la interfaz.
- **Codificación.** Las representaciones del diseño deberán ser traducidas a un lenguaje artificial (un lenguaje de programación convencional o un lenguaje no procedimental), dando como resultado las instrucciones ejecutables en la computadora.

- **Prueba del software.** Una vez que el software ha sido implementado en una forma ejecutable por la maquina, debe ser probado para descubrir los defectos que puedan existir, en la función, en la lógica y en la implementación.

#### 1.2.4.3 Fase de Mantenimiento del Software

La fase de mantenimiento se centra en el cambio que va asociado a la corrección de errores, a las adaptaciones requeridas por la evolución del entorno del software y a las modificaciones debidas a los cambios de requisitos del usuario dirigidos a reforzar o ampliar el sistema. La fase de mantenimiento vuelve a aplicar las fases de definición y de desarrollo, pero en el contexto del software ya existente. Durante la fase de desarrollo se encuentran tres tipos de cambio :

- **Corrección.** Incluso llevando a cabo las mejores actividades de garantía de calidad, es muy probable que el cliente descubra defectos en el software. El mantenimiento correctivo cambia el software para corregir los defectos.
- **Adaptación.** Con el paso del tiempo es probable que cambie el entorno original (sistemas operativos, equipos periféricos, etc.) para los que se desarrollo el software. El mantenimiento adaptivo consiste en modificar el software para acomodarlo a los cambios de su entorno externo.
- **Mejora.** Conforme utilice el software, el usuario puede descubrir funciones adicionales que podrían interesar que estuvieran incorporadas en el software. El mantenimiento perfectivo amplía el software mas allá de sus requisitos funcionales originales.



## **1.2.5 CALIDAD DE SOFTWARE**

### **1.2.5.1 Definición de Calidad de Software**

"La Calidad de Software, es la concordancia con los requisitos funcionales y de rendimiento, establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente".<sup>9</sup>

Para el autor, la calidad de software pretende que los productos desarrollados satisfagan las necesidades de los usuarios, para ser competitivos en el entorno, alcanzando la ausencia de errores de funcionamiento, la adecuación a las necesidades del usuario, y el alcance del desempeño apropiado (tiempo, volumen, espacio), además, el cumplimiento de estándares con el único fin de producir software de gran calidad.

### **1.2.5.2 La calidad en Ingeniería de Software**

El software es un producto con características muy especiales, hay que tener en cuenta que es un producto que se desarrolla y se centra en su diseño, con una existencia lógica de instrucciones sobre un soporte, siendo un producto que no se gasta con el uso y repararlo no significa restaurarlo al estado original, sino corregir algún defecto de origen.

"Software, son los programas de ordenador, los procedimientos, posiblemente la documentación asociada y los datos relativos a la operación del sistema informático, no limitándose al código."<sup>10</sup>

"Calidad, el grado con el que un sistema componente o proceso cumple con los requisitos especificados y las necesidades o expectativas del cliente o usuario".<sup>11</sup>

---

<sup>9</sup> Pressman, Roger S; Ingeniería del Software; Un enfoque práctico; Tercera Edición; 1992

<sup>10</sup> IEEE estándar de ingeniería del software (IEEE, 1990)

<sup>11</sup> IEEE 6.10 -1990 (IEEE, 1990)

La aplicación de estándares de desarrollo y de normas para el software permitirá lograr calidad técnica del mismo. La calidad del software se puede ver a nivel empresa como implantación de un sistema de calidad y a nivel de proyecto aplicando las técnicas de evaluación y control de la calidad del software a lo largo del ciclo de vida.

La calidad de Ingeniería de Software son los requisitos explícitos, ya sean funcionales, de seguridad, de rendimiento, de interfaces, son la culminación de la etapa de análisis y quedan establecidos en el documento de especificación de requisitos del software y es en la etapa de análisis donde muchos de los requisitos implícitos no expresados formalmente por el usuario quedarán declarados en el documento de especificación.

### **1.2.5.3 Aseguramiento de Calidad de Software**

- El aseguramiento de calidad de software consiste en los procedimientos, técnicas y herramientas aplicadas por profesionales para asegurar que un producto alcance o exceda estándares preespecificados durante el ciclo de vida de desarrollo de software.
- El aseguramiento de calidad de software es el conjunto de actividades planificadas y sistemáticas, necesarias para aportar la confianza en que el producto (software) satisfaga los requisitos dados de calidad.
- El aseguramiento de calidad del software se diseña en cada aplicación antes de comenzar a desarrollarla y no después de ser desarrollada.
- Algunos autores prefieren decir **garantía** de calidad en vez de aseguramiento.
  - Garantía, puede confundir con garantía de productos
  - Aseguramiento pretende dar confianza en que el producto tiene calidad

- El aseguramiento de calidad del software está presente en
  - Métodos y herramientas de análisis, diseño, programación y prueba
  - Inspecciones técnicas formales en todos los pasos del proceso de desarrollo del software
  - Estrategias de prueba multiescala
  - Control de la documentación del software y de los cambios realizados
  - Procedimientos para ajustarse a los estándares
  - Mecanismos de medida (métricas)
  - Registro de auditorias y realización de informes
  
- Actividades para el aseguramiento- de calidad del software
  - Métricas de software para el control del proyecto
  - Verificación y validación del software a lo largo del ciclo de vida
  - Incluye las pruebas y los procesos de revisión e inspección
    - La gestión de la configuración del software

#### **1.2.5.4 Gestión de la Calidad de Software**

- Gestión de la calidad (ISO 9000)
  - Conjunto de actividades que determina la calidad, los objetivos y las responsabilidades y se implanta con la planificación de la calidad, el control de la calidad, el aseguramiento (garantía) de la calidad y la mejora de la calidad, en el marco del sistema de calidad, la gestión de la calidad se aplica normalmente a nivel de empresa ò también puede haber una gestión de calidad dentro de la gestión de cada proyecto.
  
- Política de calidad (ISO 9000)
  - Directrices y objetivos generales de una organización, relativos a la calidad, tal como se expresan formalmente por la alta dirección

### 1.2.5.5 Control de la Calidad de Software

- El Control de la calidad de Software son las técnicas y actividades de carácter operativo, utilizadas para satisfacer los requisitos relativos a la calidad, centradas en dos objetivos fundamentales:
  - mantener bajo control un proceso
  - eliminar las causas de los defectos en las diferentes fases del ciclo de vida
- En general son las actividades para evaluar la calidad de los productos desarrollados

### 1.2.5.6 Factores que determinan la Calidad de Software

Los factores que determinan la Calidad de Software se clasifican en tres grupos:

- **Operaciones del producto:** Son las características operativas.
  - **Corrección** (¿Hace lo que se le pide?).- El grado que una aplicación satisface sus especificaciones y consigue los objetivos encomendados por el cliente
  - **Fiabilidad** (¿Lo hace de forma fiable todo el tiempo?).- El grado que se espera de una aplicación lleve a cabo las operaciones especificadas y con la precisión requerida
  - **Eficiencia** (¿Qué recursos hardware y software necesito?).- La cantidad de recursos hardware y software que necesita una aplicación para realizar las operaciones con los tiempos de respuesta adecuados

- **Integridad** (¿Puedo controlar su uso?).- El grado que puede controlarse el acceso al software o a los datos a personal no autorizado
- **Facilidad de uso** (¿Es fácil y cómodo de manejar?).- El esfuerzo requerido para aprender el manejo de una aplicación, trabajar con ella, introducir datos y conseguir resultados
- **Revisión del producto:** Es la capacidad para soportar cambios
  - **Facilidad de mantenimiento** (¿Puedo localizar los fallos?).- El esfuerzo requerido para localizar y reparar errores
  - **Flexibilidad** (¿Puedo añadir nuevas opciones?).- El esfuerzo requerido para modificar una aplicación en funcionamiento
  - **Facilidad de prueba** (¿Puedo probar todas las opciones?).-El esfuerzo requerido para probar una aplicación de forma que cumpla con lo especificado en los requisitos
- **Transición del producto:** Es la adaptabilidad a nuevos entornos
  - **Portabilidad** (¿Podré usarlo en otra máquina?).-El esfuerzo requerido para transferir la aplicación a otro hardware o sistema operativo
  - **Reusabilidad** (¿Podré utilizar alguna parte del software en otra aplicación?).- Grado en que partes de una aplicación pueden utilizarse en otras aplicaciones
  - **Interoperabilidad** (¿Podrá comunicarse con otras aplicaciones o sistemas informáticos?).-El esfuerzo necesario para comunicar la aplicación con otras aplicaciones o sistemas informáticos

## **1.3 PARADIGMAS DE DESARROLLO DE SOFTWARE**

### **1.3.1 DEFINICIÓN DE PARADIGMAS DE DESARROLLO DE SOFTWARE**

Los paradigmas de la ingeniería del software, se describen como métodos alternativos para la ingeniería del software. En muchos casos, los paradigmas pueden y deben combinarse de forma que se puedan utilizar las ventajas de cada uno a un único proyecto. Representan un enfoque particular o filosofía para la construcción del software. No es mejor uno que otro sino que cada uno tiene ventajas y desventajas.

Entre los paradigmas más utilizados son: paradigma estructurado, orientado a objetos y orientado a la web.

### **1.3.2 PARADIGMA ESTRUCTURADO**

El paradigma estructurado proporciona alineamientos para desarrollar Sistemas de Información. Es un conjunto de pasos que si bien son secuenciales no necesariamente deben llevarse con rigidez en cualquier momento q se requiere puede devolverse a la fase anterior de hecho es muy común que si en una fase se requiera modificaciones alguna de una fase previa o hasta repetir varias veces una misma tarea para comparar.

El objetivo que persigue es estructurar u organizar las tareas asociadas con la determinación de requerimientos para obtener la comprensión completa y exacta de una situación dada.

Se concentra en especificar lo que se requiere que haga el sistema o la aplicación. No se establece como cumplirán los requerimientos o la forma en que implantarán la aplicación. Más bien permite que las personas observen los elementos lógicos separados de los componentes físicos.

### 1.3.2.1 Especificaciones Estructuradas

Existen varias cualidades que debe poseer la especificación estructurada:

- **Gráfica.-** Formadas por una colección de diagramas, DFD's, deben representar una visión bien sustanciosa de lo que está siendo especificado, una concepción fácil de entender sobre el problema bajo estudio.
- **Particionadas.-** Los procesos incluidos en los DFD'S, son los elementos básicos en los que se descompone el sistema. Este particionamiento será implementado de una forma top-down de forma tal que haya una progresión gradual desde los aspectos más abstractos a los más detallados.
- **Mínimamente redundantes.** De forma que los cambios en los requisitos necesita reflejarse en un sólo punto de la especificación. El proceso de cambio de la especificación estructurada debe ser fácil de ejecutar.

### 1.3.2.2 Análisis Estructurado

Siguiendo las técnicas del análisis estructurado, que consiste en un conjunto integrado de: Modelo de Procesos y Modelo de Datos

#### **Modelo de Procesos**

Es la parte más conocida y extensa de la especificación estructurada. Está formada por un conjunto de Diagramas de Flujo de datos, diccionario de datos y mini especificaciones.

- **Diagrama de Flujo de Datos**

El diagrama de flujo de datos (DFD), es una herramienta que permite visualizar un sistema como una red de procesos funcionales, conectados entre sí por "conductos" y "tanques de almacenamiento" de datos.

Los componentes de un diagrama típico de flujo de datos:

- Proceso
- Flujo
- Almacén de datos
- Terminador

○ **Proceso.**

Es el primer componente del DFD, los sinónimos comunes son burbuja, función, transformación. El proceso muestra una parte del sistema que transforma entradas en salidas, se representa gráficamente como un círculo, un óvalo o un rectángulo con esquinas redondeadas; se nombra o describe con una sola palabra, frase u oración sencilla. Un buen nombre para un proceso generalmente consiste en una frase verbo-objeto tal como **validar entradas o calcular impuesto**, o puede contener el nombre de una persona o un grupo.

○ **Flujos de Datos**

Un flujo se representa gráficamente por medio de una flecha que entra o sale de un proceso, describe el movimiento de bloques o paquetes de información de una parte del sistema a otra, permitiendo conectar los procesos.

○ **Almacén de Datos**

El almacén se utiliza para modelar una colección de paquetes de datos en reposo. Se denota por dos líneas paralelas, el nombre que se utiliza es el plural de los flujos. Los almacenes se conectan por flujos a los procesos. Así, el contexto en el que se muestra en un DFD es:

- Un flujo desde un almacén.
- Un flujo hacía un almacén.



- **Terminador.**

El terminador gráficamente se representa como un rectángulo, estos representan entidades externas con las cuales el sistema se comunica. Comúnmente, puede ser una persona, o un grupo, que esté dentro de la misma compañía u organización, pero fuera del control del sistema que se está modelando. Un terminador puede ser otro sistema, como algún otro sistema computacional con el cual se comunica éste.

- **Diccionario de Datos**

Un análisis del dominio de la información puede ser incompleto si solo se considera el flujo de datos. Cada flecha de un diagrama de flujo de datos representa uno o más elementos de información. El diccionario de datos contiene las definiciones de todos los datos mencionados en el DFD.

- **Miniespecificaciones**

Las miniespecificaciones describen las reglas sobre cómo realizar el proceso para transformar las entradas en salidas; indican el proceso a realizar, la transformación de datos, no el algoritmo que se selecciona en la etapa de diseño.

Las herramientas son:

- Lenguaje estructurado
- Tablas de decisión
- Árboles de Decisión
- Pre y Post-condiciones

## **Modelo de Datos**

El modelado de datos se centra únicamente en los datos que maneja el proceso y en qué relaciones se establecen entre estos datos, dejando de lado las funciones que realiza el sistema y el control del mismo

- **Diagrama Entidad- Relación**

El diagrama de entidad-relación (también conocido como DER, o diagrama E-R) es un modelo de red que describe con un alto nivel de abstracción la distribución de datos almacenados en un sistema, enfatiza las relaciones entre almacenes de datos en el DFD.

Hay cuatro componentes principales en un diagrama de entidad-relación:

- Tipos de objetos.
- Relaciones.
- Indicadores asociativos de tipo de objeto.
- Indicadores de supertipo/subtipo.

- **Tipos de objetos**

El tipo de objeto se representa en un diagrama de entidad-relación por medio de una caja rectangular; representa una colección de objetos (cosas) del mundo real cuyos miembros individuales o instancias tienen las siguientes características:

- Cada una puede identificarse de manera única por algún medio
- Cada uno juega un papel necesario en el sistema que se construye. Es decir, para que el tipo de objeto sea legítimo, debe poder decirse que el sistema no puede operar sin tener acceso a esos miembros.
- Cada uno puede describirse por uno o más datos. Es decir, un **cliente** puede describirse por medio de datos tales como nombre, domicilio, límite de crédito y número telefónico.

- **Relaciones**

Una **relación** es una asociación o relación matemática entre varias entidades. Las relaciones también se nombran. Se representan en el diagrama E-R mediante flechas y rombos. Cada entidad interviene en una relación con una determinada **cardinalidad**. La cardinalidad (número de instancias o elementos de una entidad que pueden asociarse a un elemento de la otra entidad

relacionada) se representa mediante una pareja de datos, en minúsculas, de la forma (*cardinalidad mínima, cardinalidad máxima*), asociada a cada uno de las entidades que intervienen en la relación.

El **tipo de relación** se define tomando los máximos de las cardinalidades que intervienen en la relación. Hay cuatro tipos posibles:

- Una a una (1:1). En este tipo de relación, una vez fijado un elemento de una entidad se conoce la otra. Ejemplo: nación y capital.
- Una a muchas (1:N). Ejemplo: cliente y pedidos.
- Muchas a una (N:1). Simetría respecto al tipo anterior según el punto de visto de una u otra entidad.
- Muchas a muchas (N:N). Ejemplo: personas y viviendas.

Toda entidad debe ser unívocamente identificada y distinguible mediante un conjunto de atributos denominado **identificador** o **clave principal** o **primaria**.

Puede haber varios posibles identificadores para una misma entidad, en cuyo caso se ha de escoger uno de ellos como identificador principal siendo el resto identificadores alternativos.

- **Indicadores asociativos de tipo de objeto**

El indicador asociativo de tipo de objeto representa algo que funciona como objeto y como relación. Esto pretende indicar que funciona como:

- Un tipo de objeto, algo acerca de lo cual se desea almacenar información.
- Una relación que conecta los dos tipos de objetos cliente y artículo.

- **Indicadores de subtipo/supertipo**

Los tipos de objetos de subtipo/supertipo consisten en tipos de objeto de una o más subcategorías, conectados por una relación. El supertipo se conecta con una línea que contiene una barra.

### **1.3.3 PARADIGMA ORIENTADO A OBJETOS**

Paradigma Orientado a Objeto(O.O): disciplina de ingeniería de desarrollo y modelado de software que permite construir más fácilmente sistemas complejos a partir de componentes individuales. Objetos + Mensajes = Programa.

Su historia va unida a la evolución de los lenguajes de programación orientada a objeto, los más representativos: a fines de los 60's SIMULA, a fines de los 70's Smalltalk-80, la primera versión de C++ por Bjarne Stroustrup en 1981 y actualmente Java o C# de Microsoft. A fines de los 80's comenzaron a consolidarse algunos métodos Orientadas a Objeto.

En 1995 Booch y Rumbaugh proponen el Método Unificado con la ambiciosa idea de conseguir una unificación de sus métodos y notaciones, que posteriormente se reorienta a un objetivo más modesto, para dar lugar al Unified Modeling Language (UML), la notación OO más popular en la actualidad.

Algunos métodos OO con notaciones predecesoras de UML son: OOAD (Booch), OOSE (Jacobson), Coad & Yourdon, Shaler & Mellor y OMT (Rumbaugh).

Algunas metodologías orientadas a objetos que utilizan la notación UML son: Rational Unified Process (RUP), OPEN, MÉTRICA.

### 1.3.3.1 Especificaciones Orientadas a Objetos

El paradigma Orientado a Objetos esta formado por los siguientes conceptos:

- **Objetos**

Un objeto encapsula datos, operaciones, otros objetos, constantes y otra información relacionada. Los datos dentro de un objeto son atributos. Un objeto sólo puede ser manipulado a través de su interfaz. Interfaz es el conjunto de métodos visibles desde el exterior. Pueden ser: Entidades Externas, Ocurrencias o eventos, papeles o roles, unidades organizacionales, lugares, estructuras.

Existen cinco características que deben ser usadas para incluir o un objeto en el modelo de análisis: información retenida, servicios necesarios, atributos múltiples, atributos comunes, requisitos esenciales.

- **Las Clases**

Una clase encapsula las abstracciones de datos y procedimientos que se requieren para describir el contenido y comportamiento de alguna entidad del mundo real. Por definición, todos los objetos que existen dentro de una clase heredan sus atributos y las operaciones disponibles para la manipulación de los atributos.

Una clase es un tipo de dato que contiene uno o más elementos de datos llamados miembro dato, y cero, una o más funciones que manipulan esos datos (llamados miembro función o funciones miembros). Una clase se puede definir con una estructura (estruct), una unión (union) o una clase class.

### 1.3.3.2 Propiedades Orientadas a Objetos

- **Abstracción**

La abstracción es la propiedad que permite representar las características esenciales de un objeto, sin preocuparse de las restantes características (no esenciales). Una abstracción se centra en la vista externa de un objeto, de modo que sirva para separar el comportamiento esencial de un objeto de su implementación.

El elemento clave de la programación orientada a objetos es la clase. Una clase se puede definir como una descripción abstracta de un grupo de objetos, cada uno de los cuales se diferencia por su estado específico y por la posibilidad de realizar una serie de operaciones.

- **Encapsulamiento**

El Encapsulamiento o encapsulación asegura que el contenido de la información de un objeto está oculta al mundo exterior. La encapsulación (también se conoce como ocultación de la información), en esencia, es el proceso de ocultar todos los secretos de un objeto que no contribuyen a sus características esenciales, también le permite controlar la forma en que se utilizan los datos y los procedimientos.

La encapsulación permite la división de un programa en módulos. Estos módulos se implementan mediante clases, de forma que una clase representa la encapsulación de una abstracción. En la práctica, esto significa que cada clase debe tener dos partes: una interfaz y una implementación. La interfaz de una clase captura sólo su vista externa y la implementación contiene la representación de la abstracción, así como los mecanismos que realizan el comportamiento adecuado.

- **Modularidad**

La Modularidad es la propiedad que permite subdividir una aplicación en partes más pequeñas (llamadas módulos), que se puedan compilar por separado, cada una de las cuales debe ser tan independiente como sea posible de la aplicación en sí y de las restantes partes.

Para dividir una aplicación en módulos, se debe tener en cuenta los conceptos asociados de dependencia, acoplamiento, cohesión, interfaz, encapsulación y abstracción. Una vez identificado lo que es un buen módulo, se puede contemplar la reutilización de un módulo como componente.

- **Jerarquía**

La Jerarquía permite la ordenación de las abstracciones. Las dos jerarquías más importantes de un sistema complejo son: estructura de clases (jerarquía “es-un” (is-a): generalización / especialización) y estructura de objetos (jerarquía “parte-de” (part-of): agregación).

Las jerarquías de generalización / especialización se conocen como herencia. Básicamente, la herencia define una relación entre clases, en donde una clase comparte la estructura o comportamiento definido en una o más clases (herencia simple y herencia múltiple, respectivamente).

- **Polimorfismo**

Es la propiedad que indica, la posibilidad que una entidad tome muchas formas. En términos prácticos, el polimorfismo permite referirse a objetos de clases diferentes mediante el mismo elemento de programa y realizar la misma operación de diferentes formas, según el objeto que se refiera en ese momento.

### 1.3.4 PARADIGMA ORIENTADO A LA WEB

La World Wide Web e Internet son los avances más importantes en la historia de la informática; los sistemas y aplicaciones basados en Web hacen posible disponer de una gran variedad de contenido y funcionalidad. A medida que las Web se integran cada vez más en grandes y pequeñas compañías, surge la necesidad de construir sistemas fiables y adaptables.

#### 1.3.4.1 Atributos de Aplicaciones basadas en la Web

- **Intensivas de Red.-** Por naturaleza una Web reside en una red dando servicio a las necesidades del cliente.
- **Controlada por el contenido.-** Utiliza hipermedia para presentar al usuario el contenido de textos, gráficos, sonido y vídeo.
- **Evolución Continua.-** Están en constante evolución, inusualmente se pueden actualizar cada hora.

#### 1.3.4.2 Características de la Web

- **Inmediatez.-** Las aplicaciones basadas en la Web tienen una inmediatez que no se encuentra en otros tipos de software. El tiempo que se tarda en comercializar un sitio Web completo puede ser cuestión de días o semanas.
- **Seguridad.-** Se debe implementar fuertes medidas de seguridad en toda la infraestructura Web y dentro de la misma aplicación, porque se encuentran disponibles a través de la red.
- **Estética.-** El atractivo diseñado de la Web su apariencia e interacción.



#### 1.3.4.3 Aplicaciones de la Web

- **Informativa.-** Información solo de lectura con navegación y enlaces simple.
- **Descarga.-** Descarga de información desde servidores apropiados.
- **Personalizable.-** Personaliza el contenido a necesidades específicas.
- **Interacción.-** Comunicación entre usuarios, mediante el chat, mensajería instantánea, entre otros.
- **Entrada del usuario.-** Es el mecanismo de la necesidad de comunicación, por la entrada basada en formularios.
- **Orientada a transacciones.-** Solicitudes que se realiza.
- **Orientada a servicios.-** Proporciona ayuda al usuario.
- **Portal.-** Canaliza al usuario llevándole a otros contenidos fuera del dominio de la aplicación.
- **Acceso a base de datos.-** Consulta de base de datos y extracción de información.

#### 1.3.4.4 Tecnologías de la Web

El diseño y la implementación de sistemas basados en la Web incorporan tres tecnologías: el desarrollo basado en componentes, la seguridad y los estándares de Internet.

- **Desarrollo basado en Componentes.-** Las tecnologías han evolucionado en gran parte gracias al crecimiento explosivo de los sistemas y aplicaciones basados en Web. Se dispone de tres estándares importantes para la infraestructura: CORBA, COM/DCOM Y Java Beans, los mismos proporcionan una infraestructura que permite a los que diseñan emplear y personalizar componentes de terceras partes, para así comunicarse unos con otros y con servicios a nivel de sistemas.

- **Seguridad.-** Mediante la infraestructura de red se proporciona una variedad de medidas de seguridad, tales como encriptación, cortafuegos, entre otras.
- **Estándares de Internet.-** El estándar dominante en la creación de contenido y la estructura de la Web ha sido HTML, un lenguaje de marcas que posibilita al desarrollador proporcionar una serie de etiquetas que describen una variedad de objetos de datos. Sin embargo, a medida que las aplicaciones crecen en tamaño y complejidad, se adoptado un nuevo estándar XML, lenguaje de marcas extensible es un subconjunto estrictamente definido del metalenguaje SGML, permitiendo que los diseñadores definan etiquetas personalizadas en las descripciones de página Web.

#### 1.3.4.5 Proceso de la Web

- **Formulación.-** Identifica las metas y objetivos y establece el ámbito del primer incremento.
- **Planificación.-** Estima el coste global del proyecto, evalúa riesgos asociados con el esfuerzo del desarrollo.
- **Análisis.-** Establece los requisitos técnicos e identifica los elementos del contenido que se van a incorporar.
- **Ingeniería.-** Consta del diseño de contenido y la producción, que tienen como objetivo diseñar, producir, y adquirir todo el contenido de texto gráfico y video.
- **Generación de páginas.-** Usa herramientas automatizadas para la creación de la Web, se lleva a cabo la integración con los diseños arquitectónicos, de navegación, de la interfaz y otros lenguajes orientados a procesos.
- **Pruebas.-** Ejercitan la navegación, intentan descubrir los errores de las applets, guiones y formularios, asegurando que funcionará correctamente en diferentes entornos.
- **Evaluación del cliente.-** Solicitud de cambios por parte del cliente

#### 1.3.4.6 Diseño para Aplicaciones en la Web

En el desarrollo de sitios WEB podemos identificar dos niveles de diseño, que están relacionados al nivel de abstracción, necesario para modelar las estructuras y los componentes. El primer nivel de diseño llamado enfoque sistemático analiza la estructura. El segundo nivel de diseño llamado enfoque de detalle, caracterizado por el análisis en detalle de sus elementos y como pueden integrarse para construir la página Web.

##### ◆ Enfoque Sistemático

El enfoque sistemático, construye modelos de las estructuras que soportan la navegación del sitio Web. Los primeros modelos estaban ligados al paradigma estructurado, sin embargo tienen la limitación de que son metodologías para el desarrollo de software multimedia y que requieren de pequeñas adaptaciones para llegar a ser metodologías de diseño de sitios Web.

La mayoría de metodologías de diseños de sitios Web están ligadas al paradigma de orientación a objetos, el cual captura la estructura del sitio Web y el comportamiento de los elementos de dicha estructura.

Los modelos más utilizados en el diseño de herramientas multimedia son:

- **Modelo de diseño de hipertexto HDM (Hypertext Design Model).**- Introduce la terminología de la multimedia y los conceptos de entidad, tipos de entidades y estructura de acceso, permite presentar la información de varias formas.

- **Modelo de referencia Dexter.**- Es un modelo genérico propuesto que divide al sistema en tres capas o niveles: La capa de almacenamiento, donde se escribe la estructura de un conjunto finito de componentes.
- **Modelo Ámsterdam.**- Cubre las fallas del modelo Dexter, es decir, toma en consideración el tiempo y cumple con el criterio de sincronización indispensable en los sistemas multimedia.
- **Modelo de Diseño Hipermedia Orientado a Objeto OOHDM(The Object-Oriented Hipermedia Design Model).**- Es el sucesor del modelo de diseño de hipertexto HDM, se trata de una metodología que se fundamenta en la orientación a objeto. Tiene las fases: Diseño conceptual o análisis de dominio que utiliza el método de análisis orientado a objeto para obtener esquemas conceptuales de las clases y de las relaciones entre las mismas. Utiliza las “técnicas de modelado de objeto” llamado notación OMT para el diseño de navegación, donde se define la estructura de navegación por medio de modelos, es decir, a través de diferentes vistas del esquema conceptual; la fase de diseño de interfaz abstracta, se apoya en un modelo orientado a objeto para especificar la estructura y el comportamiento de la interfaz del sistema, se crea a través de tres tipos de diagramas: diagramas abstractos para cada clase, diagramas de configuración para reflejar los eventos externos y el diagrama de estado para señalar el comportamiento dinámico; y por último la fase de implementación, la construcción de programas en programación orientada a objetos.
- **Modelo DRMM (Dynamic Relation ship Management Model),** Esta basado en el modelo RMM sólo considera las aplicaciones dinámicas, permitiendo el diseño de pantallas con resultados de consultas realizadas por el usuario a una base de datos

Los modelos más utilizados en el diseño de sitios Web son:

- **Método del diseño de sitios Web centrados en el usuario (WSDM.- A User-centered design meted for Web sities).**- Este método toma muchos de los conceptos previos de diseño de documentos hypermedia, y adiciona un concepto clave para los sitios Web, la audiencia, definiéndola como la clase de los usuarios que determina los requerimientos del sitio Web.
- **Técnicas de Diseño de Hipermedia Estructurado (Structured Hypermedia Design Technique SHDT).**- Es una técnica de modelado y un ambiente de diseño basado en la computadora, útil para facilitar la creación de prototipos, considera las páginas estáticas y dinámicas.
- **Metodología de enfoque de ingeniería para el desarrollo de cursos basados en la Web (A software Engineering Approach to the Development of Web-based Courses).**- Basada en la ingeniería de software que incluye todas las fases para el desarrollo de cursos basados en la Web. Describe la estructura típica de una guía de estudio para la Web, se basa en la Orientación a Objeto y presenta cuatro fases: análisis de dominio de curso, para establecer el modelo conceptual de las clases que componen el sistema; la fase de definición de requerimientos, requerimientos de aprendizaje, de interacción, de la interfaz y de calidad; la fase de diseño de la guía de estudio, donde se diseña la estructura de la guía, de las unidades, las páginas Web, y por último, la fase de producción donde se realiza el desarrollo, se ensamblan las unidades y se verifican los requerimientos.

## ◆ **Enfoque de Detalle**

Un enfoque detallado en el diseño de sitios Web esta más ligado a las características propias de sus elementos, y en cómo éstos pueden integrarse para la construcción del mensaje.

- **Herramientas de autor.-** Presentan un ambiente integrado para la creación de páginas Web y ofrecen plantillas de diseño, sin embargo, no dan ninguna orientación sobre cómo componer, y equilibrar los pesos de los elementos que la integran, y más aun, si se sigue algunas plantillas de diseño se puede llegar a serias contradicciones de los lineamientos de usabilidad
- **Marcos (frames).-** Es una herramienta de detalle para crear páginas Web, su ventaja consiste en que las páginas de contenido se desarrollan independiente de las de navegación, el resultado final, un estruendo fracaso, viola muchos principios de usabilidad y lamentablemente por su éxito inicial existen muchos sitios que utilizan los marcos y rompen con los “criterios tradicionales” de la WEB.
- **Hojas De Estilo en Cascada (CCS).-** Son herramientas que permiten administrar los estilos de presentación de un sitio Web, tanto el diseñador como al usuario final, su uso garantiza la consistencia, sin embargo, no todos los navegadores lo soportan. Antes de la introducción de las hojas de estilo, los creadores de paginas Web sólo tenían un control parcial sobre el aspecto final de sus páginas.

- **Lenguaje Scripting.**- son lenguajes de programación donde se hacen pequeñas rutinas para ser ejecutadas de lado del cliente y así dar más poder de dinamismo e interactividad a las páginas Web. Permiten empotrar subprogramas en los documentos Web utilizando un lenguaje de programación o permitir cargar y ejecutar applets basados en Java, independientemente de la plataforma utilizada para la comunicación en Internet. Durante su ejecución puede generar contenido dinámico, interactuar con el usuario, validar datos de formularios, o incluso, crear ventanas y ejecutar aplicaciones independientes de las páginas. Estos programas sobrepasan el modelo HTML original. Los applets son una desviación del modelo básico de la Web; permiten extender las características de los navegadores de los clientes sin obligar a los usuarios a cambiar de navegador. La ventaja de los applets es que a través de ellos se puede proporcionar interfaces de usuario más ricas a la Web.

**FIGURA 1.1 DIFERENCIA ENTRE PARADIGMAS DE DESARROLLO SOFTWARE**

	PARADIGMA ESTRUCTURADO	PARADIGMA ORIENTADO A OBJETOS	PARADIGMA ORIENTADO A LA WEB
Descripción	<p>También se denomina "clásica" o "Ingeniería del Software orientada a funciones".</p> <ul style="list-style-type: none"> <li>• Es aplicada desde principios de los años 70.</li> <li>• Se basa en la idea de que un sistema software se compone de funciones que procesan datos. (Existen unas funciones generales y otras funciones específicas en las que se descomponen las primeras).</li> <li>• La Ingeniería del Software estructurada considera que el desarrollo de software ha de hacerse de forma descendente (top-down: desde una visión general cercana al usuario, hasta un nivel de abstracción mas detallado, cercano al programador), proponiendo la creación de modelos del sistema que representen de manera descendiente los siguientes aspectos. <ul style="list-style-type: none"> <li>○ Las funciones (también llamadas "proceso") llevadas a cabo por el sistema.</li> <li>○ Los flujos de datos de entrada, salida e internos de cada función del sistema.</li> <li>○ La estructura de los datos procesados por las funciones del sistema.</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Se empieza a aplicar a finales de los años 80.</li> <li>• Se basa en la idea de que un sistema software se compone de objetos software que interactúan entre si.</li> <li>• La funcionalidad de un sistema se reparten entre los objetos, asignando a cada objeto funciones específicas.</li> <li>• El objetivo final de la Ingeniería del software Orientado a Objetos es construir software de la misma forma como se construye el hardware: mediante el ensamblaje de componentes.</li> <li>• La ventaja de la orientación a objetos son: <ul style="list-style-type: none"> <li>○ Facilidad para la reutilización del software.</li> <li>○ Simplificación del mantenimiento software.</li> <li>○ Mejora de la calidad del software</li> </ul> </li> </ul>	<p>También denominada "WEB ENGINEERING".</p> <ul style="list-style-type: none"> <li>• Se trata de un tipo de Ingeniería del Software orientada a la naturaleza multidimensional de las aplicaciones web, que implican, además de programación: <ul style="list-style-type: none"> <li>○ Definición de estructuras complejas de información (XML, DTD,...)</li> <li>○ Diseño de estructuras de navegación.</li> <li>○ Gestión de contenidos.</li> <li>○ Diseño gráfico.</li> <li>○ Gestión De seguridad.</li> <li>○ Gestión de diferentes perfiles de usuario.</li> </ul> </li> </ul>
Técnicas de Análisis	<ul style="list-style-type: none"> <li>• Diagrama de Flujo de Datos</li> <li>• Diagrama de Entidad Relación.</li> <li>• Diagrama de historia de la vida de una entidad.</li> <li>• Matriz de funciones-entidades.</li> </ul>	<ul style="list-style-type: none"> <li>• Diagrama de Casos de Uso.</li> <li>• Diagrama de Actividades.</li> <li>• Diagrama de Secuencia.</li> <li>• Diagrama de Colaboración.</li> <li>• Diagrama de Estados.</li> </ul>	<ul style="list-style-type: none"> <li>• Diagrama de Entidad Relación.</li> <li>• Diagrama de Slices (Relación extendido).</li> <li>• Diagrama de Clases.</li> </ul>
Técnicas de Diseño	<ul style="list-style-type: none"> <li>• Diagrama de Estructura Modular.</li> <li>• Diagrama de Módulos</li> <li>• Diagrama de Estructura Lógica de un programa.</li> <li>• Diagrama de Árbol Programático</li> <li>• Diagrama de Estructura de Datos.</li> <li>• Diagrama de Tablas.</li> <li>• Diagrama de Flujos de Control</li> <li>• Pseudocódigo</li> </ul>	<ul style="list-style-type: none"> <li>• Las de análisis O.O.</li> <li>• Diagrama de Componentes</li> <li>• Diagrama de Despliegue.</li> <li>• Patrones de Diseño.</li> </ul>	<ul style="list-style-type: none"> <li>• Diagrama de Presentación.</li> <li>• Diagrama de Navegación.</li> <li>• Diagrama de Sincronización.</li> <li>• Diagrama de Composición.</li> <li>• Diagrama de Personalización.</li> </ul>
Técnicas de Programación	<ul style="list-style-type: none"> <li>• Estructura de Control, repetición.</li> <li>• Subprogramas</li> <li>• Lenguajes de programación estructurada: COBOL, FORTRAN, C, PASCAL, NATURAL</li> </ul>	<ul style="list-style-type: none"> <li>• Herencia: Objetos basados entre otros objetos.</li> <li>• Polimorfismo: Funciones con el mismo nombre.</li> <li>• Agregación: Objetos que contienen objetos.</li> <li>• Lenguajes de programación orientada a objetos: JAVA, C++, C#, VISUAL BASIC.</li> </ul>	<ul style="list-style-type: none"> <li>• Lenguajes de marcado (HTML, XML,...)</li> <li>• Lenguajes De script (JavaScrip, VBScript).</li> <li>• Programación GGI utilizando cualquier lenguaje (C, C++, Java,...)</li> <li>• Lenguajes de programación: estructurada y orientados a objetos (XML, Java, C, C++,...)</li> </ul>
Metodologías	<ul style="list-style-type: none"> <li>• MERISE</li> <li>• SSADM</li> <li>• MÉTRICA</li> <li>• Otras.</li> </ul>	<ul style="list-style-type: none"> <li>• METRICA3</li> <li>• UNIFIED PROCESS</li> <li>• FUSION</li> <li>• OPEN</li> <li>• Otras.</li> </ul>	<ul style="list-style-type: none"> <li>• OPEN</li> <li>• HDM</li> <li>• OOHDM</li> <li>• RMM</li> <li>• EORM</li> <li>• Otras.</li> </ul>

Fuente: www.tema1-1.3.pdf.

Elaborado por: Jenny Chicaiza & Luis Quispe



## **1.4 METODOLOGIAS DE DESARROLLO DE SOFTWARE**

### **1.4.1 INTRODUCCION A LAS METODOLOGÍA DE DESARROLLO DE SOFTWARE**

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software, se van indicando paso a paso todas las actividades a realizar para lograr el producto informático, las personas deben participar y el qué papel que deben de tener. Además detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzar.

Actualmente es imprescindible considerar los riesgos, aunque habitualmente las empresas, no han sido de los riesgos inherentes al procesamiento de la información mediante ordenadores, a lo que han contribuido, a veces, los propios responsables de informática, que no han sabido explicar con la suficiente claridad las consecuencias de una política de seguridad insuficiente o incluso inexistente. Por otro lado, debido a una cierta deformación profesional en la aplicación de los criterios de coste / beneficio, el directivo desconocedor de la informática no acostumbra a autorizar inversiones que no lleven implícito un beneficio demostrable, tangible y mensurable.

Las técnicas indican cómo debe ser realizada una actividad, combina el empleo de unos modelos o representaciones gráficas junto con el empleo de unos procedimientos detallados; en consideración que una técnica determinada puede ser utilizada en una o más actividades de la metodología de desarrollo de software.

Las metodologías tradicionales hacen énfasis en la planeación, y las metodologías ágiles enfatizan la adaptabilidad del proceso. De manera paralela, el tema de modelos para el mejoramiento de los procesos de

desarrollo ocupa un lugar importante en la búsqueda de la metodología adecuada para producir software de calidad en cualquier contexto de desarrollo.

#### **1.4.2 METODOLOGIAS TRADICIONALES**

Las metodologías no ágiles son aquellas que están guiadas por una fuerte planificación durante todo el proceso de desarrollo; llamadas también metodologías tradicionales o clásicas, donde se realiza una intensa etapa de análisis y diseño antes de la construcción del sistema.

Se caracterizan por exponer procesos basados en planeación exhaustiva. Esta planeación se realiza esperando que el resultado de cada proceso sea determinante y predecible. La experiencia ha mostrado que, como consecuencia de las características del software, los resultados de los procesos no son siempre predecibles y sobre todo, es difícil predecir desde el comienzo del proyecto cada resultado. Sin embargo, es posible por medio de la recolección y estudio de métricas de desarrollo lograr realizar estimaciones acertadas en contextos de desarrollo repetibles.

Remontándose a la historia, el modelo de cascada fue uno de los primeros modelos de ciclo de vida (MCV) que formalizó un conjunto de procesos de desarrollo de software. Este MCV describe un orden secuencial en la ejecución de los procesos asociados. El modelo espiral se postuló como una alternativa al modelo de cascada. La ventaja de este modelo radica en el perfeccionamiento de las soluciones encontradas con cada ciclo de desarrollo, en términos de dar respuesta a los requerimientos inicialmente analizados. El modelo de cascada y el modelo espiral suponen, de manera general, que los requerimientos del cliente no cambian radicalmente en el transcurso del desarrollo del sistema.

Por otro lado, la realización de prototipos es una herramienta en la que se apoyan diferentes MCV. Un prototipo debe tener el objetivo de mostrar al cliente o a la gerencia del proyecto el resultado que se obtendrá de la implementación de cada uno de los requerimientos del cliente una vez terminado el desarrollo. Con los prototipos se tiene la posibilidad de obtener retroalimentación de manera temprana.

La solución a algunos de los problemas presentados por las metodologías tradicionales se logra con una gran evolución del modelo espiral. El proceso unificado propone la elaboración de varios ciclos de desarrollo, donde cada uno finaliza con la entrega al cliente de un producto terminado. Este se enmarca entre los conocidos modelos iterativo-incremental.

### 1.4.3 METODOLOGIAS AGILES

En febrero de 2001, tras una reunión celebrada en Utah-EEUU, nace el término “ágil” aplicado al desarrollo de software. En esta reunión participan un grupo de 17 expertos de la industria del software, incluyendo algunos de los creadores o impulsores de metodologías de software. Su objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto.

Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas.

Tras esta reunión se creó The Agile Alliance<sup>3</sup>, una organización, sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que adopten dichos conceptos. El punto de partida fue el Manifiesto Ágil, un documento que resume la filosofía “ágil”.

Los métodos ágiles son también denominados livianos (lightweight), adaptativos e iterativos.

- **Livianos**, puesto que ellos se consideran más fáciles de usar y no enfatizan la planificación y documentación detallada como sí lo hacen los métodos tradicionales más formales, que en contraste con las ágiles se denominan pesados (heavyweight).

- **Adaptativos**, porque consideran los cambios como una realidad inevitable y no como excepciones. Los métodos ágiles permiten una rápida reacción frente a estos.
- **Iterativos**, porque dividen el desarrollo del proyecto en ciclos muy cortos. Al final de cada ciclo una porción ejecutable del sistema es entregada al usuario para que éste la valide.

Existen diversas metodologías que coinciden en llamarse metodologías ágiles. Y aunque entre ellas comparten muchas características tienen también diferencias significativas. A continuación se presentan algunas de las metodologías ágiles más representativas.

- Programación Extrema XP (Extreme Programming),
- Open Source
- Crystal de Cockburn
- Desarrollo de Software Adaptable de Highsmith
- Scrum
- MIDAS
- Desarrollo Guiado por Aspectos
- DSDM (Método de Desarrollo de Sistema Dinámico)

- 1 **Extreme Programming (XP)** - La programación extrema (Beck, 1999; Mc Breen, 2000) concede una gran importancia a las pruebas del software (testing). Aunque la mayoría de los procesos las tienen en cuenta, generalmente lo contemplan de una forma demasiado ligera y superficial. Sin embargo, XP lo toma como base para el desarrollo y cada programador que escribe código crea casos de prueba. Estos forman parte del proceso continuo de generación de código y se integra continuamente con ello, lo que garantiza una plataforma estable para el futuro desarrollo. Sobre dicha plataforma se genera un proceso de diseño evolutivo, que es la base del sistema y que se enriquece con cada iteración. Nunca se generan diseños futuros. Según afirma Fowler (2001), el resultado es un proceso de diseño que combina adecuadamente la disciplina con la adaptabilidad.

- **Open source - Open source** apuesta por la distribución de trabajo entre diferentes equipos, al igual que ocurre con la mayoría de los procesos adaptativos. La mayoría de los proyectos open source cuentan con supervisores de código. Estos supervisores de código, son las únicas personas autorizadas para realizar un cambio en el repositorio del código fuente. Por otra parte, el resto del personal puede realizar cualquier cambio en el código base. Sin embargo, el supervisor del código es la persona responsable de coordinar y de mantener la consistencia del diseño del software.

Una de las principales ventajas de los desarrollos open source es que la depuración es altamente paralelizable, aunque un gran número de personas puedan verse involucradas. Cuando se soluciona un error, se envía la solución al supervisor de código, lo que garantiza que alguien realiza la modificación de forma fiable mientras otra parte del personal se dedica a las tareas de depuración.

- **Feature -Driven Development.** Define un proceso iterativo que consta de 5 pasos. Las iteraciones son cortas (hasta 2 semanas). Se centra en las fases de diseño e implementación del sistema partiendo de una lista de características que debe reunir el software. Sus impulsores son Jeff De Luca y Peter Coad.
- **Crystal Methodologies.** Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo y la reducción al máximo del número de artefactos producidos. Han sido desarrolladas por Alistair Cockburn. El desarrollo de software se considera un juego cooperativo de invención y comunicación, limitado por los recursos a utilizar. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener definidas políticas de trabajo en equipo. Estas políticas dependerán del tamaño del equipo, estableciéndose una clasificación por colores, por ejemplo Crystal Clear (3 a 8 miembros) y Crystal Orange (25 a 50 miembros).

- **SCRUM5** Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años.

Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas *sprints*, con una duración de 30 días. El resultado de cada *sprint* es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.

- **Dynamic Systems Development Method.** Define el marco para desarrollar un proceso de producción de software. Nace en 1994 con el objetivo de crear una metodología RAD unificada. Sus principales características son: proceso iterativo e incremental, el equipo de desarrollo y el usuario trabajan juntos. Propone cinco fases: estudio viabilidad, estudio del negocio, modelado funcional, diseño y construcción, y finalmente implementación.
- **MIDAS**, es un marco metodológico orientado a modelos para el desarrollo de sistemas de Información Web con tecnología XML y OR. En la actualidad existen multitud de plataformas de explotación para los sistemas de información Web, así como diferentes tecnologías de implementación para el desarrollo de dichos sistemas. Con el fin de garantizar una especificación independiente de la plataforma de explotación y de la tecnología de implementación, OMG propone realizar una arquitectura basada en modelos (MDA), que garantiza la especificación completa de un sistema en base a modelos, independientes y específicos de una tecnología y plataforma. En base a esta arquitectura, se define un marco metodológico basado en modelos para el desarrollo de sistemas de información Web, denominado MIDAS. Los modelos propuestos se agrupan según una dimensión estructural una dimensión de comportamiento, en base a las diferentes dimensiones del modelado de los SIW. Por otra parte, el límite entre los modelos independientes de plataforma y específicos de ella, lo fija el diseño lógico.

La aparición de aplicaciones y sitios Web proporciona la explotación de otros mercados y servicios antes impensables como el comercio electrónico, la enseñanza virtual, etc., y esto conlleva un importante crecimiento en el desarrollo del software sobre dicha tecnología. Ahora bien, desde el punto de vista de la ingeniería del software es importante dotar de los mecanismos adecuados, para que la realización de este tipo de aplicaciones satisfaga las necesidades tanto de los usuarios como de los clientes que contratan el desarrollo de este tipo de aplicaciones.

- **Lean Development (LD).** Definida por Bob Charette's a partir de su experiencia en proyectos con la industria japonesa del automóvil en los años 80 y utilizada en numerosos proyectos de telecomunicaciones en Europa. En LD, los cambios se consideran riesgos, pero si se manejan adecuadamente se pueden convertir en oportunidades que mejoren la productividad del cliente. Su principal característica es introducir un mecanismo para implementar dichos cambios.

#### **1.4.4 DIFERENCIAS ENTRE METODOLOGIAS TRADICIONALES Y METODOLOGIAS AGILES**

Las metodologías tradicionales imponen una disciplina de trabajo sobre el proceso de desarrollo del software, con el objetivo de conseguir un software más eficiente y predecible.

Las metodologías tradicionales no se adaptan a los cambios, por lo que no son métodos adecuados cuando se trabaja en un entorno, donde los requisitos no pueden predecirse o bien pueden variar.

En contraposición a estas metodologías clásicas, en los últimos años ha aparecido un nuevo grupo de metodologías, que se identifica como metodologías ágiles. Aportan como novedad, nuevos métodos de trabajo que apuestan por una cantidad apropiada de proceso. Es decir, ni se pierden en una excesiva cantidad de cuestiones burocráticas ni defienden tampoco la falta total de proceso.

Las diferencias existentes entre ambos grupos de metodologías surgen por un enfoque y objetivos diferentes. Y como principales diferencias, tenemos:

- Las metodologías ágiles son adaptativas más que predictivas. Las metodologías tradicionales potencian la planificación detallada de prácticamente todo el desarrollo software a largo plazo. Pero cuando se produce un cambio, toda esta planificación puede venirse abajo. Sin embargo, las metodologías ágiles proponen procesos que se adaptan y progresan con el cambio, llegando incluso hasta el punto de cambiar ellos mismos.
- Las metodologías ágiles están orientadas al personal más que orientadas al proceso. Intentan trabajar con la naturaleza del personal asignado al desarrollo, mas que contra ellos, de tal forma que permiten que la actividad de desarrollo software se convierta en una actividad grata e interesante.

**FIGURA 1.2 DIFERENCIA ENTRE METODOLOGÍAS AGILES Y TRADICIONALES**

<b>METODOLOGIAS AGILES</b>	<b>METODOLOGIAS TRADICIONALES</b>
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a cambios
Impuestas internamente (por el equipo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas / normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura de software	La arquitectura del software es esencial y se expresa mediante modelos



# CAPITULO II

## II. ESTUDIO DE LA METODOLOGIA MIDAS

En la industria software surgen continuamente nuevas tecnologías (Java, Linux, XML, SOAO, NET, etc) que se hacen cada vez más populares, obligando a las compañías a incorporar rápidamente dichas tecnologías en sus productos y procesos de desarrollo; además, con intervalos de tiempo cada vez más cortos, van apareciendo nuevas versiones de esas mismas tecnologías.

Hoy en día, el enfoque para el desarrollo de sistemas de información de las empresas se realiza a partir de la tecnología de desarrollo. Desde el punto de vista de la tecnología de información, este planteamiento es poco adecuado puesto que generalmente es más importante el conocimiento del propio negocio, de los productos y de los procesos de desarrollo implicados, que la plataforma o tecnología específica de implementación

Los sistemas de información Web (SIW) son, en la actualidad, instrumentos imprescindibles para la divulgación de información, así como para la dotación de servicios a los usuarios de la red.

Uno de los problemas con los que nos encontramos en el desarrollo de SIW, es que aunque existen algunas aproximaciones metodológicas, no existe una metodología universalmente aceptada que permita el desarrollo de este tipo de sistemas. pues es necesaria una metodología que ayude y guíe en el proceso de desarrollo de SIW. Y la tendencia actual en el campo del desarrollo software, apuesta por un lado, por el uso de prácticas ágiles como las de Extreme Programming, y por otra parte, por arquitecturas dirigidas por modelos, como propone OMG (Object Management Group) con MDA (Model Driven Architecture).

Con la idea de que el modelado sea independiente de la tecnología final de implementación, OMG ha creado MDA. MDA es un marco de trabajo para el desarrollo de software, orientado a modelos, que plantea la obtención del modelado del negocio, en base a modelos independientes de la plataforma (PIM), para después transformarlos en modelos específicos de la plataforma (PSI) a partir de unas guías de transformación entre los diferentes modelos.

La principal ventaja de MDA es la posibilidad de transformar, mediante las guías de transformación entre modelos, un único PIM en distintos PSM para las diferentes plataformas y tecnologías, y la posterior transformación de esos PSM en código. Para aprovechar estas ventajas, en los últimos años han aparecido numerosas herramientas basadas en MDA, que en la mayoría de los casos permiten la generación de código para distintas plataformas

Hoy en día, el desarrollo de los sistemas de información Web (SIW) ha despertado un gran interés tanto a nivel empresarial como a nivel investigador y docente; por otra parte, la creciente aparición de diferentes tecnologías y plataformas asociadas al desarrollo de sistemas de información, hace demasiado específico el modelado de un sistema. En este sentido, OMG ha propuesto la Arquitectura Dirigida por Modelos (MDA), (Miller y Mukerji, 2001) que garantiza la especificación completa de un sistema en base a modelos, independientes y específicos de una tecnología y plataforma.

Estos motivos han llevado a plantear un proceso software dirigido por modelos para el desarrollo ágil de SIW dentro del marco metodológico de MIDAS, este proceso metodológico combina además, la arquitectura de MDA con una arquitectura de capas, típica de las aplicaciones de negocio.

## **2.1 ARQUITECTURA DIRIGIDA POR MODELOS (MDA)**

### **2.1.1 INTRODUCCION DE MDA**

En los últimos años se ha tomado mayor interés e importancia al modelado en el desarrollo de cualquier tipo de software, debido a la facilidad que ofrece un buen diseño tanto en la hora de desarrollar como al hacer la integración y mantenimiento de sistemas de software.

OMG (Object Management Group), es una organización de compañías de sistemas de información creada en 1990 con el fin de potenciar el desarrollo de aplicaciones orientadas a objetos distribuidas; la misma define una arquitectura orientada a modelos y guías de transformación para recoger las especificaciones de un sistema, donde los productos que se generan son modelos formales, que pueden llegar a ser comprendidos por ordenadores.

“En el año 2001, el OMG definió un marco de trabajo nuevo llamado MDA (Model Driven Architecture); La clave del MDA es la importancia de los modelos en el proceso de desarrollo de software, la misma que propone la definición y uso de modelos a diferente nivel de abstracción, así como la posibilidad de la generación automática de código a partir de los modelos definidos y de las reglas de transformación entre dichos modelos.

### **2.1.2 DEFINICION DE MDA**

“MDA, es una aproximación para la especificación de sistemas cuyo principio es la separación de la funcionalidad del sistema de los aspectos dependientes de la plataforma destino. Esto lo realiza mediante una

arquitectura de modelos que provee un conjunto de guías para estructurar las especificaciones expresadas como modelos.”<sup>12</sup>

“MDA, es un mejor diseño de un sistema software, mediante un conjunto de modelos que incrementen la flexibilidad de la implementación, integración, mantenimiento, prueba y simulación de las aplicaciones finales.”<sup>13</sup>

“MDA, define una forma de construir software en la que se usan modelos del sistema, a distintos niveles de abstracción, para guiar todo el proceso de desarrollo, desde el análisis y diseño hasta el mantenimiento del sistema y su integración con futuros sistema”<sup>14</sup>

“MDA, ofrece un nivel mayor de abstracción, y luego usar las transformaciones MDA para refinar el modelo.”<sup>15</sup>

Para el autor, MDA es una arquitectura de modelos que guía todo el proceso de desarrollo, permitiendo flexibilidad en la implementación, integración, mantenimiento y prueba de las aplicaciones.

### **2.1.3 IMPORTANCIA DE MDA**

La arquitectura dirigida por modelos (Model Driven Arquitectura) es una especificación detallada por el OMG (Object Management Group) que integra diferentes especificaciones y estándares definidos con la finalidad de ofrecer una solución a los problemas relacionados con los cambios en los modelos de negocio, la tecnología y la adaptación de los sistemas de información.

Además permite el expansión de aplicaciones empresariales, diseñadas sin dependencias de plataforma de despliegue y expresado su diseño mediante el

---

<sup>12</sup> Object Management Group (OMG), 2001

<sup>13</sup> MDA Guide (Draft Versión 2), 2003

<sup>14</sup> Marcos, E. Cáceres, P., Vela, B. y Cavero, J.M. *MIDAS/DB: a Methodological Framework for Web Database Design*. DASWIS 2001. Yokohama (Japan), Noviembre, 2001. LNCS-2465. Springer Verlag. ISBN 3-540-44122-0. Septiembre, 2002.

<sup>15</sup> C. Szyperski, *Component Software. Beyond Object-Oriented Programming*, 2 edn. Addison-Wesley, 2002..

uso de UML y otros estándares, potencialmente en cualquier plataforma existente, abierta o propietaria, como servicios web, .Net, PHP, J2EE, u otras.

#### 2.1.4 CONCEPTOS DE MDA

Para entender MDA, sus características, funcionamiento y aplicación al proceso de desarrollo, revisaremos los conceptos básicos de MDA y su forma de uso.

- **Sistema.-** Los conceptos de MDA se definen centrados en la existencia o planteamiento de un sistema, que puede contener un simple sistema informático, o combinaciones de componentes en diferentes sistemas informáticos, o diferentes sistemas en diferentes organizaciones, etc.
- **Arquitectura.-** La arquitectura es la especificación de las partes del mismo, determina los tipos de modelos que deben ser usados, como preparar dichos modelos y las relaciones que existen entre los diferentes modelos.
- **Modelo.-** Un modelo de un sistema es una representación de una parte de la función, estructura y comportamiento de un sistema. Los modelos se presentan normalmente como una combinación de texto y dibujos. El texto se puede presentar en lenguaje de modelado, o en lenguaje natural.
- **Dirigido por modelos.-** MDA es un acercamiento al desarrollo de sistemas, que potencia el uso de modelos en el desarrollo, dirigir el ámbito del desarrollo, el diseño, la construcción, el despliegue, la operación, el mantenimiento y la modificación de los sistemas.
- **Metamodelos.-** Los modelos en sí mismos pueden ser considerados como un dominio cuyas características pueden abstraerse en otros modelos llamados metamodelos. Un *metamodelo* es un modelo cuyo dominio consiste de modelos. Los metamodelos describen la sintaxis y la semántica de los modelos.
- **Aplicación.-** En MDA se define el término aplicación como una funcionalidad que tiene que ser desarrollada. Por tanto podemos definir un

sistema en términos de la implementación de una o más aplicaciones, soportadas por una o más plataformas.

- **Punto de vista.-** Un punto de vista es una abstracción que hace uso de un conjunto de conceptos de arquitectura y reglas estructurales para centrarse en aspectos particulares del sistema, obteniendo un modelo simplificado.
- **Plataforma.-** Una plataforma es un conjunto de subsistemas y tecnologías que aportan un conjunto coherente de funcionalidades a través de interfaces y determinados patrones de uso, que cualquier aplicación que se construya para esa plataforma puede usar sin preocuparse por los detalles de la implementación o como se lleva a cabo la misma dentro de la plataforma.
- **Vista.-** Una vista es una representación del sistema desde un determinado punto de vista.
- **Independencia de la plataforma.-** La independencia de la plataforma es una cualidad que tienen que presentar los modelos. Lo que significa que un modelo es independiente de las facilidades o características que implementan las plataformas, de cualquier tipo.

### 2.1.5 MODELOS EN MDA

Los modelos en MDA tienen gran importancia, permiten la interpretación automática por parte de transformadores o compiladores de modelos, el cual supone que el modelo tiene asociadas sintaxis y semántica bien definidas.

MDA para el proceso de desarrollo propone los siguientes modelos:

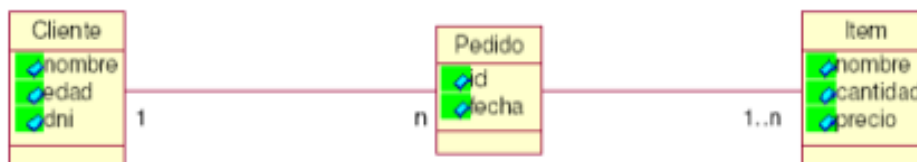
- **CIM (Modelos independientes de computación).-** que son modelos del más alto nivel de abstracción que identifican el contexto del sistema; se centra en el entorno del sistema y los requisitos para el mismo. Los detalles de la estructura y procesamiento del sistema no se muestran, o no están especificados.

- **PIM (Platform Independent Model).**- Un modelo Independiente de Plataforma, es un modelo del sistema de alto nivel que representa la estructura, funcionalidad y restricciones del sistema sin tener en cuenta aspectos técnicos e independientes de cualquier tecnología de implementación. Este modelo servirá de base para todo el proceso de desarrollo, y es el único que debe ser creado íntegramente por el desarrollador.

Al no incluir detalles específicos de una tecnología determinada, este modelo es útil en dos aspectos:

- Es funcional comprensible por los usuarios del sistema, y por lo tanto, resulta más sencillo validar la corrección del sistema.
- Facilita la creación de implementaciones del sistema en diferentes plataformas, dejando intacta la estructura y funcionalidad básica.

**FIGURA 2.1 PIM**



Fuente: proyecto-mda.pdf

Elaborado por: Jenny Chicaiza & Luis Quispe

- **PSM (Platform Specific Model).**- Un Modelo Especifico de Plataforma, es un modelo del sistema con detalles específicos de la plataforma en la que será implementado. Se genera a partir del PIM, así que representa el mismo sistema pero a distinto nivel de abstracción. Podemos decir que un PSM es un PIM al que se le añaden detalles específicos para ser implementado en una plataforma determinada.

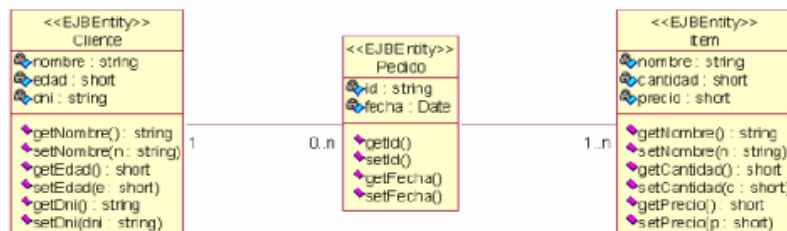
El PSM es, pues, un modelo del sistema de más bajo nivel, mucho más cercano a la vista de código que el PIM. Puede incluir más o menos detalle, dependiendo de su propósito.

PSM es construido a partir del PIM, representado también mediante un diagrama de clases UML; en el paso del PIM al PSM se han producido varias transformaciones:

- Se han añadido el estereotipo a cada clase, para indicar que la clase representa un Entity (Entidad).
- Se ha modificado la visibilidad de los atributos del PIM de público a privado.
- Se han generado métodos públicos de lectura y modificación (get y set) para cada atributo.

Hay que destacar que a partir de un mismo PIM pueden generarse varios PSMs, cada uno describiendo el sistema desde una perspectiva diferente

**FIGURA 2.2 PSM**



Fuente: proyecto-mda.pdf

Elaborado por: Jenny Chicaiza & Luis Quispe

### 2.1.6 OBJETIVOS DE MDA

Los principales objetivos de MDA son mejorar la productividad, la portabilidad, la interoperabilidad y la reutilización de los sistemas mediante el mantenimiento y documentación.

- **Productividad.-** El desarrollo recae sobre el PIM. Los PSMs se generan automáticamente a partir del PIM. Este enfoque centrado en el PIM aísla los



problemas específicos de cada plataforma y encaja mucho mejor con las necesidades de los usuarios finales, puesto que se puede añadir funcionalidad con menos esfuerzo.

- **Portabilidad.-** Se logra enfocando el desarrollo sobre el PIM. Al ser un modelo independiente de cualquier tecnología, todo lo definido en él es totalmente portable. El peso recae sobre las herramientas de transformación, que realizarán automáticamente el paso del PIM al PSM de la plataforma deseada.
- **Interoperabilidad.-** Genera no solo los PSMs, sino también los puentes entre ellos. Estos puentes son construidos por las herramientas de transformación, que es un pilar de MDA.
- **Mantenimiento y Documentación.-** El PIM desempeña el papel de la documentación de alto nivel que se necesita para cualquier sistema software. Pero la gran diferencia es que el PIM no se abandona tras la codificación. Los cambios realizados en el sistema se reflejarán en todos los niveles, mediante la regeneración de los PSMs y del código.

### 2.1.7 VENTAJAS DE MDA

“Las principales ventajas que aporta MDA son:

- Integración con sistemas legados.
- Utilización de estándares como UML, XMI, CWM, que proporcionen la posibilidad de intercambiar modelos a cualquier herramienta que sea compatible UML.
- La formalización a través MOF y OCL de todos los modelos definidos a partir de UML.

- Rápida inclusión de los beneficios de las tecnologías emergentes en sus sistemas existentes.”<sup>16</sup>

### 2.1.8 ESTANDARES DE MDA

MDA se apoya sobre los siguientes estándares para llevar a cabo su función, cada uno de estos permite las transformaciones de los modelos para el desarrollo de las aplicaciones:

- **UML:** Empleado para la definición de los modelos independientes de la plataforma y los modelos específicos de las plataformas destino. Es un estándar para el modelado introducido por el OMG. UML es un lenguaje gráfico para el modelado de sistemas, y nos permite modelar la arquitectura, los objetos, las interacciones entre objetos, datos y aspectos del ciclo de vida de una aplicación, así como otros aspectos más relacionados con el diseño de componentes incluyendo su construcción y despliegue.

Los elementos del modelado de UML, es decir, clases, interfaces, casos de uso, diagramas de actividad, etc. además pueden ser intercambiados entre herramientas del ciclo de vida utilizando XML. Actualmente nos encontramos con algunos perfiles UML para diferentes tecnologías como pueden ser PHP, EJB, etc. y otros muchas que están en desarrollo.

De entre los distintos tipos de modelos definidos en UML, los Modelos de Clases, que muestran la vista estática del sistema, son los más importantes dentro de MDA, ya que el PIM y la mayoría de PSMs son modelos de este tipo. Estos modelos se representan mediante Diagramas de Clases de UML.

- **MOF:** La OMG ha creado el estándar MOF, Meta Object Facility que extiende UML para que este sea aplicado en el modelado de diferentes sistemas de información. El estándar MOF define diversos metamodelos, esencialmente abstrayendo la forma y la estructura que describe los metamodelos.

MOF es un ejemplo de un meta modelo o un modelo del metamodelo orientado a objetos por naturaleza. Define los elementos esenciales,

---

<sup>16</sup> [www.ewh.ieee.org/reg/9/etrans/Marzo2005/paper141.pdf](http://www.ewh.ieee.org/reg/9/etrans/Marzo2005/paper141.pdf).

sintaxis y estructuras de metamodelos que son utilizados para construir modelos orientados a objetos. Además, proporciona un modelo común para los metamodelos de CWM y UML. Más en concreto, el estándar MOF proporciona:

- Un modelo abstracto de los objetos genéricos MOF y sus asociaciones.
- Un conjunto de reglas para el mapeo de metamodelos basados en MOF a interfaces independientes del lenguaje de programación definidos por medio del estándar PHP IDL. Una implementación de estos interfaces para un modelo concreto podría ser utilizado para acceder y modificar cualquier modelo basado en ese metamodelo.
- Reglas que definen el ciclo de vida, composición y semántica de los elementos basados en metamodelos MOF.
- Una jerarquía de interfaces reflexivos. Estos definen un conjunto de operaciones para localizar y manipular modelos basados en MOF pero de los cuales no conocemos sus interfaces.

El gran potencial de MOF reside en ínter operar entre metamodelos muy diferentes que representan dominios diversos. Las aplicaciones que utilizan MOF no tiene por que conocer los interfaces del dominio específico de algunos de sus modelos, pero pueden leer y actualizar los modelos utilizando las operaciones genéricas y reflexivas ofrecidas en los interfaces.

El meta- metamodelo, definido en la capa del metamodelo, es un metamodelo que describe el contenido de los metamodelos, es decir, los tipos de entidades que son compartidas a través de los diferentes sistemas de información. El estándar MOF proporciona los mecanismos para poder describir todos los tipos de sistemas de información y puede ser ampliado para incluir más sistemas de los cuatro predefinidos.

El MOF tiene bastantes ventajas sobre los sistemas de modelado simples y permite:

- Soportar cualquier tipo de modelo y paradigma de modelado imaginable.

- Permite relacionar diferentes tipos de metadatos.
- Permite añadir de forma incremental metamodelos y nuevos tipos de metadatos.
- Permite intercambiar modelos y metamodelos entre elementos que usen el mismo metametamodelo.

Esta arquitectura también tiene un conjunto de características que la diferencian de otras arquitecturas:

El modelo MOF está orientado a objetos y formado por elementos totalmente compatibles con el lenguaje de modelado UML; Las capas de la arquitectura MOF no son fijas y aunque normalmente utilizaremos cuatro niveles estos podrían ser más o menos en función de cómo se quiera utilizar MOF. Estas capas son simplemente un mecanismo para entender las relaciones entre los diferentes tipos de datos y metadatos; además este modelo está representado por medio del dibujo del paquete UML.

Esta característica tiene unas consecuencias importantes:

- Indica que el modelo MOF es lo suficientemente expresivo para realizar metamodelos.
- Permite a los interfaces MOF ser definidos por medio de mapeo IDL.
- Proporciona las bases de una arquitectura para su extensión y modificación.
- Permite nuevas implementaciones del repositorio del metamodelo y herramientas asociadas.
- **XMI:** XML Metada Interchange (XMI) es un estándar de la OMG que mapea MOF a XML. XMI define como deben ser usadas las etiquetas XML que son usadas para representar modelos MOF serializados en XML.

Los metamodelos MOF son convertidos en DTD (Document Type Definitions) y los modelos son convertidos en documentos XML que son consistentes con su DTD correspondiente. XMI resuelve muchos de los problemas encontrados cuando intentamos utilizar un lenguaje basado en etiquetas para representar objetos y sus asociaciones, y además el hecho

de que XMI esté basado en XML significa que tanto los metadatos (etiquetas) y las instancias que describen (elementos) pueden ser agrupados en el mismo documento, permitiendo a las aplicaciones entender rápidamente las instancias por medio de los metadatos. Muchas de las herramientas CASE como Rational Rose, Together, Omondo, etc.. soportan XMI y el estándar para importar y exportar el formato. XMI no solo puede ser usado como un formato de intercambio UML, sino que puede ser utilizado para cualquier formato descrito por medio de un metamodelo MOF.

XMI define muchos de los aspectos importantes involucrados en la descripción de objetos en XML:

- La representación de objetos en términos de elementos y atributos XML.
- Como los objetos están normalmente interconectados, XMI incluye un mecanismo estándar para enlazar objetos dentro del mismo fichero o entre fichero diferentes.
- La identidad de los objetos permite que estos sean referenciados por otros objetos en términos de ID y UUID.
- El versionado de objetos y sus definiciones son gestionadas por el modelo XMI
- La validación de documentos XMI se realiza por medio de sus DTD o XMI Schemas.

XMI proporciona reglas para la creación de DTD, XML Schemas y documentos XML, y para ello define varios tipos de reglas de producción:

- Producción de DTD desde un modelo de objetos.
  - Producción de XML Schemas desde un modelo de objetos.
  - Producción de documentos XML desde un modelo de objetos.
- **CWM:** Define la transformación de los modelos de datos en el modelo de negocio a los esquemas de base de datos.

## 2.1.9 PROCESO DE DESARROLLO SOFTWARE CON MDA

En el Proceso de Desarrollo de Software con MDA, se realiza el mismo proceso de recolección de información, prueba y implementación son iguales; lo que cambia son el análisis, diseño y codificación.

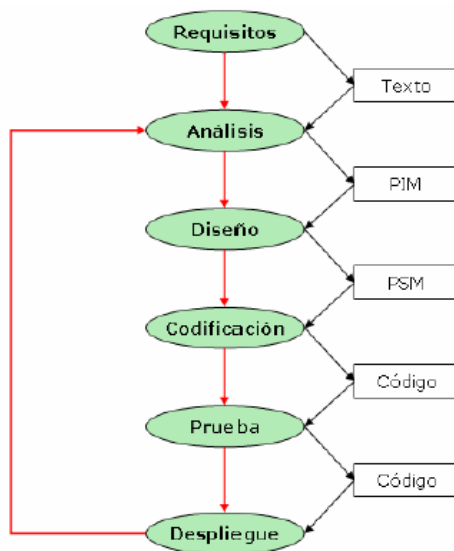
- **Análisis:** Un grupo especial de personas desarrollaran el PIM, guiados por las necesidades del negocio y la funcionalidad que debe incorporar el sistema.
- **Diseño.-** Otro grupo diferente de personas se encargarán de la transformación del PIM a uno o más PSMs. Estas personas tendrán conocimientos sobre distintas plataformas y arquitecturas, y conocerán también las transformaciones disponibles en las herramientas que usan.

De este modo podrán elegir la plataforma o arquitectura que mejor se adapte a los requisitos del sistema y establecer los parámetros de las distintas transformaciones. Los creadores de PSMs Tendrán comunicación constante con los diseñadores del PIM para tener más información sobre el sistema.

- **Codificación.-** Esta fase se reduce a generar el código del sistema mediante herramientas especializadas. Los programadores únicamente tendrán que añadir la funcionalidad que no puede reflejarse en los modelos, y si es necesario, “retocar” el código generado. Según los expertos, en un futuro desaparecerá esta fase de codificación dentro del MDA, y se pasará directamente del PSM a la fase de prueba.

Pero en este nuevo proceso de desarrollo aún falta un tercer grupo de personas, aquellos que escriben definiciones de transformaciones para empresas de construcción de software o principalmente, para los vendedores de herramientas MDA. Estas transformaciones son fundamentales para construir software con MDA, pues permiten a las herramientas pasar de PIM a PSM y de PSM a código.

### FIGURA 2.3 PROCESO DE DESARROLLO CON MDA



Fuente: proyecto-mda.pdf

Elaborado por: Jenny Chicaiza & Luis Quispe

### 2.1.10 FASES DEL DESARROLLO SOFTWARE CON MDA

El proceso de desarrollo de software con MDA se puede dividir en tres fases:

- **Construcción de un Modelo Independientemente de la Plataforma PIM,** un modelo de alto nivel del sistema independiente de cualquier tecnología o plataforma.
- **Transformación del modelo anterior a uno o varios Modelos Específicos de Plataforma.-** Un PSM es un modelo de más bajo nivel que el PIM que describe el sistema de acuerdo con una tecnología de implementación determinada.
- **Generación de código a partir de cada PSM.-** Debido a que cada PSM está muy ligado a una tecnología concreta, la transformación de cada PSM a código puede automatizarse.

**FIGURA 2.4 FASES DE DESARROLLO CON MDA**



Fuente: proyecto-mda.pdf

Elaborado por: Jenny Chicaiza & Luis Quispe

## **2.2 METODOLOGIA AGIL Y GENERICA PARA EL DESARROLLO DE SISTEMAS DE INFORMACION EN LA WEB (MIDAS)**

### **2.2.1 DEFINICIÓN DE MIDAS**

“MIDAS, es un marco metodológico dirigido por modelos para el desarrollo ágil de Sistemas de Información Web (SIW) basado con MDA, que propone el modelado de SIW teniendo en cuenta dos dimensiones: por un lado, el grado de dependencia de plataforma, es decir a nivel de CIM, PIM y PSM; y los aspectos más revelantes de un sistema de información: contenido, hipertexto y comportamiento”.<sup>17</sup>

“MIDAS, es un marco metodológico orientado a modelos para el desarrollo de Sistemas de Información Web con tecnología XML y OR.”<sup>18</sup>

“MIDAS consiste en un conjunto abierto de especificaciones técnicas que definen un modelo genérico orientado a la implementación de Software

<sup>17</sup> Cáceres, P, Marcos; Vela, B; A MDA-Based Approach for Web Information System Development. Workshop in Software Engineering in conjunction with the UML conference. San Francisco, 2003

<sup>18</sup> Marcos, E. Cáceres, P., Vela, B. y Cavero, J.M. *MIDAS/DB: a Methodological Framework for Web Database Design*. DASWIS 2001. Yokohama (Japan), Noviembre, 2001. LNCS-2465. Springer Verlag. ISBN 3-540-44122-0. Septiembre, 2002.



“MIDAS es una metodología genérica para el desarrollo de SIW, en el que se propone un proceso de desarrollo ágil integrado en una arquitectura dirigida por modelos, alineándose así con la propuesta MDA del OMG.”<sup>20</sup>

## 2.2.2 IMPORTANCIA DE MIDAS

MIDAS, es una metodología que presenta una visión unificada para desarrollar soluciones en las diferentes plataformas (Linux, Windows) como en el contenido, componentes del hardware y de software, con el análisis y modelamiento de SIW complejos, con énfasis especial en la tarea de la captura de los requisitos, los meta-modelos de especificación, y los métodos de hacer seguimiento en todo el desarrollo software.

MIDAS es una metodología dirigida por modelos para el desarrollo ágil de SIW, que a diferencia de las metodologías tradicionales que contemplan sólo la dimensión estructural para construir el modelo navegacional, realizando primero, la introducción de la dimensión de comportamiento y segundo, un enfoque desde una perspectiva orientada a servicios de usuario, es decir, los servicios del SIW que serán requeridos por el usuario, denominados servicios de usuario conceptuales.

MIDAS se dirige hacia un proceso iterativo e incremental. Un meta-modelo que se presenta para especificar apropiadamente sistemas SIW y se indican las notaciones correspondientes, que se basan totalmente en la notación de UML. El modelo de proceso también se presenta y para cada tarea que pertenezca a él, se describe el método correspondiente. La descripción de los

---

<sup>19</sup> Marcos, E. Cáceres, P., Vela, B. y Cavero, J.M. *MIDAS/DB: a Methodological Framework for Web Database Design*. DASWIS 2001. Yokohama (Japan), Noviembre, 2001. LNCS-2465. Springer Verlag. ISBN 3-540-44122-0. Septiembre, 2002.

<sup>20</sup> <http://www.di.uminho.pt/~jmf/PUBLI/papers/phd-Fernandes.pdf>, 2002.

métodos incluye las pautas que se pueden utilizar para lograr la tarea encomendada.

El uso de representaciones gráficas en MIDAS se muestra útil, siempre que las especificaciones no estuvieren sobrecargadas con demasiados pormenores.

### 2.2.3 CARACTERISTICAS DE MIDAS

- **Priorización de requisitos** - Secuencia de fases o incrementos. Selección de la fase de arranque según las necesidades.
- **Pequeñas versiones** – Proceso iterativo, en cada iteración terminada el resultado es un producto software (documento, prototipo o programa).
- **Diseño sencillo** – Técnicas y actividades necesarias no se fuerza la realización y aplicación de todas ellas.
- **Pruebas** – En cada iteración se realiza un producto Sw, que debe ser testeado.
- **Producción por pares** – Producción de productos Sw con pares de personas.
- **Cliente en el equipo** – El cliente debe incorporarse al equipo para cualquier duda que pueda surgir. Evita retrasos en el proyecto. Importante en cualquier proyecto.
- **Codificación estándar** – Acuñación de términos normalizados entre todos los participantes, utilización de lenguajes estándar.

- **Ingreso Operacional**, usa modelos de especificación que son vistos como programas escritos en un lenguaje de alto nivel. Esos programas son compilados y ejecutados a fin de que se detecten, durante el desarrollo del sistema, fallos e inconsistencias
- **Representación unificada.**- La representación del sistema va siendo progresivamente refinada hasta tener un modelo valido del sistema.

#### 2.2.4 MODELOS DE MIDAS

MIDAS se basa en la Arquitectura MDA; la cual tiene los siguientes modelos:

- **Modelos independientes de computación – CIM.**- Los CIM se corresponden con los modelos de dominio y de negocio del sistema. En MIDAS, se propone el diagrama de clases y el diagrama de casos de uso, respectivamente para modelar el contexto del sistema.
- **Modelos independientes de la plataforma – PIM.**- Los PIM proporcionan la especificación formal del sistema sin tener en cuenta aspectos técnicos ni la tecnología específica de implementación. A continuación se presentan los diferentes modelos, agrupados según una dimensión estructural y una dimensión de comportamiento:
  - **PIM de la Dimensión Estructural.** Recogen, a nivel conceptual, el contenido, el hipertexto y la presentación. A nivel de contenido se propone el diagrama de clases como modelo conceptual de datos. A nivel de hipertexto se propone realizar el modelo de hipertexto con dos técnicas: el modelado conceptual de fragmentos y de navegación propuestos por RMM, usando respectivamente los diagramas de fragmentos y de navegación propuestos en UWE.
  - **PIM de la Dimensión de Comportamiento.** Recogen, a nivel conceptual, el comportamiento y la lógica de negocio del sistema, en base a la identificación de servicios. Se propone el modelo de casos de uso, el modelo de servicios y el modelo de composición de servicios. Como modelo de servicio se propone el diagrama de colaboración, propuesto en UML y el diagrama de servicio, que modela cada uno de

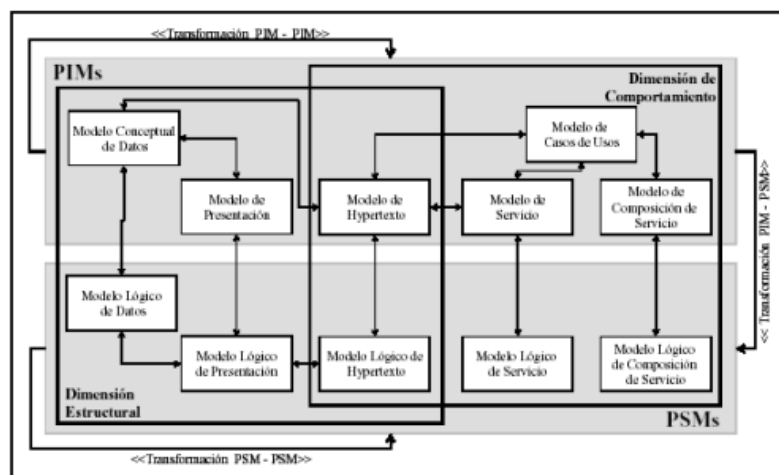
los servicios identificando las operaciones que realiza. Como modelo de composición de servicio se proponen los diagramas de actividad y de transición de estados, además del diagrama de secuencia, todos ellos propuestos en UML.

- **Modelos dependientes de la plataforma – PSM**

Los PSM proporcionan modelos en términos de constructores de implementación disponibles en una tecnología específica. Los modelos se presentan agrupados según una dimensión estructural y de comportamiento.

- **PSM de la Dimensión Estructural.** Son los modelos que representan, desde el nivel de diseño lógico hasta la implementación. la presentación y el hipertexto con la tecnología XML y la persistencia o contenido del sistema con la tecnología OR. A nivel de presentación, el modelo lógico se implementaría con XSL. A nivel de hipertexto, se propone representar el modelo lógico de fragmentos y de navegación con Xlink en UML extendido; y los fragmentos a nivel lógico con XMLschema en UML extendido. A nivel de contenido, el modelo lógico de datos.
- **PSM de la Dimensión de Comportamiento.** Son los modelos enfocados a recoger el comportamiento del sistema en base a lenguajes específicos. Para el modelado lógico de servicio se propone una extensión de UML para WSDL

**FIGURA 2.5 MODELO DE MIDAS**



Fuente: MIDAS.pdf

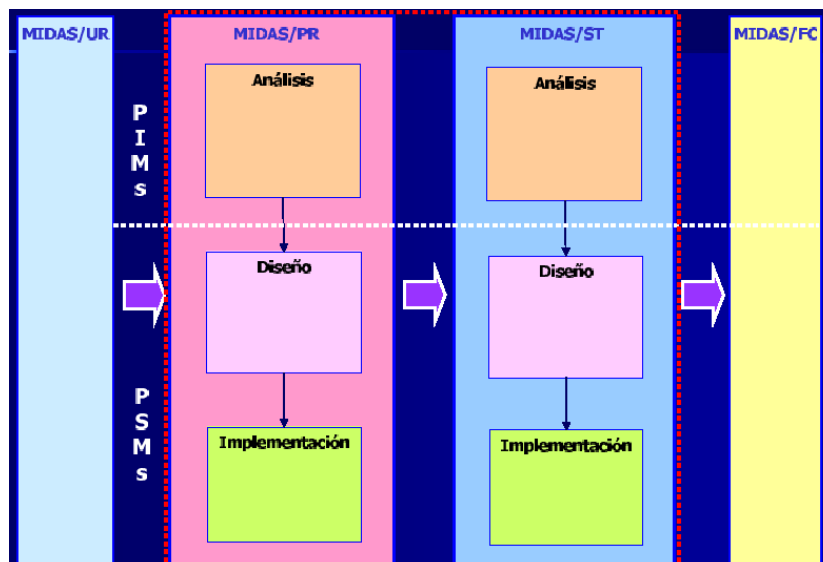
Elaborado por: Jenny Chicaiza & Luis Quispe

## 2.2.5 TRANSFORMACION DE MODELOS EN MIDAS

En MDA, las reglas de transformación proporcionan especificaciones para la transformación de modelos. Estas reglas se aplican para transformar modelos **PIM a PIM**, generalmente asociadas a los diferentes pasos que se suceden entre los modelos de especificación, análisis y diseño; modelos **PIM a PSM**, cuando el PIM está lo suficientemente refinado como para poder ser transformado en un modelo dependiente de la infraestructura final; modelos **PSM a PSM** aplicable al refinamiento de los modelos dependientes de la plataforma; y modelos **PSM a PIM** para permitir la abstracción de modelos a partir de implementaciones específicas de una plataforma y dependientes de una tecnología concreta.

En MIDAS, en la dimensión de comportamiento existen diversas propuestas relacionadas con la obtención del modelo de hipertexto a partir del modelo de casos de uso y con la obtención del modelo lógico de servicio a partir del modelo de servicio.

FIGURA 2.6 MODELO DE MIDAS



Fuente: MIDAS.pdf

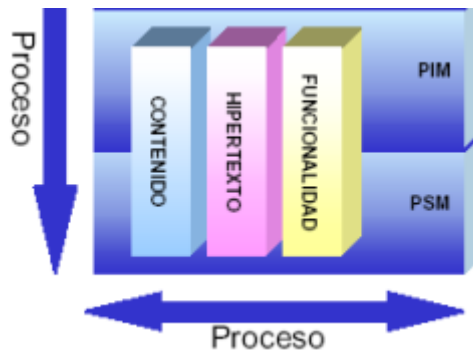
Elaborado por: Jenny Chicaiza & Luis Quispe

## 2.2.6 ARQUITECTURA DIRIGIDA MODELOS EN MIDAS

En MIDAS se ha combinado la arquitectura de MDA con una arquitectura de capas donde cada capa representa una vista del sistema (Kulkarni y Reddy, 2003). De

esta forma tenemos una representación bidimensional de nuestro proceso metodológico.

**FIGURA 2.7 DIMENSIONES DEL PROCESO METODOLOGICO DE MIDAS**



Fuente: MIDAS.pdf

Elaborado por: Jenny Chicaiza & Luis Quispe

La dimensión correspondiente al eje x, representa las diferentes vistas del SIW que son la vista del hipertexto, la vista del contenido y la vista de la funcionalidad. La dimensión correspondiente al eje y, representa la parte independiente (PIM) y dependiente de tecnología (PSM).

La intersección de ambas dimensiones ha dado lugar a un conjunto de modelos que representan, a nivel de PIM el modelado del hipertexto, del contenido y de la funcionalidad del sistema, desde el modelado conceptual hasta la etapa de diseño (excluido el diseño lógico); y a nivel de PSM representan también el modelado del hipertexto, del contenido y de la funcionalidad del sistema, desde la etapa de diseño lógico hasta la implementación.

El proceso de desarrollo de MIDAS, comienza siempre en la definición de los PIM, y posteriormente evolucionará hasta los PSM. Sin embargo desde el punto de vista del eje x, el proceso puede comenzar en diferentes puntos a nivel PIM. Puede comenzar en la vista de contenido o bien en la vista de funcionalidad con la definición del modelo conceptual de datos o de casos de uso, respectivamente. A continuación, las guías de transformación permiten pasar a definir, bien la vista de hipertexto a nivel de PIM, o bien continuar dentro de las mismas vistas para definir, ya a nivel de PSM, el modelado lógico. Esta decisión vendrá dada por las necesidades del cliente o bien por una priorización de requisitos realizada la etapa de especificación de requisitos.

**FIGURA 2.8 PROCESO DE DESARROLLO MIDAS**

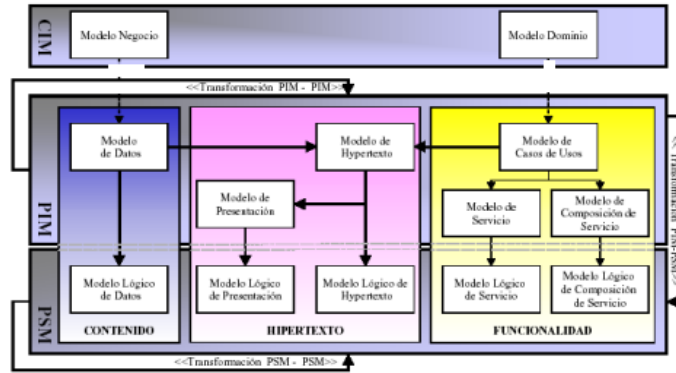


Fig. 3.- Proceso de desarrollo de MIDAS

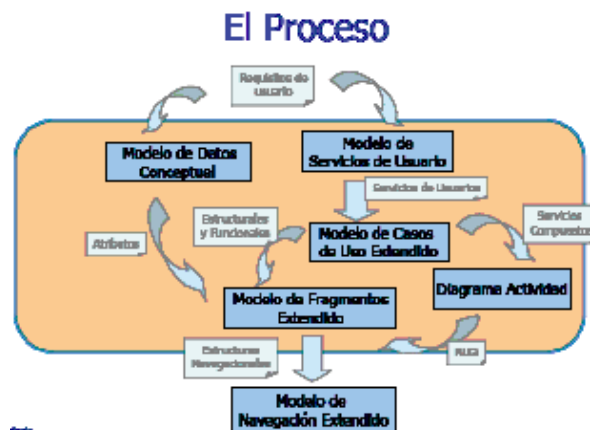
Fuente: MIDAS.pdf

Elaborado por: Jenny Chicaiza & Luis Quispe

## 2.2.7 PROCESO DE MIDAS

El modelo de proceso asociado en la Metodología MIDAS es presentado de una forma secuencial. En la realidad, su aplicación seguirá un modelo mas interactivo (con posibles conexiones hacia atrás). Típicamente el proceso es secuencial al inicio del desarrollo (estudio de viabilidad y análisis) pero, mas al frente se hace una mezcla de actividades (por ejemplo, la reutilización de un objeto, a partir de la instancia de una clase transformada inmediatamente un objeto analizado y codificado).

FIGURA 2.9 PROCESO DE MIDAS



Fuente: MIDAS.pdf

Elaborado por: Jenny Chicaiza & Luis Quispe

### 2.2.7.1 Modelo de Servicios de Usuario

Un **Servicio de Usuario Conceptual** representa un servicio requerido por el usuario, que debe estar fácilmente accesible en la interfaz de usuario.

**FIGURA 2.10 MODELO DE SERVICIOS DE USUARIO**



Fuente: MIDAS.pdf

Elaborado por: Jenny Chicaiza & Luis Quispe

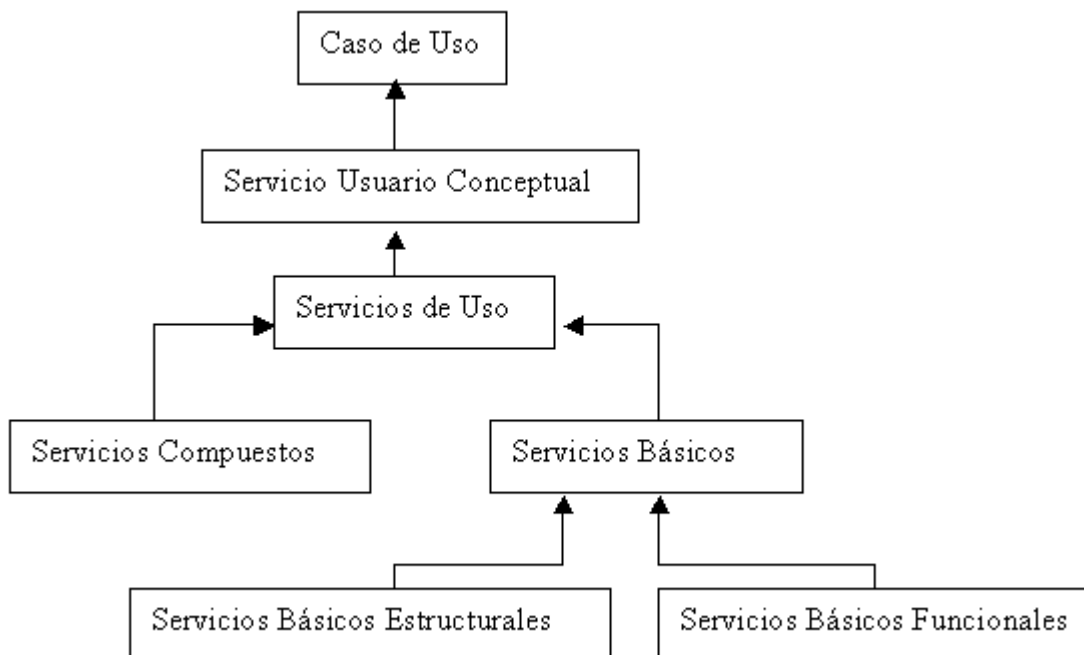
**Actividades:**

- Capturar los requisitos de usuario.
- Identificar los servicios de usuario.
- Representar el Modelo de Servicios de Usuario, estereotipando los servicios de usuario con <<CUS>>.

**2.2.7.2 Modelo de Casos de Uso Extendido**

**Servicio de Uso** representa las funcionalidades del sistema necesarios para llevar a cabo el servicio de usuario conceptual.

**FIGURA 2.11 MODELO DE CASOS DE USO EXTENDIDO**



Fuente: MIDAS.pdf



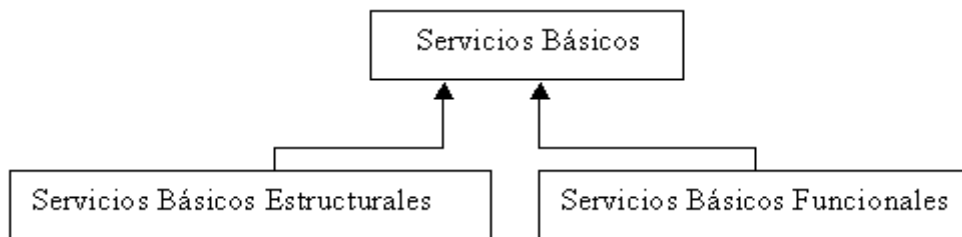
**Actividades:**

- Identificar los servicios de uso, descomponiendo los compuestos
- Incluir las relaciones include y extend
- Realizar los diagramas de actividad asociados a los servicios de uso compuestos

**2.2.7.3 Modelo de Fragmento Extendido**

Los **Fragmentos Estructurales y Funcionales** son unidades significativas de información o funcionalidad para el hipertexto

**FIGURA 2.12 MODELO DE FRAGMENTO EXTENDIDO**



Fuente: MIDAS.pdf

Elaborado por: Jenny Chicaiza & Luis Quispe

**Actividades:**

- Transformar los servicios básicos estructurales y funcionales en fragmentos estructurales y funcionales, respectivamente.
- Identificar los atributos de los fragmentos de los atributos procedentes del modelo de datos conceptual.
- Marcar la ruta asociada a los fragmentos procedentes de cada servicio de uso compuestos.

#### **2.2.7.4 Modelo de Navegación Extendido**

##### **Actividades:**

Añadir las estructuras navegacionales necesarias (índices, menús, etc...) al modelo de fragmentos extendido.

### **2.2.8 ANALISIS DE LA METODOLOGÍA MIDAS**

#### **2.2.8.1 Diagrama de Contexto**

El diagrama de contexto muestra el sistema, representado como una entidad única rodeado por todos los actores que con él interactúan. Además de eso, permite caracterizar la interfaz del sistema con el exterior, o sea, los mensajes y los eventos que fluyen entre el sistema y el respectivo ambiente.

Todos los actores del sistema le son externos, por lo que no deben ser desarrollados en el ámbito del proyecto del sistema. De esa forma, hay una definición de aquello que es el sistema que interesa desarrollar.

La estrategia para crear el diagrama de contexto consiste en los 3 pasos siguientes.

- Construir una lista de actores, entradas y salidas.
- Diseñar el diagrama de contexto, escogiendo el tipo adecuado para cada una de las conexiones entre actores y el sistema.
- Especificar pormenorizadamente las conexiones en una tabla.

Identificar los actores de un sistema, a pesar de ser una tarea simple puede revelarse difícil, pues muchas veces es complicado justificar si una funciónabilidad esta dentro o fuera del sistema, normalmente, es siempre necesaria alguna interacción de ese proceso hasta alcanzarse una solución estable.


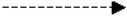


#### **2.2.8.2 Comunicación Interna – Objetos**

Las comunicaciones aquí descritas son igualmente aplicables a todos los diagramas en que existan conexiones entre objetos, expresamente, diagramas de clase, diagramas de objetos, diagramas de secuencia y diagramas de colaboración.

Regla general, la Metodologías para el desarrollo de Software usa solo, como medio de comunicación mensajes.

En midas, son posibles 4 tipos distinguidos de interconexiones entre objetos .

**FIGURA 2.13 COMUNICACIÓN INTERNA – OBJETOS EN MIDAS**

Tipo de Comunicación	Símbolo	Estereotipo
Interacción		<<Interacción>>
Evento		<<Evento>>
Flujo discreto de la Información		<<Información <u>discreta</u> >>
Flujo Continuo de la Información		<<Información <u>continua</u> >>

Fuente: MIDAS.pdf

Elaborado por: Jenny Chicaiza & Luis Quispe

- **Interacción:** retrata el habitual mecanismo de pasada de mensajes en software, siendo su semántica similar a la llamada de una función (envía, espera, y responde).
- **Evento:** Es usado, por un objeto, para señalar los cambios de su propio estado que pueden ser relevantes a otros objetos. Se trata de un mecanismo de comunicación fundamental para sistemas reactivos, usado con frecuencia en la especificación de los diagramas de estados de los objetos.
- **Flujo discreto de Información:** Expresa la transferencia de datos entre objetos de una forma directa.
- **Flujo continuo de información:** Este mecanismo de conexión es muy idéntico al anterior, a diferencia que la transferencia de datos entre objetos es de un modo continuo. Directamente el receptor de la

información tiene solo disponible para usar el ultimo valor de la información

### **2.2.8.3 Diagramas de Casos de Uso**

Para identificar los casos de uso de un sistema, la mejor “estrategia “ consiste en intentar encontrar, de entrada, los actores del sistema y, después, para cada uno de aquellos, recolectar una lista de posibles casos de uso.

Cada flujo completo de eventos, iniciado por un actor, representa un caso de uso del sistema en análisis. Los casos de uso pueden ser decompuestos en otros casos de uso repitiéndose el proceso hasta llegar a los escenarios que muestran secuencias de interacción entre objetos.

Para construir de los diagramas de casos de uso consiste en la identificación de los escenarios del sistema, a partir de los cuáles es posible, por agregación, inferir los casos de uso. Se sugiere la utilización de un estereotipo <interna> asociado a los casos de uso que representan esas tareas iniciadas internamente

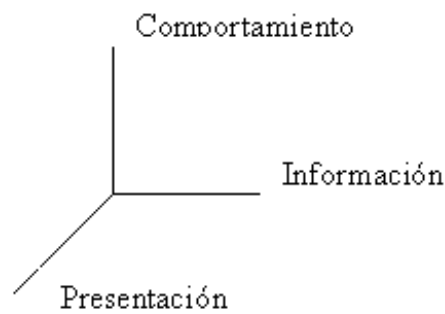
### **2.2.8.4 Diagramas de Objetos**

Una clase puede ser vista como un patrón que permite crear objetos para la aplicación en causa y, probablemente, para otras aplicaciones futuras, mientras que los objetos representan los elementos que constituyen realmente la aplicación.

Esta perspectiva permite concluir que es preferible no incorporar en el mismo modelo las clases y los objetos.

El espacio de análisis puede dividirse en tres dimensiones ortogonales: información, comportamiento y presentación.

**FIGURA 2.14 DIAGRAMA DE OBJETOS**



Fuente: MIDAS.pdf

Elaborado por: Jenny Chicaiza & Luis Quispe

- La dimensión información describe los datos del sistema, lo que permite especificar su estado interno.
- La dimensión comportamiento indica cuando y como el estado del sistema es alterado.
- La dimensión presentación procura los pormenores para presentar el sistema al exterior.

El modelo de análisis se construye a través de la especificación de objetos en este espacio tri - dimensional

- Una hipótesis, que no se recomienda, es usar objetos que expresan sólo una única dimensión.
- Otra hipótesis, que tampoco se recomienda, consiste en la utilización de objetos, que pueden ser colocados en cualquier posición del espacio.

- **Categorías de Objetos**

En MIDAS los diagramas de objetos se adoptaron la perspectiva que asocia, cada objeto, permitiendo obtener una estructura que puede, más fácilmente, adaptarse a los cambios

Cada una de las categorías está fuertemente relacionada con una de las dimensiones, los objetos pueden clasificarse en tres categorías: Objetos-interfaz, Objetos-entidad, Objetos-función.

- Un **Objeto-interfaz** modela comportamiento e información que depende de la interfaz del sistema, de la comunicación del sistema con los actores que con el interaccionan.
- Un **objeto-entidad** modela principalmente información, cuya existencia debe ser prolongada (no se incluyen, por lo tanto, datos con un carácter temporal). Todo el comportamiento asociado a la manipulación de esa información debe ser incluido en ese objeto-entidad; Ejemplo de un objeto-entidad considérese una cuenta bancaria con los respectivos atributos y operaciones.
- Un **objeto-función** modela comportamiento que no puede ser asociado, de una forma natural, a ningún otro objeto; Ejemplo: la funcionalidad que opera sobre varios objetos y queda vuelve el resultado a un objeto-interfaz.– Considérese un objeto que calcula el total de los saldos de un conjunto de cuentas bancarias.

#### 2.2.8.5 Diagramas de Clase

El diagrama de clases es un referencial para una categoría de aplicaciones que a partir de él se pueden construir el sistema. El modelo de clases es una generalización de alto nivel de los sistemas, cuando se define la forma como las clases se relacionan entre sí, está indica cuáles son los posibles sistemas que pueden ser contruidos a partir de esas clases.

La Metodología Midas ve el diagrama de clases sólo con un repositorio de especificaciones pre-definidas de objetos (“un almacén de materia prima”), que puede para cualquier aplicación que se pretenda desarrollar. En ese sentido, la estructura de clases puede ser compuesta por varios árboles no conectados entre si, no imponiéndose por lo tanto una única estructura para incluir todas las clases relevantes.

En Midas, la sub-clases son como semánticas relacionadas con las superclases. Esta perspectiva no viabiliza la reutilización de las implementaciones, que ocurre como consecuencia natural de las relaciones semánticas entre clases. A pesar del mecanismo de herencia no impone ninguna perspectiva, se adopta la practica de ver las superclases como más abstractas que las subclases, lo que permite que la jerarquía de las clases sea igualmente una jerarquía de abstracciones y que el mecanismo de herencia sea un esquema de clasificación, en el verdadero sentido de la palabra.

#### **2.2.8.5.1 Diagramas de Secuencia**

Los diagramas de casos de uso, para que sean una técnica de descripción útil y poderosa, no pueden ser considerados aisladamente, ya que contienen poca información sobre la funcionalidad del sistema. Pero, su utilización como estructura referencial para otros diagramas (expresamente diagramas de objetos y de estados) que describen, en por menor , la secuencia de acciones del sistema es extremadamente pertinente.

### **2.2.9 ACTIVIDADES DE DESARROLLO DE MIDAS**

#### **2.2.9.1 Análisis**

Cada paso a ejecutar en la fase de análisis indica la información que es usada como entrada y generada como salida; en los cuales corresponde a la creación de un único tipo de información útil para la inicialización de la fase de análisis; para la realización de estos pasos es necesario documentos y otras informaciones como: entrevistas con usuarios o conocimientos que el equipo de desarrollo adquirió por lectura o consulta de documentación.

FIGURA 2.15 ANALISIS DE MIDAS

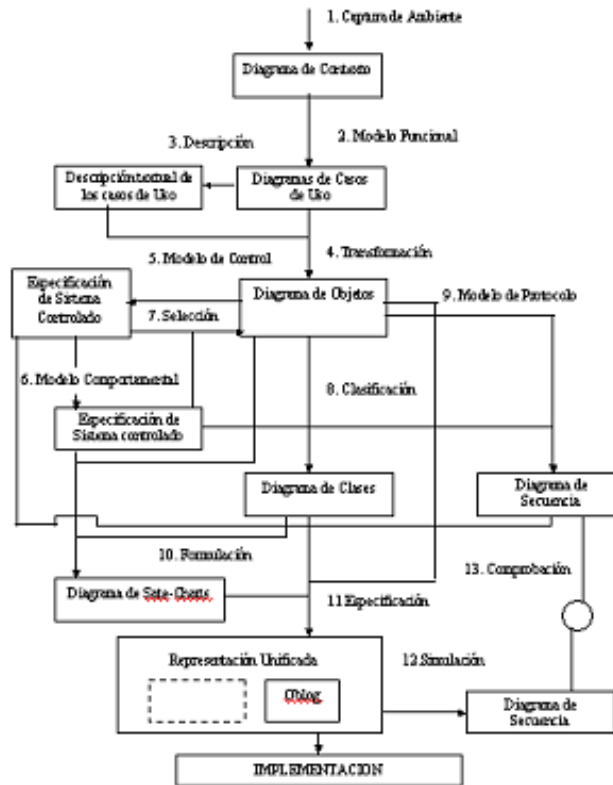


Figura 4 Fase Análisis en MIDAS

Fuente: MIDAS.pdf

Elaborado por: Jenny Chicaiza & Luis Quispe

MIDAS hace resumen descriptivo de esos pasos como son:

- 1. Captura del Ambiente:** Se obtiene el diagrama de contexto que muestra el sistema entendido como una entidad única y todo los actores visualizados, como entidades exteriores del sistema, con el que interaccionan.
- 2. Modelo funcional:** Obtenido del diagrama del contexto, el paso siguiente reside en la construcción del diagrama de casos de uso, que es entendido como un refinamiento de aquel. Para cada actor del sistema, especificado en el diagrama de contexto, deben identificarse los respectivos casos de uso.
- 3. Descripción:** En este paso, se describe para cada uno de los casos de uso indicados en el respectivo diagrama, una descripción textual que



caracteriza sus aspectos funcionales. Esa descripción debe indicar el tipo de información utilizada por el caso de uso.

4. **Transformaciones:** Los casos de uso son transformados en objeto; este paso constituye una de las innovaciones que la metodología MIDAS introdujo, pues no es conocida en otra metodología que siga un Ingreso Operacional
5. **Modelo de Control:** Para sistemas con una fuerte componente de control este paso es considerado como crucial, pues permite determinar las características del sistema controlado.
6. **Modelación de Comportamiento:** teniendo en cuenta el sistema controlado, este paso especifica la forma como el sistema será usado.
7. **Selección:** Este paso permite escoger los objetos a tener en consideración para el desarrollo.
8. **Clasificación:** Para los objetos creados se indican la respectiva clase. A través de este proceso de clasificación, deben identificarse objetos que pertenezcan a la misma clase, también relacionar las varias clases entre si.
9. **Modelo de Protocolos:** Teniendo en cuenta los objetos que constituyen el sistema y las conexiones entre ellos, se describen algunas interacciones (aquellas consideradas mas típicas o relevantes) entre algunos de los objetos del sistema. El numero y el nivel de por menores de las secuencias depende del sistema en causa y del tipo de información que consta del documento de requisitos.
10. **Formalización:** Por análisis de los diagramas de objetos y de clases para cualquier objeto /clase cuyo comportamiento dinámico se considere merecedor de ser especificado con tal formalismo.
11. **Especificaciones:** En este paso, con base en los diagramas de objetos, y de clases , se procede a la creación de una especificación; esta

transformación constituye igualmente una de las contribuciones innovadoras de la Metodología Midas.

12. **Simulación:** Usando las especificaciones como entrada, este paso permite simular el sistema (o partes de él), siendo creados, como salida, diagramas de secuencia que describen situaciones de funcionamiento del modelo especificado.

13. **Comprobación:** Con base en los dos conjuntos de diagramas de secuencia obtenidos en los pasos 10 y 12, se hace una comprobación de los resultados de ambos conjuntos, verificando si el comportamiento que se pretende obtener (descritos por los diagramas de secuencia del paso 10) corresponden al comportamiento especificado (descrito por los diagramas de secuencia del paso 12).

#### **2.2.9.2 Concepto e Implementación**

En el paso 9 puede también ser aplicado el diagrama de contexto, creándose así diagramas de secuencia para interacciones entre el sistema y sus actores.

El paso 11 impone la generación de una especificación unificada usando un lenguaje dado.

En el paso 12, es posible simular el funcionamiento del sistema mas correctamente, se puede afirmar que la Metodología MIDAS facilita la posibilidad de seguir, durante el proceso de desarrollo, a una Ingreso Operacional.

El resultado de la comparación que se obtiene en el paso 13 permite que se pueda determinar si el sistema esta o no correctamente especificado. En caso de ser afirmativo, los resultados obtenidos pueden ser usados como entrada para la fase de concepto. De lo contrario , debe ejecutarse una nueva interacción de análisis en la expectativa de encontrar y corregir totalmente las no conformidades destacadas en la comparación. Claramente , esta posibilidad

encaja en el uso de un modelo de proceso iterativo (expresamente, el modelo en espiral).

Durante el levantamiento de requisitos , las restricciones de tiempo real deben ser recogidas y analizadas, lo que se hace asociado esas restricciones a los casos de uso del sistema. Pero, puede hacerse asociar las restricciones de tiempo real de cada caso de uso de los objetos que dio origen, facilitando así el desarrollo del sistema en las fases de concepto e implementación. Es solo en estas fases que esas restricciones deben ser consideradas.

En la fase de concepto, el diagrama de objetos es la base para introducir, en los modelos, los promotores de implementación a tomar en consideración. Así, para que esa transformación sea posible, deben ejecutarse los siguientes pasos:

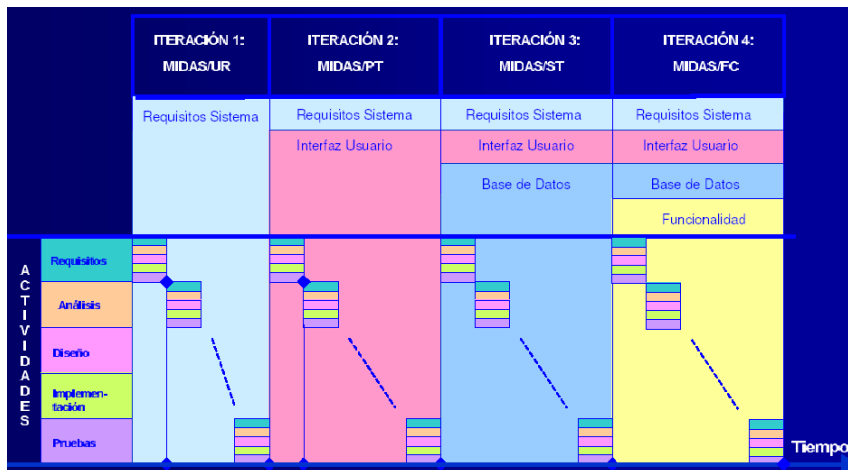
- Identificar el ambiente de implementación que va a soportar el sistema.
- Desarrollar una primera versión del ambiente de concepto.
- Describir la forma como los objetos interactúan.

El proceso de concepto puede ser conducido según dos alternativas: traducción o elaboración.

- En la traducción, el modelo de análisis es transformado, de una forma mas o menos automática, en un sistema ejecutable, a través de un traductor. Es importante tener gran cuidado en la construcción de traductor, que es normalmente usado en un dato del dominio de aplicación.
- En la elaboración el modelo de análisis, a través de la adición de por menores de concepto hasta que el sistema este completamente especificado.

Una cuestión que tendrá que ser tratada en la fase de conceptualización es la participación del hardware / software de las funcionalidades del sistema. Dependiendo del nivel de complejidad, la partición, según un determinado criterio, podrá determinar la restauración de los componentes del sistema (en el caso de la Metodología Midas, los Objetos).

**FIGURA 2.16 ACTIVIDADES DE DESARROLLO DE MIDAS**



Fuente: MIDAS.pdf

Elaborado por: Jenny Chicaiza & Luis Quispe

## 2.2.10 FASES DE DESARROLLO DE MIDAS

### 2.2.10.1 Fase de Definición MIDAS/UR

En esta fase se realiza las siguientes actividades: Definición del Caso Uso y Especificación de Requisitos; los mismos que están descritos en el Capítulo IV.

### 2.2.10.2 Fase de Hipermedia a la Web MIDAS/PR

En esta fase se realiza las siguientes actividades: Análisis, Diseño y Implementación las mismas que cumplen con una tarea, técnica y notación como podemos observar en la Tabla 2.1

**TABLA 2.1 ACTIVIDADES DE DESARROLLO WEB MIDAS/PR**

Actividad	Tarea	Técnica	Notación
<b>Análisis</b>	Diseño Conceptual de Datos	Modelo Conceptual de Datos (O.O)	Diagrama de Clases UML
	Diseño Conceptual del Hipertexto	Modelo Conceptual de	Diagrama de Fragmentos (UWE)

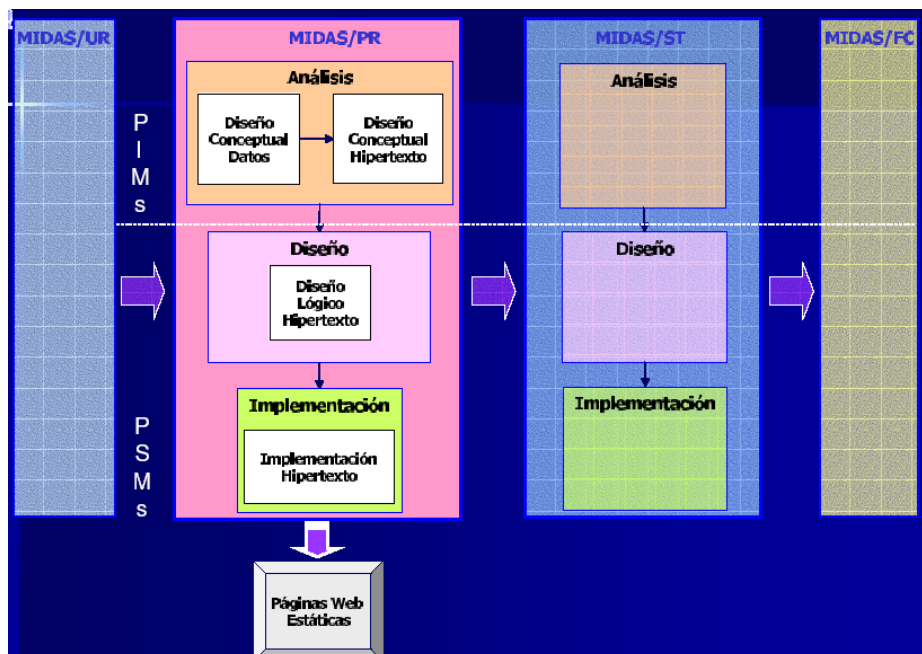
		Fragmentos Modelo Conceptual de Navegación	Diagrama de Navegación (UWE)
<b>Diseño</b>	Diseño Lógico del Hipertexto	Prototipado con herramientas de diseño gráfico	
<b>Implementación</b>	Implementación de la UI	Dream Weaver, XML.	HTML, XML

Fuente: MIDAS.pdf

Elaborado por: Jenny Chicaiza & Luis Quispe

Las actividades realizadas en el desarrollo de MIDAS/PR las podemos observar en la Figura 2.17; como es el desarrollo durante el proceso de esta fase y que técnicas realizamos en cada una de estas.

**FIGURA 2.17 ACTIVIDADES DE DESARROLLO DE MIDAS/PR**



Fuente: MIDAS.pdf

Elaborado por: Jenny Chicaiza & Luis Quispe

### 2.2.10.3 Fase de Base de Datos a la Web MIDAS/ST

En esta fase al igual que la fase de MIDAS/PR, se realiza las siguientes actividades: Análisis, Diseño y Implementación las mismas que cumplen con una tarea, técnica y notación como podemos observar en la Tabla 2.2.

**TABLA 2.2 ACTIVIDADES DE DESARROLLO WEB MIDAS/ST**

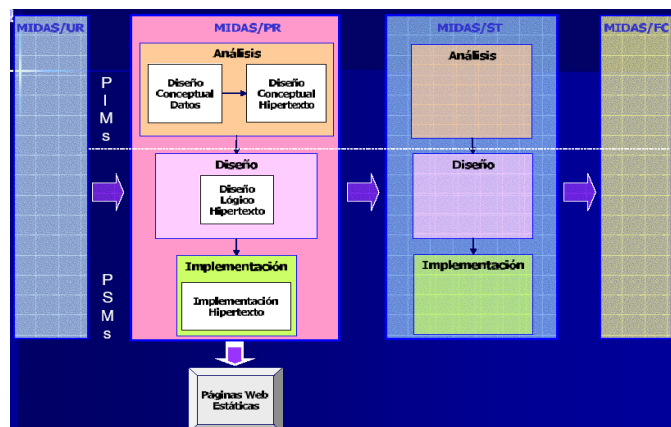
<b>Actividad</b>	<b>Tarea</b>	<b>Técnica</b>	<b>Notación</b>
<b>Análisis</b>	Diseño Conceptual refinado	Modelos Conceptuales	UML
<b>Diseño</b>	Diseño Lógico de Datos  Diseño Lógico del Hipertexto	Modelo (Objeto) – Realacional  Modelo Lógico de Fragmentos  Modelo Lógico de Navegación	Diagrama Objetos – Relacional  MIDAS – UML XML Schemas MIDAS – UML Xlink MIDAS - UML
<b>Implementación</b>	Implementación BD Implementación Hipertexto + Implementación Refinada de la IU Integración XML – SQL	SQL del Producto Concreto Tecnología XML MSQL XML XML Schema XLINK, XSL  PHP, JSP, ASP...	

Fuente: MIDAS.pdf

Elaborado por: Jenny Chicaiza & Luis Quispe

Las actividades realizadas en el desarrollo de MIDAS/ST las podemos observar en la Figura 2.18; como es el desarrollo durante el proceso de esta fase y que técnicas realizamos en cada una de estas.

**FIGURA 2.18 ACTIVIDADES DE DESARROLLO DE MIDAS/ST**



Fuente: MIDAS.pdf

Elaborado por: Jenny Chicaiza & Luis Quispe

Las fases de MIDAS/PR y MIDAS/ST van estrechamente relacionadas es por eso que estas dos fases se realizan conjuntamente, las mismas cumplen con ciclo de vida iterativo incremental, que nos permite realizar el análisis, diseño y implementación en cada una de ellas; como podemos observar en la Tabla 2.3.

**TABLA 2.3 ACTIVIDADES DE DESARROLLO MIDAS/PR y MIDAS/ST**

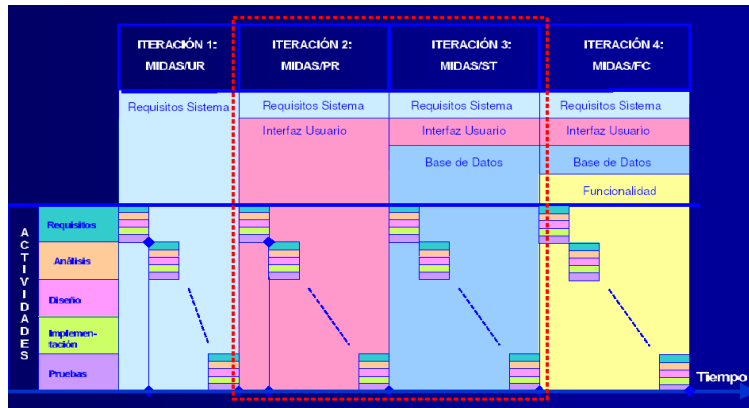
Actividad	Tarea	Técnica	Notación
<b>1.1.1..1.1 Análisis</b>	Diseño Conceptual de Datos Diseño Conceptual del Hipertexto	Modelo Conceptual de Datos (O.O) Modelo Conceptual de Fragmentos Modelo Conceptual de Navegación	Diagrama de Clases UML Diagrama de Fragmentos (UWE) Diagrama de Navegación (UWE)
<b>Diseño</b>	Diseño Lógico del Hipertexto	Prototipado con herramientas de diseño gráfico	
<b>Implementación</b>	Implementación de la UI	Dream Weaver, XML.	HTML, XML

Fuente: MIDAS.pdf

Elaborado por: Jenny Chicaiza & Luis Quispe

La relación que existe entre MIDAS/PR y MIDAS/ST las podemos observar en la Figura 2.19.

**FIGURA 2.19 ACTIVIDADES DE DESARROLLO DE MIDAS/ST Y MIDAS/PR**

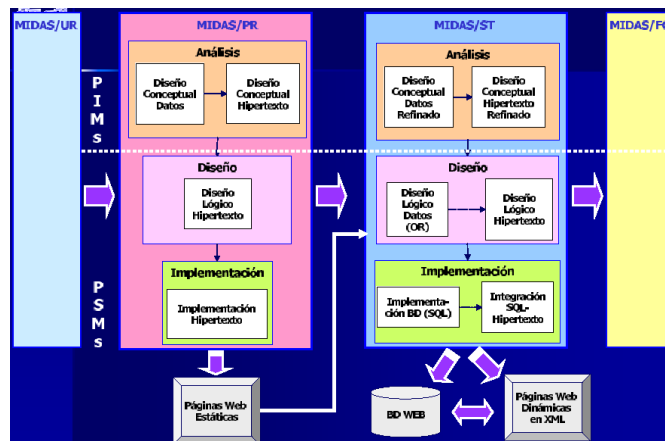


Fuente: MIDAS.pdf

Elaborado por: Jenny Chicaiza & Luis Quispe

El modelo de proceso asociado en la Metodología MIDAS es presentado de una forma secuencial. En la realidad, su aplicación sigue un modelo mas interactivo, como observamos en la Figura 2.19

**FIGURA 2.19 ACTIVIDADES DE DESARROLLO DE MIDAS/ST Y MIDAS/PR**



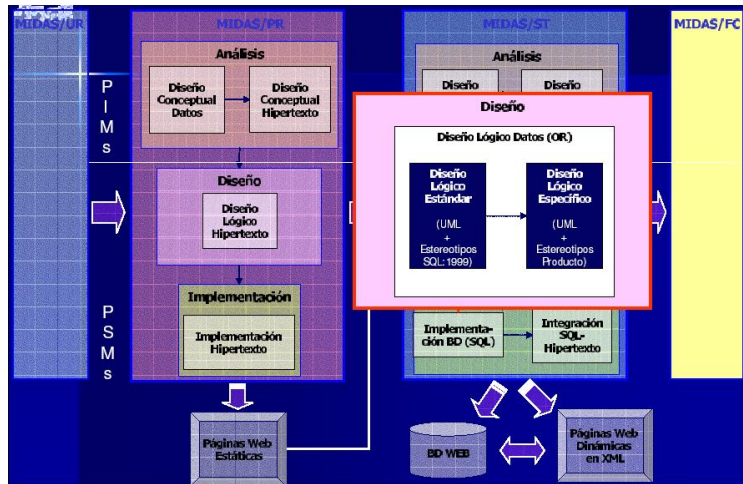
Fuente: MIDAS.pdf

Elaborado por: Jenny Chicaiza & Luis Quispe

Típicamente el proceso es secuencial al inicio pero durante su proceso se hace una mezcla de actividades, como observamos en la Figura 2.20

**FIGURA 2.20 ACTIVIDADES DE DESARROLLO DE MIDAS/ST Y MIDAS/PR**



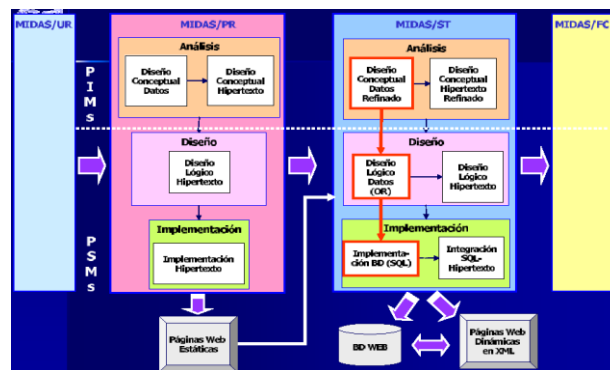


Fuente: MIDAS.pdf

Elaborado por: Jenny Chicaiza & Luis Quispe

Al realizar el proceso de desarrollo se puede determinar si el sistema esta o no correctamente especificado. En caso de ser afirmativo, los resultados obtenidos pueden ser usados como entrada para la fase de concepto. De lo contrario , debe ejecutarse una nueva interacción de análisis en la expectativa de encontrar y corregir totalmente las no conformidades. Claramente , esta posibilidad encaja en el uso de un modelo de proceso iterativo (expresamente, el modelo en espiral).

**FIGURA 2.21 ACTIVIDADES DE DESARROLLO DE MIDAS/ST Y MIDAS/PR**

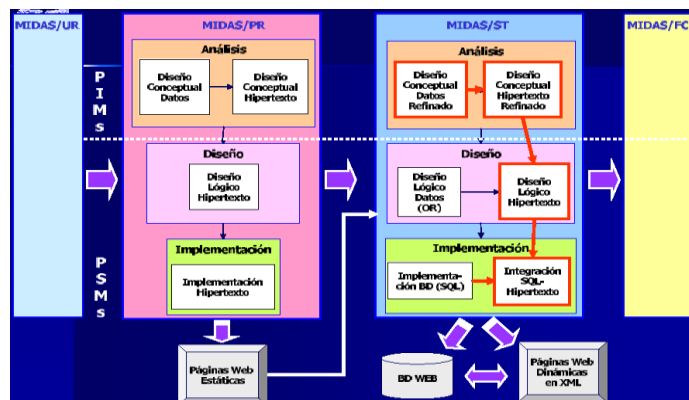


Fuente: MIDAS.pdf

Elaborado por: Jenny Chicaiza & Luis Quispe

Dependiendo del nivel de complejidad, la partición, según un determinado criterio, podrá determinar la restauración de los componentes del sistema (en el caso de la Metodología Midas, los Objetos).

**FIGURA 2.21 ACTIVIDADES DE DESARROLLO DE MIDAS/ST Y MIDAS/PR**



Fuente: MIDAS.pdf

Elaborado por: Jenny Chicaiza & Luis Quispe

# CAPITULO III

## III. ESTUDIO CONTABLE FINANCIERO

La Contabilidad se remonta desde tiempos muy antiguos, cuando el hombre se ve obligado a llevar registros y controles de sus propiedades porque su memoria no bastaba para guardar la información requerida. Se ha demostrado a través de diversos historiadores que en épocas como la egipcia o romana, se empleaban técnicas contables que se derivaban del intercambio comercial.

El inicio de la literatura contable queda circunscrito a la obra del franciscano Fray Luca Paccioli de 1494 titulado "La Suma de Aritmética, Geometria Proportioni et Proportionalitá", en donde se considera el concepto de partida doble por primera vez.

Actualmente, dentro de los sistemas de información empresarial, la contabilidad se funda como uno de los sistemas más notables y eficaces para

dar a conocer los diversos ámbitos de la información de las unidades de producción o empresas. El concepto ha evolucionado sobremanera, de forma que cada vez es mayor el grado de "especialización" de ésta disciplina dentro del entorno empresarial.

## **3.1 LA EMPRESA Y LA CONTABILIDAD**

### **3.1.1 LA EMPRESA**

#### **3.1.1.1 Definición de la Empresa**

La contabilidad se ha insertado en el sistema de información; por tanto, constituye parte esencial del mismo, y su ámbito natural es el negocio o empresa; además, su aplicación es muy apreciada en toda organización que maneje fondos y recursos con propósitos de asistencia social.

“La empresa, es todo ente económico cuyo esfuerzo se orienta a ofrecer bienes y/o servicios que, al ser vendidos, producirán una renta.”<sup>21</sup>

“En toda empresa, grande o pequeña, deben existir tres factores para que pueda realizar su actividad: personas, capital y trabajo.

- **Las Personas**, lo representan los propietarios, los administradores y los trabajadores que trabajan en la empresa.
- **El Capital**, lo constituyen los aportes que hacen los propietarios de la empresa, y puede estar representado en dinero en efectivo, mercancías, maquinaria, equipo de tecnología, muebles y otros bienes.
- **El Trabajo**, es la actividad que realizan las personas para lograr el objetivo de la empresa, el cual puede ser la administración, la

---

<sup>21</sup> Zapata, Pedro; Contabilidad General; Cuarta Edición; 2003; Pág. 13.

producción de bienes, la compra – venta de mercancías o la prestación de un servicio.”<sup>22</sup>

### 3.1.1.2 Clasificación de la Empresa

La clasificación de las empresas se realiza bajo los siguientes criterios:

#### a) Según su naturaleza

- **Industriales:** Aquellas que se dedican a la elaboración de productos terminados.
- **Comerciales:** Las que realizan actividades de compra – venta de productos.
- **De Servicios:** Son aquellas que prestan servicios intangibles, con el objeto de satisfacer necesidades complementarias de protección, mantenimiento, etc.

#### b) Según el sector al cual pertenezcan

- **Públicas:** Son aquellas que se financian con el presupuesto del Estado.
- **Privadas:** Aquellas que se constituyen con el aporte de capital del sector privado.
- **Mixtas:** Se constituyen con el aporte de capital tanto del sector privado como sector público.

#### c) Según la integración del capital

- **Unipersonales:** El capital aportado lo realiza una sola persona.
- **Sociedades o Compañías:** El aporte de capital lo realizan varios socios.

---

<sup>22</sup> Coral, Lucy; Gudiño, Emma; Contabilidad Universitaria, Cuarta Edición, 2001; Pág. 3

### **3.1.2 DEFINICIÓN DE CONTABILIDAD**

La contabilidad puede definirse, como la ciencia cuya finalidad consiste en recopilar, clasificar y registrar las operaciones mercantiles de una empresa, con objeto de poder conocer los resultados obtenidos y la situación en que la misma se encuentra que sirva de base para la toma de decisiones.

"La contabilidad, es el arte de recoger, resumir analizar e interpretar datos financieros, para obtener así las informaciones necesarias relacionadas con las operaciones de una empresa".<sup>23</sup>

"La contabilidad, es aquella que representa el registro de las transacciones comerciales, la preparación de estados financieros con los resúmenes de resultados de las transacciones y el análisis e interpretación de esos estados financieros y cifras estadísticas afines".<sup>24</sup>

"La contabilidad, es la ciencia y técnica que enseña a recopilar, clasificar y registrar, de una forma sistemática y estructural, las operaciones mercantiles realizadas por una empresa, con el fin de producir informes que, analizados e interpretados, permitan planear, controlar y tomar decisiones sobre la actividad de la empresa".<sup>25</sup>

### **3.1.3 OBJETIVOS DE CONTABILIDAD**

Proporcionar información a dueños, accionistas, bancos y gerentes, con relación a la naturaleza del valor de las cosas que el negocio deba a terceros, las cosas poseídas por los negocios. Sin embargo, su primordial objetivo es suministrar información razonada, con base en registros técnicos, de las operaciones realizadas por un ente privado o público. Para ello deberá realizar:

- Registros con bases en sistemas y procedimientos técnicos adaptados a la diversidad de operaciones que pueda realizar un determinado ente.
- Clasificar operaciones registradas como medio para obtener objetivos propuestos.

---

<sup>23</sup> Casas, Enrique; Biblioteca Moderna de Contabilidad; Octava Edición; Tomo VII, 2004; Pág. 7.

<sup>24</sup> Hernard, J; Hargadon, Jr; Munera Armando; Principios de Contabilidad; Editorial Norma; Pag 9.

<sup>25</sup> Coral, Lucy; Gudiño, Emma; Contabilidad Universitaria; Cuarta Edición, 2001; Pág. 98

- Interpretar los resultados con el fin de dar información detallada y razonada.

Con relación a la información suministrada, esta deberá cumplir con un objetivo administrativo y un financiero

- **Administrativo:** Ofrecer información a los usuarios internos para suministrar y facilitar a la administración intrínseca la planificación, toma de decisiones y control de operaciones.
- **Financiero:** Proporcionar información a usuarios externos de las operaciones realizadas por un ente, fundamentalmente en el pasado por lo que también se le denomina contabilidad histórica.

### **3.1.4 PRINCIPIOS DE CONTABILIDAD GENERALMENTE ACEPTADOS (PCGA)**

Los Principios de Contabilidad Generalmente Aceptados están determinados mediante Resolución por la Superintendencia de Compañías que se aplicarán obligatoriamente en la elaboración de los Balances de las compañías sujetas a su control; constituyen el conjunto de reglas y procedimientos necesarios para definir una práctica contable. Esta resolución contiene los principios de contabilidad que permitirán a las sociedades elaborar sus estados financieros con uniformidad; tomando como base la experiencia, la razón, el uso, la costumbre y en gran medida la necesidad de la práctica.

Los Principios de Contabilidad vigentes en el Ecuador están divididos en tres grupos:

- **Conceptos Básicos.-** Se consideran fundamentales por cuanto orientan la acción de la profesión contable y deben considerarse en la aplicación de los principios contables.

- **Conceptos Esenciales.-** Especifican el tratamiento que debe aplicarse al reconocimiento y medición de hechos ciertos que afectan la posición financiera y los resultados de las operaciones de las empresas. Estos principios proporcionan bases para la formación de otros principios.
- **Conceptos Generales de Operación.-** Los principios generales de operación determinan el registro, medición y presentación de la información financiera.

#### 3.1.4.1 Descripción de los Principios de Contabilidad.

- **“Principios de Equidad.-** Es el primer principio fundamental que debe orientar la acción del profesional contable, al determinar las operaciones económicas hechas por la empresa, debe prevalecer su ética de principio y moral esto quiere decir que el contador profesional debe ser imparcial entre la empresa donde trabaja y el estado.
- **Partida Doble.-** Permite expresar hechos económicos, y consiste en realizar una doble anotación, o sea, anotar en una cuenta en el debe y en otra cuenta en el haber. Siendo el importe anotado en el debe igual al anotado en el haber. Esta igualdad en el registro de cada operación hace que no se altere la igualdad patrimonial:
  - **$A=P+PN$**
- **Ente.-** Se refiere a que se debe llevar los estados financieros de la empresa como una organización económica y tomar a los dueños del negocio como personas terceras ajenas al patrimonio.
- **Bienes económicos.-** Son aquellos bienes materiales e inmateriales que posee la empresa, que le sirven para realizar sus operaciones económicas; y que están valorizado en términos económicos.
- **Moneda común denominador.-** Los hechos económicos, que conforman los estados financieros deben ser registrados en una sola moneda con la finalidad de poder compararlos y homogenizarlos.

- **Empresa en marcha.-** Todo organismo económico se considera en marcha cuando tiene continuidad es decir que sigue operando, esto se refleja en el éxito que se mide por la diferencia entre el valor de lo que vende o del servicio que presta y el costo de los recursos que se usan para obtener esos ingresos.
- **Valuación al costo.-** Es un concepto fundamental de la contabilidad en que los activos se registran al precio que se pagó por adquirirlos, cuyo concepto se relaciona con la continuidad de la empresa. La razón de valorizar estos activos a su precio de compra constituye el intento de estimar valores actuales del mercado.
- **Periodo.-**Es el tiempo que comprende una fecha a otra del 01 de enero al 31 de diciembre de cada año, cuya finalidad es para conocer el resultado de operación (perdida o ganancia) , la situación financiera y económica de la entidad , y verificar los cambios habidos en este periodo y comparándolos con otros.
- **Devengado.-** Se refiere a los derechos y obligaciones que habrá de vencer en fecha normal del ejercicio y/o posterior al cierre del ejercicio económico que tienen que ser regularizados al cierre del periodo económico, teniendo en cuenta el tiempo, sea a corto o mediano plazo.
- **Objetividad.-** La información que se procese debe ser registrada libre de prejuicios, de manera que refleje los acontecimientos con incidencia económica - financiera, de forma transparente y ser susceptible de verificación por parte de terceros independientes.
- **Realización.-** La contabilidad cuantifica en términos monetarios las operaciones que han sido realizadas, es decir concretadas con otros participantes de la actividad, siendo dada a través de una transacción económica.
- **Prudencia.-** Los ingresos y gastos no se deben sobreestimar, subestimar ni anticipar o diferir. Cuando se deba elegir entre más de una alternativa para medir un hecho económico - financiero, se elegirá aquella que no sobrevalúe los activos ni subvalúe los pasivos.



- **Uniformidad.-** Se refiere a que las operaciones que se registran en una empresa deben ser de una manera uniforme, es decir que si se ha decidido en uso de un método entonces deberá manejarse todas la subsecuentes operaciones iguales en la misma forma y aplicables uniformemente en un ejercicio u otro.
- **Significación o importancia relativas.-** Se refiere que el contador debe pasar hechos de poca importancia que no encuadran dentro de los principios o normas establecidos, estos no tienen un efecto relativo en el activo, pasivo, patrimonio o el resultado de operaciones.
- **Exposición.-** La información contable se encuentra expresada en los estados financieros estos deben ser claros y comprensibles para juzgar los resultados de operación y conocer la situación económica en la que se encuentra la empresa y poder hacer dediciones futuras.”<sup>26</sup>

## 3.2 REGISTRO CONTABLE

### 3.2.1 LA CUENTA CONTABLE

Es el nombre o designación que se otorga a un grupo de valores, obligaciones, bienes y servicios de la misma naturaleza, que sirve en Contabilidad para registrar, clasificar y resumir las transacciones que ocurren en la empresa.

“La cuenta se representa en forma de “T”, en la cual se identifican cuatro partes:

- 1) El título o nombre de la cuenta.
- 2) El sector izquierdo en el que se registran los débitos o cargos, que se denomina **Debe**; aquí se anotan los valores que incrementan el activo, los valores que representan pérdidas o gastos y la disminución del pasivo.
- 3) El sector derecho en el que se registran los créditos o abonos, que se denomina **Haber**; aquí constan los valores de aportes de capital

---

<sup>26</sup> <http://www.monografias.com/trabajos28/principios-contabilidad/principios-contabilidad.shtml>

(**Patrimonio**), de deudas contraídas (**Pasivos**), las rentas o ganancias y la disminución de los activos.

- 4) El saldo que se obtiene de la diferencia entre el Debe y el Haber; si la suma de Debe es mayor que la suma del Haber, se tiene un saldo deudor; si la sumatoria del Haber es mayor que la del Debe, se obtiene saldo acreedor, si las sumas son iguales, el saldo será nulo.”<sup>27</sup>

**FIGURA 3.1 LA CUENTA CONTABLE**

<b>TITULO O NOMBRE DE LA CUENTA</b>	
<b>DEBE</b>	<b>HABER</b>

Fuente: Zapata, Pedro; Contabilidad General; Cuarta Edición; 2003; Pág. 26

Elaborado por: Jenny Chicaiza & Luis Quispe

## **3.2.2 CODIGO Y PLAN DE CUENTAS**

### **3.2.2.1 Código de Cuentas**

Consiste en la simbología a través de números, letras y / o símbolos, en consecuencia el símbolo viene hacer el equivalente de las cuentas, que deben registrarse, informarse y procesarse.

Para una adecuada codificación, es recomendable lo siguiente:

- Clasificar los datos bajo algún criterio predeterminado.
- Hay que considerar la jerarquía de los datos, es un simple subrayado.
- Dentro de los sistemas es conveniente que se ajuste a las necesidades de información y naturaleza de los elementos a codificarse.

Para que los sistemas por computador y mecanizados funcionen bien son indispensables los códigos de cuentas, pero aún en los sistemas manuales, los códigos de cuentas ofrecen muchas ventajas.

---

<sup>27</sup> Zapata, Pedro; Contabilidad General; Cuarta Edición; 2003; Pág. 26.

## Estructura del Código de Cuentas

0	Elemento del Estado Financiero
0 0	Grupo
0 0 00	Cuenta
0 0 00 00	Subcuenta
0 0 00 00 00	Subcuenta Analítica

### 3.2.2.2 Plan de Cuentas

El plan de cuentas esta constituido por un listado lógico y ordenado de las cuentas que conforman un proceso contable, utilizando para el libro mayor y libros auxiliares aplicables a una entidad, con la identificación de un código respectivo

Este plan debe ser desarrollado de tal manera que no haya duplicación de las cuentas permitiendo un fácil entendimiento de la información contable en la presentación de los Estados Financieros en un período contable.

“Un plan de cuentas debe ser específico y particularizado, además debe reunir las siguientes características:

- Sistemático en el ordenamiento y presentación
- Flexible y capaz de aceptar nuevas cuentas
- Homogéneo en los agrupamientos practicados
- Claro en la denominación de las cuentas seleccionadas”<sup>28</sup>

## PLAN DE CUENTAS

---

<sup>28</sup> Zapata, Pedro; Contabilidad General; Cuarta Edición; 2003; Pág. 28.

Los planes de cuentas se realizan bajo el criterio de tipo de empresa y necesidades de información que requiera, para lo cual se debe realizar un estudio previo de sus requerimientos. Entre las principales cuentas tenemos:

## **1. ACTIVO (Según el criterio de liquidez)**

### **1.1. Activo corriente**

- 1.1.1. Caja
- 1.1.2. Caja Chica
- 1.1.3. Bancos
- 1.1.4. Cuentas por cobrar
- 1.1.5. Documentos por cobrar
- 1.1.6. Mercaderías
- 1.1.7. Inversiones financieras

### **1.2. Propiedad, planta y equipo (activo fijo)**

- 1.2.1. Edificios
- 1.2.2. Depreciación acumulada de edificios
- 1.2.3. Muebles y Enceres
- 1.2.4. Depreciación acumulada de muebles y enceres
- 1.2.5. Equipo de oficina
- 1.2.6. Depreciación acumulada de equipo de oficina
- 1.2.7. Equipo de computo
- 1.2.8. Depreciación acumulada de equipo de computo
- 1.2.9. Vehículo
- 1.2.10. Depreciación acumulada de vehículo

### **1.3. Diferidos y otros activos**

- 1.3.1. Gastos de constitución
- 1.3.2. Amortización acum. de gastos de constitución
- 1.3.3. Gastos de instalación

## **2. PASIVO (Según el criterio de temporalidad)**

### **2.1. Pasivo corriente**

2.1.1. Cuentas por pagar

2.1.2. Proveedores

2.1.3. Prestamos Bancarios a corto plazo

2.1.4. Obligaciones al IESS

2.1.5. IVA Cobrado

2.1.6. IVA por Pagar

### **2.2. Pasivo a largo plazo (pasivo fijo)**

2.2.1. Documentos por pagar

2.2.2. Hipotecas por pagar

2.2.3. Préstamo bancario a largo plazo

### **2.3. Diferidos y otros pasivos**

## **3. PATRIMONIO (Según el criterio de inmovilidad)**

### **3.1. Capital**

3.1.1. Capital

### **3.2. Reservas**

3.2.1. Reserva legal

3.2.2. Reserva Estatutaria

3.2.3. Reserva Facultativa

**3.3. Superávit de capital**

3.3.1. Superávit de Capital

**3.4. Resultados**

3.4.1. Resultados Acumulados

3.4.2. Resultado Ejercicio Anterior

3.4.3. Resultados Presente

**4. INGRESOS**

**4.1. Operacionales**

4.1.1. Ventas

**4.2. No operacionales**

4.2.1. Comisiones Ganadas

4.2.2. Intereses Ganados

**4.3. Extraordinarios**

**5. EGRESOS**

**5.1. GASTOS**

5.1.1. Gastos Administrativos

5.1.2. Gastos en Ventas

5.1.3. Gastos Financieros

**5.2. COSTOS**

5.2.1. Costos de Ventas

5.2.2. Costos de Producción

### 3.2.3 ANALISIS DEL PLAN DE CUENTAS

Con la finalidad de fortalecer conceptos del plan de cuentas, se presenta una breve descripción de los grupos y subgrupos que lo conforman y ejemplos de las principales cuentas que lo conforman.

- **Activo.-** Representa los bienes materiales, derechos de propiedad de la empresa apreciables en dinero y estén destinados al logro de sus objetivos.
  - **Activo Corriente.-** Integra el efectivo, cuentas corrientes, otros recursos y derechos que se espera convertirlos, en efectivo durante el período normal del negocio; que no exceda un año. Ejemplos: caja, caja chica, bancos, clientes, cuentas por cobrar, documentos por cobrar, mercaderías, etc.
  - **Propiedad, planta y equipo (activo fijo).-** Lo constituyen los bienes de propiedad de la empresa, generalmente se adquieren estos bienes en función a la razón de la empresa que están destinados y que tienen una vida más o menos duradera. Ejemplos: edificios, muebles de oficina, equipo de oficina, vehículos, equipo de computación, etc.
  - **Activos diferidos y otros activos.-** Registra los pagos por bienes o servicios que serán vencidos y amortizados en un período determinado. Ejemplos: gastos de constitución, arriendos pagados por anticipado, etc.
- **Pasivo.-** Son las deudas u obligaciones contraídas por la empresa con terceras personas. Son derechos que tienen los acreedores sobre un negocio personas o entidades ajenas.
  - **Pasivo Corriente.-** Conocido también como pasivo a corto plazo. Son las deudas contraídas por la empresa, cuya cancelación esta próximo a no más de un año. Ejemplos: proveedores, cuentas por pagar, documentos por pagar, retenciones por pagar, sueldos por pagar, etc.
  - **Pasivo a largo plazo.-** Son las obligaciones que tiene la empresa a entidades ajenas a esta, cuya cancelación deberá realizarse en un plazo

superior a un año. Ejemplos: documentos por pagar largo plazo, hipotecas por pagar, préstamos bancarios a largo plazo, etc.

- **Pasivos diferidos y otros pasivos.-** Son los valores cobrados por anticipado que serán devengados a través de la prestación de servicios en plazos mayores a un año. Ejemplos: arriendos cobrados por anticipado, comisiones cobradas por anticipado, ingresos cobrados por anticipado, etc.
- **Patrimonio.-** Esta representada por las aportaciones originales de los socios de la empresa y los resultados del ejercicio.
  - **Capital.-** Es el conjunto de valores, de diversas características, utilizados por una empresa para el desarrollo de sus operaciones económicas; está constituida por los derechos de los propietarios sobre la empresa o negocio.
  - **Reservas.-** Son fondos obtenidos de los rendimientos brutos anuales. Con el objetivo determinado por la Ley o Estatutos. Ejemplos: reserva legal, reserva facultativa, reserva estatutaria.
  - **Superávit.-** Registran las cuentas que incrementan el patrimonio de la empresa por donaciones de capital recibidas. Ejemplos. Donación de capital, reserva de capital, etc.
  - **Resultados.-** En esta parte se refleja la utilidad o pérdida obtenida en el ejercicio. Ejemplos: Utilidad de ejercicios anteriores, utilidad del presente ejercicio, pérdida de ejercicios anteriores, pérdida del presente ejercicio.
- **Ingresos.-** Son los valores que recibe la empresa por la actividad que realiza, o la prestación de servicios con el fin de obtener lucro.
  - **Ingresos operacionales.-** Son los que provienen de las operaciones propias de la actividad empresarial. Ejemplos: Ventas, ganancia bruta en ventas, ingresos por servicios prestados, utilidades realizadas, etc.
  - **Ingresos no operacionales.-** Son los valores adicionales que la empresa recibe, por actividades diferentes del giro normal del negocio.



Ejemplos: intereses ganados en cuenta corriente, arriendos ganados, comisiones ganadas, ingresos por multas, etc.

- **Gastos.-** Constituyen los diversos desembolsos que realiza la empresa para su buen funcionamiento.
  - **Gastos de administración.-** Pagos efectuados en el curso normal de las actividades de la empresa, dentro del área administrativa. Ejemplos: sueldos, bonificaciones, servicios básicos, útiles de oficina, etc.
  - **Gastos de venta.-** Proviene de los desembolsos que se realiza dentro de la gestión normal de ventas, con el objeto de alcanzar mayor eficiencia en la distribución de los productos. Ejemplos: sueldo, comisiones, publicidad, etc.
  - **Gastos financieros.-** Gastos originados en la obtención y uso de capitales de terceros, para financiar la compra de activos y las operaciones de la empresa. Ejemplos: intereses pagados, comisiones bancarias, etc.
  
- **Costos.-** Representa el valor de adquisición de los artículos para la elaboración productos terminados.
  - **Costos de ventas.-** Registra el costo de fabricación que representa para la empresa cada uno de los artículos terminados y listos para ser comercializados.
  - **Costos de producción.-** Esta constituido por el costo de los materiales indirectos, mano de obra indirecta y todos los gastos incurridos en la elaboración del producto

### **3.3 CICLO O PROCESO CONTABLE**

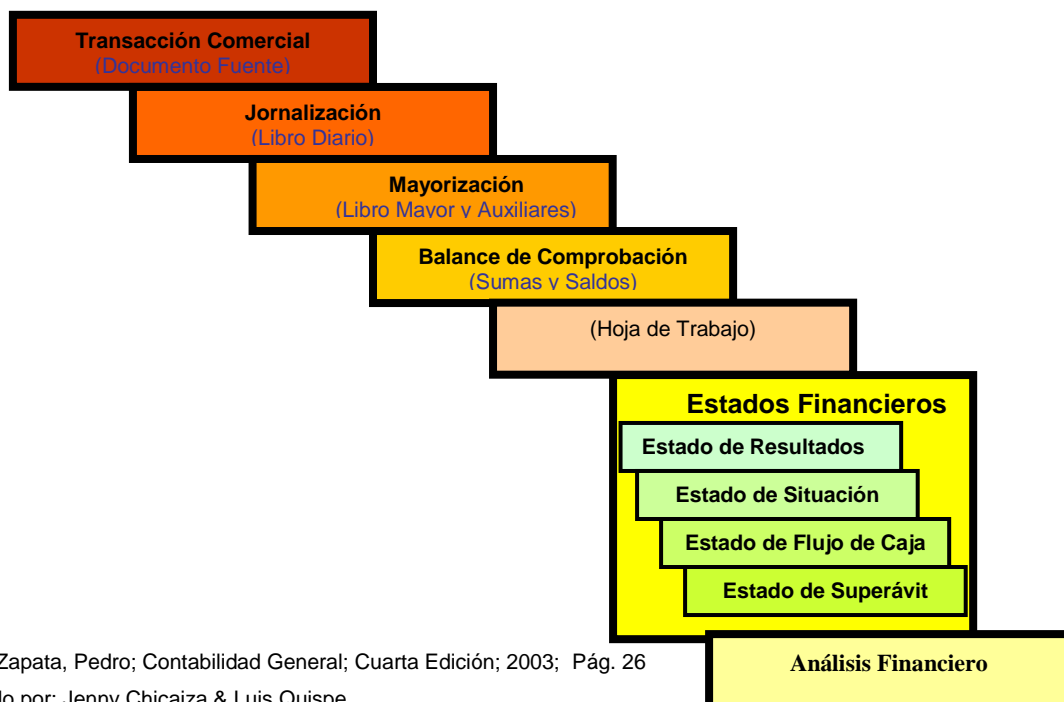
#### **3.3.1 DEFINICION**

Es el conjunto de pasos o fases de la contabilidad que se repiten en cada período contable, durante la vida de un negocio. Se inicia con el registro de las transacciones, su traspaso al diario, al libro mayor, la elaboración del balance de comprobación, la hoja de trabajo, los estados financieros.

El ciclo contable se refiere al proceso de registros que va desde el registro inicial de las transacciones hasta los estados financieros finales. Además de registrar las transacciones explícitas conforme van ocurriendo, el ciclo contable incluye los ajustes para las transacciones implícitas.

### 3.3.2 ESTRUCTURA DEL PROCESO CONTABLE

FIGURA 3.2 ESTRUCTURA DEL PROCESO CONTABLE



Fuente: Zapata, Pedro; Contabilidad General; Cuarta Edición; 2003; Pág. 26  
Elaborado por: Jenny Chicaiza & Luis Quispe

#### 3.3.2.1 Transacción Comercial

Intercambio de bienes y / o servicios, información que se obtiene del documento de origen o fuente de los registros contables, respaldan todas y cada una de las transacciones que se realizan en la empresa.

- **Documentos Fuente**

Constituyen la evidencia escrita que da origen a los registros contables y respaldan todas las transacciones que realiza la empresa. La clasificación de los documentos fuentes se realiza bajo los siguientes criterios:

- **Según su Importancia**

- **Documentos Principales:** Aquellos que se consideran indispensables para el registro de las operaciones. Ejemplo: Factura, Nota de Venta.
- **Documentos Secundarios:** Aquellos que complementan el expediente de la transacción. Ejemplo: Proformas, Actas de Entrega – Recepción.
- **Según su origen**
  - **Documentos Internos:** Aquellos documentos expedidos por la empresa.
  - **Documentos Externos:** Son los documentos que llegan a la empresa, con la finalidad de evidenciar las operaciones efectuadas.

### **Estudio de los Documentos Fuente**

Entre los documentos principales autorizados en el reglamento de comprobantes de venta y de retención del (Registro oficial 679 del 8 de octubre /02) tenemos:

- Comprobantes de Venta
  - Factura
  - Notas de Venta
- Documentos Complementarios
  - Notas de Crédito
  - Notas de Debito
- Comprobantes de Retención

Todas las empresas constituidas en sociedad y las personas naturales empresarias tienen la Obligación de Emisión de Comprobantes de Venta para lo cual se sujetan al Reglamento de Comprobantes de Venta y Retención.

“Están obligados a emitir y entregar comprobantes de venta todos los sujetos pasivos de los impuestos a la renta, al valor agregado y a los consumos especiales, sean sociedades o personas naturales, incluyendo las sucesiones indivisas, obligados o no a llevar contabilidad.

La obligación de la emisión de los comprobantes de venta nace con ocasión de la transferencia de bienes, aún cuando se realice a título gratuito, o de la prestación de

servicios de cualquier naturaleza, o se encuentren gravados con tarifa cero (0%) del impuesto al valor agregado.

Se encuentran exentos de la obligación de emitir comprobantes de venta, las instituciones del Estado que presten servicios que generan la emisión de documentos públicos tales como: cédula de identidad, licencia de conducir, matrícula de vehículos, entre otros.

### **Numeración de los Comprobantes de Venta**

Nº Comprobantes de venta:	<b>001 – 002 – 0000003</b>
Código del establecimiento	<b>001</b>
Punto de emisión	<b>002</b> <b>001</b> <b>002</b> <b>003</b>
Numeración secuencial	<b>0000001</b> <b>0000002</b> <b>0000003</b>

Podrá omitirse la impresión de los ceros a la izquierda del número secuencial pero deberán completarse los siete dígitos antes de iniciar la nueva numeración”<sup>29</sup>

#### **◆ Factura**

Es un documento con el que se documenta la venta de mercaderías u otros efectos. En él se hacen constar las mercaderías vendidas, en cantidades, precios e importes, las condiciones de pago y otros datos relativos a la operación. Una factura es de venta cuando sale de la empresa y es de compra cuando ingresa a la empresa.

Se emitirán y entregarán facturas, cuando se realiza transferencia de bienes o prestación de servicios a sociedades o personas naturales que tengan derecho al uso de crédito tributario.

---

<sup>29</sup> MATERIAL DE USO DIDÁCTICO PARA ESTUDIANTES DE LA U.T.P DE LOJA, Pág. 100

Al momento de su emisión se requiere de manera obligada identificar al comprador con su nombre o razón social, RUC. Debe constar el desglose del IVA.

**FIGURA 3.3 FACTURA**

AUTOMOVILES NACIONALES S.A. AUTOMAN		RUC 1790182345001 FACTURA No. 002-001-0000001	
MATRIZ: Av. 6 de Diciembre N24-50 y Gral. Foch Quito		No. Autorización 159984132	
SUCURSAL: Luis Plaza Danin 818 y M. Alcivar Guayaquil			
Sr. (es): .....	Fecha Emisión: .....		
RUC: .....	Guía de Remisión: .....		
Dirección: .....			
Cantidad	Descripción	Valor Unitario	Valor de Venta
		Subtotal	
		Descuento	
		IVA 0%	
		IVA 12%	
		TOTAL	
Carlos Alberto Bolívar Mora / Impresión Bolívar RUC: 171050142001 / No. Autorización 2540		Original: Adquiriente Copia: Emisor	

Válido para su emisión hasta 05/2004

Fuente: MATERIAL DE USO DIDÁCTICO PARA ESTUDIANTES DE LA U.T.P DE LOJA, Pág. 101

Elaborado por: Jenny Chicaiza & Luis Quispe

◆ **Notas de Crédito**

Es una notificación que se envía al cliente a favor de su cuenta. Estas se orientan a favor de los clientes en el Diario de Ventas y las Notas de Crédito a favor de la empresa en el Diario de Compras. Entre las causas más usuales son:

- Error en la facturación
- Devoluciones en compras de mercaderías
- Aceptación de descuentos o bonificaciones

El que reciba la nota de crédito, deberá consignar en ella su nombre, RUC O C.I, Fecha de recepción y de ser necesario, el sello de la empresa.

**FIGURA 3.4 NOTA DE CREDITO**

SISTEMAS COMPUTARIZADOS S.A. COMPUSISTEM		RUC 0990561448001 NOTA DE CREDITO No. 002-001-0000001	
MATRIZ: Av.de la Prensa Oe4-50 y Vaca de Castro Quito		No. Autorización 159984132	
SUCURSAL: Luis Plaza Danin 818 y M. Alcivar      Guayaquil			
Sr. (es): .....	DOCUMENTO QUE MODIFICA: FACTURA		
RUC: .....	No. : 002-003-0000230		
Dirección: .....	Fecha de emisión: .....		
Cantidad	Descripción	Valor Unitario	Valor de Venta
Carlos Alberto Bolívar Mora / Impresión Bolívar RUC: 171050142001 / No. Autorización 2540		Subtotal	
		Descuento	
		IVA 0%	
		IVA 12%	
		TOTAL	
Válido para su emisión hasta .....		Original: Adquiriente Copia: Emisor	

Fuente: MATERIAL DE USO DIDÁCTICO PARA ESTUDIANTES DE LA U.T.P DE LOJA, Pág. 104  
Elaborado por: Jenny Chicaiza & Luis Quispe

#### ◆ Notas de Débito

Se utiliza para contabilizar una partida a favor de la empresa y contra un acreedor o cliente. Entre las causas más usuales son:

- Devoluciones en ventas de mercadería
- Error en la facturación

Se utilizan para el cobro de intereses por mora y para recuperar costos y gastos cometidos por el vendedor. Deberán consignar la serie y número de los comprobantes de venta a los cuales se refieren.

FIGURA 3.5 NOTA DE DEBITO

<b>SISTEMAS COMPUTARIZADOS S.A.</b>		<b>RUC 0990561448001</b>	
<b>COMPUSYSTEM</b>		<b>NOTA DE DEBITO</b>	
<b>MATRIZ: Av.de la Prensa Oe4-50 y Vaca de Castro Quito</b>		<b>No. 002-001-0000001</b>	
<b>SUCURSAL: Luis Plaza Danin 818 y M. Alcivar</b>		<b>No. Autorización</b>	
<b>Guayaquil</b>		<b>159984132</b>	
Sr. (es): .....	DOCUMENTO QUE MODIFICA: FACTURA		
RUC: .....	No. : 002-004-0000560		
Dirección: .....	Fecha de emisión: .....		
Cantidad	Descripción	Valor Unitario	Valor de Venta
		Subtotal	
		Descuento	
		IVA 0%	
		IVA 12%	
		TOTAL	
Carlos Alberto Bolívar Mora / Impresión Bolívar			
RUC: 171050142001 / No. Autorización 2540			
Válido para su emisión hasta .....		Original: Adquiriente	
		Copia: Emisor	

Fuente: MATERIAL DE USO DIDÁCTICO PARA ESTUDIANTES DE LA U.T.P DE LOJA, Pág.105

Elaborado por: Jenny Chicaiza & Luis Quispe

### ◆ Comprobante de Retención

Son documentos que acreditan las retenciones de impuestos realizadas por los Agentes de Retención que lo entregarán en el momento que se realiza el pago dentro de los plazos previstos en cumplimiento de lo dispuesto en la Ley de Régimen Tributario Interno.

Este documento sirve como constancia de las retenciones realizadas por los siguientes conceptos: Retención en la fuente del impuesto a la renta, Retención del IVA.

FIGURA 3.6 COMPROBANTES DE RETENCION

Razón Social	← ALMACENES RIO 'S S.A.	RUC: 1790145045001	→ RUC
Nombre Comercial	← RIO CENTER	COMPROBANTE DE RETENCIÓN	→ Denominación
Lugar de emisión y dirección de la matriz	← MATRIZ: 6 de Diciembre N9-02 y Naciones Unidas SUCURSAL: Av. Juan Tanca Marengo	Quito Guayaquil	→ Numeración
Identificación sujeto pasivo retenido	Sr. (es): ..... RUC: ..... Dirección: .....	No. Autorización: 1005841321 Fecha Emisión: ..... Tipo de Comprobante de Venta: ..... No. de Comprobante de Venta: .....	→ Número de autorización → Fecha de emisión → Tipo y No. comprobante de venta
Ejercicio Fiscal	←		→ Valor retenido
Valor de la transacción objeto de la retención	←		→ Porcentaje de la retención
Impuesto	←		
Firma del agente de retención	←		→ Validez
Datos de la imprenta	←		→ Destinatarios

Ejercicio Fiscal	Base imponible para la retención	Impuesto	% de retención	Valor retenido

Válido para su emisión hasta 09/2004

Firma del agente de retención

José Carlos Álvarez Díaz / RUC:09015401420001 /  
No. Autorización 4502

Original: Sujeto pasivo retenido  
Copia: Agente de retención

Fuente: MATERIAL DE USO DIDÁCTICO PARA ESTUDIANTES DE LA U.T.P DE LOJA, Pág. 106

Elaborado por: Jenny Chicaiza & Luis Quispe

### ◆ Cheque

Es una orden o mandato de pago incorporado a un título de crédito que permite al librador disponer, en favor de una determinada persona o al portador, de fondos que tenga disponibles en un banco.

El cheque deberá contener: la denominación de cheque inserta en el texto mismo del título, el mandato puro y simple de pagar una suma determinada de dinero, el nombre del que debe pagar(librador), el lugar de pago, la fecha y el lugar de la emisión del cheque, la firma del que lo expide, al que se denomina librador.

**FIGURA 3.7 CHEQUE**



CHEQUE NRO. _____	BANCO ESTUDIANTIL VENUS	00-001	CTA. CTE
FECHA: _____	COMPANÍA ANÓNIMA	763	CHEQUE NO.
PAGADO A: _____	PÁGUESE A		
CONCEPTO: _____	LA ORDEN DE _____		\$.
SALDO ANTERIOR _____	LA SUMA DE: _____		DÓLARES
DEPÓSITO _____	_____ DEL 200 _____		
SUBTOTAL _____	CIUDAD _____	FECHA _____	
N/D _____			FIRMA _____
CHEQUE _____	NO INVADIR ESTA ZONA CON RASGOS CALIGRÁFICOS NI SELLOS		
SALDO ACTUAL _____			

Fuente: MATERIAL DE USO DIDÁCTICO PARA ESTUDIANTES DE LA U.T.P DE LOJA, Pág. 107

Elaborado por: Jenny Chicaiza & Luis Quispe

### 3.3.2.2 Jornalización

Es el registro y el control de toda transacción mediante el asiento contable en el libro diario.

**Asiento Contable.-** Es la fórmula técnica de anotación de las transacciones, bajo el principio de Partida Doble.

Todo Asiento Contable contiene:

- Fecha
- Cuenta o cuentas deudoras y el valor afectado
- Cuenta o cuentas acreedoras y el valor afectado
- Explicación del asiento y el comprobante que originó la jornalización

En contabilidad se distinguen tres clases de asientos contables:

- **Asiento Simple:** Cuando el registro de la transacción tiene una cuenta deudora y una cuenta acreedora.
- **Asiento compuesto:** Cuando el registro de las operaciones tiene dos o más cuentas deudoras y dos o más cuentas acreedoras.
- **Asiento Mixto:** En el registro de las operaciones tiene una cuenta deudora y dos o más acreedoras o viceversa.

### 3.3.2.2.1 Libro Diario

El libro diario es documento numerado, que le permite registrar o jornalizar en forma cronológica todas las transacciones realizadas por la empresa; en el que se anotan todas las operaciones. Este diario posee la fecha, descripción, dos columnas que sirven para anotar las cantidades, las mismas que son débito y crédito, así como la referencia, en la parte superior derecha se anotará el nombre de la empresa, y el nombre del documento en este caso diario general en la parte superior izquierda el número de página.

Cada página del libro debe enumerarse en orden ascendente, las columnas del rayado se les realiza de la siguiente forma:

FIGURA 3.8 LIBRO DIARIO

**EMPRESA "XYZ"**  
**LIBRO DIARIO**

**FOLIO N° 001**

<b>FECHA</b>	<b>DETALLE</b>	<b>REF.</b>	<b>PARCIAL</b>	<b>DEBE</b>	<b>HABER</b>
	VIENEN				
	PASAN				

Fuente: Zapata, Pedro; Contabilidad General; Cuaderno de Practicas Contables; Cuarta Edición; 2003; Pág. 141  
Elaborado por: Jenny Chicaiza & Luis Quispe

“El procedimiento para anotar las transacciones en libro diario:

- a) En la columna fecha, se escribe el día, mes y año de la transacción, según el documento fuente.
- b) En la columna de detalle, se inicia asignando un número a cada uno de los registros contables. Luego se registra las cuentas deudoras y acreedoras de la transacción, con la correspondiente explicación que debe ser clara y precisa.

- c) En las columnas del debe y haber se escribe los valores respectivos, existiendo siempre igualdad entre la parte deudora y acreedora.
- d) La columna de parcial sirve para anotar los valores de las cuenta auxiliares.”<sup>30</sup>

### 3.3.2.3 Mayorización

Agrupar los valores de cada una de las cuentas del libro diario, con el propósito de conocer su movimiento y saldo en forma particular.

#### 3.3.2.3.1 Libro Mayor

Es la clasificación de las anotaciones o asientos del libro diario, clasificación que se produce en unos estados llamados cuentas, que registran todas las operaciones de idéntica naturaleza o características, con el propósito de conocer su movimiento y saldo en particular.

**FIGURA 3.9 LIBRO MAYOR GENERAL**

**EMPRESA “XYZ”  
LIBRO MAYOR GENERAL**

FECHA	DESCRIPCION	REF.	DEBE	HABER	SALDO
	SUMAN				

Fuente: Zapata, Pedro; Contabilidad General; Cuaderno de Practicas Contables; Cuarta Edición; 2003; Pág. 142.

Elaborado por: Jenny Chicaiza & Luis Quispe

<sup>30</sup> MATERIAL DE USO DIDÁCTICO PARA ESTUDIANTES DE LA U.T.P DE LOJA, Pág. 116

“El procedimiento para mayorizar es:

- a) Tome cuenta por cuenta, las que constan en el libro diario.
- b) A cada cuenta, asígnele una tarjeta (o archivo magnético) y escriba el nombre completo de la cuenta (o auxiliar). Ponga el número del código que le corresponda (según el Plan General o Catalogo de cuentas vigente)
- c) Pase los valores de la cuenta que consta en el diario; si están en el debe, al DEBE; si están en el haber, al HABER de la respectiva cuenta.
- d) Obtenga los saldos así, asigne a los valores del debe el signo (+), y si están en el haber el signo (-), obtenga la diferencia matemática; si el resultado es positivo, se trata de un saldo deudor, por el contrario, si el valor es negativo, se trata de un saldo acreedor.

### **Tipos de Saldos**

- **Saldo Deudor:** Se obtiene cuando los valores debitados en una cuenta son mayores que los valores acreditados. Generalmente, este tipo de saldo es propio de las cuentas de Activo, Costos y Gastos.
- **Saldo Acreedor:** Se obtiene cuando los valores acreditados en una cuenta son mayores que los valores debitados. Generalmente, este tipo de saldo es propio de las cuentas de Pasivo, Patrimonio y Rentas.
- **Saldo Nulo:** Se Obtiene cuando los valores debitados en una cuenta son exactamente iguales. Este tipo de saldo es excepcional; se puede dar en cuentas de Activo, Pasivo y Patrimonio, pero momentáneamente, y en las cuentas de Costos, Gastos y Rentas, al final del ejercicio a través de los asientos de cierre.”<sup>31</sup>

---

<sup>31</sup> Zapata, Pedro; Contabilidad General; Cuarta Edición; 2003; Pág. 51.

### 3.3.2.4 Balance de Comprobación

Listado y verificación de todas las cuentas del libro mayor con sus respectivas sumas y saldos. Es trasladar sistemáticamente de manera clasificada los valores que se encuentran en cada cuenta que han intervenido en un ejercicio económico

El propósito del balance de comprobación es comprobar la eficacia de la aplicación de la partida doble y permitir una verificación de la calidad de trabajo en la transacción, la contabilización y la mayorización. El balance de comprobación se lo realiza en el último mes del año con la información de los saldos.

Su formato es el siguiente:

**FIGURA 3.10 BALANCE DE COMPROBACIÓN DE SUMAS Y SALDOS**

**EMPRESA "XYZ"**  
**BALANCE DE COMPROBACIÓN DE SUMAS Y SALDOS**  
**AL.....**

Nº	CUENTAS	SUMAS		SALDOS	
		DEBE	HABER	DEUDOR	ACREEDOR
	SUMAN				

Fuente: Zapata, Pedro; Contabilidad General; Cuarta Edición; 2003; Pág. 118

Elaborado por: Jenny Chicaiza & Luis Quispe

“Procedimiento para la elaboración del balance de comprobación:

- a) Según consta en cada libro mayor, se traslada el código de las cuentas a nivel de mayor con su respectivo nombre.
- b) Cada libro tiene que estar totalizado las columnas del debe y haber, pasando dichos valores a la columna de sumas con debe y haber, respectivamente.
- c) En la columna de saldos deudor y acreedor se presenta el saldo correspondiente.
- d) Hay ocasiones que las cuentas del libro mayor han quedado con saldo cero, sin embargo se las incluye en el balance de comprobación de sumas y saldos.
- e) Una vez que se traslado la información de todas las cuentas contables, se procede a totalizar la columna de sumas con debe y haber. Una manera de verificar la exactitud de balance de comprobación es que la sumatoria de las dos primeras columnas sea igual al total del libro diario; en los saldos deudor y acreedor también tiene que existir igualdad.”<sup>32</sup>

### **3.3.2.5 Hoja de Trabajo**

Es un medio auxiliar y didáctico que permite resumir en un solo informe gran parte del proceso contable. El mismo que contiene:

#### **1. Balance de Comprobación**

---

<sup>32</sup> MATERIAL DE USO DIDÁCTICO PARA ESTUDIANTES DE LA U.T.P DE LOJA, Pág. 118

2. Estado de Resultados.
3. Estado de Situación.

Es decir que la hoja de trabajo es un esquema que presenta el conjunto de Balances necesarios para cerrar un ejercicio económico.

**FIGURA 3.11 HOJA DE TRABAJO**

**EMPRESA “XYZ”**  
**HOJA DE TRABAJO**  
**DEL.....**

Nº	CUENTAS	SUMAS		SALDOS		ESTADO DE RESULTADOS		ESTADO DE SITUACION	
		DEBE	HABER	DEUDOR	ACREEDOR	GASTOS	INGRESOS	ACTIVO	PASIVO
	TOTALES								
	GANANCIA LIQUIDA								
	TOTALES								

Fuente: Apuntes de Procesos Contables, 2003.

Elaborado por: Jenny Chicaiza & Luis Quispe

“Procedimiento para la elaboración de la hoja de trabajo:

- Encabezamiento formado por:
  - Nombre o razón social de la empresa
  - Nombre del documento contable, hoja de trabajo
  - Período al que corresponde la hoja de trabajo
- Registro en el balance de comprobación:

- Nombres y valores de las cuentas del débito
  - Nombres y valores de las cuentas del crédito
  - Sumas iguales del débito y el crédito del balance
- Traslado de los valores de las cuentas registradas en el débito del balance de comprobación, al débito del estado de resultados o al débito del estado de situación.
  - Traslado de los valores de las cuentas registradas en el crédito del balance de comprobación, al crédito del estado de resultados o al crédito del estado de situación.
  - Sumar el débito y el crédito de las columnas del estado de resultados y determinar la pérdida o ganancia neta de la empresa y trasladar al débito o crédito del estado de situación, según la naturaleza del estado.
  - Sumar las columnas del débito y del crédito del estado de situación, para comprobar la correcta realización de la hoja de trabajo.”<sup>33</sup>

### **3.3.2.6 Estados Financieros**

Son informes que elaboran al finalizar un período contable con el propósito de demostrar los cambios o variaciones de la situación financiera de una empresa, ocurridos en un ejercicio económico de conformidad con los principios de contabilidad vigentes en el país.

#### **Los Estados Financieros nos ayudan a:**

- Tomar decisiones de inversión y crédito.
- Evaluar la gestión gerencial, la solvencia y la liquides de la empresa, además su calidad para generar fondos.

---

<sup>33</sup> Vásconez, José; Introducción a la Contabilidad; Pág. 94



- Conocer origen y las características de sus recursos para estimar la capacidad financiera crecimiento.
- Formular un juicio sobre los resultados financieros de la administración en cuanto a la rentabilidad, secuencia, generación de fondos y capacidad de crecimiento.

### **Clases de Estados Financieros**

Existen dos clases de Estados Financieros:

- **Los que piden la situación económica:**
  - Estado de Resultados o de Perdidas y Ganancias.
  - Estado de Ganancias Retenidas o de Superávit.
- **Los que miden la situación financiera:**
  - Balance General o Estado de Situación Financiera.
  - Estado de Flujo de Efectivo

#### **3.3.2.6.1 Estados de Resultados**

Establecer el resultado económico (Pérdida y Ganancias o de Resultados) del ejercicio contable; con el propósito de demostrar si existe utilidad o pérdida. Este muestra el resultado de las operaciones (utilidad, pérdida remanente y excedente) de una entidad durante un periodo determinado.

Presenta la situación financiera de una empresa a una fecha determinada, tomando como parámetro los ingresos y gastos efectuados; proporciona la utilidad neta de la empresa. Generalmente acompaña a la hoja del Balance General.

“Partes y formas de presentación del estado de resultados

- **Encabezamiento.-** Contendrá nombre de la empresa, nombre del estado, período al que corresponde y denominación de la moneda en que se presenta.

- **Texto.-** Parte sustancial que presenta todas las cuentas de rentas, costos y gastos debidamente clasificadas y ordenadas, a fin de ir estableciendo los tipos de utilidad o pérdida.
- **Firmas de legalización.-** En la parte inferior se insertan la firma y rúbrica del gerente y del contador.”<sup>34</sup>

**FIGURA 3.12 ESTADO DE RESULTADOS**

**EMPRESA “XYZ”  
ESTADO DE RESULTADOS – US\$  
DEL..... AL.....**

Ventas	XXX		
(-) Devolución en ventas	XXX		
(-) Descuento en ventas	<u>XXX</u>		
=Ventas netas			XXX
(-)Costo de ventas			- <u>XXX</u>
Mercadería inventario inicial		XXX	
+ Compras	XXX		
(-) Devolución en compras	XXX		
(-) Descuento en compras	<u>XXX</u>		
Compras netas		<u>XXX</u>	
Mercadería disponible para la venta		XXX	
(-) Mercadería Inventario final		- <u>XXX</u>	
Utilidad bruta en ventas			XXX
(-) Gastos de administración		XXX	
(-) Gastos de venta		<u>XXX</u>	

<sup>34</sup> Zapata, Pedro; Contabilidad General; Cuarta Edición; 2003; Pág. 71.

Total gastos de operación		<u>XXX</u>
Utilidad operacional		XXX
+ Otros ingresos	XXX	
(-) Otros gastos	<u>XXX</u>	<u>XXX</u>
Utilidad del ejercicio		<u>XXX</u>
	(Firma)	(Firma)
	Gerente	Contador

Fuente: Zapata, Pedro; Contabilidad General; Cuarta Edición; 2003; Pág. 118

Elaborado por: Jenny Chicaiza & Luis Quispe

### 3.3.2.6.2 Balance General o de Situación Financiera.

Demuestra la situación económica de la empresa al cierre del ejercicio contable; o al inicio según el caso. Tiene por objeto rendir un claro y preciso informe a las partes interesadas en la empresa sobre la situación de la misma al final de un año comercial. Se prepara con las cuentas reales del balance comprobación.

“Partes y formas de presentación del estado de situación financiera:

- **Encabezamiento.-** Contendrá nombre de la empresa, nombre del estado, la fecha de corte o cierre del Balance y denominación de la moneda en que se presenta.
- **Texto.-** Es la parte esencial que presenta las cuentas de Activo, Pasivo y Patrimonio, distribuidas de tal manera que permitan efectuar un análisis financiero certero y objetivo.
- **Firmas de legalización.-** En la parte inferior del Estado se deberán insertar la firma y rúbrica del gerente y del contador.”<sup>35</sup>

<sup>35</sup> Zapata, Pedro; Contabilidad General; Cuarta Edición; 2003; Pág. 72.

**FIGURA 3.13 ESTADO DE SITUACION**

<b>EMPRESA "XYZ"</b>			
<b>BALANCE GENERAL</b>			
<b>AL..... DE .....DE .....</b>			
<b>ACTIVO</b>		<b>PASIVO</b>	
<b>Corriente</b>		<b>Corriente</b>	
Caja	XXX	Cuentas por pagar	XXX
Caja Chica	XXX	Documentos por pagar	XXX
Bancos	XXX	Sueldos por pagar	<u>XXX</u>
Mercaderías	XXX	Total pasivo corriente	XXX
Cuentas por cobrar	<u>XXX</u>		
Total activo corriente	XXX		
<b>Propiedad, planta y equipo</b>		<b>A largo plazo</b>	
Muebles de oficina	XXX	Préstamos bancarios	XXX XXX
(-)Deprec. Acum. de muebles de of.	(XXX)	Hipotecas por pagar	<u>XXX</u>
Equipo de oficina	XXX	Total pasivo a largo plazo	
(-) Deprec. acum. De eq. De OF.	(XXX)		
Total propiedad, planta y equipo	XXX		
<b>Otros activos</b>		<b>PATRIMONIO</b>	
Gastos de constitución	XXX	Capital	XXX <u>XXX</u>
Inversiones financieras largo plazo	<u>XXX</u>	Utilidad del ejercicio	<u>XXX</u>
Total otros activos	<u>XXX</u>	Total patrimonio	
<b>TOTAL ACTIVO</b>	<u>XXX</u>	<b>TOTAL PASIVO Y</b>	<u>XXX</u>
		<b>PATRIMONIO</b>	
(Firma)		(Firma)	
<b>Gerente</b>		<b>Gerente</b>	

Fuente: Apuntes de Procesos Contables, 2003.

Elaborado por: Jenny Chicaiza & Luis Quispe

### 3.3.2.6.3 Estado de Flujo de Efectivo

Es un informe que incluye las entradas y salidas de efectivo para así determinar el saldo final o el flujo neto de efectivo, factor decisivo para evaluar la liquidez de un negocio.

El propósito del Estado de Flujo de Efectivo es mostrar la información pertinente sobre los cobros y pagos de efectivo de una empresa durante un período. Este es un estado financiero básico que junto con el balance general y el estado de resultados proporcionan información acerca de la situación financiera de un negocio. Evaluando la habilidad de la empresa para cumplir con sus necesidades de financiamiento; evaluar las razones de la diferencia entre las utilidades netas y los cobros y pagos de efectivo asociado.

Los estados financieros deben reflejar una información financiera que ayude al usuario a evaluar, valorar, predecir o confirmar el rendimiento de una inversión y el nivel percibido de riesgo implícito.

“El Estado de flujo del efectivo contiene:

- **Encabezado:** Razón social, nombre del estado y período al que corresponde
- **Texto:** Parte esencial del estado; reflejará el movimiento del efectivo que ingresa y egresa.
- **Firmas de legalización:** debe aparecer la firma del contador que está obligado a preparar este estado, y la del gerente, que avala el trabajo del trabajo del profesional contable.”<sup>36</sup>

(ANEXO 3.1)

---

<sup>36</sup> Zapata, Pedro; Contabilidad General; Cuarta Edición; 2003; Pág. 304

#### **3.3.2.6.4 Estados de Ganancias o de Superávit**

Demuestra las variaciones de las utilidades retenidas de años anteriores más o menos las utilidades o pérdidas del presente año, se elabora básicamente para conocimiento de los socios o accionistas de la empresa.

Se trata de un estado financiero cuya finalidad es exhibir los cambios que ocurren en la cuenta Superávit – Ganancias retenidas, durante un periodo contable.<sup>37</sup>

El Estado de ganancias retenidas es el informa que demuestra los cambios ocurridos en las utilidades de ejercicios anteriores que no han sido entregadas o distribuidas a socios y accionistas en el ejercicio presente.<sup>38</sup>

Las causas que se observan en la cuenta de superávit son:

- Ganancia o pérdida neta de ejercicios anteriores.
- Ganancia o pérdida neta del presente ejercicio.
- Dividendos pagados.
- Dividendos de capital.

(ANEXO 3.2)

#### **3.3.2.7 Análisis Financiero**

Es la evaluación y diagnóstico económico-financiero de la empresa, mediante la relación de ciertos índices que permite analizar con objetividad el

---

<sup>37</sup> Kohler, Eric; Diccionario para contadores; Uteha; 1976

<sup>38</sup> Hargadon, Bernard; Principios de contabilidad; Norma; 1996

comportamiento de las cuentas que interesa conocer, para la toma oportuna de decisiones.

El análisis de estados financieros es el conjunto de procedimientos que permite evaluar e interpretar objetivamente la información contable que consta en los estados financieros, cuyos resultados facilitarán la toma de decisiones futuras.

El análisis debe ser sistemático y periódico, es decir, ha de responder a objetivos preestablecidos; por lo tanto, es conveniente definir ciertas fechas para efectuar el “diagnóstico” del “estado de salud” económica, financiera y administrativa de la empresa y, en vista de que la base del estudio lo constituyen los estados financieros, es recomendable que el análisis se realice en las fechas de elaboración de estados financieros. Luego de un primer análisis, convendrá efectuar tantos cuantos sean necesarios.

El análisis financiero permite obtener información trascendente para determinar la capacidad que tiene la empresa para cumplir con sus obligaciones de pago, relacionando el activo según su liquidez, con el pasivo según su vencimiento.

### **3.4 MARCO LEGAL Y TÉCNICO DEL CICLO CONTABLE**

#### **3.4.1 NORMAS INTERNACIONALES Y ECUATORIANAS DE CONTABILIDAD, NIC YNEC**

(ANEXO 3.3)

### **3.4.2 DL. 825 LEY SOBRE IMPUESTO A LAS VENTAS Y SERVICIOS**

(ANEXO 3.4)

### **3.4.3 EL IVA EN LAS COMPRAS Y VENTAS**

(ANEXO 3.5)

## **3.5 CONVERSION**

### **3.5.1 DEFINICION**

La “conversión” implica transformar, re-expresar o trasladar los estados contables de moneda de un país a otro, como si todas las operaciones ocurridas en moneda original, hubieran ocurrido en la moneda de conversión, en el país de origen.

Una conversión es una reexpresión monetaria en la cual:

- El objeto (lo que se expresa) es una medición en moneda de origen;
- El producto es una medición (equivalente a la anterior) en una moneda de conversión;
- La segunda medición se obtiene a partir de la primera y de algún tipo de cambio que indica la cantidad de unidades de la moneda de origen por las que se puede cambiar una unidad de la moneda de conversión, o viceversa.

Cuando lo que se convierte es el conjunto de datos que integra un juego de estados contables, se habla de conversión de estados contables.

### **3.5.2 OBJETIVO DE LA CONVERSION**

Fowler Newton establece que las conversiones de datos aislados son imprescindibles cuando en una contabilidad llevada en una moneda deben reconocerse los efectos de compras, ventas, pagos y cobranzas en otras monedas; y los efectos de los cambios en los valores de activos y pasivos en esas monedas. También es necesaria la conversión cuando se desea llevar la contabilidad en más de una moneda.



Con respecto a estados contables, la conversión de los mismos a otra moneda puede obedecer los siguientes propósitos:

- Obtener información para la toma de decisiones en una moneda distinta de la local.
- Brindar información a accionistas del exterior que se sienten más cómodos usando datos expresados en la moneda de su propio país o en otra de uso universal.
- Para que una sociedad que prepara estados contables en otra moneda y posee acciones del emisor de los estados contables pueda consolidar o combinar los estados contables de la emisora o aplicar el método del Valor Patrimonial Proporcional.

### **3.5.3 TIPOS DE CAMBIO A EMPLEAR**

#### **Para las conversiones de datos a otra moneda**

Hay acuerdo en que para estas conversiones se utilice el tipo de cambio correspondiente al mercado al que deba recurrirse para pagar dividendos. Esto considera el punto de vista de los inversores extranjeros, que son las personas en cuyo interés suelen efectuarse las conversiones de estados contables.

Por otra parte, el tipo de cambio a emplear debería ser el de la fecha en cuyo poder adquisitivo está expresada la medición en moneda de origen, pues la conversión solo debería apuntar al cambio de unidad de medida. Por lo tanto, antes de cualquier conversión deberían excluirse los efectos de los

ajustes por inflación que hayan incidido sobre las mediciones en moneda de origen a convertir.

Como en el ajuste por inflación, debe tenerse un cuidado especial con las partidas en moneda de origen que resultan de comparar dos o más mediciones expresadas en distinto poder adquisitivo definido, como las correspondientes a los resultados de tenencia (financieros y otros).

En estos casos, lo que debe hacerse es:

- a) convertir cada una de las mediciones comparadas;
- b) volver a efectuar la comparación utilizando las mediciones convertidas.

# **CAPITULO IV**

## **IV. APLICACIÓN DE LA METODOLOGÍA MIDAS EN EL DESARROLLO DEL SISTEMA CONTABLE Y FINANCIERO ONLINE PARAMETRIZABLE “SCFONLINE”**

En este capítulo se describen los pasos para el desarrollo de la Aplicación del Sistema Integrado Parametrizable Contable y Financiero OnLine para empresas comerciales y de servicios (SCFONLINE) basado en la Metodología MIDAS (Metodología para el desarrollo de sistemas de información en la Web), describiendo el proceso de análisis de Especificación de Requisitos, de diseño utilizando UML, con la implementación de SCFONLINE.

## **4.1 FASE DE DEFINICIÓN MIDAS/UR**

### **4.1.1 ESPECIFICACION DE LOS REQUISITOS DE SOFTWARE**

#### **4.1.1.1 Introducción**

Este documento es la Especificación de Requerimientos de Software (ERS) del Sistema (SCFONLINE), es un prototipo unificado de varios procesos contables que se realizan los principios de contabilidad mediante la journalización para generar así los reportes tales como el libro diario, mayores, balance de comprobación, estados resultados, estados situación, al trabajar con módulos tales como de facturación, tesorería que genera asientos contables automáticamente.

Esta especificación se ha estructurado basándose en directrices dadas por el estándar "IEEE Recommended Practice for Software Requirements Specification ANSI/IEEE 830 1998".

#### **4.1.2 PROPÓSITO**

El objeto de la especificación es definir de manera clara y concisa las funcionalidades y restricciones del sistema (SCFONLINE). Este documento va dirigido al personal de desarrollo, a las personas que lo pondrán a prueba y a los usuarios finales.

Esta especificación está sujeta a revisiones y modificaciones por parte de las partes implicadas, de acuerdo a sus necesidades, hasta lograr la aprobación por todas las partes. Una vez aprobado servirá de base al equipo de desarrollo para la construcción del sistema.

#### **4.1.3 AMBITO DEL SISTEMA**

La futura aplicación recibirá el nombre de "SCFONLINE", abreviatura de Sistema Integrado parametrizable Contable y Financiero Online, el mismo que

permitirá tener todos los procesos de una empresa tanto de productos como de servicios conteniendo los módulos de Contabilidad, Facturación, Tesorería, Inventarios. En sí, el objetivo del desarrollo del sistema es generar actualizaciones inteligentes y completamente automatizados los procesos del sistema.

El sistema gestiona aplicaciones independientes para labores administrativas y de usuario cliente. Cada una de estas aplicaciones debe comunicarse con el resto del sistema a través de conexiones independientes y excluyentes.

Permite el ingreso a usuarios, los mismos que tienen perfiles de acuerdo al cargo que estos tengan, al igual que se ingresa empresas para poder interactuar con estas en forma individual o consolidada.

El sistema debe ofrecer funcionalidades relacionadas e integradas con los siguientes módulos:

- **Módulo Contabilidad.-** El modulo de Contabilidad registra los hechos (jornalización) económicos de la gestión financiera de la empresa. Realiza un control exhaustivo de ingreso de datos que garantiza la consistencia de los movimientos que se registran; configurando tipo de moneda, plan contable, tipo de comprobante y periodos contables. Además le permitirá realizar asientos contables en diferentes monedas pero mayorizarlos en la moneda origen, generando reportes como: libro diario, mayores, balance de comprobación, estado de resultados, estado de situación y gastos fijos.
- **Módulo de Inventario.-** El modulo de Inventarios permite el ingreso del inventario inicial, transferencias, cálculo de depreciación; familia de activos, porcentajes de retención e ingreso de Activos; generando Kardex que permite identificar y controlar cada uno de los activos de la empresa, registrando sus características, ubicación y responsable.
- **Módulo Facturación.-** El modulo de Facturación es de Compras como Ventas le permite realizar facturas, notas de crédito de servicios, inventarios y automáticamente enlazarse con los módulos de tesorería para

la emisión de cheques y realizar al mismo tiempo los asientos contables en el módulo Contable; generando el reporte de cartera vencida, vigente y por vencer

- **Módulo de Tesorería.-** El modulo de Tesorería realiza automáticamente los respectivos asientos en el módulo de Contabilidad, actualizando los saldos de las cuentas contables (Cuentas impuestos, facturas por pagar, cuentas proveedores, reporte de impuestos).

Además en este módulo se ingresa bancos, y mediante el flujo de caja se realiza cheques los mismos que interactúan con el módulo de facturación; a la vez el mismo que genera reporte de los mismos.

- **Módulo de Cuentas por Cobrar.-** El modulo de Cuentas por Cobrar emite un reporte de pago de los diversos servicios y productos que se ofrece, esta orden será impresa.

En el momento en que el contribuyente realice efectivamente el pago o el cobro, el Módulo de Tesorería realiza automáticamente los respectivos asientos en el módulo de Contabilidad, actualizando los saldos de las cuentas contables en línea (Ingreso Devengado e Ingreso Percibido y Egreso pagado).

- **Modulo de Cliente.-** El modulo de Clientes se ingresa los datos de clientes y proveedores de la empresa que genera reportes de los mismos Clientes y Proveedores interactúa directamente con el modulo de facturación, cuentas por cobrar y tesorería.

#### 4.1.4 DEFINICIONES, ACRONIMOS Y ABREVIATURAS

A continuación se definen las definiciones, acrónimos y abreviaturas que utilizaremos en el ERS.

##### 4.1.4.1 Definiciones

Administrador Inicial	Persona encargada de inicializar los parámetros necesarios para el ingreso al sistema por usuarios.
Administrador	Persona encargada de la administración completa del Sistema
Auxiliar Contable	Persona encargada del análisis del sistema contable
Gerente	Persona encargada del análisis de los reportes generados en el sistema contable
Usuario	Persona encargada del ingreso de la información que alimentara al sistema contable

**Tabla 4. 1 Definiciones**

##### 4.1.4.2 Acrónimos

ERS	Especificación de Requisitos de Software
-----	--

**Tabla 4. 2 Acrónimos**

##### 4.1.4.3 Abreviaturas

SCFOline	Sistema Integrado parametrizable Contable y Financiero OnLine para empresas comerciales y de servicios (SCFONLINE).
----------	---

**Tabla 4. 3 Abreviaturas**

#### **4.1.5 REFERENCIAS**

IEEE Recommended Practice for Software Requirements Specification. ANSI/IEEE std 830, 1998.

##### **4.1.5.1 Visión General del Documento**

Este documento consta de tres secciones. Esta sección es la Introducción y proporciona una visión general de la ERS del sistema SCFONLINE. En la Sección 4.2 se da una descripción general del sistema, con el fin de conocer las principales funciones que debe realizar SCFONLINE, los datos asociados, los factores, restricciones, supuestos y dependencias que afectan al desarrollo, sin entrar en mayor detalle.

En la sección 4.3 se definen detalladamente los requisitos que debe satisfacer el sistema SCFONLINE.

#### **4.1.1 DESCRIPCIÓN GENERAL**

Esta sección nos permite conocer de una manera general las funciones principales que el sistema debe realizar.

##### **4.1.1.1 Perspectiva del producto**

El sistema SCFONLINE interactuará entre los diferentes departamentos al estar integrado en un solo paquete, generando de manera automáticamente reportes contables que interactúan entre los diferentes módulos del sistemas, evitando la molestia de que los usuarios ingresen a diferentes sistemas y la creación de varias interfaces para obtener un mismo resultado.

#### **4.2 FUNCIONES DEL SISTEMA**

En términos generales, SCFONLINE deberá proporcionar soporte a las siguientes tareas:

#### **4.2.1 GESTIÓN ADMINISTRACION**

Permite realizar las funciones de ingreso, eliminación, modificación, consulta individual o general; recalando que en la eliminación y modificación se podrán hacer solo en campos que el sistema permita, y no tengan datos relacionados con ellos.

El primer usuario que interactúa con el sistema se le llamara Administrador, es la persona encargada de verificar que la configuración inicial de las variables globales del sistema este de acuerdo al tipo de empresa a crear, cabe indicar que este proceso será realizado de forma manual. En el caso de que los datos de la Empresa sean diferentes por ejemplo se implementen en diferentes países deberá crearse los usuarios del sistema y modificarse cada uno de los parámetros iniciales.

#### **4.2.2 GESTIÓN CONTABILIDAD**

Permite realizar las funciones de ingreso, eliminación, modificación, consulta individual o general(reportes); recalando que en la eliminación y modificación se podrán hacer solo en aquellos campos que el sistema permita, y no se encuentren relacionados con otros datos.

En la contabilidad se registrara todas las transacciones financieras de la empresa como facturas, notas de venta, notas de crédito, pagos realizados, retenciones, emisión de cheques, depósitos mediante la jornalización con su respectivo detalle, que se mayoriza y automáticamente se genera los reportes: libro diario, mayores tanto auxiliar como general, balance de comprobación, estado de resultados, estado de situación y gastos fijos para lo cual se debe ingresar una fecha inicial y una fecha final para visualizar el reporte; cabe recalcar que para poder jornalizar debemos configurar el tipo de moneda con el



que vamos a realizar los asientos contables, tener un plan de cuenta, definir los tipos de comprobantes que vamos a utilizar en la empresa, y abrir un periodo contable, que por lo general se realiza cada inicio de mes y se cierra cada fin de mes, dependiendo el tipo de contabilidad que realiza la empresa, al cerrar el periodo se define a que cuenta se pasa la utilidad de la empresa en el siguiente periodo.

#### **4.2.3 GESTIÓN CLIENTES**

Permite realizar las funciones de ingreso, eliminación, modificación, consulta individual o general; recalcando que el proceso de eliminación y modificación se lo podrá realizar si no se encuentran los datos relacionados.

#### **4.2.4 GESTIÓN INVENTARIOS**

Permite realizar las funciones de ingreso, eliminación, modificación, consulta individual o general; recalcando que en la eliminación y modificación se podrán hacer solo en aquellos campos que el sistema permita, y no se encuentren relacionados con otros datos.

En inventarios se registra el inventario inicial de la empresa, los consumos ajustes, transferencias y calculo de depreciación; mediante el kardex que se genera automáticamente, podemos observar los movimientos realizados por los activos que posee la empresa tanto fijos como corrientes; antes de registrarlos se debe configurar los parámetros de activos, familias de activos y porcentajes de retención.

#### **4.2.5 GESTIÓN FACTURACION**

Permite realizar las funciones de ingreso, eliminación, modificación, consulta individual o general(reportes); en la eliminación y modificación se podrán hacer siempre y cuando no se encuentre aprobada la factura.

Este módulo permite registrar las facturas de compra y venta de productos y de servicios, para realizar una factura de compra antes se debe realizar una solicitud de pedido, la misma que debe estar aprobada para proceder a la facturación de compras; la facturación interactúa con el módulo de clientes, cuentas por pagar, contabilidad, tesorería; los mismos que generan los reportes correspondientes en cada uno de estos módulos de una manera automática.

#### **4.2.6 GESTIÓN TESORERIA**

Permite realizar las funciones de ingreso, eliminación, modificación, consulta individual o general de bancos. Este módulo permite realizar el pago de facturas aprobadas mediante la emisión de los cheques, adicionalmente podrá anular algunos pagos, visualizar saldos, cuentas por pagar; interactuando con el módulo de contabilidad y facturación.

#### **4.2.7 GESTIÓN CUENTAS POR COBRAR**

Permite realizar las funciones de ingreso, eliminación, modificación, consulta de clientes que tengan cuentas pendientes de cobro en la empresa; con el módulo de contabilidad y facturación.

##### **4.2.7.1 Requisitos no Funcionales**

- SCFONLINE, permite el control de concurrencia de múltiples clientes a las Bases de Datos garantizando la integridad de la información.
- SCFONLINE, ofrece un mecanismo de autenticación para los (claves de acceso).
- SCFONLINE, administra aplicaciones independientes para labores administrativas y de usuario.

- SCFONLINE, esta disponible las 24 horas del día los 7 días de la semana.
- El 95% de las transacciones deberán ser procesadas en menos de 5 segundos.
- La interfaz de usuario deber ser orientada a ventanas y el manejo de SCFONLINE debe ser a través de teclado y mouse
- SCFONLINE, no genera gráficos estadísticos
- SCFONLINE, genera automáticamente la mayorización de las transacciones del módulo de facturación y tesorería
- SCFONLINE, genera automáticamente el reporte de diarios, mayores, balance de comprobación, estado de situación y de gastos fijos.
- SCFONLINE, genera y actualiza automáticamente el kardex después de realizada una transacción.

## **4.2.8 SUPOSICIONES Y DEPENDENCIAS**

### **4.2.8.1 Suposiciones**

Se asume que los datos Financieros ingresados por el usuario son exactos y verdaderos, de tal manera que los gerentes obtengan datos confiables y reales de los estados y situación financiera de la empresa, por lo tanto la responsabilidad sobre la credibilidad de la información ingresada al sistema es del Usuario.

### **4.2.8.2 Dependencias**

SCFONLINE funciona autónomamente sin la necesidad de otros sistemas.

## **4.3 REQUISITOS ESPECÍFICOS**

En este apartado se presentan los requisitos funcionales que deberán ser satisfechos por SCFONLINE. Todos los requisitos aquí expuestos son esenciales, para el buen y correcto funcionamiento del sistema.

#### **4.3.1 GESTIÓN CONFIGURACION INICIAL**

Req(1) Ingreso a SCFONLINE desde el sitio web (Intranet).

Req(2) Ingreso por primera vez a SCFONLINE, con el login "Adm" y el password "admin".

#### **4.3.2 GESTIÓN ADMINISTRADOR INICIAL**

Los usuarios que pueden realizar la administración inicial son el administrador inicial y el administrador.

##### **4.3.2.1 GESTION USUARIOS**

Req(3) Ingreso de datos del usuarios con la siguiente información: Nombre, Apellido, Dirección, Teléfono, E-mail, Observaciones, Login (max 12), Password, Confirme Password, perfil (Administrador, Auxiliar Contable, Gerente y Usuario), Tipo(Activo, Inactivo).

Req(4) Verificación de los datos ingresados en el Req(3), con presentación de mensajes de datos correctos, si ingreso todos los datos obligatorios; datos incorrectos si esta incompleto.

Req(5) Control de e-mail y confirmación de password

Req(6) Eliminación de usuario que no tenga relación con otros datos, notificando con mensajes la actividad realizada.

Req(7) Actualización de campos de usuarios notificando con mensajes la actividad realizada.

Req(8) Buscar usuarios por código o nombre.

Req(9) Imprimir usuarios en forma particular o general

#### **4.3.2.2 GESTION EMPRESAS**

Req(10) Ingreso de datos de empresas con la siguiente información: Código, Nombre, Descripción, Zona, # de cuenta, Fecha de apertura, Estado(Activo o Inactivo), Consolidado (Si o NO).

Req(11) Verificación de los datos ingresados en el Req(10), con presentación de mensajes de datos correctos, si ingreso todos los datos obligatorios; datos incorrectos si esta incompleto.

Req(12) Eliminación de empresa que no tenga relación con otros datos, notificando con mensajes la actividad realizada.

Req(13) Actualización de campos de empresa notificando con mensajes la actividad realizada.

Req(14) Buscar empresas por código o nombre.

Req(15) Imprimir empresas en forma particular o general

#### **4.3.2.3 GESTION VARIABLES GLOBALES**

Req(16) Ingreso de datos de variables globales con la siguiente información: Variable, Tipo, Valor, Id Compañía, Descripción.

#### **4.3.3 GESTIÓN ADMINISTRADOR**

Req(17) Ingreso a SCFONLINE, a través del login y password.

Req(18) Verificación los datos ingresados en el Req(17), con presentación de mensajes de datos incorrectos, si ingreso datos erróneos.

#### **4.3.3.1 GESTION CONTABILIDAD**

##### **4.3.3.1.1 GESTION CONFIGURACIÓN**

Los usuarios que pueden realizar las configuraciones en el modulo de contabilidad son el administrador y el auxiliar contable.

#### **4.3.3.1.2 GESTION MONEDA**

Req(19) Ingreso de datos de monedas con la siguiente información:  
Nombre, Moneda local (Sí) , Símbolo, TC Compra y TC Venta.

Req(20) Verificación de los datos ingresados en el Req(19), con presentación de mensajes de datos correctos, si ingreso todos los datos obligatorios; datos incorrectos si esta incompleto.

Req(21) Eliminación de moneda que no tenga relación con otros datos, notificando con mensajes la actividad realizada.

Req(22) Actualización de campos de moneda notificando con mensajes la actividad realizada.

Req(23) Imprimir tipos de moneda en forma particular o general

#### **4.3.3.1.3 GESTION PLAN CONTABLE**

Req(24) Ingreso del Plan Contable con la siguiente información: Código, Nombre, Tipo (Acumulación, Movimiento).

Req(25) Verificación de los datos ingresados en el Req(24), con presentación de mensajes de datos correctos, si ingreso todos los datos obligatorios; datos incorrectos si esta incompleto.

Req(26) Eliminación de cuentas contables que no tenga relación con otros datos, notificando con mensajes la actividad realizada.

Req(27) Actualización de campos de una cuenta contable notificando con mensajes la actividad realizada.

Req(28) Buscar cuentas contables por código o nombre.

Req(29) Imprimir plan contable en forma general .

#### **4.3.3.1.4 GESTION TIPO DE COMPROBANTE**

Req(30) Ingreso de Tipo Comprobante con la siguiente información:  
Código, Nombre, Numeración Actual.

Req(31) Verificación de los datos ingresados en el Req(30), con presentación de mensajes de datos correctos, si ingreso todos los datos obligatorios; datos incorrectos si esta incompleto.

Req(32) Eliminación de tipo de comprobante que no tenga relación con otros datos, notificando con mensajes la actividad realizada.

Req(33) Actualización de campos de un tipo de comprobante notificando con mensajes la actividad realizada.

Req(34) Buscar tipo de comprobantes por código o nombre.

Req(35) Imprimir tipo comprobante en forma particular o general .

#### **4.3.3.1.5 GESTION PERIODOS CONTABLES**

Req(36) Ingreso del Periodo Contable con la siguiente información: la fecha de inicio y la fecha de fin.

Req(37) Verificación de los datos ingresados en el Req(36), con presentación de mensajes de datos correctos, si ingreso todos los datos obligatorios; datos incorrectos si esta incompleto.

Req(38) Eliminación de un periodo contable que no tenga relación con otros datos, notificando con mensajes la actividad realizada.

Req(39) Actualización de fecha de un período contable notificando con mensajes la actividad realizada.

Req(40) Buscar período contable por fecha.

Req(41) Imprimir periodos contables en forma general .

#### **4.3.3.1.6 GESTION PROCESOS**

##### **4.3.3.1.6.1 GESTION JORNALIZACION**

Los usuarios que pueden realizar las configuraciones en el modulo de contabilidad son el administrador y el auxiliar contable.

Req(42) Ingreso de un Asiento Contable con la siguiente información: Tipo de comprobante, comprobante No. fecha comprobante, moneda, valor original, valor moneda Local, concepto, estado, generado.

Req(43) Verificación de los datos ingresados en el Req(42), con presentación de mensajes de datos correctos, si ingreso todos los datos obligatorios; datos incorrectos si esta incompleto.

Req(44) Una vez cumplido con el Req(43), ingreso de detalle con la siguiente información: cuenta contable, valor en el debe o haber.

Req(45) Verificación de los datos ingresados en el Req(44), con presentación de mensajes de saldos incorrectos, si ingreso valor incorrecto.

Req(46) Mayorización de asiento contable, con presentación de mensajes de estado mayorizado o no mayorizado.

Req(47) Distribución de Comprobante con la siguiente información: escoger el centro de costos, el valor que distribuye en el centro de costos y la descripción.

Req(48) Verificación de los datos ingresados en el Req(47), con presentación de mensajes de datos correctos, si ingreso todos los datos obligatorios; datos incorrectos si esta incompleto.

Req(49) Buscar asientos contables por código.

Req(50) Imprimir asientos contables en forma particular o general.



Req(51) Eliminación de asiento contable que no tenga relación con otros datos, y su estado sea no mayorizado notificando con mensajes la actividad realizada.

Req(52) Actualización de campos de un asiento contable notificando con mensajes la actividad realizada.

#### **4.3.3.1.7 GESTION REPORTES**

Los usuarios que pueden observar reportes en el modulo de contabilidad son el administrador, auxiliar contable y el gerente.

##### **4.3.3.1.7.1 GESTION LIBRO DIARIO**

Req(53) Consulta del Libro Diario con los siguientes datos: el periodo contable, el estado de jornalización, y tipo de cuenta.

##### **4.3.3.1.7.2 GESTION MAYORES**

Req(54) Consulta de Mayores con los siguientes datos: un rango de fechas (inicio, finalización),

##### **4.3.3.1.7.3 GESTION BALANCE DE COMPROBACIÓN**

Req(55) Consulta del Balance de Comprobación con los siguientes datos: periodo, nivel.

##### **4.3.3.1.7.4 GESTION BALANCE DE RESULTADOS**

Req(56) Consulta del Balance de Resultados con los siguientes datos: periodo, nivel.

##### **4.3.3.1.7.5 GESTION ESTADO DE SITUACIÓN**

Req(57) Consulta del Estado de Situación con los siguientes datos: periodo, nivel.

##### **4.3.3.1.7.6 GESTION GASTOS FIJOS**

Req(58) Consulta de Gastos Fijos con los siguientes datos: periodo, nombre de la empresa, moneda.

#### **4.3.3.2 GESTION INVENTARIOS**

##### **4.3.3.2.1 GESTION CONFIGURACIÓN**

Los usuarios que pueden configurar los parámetros en el módulo inventarios son el administrador, auxiliar contable.

##### **4.3.3.2.2 GESTION FAMILIA DE ACTIVOS**

Req(59) Ingreso de datos de Familia de Activos con la siguiente información: Código, Nombre.

Req(60) Verificación de los datos ingresados en el Req(59), con presentación de mensajes de datos correctos, si ingreso todos los datos obligatorios; datos incorrectos si esta incompleto.

Req(61) Eliminación de familia de activos que no tenga relación con otros datos, notificando con mensajes la actividad realizada.

Req(62) Actualización de campos de familia de activos notificando con mensajes la actividad realizada.

Req(63) Imprimir familia de activos en forma particular o general

##### **4.3.3.2.3 GESTION PORCENTAJES DE RETENCION**

Req(64) Ingreso de Porcentajes de Retención con la siguiente información: tipo de Porcentaje de Retención (IRF, IVA) , valor del porcentaje de retención.

Req(65) Verificación de los datos ingresados en el Req(64), con presentación de mensajes de datos correctos, si ingreso todos los datos obligatorios; datos incorrectos si esta incompleto.

Req(66) Eliminación de porcentajes de retención que no tenga relación con otros datos, notificando con mensajes la actividad realizada.

Req(67) Actualización de campos de porcentajes de retención notificando con mensajes la actividad realizada.

Req(68) Imprimir porcentajes de retención en forma particular o general

#### **4.3.3.2.4 GESTION ACTIVOS**

Req(69) Ingreso de Activos con la siguientes información: Tipo, Familia, Código, Descripción, Marca, Modelo, Cuenta de Activo Fijo, Cuenta de Gasto, Valor Reposición, Porcentaje de Depreciación, Vida Útil, Unidad, Valor Impuesto.

Req(70) Verificación de los datos ingresados en el Req(69), con presentación de mensajes de datos correctos, si ingreso todos los datos obligatorios; datos incorrectos si esta incompleto.

Req(71) Eliminación de activo que no tenga relación con otros datos, notificando con mensajes la actividad realizada.

Req(72) Actualización de campos de activos notificando con mensajes la actividad realizada.

Req(73) Imprimir activos en forma particular o general

#### **4.3.3.2.5 GESTION PROCESOS**

Los usuarios que pueden configurar los parámetros en el módulo inventarios son el administrador, auxiliar contable.

##### **4.3.3.2.5.1 GESTION INVENTARIO INICIAL**

Req(74) Ingreso del Inventario Inicial con la siguiente información: Fecha, Centro de Costo, Moneda, tipo de activo, el tipo de producto, cantidad, precio, la fecha de ingreso .

Req(75) Verificación de los datos ingresados en el Req(74), con presentación de mensajes de datos correctos, si ingreso todos los datos obligatorios; datos incorrectos si esta incompleto.

Req(76) Eliminación de un producto de inventario inicial que no tenga relación con otros datos, notificando con mensajes la actividad realizada.

Req(77) Actualización de campos de un producto del inventario inicial notificando con mensajes la actividad realizada.

Req(78) Imprimir inventario inicial en forma general

#### **4.3.3.2.5.2 GESTION CONSUMO AJUSTE TRANSFERENCIA**

Req(79) Ajuste, Transferencia de Activos con la siguiente información: Fecha, Tipo, Recuerde, Empresa Origen, Cuenta Consumo, Descripción, tipo de activo y centro de costo.

Req(80) Verificación de los datos ingresados en el Req(79), con presentación de mensajes de datos correctos, si ingreso todos los datos obligatorios; datos incorrectos si esta incompleto.

Req(81) Eliminación de una transferencia de Ajuste y Transferencia que no tenga relación con otros datos, notificando con mensajes la actividad realizada.

Req(82) Actualización de campos de productos de la transferencia o ajuste de activos notificando con mensajes la actividad realizada.

Req(83) Buscar Asientos de Consumo, Ajuste y Transferencia de Activos por nombre.

Req(84) Imprimir empresas en forma particular o general

#### **4.3.3.2.5.3 GESTION CALCULO DEPRECIACIÓN**

Req(85) Generación del Calculo de Depreciación con los siguiente información: periodo; con presentación de mensajes si existe datos o no en el periodo señalado.

#### **4.3.3.2.6 GESTION REPORTES**

Los usuarios que pueden observar los reportes en el módulo inventarios son el administrador, auxiliar contable y gerente.

#### **4.3.3.2.6.1 GESTION KARDEX**

Req(86) Consulta de Kardex con la siguiente información: centro de costos, tipo de activo, material, fecha de inicio, fecha de corte, enviar a, moneda.

#### **4.3.3.3 GESTION FACTURACIÓN**

Los usuarios que pueden observar los reportes en el módulo facturación son el administrador, auxiliar contable y usuario.

##### **4.3.3.3.1 GESTION SOLICITUD PEDIDO**

###### **4.3.3.3.1.1 GESTION SOLICITUD PEDIDO**

Req(87) Ingreso de Solicitud de Pedido con la siguiente información: proveedor, fecha de emisión, tipo (servicios, inventarios), cuenta contable, centro de costos, descripción, moneda, subtotal, tipo, articulo, cantidad, valor.

Req(88) Verificación de los datos ingresados en el Req(87), con presentación de mensajes de datos correctos, si ingreso todos los datos obligatorios; datos incorrectos si esta incompleto.

Req(89) Eliminación de un producto ingresado en la solicitud de pedido, notificando con mensajes la actividad realizada.

Req(90) Actualización de un producto ingresado en la solicitud de pedido, notificando con mensajes la actividad realizada.

Req(91) Revisar solicitud de Pedido

Req(92) Rechazar Solicitud de Pedido

Req(93) Desautorizar Solicitud de Pedido

Req(94) Autorizar Solicitud de Pedido

Req(95) Imprimir Solicitud de Pedido.

#### **4.3.3.3.1.2 GESTION APROBAR SOLICITUD**

Req(96) Buscar Solicitud de Pedido con la siguiente información: serial, elaborado por, la fecha desde, fecha hasta, tipo de orden y estado.

#### **4.3.3.3.2 GESTION COMPRAS**

##### **4.3.3.3.2.1 GESTION FACTURAS DE SERVICIOS**

Req(97) Ingreso de Facturas de Servicios con la siguiente información: orden de compra N<sup>o</sup>, Factura N<sup>o</sup>, Fecha de emisión de la factura, Fecha de pago de contado, tipo (factura, nota de crédito), moneda y descripción.

Req(98) Verificación de los datos ingresados en el Req(97), con presentación de mensajes de datos correctos, si ingreso todos los datos obligatorios; datos incorrectos si esta incompleto.

Req(99) Ingreso de Forma de Pago con la siguiente información: la forma de pago de la factura (crédito, contado).

Req(100) Aprobación de Factura, con presentación de mensaje si los datos son correctos.

Req(101) Imprimir factura en forma individual por número o nombre proveedor.

Req(102) Buscar facturas de servicios por: proveedor, fecha y factura.

##### **4.3.3.3.2.2 GESTION NOTA DE CREDITO SERVICIOS**

Req(103) Ingreso de Nota de crédito de Servicios con la siguiente información: proveedor, Factura N<sup>o</sup>, fecha, Nota de crédito N<sup>o</sup>, cuenta contable, descripción, cantidad.

Req(104) Verificación de los datos ingresados en el Req(103), con presentación de mensajes de datos correctos, si ingreso todos los datos obligatorios; datos incorrectos si esta incompleto.

Req(105) Buscar notas de créditos por: número o por nombre de proveedor

#### **4.3.3.3.2.3 GESTION FACTURAS DE INVENTARIO**

Req(106) Ingreso de Facturas de Inventarios con la siguiente información: orden de compra N°, Factura N°, Fecha de emisión de la factura, Fecha de pago de contado, tipo (factura, nota de crédito), moneda y descripción, tipo, artículo, cantidad, valor.

Req(107) Verificación de los datos ingresados en el Req(106), con presentación de mensajes de datos correctos, si ingreso todos los datos obligatorios; datos incorrectos si esta incompleto.

Req(108) Ingreso de Forma de Pago con la siguiente información: la forma de pago de la factura (crédito, contado).

Req(109) Aprobación de Factura, con presentación de mensaje si los datos son correctos.

Req(110) Imprimir factura en forma individual por número o nombre proveedor.

Req(111) Buscar facturas por: proveedor, fecha y factura.

#### **4.3.3.3.2.4 GESTION NOTA DE CREDITO INVENTARIO**

Req(112) Ingreso de Nota de crédito de Inventario con la siguiente información: proveedor, Factura N°, fecha, Nota de crédito N°, cuenta contable, descripción, cantidad

Req(113) Verificación de los datos ingresados en el Req(112), con presentación de mensajes de datos correctos, si ingreso todos los datos obligatorios; datos incorrectos si esta incompleto.

Req(114) Buscar notas de créditos por: número o por nombre de proveedor

#### **4.3.3.3.3 GESTION VENTAS**

##### **4.3.3.3.3.1 GESTION FACTURAS DE PRODUCTOS**

Req(115) Ingreso de Facturas de Productos con la siguiente información:  
Factura N<sup>o</sup>, Fecha de emisión de la factura, Compañía, cliente, moneda, tipo, tipo de comprobante y descripción, tipo, artículo, cantidad, valor.

Req(116) Verificación de los datos ingresados en el Req(115), con presentación de mensajes de datos correctos, si ingreso todos los datos obligatorios; datos incorrectos si esta incompleto.

Req(117) Eliminación de un producto ingresado en la factura de productos, notificando con mensajes la actividad realizada.

Req(118) Actualización de un producto ingresado en la factura de productos, notificando con mensajes la actividad realizada.

Req(119) Ingreso de Forma de Pago con la siguiente información: la forma de pago de la factura (crédito, contado).

Req(120) Aprobación de Factura, con presentación de mensaje si los datos son correctos.

Req(121) Imprimir factura en forma individual por número o nombre cliente.

Req(122) Buscar facturas por: proveedor, fecha y factura.

##### **4.3.3.3.3.2 NOTA DE DEBITO FACTURAS DE PRODUCTOS**

Req(123) Ingreso de Nota de débito de Inventario con la siguiente información: cliente, Factura N<sup>o</sup>, fecha, Nota de crédito N<sup>o</sup>, tipo de comprobante, descripción

Req(124) Verificación de los datos ingresados en el Req(123), con presentación de mensajes de datos correctos, si ingreso todos los datos obligatorios; datos incorrectos si esta incompleto.



Req(125) Buscar notas de débito por: número o por nombre de proveedor

#### **4.3.3.3.3 GESTION FACTURAS DE SERVICIOS**

Req(126) Ingreso de Facturas de Servicios con la siguiente información:  
Factura N<sup>o</sup>, Fecha de emisión de la factura, Centro de costo, cliente, moneda, subtotal , IVA, iva y descripción.

Req(127) Verificación de los datos ingresados en el Req(126), con presentación de mensajes de datos correctos, si ingreso todos los datos obligatorios; datos incorrectos si esta incompleto.

Req(128) Ingreso de Forma de Pago con la siguiente información: la forma de pago de la factura (crédito, contado).

Req(129) Aprobación de Factura, con presentación de mensaje si los datos son correctos.

Req(130) Imprimir factura en forma individual por número o nombre cliente.

Req(131) Buscar facturas por: proveedor, fecha y factura.

#### **4.3.3.3.4 NOTA DE DEBITO FACTURAS DE SERVICIOS**

Req(132) Ingreso de Nota de débito de Servicio con la siguiente información:  
cliente, Factura N<sup>o</sup>, fecha, Nota de crédito N<sup>o</sup>, tipo de comprobante, descripción, cantidad.

Req(133) Verificación de los datos ingresados en el Req(123), con presentación de mensajes de datos correctos, si ingreso todos los datos obligatorios; datos incorrectos si esta incompleto.

Req(134) Buscar notas de débito por: número o por nombre de cliente

#### **4.3.3.3.4 GESTION REPORTES**

##### **4.3.3.3.4.1 GESTION CARTERA**

Req(135) Consulta de Cartera con la siguiente información: fecha fin, proveedor, tipo consulta (global, detallado), cartera, tipo factura (inventarios, facturas).

#### **4.3.3.4 GESTION TESORERIA**

Los usuarios que pueden observar los reportes en el módulo facturación son el administrador, auxiliar contable.

##### **4.3.3.4.1 GESTION BANCOS**

###### **4.3.3.4.1.1 GESTION BANCOS**

Req(136) Ingreso de Bancos con la siguiente información: Nombre, N° de Cuenta, Tipo (corriente, ahorros), cuenta contable (Bancos).

Req(137) Verificación de los datos ingresados en el Req(136), con presentación de mensajes de datos correctos, si ingreso todos los datos obligatorios; datos incorrectos si esta incompleto.

Req(138) Eliminación de bancos que no tenga relación con otros datos, notificando con mensajes la actividad realizada.

Req(139) Actualización de campos de bancos notificando con mensajes la actividad realizada.

Req(140) Buscar empresas por código o nombre.

Req(141) Imprimir bancos en forma particular o general

###### **4.3.3.4.1.2 GESTION CHEQUES**

Req(142) Ingreso de Cheques con la siguiente información: Cheque N°, Fecha de emisión del cheque, Fecha de pago del cheque.

Req(143) Verificación de los datos ingresados en el Req(142), con presentación de mensajes de datos correctos, si ingreso todos los datos obligatorios; datos incorrectos si esta incompleto.

Req(144) Imprimir cheques en forma particular.

#### **4.3.3.4.1.3 GESTION ANULAR CHEQUES**

Req(145) Anular cheques por número, notificando con mensajes la actividad realizada.

#### **4.3.3.4.1.4 GESTION REPORTE CHEQUES**

Req(146) Consulta de Reporte Cheques por: N° de cheque, fecha inicial y fecha final.

#### **4.3.3.4.2 GESTION CUENTAS POR PAGAR**

##### **4.3.3.4.2.1 GESTION CUENTAS IMPUESTOS**

Req(147) Ingreso de Cuentas de Impuestos con la siguiente información: porcentaje de retención y cuenta contable.

Req(148) Verificación de los datos ingresados en el Req(147), con presentación de mensajes de datos correctos, si ingreso todos los datos obligatorios; datos incorrectos si esta incompleto.

Req(149) Eliminación de cuenta de impuestos que no tenga relación con otros datos, notificando con mensajes la actividad realizada.

Req(150) Actualización de campos de cuentas de impuestos notificando con mensajes la actividad realizada.

##### **4.3.3.4.2.2 GESTION FACTURAS POR PAGAR**

Req(151) Consulta de Facturas por Pagar con la siguiente información: fecha, factura N°, descripción y valor.

##### **4.3.3.4.2.3 GESTION CUENTA PROVEEDORES**

Req(152) Consulta cuentas de proveedores con la siguiente información: proveedor, fecha inicial, fecha final.

#### **4.3.3.4.2.4 GESTION REPORTE IMPUESTOS**

Req(153) Consulta reporte impuestos con la siguiente información: fecha inicial y fecha final.

#### **4.3.3.4.3 GESTION FLUJO DE CAJA**

Req(154) Consulta de Flujo de Caja por fecha.

Req(155) Consulta de Pagos pendientes por fecha.

#### **4.3.3.5 GESTION CUENTAS POR COBRAR**

##### **4.3.3.5.1 GESTION COBROS SERVICIOS**

Req(156) Consulta de Cobros Servicios por clientes, empresa o factura.

##### **4.3.3.5.2 GESTION COBROS PRODUCTOS**

Req(157) Consulta de Cobros Productos por clientes, empresa o factura.

#### **4.3.3.6 GESTION CLIENTES**

##### **4.3.3.6.1 GESTION CLIENTES**

Req(158) Ingreso de datos de clientes con la siguiente información: Código, Nombre, Cedula de Identidad, Nacionalidad, Fecha de Nacimiento, Dirección del Domicilio, Teléfono del Domicilio, Fax Celular, Contacto, E-mail, Descripción, estado civil, RUC tipo, porcentaje de retención IRF, IVA.

Req(159) Verificación de los datos ingresados en el Req(158), con presentación de mensajes de datos correctos, si ingreso todos los datos obligatorios; datos incorrectos si esta incompleto.

Req(160) Eliminación de cliente que no tenga relación con otros datos, notificando con mensajes la actividad realizada.

Req(161) Actualización de campos de cliente notificando con mensajes la actividad realizada.

Req(162) Imprimir clientes en forma particular o general

#### **4.3.3.6.2 GESTION PROVEEDORES**

Req(163) Ingreso de datos de proveedores con la siguiente información: Código, Nombre, RUC, Dirección, Teléfono, Fax, Contacto, E-mail, Descripción, Días de Crédito, Tipo, IVA, IRF, Provee.

Req(164) Verificación de los datos ingresados en el Req(163), con presentación de mensajes de datos correctos, si ingreso todos los datos obligatorios; datos incorrectos si esta incompleto.

Req(165) Eliminación de proveedor que no tenga relación con otros datos, notificando con mensajes la actividad realizada.

Req(166) Actualización de campos de proveedor notificando con mensajes la actividad realizada.

Req(167) Imprimir proveedor en forma particular o general

### **4.4 REQUISITOS DE INTERFACES EXTERNAS**

#### **4.4.1 INTERFACES DE USUARIO**

SCFONLINE tendrá una interfaz de usuario orientada a ventanas y el manejo del sistema se realizará a través del teclado y el ratón.

#### **4.4.2 INTERFACES DE SOFTWARE**

SCFONLINE requiere de una interfaz basada en HTML.

#### 4.4.3 INTERFACES DE COMUNICACIÓN

SCFONLINE está orientado para funcionar en línea sobre Internet

#### 4.5 REQUISITOS DE DESARROLLO

El ciclo de vida elegido para desarrollar el producto será el de prototipo iterativo - incremental, de manera que se puedan incorporar fácilmente cambios y nuevas funciones.

#### 4.6 REQUISITOS TECNOLÓGICOS DE HARDWARE Y SOFTWARE

	<b>Cliente</b>	<b>Servidor</b>
<b>Sistema Operativo</b>	Windows Me, Windows XP,	Windows 2000 Server, Windows 2000 o Superior
<b>Memoria (MB)</b>	128 o superior	256 o superior
<b>Procesador</b>	Pentium II o superior	Pentium III o superior
<b>Velocidad</b>	1.7 GHz o superior	1.7 GHz o superior
<b>Browser</b>	IE 6.0.26 o superior	IE 6.0.26 o superior
<b>Programas instalados</b>	_____	PHP 5.0 MSQL 4.1 Apache2.0.4 PHP MYADMIN 8.2.1
<b>Dispositivos adicionales</b>	_____	CD ROM

Tabla 4. 4 Requisitos Tecnológicos

#### 4.7 ATRIBUTOS

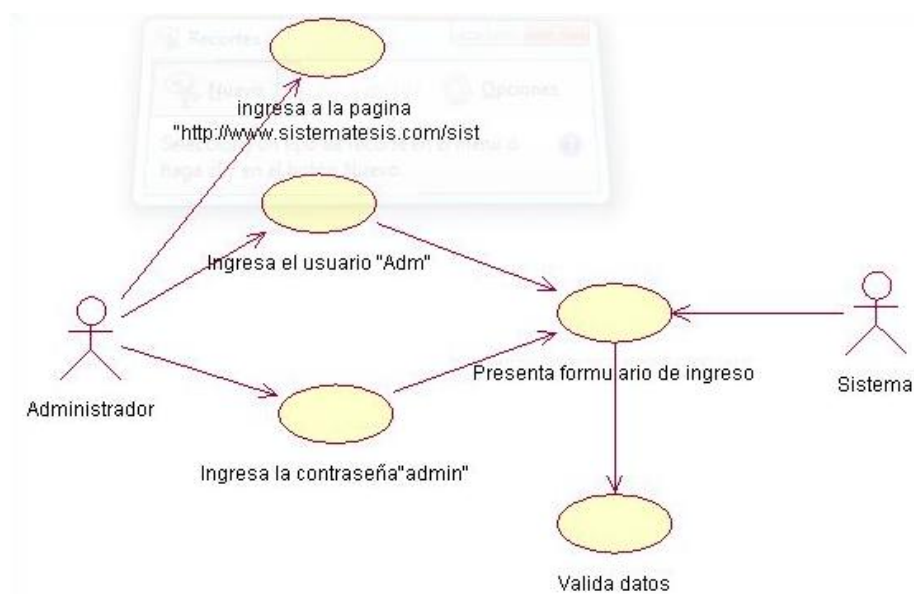
## 4.7.1 SEGURIDAD

El sistema permitirá el manejo de los datos de contabilidad, tesorería, facturación, inventarios únicamente a las personas autorizadas, es decir a los administradores, contador y usuarios que poseen un login y un password, para que no ingresen personas no autorizadas, el administrador bloqueará los accesos a dichos usuarios.

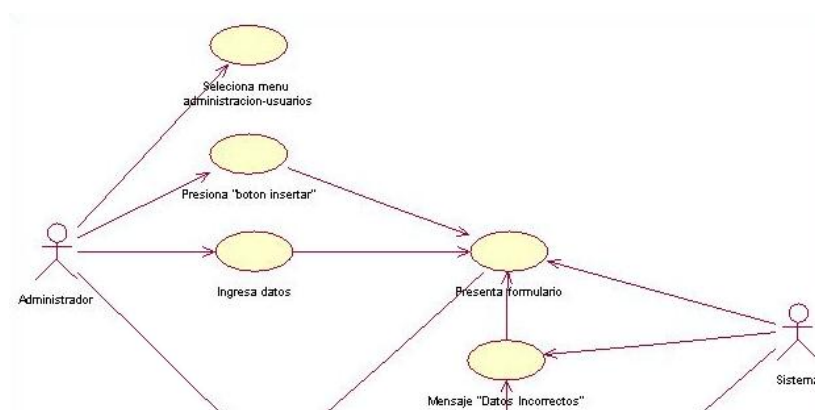
## 4.8 FASE DE ANÁLISIS

### 4.8.1 DIAGRAMAS DE CASOS DE USO

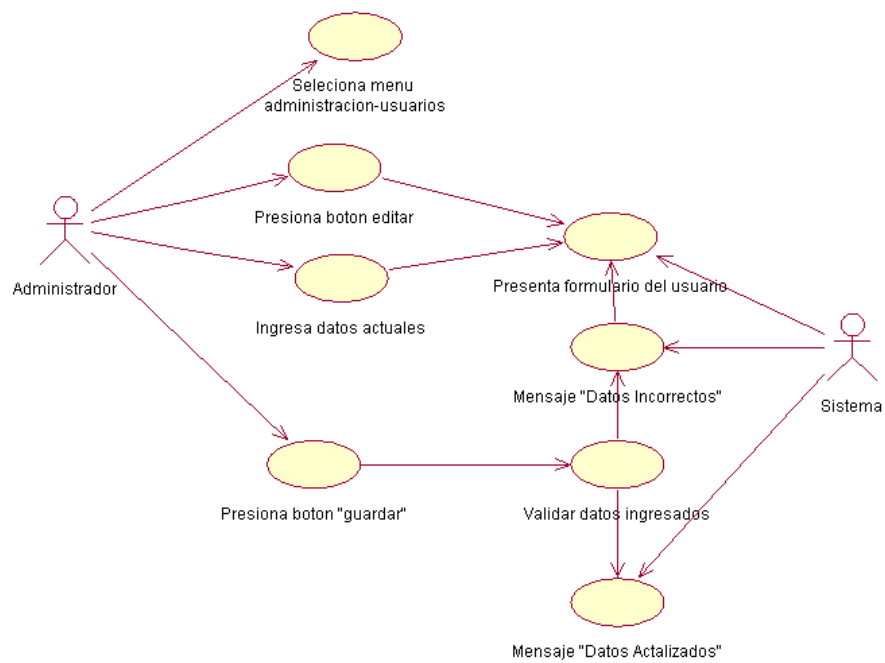
#### 4.8.1.1 INGRESO AL SISTEMA



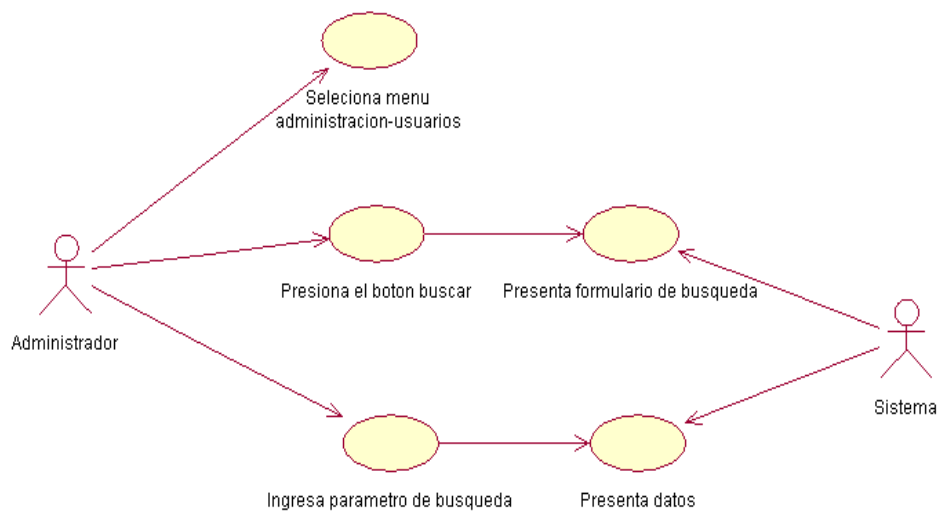
#### 4.8.1.2 INSERTAR USUARIO



### 4.8.1.3 EDITAR USUARIO

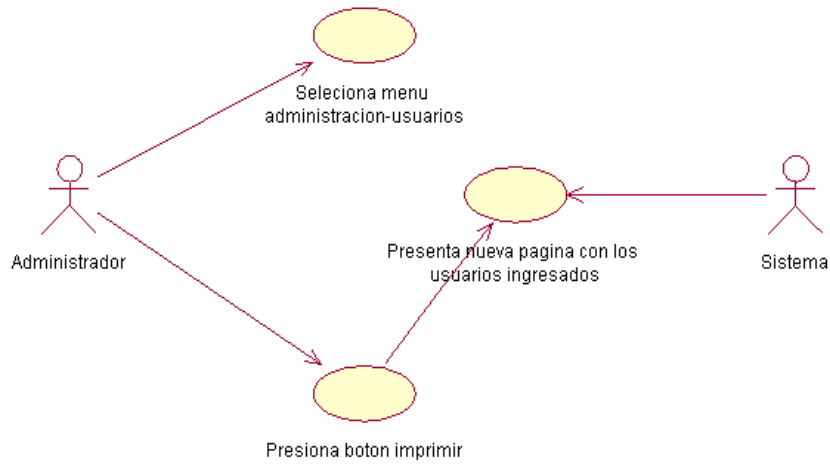


### 4.8.1.4 BUSCAR USUARIO

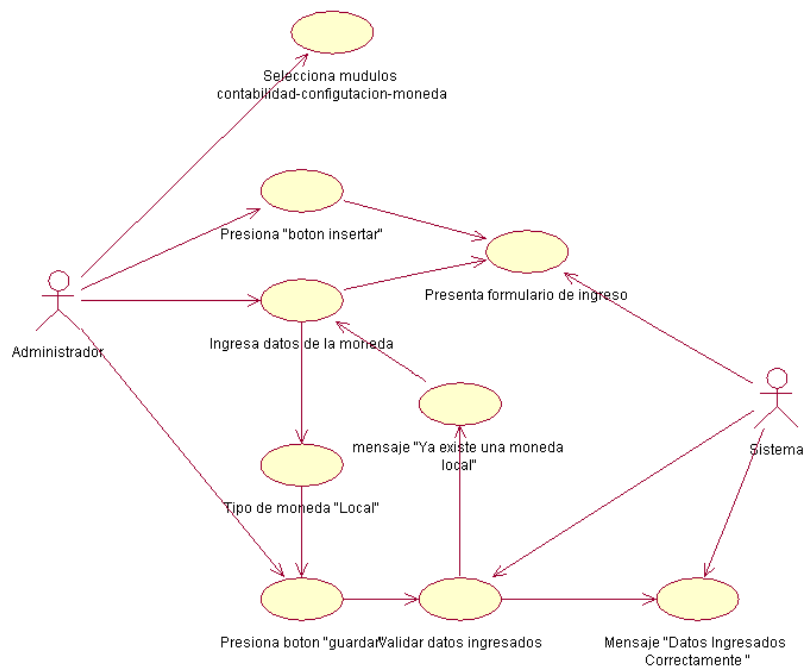




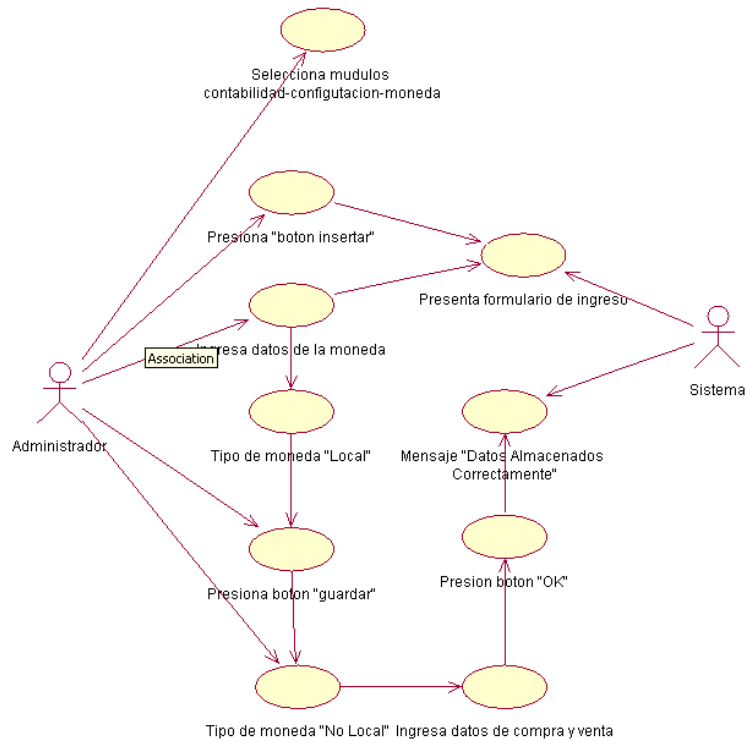
#### 4.8.1.5 IMPRIMIR USUARIO



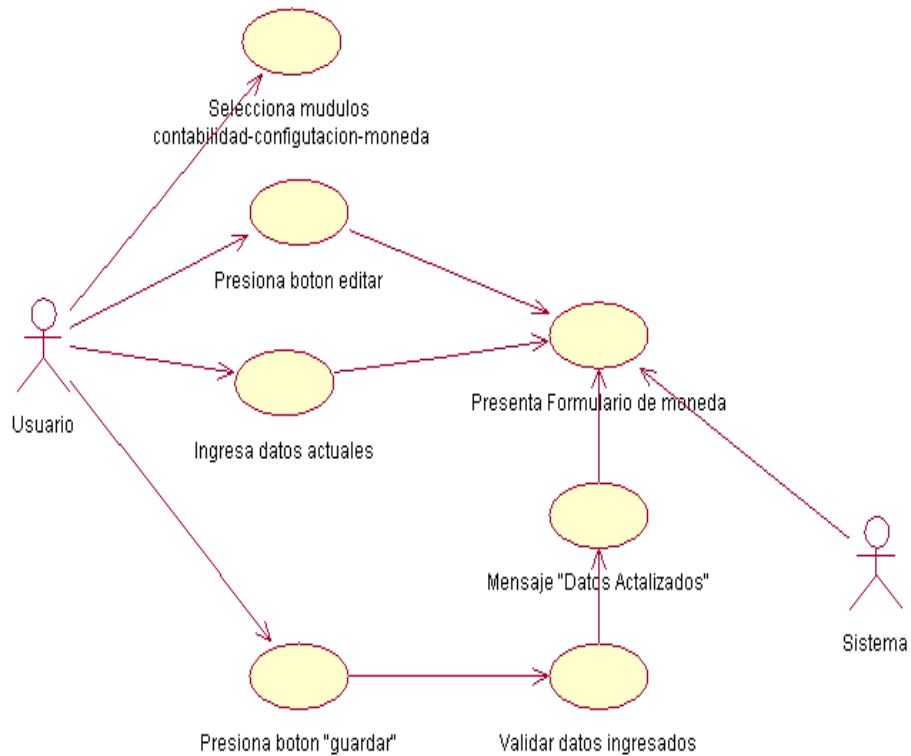
#### 4.8.1.6 INGRESAR MONEDA LOCAL



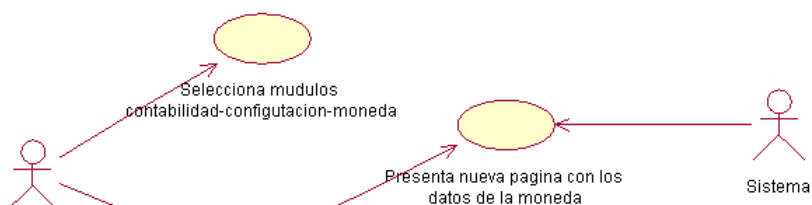
#### 4.8.1.7 INGRESAR MONEDA NO LOCAL



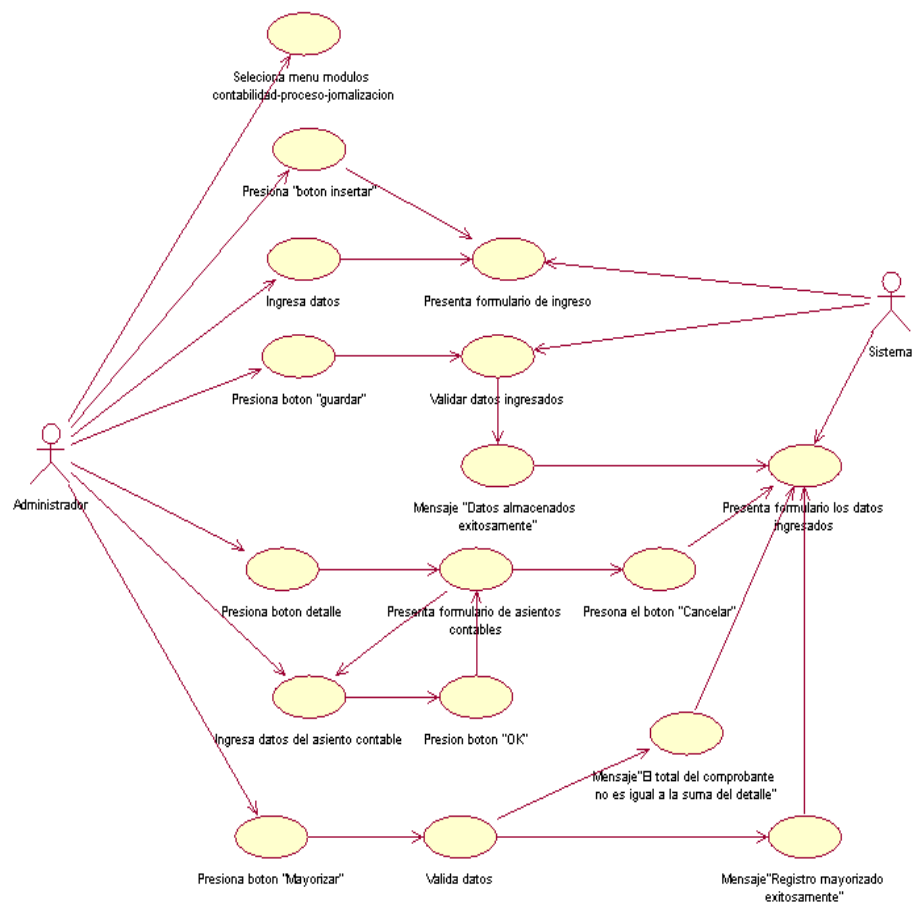
#### 4.8.1.8 EDITAR MONEDA



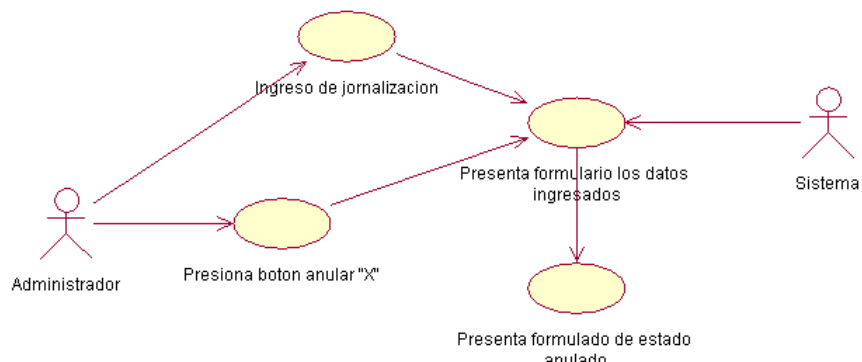
#### 4.8.1.9 IMPRIMIR MONEDA



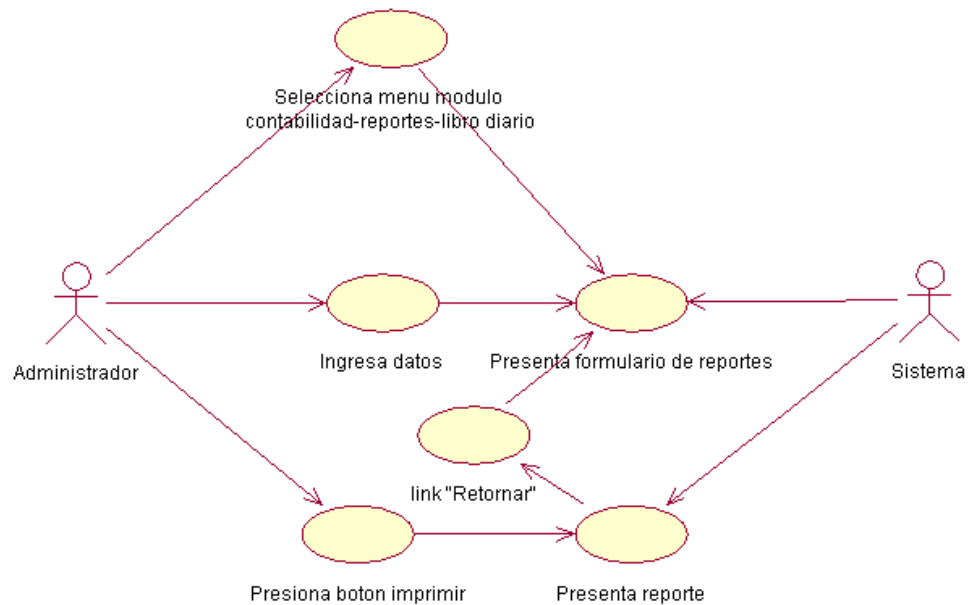
#### 4.8.1.10 INGRESO DE JORNALIZACION



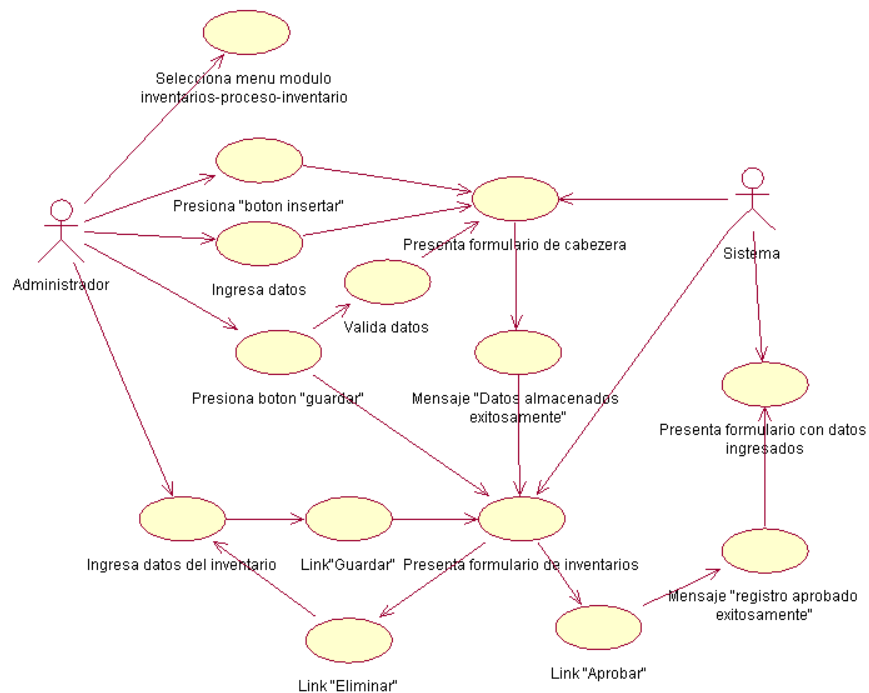
#### 4.8.1.11 ANULACION DE JORNALIZACION



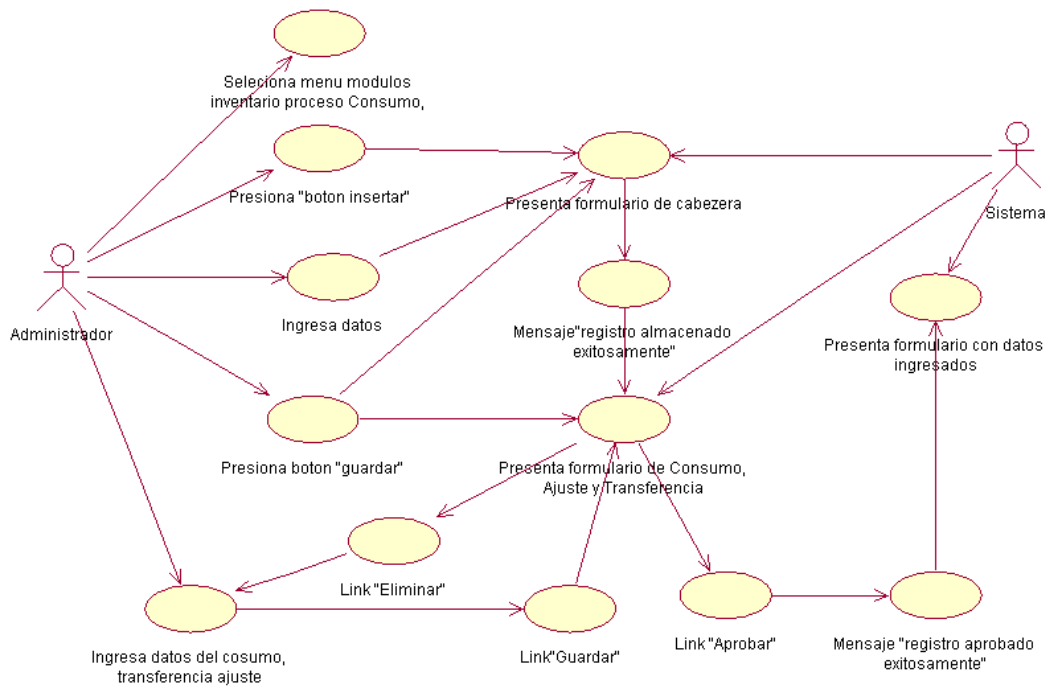
#### 4.8.1.12 REPORTE DE LIBRO DIARIO



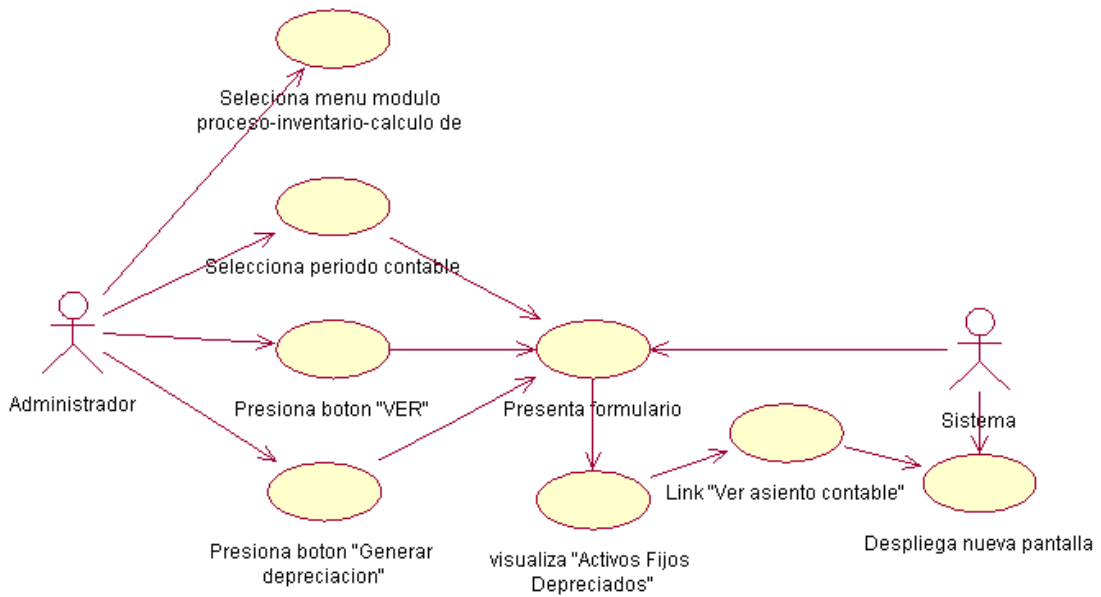
#### 4.8.1.13 INGRESO DE INVENTARIOS



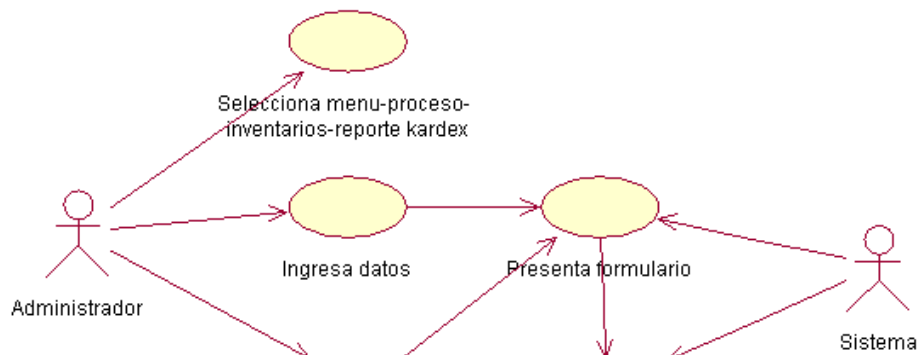
#### 4.8.1.14 CONSUMO AJUSTE Y TRANSFERENCIA



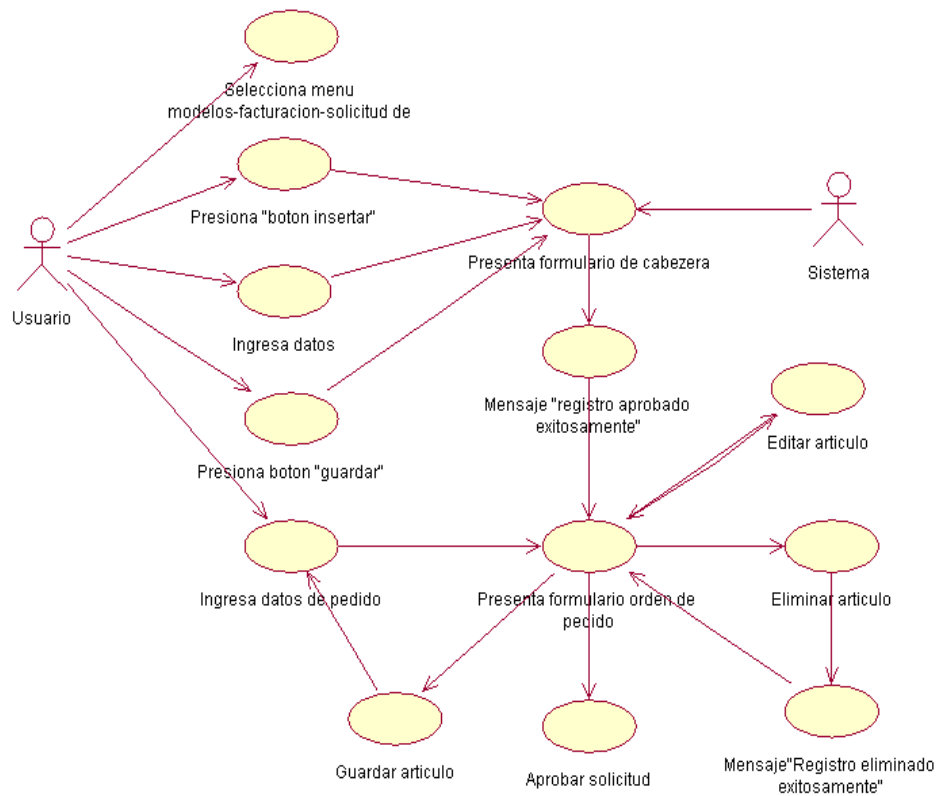
#### 4.8.1.15 CALCULO DE DEPRECIACION



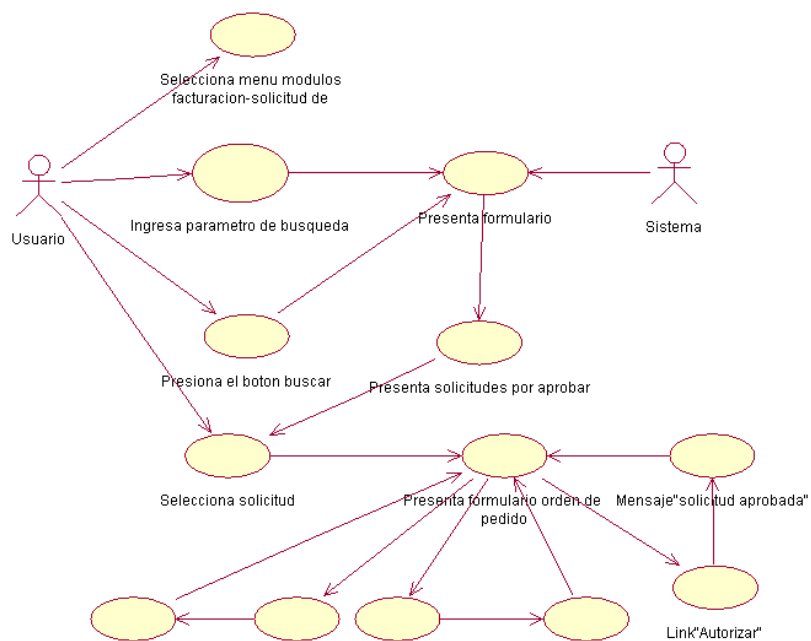
#### 4.8.1.16 REPORTE KARDEX



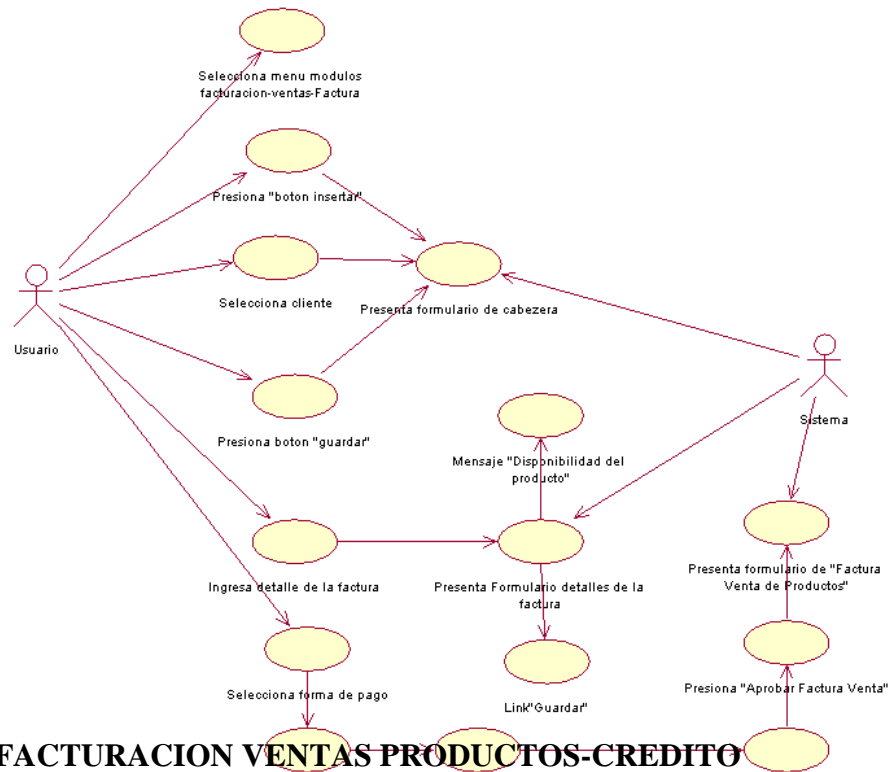
#### 4.8.1.17 SOLICITUD DE PEDIDO



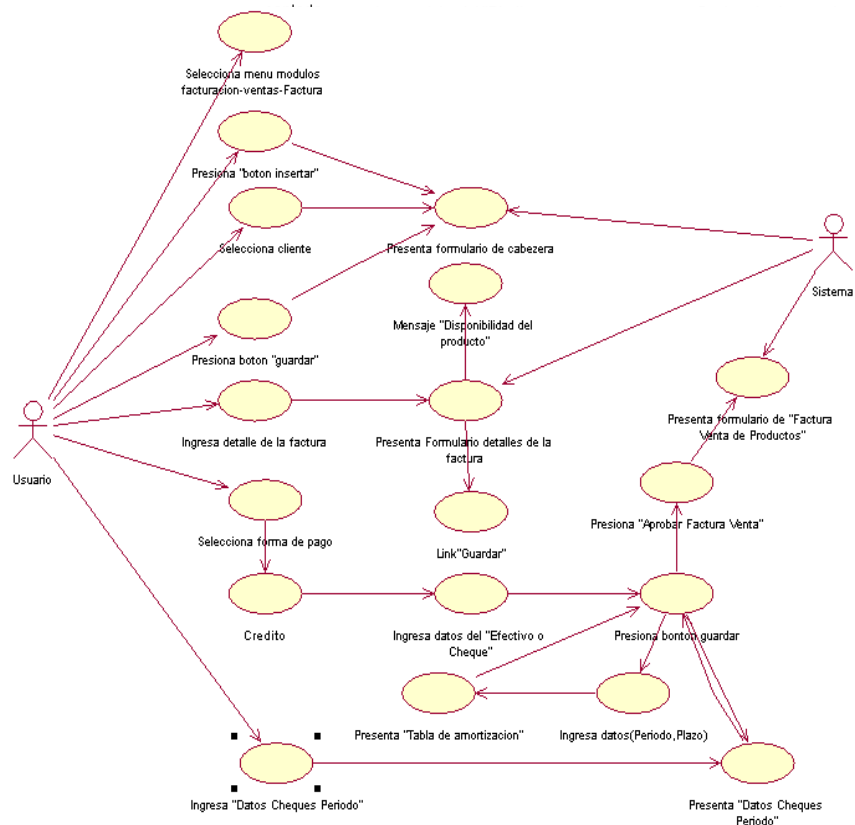
#### 4.8.1.18 APROBAR SOLICITUD



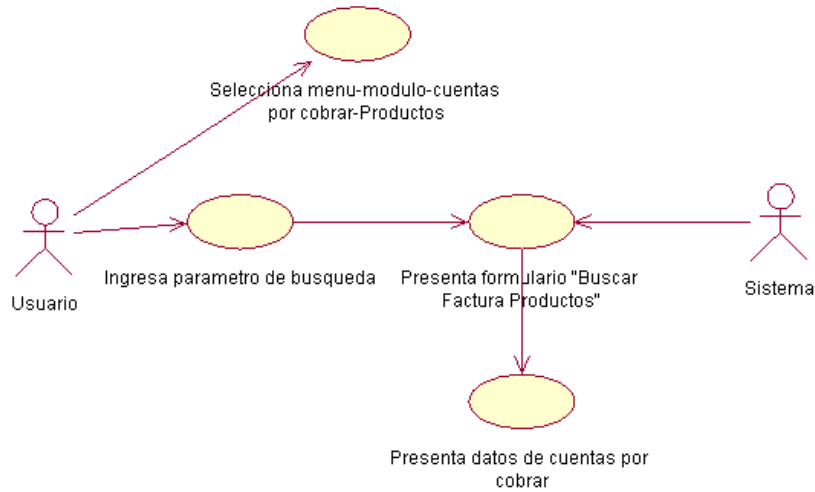
#### 4.8.1.19 FACTURACION VENTAS PRODUCTO-CONTADO



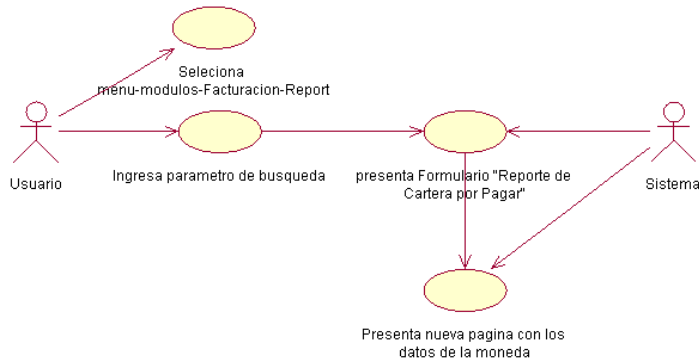
#### 4.8.1.20 FACTURACION VENTAS PRODUCTOS-CREDITO



#### 4.8.1.21 CUENTAS POR COBRAR-PRODUCTOS



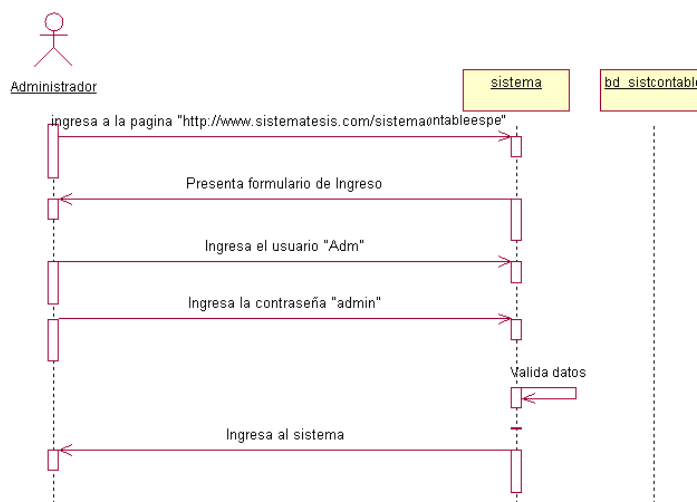
#### 4.8.1.22 REPORTE DE CARTERA



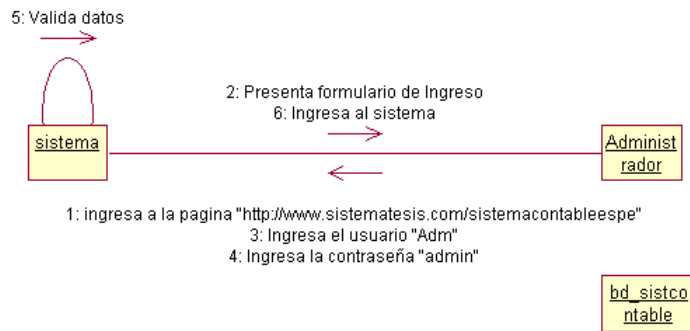
### 4.9 FASE DE DISEÑO

#### 4.9.1 DIAGRAMAS DE SECUENCIA

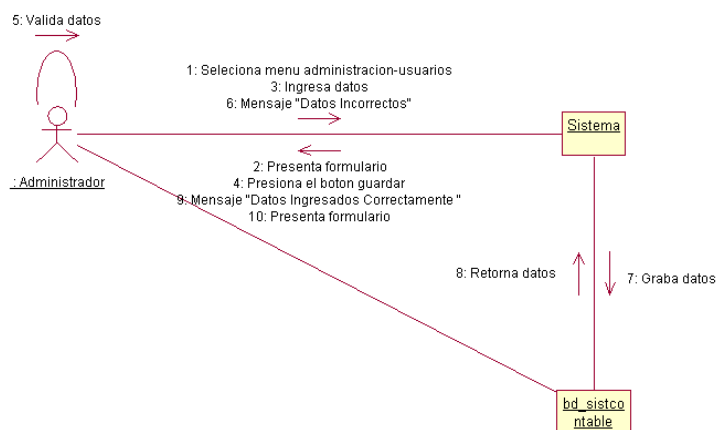
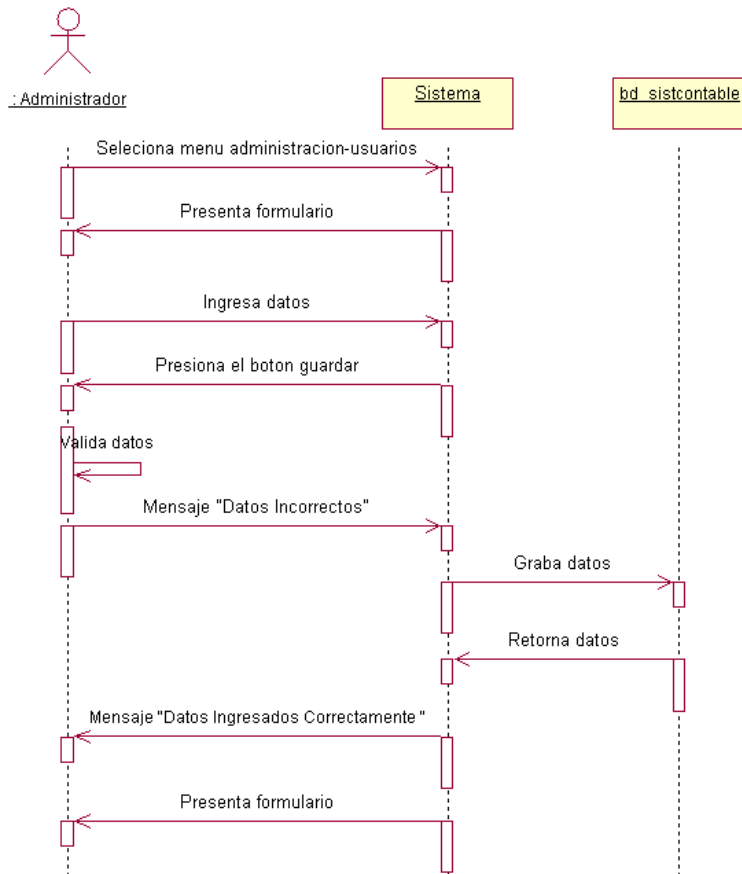
##### 4.9.1.1 INGRESO AL SISTEMA



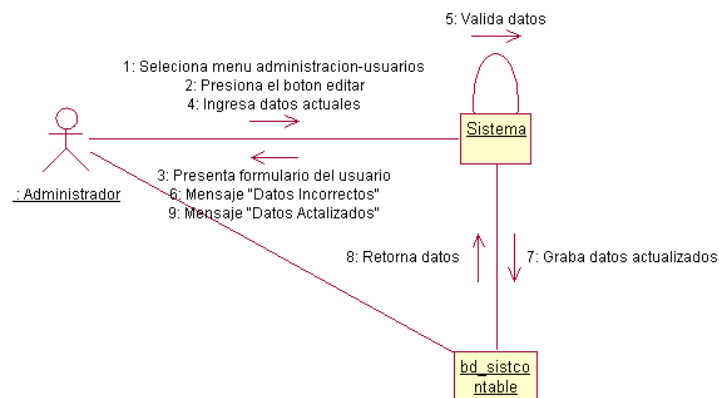
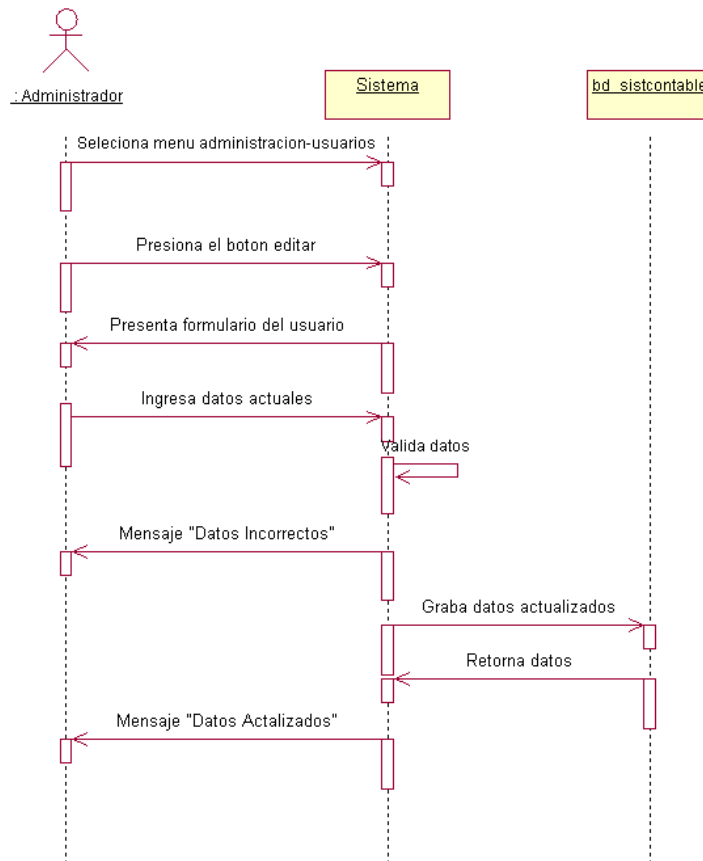




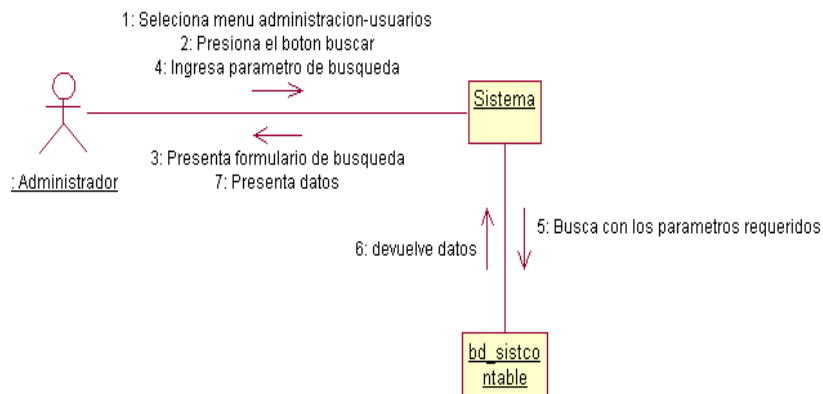
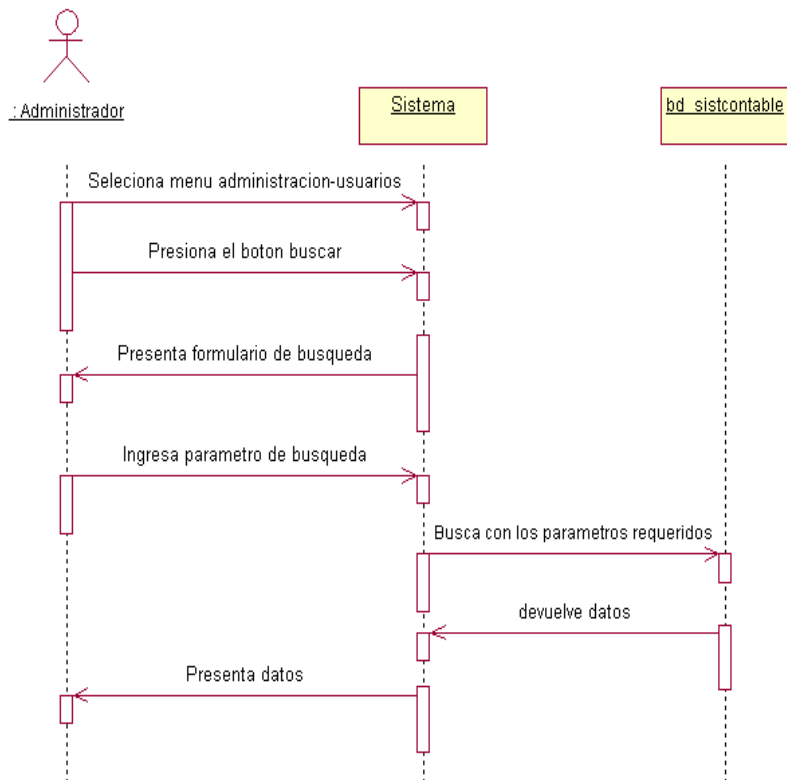
### 4.9.1.2 INSERTAR USUARIO



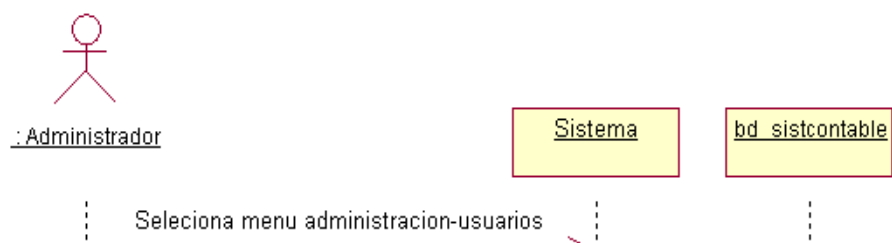
### 4.9.1.3 EDITAR USUARIO

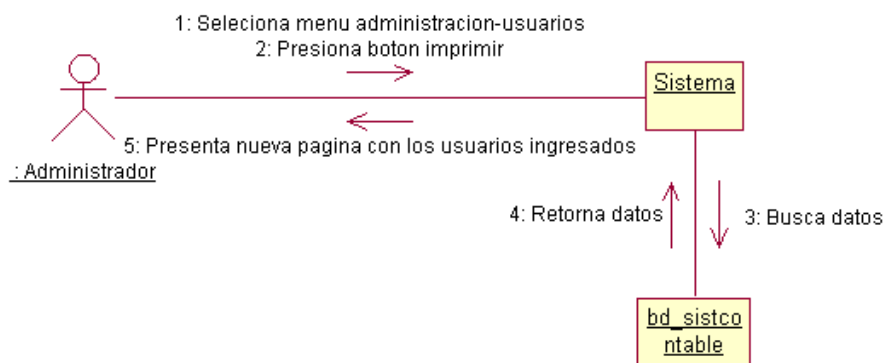


#### 4.9.1.4 BUSCAR USUARIO

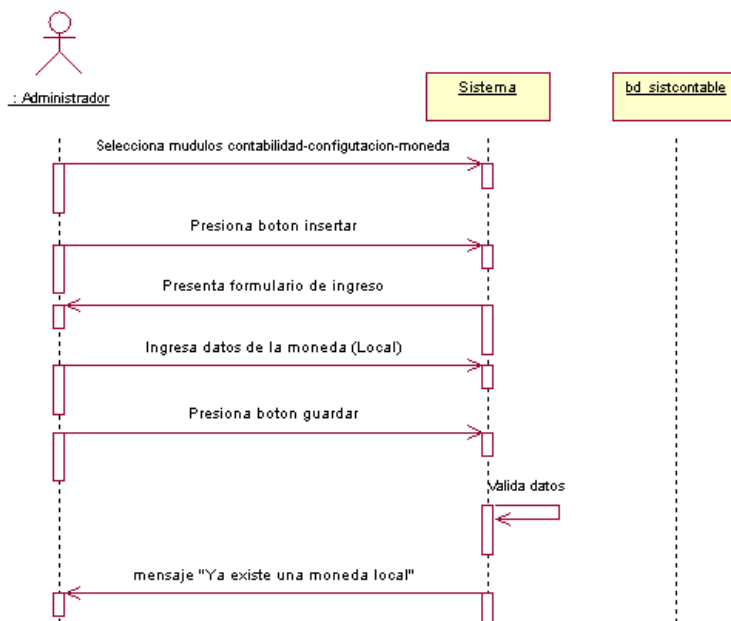


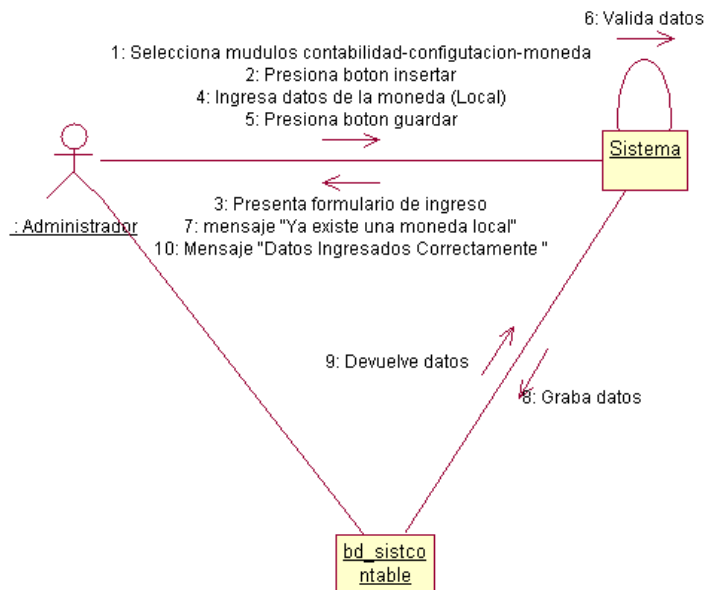
#### 4.9.1.5 IMPRIMIR USUARIO



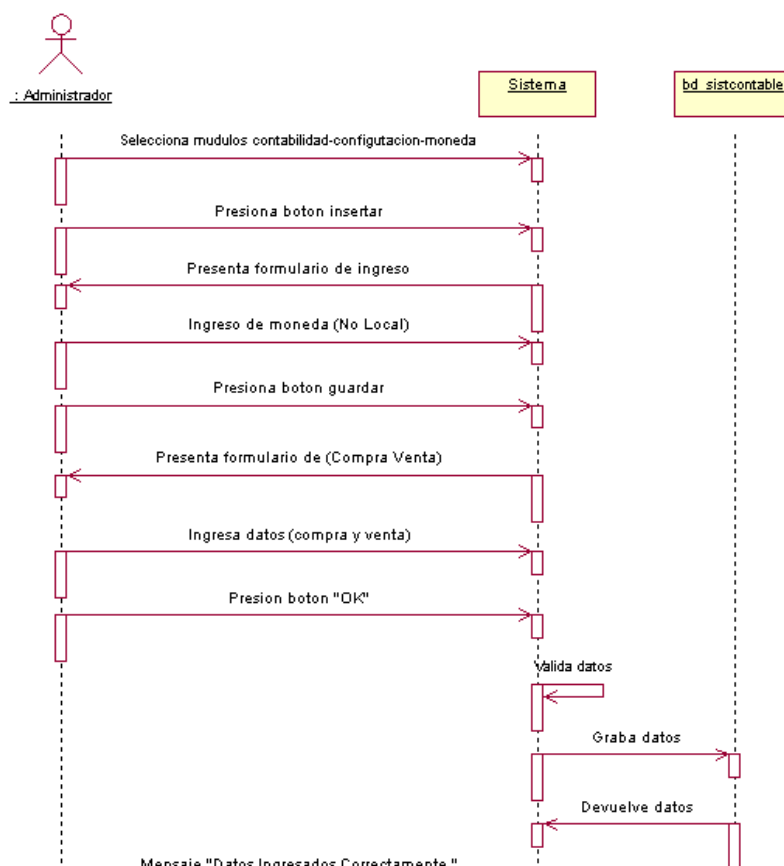


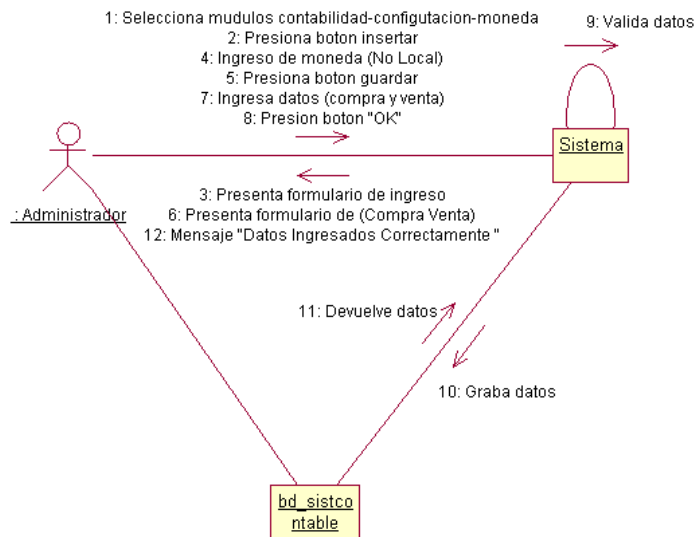
#### 4.9.1.6 INSERTAR MONEDA (LOCAL)



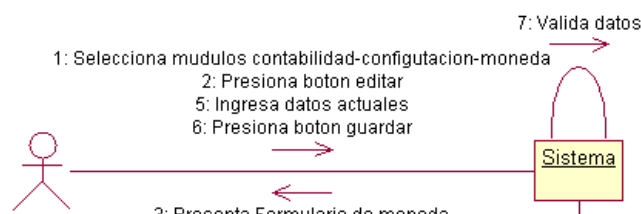
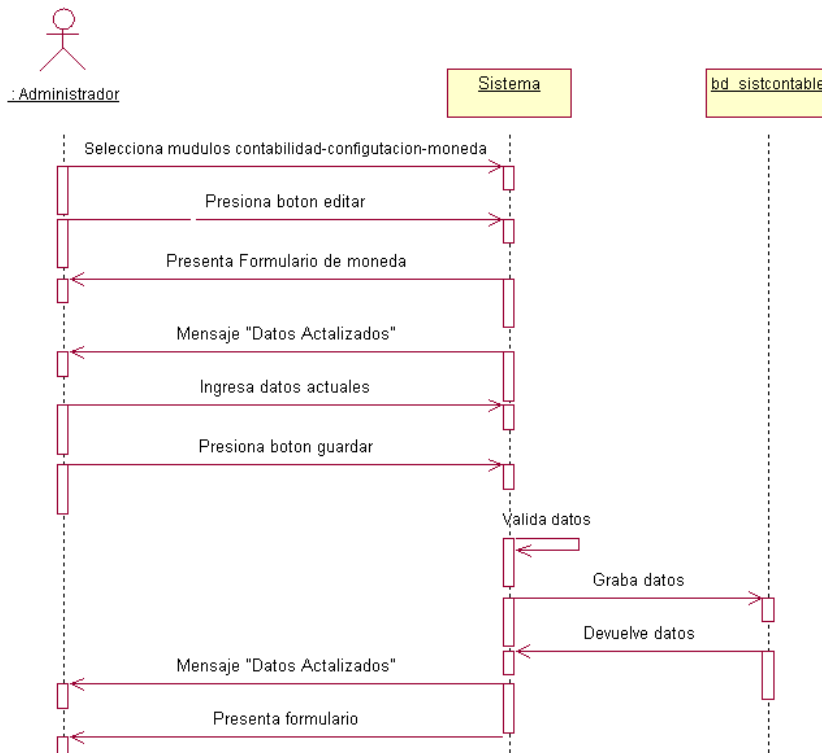


#### 4.9.1.7 INSERTAR MONEDA (NO LOCAL)

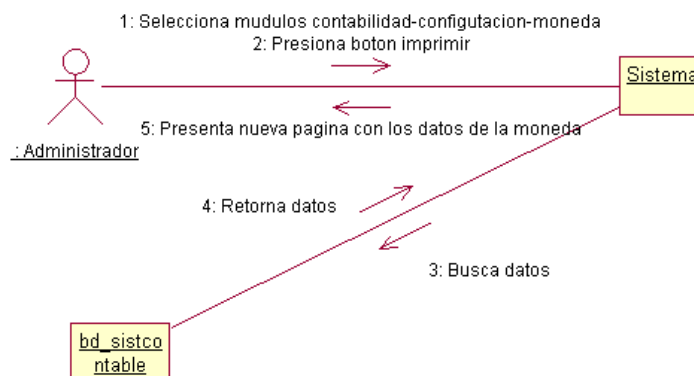
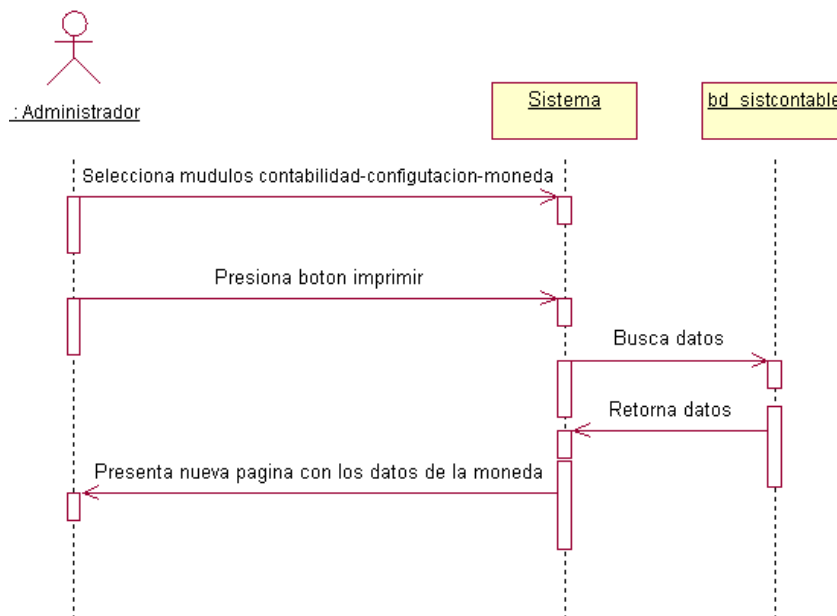




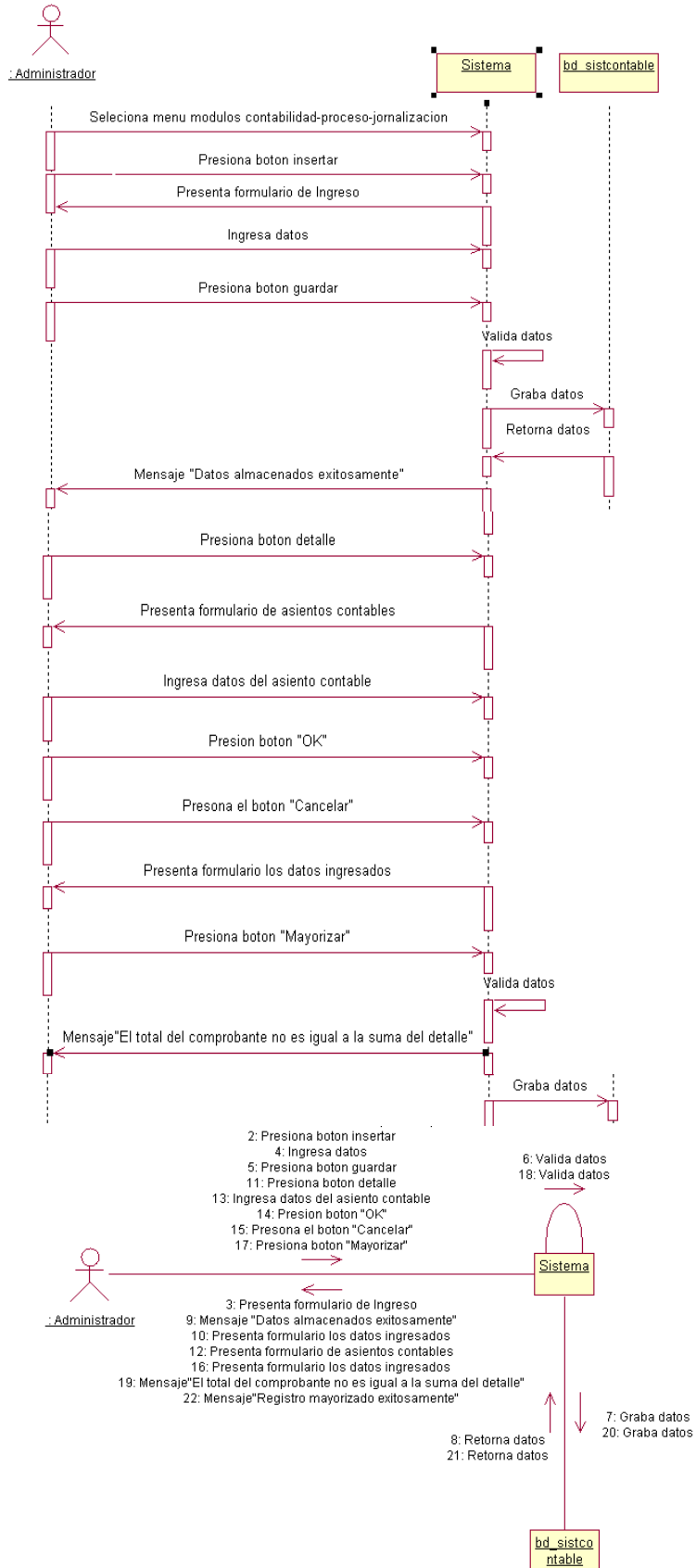
#### 4.9.1.8 EDITAR MONEDA



### 4.9.1.9 IMPRIMIR MONEDA

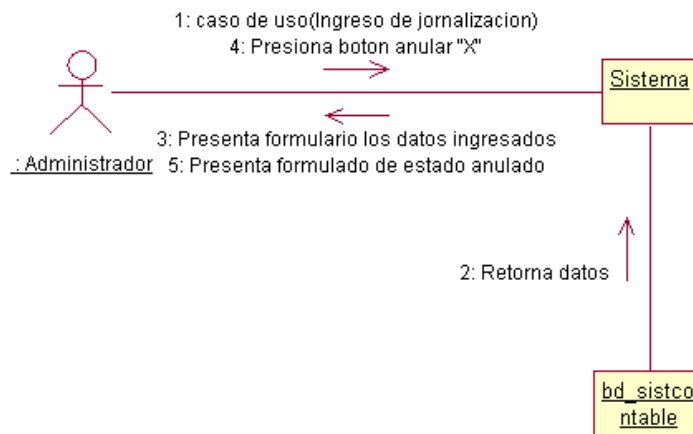
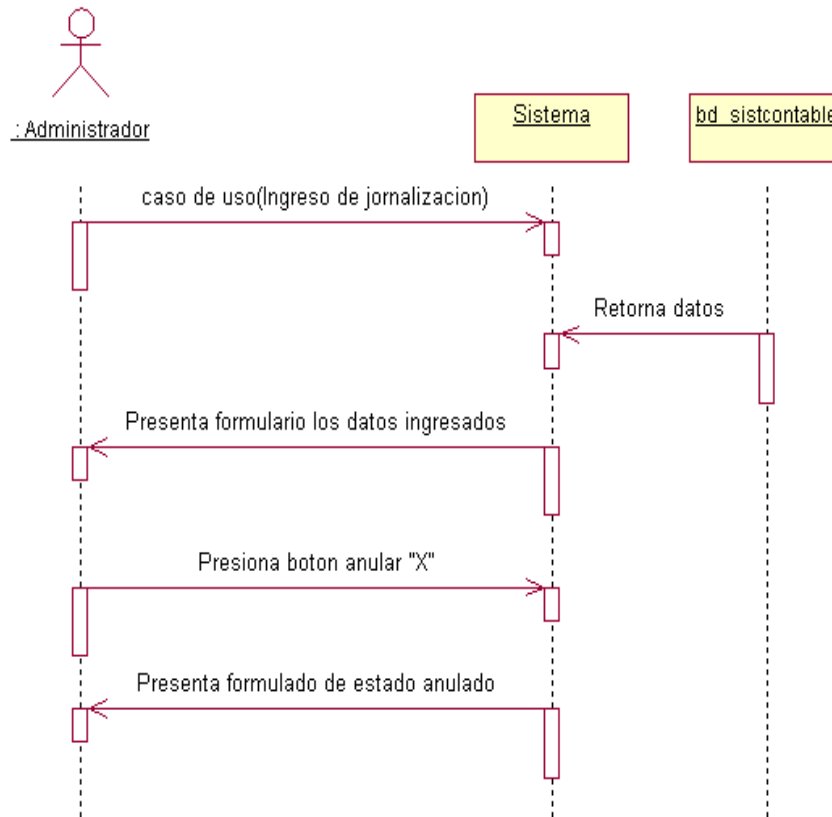


### 4.9.1.10 INGRESO DE JORNALIZACION

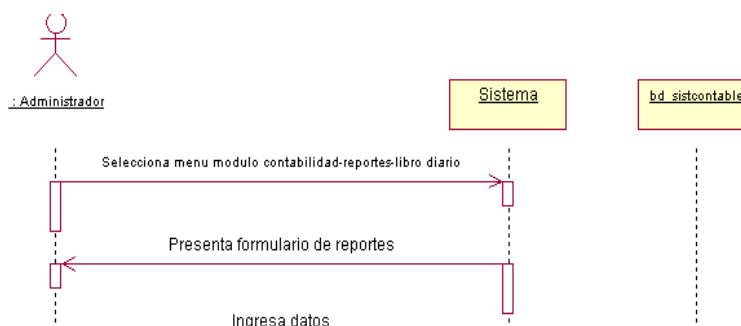


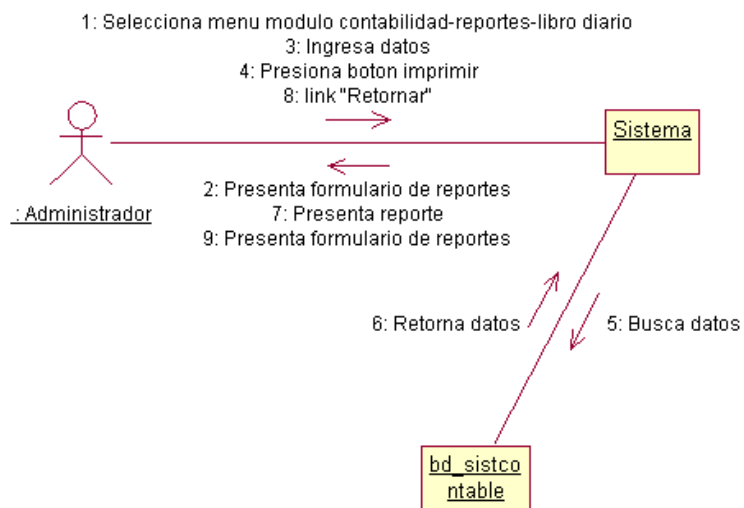


### 4.9.1.11 ANULACION DE JORNALIZACION

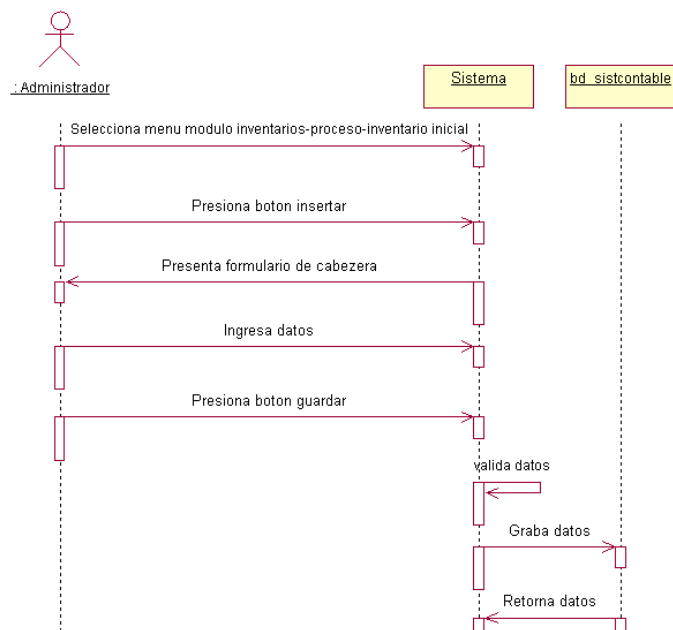


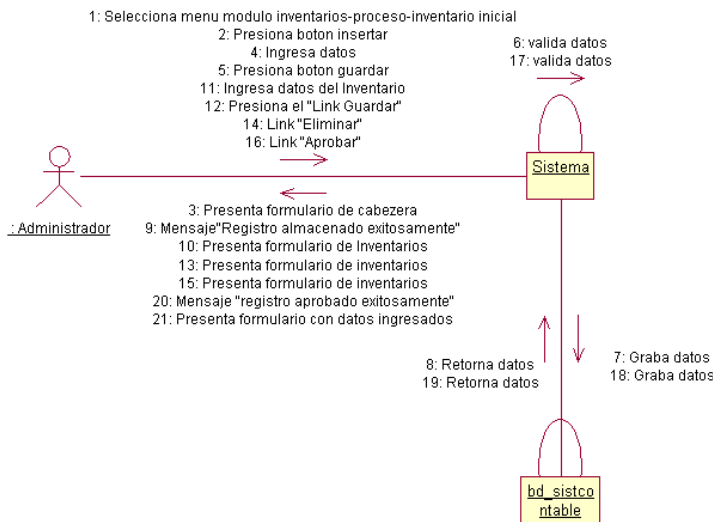
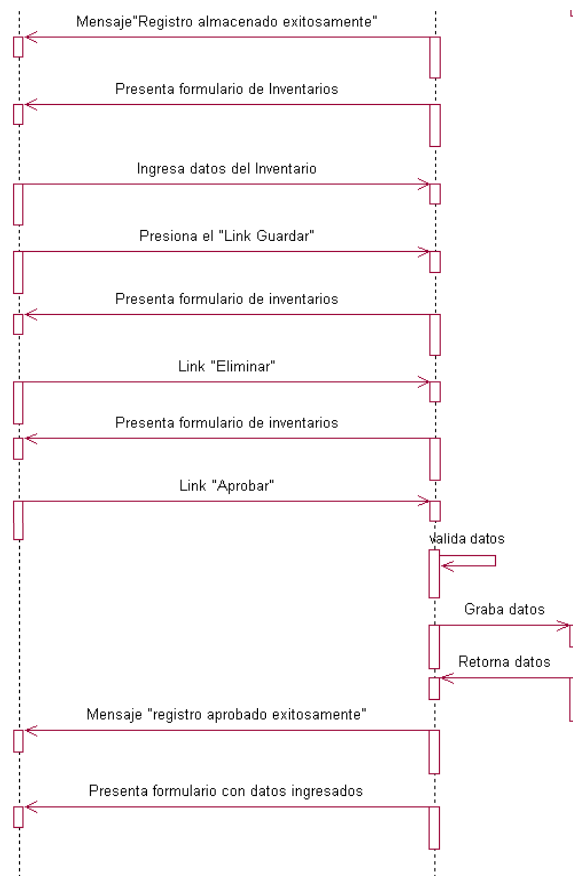
### 4.9.1.12 REPORTE DE LIBRO DIARIO



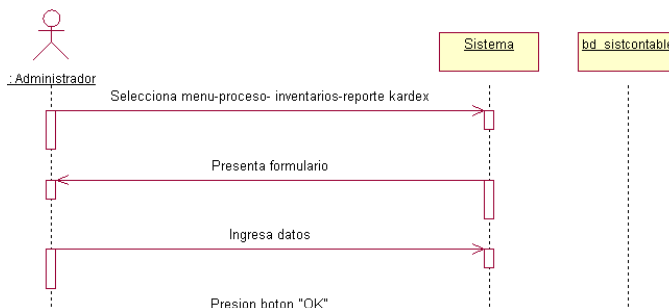


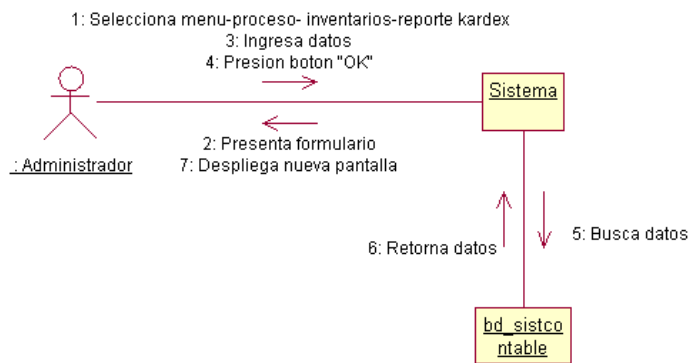
### 4.9.1.13 INGRESO DE INVENTARIOS



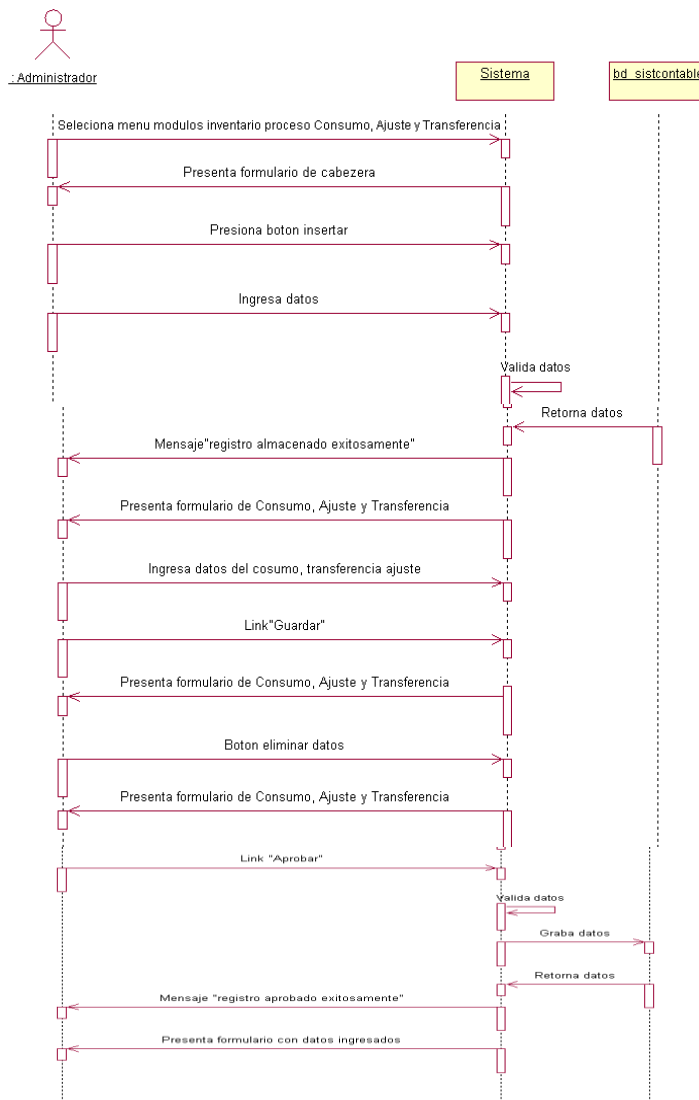


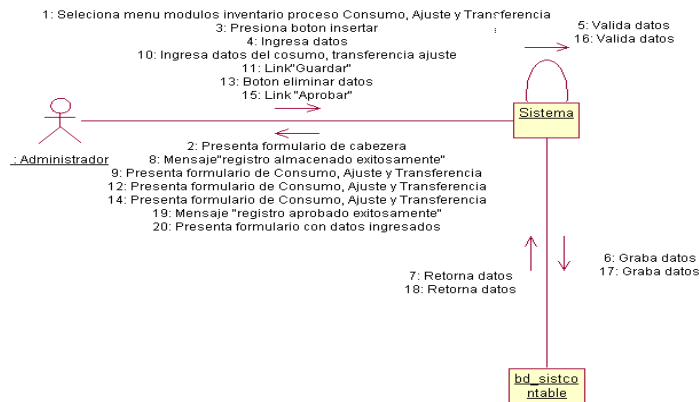
#### 4.9.1.14 REPORTE KARDEX



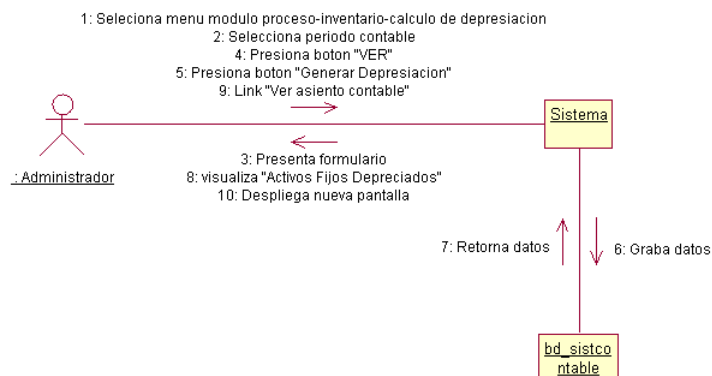
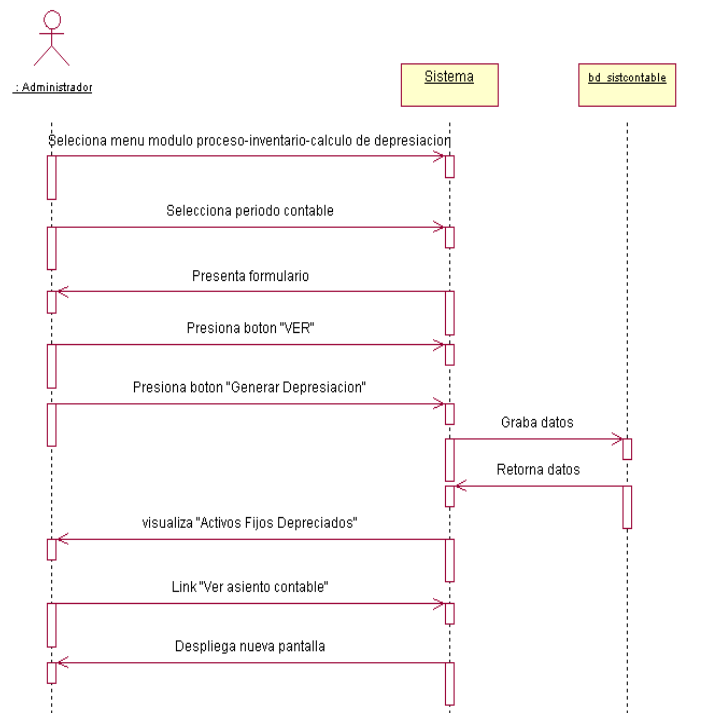


#### 4.9.1.15 CONSUMO, AJUSTES Y TRANSFERENCIA

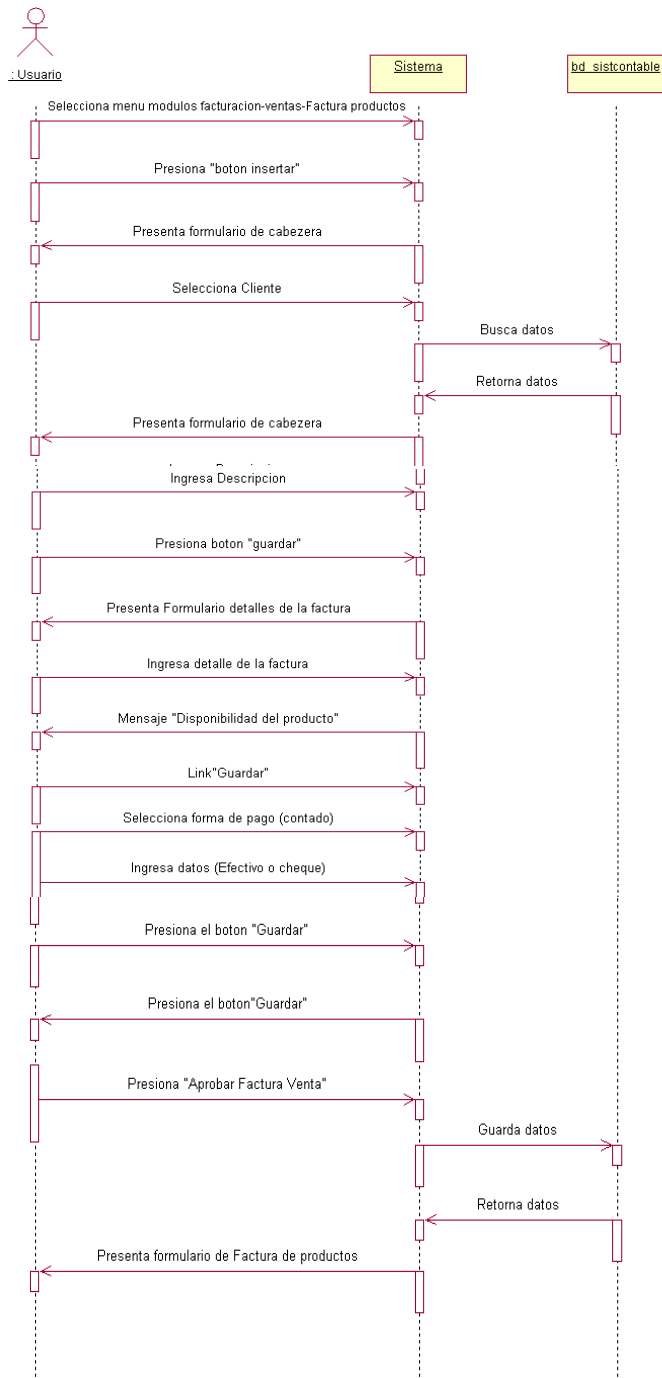


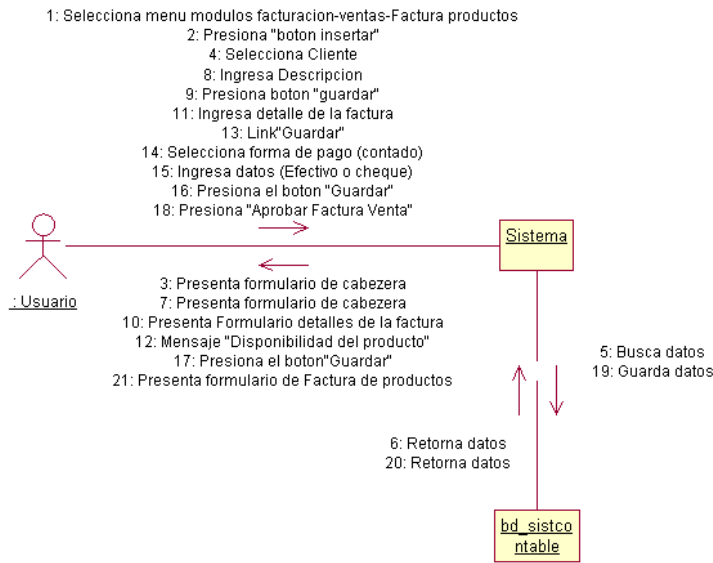


#### 4.9.1.16 CALCULO DE DEPRECIACION

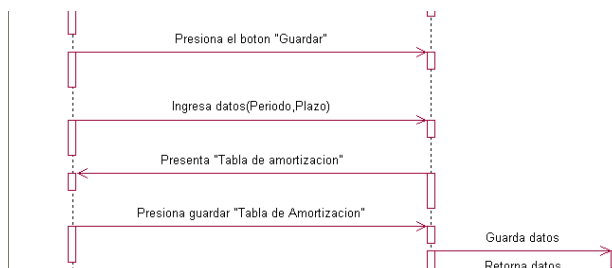
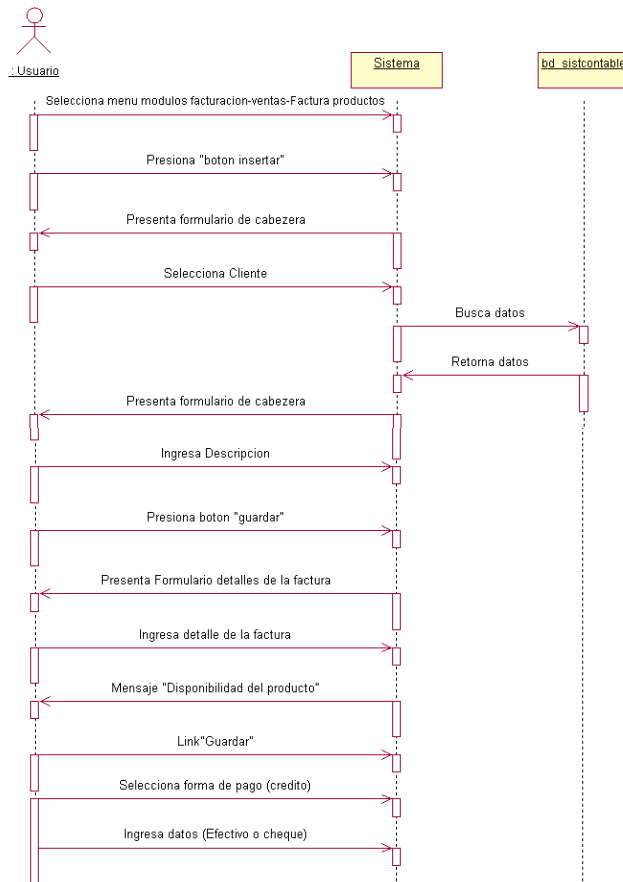


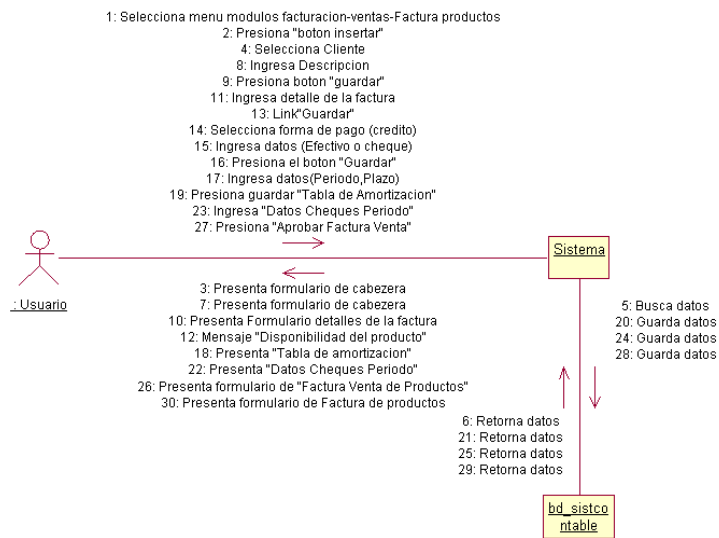
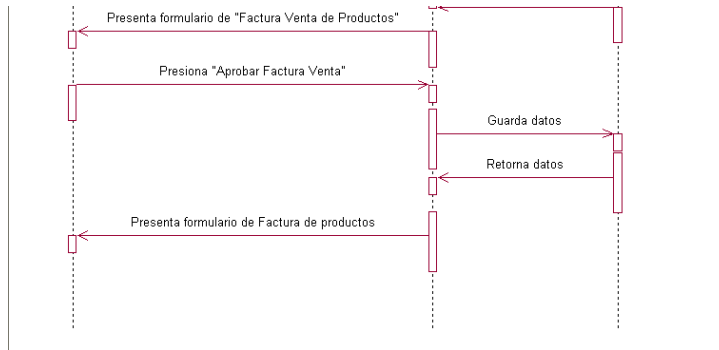
#### 4.9.1.17 FACTURACION VENTA DE PRODUCTOS-CONTADO



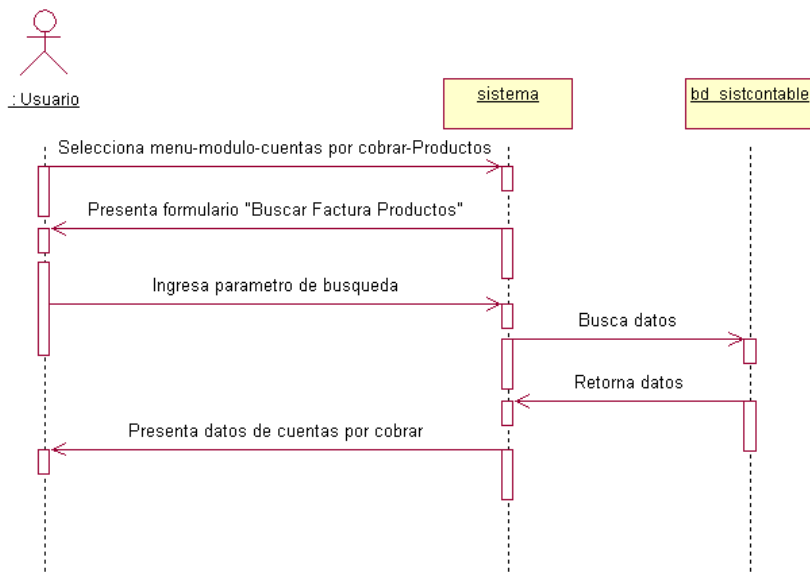


#### 4.9.1.18 FACTURACION VENTA DE PRODUCTOS –CREDITO

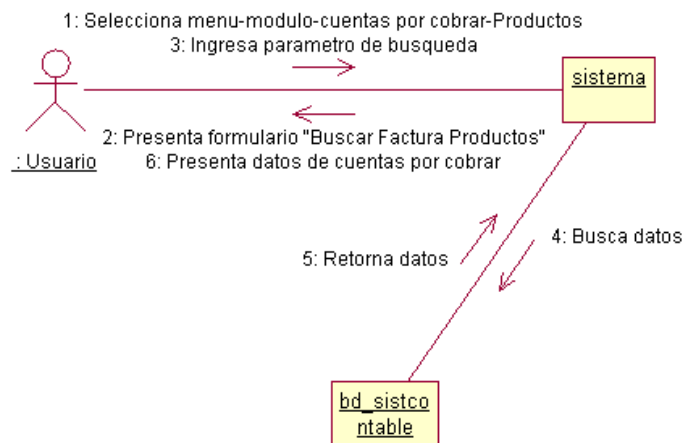




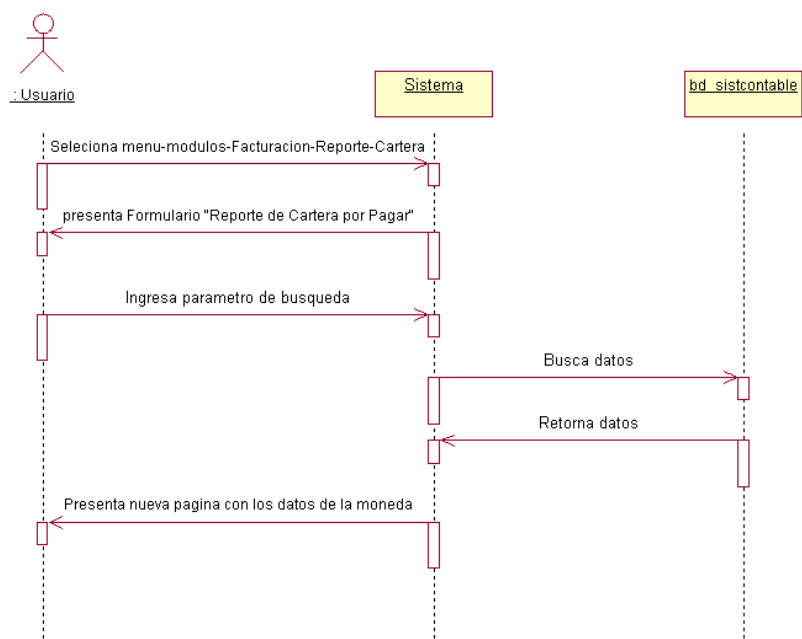
#### 4.9.1.19 CUENTAS POR COBRAR PRODUCTOS

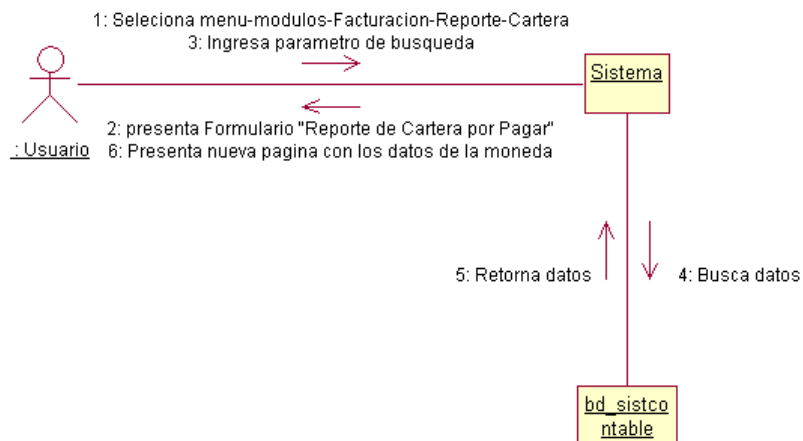






#### 4.9.1.20 REPORTE CARTERA





# CAPITULO V

## V. CONCLUSIONES Y RECOMENDACIONES

### 5.1 CONCLUSIONES

Al finalizar el presente trabajo de investigación podemos concluir lo siguiente:

- La utilización de la metodología MIDAS, es de gran ayuda para el desarrollo de SCFONLINE, por ser una Metodología orientada a Sistemas de Información WEB, visualizando desde el inicio claramente el desarrollo del sistema.

- Luego del análisis del ambiente contable y su incidencia en empresas comerciales y de servicios se llegó a concluir que lo normal en las empresas es que se utilicen programas y bases de datos independientes para cada área funcional, así, el área de contabilidad tiene su propio programa para desarrollar su trabajo, el área de facturación, tesorería e inventarios son módulos distintos. La interacción entre estos programas es escasa o nula, por lo que cada departamento tendrá que introducir los mismos datos en su respectivo software bajo unos criterios diferentes. Esto disminuye la calidad de la información y produce esfuerzos desperdiciados.
- La ventaja de utilizar un sistema totalmente integrado es que conforme van sucediéndose las operaciones se va generando la contabilidad de manera precisa y acorde en el tiempo, además de permitir hacer cierres sencillos y reversibles.
- Las empresas no deben estar sujetas a procesos manuales, lentos e ineficientes, sino, deben tener procesos de jurnalización, conversiones y mayorización de transacciones en forma automática que ayude a la toma rápida de decisiones empresariales.
- El Sistema está diseñado para satisfacer las necesidades de empresas que operan con varias divisas. Se registran operaciones en la moneda original y convierte los valores a moneda nacional dependiendo de la fecha en que fueron realizadas, también permite ver la diferencia debida al cambio.
- La metodología MIDAS emerge para dar soluciones rápidas y ágiles a la creación de Sistemas de Información Web como un producto software de calidad, priorizando los requisitos, creando versiones software y enfocándose al cliente.
- En función al análisis y del desarrollo se creó una aplicación Integrada siguiendo la Metodología Midas, la misma que viene a ser un producto

parametrizable y que se puede aplicar en empresas para llevar un mejor control y permitir la toma de decisiones por parte de Gerencia.

- Con la investigación, esfuerzo y constancia adquirimos nuevos conocimientos en el desarrollo de Software, para lo cual aplicamos Ingeniería de Software, técnicas, estándares, siguiendo de manera correcta la metodología utilizada.

## **5.2 RECOMENDACIONES**

Al finalizar el presente trabajo de investigación recomendamos lo siguiente:

- Se recomienda la utilización de la metodología MIDAS debido a que en comparación con otras metodologías el desarrollo de cualquier software es un proceso más rápido por que se enfoca principalmente en el análisis, diseño y pruebas, constituyéndose en una metodología eficiente en entornos Web.
- Al utilizar la metodología propuesta en este análisis, se debe seguir todas las directrices expuestas en la misma para conseguir los resultados esperados, y aprovechar las ventajas que esta metodología ofrece.
- Al final del análisis, debido a que la contabilidad es un factor muy importante dentro de cualquier organización, surge la necesidad de automatizar estas tareas para esto se recomienda la utilización de MIDAS debido a sus beneficios como metodología las mismas que se reflejan en la aplicación desarrollada.
- Si se toma la decisión implantar un sistema Web de contabilidad similar al expuesto en este trabajo, se recomienda tomar todas las medidas necesarias de seguridad, por su funcionamiento en Internet, además se debe contar con el personal calificado para su correcta manipulación.

## **REFERENCIAS BIBLIOGRAFICAS**

Pressman, Roger S; Ingeniería del Software; Un enfoque práctico; Tercera Edición; 1992.

Glosario Estándar de Términos de Ingeniería del Software de IEEE, 1983

Bohem, B.W; Software Engineering, IEEE; Transactions on Computers,1076

Zapata, Pedro; Contabilidad General; Cuarta Edición; 2003

Coral, Lucy; Gudiño, Emma; Contabilidad Universitaria;, Cuarta Edición, 2001.

Casas, Enrique; Biblioteca Moderna de Contabilidad; Octava Edición; Tomo VII, 2004.

Hernard, J; Hargadon, Jr; Munera Armando; Principios de Contabilidad; Editorial Norma.

Coral, Lucy; Gudiño, Emma; Contabilidad Universitaria;, Cuarta Edición, 2001

MATERIAL DE USO DIDÁCTICO PARA ESTUDIANTES DE LA U.T.P DE LOJA.

## DIRECCIONES WEB

<http://www.willydev.net/descargas/prev/TodoAgil.Pdf>

<http://www.di.uminho.pt/~jmf/PUBLI/papers/phd-Fernandes.pdf>

<http://www.ewh.ieee.org/reg/9/etrans/Marzo2005/paper141.pdf>

[http://www.monografias.com/trabajos28/principios-contabilidad/principios-](http://www.monografias.com/trabajos28/principios-contabilidad/principios-contabilidad/principios-)

<http://www.tema1-1.3.pdf>

<http://www.MIDAS.pdf>

<http://www.proyecto-mda.pdf>

<http://www.contabilidad.shtml>

[www.unab.edu.co/editorialunab/revistas/rcc/pdfs/r22\\_art5\\_c.pdf](http://www.unab.edu.co/editorialunab/revistas/rcc/pdfs/r22_art5_c.pdf)

[www.ewh.ieee.org/reg/9/etrans/Marzo2005/paper141.pdf](http://www.ewh.ieee.org/reg/9/etrans/Marzo2005/paper141.pdf)

[www.jungla.dit.upm.es/~gabriel/seminario/01-02/emarcos.ppt](http://www.jungla.dit.upm.es/~gabriel/seminario/01-02/emarcos.ppt)

<http://rmm-java.stern.nyu.edu/rmm/>, 1998.

<http://www.atarix.org/webe03/Papers/2Caceres.pdf>

<http://kybele.escet.urjc.es/Seminarios/Documentos/MIDAS.pdf>

<http://www.kazak.ws/midas/midas.php>

<http://www.monografias.com/trabajos10/conta/conta.shtml>

<http://www.mailxmail.com/curso/empresa/contabilidad/toc.htm>

<http://www.infomipyme.com/Docs/GT/Offline/balancege.htm>

## **ANEXO**

CD que contiene lo siguiente:

- **Archivo de la tesis:** “ DESARROLLO DEL SISTEMA CONTABLE Y FINANCIERO ONLINE PARAMETRIZABLE “SCFONLINE”.
  
- **Manual de Usuario SCFONLINE**