

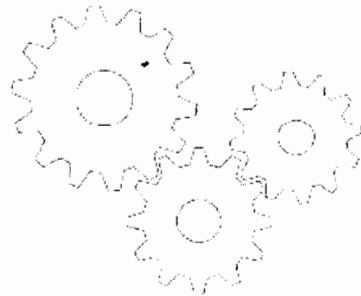


Universidad de las Ciencias Informáticas
DIRECCIÓN DE INFORMATIZACIÓN

Vinculación de los diagramas de flujo de procesos con el ciclo de desarrollo de software

Trabajo de diploma para optar por el título de

Ingeniero Informático



Autor: Yadiel Ramos Rodríguez

Tutor: Ing. Dubier Rivero Pérez

Ciudad de la Habana, Diciembre, 2005.

Resumen

En los momentos actuales, los instrumentos y la conectividad de la era digital ponen en nuestras manos los medios para obtener información, compartirla, y actuar en función de ella de muchas maneras nuevas y notables. Para funcionar en este mundo hay que desarrollar una nueva infraestructura digital, tanto mental como técnica, de tal forma que todos los elementos se relacionen de manera armónica, con capacidad para funcionar de manera fluida y eficiente. Con el avance de las tecnologías y de la gestión de software se ha detectado que el mayor problema que se tiene está centrado en que muchos de los sistemas que se hacen nunca llegan a usarse, o no alcanzan las expectativas que los clientes habían solicitado. Con el objetivo de permitir un mejor entendimiento entre los desarrolladores de software y los clientes, se han diversificado los esfuerzos, y como respuesta se ha decidido en la Universidad de las Ciencias Informáticas asignar la tarea de la identificación, caracterización y automatización de los procesos de negocio a los ingenieros industriales para que después ellos le brinden esta información a los desarrolladores, sin embargo, todavía no es posible decir que ya se ha solucionado dicho inconveniente. Existen factores como la falta de documentación acerca de este tema, ya que es algo novedoso, que atentan contra un desarrollo exitoso del software en cuestión. Este trabajo se propone investigar a fondo como vincular el trabajo que realizan los ingenieros industriales en el proceso de desarrollo del flujo de procesos con la ingeniería que llevan a cabo los desarrolladores utilizando para ello los diagramas del Lenguaje Unificado de Modelado (UML), de esta forma con el paso del tiempo y con la ayuda de todos se podrá realizar una óptima ingeniería de software en un menor tiempo.

Por tanto se aborda como **objetivo concreto**: exponer como queda reestructurado el proceso de desarrollo de software en la Universidad de las Ciencias Informáticas, al decidirse pasar parte del trabajo al ingeniero industrial. Para moldear estas investigaciones se desarrollarán ejemplos concretos de cómo se hace la transición entre los diagramas de flujo hacia los diagramas y modelos del UML que usa el desarrollador de software.

En este documento de tesis se plasman los resultados del estudio realizado con un grupo de procesos, los cuales aparecen ya llevados a diagramas de UML utilizando para esto la fundamentación teórica producto de las indagaciones en este campo. Se incluyen los conceptos relacionados con la Ingeniería de Procesos y el resultado de todas las investigaciones realizadas y las conclusiones a las que se llegaron. Finalmente se dejan algunas recomendaciones para el desarrollo futuro del mismo.

INTRODUCCIÓN.	1
CAPÍTULO I. FUNDAMENTACIÓN DEL TEMA.	5
1.1 INTRODUCCIÓN.	5
1.2 PROCESOS.	5
1.2.1 <i>Papel e importancia de los procesos en la empresa.</i>	5
1.2.2 <i>Mejora de procesos.</i>	6
1.2.3 <i>Reingeniería de Procesos.</i>	7
1.2.4 <i>Identificar los procesos.</i>	8
1.3 LOS DIAGRAMAS DE FLUJO.	8
1.3.1 <i>Cuándo se utilizan los diagramas de flujo.</i>	9
1.3.2 <i>Cómo se utilizan los diagramas de flujo.</i>	10
1.3.3 <i>Símbolos empleados.</i>	11
1.3.4 <i>Ejemplo de Diagrama de Flujo.</i>	11
1.3.5 <i>Importancia de los diagramas de flujo.</i>	13
1.4 UML.	13
1.4.1 <i>La concepción de UML.</i>	14
1.4.2 <i>Diagramas de UML.</i>	14
1.4.3 <i>UML no es un método.</i>	15
1.4.4 <i>Importancia de UML.</i>	16
1.5 EL PROCESO UNIFICADO DE DESARROLLO (RUP).	16
1.6 HERRAMIENTAS UTILIZADAS.	18
1.7 EL OBJETO DE ESTUDIO.	18
1.7.1 <i>Problema.</i>	18
1.7.2 <i>Situación Problemática.</i>	18
1.8 PROPUESTA DE SOLUCIÓN.	19
1.9 FUNDAMENTACIÓN DE LOS OBJETIVOS PROPUESTOS.	19
1.9.1 <i>Objetivo General.</i>	19
1.9.2 <i>Objetivos Específicos.</i>	19
1.10 CONCLUSIONES.	20
CAPÍTULO 2. TENDENCIAS Y TECNOLOGÍAS ACTUALES A CONSIDERAR.	21
2.1 INTRODUCCIÓN.	21
2.2 HERRAMIENTAS DE DIAGRAMAS DE FLUJO DE PROCESOS.	21
2.2.1 <i>ConceptDraw.</i>	21
2.2.2 <i>SmartDraw.</i>	23
2.2.3 <i>Edge Diagrammer.</i>	24
2.2.4 <i>WizFlow Flowcharter.</i>	26
2.2.5 <i>Microsoft Office Visio.</i>	27
2.3 HERRAMIENTAS DE MODELADO UML.	29
2.3.1 <i>Visual Paradigm.</i>	29
2.3.2 <i>Poseidón.</i>	31
2.3.3 <i>Altova Umodel.</i>	32
2.3.4 <i>Enterprise Architect.</i>	33
2.3.5 <i>Rational Rose.</i>	35
2.3.6 <i>Importancia de las herramientas Case en la actualidad.</i>	36
2.4 FUNDAMENTACIÓN DE LA HERRAMIENTA UTILIZADA EN LA UCI PARA REALIZAR LOS DIAGRAMAS DE FLUJO.	37
2.5 FUNDAMENTACIÓN DE LA HERRAMIENTA UTILIZADA EN LA UCI PARA EL MODELADO CON UML.	38
2.6 CONCLUSIONES.	39
CAPÍTULO 3. DESCRIPCIÓN DE LA PROPUESTA DISEÑADA.	40
3.1 INTRODUCCIÓN.	40

3.2	CÓMO SE REALIZAN LOS DIAGRAMAS DE FLUJO EN DETALLE.-----	40
3.3	CÓMO SE REALIZAN LOS DIAGRAMAS DE FLUJO ACTUALMENTE EN LA UCI. -----	46
3.4	INTERPRETACIÓN DE LOS DIAGRAMAS DE FLUJO.-----	50
3.4.1	<i>Objetivos de la interpretación.</i> -----	50
3.4.2	<i>Cómo se interpretan los diagramas de flujo que se crean actualmente en la universidad.</i> -----	50
3.5	DESCRIPCIÓN DEL PROCESO DE CONVERSIÓN A UML.-----	53
3.6	ESTRATEGIA DE CONTINUACIÓN.-----	64
3.6.1	<i>Diagramas de secuencia del sistema.</i> -----	64
3.6.2	<i>Diagramas de Colaboración.</i> -----	65
3.6.3	<i>Diagramas de estado.</i> -----	67
3.7	CONCLUSIONES.-----	69
CAPÍTULO 4. EJEMPLOS INFORMATIZADOS DE LA PROPUESTA DISEÑADA UTILIZANDO RUP.-----		70
4.1	INTRODUCCIÓN.-----	70
4.2	EJEMPLOS PRÁCTICOS YA ANALIZADOS.-----	70
4.2.1	<i>Proceso de consulta de producto.</i> -----	70
4.2.2	<i>Solicitud de producto.</i> -----	73
4.2.3	<i>Aprobación de solicitud.</i> -----	76
4.2.4	<i>Comprar productos básicos.</i> -----	77
4.2.5	<i>Entrega del mobiliario.</i> -----	80
4.3	CONCLUSIONES.-----	83
CAPÍTULO 5. ESTUDIO DE FACTIBILIDAD.-----		84
5.1	INTRODUCCIÓN.-----	84
5.2	BENEFICIOS TANGIBLES E INTANGIBLES.-----	84
5.3	FACTIBILIDAD PARA EL ÁREA DE INFORMATIZACIÓN.-----	85
5.4	CONCLUSIONES.-----	87
CONCLUSIONES GENERALES.-----		89
RECOMENDACIONES.-----		90
BIBLIOGRAFÍA.-----		91
GLOSARIO DE TÉRMINOS.-----		93
ANEXOS.-----		95

Figuras.

Figura 1 Método sistemático de mejora de procesos.	7
Figura 2: Representación de un diagrama de flujo.....	12
Figura 3 Representación gráfica de las etapas de RUP.	17
Figura 4: Ambiente de ConceptDraw.....	23
Figura 5: Ambiente de SmartDraw.....	24
Figura 6: Ambiente de Edge Diagrammer.....	25
Figura 7: Ambiente de Wizflow Flowcharter.	27
Figura 8: Ambiente de Microsoft Office Visio.	29
Figura 9: Ambiente de Visual Paradigm.....	30
Figura 10: Ambiente de Poseidón.....	31
Figura 11: Ambiente de Altova Umodel.....	33
Figura 12: Ambiente de Enterprise Architect.....	35
Figura 13: Representación de entidades externas.....	41
Figura 14: Representación de las actividades.	42
Figura 15: Ejemplo del uso del almacén de datos.....	43
Figura 16: Ejemplo del uso del flujo de datos.	44
Figura 17: Representación de los niveles del sistema.....	45
Figura 18: Representación del diagrama de contexto.	46
Figura 19: Símbolo de representación de entrada/salida de un proceso.	47
Figura 20: Símbolos de comienzo del proceso.	47
Figura 21: Representación de tareas en paralelo.....	48
Figura 22: Símbolo de documento con copia.....	48
Figura 23: Símbolo de documento y varias copias o varios documentos.....	48
Figura 24: Ejemplo de un modelo de flujo de procesos.....	49
Figura 25: Diagrama de flujo de procesos para la compra de productos.....	55
Figura 26: Diagrama de actividades sin marcos de responsabilidad del CU Comprar Producto. ...	58
Figura 27: Modelo conceptual del dominio del punto de venta.	60
Figura 28: Diagrama de secuencia del sistema para el caso de uso Comprar productos.....	65
Figura 29: Diagrama de colaboración introducir Producto.	66
Figura 30: Diagrama de estado para el caso de uso comprar producto.....	67
Figura 31: Dependencias de los artefactos durante la fase de construcción.....	68
Figura 32: Diagrama de flujo para el proceso Consulta de Producto.....	71
Figura 33: Diagrama de actividades para el caso de uso Consultar Existencia de Producto.....	72
Figura 34: Modelo de Objetos del negocio para el CU Consultar Existencia de Producto.....	72
Figura 35: Diagrama de flujo para el proceso Solicitud de producto.....	74
Figura 36: Diagrama de actividades para el caso de uso Solicitar Productos al Almacén.....	75
Figura 37: Modelo de objetos para el caso de uso Solicitar Productos al Almacén.	75
Figura 38: Diagrama de flujo para el proceso de Aprobación de solicitud.	76
Figura 39: Diagrama de actividades para el caso de uso aprobar Solicitud.	77
Figura 40: Modelo de objetos del negocio para el caso de uso aprobar Solicitud.....	77
Figura 41: Diagrama de flujo del proceso comprar productos básicos.	78

Figura 42: Diagrama de actividades para el caso de uso ordenar compra de productos básicos.	78
Figura 43: Modelo de objetos para el caso de uso ordenar compra de productos básicos.	79
Figura 44: Diagrama de flujo para el proceso de entrega de mobiliario.	81
Figura 45: Diagrama de actividades para el caso de uso entregar mobiliario.	82
Figura 46: Modelo de objetos para el caso de uso entregar mobiliarios.	82
Figura 47: Elementos claves de cada proyecto.	85

Tablas.

Tabla 1 Simbología de los diagramas de flujo.	11
Tabla 2: Descripción de CU. Comprar Productos.....	63
Tabla 3: Explicación de la secuencia.....	65
Tabla 4: Descripción de CU. Ordenar compra de PB	80
Tabla 5: Descripción de CU. Entregar Mobiliario.....	83

Introducción.

Actualmente el desarrollo de software juega un papel fundamental en todo el mundo. La implementación de herramientas cada vez más eficaces que permitan innovarlo es una de las tareas de la comunidad científica y universitaria a nivel mundial.

Hoy en día se emplea la Gestión por Procesos como una forma de administrar la información de las organizaciones basándose en sus actividades, es por esto que se ha decidido emplear esta técnica para un mayor avance en el ámbito del desarrollo de software.

El principal objetivo de la Universidad de Ciencias Informáticas, es llegar a abrirse camino en el amplio mundo de la industria del software que se ha vuelto tan importante para tener un negocio eficiente y que está poblado de una gran competencia. Es por eso que para lograr este propósito se ve obligada a renovar las formas de realizar con eficiencia y calidad esta tarea.

Para esto se provee al estudiante a lo largo de sus años de estudio de una gran habilidad para desarrollar y llevar a cabo la impetuosa tarea de realizar proyectos de una gran envergadura usando las facilidades de algunas herramientas de desarrollo que se tienen al alcance de la comunidad universitaria gracias al perfeccionamiento de las tecnologías de la Información y las Telecomunicaciones.

En la UCI se lleva a cabo la confección de todos los flujos de procesos con el objetivo de su posterior desarrollo, trabajo que se desarrolla por la Dirección de Informatización, fundamentalmente por ingenieros industriales que tienen esta tarea utilizando diversas técnicas de recopilación de datos y en conjunto con los informáticos se diseñan los softwares correspondientes para su futura aplicación. También existen algunas aplicaciones de Gestión de Software cuyo propósito es facilitar a los usuarios el desarrollo de una aplicación o proyecto dado.

El proceso de modelación del negocio, para la posterior captura de requisitos que se utiliza actualmente no es en varias ocasiones efectivo. Uno de los mayores problemas existentes es el hecho de que a veces dichas tareas se demoran mucho, y en muchas ocasiones se debe preparar personal para que se encarguen de la misma, y a pesar de lo antes expuesto no es aclarado todo lo necesario para comenzar a desarrollar el software, puesto que siempre se obvian cosas por no saber la forma correcta de obtener la información que se necesita del cliente.

La **situación problemática** surge a raíz de que las situaciones antes mencionadas provocaron la reestructuración del proceso de desarrollo de software en la UCI asignándole la labor de modelación de procesos a los ingenieros industriales, por lo que es necesario investigar como vincular los diagramas de

flujo de procesos desarrollados por los ingenieros industriales con el ciclo de desarrollo de software que realizan los desarrolladores.

Actualmente el proceso de trasladar la información obtenida de los diferentes procesos de la UCI hacia una metodología de desarrollo como RUP es bastante complicado, pues requiere primeramente entender las áreas que se pueden informatizar y darle prioridades, trabajo este que se detectó era más sencillo llevarlo a cabo por los ingenieros industriales, quienes después de hacer esta tarea automatizan los procesos, y elaboran los diagramas de flujo. De aquí que el **problema** consiste en ¿Cómo facilitar la interacción entre el trabajo que realizan los ingenieros industriales y los ingenieros informáticos de la Universidad de las Ciencias Informáticas para lograr una mayor eficiencia en el proceso de desarrollo de software?

Conjuntamente con el desarrollo del mercado del software se han implementado un cúmulo de técnicas ingenieriles que se encargan de facilitar el arduo trabajo que realizan los desarrolladores. Entre estas herramientas se encuentra la metodología RUP. La misma desempeña un papel fundamental pues mediante ella se obtienen los lenguajes documentales, los que contienen disímiles diagramas mediante los cuáles se desarrolla el producto en cuestión.

El querer incrementar el número de clientes en un negocio determinado es una tarea que le concierne a todas las empresas pues a cada minuto que pasa los usuarios requieren mejores sistemas con las más diversas tecnologías para no perder el lugar que ocupan en el mercado, o para intentar utilizar algún tipo de estrategia que revolucione su negocio, es aquí donde entran a jugar un papel ponderante los desarrolladores de software. Sin embargo para poder sacar provecho de todas estas posibilidades que se brindan es necesario encontrar la forma de realizar estas tareas de manera eficiente y rápida, teniendo en cuenta responder a las necesidades de los usuarios.

Este trabajo se propone desarrollar la fundamentación teórica necesaria para lograr transformar todo el flujo de procesos desarrollado por los ingenieros industriales en diagramas con los cuáles puedan trabajar los desarrolladores a la hora de informatizar determinados procesos dentro de un negocio dado.

Por tanto el **objeto de estudio** del mismo es el Proceso de Desarrollo de Software en la Universidad de las Ciencias Informáticas.

Por otra parte el **campo de acción** será el flujo de procesos de la UCI, dentro de la ingeniería de procesos y su interrelación con el UML.

Como **Hipótesis** se parte de la idea de que si se desarrolla la fundamentación teórica necesaria para lograr unificar las labores de los ingenieros industriales y los desarrolladores de software, será posible desarrollar proyectos de mayor envergadura que tendrán mayor calidad y gran usabilidad.

El **objetivo general** del trabajo será desarrollar una investigación detallada de donde se extraigan como resultados un grupo de reglas a seguir que permitan controlar de manera centralizada la interrelación entre la información referente al flujo de procesos de la UCI, en el campo de la Ingeniería de Procesos, con el UML, ejemplificándolo mediante el Proceso Unificado de Desarrollo de Software.

De acuerdo a esta propuesta se derivan los siguientes **objetivos específicos**:

- I. Realizar un estudio sobre formas y mecanismos mediante los cuáles los ingenieros industriales llevan a cabo los diagramas de flujo de procesos.
- II. Investigar si este método novedoso de trabajo se ha llevado a cabo antes en algún lugar.
- III. Investigar a fondo los diagramas UML y en específico la metodología de desarrollo RUP.
- IV. Proponer el proceso para convertir los diagramas de flujo de procesos a diagramas conocidos de UML.

Para cumplir con dichos objetivos y resolver la situación problemática planteada, se proponen las siguientes **tareas**:

- I. Búsqueda exhaustiva sobre tipos de procesos y sus clasificaciones.
- II. Explicación de los diagramas de flujo de procesos.
- III. Análisis de cómo se encuentran en la arena internacional las tecnologías que se utilizan para realizar los diagramas de flujo procesos.
- IV. Explicación de qué es UML y cómo se emplea.
- V. Implementación de una metodología a seguir para convertir los diagramas de flujo de procesos en diagramas de UML.
- VI. Desarrollar ejemplos abarcadores de la metodología empleada.

Esta propuesta de trabajo es algo novedoso en la UCI, pero no obstante se pretende dejar una metodología muy útil a la altura de las exigencias actuales de producción de software, acorde con los estándares nacionales de catalogación de los procesos y los de diseño UML utilizando para ello los patrones de diseño.

Este documento está estructurado en 5 capítulos.

En el Capítulo 1 se realiza un estudio de los principales conceptos referentes a la Ingeniería de Procesos y al Proceso Unificado de Desarrollo de Software empleando el UML. Se mencionan los problemas fundamentales que generaron la necesidad del cambio; y como conclusión, se obtienen los objetivos generales y específicos a cumplir por el trabajo.

En el Capítulo2 se analizan las tecnologías que se utilizan en la actualidad para la construcción de diagramas de flujo de procesos y cómo se trabaja con los modelos UML, además de abordar las tendencias que existen sobre estos tópicos.

En el Capítulo3 se describe la propuesta de solución que se brinda en este documento, aquí se especifica que hay que tener un dominio estándar del UML.

El capítulo 4 se encarga de poner ejemplos prácticos de casos que ya hayan sido investigados y definidos con anterioridad por los ingenieros industriales, a los cuáles se les realizará la ingeniería de software correspondiente.

En el Capítulo5 se estudia si fue factible o no las investigaciones realizadas, analizándose los beneficios que produciría este cambio en la forma de actuar en la UCI.

Capítulo I. Fundamentación del Tema.

1.1 Introducción.

En este capítulo se brinda una visión general de los aspectos relacionados con la gestión de procesos y con el UML, los conceptos afines con el estudio de ambos y las características que poseen. Además se describen las originalidades fundamentales que se deben dominar para entender la propuesta teórica brindada.

1.2 Procesos.

Un proceso se define como el conjunto de actividades secuenciales que realizan una transformación de una serie de entradas (material, mano de obra, capital, información, etc.) en las salidas deseadas (bienes y/o servicios) añadiendo valor.

1.2.1 Papel e importancia de los procesos en la empresa.

Los procesos son posiblemente el elemento más importante y extendido en la gestión de las empresas innovadoras, especialmente de las que basan su sistema de gestión en la calidad total, es por esto que se consideran actualmente como la base operativa de gran parte de las organizaciones y gradualmente se van convirtiendo en la base estructural de un número creciente de compañías. Esta tendencia llega después de las limitaciones puestas de manifiesto en diversas soluciones organizativas, en sucesivos intentos de aproximar las estructuras empresariales a las necesidades de cada momento. La importancia de los procesos fue apareciendo de forma progresiva en los modelos de gestión empresarial. No irrumpieron con fuerza como la solución, sino que se les fue considerando poco a poco como unos medios muy útiles para transformar la empresa y adecuarse al mercado. Los procesos están permanentemente sometidos a revisiones para responder a dos motivos distintos. Por un lado, desde un punto de vista interno, todo proceso es mejorable en sí mismo, siempre se encuentra algún detalle o alguna secuencia que aumenta su rendimiento en aspectos de la productividad de las operaciones o de disminución de defectos. Por otro lado, los procesos han de variar para adaptarse a los requisitos cambiantes de mercados, clientes y nuevas tecnologías.

Admitida esta necesidad de revisión y cambio, la empresa ha de buscar, en cada caso, el ritmo adecuado de cambio. Si el cambio puede ser gradual, el método recomendado, experimentado en miles y miles de empresas con buenos resultados, es la mejora progresiva. Por el contrario, si la empresa ha perdido su posición competitiva y necesita mejoras espectaculares en tiempos cortos, tendrá que recurrir a la reingeniería.

1.2.2 Mejora de procesos.

La experiencia japonesa, sobre todo en los años setenta y ochenta, con sus métodos de trabajo en equipo y la participación de todo su personal en las mejoras empresariales, popularizó las ventajas obtenidas en la revisión y retoque continuo de los procesos empresariales.

Este método se basa en el recorrido de una serie de pasos o etapas, desde la detección de un problema o de una posibilidad de mejora, pasando por un estudio en busca de posibles perfeccionamientos o soluciones, la elección de la solución o conjunto de soluciones que parecen idóneas, hasta llegar a su implantación y a la medida de las mejoras conseguidas. El diagrama de la figura 1 resume las etapas de este método y se observa que el rasgo más característico que se puede extraer de su uso es el continuo recurso a las medidas, a los datos objetivos, para la detección de los puntos a mejorar, confirmar el hallazgo de la causa real de los defectos detectados, corroborar que la solución adoptada es la apropiada y posteriormente cuantificar el nivel de mejora alcanzado.

Con él, numerosas empresas han conseguido incrementos de productividad del orden del 5 al 15 por ciento en determinados procesos en plazos inferiores a un año, bien sea por mejora de los rendimientos, por disminución de defectos o por una combinación de ambos.

FIGURA 1
MÉTODO SISTEMÁTICO DE MEJORA DE PROCESOS

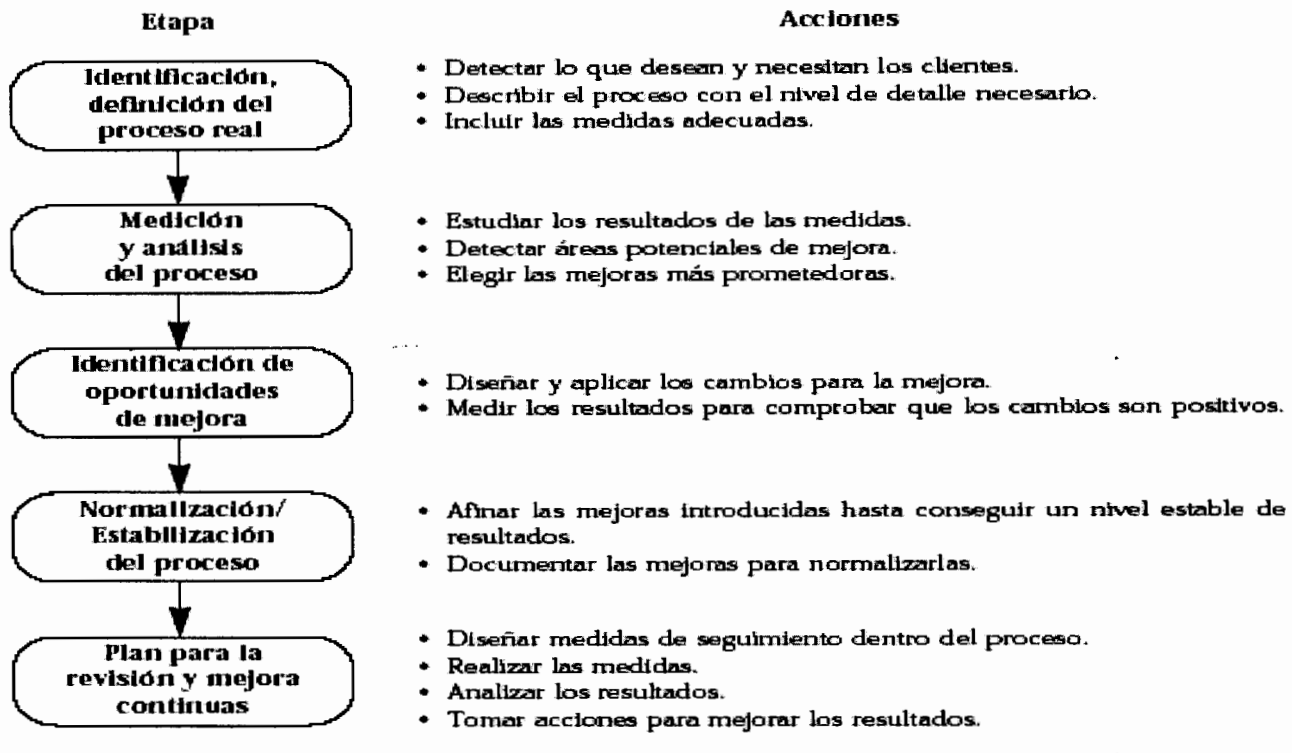


Figura 1 Método sistemático de mejora de procesos.

1.2.3 Reingeniería de Procesos.

La reingeniería de procesos supone un cambio radical en la forma de operar de la empresa, se puede definir como la reconsideración fundamental y el rediseño radical de los procesos de la empresa para conseguir mejoras espectaculares en medidas críticas, actuales, de resultados o rendimiento, como pueden ser los costos, la calidad, el servicio y la rapidez[Zaratiegui99]. Implica normalmente enfrentarse a los cambios por adoptar con la mente limpia de la historia pasada; se recomienda pensar en la situación a la que se aspira para, a partir de ahí, idear y diseñar los medios necesarios, sin sentirse atados por lo que se está haciendo, que supondrá un lastre a las nuevas ideas. El método se puede introducir en teoría proceso a proceso, pero implica cambios profundos de mentalidad, por lo que en la práctica se cambian bloques enteros de procesos relacionados, o todos los procesos básicos de una unidad de negocio, o bien directamente todos los procesos claves de la empresa. La reingeniería de procesos y de empresas ha tomado mucho ímpetu en los años noventa, pero ya se aplicaba con anterioridad, siempre que las

circunstancias lo exigiesen. Un ejemplo muy conocido es el sistema automatizado de reserva de billetes implantado por American Airlines, al que se adhirieron muchas otras compañías aéreas, y que alteró la forma de funcionar de las compañías aéreas y las agencias de viaje. Muchas veces esta necesidad de cambio se debe a saltos cualitativos introducidos por la competencia, que obligan a realizar un cambio comparable o más profundo debido a la posibilidad de quedar eliminado de un determinado mercado. Y para llevar a cabo la transformación con éxito es casi obligado (como en el caso de American Airlines) recurrir a las últimas posibilidades que ofrecen en cada momento las nuevas tecnologías.

1.2.4 Identificar los procesos.

Los procesos dentro de una empresa determinada van a tener distintas clasificaciones, debido a la importancia o papel que jueguen dentro del negocio. Existe diversidad de criterios en la clasificación de los mismos, pero se coincide en que hay 3 clasificaciones generales:

- I. Procesos estratégicos: son aquellos que proporcionan directrices a todos los demás procesos y son realizados por la dirección o por otras entidades. Se suelen referir a las leyes o normativas aplicables al producto o servicio y que no son controladas por el mismo.
- II. Procesos fundamentales: Atañen a diferentes áreas de la fabricación del producto o prestación del servicio y tienen impacto en el cliente creando valor para éste. Son las actividades esenciales del producto o servicio, su razón de ser.
- III. Procesos de soporte: Son los procesos que realizan otras áreas de la compañía y que ayudan en el momento de realizar los procesos fundamentales.

1.3 Los diagramas de flujo.

Un diagrama de flujo es una representación pictórica de los pasos en un proceso, es muy útil para determinar cómo funciona realmente el proceso para producir un resultado. El resultado puede ser un producto, un servicio, información o una combinación de los tres. Al examinar cómo los diferentes pasos en un proceso se relacionan entre sí, se puede descubrir con frecuencia las fuentes de problemas potenciales. Los diagramas de flujo se pueden aplicar a cualquier aspecto del proceso desde el flujo de materiales hasta los pasos para hacer la venta u ofrecer un producto. Por otra parte si se detallan bien llegan a describir la mayoría de los pasos en un proceso. Con frecuencia este nivel de detalle no es necesario, pero cuando se necesita, el equipo completo normalmente desarrollará una versión de arriba

hacia abajo; luego grupos de trabajo más pequeños pueden agregar niveles de detalle según sea necesario durante el proyecto.

1.3.1 Cuándo se utilizan los diagramas de flujo.

Cuando un equipo necesita ver cómo funciona realmente un proceso completo se hace necesario utilizar los diagramas de flujo. Este esfuerzo con frecuencia revela problemas potenciales tales como cuellos de botella en el sistema, pasos innecesarios y círculos de duplicación de trabajo. Algunas aplicaciones comunes son:

Definición de proyectos:

- I. Identificar oportunidades de cambios en el proceso.
- II. Desarrollar estimados de costos de mala calidad.
- III. Identificar organizaciones que deben estar representadas en el equipo.
- IV. Desarrollar una base común de conocimiento para los nuevos miembros del equipo.
- V. Involucrar a trabajadores en los esfuerzos de resolución de problemas para reducir la resistencia futura al cambio.

Identificación de las causas principales del cambio:

- I. Desarrollar planes para reunir datos.
- II. Generar teorías sobre las causas principales.
- III. Discutir las formas de estratificar los datos para el análisis e identificar las causas principales.
- IV. Examinar el tiempo requerido para las diferentes vías del proceso.

Diseño de Soluciones:

- I. Describir los cambios y efectos potenciales en el proceso.
- II. Identificar las organizaciones que serán afectadas por los cambios propuestos.

Aplicación de soluciones:

- I. Explicar a otros el proceso actual y la solución propuesta.
- II. Superar la resistencia al cambio demostrando cómo éstos simplificarán el proceso.

Control (Retener las Ganancias):

- I. Revisar y establecer controles para posteriormente monitorear el proceso.
- II. Auditar el proceso periódicamente para asegurar que se están siguiendo los nuevos procedimientos.
- III. Entrenar a nuevos empleados.

1.3.2 Cómo se utilizan los diagramas de flujo.

La metodología para preparar un diagrama de flujo es la siguiente:

- I. Analizar cómo se pretende utilizar el diagrama de flujo. Exhibir esta hoja en la pared y consultarla en cualquier momento para verificar que su Diagrama de Flujo es apropiado para las aplicaciones que se pretenden.
- II. Posteriormente hay que determinar el nivel de detalle requerido.
- III. Después de establecer los límites del proceso, enumerar los resultados y los clientes en el extremo derecho del diagrama.
- IV. Utilizando los símbolos apropiados para el Diagrama de Flujo, presentar las respuestas como los primeros pasos en el diagrama.
- V. Para cada entrada, hacer preguntas cómo:
 - ¿Quién recibe la entrada?
 - ¿Qué es lo primero que se hace con la entrada?
- VI. Documentar cada paso en la secuencia, empezando con el primer (o último) paso. Para cada paso, hacer preguntas como:
 - ¿Qué produce este paso?
 - ¿Quién recibe este resultado?
 - ¿Qué pasa después?
 - ¿Alguno de los pasos requiere de entradas que actualmente no se muestran?
- VII. Continuar la construcción del diagrama hasta que se conecten todos los resultados (salidas) definidos en el extremo derecho del diagrama. Si se encuentra un segmento del proceso que es extraño para todos en el salón, se deberá tomar nota y continuar haciendo el diagrama.
- VIII. Revisión.
- IX. Determinar oportunidades.

1.3.3 Símbolos empleados.

Aunque hay literalmente docenas de símbolos especializados utilizados para hacer diagramas de flujo, se utilizan con más frecuencia los siguientes:

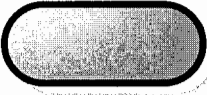
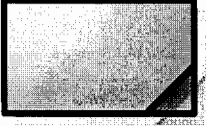
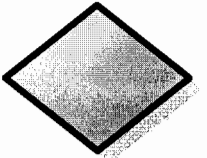
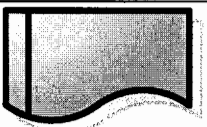
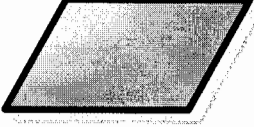

	Representa el inicio y fin de un programa. También puede representar una parada o interrupción programada que sea necesaria realizar en un programa.
	Representa cualquier tipo de operación que pueda originar cambio de valor, formato o posición de la información almacenada en memoria, operaciones aritméticas, de transformaciones, etc.
	Toma de decisiones. Indica operaciones lógicas o de comparación entre datos (normalmente dos) y en función del resultado de la misma determina (normalmente si y no) cual de los distintos caminos alternativos se debe seguir.
	Representación de documentos. En ocasiones si se quieren sacar copias del documento, se representan 2, o más símbolos de este tipo conectados.
	Elementos de Entrada y de Salida del proceso. Cualquier tipo de introducción de datos en la memoria desde los periféricos o registro de información procesada en un periférico.
	Este símbolo se emplea a la hora de especificar que existe una conexión con una base de datos. O sea que hay almacenamiento de datos.

Tabla 1 Simbología de los diagramas de flujo.

1.3.4 Ejemplo de Diagrama de Flujo.



Diagrama de flujo

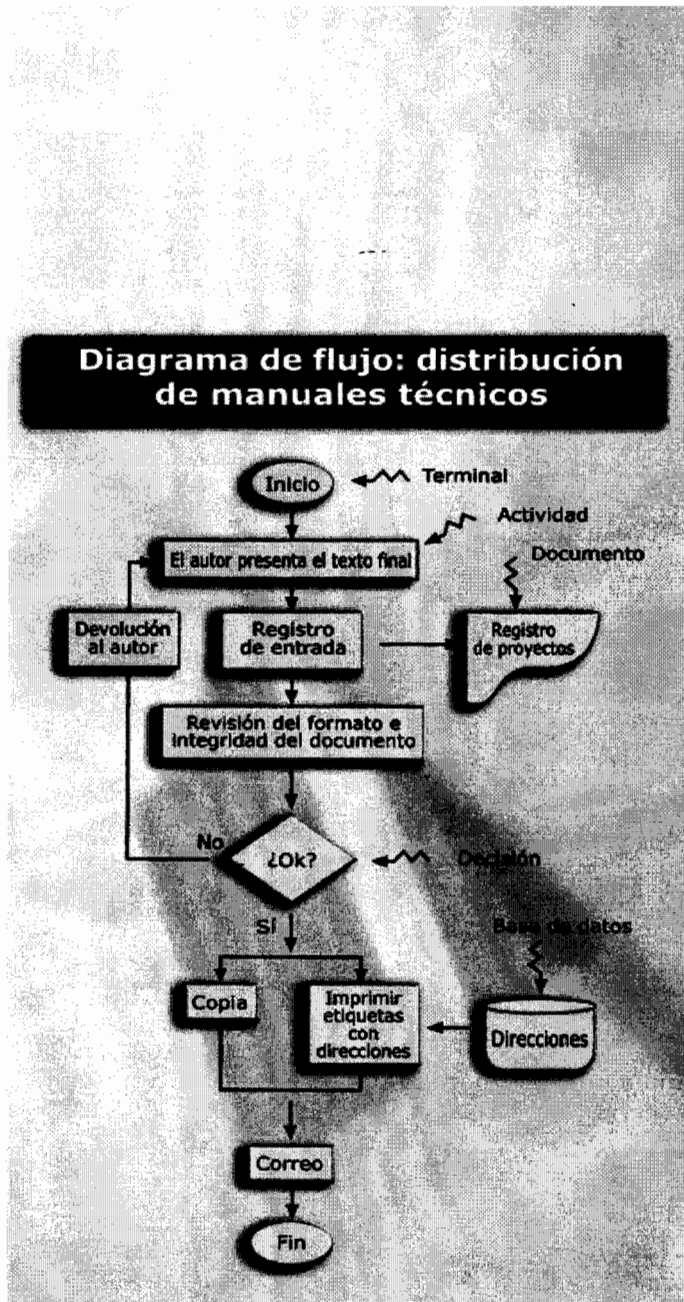


Figura 2: Representación de un diagrama de flujo.

1.3.5 Importancia de los diagramas de flujo.

Si un diagrama de flujo se construye de forma apropiada y refleja el proceso de la manera que realmente opera, todos los miembros del equipo poseerán un conocimiento común y exacto del funcionamiento del proceso. Adicionalmente, el equipo no necesitará invertir el tiempo y la energía en observarlo físicamente cada vez que se quiera identificar problemas para trabajar, discutir teorías sobre las causas principales, examinar el impacto de las soluciones propuestas o discutir las formas para mantener las mejoras. Los diagramas de flujo pueden ayudar a un equipo en su tarea de diagnóstico para lograr mejoras. Uno de sus usos es el de ayudar a un equipo a generar teorías sobre las posibles causas principales de un problema. El equipo que investiga un problema redacta una descripción del problema en un pedazo pequeño de papel y lo pega en el diagrama de flujo en el punto donde se ha detectado el problema. Luego se discute cada uno de los pasos en el proceso antes del punto donde se detectó el problema, y se producen teorías sobre las cosas que podrían salir mal en el paso que lo causó. El diagrama de flujo ayuda a examinar cada paso del proceso de forma sistemática a medida que produce teorías sobre las posibles causas principales del problema. Otra de sus utilidades es la de ayudar a identificar las formas apropiadas para separar los datos para su análisis. Por ejemplo, considérese el problema de analizar los tiempos de reparación. Una rápida revisión del diagrama de flujo puede sugerir un número de grupos posibles que pueden explicar el tiempo que se necesita para hacer una reparación.

1.4 UML.

El éxito de los proyectos de desarrollo de aplicaciones o sistemas se debe a que sirven como enlace entre quién tiene la idea y el desarrollador. El UML es una herramienta que cumple con esta función, ya que le ayuda a capturar la idea de un sistema para comunicarla posteriormente a quien está involucrado en su proceso de desarrollo; esto se lleva a cabo mediante un conjunto de símbolos y diagramas. Cada diagrama tiene fines distintos dentro del proceso de desarrollo.

UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. Por otra parte es una consolidación de muchas de las notaciones y conceptos más usados orientados a objetos. Empezó como una consolidación del trabajo de Grady Booch, James Rumbaugh, e Ivar Jacobson, aunque ya en la actualidad cuenta con innumerables seguidores.

1.4.1 La concepción de UML.

Durante los ochenta y principios de los noventa Grady Booch, James Rumbaugh, e Ivar Jacobson trabajaban en solitario desarrollando notaciones para el análisis y diseño de sistemas orientados a objetos. Los tres llegaron por separado a obtener bastante reconocimiento.

Booch había escrito "Object-Oriented Analysis and Design with Applications" un libro de referencia en el análisis y diseño orientado a objetos desarrollando su propia notación.

Por su parte James Rumbaugh había desarrollado su propia notación de diseño orientado a objetos llamada Object Modeling Technique (OMT) en su libro "Object Oriented Modelling and Design". Por otro lado Jacobson se había revelado como un visionario del análisis (padre de los casos de uso) y sobre todo del diseño orientado a objetos, sorprendiendo a todo el mundo con "Object-Oriented Software Engineering: A Use Case Driven Approach".

A mediados de los noventa empezaron a intercambiar documentos y trabajar en conjunto produciendo grandes avances en el modelado de sistemas orientados a objetos. En 1994 Rational contrató a Rumbaugh en donde ya trabajaba Booch, un año después Jacobson se unía a ellos en Rational Software Corporation. Los diagramas del UML empezaron a circular en la industria del software y las reacciones resultantes trajeron consigo considerables modificaciones. Conforme diversos corporativos vieron que el UML era útil a sus propósitos, se conformó un consorcio del UML. Entre los miembros se encuentran DEC, Hewlett-Packard, Intellicorp, Microsoft, Oracle, Texas Instruments y Rational. En 1997 el consorcio produjo la versión 1.0 del UML y lo puso a consideración del Grupo de administración de objetos (OMG) como respuesta a su propuesta para un lenguaje de modelado estándar.

El consorcio aumentó y generó la versión 1.1, la cuál se puso nuevamente a consideración del OMG. El grupo adoptó esta versión a finales de 1997. El OMG se encargó de la conservación del UML y produjo otras dos revisiones en 1998. El UML ha llegado a ser el estándar de facto en la industria del software, y su evolución continúa.

1.4.2 Diagramas de UML.

Los elementos de UML se muestran mediante diagramas que presentan múltiples vistas del sistema, ese conjunto de vistas son conocidos como modelos. UML presenta varios diagramas donde cada uno representa un aspecto del sistema. De ahí que varios investigadores según sus criterios y puntos de vista mencionan qué diagramas emplear en el desarrollo de los sistemas de información; sin llegar a mencionar cuáles son los diagramas más adecuados en las distintas etapas de desarrollo del Proceso Unificado.

Dado un sistema a desarrollar no es necesario emplear todos los diagramas; para sistemas sencillos un diagrama de clases junto con un par de diagramas de actividades e interacción sería suficiente, asimismo si los sistemas son complejos requieren de la utilización de más diagramas, debido a que necesitan etapas incrementales e iterativas (ciclos de desarrollo) en el análisis, diseño e implementación. Aquí se recogen los tipos de diagramas con que cuenta el Lenguaje Unificado de Modelado.

- I. Diagramas de Casos de Uso para modelar los procesos del negocio.
- II. Diagramas de Secuencia para modelar el paso de mensajes entre objetos.
- III. Diagramas de Colaboración para modelar interacciones entre objetos.
- IV. Diagramas de Estado para modelar el comportamiento de los objetos en el sistema.
- V. Diagramas de Actividad para modelar el comportamiento de los Casos de Uso, objetos u operaciones.
- VI. Diagramas de Clases para modelar la estructura estática de las clases en el sistema.
- VII. Diagramas de Objetos para modelar la estructura estática de los objetos en el sistema.
- VIII. Diagramas de Componentes para modelar componentes.
- IX. Diagramas de Implementación para modelar la distribución del sistema.

Una vez conocidas las herramientas, se deben usar en cada momento las más adecuadas a las necesidades. Esta no es una tarea fácil, pues hay que saber para qué sirven y qué limitaciones tienen unas y otras para conocer su utilidad. Pero se puede alcanzar este conocimiento con un poco de práctica y sentido común. Por otra parte se ha demostrado que el 80% de los problemas se pueden resolver usando tan solo el 20% del UML.

1.4.3 UML no es un método.

UML no es un método de desarrollo. No dice cómo pasar del análisis al diseño y de éste al código, no son una serie de pasos que te llevan a producir código a partir de unas especificaciones. UML al no ser un método es independiente del ciclo de desarrollo que vayas a seguir, puede encajar en un tradicional ciclo en cascada, o en un evolutivo ciclo iterativo e incremental o incluso en los métodos ágiles de desarrollo. Se plantea que más importante que seguir un proceso o método es que el desarrollador adquiera habilidades que le permitan crear un buen diseño, y que las organizaciones favorezcan este tipo de desarrollo de destrezas. Esto se logra dominando varios principios y heurísticas que permiten identificar y abstraer los objetos idóneos, asignándoles después responsabilidades. Por otra parte el UML tampoco estandariza un proceso oficial de desarrollo ya que esto posibilita aumentar las probabilidades de tener una aceptación generalizada de la notación estándar del modelado, sin la obligación de cambiar el método

a seguir, además la esencia de un proceso apropiado admite mucha variación y depende de las habilidades del personal, de la razón investigación-desarrollo, de la naturaleza del problema, de las herramientas y de muchos otros factores.

1.4.4 Importancia de UML.

En los principios de la computación, los programadores no realizaban análisis muy profundos sobre el problema por resolver. Si acaso, garabateaban algo en un papel. Con frecuencia comenzaban a escribir el programa desde el principio, y el código necesario se escribía conforme se requería. Aunque anteriormente esto agregaba un aura de aventura y atrevimiento al proceso, en la actualidad es inapropiado en los negocios de alto riesgo. Hoy en día, es necesario contar con un plan bien analizado. Un cliente tiene que comprender que es lo que hace un equipo de desarrolladores; además debe ser capaz de señalar cambios si no se han captado claramente sus necesidades (o si cambia de opinión durante el proceso). A su vez, el desarrollo es un esfuerzo orientado a equipos, por lo que cada uno de sus miembros tiene que saber que lugar toma su trabajo en la solución final, así como saber cuál es la solución en general.

Conforme aumenta la complejidad del mundo, los sistemas informáticos también deberán crecer en complejidad. La clave está en organizar el proceso de diseño de tal forma que los analistas, clientes, desarrolladores y otras personas involucradas en el desarrollo del sistema lo comprendan y convengan en las ideas. El UML proporciona tal organización. La idea es que así como un arquitecto le muestra un anteproyecto a la persona que lo contrató, se debe mostrar un plan de diseño al cliente. Tal plan de diseño debe ser el resultado de un cuidadoso análisis de las necesidades del mismo. La necesidad de diseños sólidos ha traído consigo la creación de una notación de diseño que los analistas, desarrolladores y clientes acepten como pauta. El UML es esa notación.

1.5 El Proceso Unificado de Desarrollo (RUP).

A través de la historia se han desarrollado varios modelos del proceso de software (paradigmas de desarrollo) cada uno con sus ventajas, desventajas y utilidad en algunos tipos de proyectos y problemas. Al igual que cualquier notación, el proceso unificado actúa como un modelo que puede adaptarse a cualquier tipo de proyecto y empresa (grandes y pequeñas). Las características del proceso unificado de modelado son:

- **Centrado en los Modelos:** Los diagramas son un vehículo de comunicación más expresivo que las descripciones en lenguaje natural. Se trata de minimizar el uso de descripciones y especificaciones textuales del sistema.
- **Guiado por lo casos de uso:** Los casos de uso reemplazan la antigua especificación funcional tradicional y constituyen la guía fundamental establecida para las actividades a realizar durante todo el proceso de desarrollo incluyendo el diseño, la implementación y las pruebas del sistema.
- **Centrado en la arquitectura:** La arquitectura involucra los elementos más significativos del sistema y está influenciada entre otros por plataformas software, sistemas operativos, manejadores de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados y requerimientos no funcionales.
- **Iterativo e incremental:** Se divide el proyecto en ciclos. Para cada ciclo se establecen fases de referencia. Permite una comprensión creciente de los requerimientos a la vez que se va haciendo crecer el sistema.

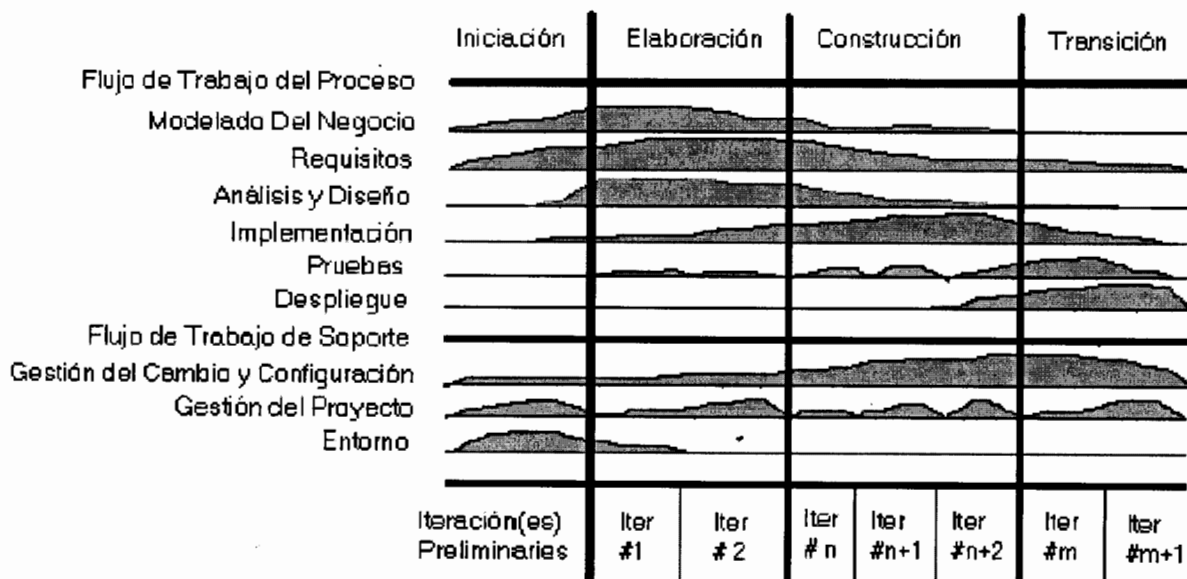


Figura 3 Representación gráfica de las etapas de RUP.

El gráfico que representa el RUP incluye las cuatro etapas importantes que son: la iniciación, elaboración, construcción y transición, las cuales muestran que para producir una versión del producto en desarrollo se aplican todas las actividades de ingeniería pero con diferente énfasis; en las versiones preliminares, como además indica la intuición, hay más énfasis en actividades de modelado del negocio, requisitos, análisis y diseño; conforme se producen versiones el énfasis pasa a las actividades de implementación, pruebas y despliegue.

1.6 Herramientas Utilizadas.

Existen varias herramientas Case visuales que permiten realizar el modelado del desarrollo de los proyectos, como son: el Analise, el Design y el Rational Rose. En este caso se utilizó para desarrollar el basamento teórico el Rational Rose.

Rational Rose es la herramienta de modelación visual que provee el modelado basado en UML. La Corporación Rational ofrece un Proceso Unificado (RUP) para el desarrollo de los proyectos de software, desde la etapa de Ingeniería de Requerimientos hasta la de pruebas. Para cada una de estas etapas existe una herramienta de ayuda en la administración de los proyectos.

1.7 El objeto de estudio.

La Universidad de las Ciencias Informáticas, institución surgida debido al amplio proceso político que está llevando a cabo la Revolución Cubana, tiene como objetivo prioritario y fundamental la formación integral y continua de profesionales para posteriormente iniciarlos en la tarea de la producción de software, siendo capaces de dar soluciones llenas de creatividad y dinamismo, para lo cual debe hacerse uso de las más innovadoras y diversas tecnologías. En busca de este fin, se provee al estudiante a lo largo de sus años de estudio de una gran capacidad cognoscitiva, enseñándole como principio básico el “aprender a aprender”.

1.7.1 Problema.

Actualmente la Universidad de las Ciencias Informáticas se encuentra informatizando todos sus procesos, y esto se complica en varias ocasiones pues requiere lograr entender correctamente lo que desea el cliente, para posteriormente implementar la infraestructura que permita este salto, y la mayor parte de las veces no se logra extraer del cliente toda la información que se necesita, es por esto que se ha decidido darle este trabajo a los ingenieros industriales, quienes posteriormente realizan diagramas de flujo de procesos con esta información. De aquí que el problema radica en: ¿Cómo facilitar la interacción entre el trabajo que realizan los ingenieros industriales y los ingenieros informáticos de la Universidad de las Ciencias Informáticas?

1.7.2 Situación Problémica.

La informatización de los procesos de la universidad es una tarea de primer orden, y en ocasiones se tarda mucho, además que muchos de los proyectos se están implementando ad-hoc ya que no se logra

todo lo que quieren los clientes, por lo que se ve la necesidad del control centralizado de la información referente a los procesos dentro del campo de la Ingeniería de Procesos y su interrelación con el Proceso Unificado de Desarrollo de Software, por lo cuál se hace indispensable la búsqueda de alguna manera productiva de mejorar este proceso, para posteriormente, si se demuestra su eficacia práctica, llevar esta metodología a toda la producción de software que se desea lograr en la UCI.

1.8 Propuesta de solución.

Después de realizar un análisis detallado, y determinar cuál es la situación actual existente se ve la necesidad de implementar una metodología entre los diagramas de flujo de procesos que llevan a cabo los ingenieros industriales con los diagramas de UML que utilizan los desarrolladores. De acuerdo a todo el análisis anterior sobre las deficiencias que presentaba la forma actual, se propone desarrollar una comunicación entre las partes involucradas

1.9 Fundamentación de los objetivos propuestos.

Debido al amplio crecimiento que está teniendo la industria del software y las tecnologías existentes, se hace más necesaria la rapidez con la que se deben informatizar las áreas, y para darle respuesta a la situación problemática planteada, se propone un conjunto de objetivos para cumplimentar la propuesta de solución planteada en la sección anterior.

1.9.1 Objetivo General.

Desarrollar una investigación detallada para obtener como resultado un grupo de reglas a seguir que permitan controlar de manera centralizada la interrelación entre la información referente al flujo de procesos de la UCI en el campo de la Ingeniería de Procesos con el UML ejemplificándolo mediante el Proceso Unificado de Desarrollo de Software.

1.9.2 Objetivos Específicos.

- I. Realizar un estudio sobre formas y mecanismos mediante los cuáles los ingenieros industriales llevan a cabo los diagramas de flujo de procesos.
- II. Investigar sobre si este método novedoso de trabajo se ha llevado a cabo antes en algún lugar.
- III. Investigar a fondo los diagramas UML y en específico la metodología de desarrollo RUP.

- IV. Proponer el proceso para convertir los diagramas de flujo de procesos a diagramas conocidos de UML.

1.10 Conclusiones.

En este capítulo se analizaron los conceptos fundamentales relacionados con el flujo de procesos y el UML y se detallaron las condiciones específicas y los problemas fundamentales que abarca el objeto de estudio, concluyéndose la necesidad de desarrollar una comunicación entre ingenieros industriales e ingenieros informáticos que sea fácil de analizar y comprender por las partes involucradas. Aunque hasta aquí se visualizan solo los conceptos que deben quedar claros, de ellos se puede extraer todo el conocimiento inicial para comprender la teoría que se desarrollará en los capítulos posteriores.

Por otra parte se dejó claro todos los objetivos que se seguirán para llegar a cumplir con el planteamiento que se propuso inicialmente.

Capítulo 2. Tendencias y Tecnologías actuales a considerar.

2.1 Introducción.

En este capítulo se analizarán las técnicas que se utilizan en la actualidad para la realización de los diagramas de flujo de procesos, así como para el trabajo con UML. Además de que se explicará lo que existe de esto internacionalmente. Por otra parte se abordará las tendencias evidentes mundialmente que justificarán el uso de varias de estas técnicas en la Universidad de las Ciencias Informáticas.

2.2 Herramientas de Diagramas de Flujo de Procesos.

Existen varias herramientas de reconocido renombre en el ámbito internacional que se dedican a la diagramación, y dentro de esta rama, se usan más específicamente para crear diagramas de flujo (flowcharts). Algunas de las más usadas son: ConceptDraw, smartdraw, Edge Diagrammer, Wizflow Flowcharter y Microsoft Office Visio. Estos programas, en general, no sólo permiten el dibujo de diagramas de procesos, sino también la introducción de tiempos aleatorios para cada actividad, y la distribución de probabilidades de entradas de clientes o transacciones, con lo que luego el diseñador puede simular el proceso y obtener estadísticas de utilización o de ocio de cada departamento, recurso humano o físico y determinar los cuellos de botella y los recursos subutilizados. Permiten también el uso de múltiples escenarios para prever fácilmente los efectos que tendrá cambiar una o más características del proceso.

2.2.1 ConceptDraw.

ConceptDraw es la familia de productos que más auge tomó en los inicios de los flujos de procesos, sus productos en la actualidad son ampliamente utilizados en el mundo de los negocios y para dibujos técnicos. Su uso en general es recomendado puesto que es multiplataforma. Entre la familia de estos productos, el más utilizado es ConceptDraw V, se dice que este producto logró nuevamente poner entre los punteros a esta compañía que ya estaba perdiendo mercado, con un conjunto de nuevas funciones y mejoras. Este software dispone de potentes capacidades de dibujo, extensas librerías de objetos y funciones de intercambio de datos avanzadas. Con este software se pueden crear casi todos los tipos de esquemas y diagramas. El trabajo con este producto no requiere aptitudes de diseño particulares. El programa permite a los usuarios concentrarse en la tarea y no en el proceso de dibujo. En particular, dispone de centenares de librerías con objetos ya listos para el uso, con el comportamiento inteligente, programados con las fórmulas necesarias para esta tarea. Por otra parte entre sus características

fundamentales se encuentran que ofrece un surtido de herramientas para el dibujo vectorial que se emplean para crear ilustraciones profesionales. Además, podrá crear estupendos diagramas con rapidez y facilidad. La "cuadrícula personalizable" y la herramienta "ajustar" ayudan a posicionar los objetos con gran precisión. Un conjunto de herramientas de alineamiento, distribución e igualación permiten automatizar operaciones con múltiples objetos para lo cuál se apoya en un número de capas ilimitado.

Entre las cualidades nuevas que presenta y que fueron capaces de hacerle recuperar el mercado se encuentran la posibilidad de contar con conectores inteligentes, lo que permite que las líneas se queden conectadas a los objetos y el usuario no necesita rediseñar diagramas cada vez que mueve un elemento. Cada objeto puede ser convertido en el conector. También presenta una interfaz flexible que no es más que una nueva apariencia excitante, barras de herramientas personalizables y diálogos flotantes que permiten acceder a las herramientas principales y configurarlas rápido y sin dificultad alguna. ConceptDraw V sigue el principio de "lo que ves es lo que obtienes" y apoya las funciones de antiescalonamiento, llevando el proceso de diagramación a nuevos niveles de calidad de presentaciones. Viene con docenas de colecciones de objetos comunes para ayudarle a crear varios tipos de diagramas, tales como diagramas de red, croquis de oficina, organigramas y otros. Es muy fácil crear sus propias herramientas de dibujo diseñando sus propios objetos o arrastrando los gráficos existentes a una nueva librería. Los asistentes orientados a diferentes tareas le ayudarán a crear documentos paso a paso. Además de lo antes expuesto permite copiar estilos de otros objetos, y crear y organizar sus propios estilos, apoya XML para Visio, permitiendo intercambiar los documentos con los usuarios de MS Visio, posibilita importar y exportar archivos a un gran número de formatos raster, vector, multimedia y textuales, facilitando el intercambio de datos con otras aplicaciones. Asimismo convierte a formatos gráficos, apoyándose en AutoCAD DXF, admite crear y redactar archivos MS PowerPoint, y exportar documentos en PDF y HTML.

Existen diversidad de criterios sobre las originalidades que posee, pero se converge en que las dos principales que tiene es que presenta objetos inteligentes que son altamente personalizables, lo que significa que puedes asignarles propiedades personalizadas, menús definidos por el usuario, crear vínculos a otros objetos, archivos o programas, y su particularidad vital es la capacidad de conectarse a cualquier base de datos compatible con ODBC y visualizar sus datos en diagramas ConceptDraw.

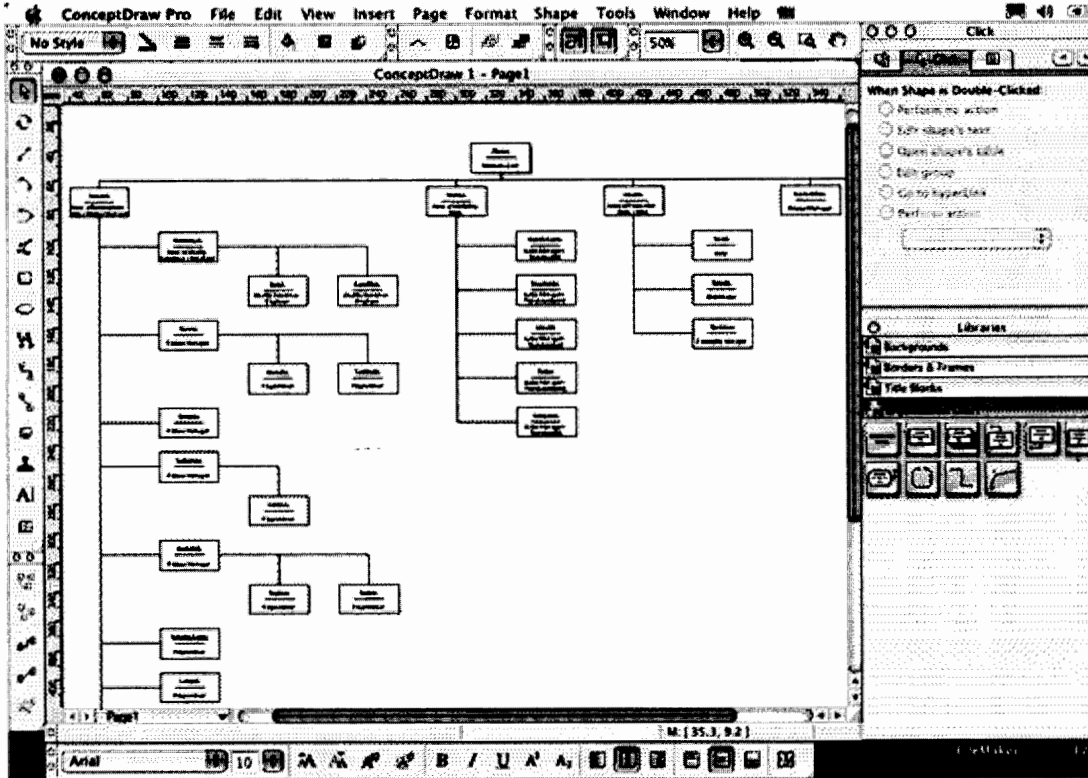


Figura 4: Ambiente de ConceptDraw

2.2.2 SmartDraw.

SmartDraw es un poderoso paquete de dibujo para crear diagramas de flujo, cuadros de organización, diagramas de redes, presentaciones, y otros tipos de diagramas. Es de aquellos programas para usuarios que necesitan crear ilustraciones profesionales de alta calidad y con rapidez, sin necesidad de invertir tiempo en aprender a usar complejas aplicaciones.

Incluye cientos de formas predibujadas y símbolos que fácilmente podrán pegarse dentro de un dibujo dado. Puedes dibujar amplios diagramas (50x50 pulgadas con cientos de símbolos) donde las formas y las líneas son automáticamente alineadas. Además si el diagrama tiene formas conectadas mediante líneas o curvas, cuando se mueve esa forma a otro lado del diagrama, no se pierde la conexión.

Es un cliente OLE, así que se pueden transferir los dibujos dentro de un procesador de texto u otros programas (por ejemplo al Microsoft Office, Word Perfect, o Lotus). Las funciones de "attach" permiten vincular cualquier archivo (inclusive una página Web o un programa) a una imagen de SmartDraw).

El programa soporta edición de texto para que las presentaciones tengan una óptima calidad, he incluye una librería de colores y estilos de sombras, y las imágenes pueden ser importadas o exportadas en formato WMF, BMP, PCX, GIF, JPG o TIFF.

También incluye un diseñador de tablas que le permiten al usuario crear formularios comerciales, tablas de datos, y formas con múltiples campos para ingreso de textos. Igualmente viabiliza la apertura de archivos de programas como Visio, MicroGrafx Flowcharter, Flowcharter PDQ, EasyFlow, entre otros.

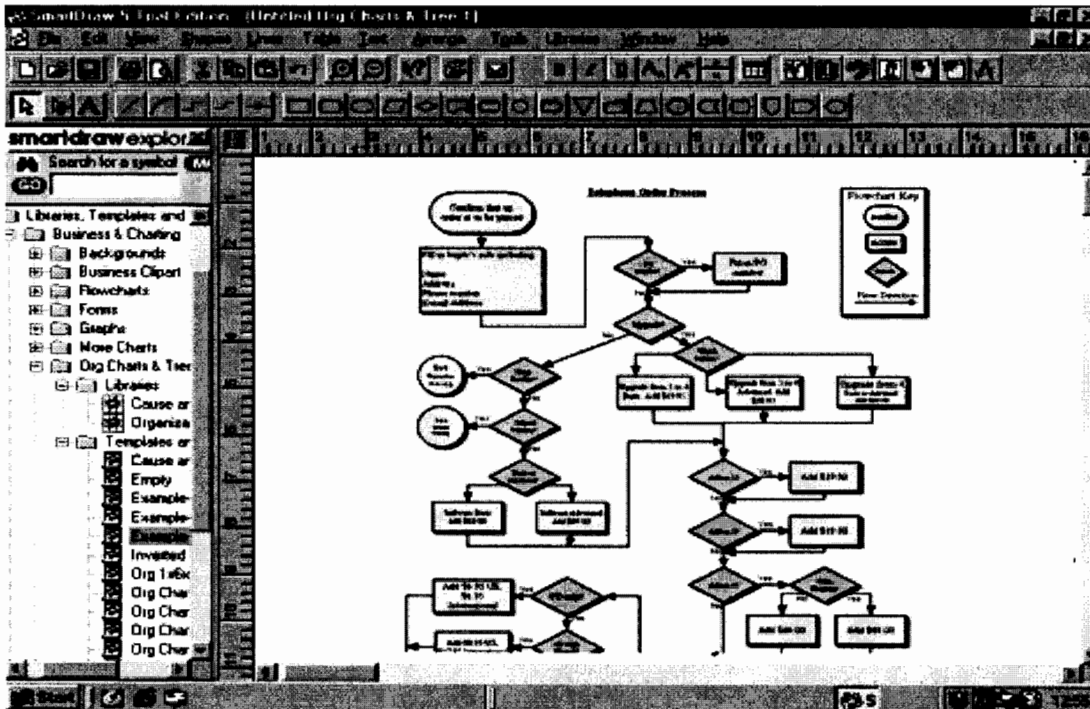


Figura 5: Ambiente de SmartDraw

2.2.3 Edge Diagrammer.

Ayuda a crear diagramas de flujo a través de una gran cantidad de herramientas prácticas que le otorgan mucha comodidad al usuario. Es un sistema versátil de dibujo para crear diagramas de flujo, diagramas organizativos, y cualquier tipo de diagrama por bloques similar. Permite trabajar con un simple objeto o grupo de objetos, dibujando cajas, símbolos o muchas formas conectadas con líneas de varios tipos. Se pueden introducir textos explicativos en cualquier localización, mediante el uso de una rejilla que permitirá tener el trabajo simétrico y alineado. También tiene opciones de zoom flexibles que ayudan a

cambiar fácilmente entre pantalla completa o una sección ampliada del diagrama. Permite incluso crear conjuntos propios de símbolos para usar con posterioridad.

Además, aparte de los cuadros rectangulares tradicionales, ahora se pueden fijar la forma del cuadro a rectangular redondeada, elíptica y romboidal. Soporta algoritmos avanzados para crear imágenes pequeñas y de calidad notable. Por otra parte posee soporte completo de los temas de Windows XP, barras de herramientas acoplables, función de imán de cuadros, soporte avanzado para plugins, mayor estabilidad del programa, selector visual de fuentes, posibilidad de usar GDI, dibujar flechas rellenas, introducir caracteres especiales usando un mapa de caracteres incorporado, y otros detalles menores. Los esquemas llevan una miniatura incrustada. Ahora programas como ACDSee, PicaView y otros muchos pueden previsualizar los esquemas sin abrir Edge Diagrammer. Una de sus características más importantes es que este programa a diferencia de otros dentro de esta rama, no necesita plugins de compresión JPG. A todo lo anteriormente planteado se le agrega que han sido implementadas nuevas propiedades de compresión JPG, como la posibilidad de cambiar el muestreo (subsampling) o insertar comentarios dentro de los archivos JPG. En su última versión ya se añade la posibilidad de elegir si se imprime el fondo o no. También controla el mejor aprovechamiento del papel para los distintos tipos de esquemas y corrige todos los fallos de impresión en versiones anteriores (imprime perfectamente el texto multilinea y los márgenes personalizados).

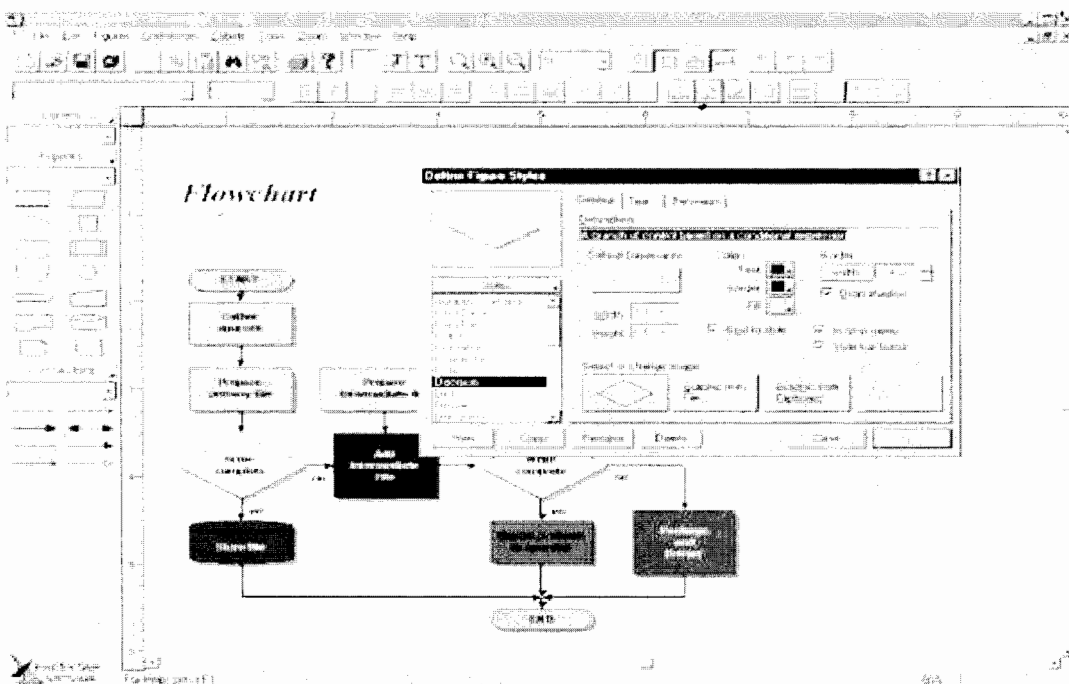


Figura 6: Ambiente de Edge Diagrammer.

2.2.4 WizFlow Flowcharter.

WizFlow Flowcharter es una utilidad para realizar diagramas de fácil uso y potentes prestaciones. Su interfaz está diseñada para la productividad, con lo que, mediante el método de arrastrar y soltar, se puede crear en pocos minutos el diagrama que se necesite. Incluye opciones muy atractivas como un zoom total, posicionamiento de las figuras en una rejilla, impresión multipágina, alineamiento, importación de gráficos y cliparts, exportación como gráficos para web, y cientos de elementos predefinidos.

Es un programa para dibujar y crear diagramas que te permite mostrar, diseñar y documentar una red local, página Web o cualquier otro sistema de comunicaciones. Permite mostrar componentes de dicho entorno, como las computadoras personales, estaciones de trabajo, servidores, y conectarlos con líneas y cables. Simplemente se seleccionan los componentes entre los símbolos predeterminados del archivo de imágenes e iconos, o incluso se puede añadir una propia imagen (BMP o JPG). Además se puede usar líneas simples, brascas, curvas, flechas, y complejas combinaciones para cualquier tipo de relación que se desee establecer. Ofrece un buen número de opciones y gráficos ya creados; permite insertar imágenes prediseñadas, seleccionar los colores de los gráficos, contornos, fuentes y de relleno y dispone de ajuste automático entre los diferentes elementos. Entendiendo la filosofía de trabajo de wizflow, se ve como es posible también introducir el diagrama en forma de sentencias o comandos para que éste lo dibuje automáticamente. Se ha demostrado producto de numerosos estudios que esta herramienta es de gran utilidad si los diagramas que se desean hacer son de moderada complejidad, ya que no sobrecarga la memoria con imágenes innecesarias, puesto que trabaja sobre texto claro posibilitando poder convertir un simple documento en un diagrama básico, suponiendo que dicho documento se encuentre en el formato adecuado para su lectura.

Entre las dificultades que presenta se encuentran que si se utiliza para crear diagramas de flujo, no admite tener procesos dentro de procesos, o sea, cada proceso debe ser tratado individualmente. Además de que no permite agregarle nuevas imágenes para crear nuevos tipos de diagramas. No obstante a lo anteriormente expresado, esta herramienta es el estándar más usado en la actualidad para uso académico ya que es fácil de emplear y tiene poca complejidad.

Se encuentra en desarrollo una nueva versión del producto, que se espera será más orientado a las empresas, y según se expresa contará con un mayor número de ventajas competitivas.

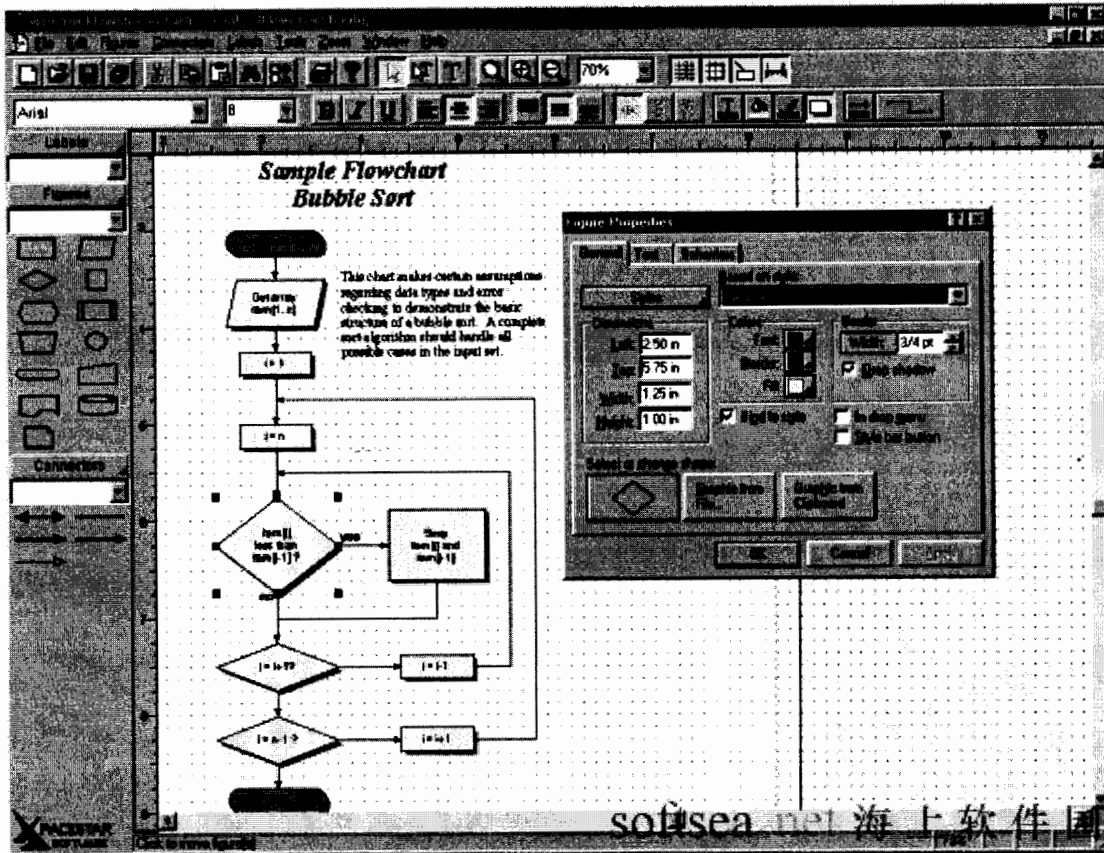


Figura 7: Ambiente de Wizflow Flowcharter.

2.2.5 Microsoft Office Visio.

Se trata de una aplicación con herramientas gráficas que ayuda a crear una gran variedad de dibujos, además de mapas de procesos o cursogramas. Visio Pro puede crear a partir de un gráfico, un documento en formato HTML para ser publicado en la Web. Importa imágenes en una gran variedad de formatos: ABC FlowCharter 2.0, 3.0, 4.0: .af3, .af2; AutoCAD; .gif; Corel Clipart .cmx; CorelDRAW! .cdr, .cfl; Macintosh .pct; tif, txt, bmp, .pcx, entre otros.

Es un programa que puede ayudarle a crear diagramas empresariales y técnicos en los que se documenten y organicen ideas, procesos y sistemas complejos. Los diagramas creados en Visio 2003 le permiten clara, concisa y eficazmente visualizar y comunicar información, de unas formas no posibles utilizando exclusivamente texto y números. También automatiza la visualización de los datos al sincronizarse directamente con los orígenes de datos para proporcionar diagramas actualizados, y se puede personalizar para cubrir las necesidades de la organización.

En general se utiliza específicamente para:

- I. Comprender fácilmente conceptos, procesos y relaciones: Se puede utilizar para crear de forma sencilla diagramas empresariales y técnicos que permitan considerar, organizar y describir mejor ideas, procesos y sistemas complejos.
- II. Comunicarse con claridad y eficacia: Puede visualizar y comunicar ideas, información y sistemas.
- III. Integrar y automatizar sistemas y procesos: Puede realizar trabajo más sofisticado, mejorar la comprensión y aumentar la productividad para crear impacto en la empresa.

Por otra parte la última versión que salió al mercado es Microsoft Office Visio 2003, la cuál presenta un cúmulo de características que lo ubican como puntero entre los software de diagramas de flujos. Entre sus particularidades se pueden ver que si el usuario nunca antes ha utilizado Visio, puede acceder a un curso eficaz para comenzar llamado Getting Started, el cuál le ayudará a construir varios tipos de gráficas rápidamente, tales como diagramas de bloques, de flujo de trabajo y de agenda de proyectos. Como en las versiones anteriores, una vez que se seleccione la clase de gráfica que desee, él tiene las herramientas y las formas adecuadas para que pueda crear un diagrama simplemente con arrastrar y soltar. Además, el programa mantiene los elementos correctos del diagrama conectados a medida que se realizan los cambios, así como una caja de ayuda que cambia según el contexto y que permanece en la pantalla en la esquina superior derecha. Si no está seguro del diagrama que satisface sus necesidades, puede dirigirse a la Diagram Gallery, donde encontrará ejemplos terminados. Con objeto de complacer el creciente interés en tales esfuerzos de renovación empresarial como Six Sigma (metodología normal de la industria que las compañías utilizan para señalar los problemas y mejorar la eficiencia), Microsoft ha introducido plantillas específicas para cada programa. También hay una práctica mejora para los grupos de trabajo que consiste en una opción con la que se puede rastrear quién agregó determinado comentario o elemento al diagrama. Además, si se desea crear un calendario que detalle el horario específico de un proyecto, el programa tiene un asistente que importará rápidamente el calendario de Outlook. Y, por último, al igual que las demás aplicaciones renovadas para el Office de Microsoft, Visio usa el XML. Con esta mejora se podrá integrar, por ejemplo, la información en vivo de las bases de datos de la compañía en los diagramas de flujo de procesos.

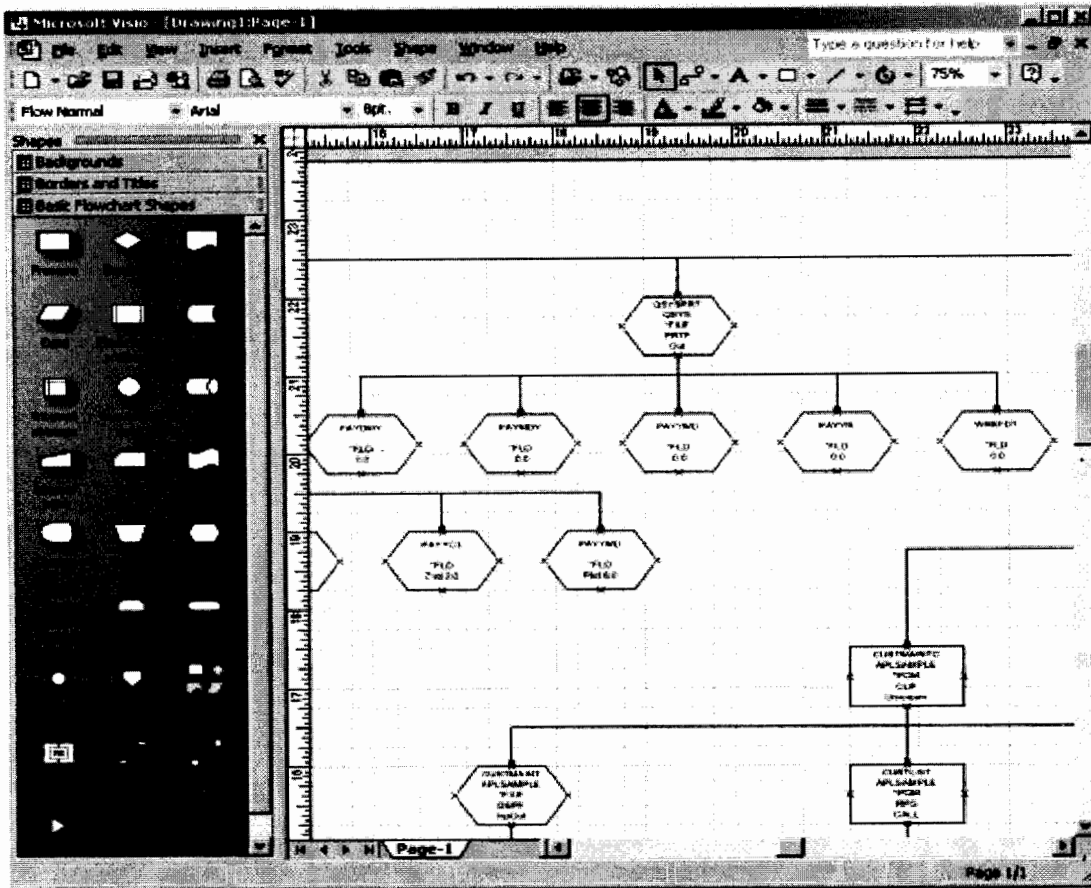


Figura 8: Ambiente de Microsoft Office Visio.

2.3 Herramientas de modelado UML.

Desde la publicación oficial del UML a fines de 1997, la cantidad de herramientas comerciales para el modelado se incrementó dramáticamente. Esto provee más alternativas al mismo tiempo que exige realizar más investigaciones para seleccionar la herramienta de modelado UML que responda mejor a los requisitos de negocio y desarrollo de software de aplicación y permite lograr el mejor retorno de la inversión. Según estudios realizados se debe escoger cada metodología de trabajo de acuerdo al proyecto que se esté realizando, aunque sí se especifica que para un mejor entendimiento debe escogerse una herramienta Case. En este grupo sobresalen Enterprise Architect (EA) de Sparx Systems y Rational Rose de Rational, IBM, aunque existen otras de muy buen uso y eficiencia como son: Visual Paradigm, Poseidón y Altova Umodel.

2.3.1 Visual Paradigm.

Visual Paradigm es una de las principales empresas creadoras de herramientas CASE. Tiene disponible distintas versiones: Enterprise, Professional, Standard, Modeler, Personal y Community (que es gratuita). El principal uso de la misma es con fines académicos, aunque también se puede utilizar para modelar casos reales. Esta herramienta fue diseñada para un amplio número de usuarios, incluyendo los ingenieros en software, analistas de sistemas, analistas de negocio y arquitectos de sistemas, los cuáles están interesados en construir sistemas de una amplia duración en su uso, y que sean fácilmente cambiables producto de la amplia documentación que tendrán. Soporta las últimas tecnologías existentes sobre el UML, lo que la hace muy factible para la enseñanza, aunque entre sus desventajas se destaca que no permite realizar diagramas de colaboración ni diagramas de despliegue. En cuanto a la generación de código, es compatible con C, C++ y JAVA. Permite generación de documentación en formato estándar y realiza ingeniería inversa según los lenguajes de programación permisibles. Posee una excelente navegación por modelos, especificándose que la misma se realiza por árbol jerárquico. Por otra parte es una herramienta multiplataforma que presenta métricas de depuración y testeo del modelo y es compatible con Rational Rose.

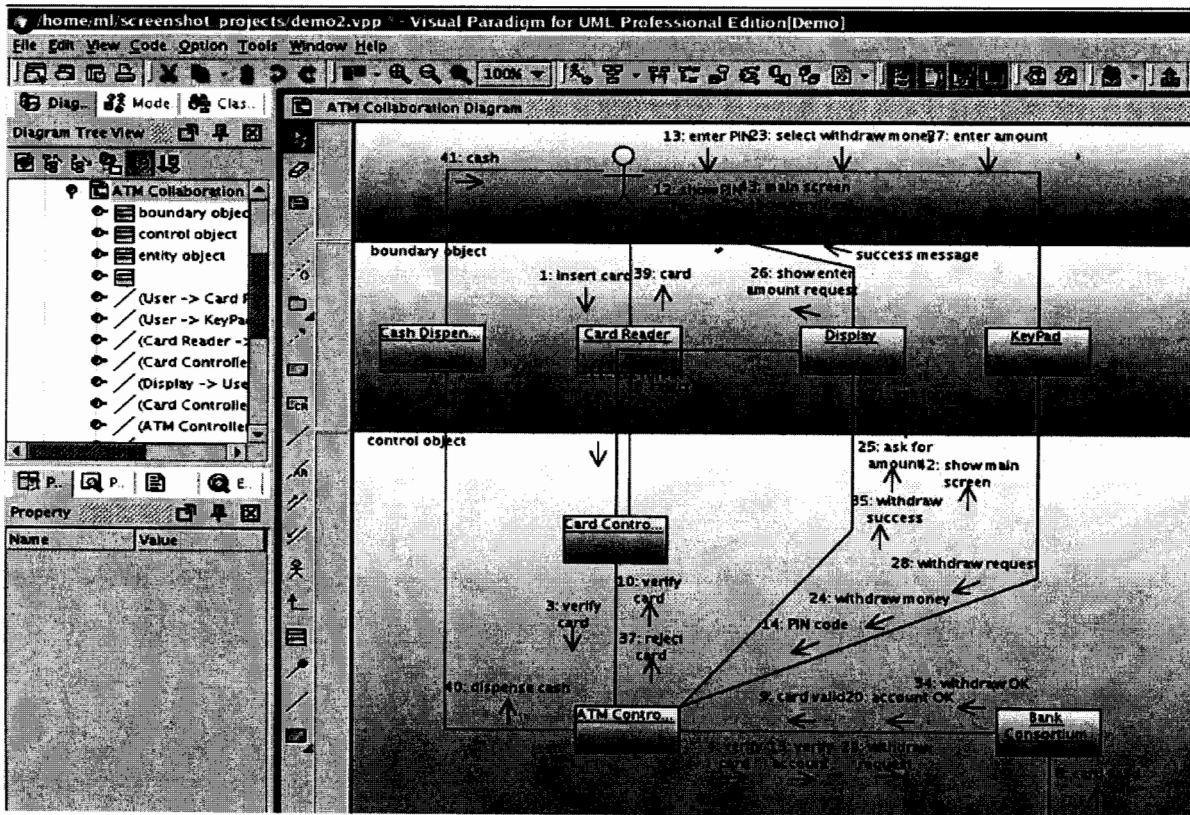


Figura 9: Ambiente de Visual Paradigm.

2.3.2 Poseidón.

Poseidón es una herramienta general para modelar cualquier clase de sistema que precise programación orientada a objetos o incluso sistemas que no tengan nada que ver con software. Entre sus características principales se destacan que está totalmente implementado en Java, es independiente de la plataforma, tiene soporte para todos los diagramas de UML, guarda en formatos compatibles con UML 2.0 Diagram Interchange Standard, Soporte para XMI 1.2, además de que los formatos estándares de almacenado. XMI 1.0, 1.1 y 1.2 son admitidos. También permite exportar los diagramas como gif, ps, eps y svg, da soporte para los formatos jpeg y png para JDK 1.4, tiene una internacionalización para Inglés, Alemán, Ruso, Francés, Español, y Chino, posee una útil generación de código Java, es de fácil instalación y se puede actualizar mediante Java Web Start. Por otra parte es gratuito, lo que le ha facilitado tener un gran número de usuarios que se inclinan por su uso.

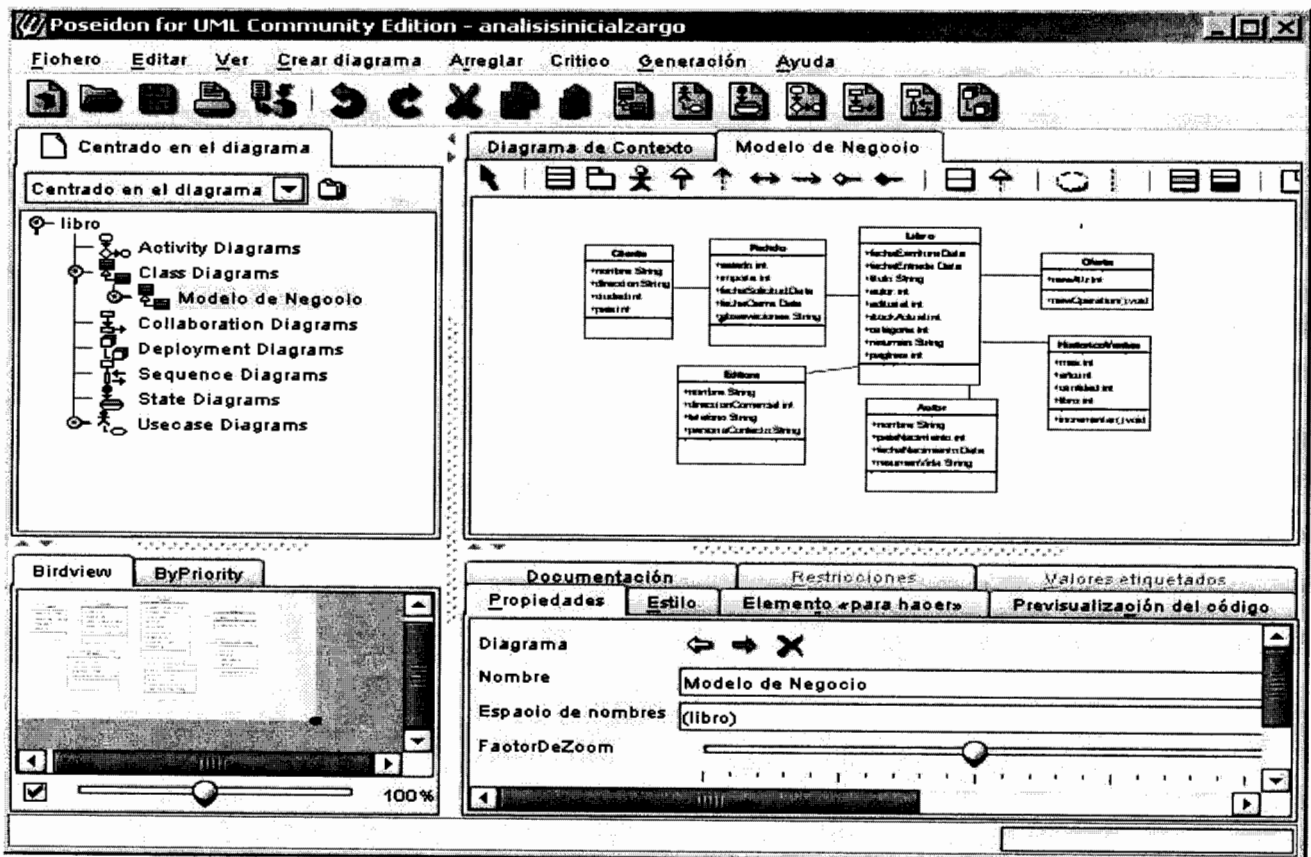


Figura 10: Ambiente de Poseidón.

2.3.3 Altova Umodel.

Es una herramienta de modelado basada en UML que permite también generación de código Java. Incluye ingeniería inversa con capacidad para leer código fuente en Java y generar modelos UML. Es compatible con Eclipse, Borland JBuilder, Microsoft Visual J# .NET, y otros entornos de desarrollo.

Por otra parte Altova UModel es el punto de partida para el desarrollo acertado del software. Permite diseñar visualmente los modelos de aplicación, o realizar ingeniería inversa de programas existentes a UML. El UML se ha reconocido como el estándar del siglo 21 para exponer y analizar requisitos del proyecto de desarrollo y para diseñar soluciones más eficientes del software. A pesar de sus ventajas claras, UML no se ha adoptado universalmente porque muchos equipos de programación eran cuidadosos de cambiar a una nueva tecnología que no les era familiar que se pudo haber percibido como elitista debido a los altos costos y complejidad relativos de herramientas anteriores de UML. Pero se planteó cuando surgió esta herramienta que ya esos tiempos habían quedado atrás ya que la misma iba a facilitar un fácil intercambio entre los usuarios y su interfaz. También se puede decir que modela las aplicaciones con un interfaz visual sin sobrecarga de ningún tipo en sus paneles y entre sus características superiores se encuentran que incluye objetos de alta funcionalidad para enriquecer a los usuarios con los aspectos más prácticos de la especificación de UML 2.0. Tiene opciones de gran utilidad para exportar en cualquier momento que se desee un diagrama como archivo de imagen de tipo png para incluir en la documentación del proyecto. Se puede incluso ejecutar desde la línea de comando para realizar tareas repetitivas tales como incluir la versión más reciente de un subproyecto compartido. Los paquetes compartidos creados en UModel pueden también aumentar la eficacia del equipo ya que son empleados para simplificar la reutilización de las estrategias de diseño o para integrar los grandes modelos creados en módulos por los miembros del equipo de programación. La última versión en existencia de esta herramienta es Altova Umodel 2005 la cuál ya presenta un cúmulo de originalidades como son la rapidez con que carga ya que lograron permitir a los usuarios comenzar a utilizar la aplicación con un mínimo de memoria requerida, incrementándose según los diagramas que se vayan creando, además de que posee unas excelentes funcionalidades para arrastrar y soltar las imágenes y permite importar diagramas.

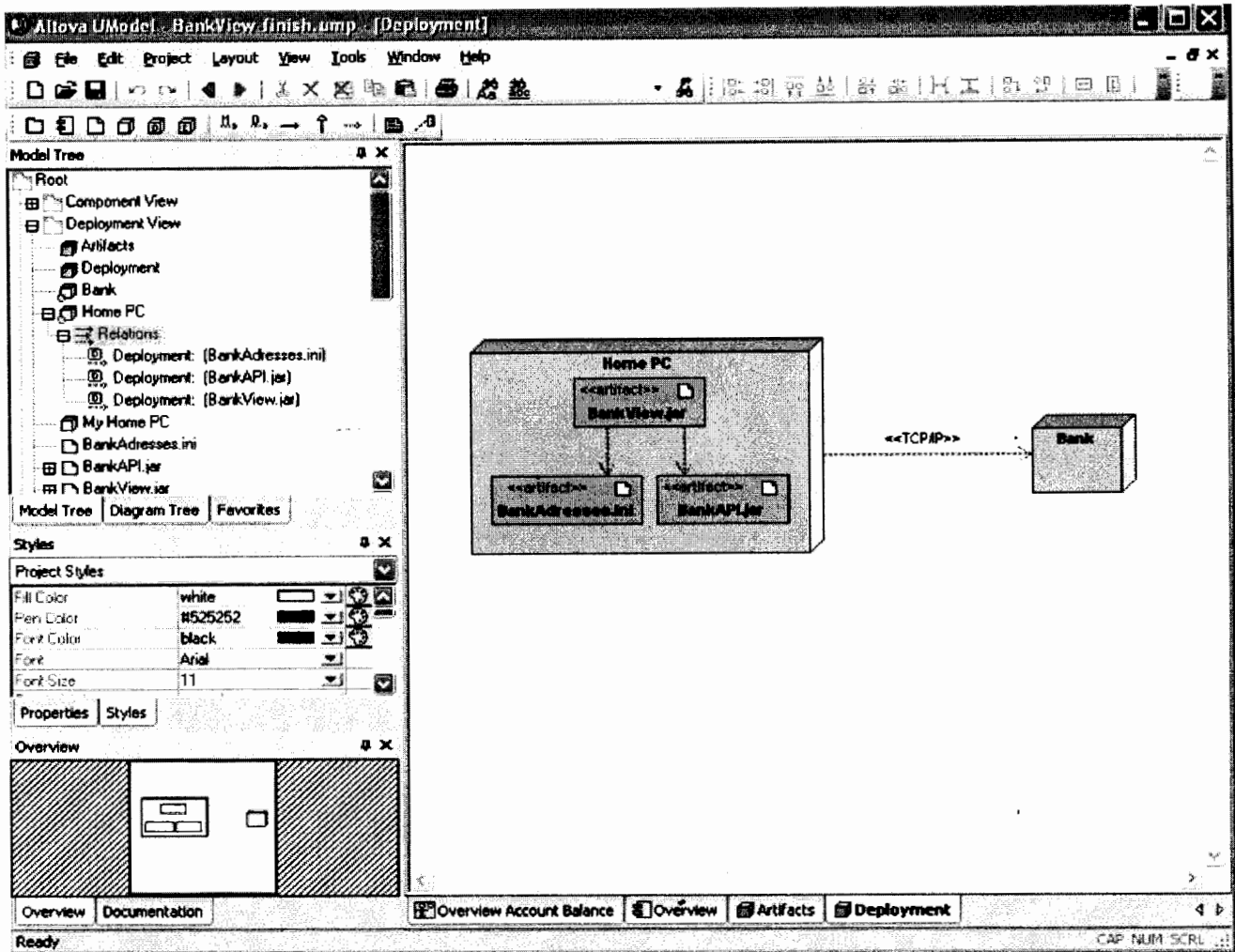


Figura 11: Ambiente de Altova Umodel.

2.3.4 Enterprise Architect.

Enterprise Architect (EA) ha sido una herramienta UML sorprendentemente efectiva y sencilla. Ha facilitado el trabajo de una gran variedad de modelados de procesos de negocio y se ha dedicado completamente a la tarea de completar los proyectos de desarrollo de software para los clientes. A pesar de ser relativamente nueva, es completamente confiable y, lo mejor de todo, cómoda. Los clientes han quedado muy impresionados con la calidad de la documentación que produce. Sparx parece estar completamente comprometido con EA, dado que cada versión brinda un conjunto de nuevas características que lo ponen cada vez más cerca de ser una herramienta UML completamente acabada, a una fracción del costo de los productos de la competencia. Los términos de licenciamiento son sencillos y la instalación es completamente más simple que para la competencia. Reduce significativamente la

cantidad de trabajo duplicado que otros softwares hacen, es un excelente producto que cuenta con un sensacional soporte técnico. Tiene uno de los conjuntos de características más integrales y exhaustivas de todas las herramientas para el ciclo de vida con UML. Su rendimiento es veloz y todo está ubicado adecuadamente en una única interfaz. La última versión de este producto es Enterprise Architect 3.60, en la cuál se agregaron elementos de subactividades y subdiagramas de estados a la barra de herramientas de UML, cada uno de ellos crea automáticamente un nuevo diagrama hijo de aquel sobre el que está. Se agregaron capacidades para mostrar un caso de uso utilizando una notación rectangular, mostrándose atributos, operaciones y restricciones del mismo, en la misma forma que una clase. Se incluyó también entre otras capacidades el eliminar una operación de un diagrama de actividad desde la vista de proyecto y crear automáticamente un elemento de actividad, se añadió el elemento de modelado artefacto para el compartimiento de despliegue de la barra de herramientas de UML, la capacidad para mostrar interfaces realizadas para clases y componentes (habilita/deshabilita utilizando un elemento del menú contextual). Se corrigieron problemas con el foco de control en los mensajes "New" de los diagramas de secuencia (por ejemplo, crear una nueva instancia). Se agregaron capacidades para generar un directorio de estructura de código completa desde un paquete o subpaquete del modelo en un paso utilizando valores por defecto. Se adicionó la apariencia de multiobjetos para los elementos de secuencia y la opción "esconder en otros diagramas" al menú contextual del conector lo que permite mostrar/ocultar un conector en todos excepto en el diagrama corriente. Se modificó el orden del ordenamiento de algunas colecciones en la interfaz de automatización para coincidir con la secuencia que se muestra en el navegador de proyecto. Se arreglaron algunos temas al agregar diagramas hijos a elementos en el navegador de proyecto. Se corrigió el asunto de bloquear paquetes empleando bloqueo de grupo. Se arregló el problema con la presentación del menú contextual en la barra de herramientas de UML, de las asociaciones entre objetos que se borran en los diagramas de colaboración cuando éste se borra del diagrama. Se modificó la ventana de tópicos para dejarlo en estado cerrado cuando el tópico se resuelve. Se corrigió el problema de sensibilidad a las mayúsculas o minúsculas del alcance de clase en la importación/exportación en XMI. Se arregló el problema con los elementos de clase asociación que muestran vínculos punteados en las imágenes que derivan de ítems solo actualmente seleccionados. Se agregó una opción al menú contextual para que elementos únicos incluyan "copiar la imagen seleccionada al portapapeles". Por otra parte esta herramienta constituye en conjunto con Rational Rose, dos de las más usadas por los desarrolladores.

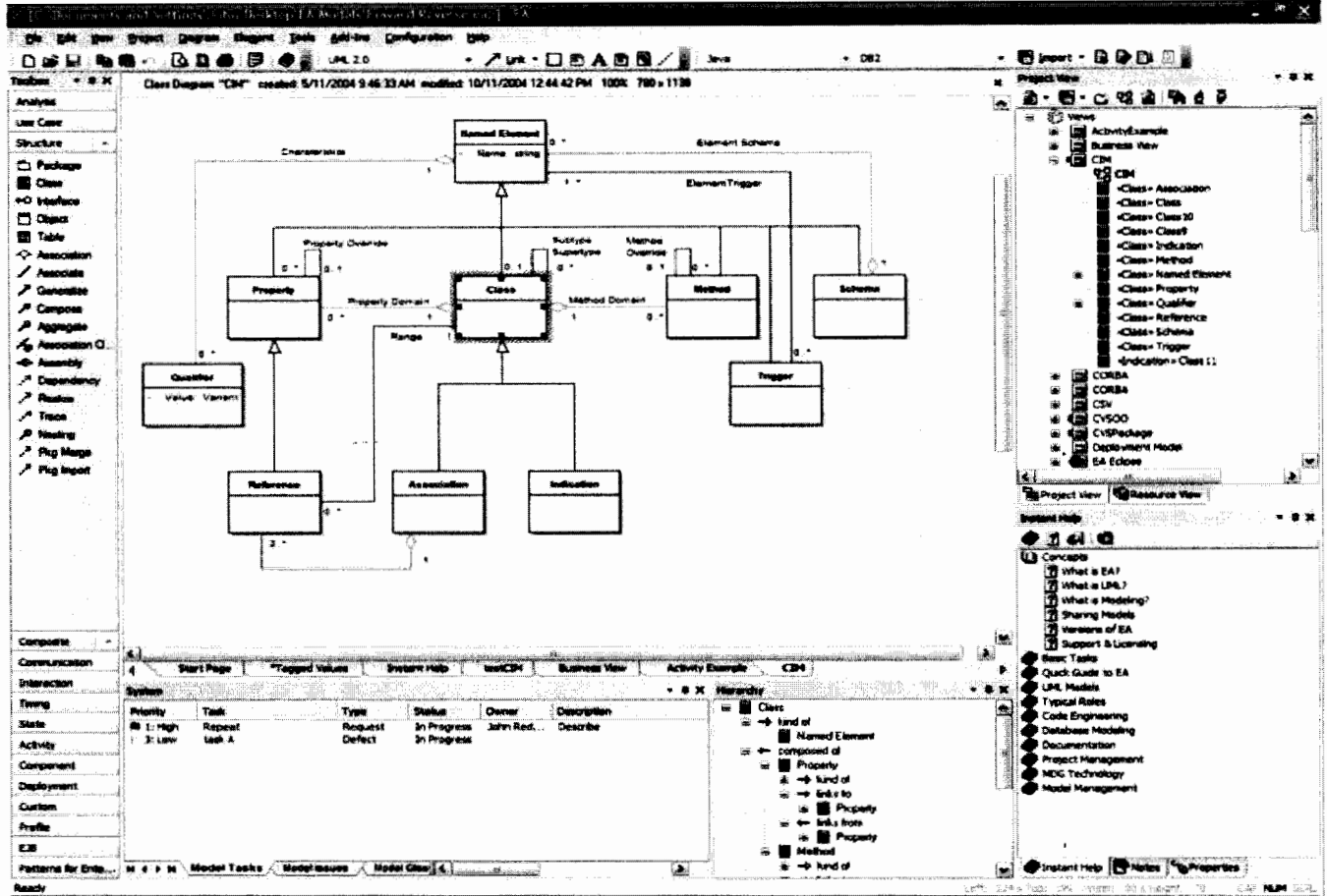


Figura 12: Ambiente de Enterprise Architect.

2.3.5 Rational Rose.

Rational Rose es la herramienta CASE que comercializan los desarrolladores de UML y que soporta de forma completa la especificación del UML 2.0. Esta herramienta propone la utilización de cuatro tipos de modelos para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software. Utiliza un proceso de desarrollo iterativo controlado, donde el desarrollo se lleva a cabo en una secuencia de iteraciones. Cada iteración comienza con una primera aproximación del análisis, diseño e implementación para identificar los riesgos del diseño, los cuales se utilizan para conducir la iteración, primero se identifican los riesgos y después se prueba la aplicación para que éstos se hagan mínimos. Cuando la implementación pasa todas las pruebas que se determinan en el proceso, esta se revisa y se añaden los elementos modificados al

modelo de análisis y diseño. Una vez que la actualización del modelo ha finalizado, se realiza la siguiente iteración. También permite que haya varias personas trabajando a la vez en el proceso iterativo controlado, para ello posibilita que cada desarrollador opere en un espacio de trabajo privado que contiene el modelo completo y tenga un control exclusivo sobre la propagación de los cambios en ese espacio de trabajo. También es posible descomponer el modelo en unidades controladas e integrarlas con un sistema para realizar el control de proyectos que permite mantener la integridad de dichas unidades. En cuanto a los mecanismos de ingeniería inversa y a las facilidades que posee para generar código se puede decir que permite generar código en distintos lenguajes de programación a partir de un diseño en UML y proporciona mecanismos para realizar la denominada Ingeniería Inversa, es decir, a partir del código de un programa, se puede obtener información sobre su diseño.

2.3.6 Importancia de las herramientas Case en la actualidad.

A medida que los sistemas que hoy se construyen se tornan más y más complejos, las herramientas de modelado con UML ofrecen muchos beneficios para todos los involucrados en un proyecto, por ejemplo, para el administrador del proyecto, los analistas, los arquitectos, los desarrolladores y otros. Las herramientas CASE de modelado con UML permiten aplicar la metodología de análisis y diseño orientados a objetos y lograr abstraerse del código fuente, en un nivel donde la arquitectura y el diseño se tornan más obvios y fáciles de entender y modificar. Cuánto más grande es un proyecto, es más importante utilizar una herramienta CASE. Al usar las herramientas CASE:

- I. Los analistas de negocio o del sistema pueden capturar los requisitos del negocio/sistema con un modelo de casos de uso.
- II. Los diseñadores y arquitectos pueden producir el modelo de diseño para articular la interacción entre los objetos o los subsistemas de la misma o de diferentes capas (los diagramas UML típicos que se crean son los de clases y los de interacción).
- III. Los Desarrolladores pueden transformar rápidamente los modelos en una aplicación funcionando, y buscar un subconjunto de clases y métodos y asimilar el entendimiento de cómo lograr interfaces con ellos.

Por otra parte el modelo actúa como el plano que guiará finalmente la construcción del sistema. De manera semejante, la administración es capaz de ver, puntualmente y desde un alto nivel, una representación del diseño y comprender lo que está sucediendo.

Por estas razones, las herramientas CASE acompañadas con metodologías, brindan una forma de representar sistemas demasiados complejos para comprenderlos a través de su código fuente subyacente y permite desarrollar la solución de software correcta más rápido y más económicamente.

Sin embargo, las mismas varían con respecto a las capacidades de modelado con UML, el soporte del ciclo de vida del proyecto, las ingenierías directa y reversa, el modelado de datos, el rendimiento, el precio, el soporte y la facilidad de uso, entre otras, por lo que como se planteó anteriormente hay que saber escoger bien acorde a nuestras exigencias.

2.4 Fundamentación de la herramienta utilizada en la UCI para realizar los diagramas de flujo.

En la Universidad de las Ciencias Informáticas, específicamente en la dirección de informatización se decidió utilizar para realizar los diagramas de flujo de procesos como estándar el Microsoft Office Visio. Entre las razones fundamentales para esta elección se encuentran:

1. Era necesario documentar y analizar procesos empresariales: para esto era idóneo el Visio ya que puede diseñar, documentar y analizar los procesos empresariales con plantillas y formas que admiten proyectos de administración de procesos empresariales (BPM, Business Process Management), lo que incluye iniciativas de mejora de la calidad Seis Sigma y documentación ISO 9000.
2. Era necesario hacer un seguimiento de los comentarios de las personas: El Visio cuenta con un modo de revisión para hacer un seguimiento de forma sencilla de los comentarios, las formas y la entrada manuscrita digital agregados por otros usuarios.
3. Se necesitaba una colaboración para en ocasiones trabajar más de una persona en el mismo proceso: Se pueden publicar diagramas de Visio en una área de trabajo de Microsoft SharePoint Portal Server o exportar diagramas con el formato de Gráficos de vectores escalables (SVG) o la funcionalidad actualizada de guardar como página Web.
4. En ocasiones se necesitaba capturar las ideas con herramientas de generación de ideas: Se pueden utilizar nuevas plantillas de Visio 2003 para capturar y estructurar el resultado de una

sesión de generación de ideas. Puede exportar diagramas de generación de ideas a Microsoft Word, Microsoft Excel o XML para dar lugar a la creación de otros archivos empresariales.

5. Se necesitaba crear y utilizar diagramas técnicos con mayor facilidad: Se puede aprovechar las numerosas mejoras de las herramientas de creación de diagramas de base de datos, ingeniería, red, software y Web de Visio 2003.
6. Se tenían que crear y modificar diagramas con mayor rapidez: Visio 2003 incluye una serie de mejoras para ayudarle a aumentar la productividad. Entre otras tareas, puede girar formas sin cambiar a un modo de giro especial e imprimir partes seleccionadas de un diagrama. Además, la característica de búsqueda de formas siempre está abierta para ayudarle a buscar rápidamente la forma que desee.
7. Se debían crear soluciones eficaces y conectadas: se puede incorporar Visio 2003 en soluciones que aprovechen los servicios Web XML y el software conectado por Microsoft .NET para conectar usuarios, información, sistemas y dispositivos. Puede conectar los diagramas a información empresarial importante para mejorar la comprensión y permitir acciones más decisivas.

2.5 Fundamentación de la herramienta utilizada en la UCI para el modelado con UML.

Después de un estudio realizado en la Universidad de las Ciencias Informáticas se decidió que se debía utilizar para la tarea de modelación con UML una herramienta Case por la importancia que estas presentan en la actualidad, y se acordó utilizar de todas ellas Rational Rose. El por qué de esta decisión está específicamente en que el UML estándar está compuesto por tres partes: bloques de construcción (tales como clases, objetos, mensajes), relaciones entre los bloques (tales como asociación, generalización) y diagramas (por ejemplo, diagrama de actividad). Los perfiles del UML son las extensiones a las notaciones estándares del UML usando los mecanismos de extensión del UML: los estereotipos, los valores etiquetados y las restricciones. Rational Rose es compatible con el UML 2.0 y soporta ocho de los nueve diagramas estándares del UML: diagrama de casos de uso, de clases, de secuencia, de colaboración, de actividad, de estados, de implementación (componentes), de despliegue y varios perfiles del UML. Si fuera necesario, el diagrama de objetos se puede crear usando los diagramas de colaboración.

Por otra parte la familia de software Rational Rose es la solución para desarrollo basado en modelación más premiada, forma parte de una solución totalmente integrada diseñada para cumplir los actuales desafíos del desarrollo de software. Integrando los ambientes de modelado y desarrollo, y utilizando el UML, permite que todos los miembros del equipo se desarrollen individualmente y se comuniquen colaborativamente para producir un mejor software. Además de esto en la UCI se necesitaba crear arquitecturas de sistemas que fueran robustas, y Rational tiene la capacidad de crear arquitecturas basadas en componentes flexibles, y permite que los procesos de software evolucionen en forma controlada, administrada e identificable, reduciendo los costos y acelerando los tiempos de entrega.

También se llegó a esta decisión ya que en la Universidad de las Ciencias Informáticas se desarrollan software de distintos tipos y se usan para su implementación distintas tecnologías, y este paquete CASE ofrece una integración completa con todas las IDEs líderes del Mercado y las últimas tecnologías, maximizando la velocidad y simplicidad de las tareas de desarrollo. Entre los mayores beneficios de su uso se encuentran que le ofrece a los diseñadores y desarrolladores de software un rico conjunto de capacidades de desarrollo y de análisis en tiempo de ejecución orientadas al modelado para construir aplicaciones de software de calidad, ofrece ambientes completos de diseño y desarrollo visual que satisfacen las necesidades de toda organización enfocada tanto a sistemas basados en J2EE como en .NET, permite que los arquitectos y diseñadores practiquen el desarrollo orientado al modelado utilizando UML. Tales usuarios pueden producir modelos independientes de la arquitectura de la plataforma de software, de las necesidades del negocio, y del nivel de administración de las comunicaciones. El soporte de UML, que es el estándar de la industria, y un poderoso motor de patrones permiten a los usuarios crear una arquitectura de aplicación semánticamente rica que satisface las necesidades del negocio y es fácilmente comprendida por el resto del equipo.

2.6 Conclusiones.

En este capítulo se analizaron las tendencias que existen en la actualidad con respecto a las herramientas del Lenguaje Unificado de Modelado y de Flujo de Procesos. Además se explicaron con detenimiento las tecnologías que más se están utilizando internacionalmente para estas tareas, dejando bien claro las particularidades de cada una de ellas. Por otra parte se concluyó explicando cuáles de estas herramientas son las utilizadas por la Universidad de las Ciencias Informáticas, abordando el por qué se hizo esta selección.

Capítulo 3. Descripción de la propuesta diseñada.

3.1 Introducción.

En el presente capítulo se abordará la metodología de trabajo que se propone. Para esto se utilizará una heurística muy útil en la actualidad, la cuál plantea que para uno interpretar algo cabalmente, debe primeramente comprender lo que desea interpretar. Es por esto que a continuación se explican los pasos a seguir a la hora de realizar los diagramas de procesos, para posteriormente abordar las distintas formas interpretativas que pueden tener, dejando bien claro que esto solo es una forma de interpretación, puesto que estará sujeto a cambios según la manera de analizar de cada persona. Por otra parte se expondrá el proceso de conversión a UML en detalles, dejando para el capítulo posterior una serie de ejemplos prácticos.

3.2 Cómo se realizan los diagramas de flujo en detalle.

El desarrollador debe tener en cuenta que los diagramas de flujo poseen ciertas características que son inviolables, por lo que en ocasiones serán bastante escuetos, por lo cuál se requerirá tener un gran poder de análisis ya que entre dichas características se encontrará que la representación que se hace de un sistema o un proceso deberá quedar resumida en pocas hojas, de preferencia en una sola. Los diagramas extensivos dificultan su comprensión y asimilación, por tanto dejan de ser prácticos.

Existen dos niveles de interpretación: comprensión del proceso y mejora del mismo. La mejor manera de adquirir conocimiento sobre un proceso en curso es recorrerlo según el diagrama de procesos, paso a paso, siguiendo el flujo indicado por las flechas. Por esto, y dado que los equipos de mejora suelen estar constituidos por desarrolladores que inicialmente sólo conocen en profundidad una de las partes del proceso a lo sumo, es recomendable plantearse como primer objetivo el de adquirir un mejor conocimiento común completo del proceso en su conjunto. El error más común es no documentar el proceso real o no actualizarlo, ya que los desarrolladores deben primeramente ir documentando la forma en que van interpretando las actividades(casos de uso), para esto se auxiliarán de la documentación auxiliar que realiza el ingeniero industrial sobre el proceso en curso. A la hora de realizarlos se debe tener claro que existen elementos básicos que aparecerán en cualquier diagrama de flujo de procesos, estos son:

I. Entidad externa: representan entes ajenos a la aplicación, pero que aportan o reciben información de la misma. Se simboliza mediante una elipse o un rectángulo con un nombre significativo dentro.

- Reglas de Construcción:

- a. Representa personas, organizaciones o sistemas que no pertenecen al sistema.
- b. En el caso que las entidades externas se comuniquen entre sí, esto no se contemplaría en el diagrama, por estar fuera del ámbito del sistema.
- c. Puede aparecer en cualquier nivel del diagrama.
- d. Puede aparecer varias veces en un mismo diagrama, para evitar entrecruzamientos de líneas.
- e. Suministra información acerca de la conexión del sistema con el mundo exterior.

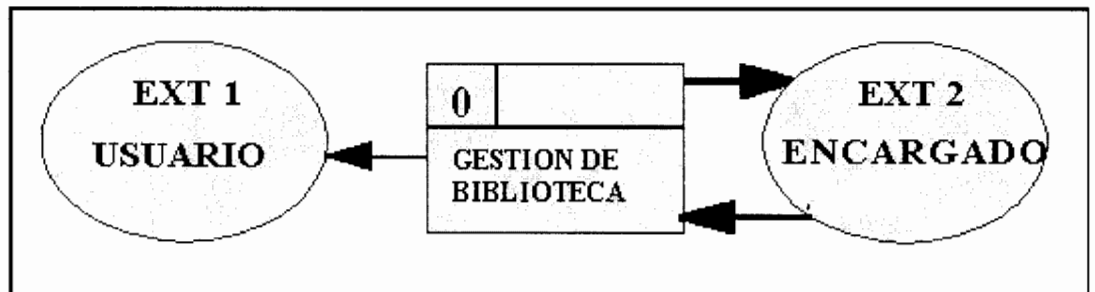


Figura 13: Representación de entidades externas.

II. Actividades: En las actividades se expresa el nombre correspondiente, dependiendo del nivel de detalle en que se encuentren dentro del diagrama, el nombre de la actividad simbolizará bien el sistema o bien acciones concretas y detalladas en niveles inferiores.

- Reglas de Construcción:

- a. Cuando un Flujo de datos entra en un proceso, sufre una transformación. Una actividad no es ni origen ni final de los datos, sólo lugar de transformación de los mismos. Por ello, cualquier flujo de datos que entre en un proceso, ha de transformarse.
- b. Una actividad o proceso puede transformar un dato, en varios, o modificar algún parámetro.

- c. Es necesario una actividad como intermediario entre una Entidad Externa y un Almacén de Datos.

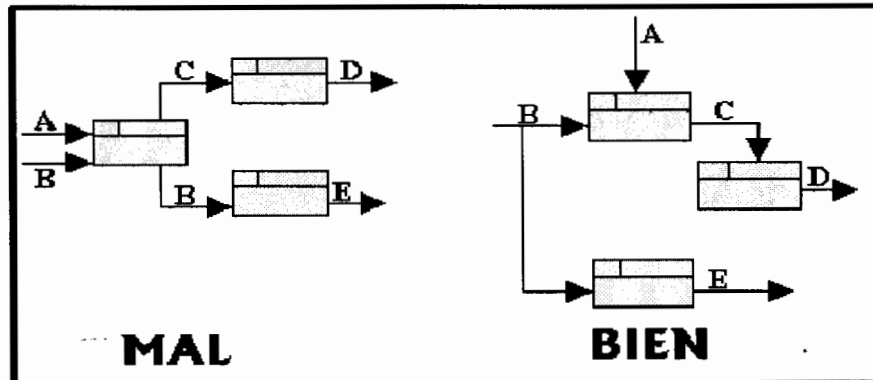
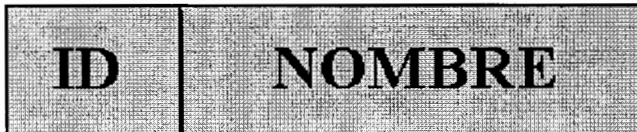


Figura 14: Representación de las actividades.

- III. Almacén de datos: Un almacén de datos representa un depósito de información dentro del sistema. Se representa con el siguiente símbolo:



En la parte derecha se indica el nombre del almacén de datos y en la parte izquierda se representa la identificación del mismo dentro del diagrama de flujo. Es conveniente distinguir las diferentes utilidades que presentan los almacenes de datos. En primer lugar, se utiliza para el almacenamiento permanente de datos, donde se guardan los datos que sirven de referencia de uso del sistema, es decir, los permanentes, sobre los que el sistema necesita guardar información. Por otra parte, se aprovechan para el almacenamiento transitorio antes de ser usados por un proceso.

Para entender el significado de estos almacenes transitorios, se puede imaginar la situación del ejemplo de la figura siguiente:

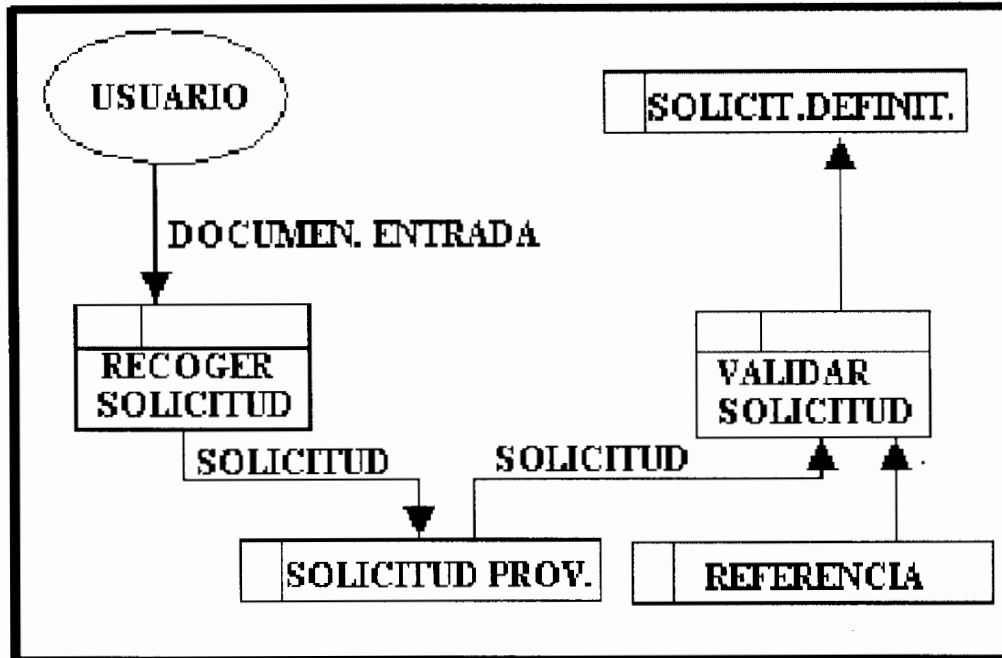


Figura 15: Ejemplo del uso del almacén de datos.

En este ejemplo, el proceso RECOGER SOLICITUD, que se ejecuta continuamente a lo largo de la jornada, genera los datos de salida representados por el flujo SOLICITUD. Estos datos constituyen la entrada al proceso VALIDAR SOLICITUD, que se ejecuta al final de la jornada. En el intervalo, esos datos de solicitud "reposarían" en el almacén SOLICITUD. PROV. cuya utilidad básica es establecer una sincronización en el funcionamiento de ambos procesos.

Los almacenes transitorios suelen representar restricciones físicas del sistema y por tanto en un diagrama de flujo, que expresa la lógica de los tratamientos realizados por el sistema, en muchos casos no será necesario representarlos.

Sin embargo, hay ocasiones en que estos almacenes simbolizan "ficheros de movimientos" donde se guardan los datos, porque el proceso siguiente necesita manejarlos todos al mismo tiempo (por ejemplo, en un proceso que compara un conjunto de registros, será necesario mantenerlos guardados en un almacén transitorio, para que dicho proceso los lea todos al mismo tiempo). En este caso sí será conveniente representarlos.

- Reglas de construcción:
 - a. Representa la información en reposo.
 - b. No puede crear, destruir ni transformar datos.
 - c. No puede estar comunicado directamente con otro Almacén o Entidad Externa.

- d. El almacén de datos aparecerá por vez primera en aquel nivel en que sea accedido por dos o más procesos y en modo lectura y/o escritura.
- e. No debe estar referido al entorno físico y por tanto, no se diferencian los ficheros convencionales de las Bases de Datos.
- f. No se representa la clave de acceso a ese almacén, sino sólo la operación que se realiza (lectura, escritura, actualización).

IV. Flujo de datos: Los flujos de datos establecen la comunicación entre procesos o actividades, almacenes y entidades externas, llevando información necesaria.

- Reglas de construcción:
 - a. El flujo de datos es la forma en que fluye una información de estructura conocida.
 - b. Los datos no pueden ser creados ni destruidos por un flujo de datos.
 - c. Sirve para conectar el resto de los componentes del diagrama.
 - d. No es un activador de actividades.
 - e. Cuando un proceso almacena datos, la flecha de flujo de datos se indica en la dirección del almacén de datos, y a la inversa, si es el proceso el que lee datos en el almacén.

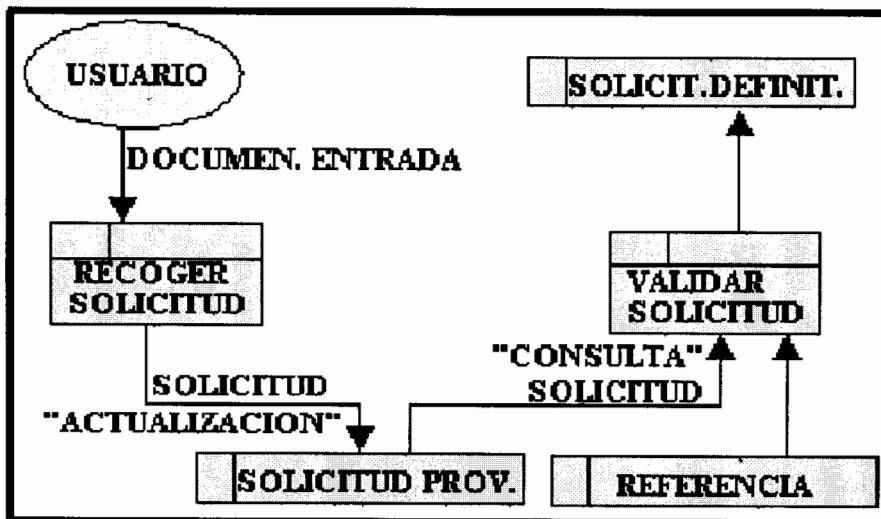


Figura 16: Ejemplo del uso del flujo de datos.

Por otra parte los diagramas de flujo de procesos deben representar las actividades de la forma más clara posible. Para ello se basarán en el principio de descomposición o explosión en distintos niveles de detalle. La descomposición por niveles permite analizar el sistema desde el ámbito general al detalle, pasando por sucesivos niveles intermedios (filosofía "top-down"). La utilización de esta técnica implica la descomposición o explosión de cada actividad en un proceso. En la figura siguiente se representan los distintos niveles que pueden surgir a la hora de desarrollar nuestro sistema de información. En cualquier momento puede aparecer un proceso que no necesite descomponerse y es lo que se llama PROCESO PRIMITIVO (PP). En él, sólo se detallará la entrada y salida que tenga, además de la descripción asociada que explique lo que realiza.

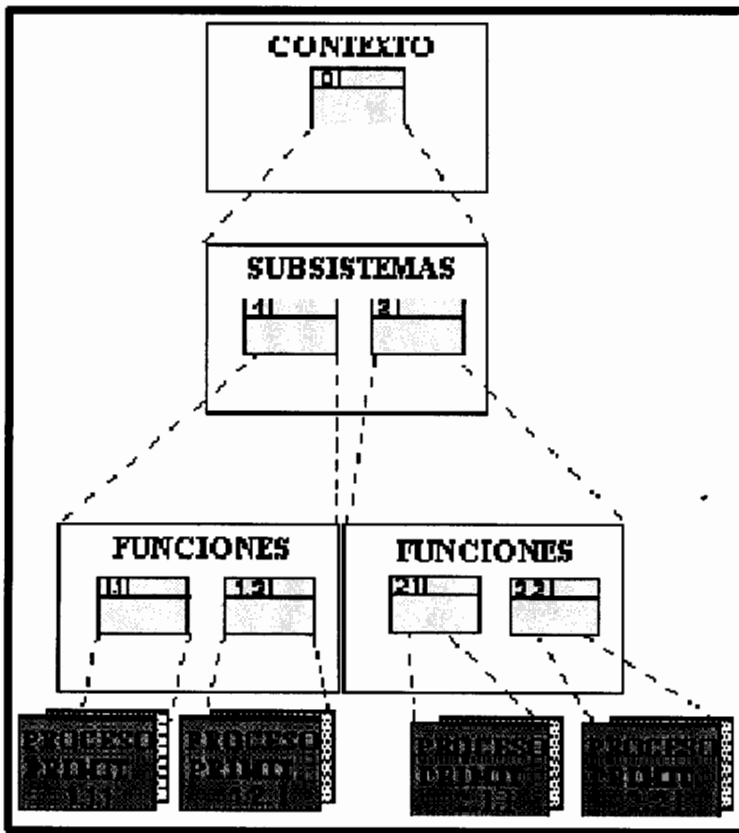


Figura 17: Representación de los niveles del sistema.

El procedimiento a seguir para la representación del sistema en diferentes niveles de detalle, se indica a continuación:

1. Representar el Diagrama de Contexto: El objetivo del diagrama de contexto es realizar una declaración formal del dominio. Un sólo proceso representará el área que se está estudiando.

El contexto queda definido por los flujos de entrada y salida y las entidades externas. Estas últimas han de aparecer en este nivel y no en ningún otro (salvo para mejorar la comprensión del resto de los diagramas). Dicho diagrama es, precisamente, el gráfico que va a proporcionar el ámbito del proyecto objeto de estudio. En él, aparecerá todo aquello que necesite o envíe datos del o hacia el sistema a desarrollar. Estos se representan por las entidades externas. El diagrama de contexto tendrá la siguiente forma:

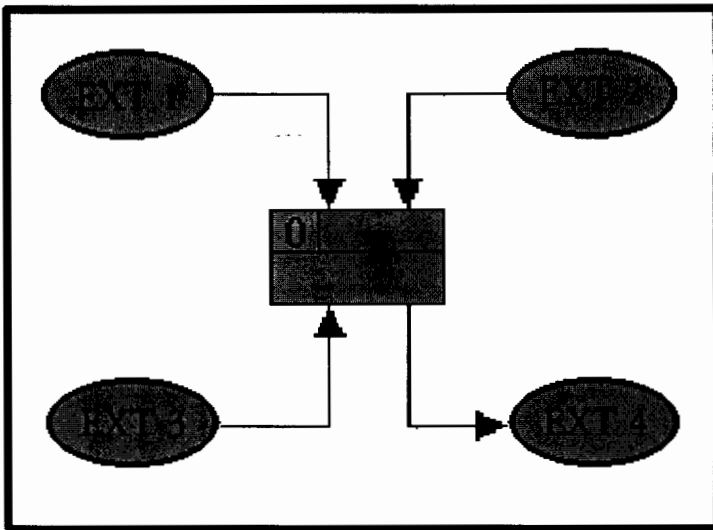


Figura 18: Representación del diagrama de contexto.

- II. Representar el diagrama de flujo de procesos, indicando los distintos subsistemas o áreas funcionales en que se descompone nuestro sistema.
- III. Descomponer cada uno de los procesos que aparecen en el diagrama, hasta llegar a un nivel suficiente de detalle.
- IV. Si es necesario, reagrupar y reorganizar los distintos subsistemas que se han identificado inicialmente, esto implicará reasignar procesos.
- V. Repetir el proceso de descomposición hasta llegar a un nivel suficiente de detalle.

Según estudios realizados, se plantea que deberán definirse más en detalle sólo las actividades o procesos primitivos, es decir, aquellos que no pueden descomponerse, pero no obstante nunca está de más una profunda descripción de los procesos.

3.3 Cómo se realizan los diagramas de flujo actualmente en la UCI.

Cómo ya se explicó con anterioridad este es un proceso completamente novedoso en la universidad, por lo cuál solo se está en las fases iniciales de diagramación, o sea todavía no se han detallado en profundidad los diagramas de flujo de procesos. A continuación se explicará la forma en que se realizan en la UCI.

El primer símbolo de casi todos los procesos es una entrada, ya que será la que le dará comienzo al proceso, o sea lo activará. El símbolo que lo representa ya fue visto anteriormente en el capítulo 1. El mismo no solo se refiere a las entradas, sino también que se utilizará para representar las salidas que tendrá un proceso dado.



Figura 19: Símbolo de representación de entrada/salida de un proceso.

Como bien se observa en esta figura y se verá también en las demás, debe especificarse el nombre de lo que ocurre dentro del símbolo. Posteriormente a esto es que comienza el proceso como tal, ya que se unirá mediante una flecha al símbolo de comienzo del proceso.



Figura 20: Símbolos de comienzo del proceso.

Después de esto ya deben ir comenzando a aparecer las actividades que se realizan durante el proceso, tales actividades en este caso específico serían algunas tales como: evaluación y aprobación por la rectoría y comunicación a la contra inteligencia para abrir una investigación. Puede darse el caso que hayan actividades que se realicen de forma paralela, esto se representa sacando dos flechas de un mismo proceso:

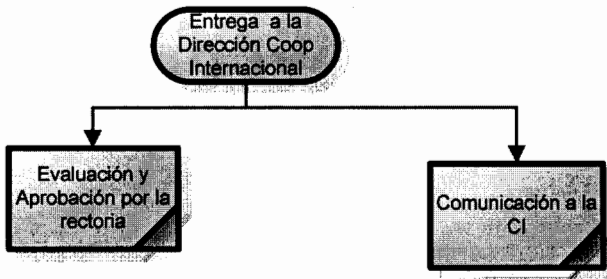


Figura 21: Representación de tareas en paralelo.

Después se continuaría el flujo siguiendo las actividades que vayan apareciendo junto con la toma de decisión. De aparecer una toma de decisión esta se representaría como mismo se hace en el UML, así que esto no debe tener mayores contratiempos. Los demás símbolos que pueden ir apareciendo en el diagrama serían los de creación de documentos. En caso de ser 1 documento y una copia se representaría así:



Figura 22: Símbolo de documento con copia.

Por el contrario, si se quiere que sean más de una copia, entonces se utilizaría el siguiente símbolo:

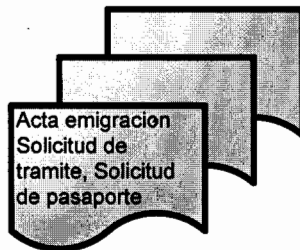


Figura 23: Símbolo de documento y varias copias o varios documentos.

A continuación se verá y analizará un ejemplo completo que ya ha sido realizado por los ingenieros industriales que trabajan en esta área de informatización:

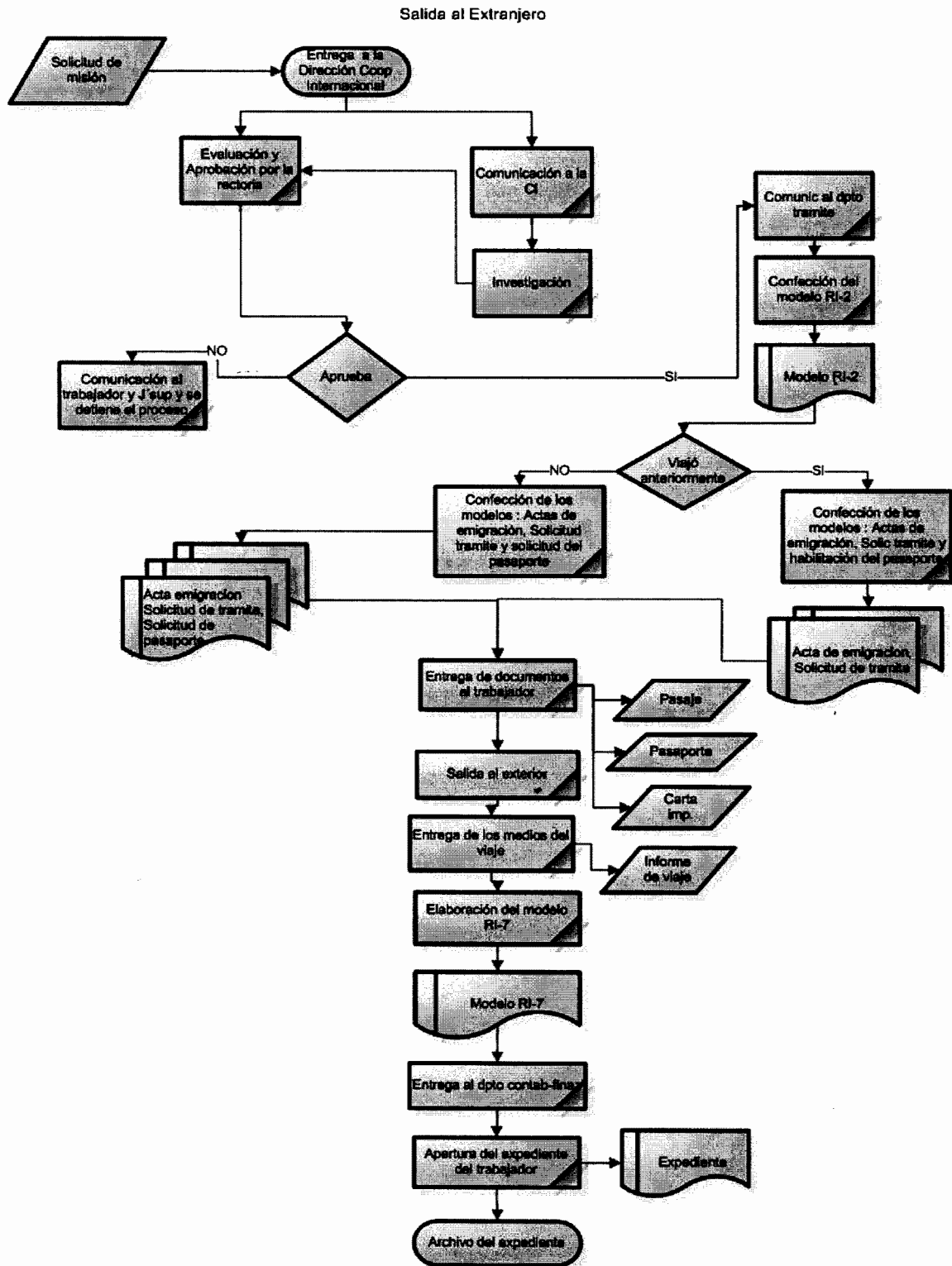


Figura 24: Ejemplo de un modelo de flujo de procesos.

3.4 Interpretación de los diagramas de Flujo.

Se va a partir del hecho de que aquí solo se mostrará la forma de interpretar los diagramas, el proceso de interpretación es algo que cada desarrollador hará de forma distinta, atendiendo a la forma en que desee trabajar y a la complejidad presentada, más abajo se abundará más sobre por qué se plantea esto. Se tocará a continuación los objetivos de la interpretación, también se verá que como esto se encuentra en fase de prueba y adaptación en la UCI, no se abarca de forma tan profunda, por lo que aquí se abordará la forma en que actualmente ocurre esta diagramación

3.4.1 Objetivos de la interpretación.

El interpretar correctamente los diagramas de flujo es un paso de gran importancia ya que no bastará con saber como intercambiar un símbolo por otro de UML, sino que además se deben poder extraer cosas extras de los diagramas que no se podrán entender sin una profunda comprensión del mismo. O sea, como bien se explicó con anterioridad, dichos diagramas de flujo de procesos no son muy explicativos en cuanto a la forma en que se realizan las cosas, ya que ellos solo recogerán el qué se hace, pero en ocasiones si se comprende bien lo que se está haciendo, es posible entender correctamente cómo se debe hacer, claro está que en este primer paso, solo necesita llevarse una idea de lo que se pretende hacer, ya que existirán otras iteraciones para una mejor comprensión. Por otra parte ya verá que una vez que se familiariza con ellos, prefiere hacerlos por la utilidad que presentan a la hora de explicar lo que se está haciendo a nuevos desarrolladores que se hayan incorporado, o al mismo cliente, ya que él necesita saber el qué se va a hacer. Por otra parte muchos administrativos ya conocen cómo trabajar con estos diagramas y será más fácil a la hora de un correcto entendimiento entre ellos y los desarrolladores.

Como conclusión previa se puede decir que como resultado de este proceso se obtendrá un modelo del sistema completamente independiente de las restricciones físicas del entorno, lo que facilitará su mantenimiento y portabilidad.

3.4.2 Cómo se interpretan los diagramas de flujo que se crean actualmente en la universidad.

La forma de leer un diagrama sería simplemente seguir el flujo descrito por las flechas correspondientes, teniendo en cuenta el significado de cada símbolo que contiene. En este caso en

particular se demostrará lo anteriormente planteado explicando el diagrama de la figura 24: el proceso se inicia cuando se entrega la solicitud de misión, la cuál es presentada a la dirección de cooperación internacional, de aquí se realizarían 2 actividades (por ahora se le llamará así) en paralelo: evaluación y aprobación por la rectoría y la comunicación a la contra inteligencia para la realización de la investigación pertinente, y posteriormente se enviarían los resultados de la investigación a rectoría para continuar con su evaluación, en caso de no aprobarse, se le comunicaría al trabajador y al jefe superior, culminando así el proceso, en caso de que se aprobara, se le comunica al departamento de trámite, el cuál confecciona el modelo RI-2 que no es más que un documento que se tiene como salida. De aquí se analiza si viajó anteriormente o no, en caso de no haberlo hecho nunca se confeccionarían los modelos: Actas de emigración, Solicitud de trámite y Solicitud del pasaporte, de los cuáles se obtendría como nueva salida, dichos 3 documentos. En caso de haber viajado anteriormente, se confeccionarían los modelos: Actas de emigración y Solicitud de trámite y se pediría la habilitación del pasaporte. Después de esto, se le entregarían los documentos al trabajador, actividad esta que tendrá como salida: el pasaporte, el pasaje y la carta de salida impresa, y además los medios de viaje, que daría una nueva salida: el informe de viaje. Para finalizar se elaboraría el modelo RI-7, documento que se le entregará al departamento de contabilidad y finanzas para la apertura del expediente del trabajador, donde se elaboraría este nuevo documento, y posteriormente se concluiría el proceso al archivar el expediente.

Después de que los desarrolladores interpreten este diagrama ya es hora de verlo más en detalle, para esto verían un documento que elaboran los ingenieros industriales dedicados a esta tarea en el cuál ellos dan explicaciones más concretas con las cuáles se contará a la hora de llevarlos a UML. En este caso que se está abordando sería así:

El proceso de salida al extranjero comienza con la entrega de la solicitud de misión elaborada de antemano con la aprobación del PCC y/o UJC a la Dirección de Cooperación Internacional y es archivada hasta el día de discusión y aprobación por la Rectoría, en la cuál se decide aprobar o no dicha solicitud. Además se le informa a los Órganos de contra inteligencia los nombres y posibles fechas de salidas de los misioneros, dicho órgano es el encargado de analizar al trabajador y comunicar los resultados a la Dirección, la cuál será la encargada de decidir la continuidad del proceso y de informar al solicitante la decisión tomada en la discusión por la rectoría. Si no se aprueba la solicitud se detiene y anula el proceso inmediatamente.

Si la solicitud es aprobada por la Rectoría, la Dirección de Cooperación pone en conocimiento la decisión al Grupo de Trámites el cual comienza a localizar al trabajador emitiéndose el modelo RI-2 con el objetivo de llenar los datos personales del viajero. En el Grupo de Trámites se emiten también los

modelos: Acta de emigración, Solicitud de trámite (AO-1) y Solicitud de pasaporte (AO-4) en el caso de que el trabajador no haya viajado con anterioridad, si ha viajado sólo se procede a la habilitación del pasaporte. El trabajador conforme a los gastos firma el modelo de solicitud de trámite (RI-3) en el cual quedan plasmados los gastos de tramitación en que se incurren. Llenados los modelos de los trabajadores que deben salir de viaje, se llevan a Inmigración: el AO-1, el acta de inmigración, si no tiene pasaporte se entrega el modelo de solicitud con la fotocopia del carné de identidad para confeccionarlo; este proceso de confeccionar el pasaporte se demora 7 días al igual que el de habilitación (si no es con inmediatez), pero si es con inmediatez (está abalado por un sello que se pone al AO-1) se demora 48 horas. Conjuntamente con la confección o habilitación del pasaporte, según se requiera, se procede a reservar el boleto de pasaje aéreo y solicitar la visa según el país (existen países que no llevan visado). Se hace la solicitud de cheque para pagar boleto y si el país lleva visa se solicita el cheque para pagar visado. Una vez hechas las solicitudes anteriores se procede a solicitar el anticipo de viaje (RI-6) donde se muestran otros gastos. Este modelo firmado por el Rector y la Directora de Cooperación Internacional se dirige a economía con el objetivo de sacar el dinero en usd o en euros según sea el país de destino. Conjuntamente con este proceso se llenan los modelos control de salida de personal (RI-4) y control de misiones (RI-5), se confecciona la carta impuesto aeropuerto firmada por el MIC y se hace entrega al trabajador de la resolución sobre las obligaciones del viajero que se encuentra en trámites de aprobación, la misma incluye como actividad importante la lectura del código de ética documento que puede ser consultado en la propia dirección de cooperación internacional. Concluido estos trámites se entrega la documentación de viaje al interesado (pasaje, pasaporte y carta de impuesto de aeropuerto). También se le informa al trabajador que debe presentarse en la caja central del departamento de contabilidad y finanzas perteneciente a la vice-rectoría económica para recibir el efectivo que cubrirá los gastos del viaje los cuales deberá liquidar una vez que regrese al país. Una vez de regreso el trabajador debe entregar su pasaporte en la Dirección de Cooperación Internacional antes de las 72 horas y se procede a llenar el modelo RI-7 de liquidación de gastos de viaje, además debe traer las justificantes de los gastos incurridos por conceptos que no son de dietas, incluyendo el comprobante de la tasa de cambio vigente del país en cuestión, se adjunta a dicho modelo una fotocopia del pasaporte donde se ponen los cuños de entrada y salida del país, el cual es firmado por Departamento de trámite y por dicho trabajador. Este modelo RI-7 debe ser entregado a economía en un plazo no mayor de 72 horas, la liquidación en economía debe efectuarse antes de los 10 días hábiles. En el Grupo de Trámites procede a abrir un expediente por cada persona, el informe de viaje conformado por el trabajador (de acuerdo a un modelo que se le entrega)

debe entregarse antes de los 15 días a los especialistas del Grupo de Cooperación en función del país que atiendan.

3.5 Descripción del proceso de conversión a UML.

Cuando se va a informatizar una empresa, el primer paso consiste en analizar lo que debe hacer, o sea, sus procesos de negocio. Si se lleva esto al proceso unificado de desarrollo del software, se estaría hablando del análisis de requerimientos, en el cuál se descubren los procesos y las necesidades de los negocios, y se expresan en los casos de uso. Es por eso que se puede decir que al enfrentarse a un diagrama de flujo de procesos, si ya se tienen bien claro de los capítulos y apéndices anteriores el concepto de dichos diagramas, se verá cómo estos procesos a los cuáles se les confecciona el diagrama de flujo posterior a su automatización, no serían más que los casos de uso. El siguiente paso sería identificar los papeles de las personas que intervendrán en los procesos, siguiendo algún tipo de metodología como RUP, este paso les recuerda el análisis del dominio orientado a objetos, que se expresa con un modelo conceptual. Este modelo representa las diversas categorías de las cosas en el dominio, no solo los papeles de las personas, sino también todo lo que tenga interés para los desarrolladores. Y este segundo paso también se encuentra de alguna forma en la descripción que hacen los ingenieros industriales de los diagramas de flujo de procesos. Así que como se puede ver, ya al unificar los esfuerzos de ambas ramas, se lograría mejorar la calidad y la rapidez con que se realizaría el proceso de informatización. Una vez identificados los procesos de la empresa y el personal, es el momento de determinar la manera de cumplir los procesos. También se deberá definir de que manera los empleados colaborarán o compartirán el trabajo, esta actividad será la asignación de responsabilidades, cosa que en la mayoría de los casos se expresa gráficamente con diagramas de diseño de clases y con diagramas de colaboración. La parte de análisis de esta tarea ya estará casi realizada si se toma como guía los diagramas de flujo de procesos, como aquí se propone. Con esto se quiere decir que si se sigue esta metodología, ya se tendría una completa definición de la planeación inicial. Solo quedaría ver la forma de lograr interpretar el proceso a seguir para su conversión a UML. Esto se mostrará paso a paso mediante un ejemplo. Para ello se utilizarán estándares que estén presentes en la mayoría de los procesos de desarrollo. El caso de estudio que se ha escogido para presentar este proceso será el punto de venta ya que ha sido abordado en varias bibliografías (aquí solo se escoge del caso de estudio los elementos necesarios para presentar la metodología deseada, si se desea estudiar más a fondo este ejemplo, consúltese Coad95 o Larman04), por lo que todos deben estar familiarizados con él, así que

servirá también para comparar las ventajas de esta unificación de áreas. A continuación se comenzará describiendo en que consiste el mismo:

La Terminal de Punto de Venta es un sistema automatizado con el que se registran las ventas y se realizan los pagos; normalmente se utiliza en las tiendas. Abarca componentes de hardware (una computadora y un lector de código barras) y software para correr el sistema.

Suponga que se ha pedido crear un programa para una Terminal de Puntos de Venta. Con una estrategia de desarrollo que partirá de los diagramas de flujo de procesos y la documentación generada por los ingenieros industriales, es por esto que cuando vaya a modelar este software, ya se contará con los diagramas y descripciones de todo el flujo de procesos. En este paso los ingenieros industriales habrán analizado todos los procesos del negocio, entre los que se destacan: Registrar los productos, entrega del cambio, compra de productos y pago de productos. En cada documento generado por los mismos se explicarán en detalles los pasos a seguir en cada uno de ellos. A continuación se toma compra de productos para explicar la metodología por ser uno de los más abarcadores. De este proceso se tendrá al comenzar el desarrollo del software, su respectivo diagrama de flujo de procesos y un documento con su explicación:

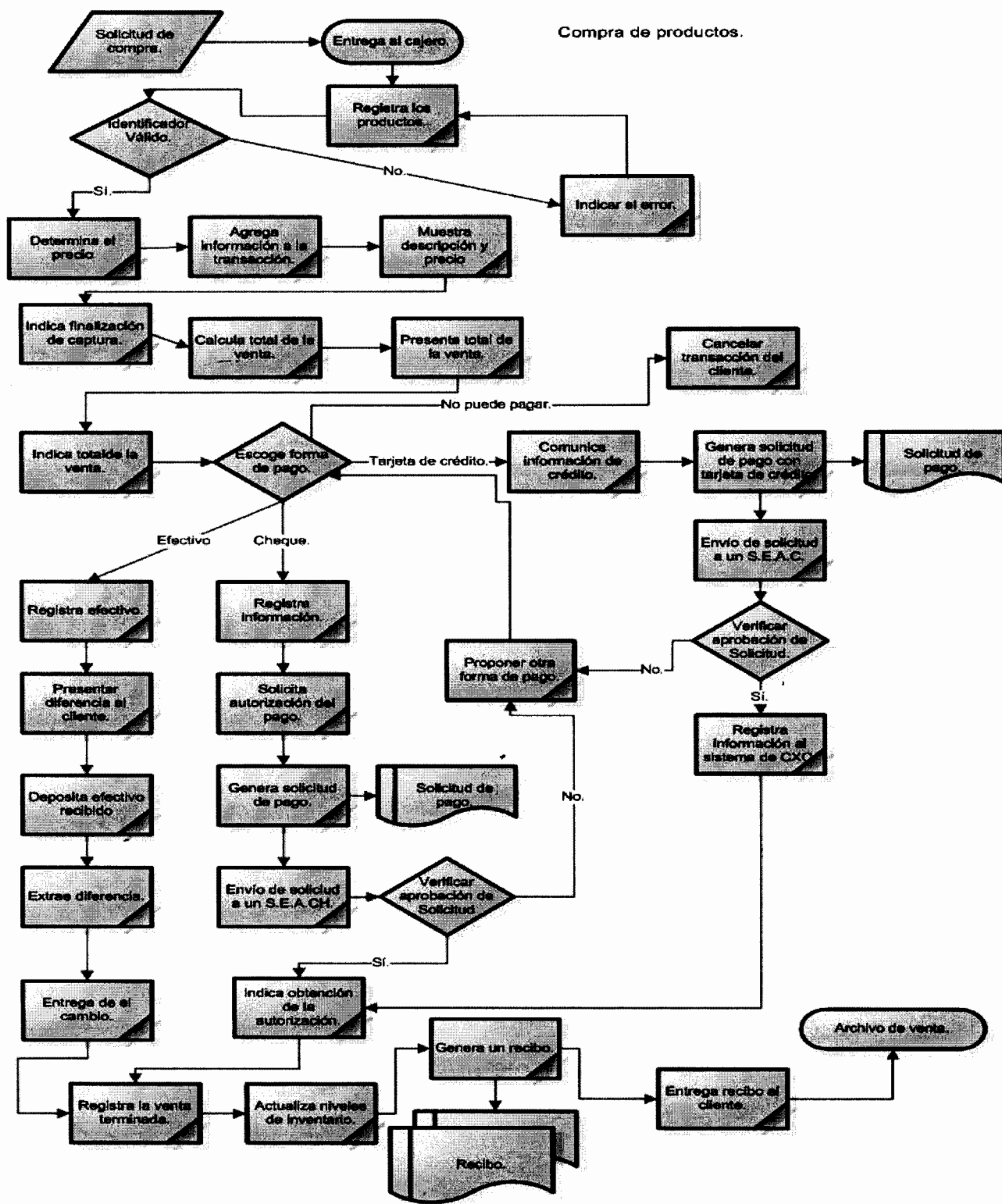


Figura 25: Diagrama de flujo de procesos para la compra de productos.

El documento donde aparecerá la descripción mostrará lo siguiente:

“El proceso de compra de productos comienza cuando un Cliente llega a la caja de la Terminal de punto de venta (TPDV) con productos que desea comprar, los cuales le muestra al cajero, para esperar a que este le registre el código universal de los productos, si se introduce un identificador no válido del producto se debe indicar el error. En caso contrario se determina el precio de cada uno, agregándose la información a la transacción de venta y se le muestra la descripción y el precio del producto actual al cliente. Se debe permitir al Cajero seleccionar una cantidad dada de un producto. Al terminar la captura de los productos el Cajero indica a TPDV que finalizó esta tarea y se calcula y muestra el total de la venta al Cliente, el cual escoge la forma en la que desea pagar. Si desea pagar en efectivo, y no tiene suficiente dinero, puede cancelar la transacción o inicializar otra forma de pago. En caso de tener más efectivo que el saldo total de la transacción entonces el Cajero registraría el efectivo ofrecido, presenta la diferencia al cliente, deposita el efectivo recibido y extrae de la caja la diferencia existente la que le es entregada al Cliente. Si desea pagar con tarjeta de crédito (Credit card), el Cliente comunica su información de crédito, las cuáles se encuentran en una tarjeta magnética que el presenta, se generará una solicitud de pago con tarjeta de crédito, que no es más que un documento que será posteriormente enviado a un servicio externo de autorización de crédito (SAC) el cuál se encargará de comprobar y autorizar o denegar el pago. En caso de autorizar el pago por el S.A.C., se recibe una autorización de pago, y en el sistema de cuentas por cobrar se registra la información referente al pago con tarjeta de crédito y la respuesta de aprobación. El servicio de autorización de crédito debe dinero a la tienda, por tanto, se le debe dar seguimiento por parte de las CxC. En caso de que el S.A.C. deniegue el pago se le propone al cliente otro método de pago. Si desea pagar con un cheque, el cajero registra la información sobre la identificación y solicita la autorización del pago con cheque, para esto genera una solicitud de pago con cheque que no es más que un documento que será enviado a un servicio externo de autorización de cheques para que se verifique el cheque. En caso de negar la solicitud del cheque el cajero propondría un nuevo método de pago. En caso de aprobarse la solicitud, se recibe una autorización de pago, y se indica la obtención de la autorización. Si el cliente no puede pagar de ninguna forma, se procedería a cancelar la venta. Después de ya haberse pagado el precio resultante de la venta, se registra en el sistema la venta como terminada, se actualizan los niveles de inventario y se genera un recibo, el cuál se le entrega al cliente.”

Cuando el desarrollador de software se enfrente a esto, lo primero que tiene que tener claro es que un diagrama de flujo de procesos será prácticamente un diagrama de actividades sin marco de responsabilidad, pero no obstante, mediante una correcta interpretación del documento que lo acompaña se pueden obtener los marcos de responsabilidad, ya que en este documento se explica en detalles quien

realiza cada actividad. A continuación se representa el diagrama de actividades para el proceso que hemos venido desarrollando (compra de productos), para no hacerlo muy complejo, ya que ese no es el objetivo, se formula aquí que pago en efectivo, pago con tarjeta de crédito y pago con cheque van a ser casos de uso aparte que serán usados por el caso de uso comprar productos, de hecho esto es verdad, aunque se notaría en fases posteriores, incluso se verá también como dichos 3 casos de uso se generalizarían en uno que fuera pagar dinero, por ejemplo, pero bueno, esto se sale del análisis propuesto. Se debe tener en cuenta que este diagrama de actividades que aquí se muestra no tiene marcos de responsabilidad, esto se hace para mostrar el gran parecido que tiene con respecto a los diagramas de flujo de procesos, no obstante si desea verlo con las responsabilidades, en el capítulo posterior se muestran ejemplos con el diagrama de actividades con marcos de responsabilidad, dichas responsabilidades fueron extraídas del documento auxiliar que elaboran los ingenieros industriales, aunque como en todos los casos, siempre es necesario un poco de creatividad e ingenio. Obsérvese también que en este ejemplo inicial del proceso de conversión se dejaron fuera de ámbito las entidades que tienen que aparecer también en este diagrama, en los ejemplos abordados posteriormente sí se van a mostrar, aquí se obvian para evitar la complejidad del proceso de estudio en sus inicios, ya que este capítulo solo se referirá a la metodología de conversión, no se abordará mucho en cuanto a cada uno de los componentes que están presentes dentro de los artefactos del UML. Esta complejidad se dejará para la sección posterior.

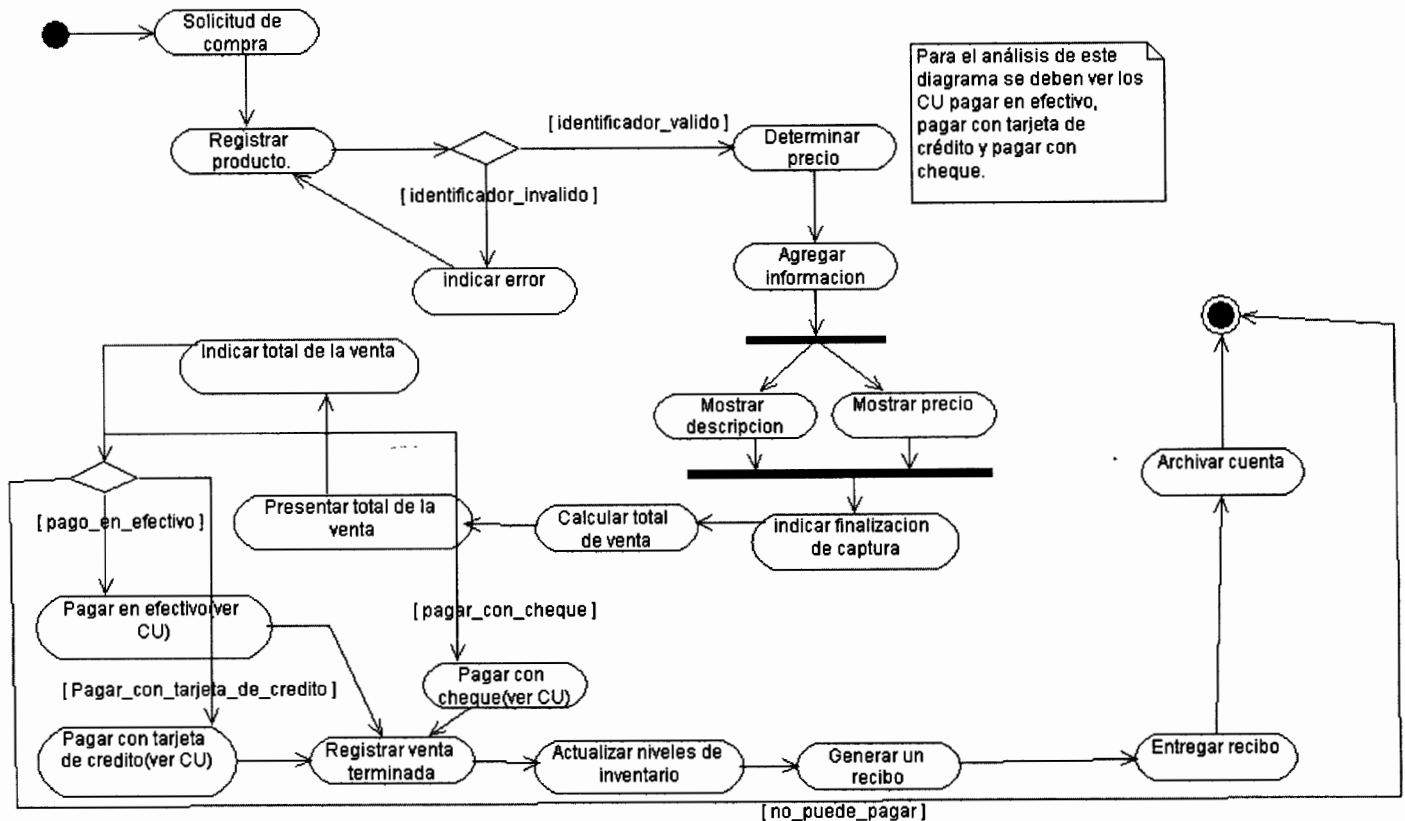


Figura 26: Diagrama de actividades sin marcos de responsabilidad del CU Comprar Producto.

Por otra parte, se debe dominar que un caso de uso es una descripción de un proceso de principio a fin relativamente amplia, descripción esta que suele abarcar muchos pasos o transacciones, por consiguiente un proceso no sería más que un caso de uso, pero a la hora de convertirlo se debe cambiar el nombre, ya que los mismos deben comenzar con un infinitivo. Además, aunque ya esto se sale de los objetivos de la metodología, es mejor aclarar que en numerosas ocasiones se considera uno que otro paso de un caso de uso como un CU en específico, y esto constituiría un error, se debe tener mucho cuidado con esto, aunque tampoco debe uno confiar ciegamente en los diagramas de flujo de procesos. Además cuando vayan a convertirlos a casos reales de uso del sistema, puede que una de estas actividades se convierta en un caso abstracto de uso. Como ya se dijo, la definición de este proceso constituiría un caso de uso del negocio, la descripción textual del mismo se puede extraer de la descripción del flujo, a continuación se muestra como quedaría:

CU Comprar productos: Este caso de uso se inicia cuando un cliente llega a una caja con productos que desea comprar. El Cajero registra los productos y obtiene el pago. Al terminar la transacción, el cliente se marcha con los productos.

También, después de haber extraído de los diagramas de flujo de procesos los casos de uso del negocio, serán capaces de crear el artefacto más importante a crear durante el análisis orientado a objetos: el modelo conceptual o modelo de negocio, en este caso de estudio, como no se tiene un negocio bien definido, se hará un modelo de dominio (modelo conceptual), ya que este explica los conceptos significativos de un dominio del problema. No se entrará mucho en detalles sobre que es un modelo conceptual, pero sí es importante aclarar que el mismo debe representar como cualidad esencial las cosas del mundo real, no los componentes del software.

Una técnica muy útil para la conformación del modelo conceptual consiste en la obtención de los conceptos a partir de las frases nominales de un problema y considerarlas conceptos o atributos idóneos. Se dice que esto también saldría de los diagramas de flujo de procesos ya que los casos de uso se utilizan para este análisis, no obstante también salen de la definición de los conceptos, y estos de una u otra forma también son abordados en la descripción textual de los diagramas de procesos. Por ejemplo del proceso en función (compra de productos) se pueden extraer los siguientes conceptos: Cliente, caja de TPDV, productos, Cajero, Código universal de productos (CUP), cantidad, precio del producto, transacción de ventas, descripción, entre otros. A continuación se muestra como debería quedar el modelo conceptual una vez analizados todos los conceptos que forman parte de nuestro caso de estudio, para una mayor referencia consultar [Larman04]:

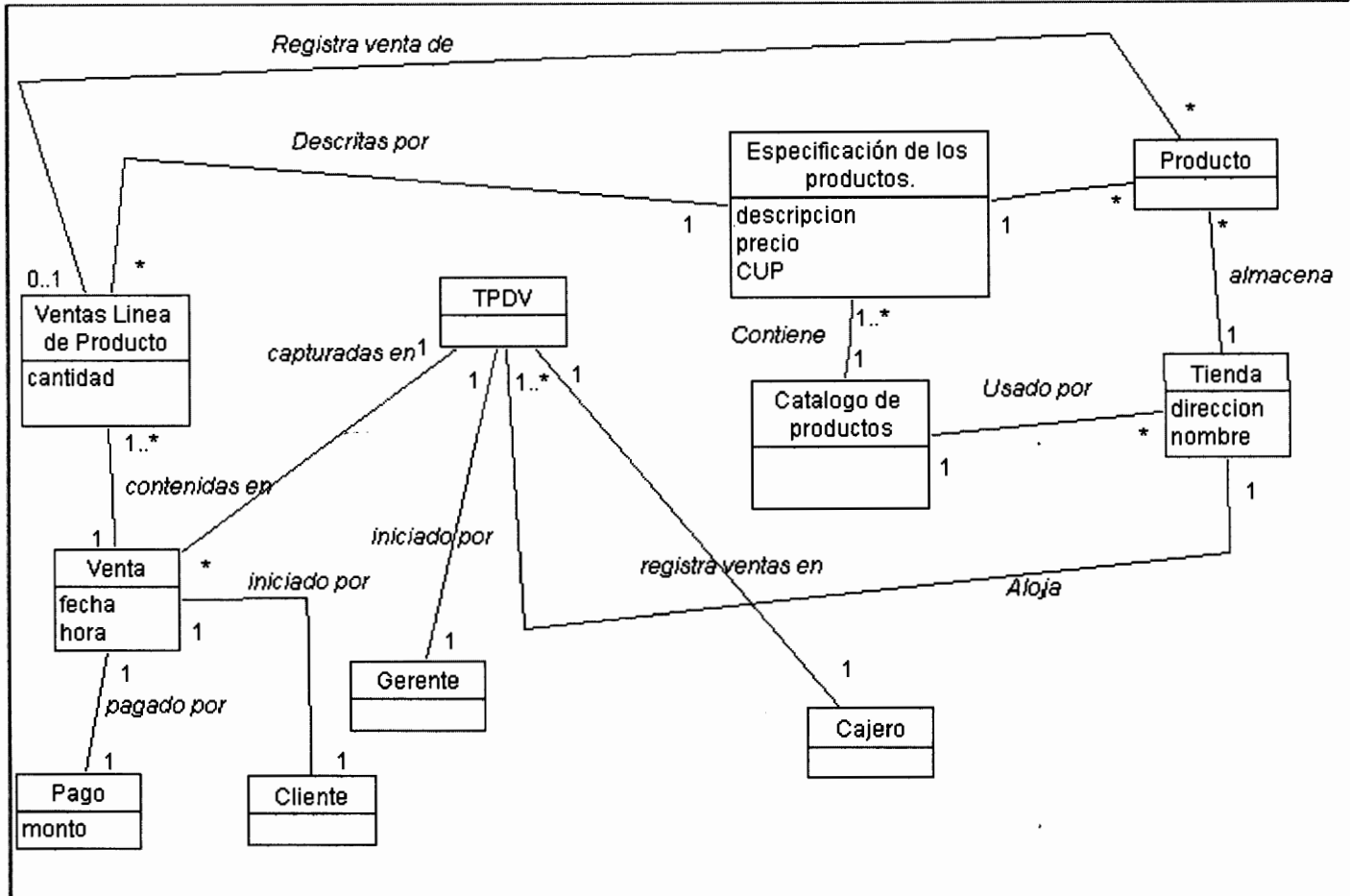


Figura 27: Modelo conceptual del dominio del punto de venta.

Después de haber analizado todos los procesos que se tengan, se extraerían los requisitos, tanto funcionales como no funcionales, y se modelan los funcionales en términos de casos de uso del sistema ya que un proyecto no puede tener éxito sin una especificación correcta y exhaustiva de los requerimientos. Para ello se necesitan muchas habilidades, un planteamiento riguroso de la forma de hacerlo ya se saldría del campo de acción propuesto, no obstante, de este caso de uso que se está siguiendo (comprar productos) se sacarían los siguientes requisitos funcionales:

- 1.1. Registra la venta en proceso (los productos comprados).
- 1.2. Calcula el total de la venta actual; aquí se incluyen el impuesto y los cálculos de cupón.
- 1.3. Captura la información sobre el objeto comprado usando su código de barras y un lector o usando una captura manual de un código del producto; por ejemplo el código universal del producto (CUP).
- 1.4. Ofrece un mecanismo de almacenamiento persistente.

1.5. Muestra la descripción y el precio del producto registrado.

1.6. Maneja los pagos en efectivo, capturando la cantidad ofrecida y calculando el saldo deudor.

Este mismo caso de uso se podría representar en su formato de alto nivel o si se requiere, se podría usar el formato expandido. En este caso se va a representar por su nivel de detalle en un caso esencial de uso expandido, aunque esto no quita que quizás por problemas de no querer complicar mucho el entendimiento en las fases iniciales, no se quiera utilizar su formato expandido, aunque siempre se deben especificar en las fases iniciales al menos los más importantes (los casos primarios de uso). A continuación se muestra como quedaría este proceso convertido a diagrama esencial de caso de uso expandido:

Caso de Uso:	Comprar productos.
Actor(es):	Cliente (inicia), cajero
Propósito:	Capturar una venta y su pago en efectivo.
Resumen:	El caso de uso inicia cuando un Cliente llega a la caja registradora con los artículos que comprará. El Cajero registra los artículos y cobra el importe. Al terminar la operación, el Cliente se marcha con los productos.
Tipo:	Primario y esencial.
Referencias:	R 1.1, R 1.2, R 1.3, R 1.7, R 1.9, R 2.1.
Curso normal de los eventos:	
Acción del actor:	Respuesta del sistema:
1. Este caso de uso comienza cuando un Cliente llega a una caja de TPDV (Terminal de Punto de Venta) con productos que desea comprar.	
2. El Cajero registra el código universal de productos (CUP). Si hay más de un producto, también puede introducirse la cantidad.	3. Determina el precio del producto e incorpora a la transacción de venta actual la información correspondiente. Se muestran la descripción y el precio del producto actual.
4. Al terminar de introducir el producto, el Cajero	5. Calcula y presenta el total de la venta.

indica a TPDV que se concluyó la captura de los productos.	
6. El Cajero le indica el total al Cliente.	
7. El Cliente escoge la forma de pago: a. Si paga en efectivo, ver la sección Pagar en efectivo. b. Si paga con tarjeta de crédito ver sección Pagar con tarjeta de crédito. c. Si paga con cheque, ver la sección Pagar con cheque.	8. Registra la venta terminada.
	9. Actualiza los niveles de inventario.
	10. Genera un recibo.
11. El Cajero entrega el recibo al cliente.	
12. El Cliente se marcha con los productos comprados.	
Cursos Alternos: Línea 2: se introduce un identificador inválido del producto. Indique el error. Línea 7: el Cliente no pudo pagar. Cancele la transacción de venta.	
Sección: Pagar en efectivo.	
Curso normal de los eventos:	
Acción de los actores:	Respuesta del sistema.
1. El Cliente da un pago en efectivo, posiblemente mayor que el total de la venta.	
2. El Cajero registra el efectivo ofrecido.	3. Presenta la diferencia al Cliente.
4. El Cajero deposita el efectivo recibido y extrae la diferencia.	
5. El Cajero le entrega el cambio al Cliente.	
Cursos Alternos: Línea 1: el Cliente no tiene suficiente efectivo. Puede cancelar o iniciar otro método de pago.	
Sección: Pagar con tarjeta de crédito.	
Curso normal de los eventos:	

Capítulo 3: Descripción de la propuesta diseñada.

Acción de los actores:	Respuesta del sistema.
1. El Cliente comunica su información de crédito para pagar con tarjeta.	2. Genera una solicitud de pago con tarjeta de crédito y la envía a un servicio externo de autorización de crédito.
3. El servicio de autorización de crédito autoriza el pago.	4. Recibe una respuesta aprobatoria de crédito del servicio de autorización de crédito.
	5. En el sistema de cuentas por cobrar registra la información sobre el pago con tarjeta de crédito y la respuesta de aprobación. El servicio de autorización de la Tienda debe dinero, por tanto, cuentas por cobrar debe darle seguimiento.
	6. Muestra el mensaje aprobatorio de autorización.
Cursos Alternos: Línea 3: Solicitud de crédito negada por el servicio de autorización de crédito. Proponer otra forma de pago.	
Sección: Pagar con cheque. Curso normal de los eventos:	
Acción de los actores:	Respuesta del sistema.
1. El Cliente extiende un cheque y se identifica.	
2. El Cajero registra la información sobre la identificación y solicita la autorización del pago con cheque.	3. Genera una solicitud de pago con cheque y la envía a un servicio externo de autorización de cheques.
4. Verifica que el pago haya sido autorizado por el servicio de autorización de cheques.	5. Recibe una respuesta aprobatoria del servicio de autorización de cheques.
	6. Indica la obtención de la autorización.
Cursos Alternos: Línea 4: Verificar solicitud negada por el servicio de autorización de cheques. Proponer otra forma de pago.	

Tabla 2: Descripción de CU. Comprar Productos.

Como se puede observar, el proceso no es realmente difícil, lo único que hay que hacer es ir siguiendo el diagrama de flujo de procesos y el documento generado por los ingenieros industriales, y utilizar un

poco de sentido común a la hora de clasificar cada actividad concreta como una acción específica del actor o una respuesta del sistema. Para continuar con la tarea debe quedar claro de las secciones anteriores los símbolos que componen los diagramas de flujo, ya que ellos son, como ya se dijo, diagramas de actividades haciéndoles uno que otro cambio.

Después de haber realizado esta labor se seguirían guiando por la metodología de desarrollo que se esté siguiendo, teniendo siempre presente que ya se cuenta con una herramienta más eficiente que ayudaría a la hora de realizar otras tareas como por ejemplo que no se tendría que pasar mucho trabajo realizando los diagramas de actividades, además de que el trabajo desarrollado por los ingenieros industriales sería muy útil pues daría un mayor conocimiento sobre el dominio que se esté modelando pues sería una fuente de información extra con que se contaría, y a la hora de elaborar los diagramas de UML será de gran ayuda como se verá en la siguiente sección.

3.6 Estrategia de continuación.

En el apéndice anterior ya quedó claro todo lo referente a la forma de incluir los diagramas de flujo de procesos dentro del ciclo de vida del desarrollo del software, a continuación se mostrará una estrategia de continuación para crear los diagramas pertinentes de UML. No se mostrarán todos, se hará solo para el caso de uso mostrar productos, ya que con esto bastará para comprenderlo. Aunque ya el momento de creación de cada uno de estos diagramas dependerá del ciclo en que se encuentren, y de la metodología que se esté utilizando.

3.6.1 Diagramas de secuencia del sistema.

El diagrama de secuencia de un sistema muestra gráficamente los eventos que fluyen de los actores al sistema. El UML ofrece una notación con los diagramas de secuencia que muestran gráficamente los eventos que pasan de los actores al sistema. Normalmente estos diagramas se deben realizar durante la fase del análisis de un ciclo de desarrollo. Su descripción depende de la formulación previa de los casos de uso. El comportamiento del sistema es una descripción de lo que hace, sin entrar en detalles sobre el cómo se hace.

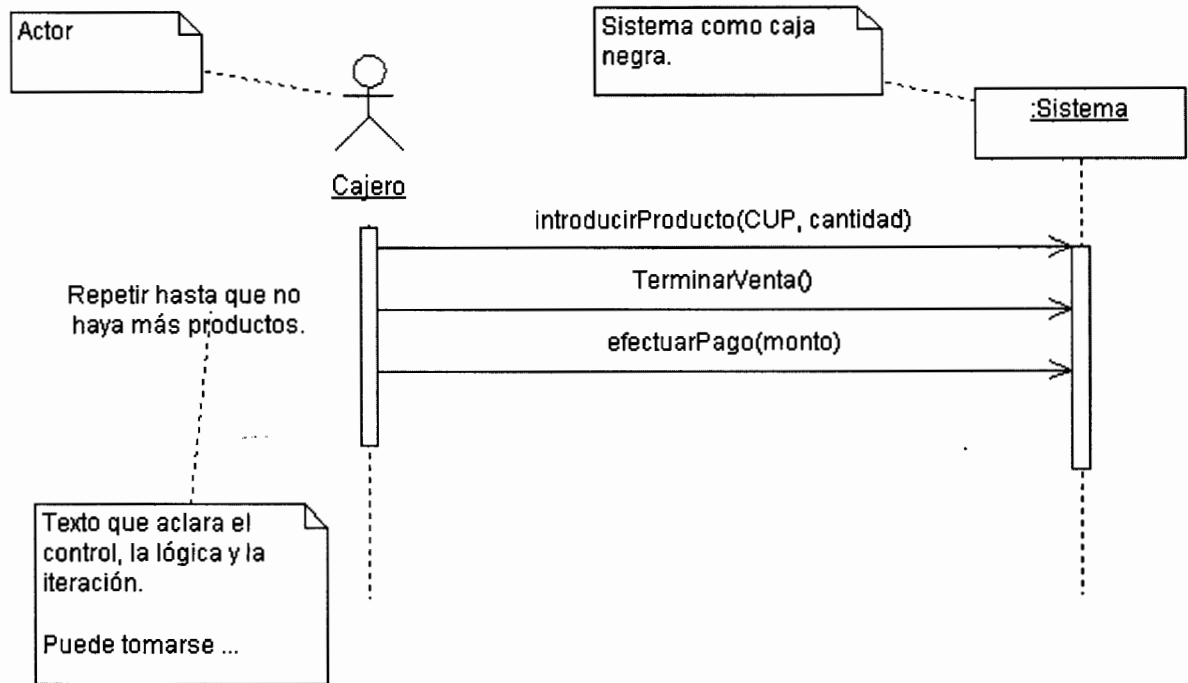


Figura 28: Diagrama de secuencia del sistema para el caso de uso Comprar productos.

Por si quedan dudas de cómo se extrajeron estas operaciones, aquí se muestra de donde salió cada una de ellas.

En todos los productos, el Cajero registra el código universal de producto (CUP) y la cantidad.	introducirProducto(CUP,cantidad)
Al terminar de capturar el producto, el Cajero indica a la TPDV que la venta concluyó.	terminarVenta()
El Cajero le indica el total al Cliente, y éste le da un pago. El Cajero registra el importe recibido en efectivo.	efectuarPago(monto)

Tabla 3: Explicación de la secuencia.

3.6.2 Diagramas de Colaboración.

Otro de los diagramas que posee UML son los diagramas de interacción, que explican gráficamente cómo los objetos interactúan a través de mensajes para llevar a cabo las tareas. Por lo general estos diagramas se realizan en la fase de diseño de un ciclo de desarrollo, no es posible realizarlos si antes no fueron generados los siguientes artefactos:

- I. Un modelo conceptual: a partir de este modelo el diseñador podrá definir las clases del software correspondiente a los conceptos. Los objetos de las clases participan en las interacciones que se describen gráficamente en los diagramas.
- II. Contratos de la operación de sistema: A partir de ellos los diseñadores identifican las responsabilidades y las poscondiciones que han de llenar los diagramas de interacción.
- III. Casos reales de Uso(o esenciales): a partir de ellos el diseñador recaba información sobre las tareas que realizan los diagramas de interacción, además de lo estipulado en los contratos.

Los diagramas de colaboración son considerados de mayor usabilidad que los de secuencia por su excepcional expresividad, su capacidad de comunicar más información y su economía de espacio, para su diagramación se utilizará el caso de estudio que aquí se presentó y con el cuál todos deben estar familiarizados, en el diagrama de colaboración que se mostrará como ejemplo, se incluye al actor Cajero, debe quedar claro que esto es solo para aclarar que él sería el que iniciaría esta operación, pero normalmente esto no es necesario ponerlo. Además también se tuvieron en cuenta algunos patrones de diseño para asignar responsabilidades (Grasp) como es el caso del patrón creador, y el patrón experto:

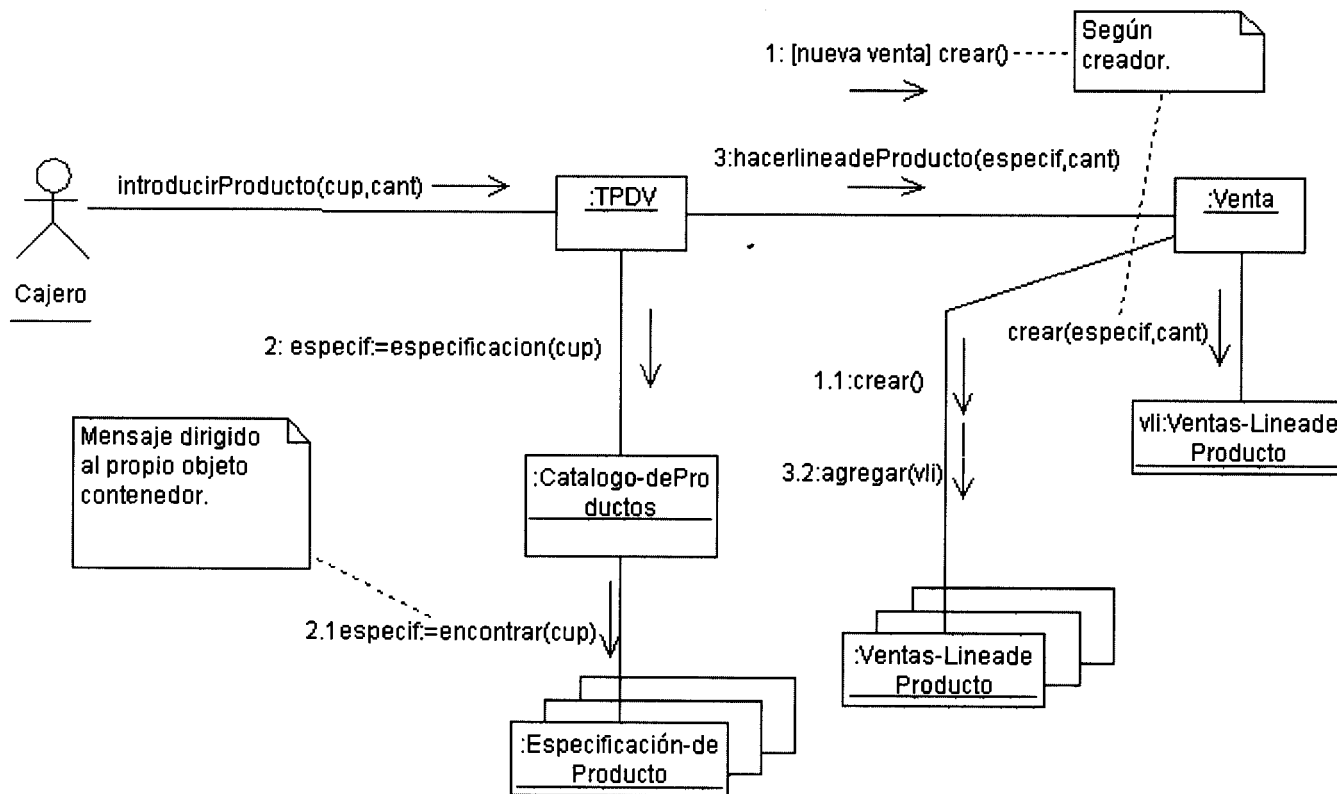


Figura 29: Diagrama de colaboración introducir Producto.

3.6.3 Diagramas de estado.

El UML cuenta con una notación para los diagramas de estados que describen gráficamente los eventos y los estados de los objetos, así como los diversos comportamientos. No es necesario que muestre todos los eventos posibles, si tiene lugar un evento que no está representado en el diagrama, se excluye siempre y cuando no sea relevante en el diagrama de estados. Se pueden utilizar para clases de software, tipos (conceptos), aunque su mayor uso es para los casos de uso. Para estos últimos son de gran utilidad ya que con un conjunto de diagramas de estado, el diseñador podrá desarrollar metódicamente un diseño que garantice el orden correcto de eventos del sistema. Además en un dominio con muchos eventos del sistema, la concisión y el rigor al emplear los diagramas de estado de casos de uso le ayuda al diseñador a cerciorarse de que no está omitiendo nada. A continuación se muestra como quedaría el diagrama de estado del caso de uso comprar productos.

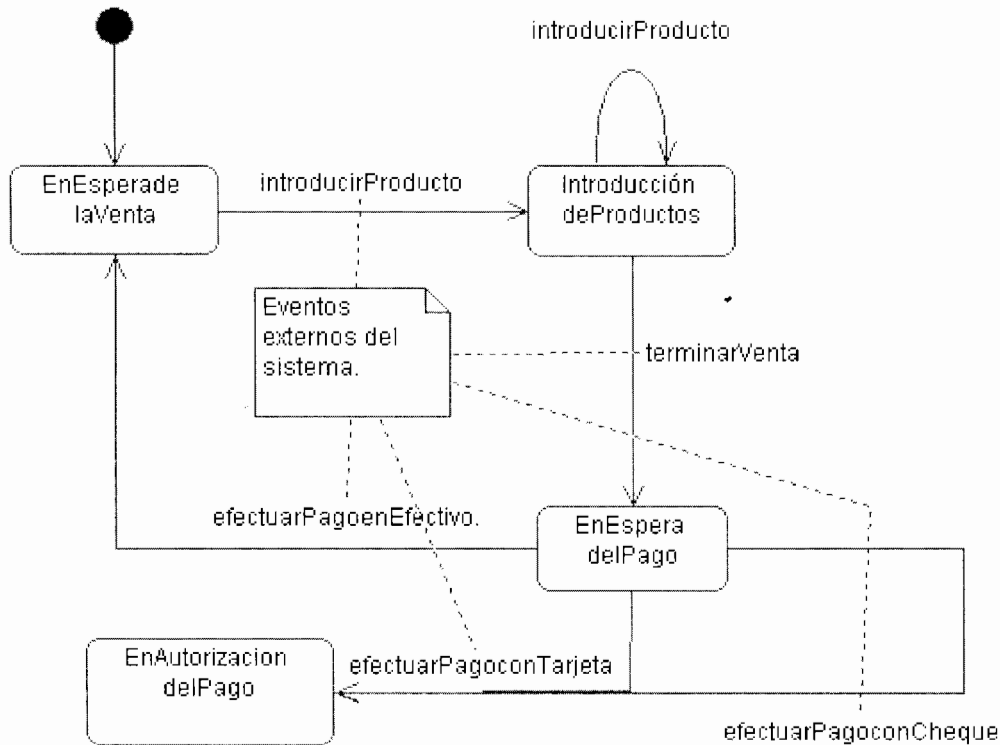


Figura 30: Diagrama de estado para el caso de uso comprar producto.

Si aquí se hubiera representado este diagrama con todos los estados que puede contener, se observara la gran semejanza que tiene con el de actividades descrito anteriormente, pero por cuestiones

entendimiento se omitieron elementos. Para una mayor complejidad de este diagrama se pueden consultar los patrones de GoF y Grasp [GHJV95].

Para una mayor información respecto al modo de continuar trabajando puede consultarse la bibliografía recomendada, de aquí ya se pueden representar todos los modelos de UML que se necesiten para posterior desarrollo. Debe aclararse que esta representación de diagramas que se están mostrando corresponden solo a las primeras iteraciones, el objetivo que se persigue no es el de volverlos expertos en este caso de estudio, solo se desea ver la forma de cumplimentar la situación problemática. Por si aún no se ha comprendido la gran ayuda que reportan los diagramas de flujo de procesos, a continuación se representarán las dependencias de los artefactos del UML.

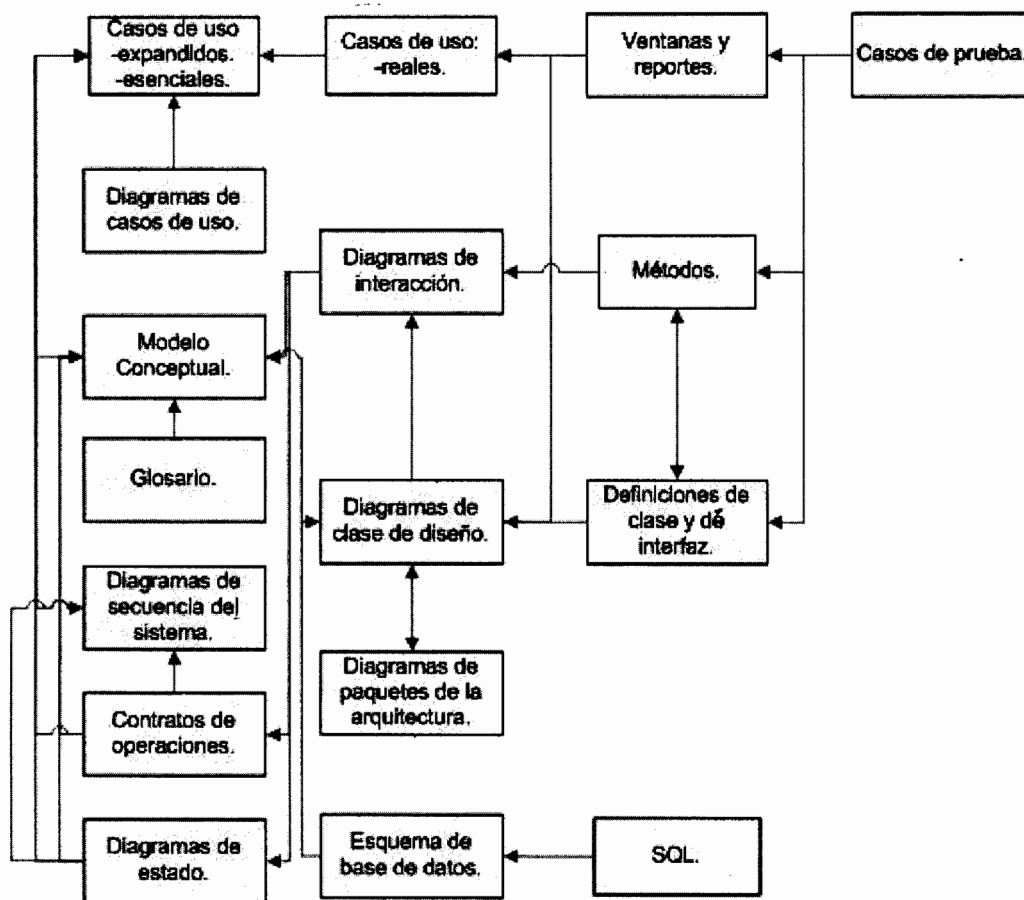


Figura 31: Dependencias de los artefactos durante la fase de construcción.

3.7 Conclusiones.

En este capítulo se abordó la estrategia que debe usarse en el área de informatización para lograr todos los objetivos que se propongan en el menor tiempo posible y con una calidad incomparable. Al culminar el mismo deben haber quedado claro que los diagramas de flujo de procesos son de gran ayuda para el ciclo de desarrollo del software, ya que los mismos no representan la forma ordinaria de realizarse el proceso de negocio, sino que dicho proceso es primero automatizado extrayendo y eliminándose todos los cuellos de botellas. No obstante, por si aún quedan cosas sin aclarar, con el próximo capítulo deben quedar claras.

Capítulo 4. Ejemplos informatizados de la propuesta diseñada utilizando RUP.

4.1 Introducción.

El presente capítulo se encargará de despejar cualquier duda que haya quedado referente a la sección anterior, ya que se verán varios ejemplos mediante los cuáles se tratará de cubrir el proceso completo de conversión partiendo desde los diagramas de flujo de procesos, y demostrando de forma práctica toda la información que se puede sacar de ellos.

4.2 Ejemplos prácticos ya analizados.

En esta sección lo que se hará es desarrollar algunos ejemplos que ya fueron definidos por los ingenieros industriales, como ya se dijo, este es un proceso novedoso, y aún se encuentra en las fases iniciales de su implementación, en la universidad no se ha aplicado a todas las áreas de la misma forma, lo que sí queda claro es que esto es de gran ayuda a la hora de definir los negocios puesto que entonces el ingeniero informático no informatizaría un proceso que presenta fallas en su funcionamiento, sino que desarrollaría un proceso completamente automatizado. Una de las primeras áreas que se les dio la tarea de informatizar sus procesos a los ingenieros industriales fue la de los almacenes que existen en la UCI. A continuación se muestran los procesos que se consideraron de interés para entender su funcionamiento a la hora de informatizar dicha área. Se comenzará mostrando el modelo de flujo de procesos de los que fueron seleccionados para su análisis, con su respectiva descripción, para posteriormente demostrar el proceso explicado en el capítulo anterior.

4.2.1 Proceso de consulta de producto.

Lo primero que se realiza para consultar la existencia o no de un producto es buscar en la base de datos de productos si es que él se encuentra, en caso de encontrarse (se sabe que se encuentra en existencia si la cantidad real es mayor que cero) se informaría la disponibilidad del producto informándose además la cantidad real que se tiene en existencia. En caso contrario (la cantidad es cero) se informaría al usuario que el producto no se encuentra en existencia, además se mostraría de ser uno de los productos básicos si está en trámites de aprobación de solicitud o no, también el usuario tendrá la opción de realizar

una nueva búsqueda o finalizarla. Todo el proceso de manipulación de si el producto se encuentra disponible o no será realizado por el sistema informático ASSETS.

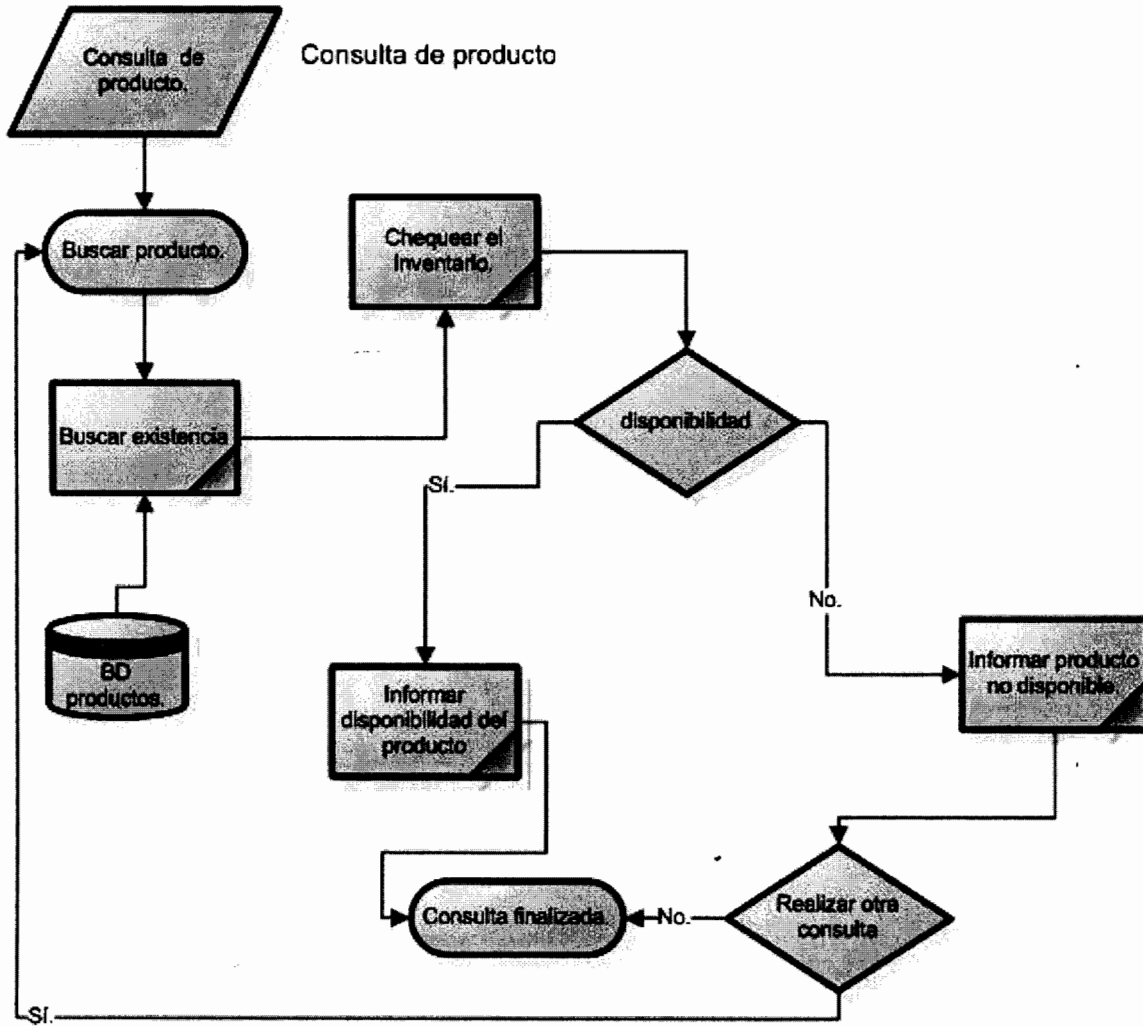


Figura 32: Diagrama de flujo para el proceso Consulta de Producto.

A continuación se mostrará la forma de llevar este proceso al UML, como lo que se está modelando aquí es el negocio, por ende este proceso sería convertido en un caso de uso del negocio, y de este se extraería su respectivo diagrama de casos de uso del negocio y su diagrama de actividades.

CU Consultar Existencia de Producto: Inicia cuando el usuario pregunta por la existencia de un producto determinado, el trabajador del almacén consulta el inventario a través del sistema ASSETS. Si la disponibilidad que aparece asociada al producto en cuestión es mayor que cero, entonces el producto está disponible; si es igual a cero, el producto no está disponible. El trabajador del almacén informa este resultado al usuario y finaliza el CU. Como se ve, este proceso de negocio propone una modificación con

respecto a cómo se está realizando en la actualidad. En lugar de consultar el inventario manual, el trabajador del almacén consulta directamente el inventario que mantiene ASSETS, que está centralizado.

Esta vez si se va a utilizar el diagrama de actividades con sus marcos de responsabilidades, cualquier duda se puede consultar el documento de interpretación del diagrama de flujo.

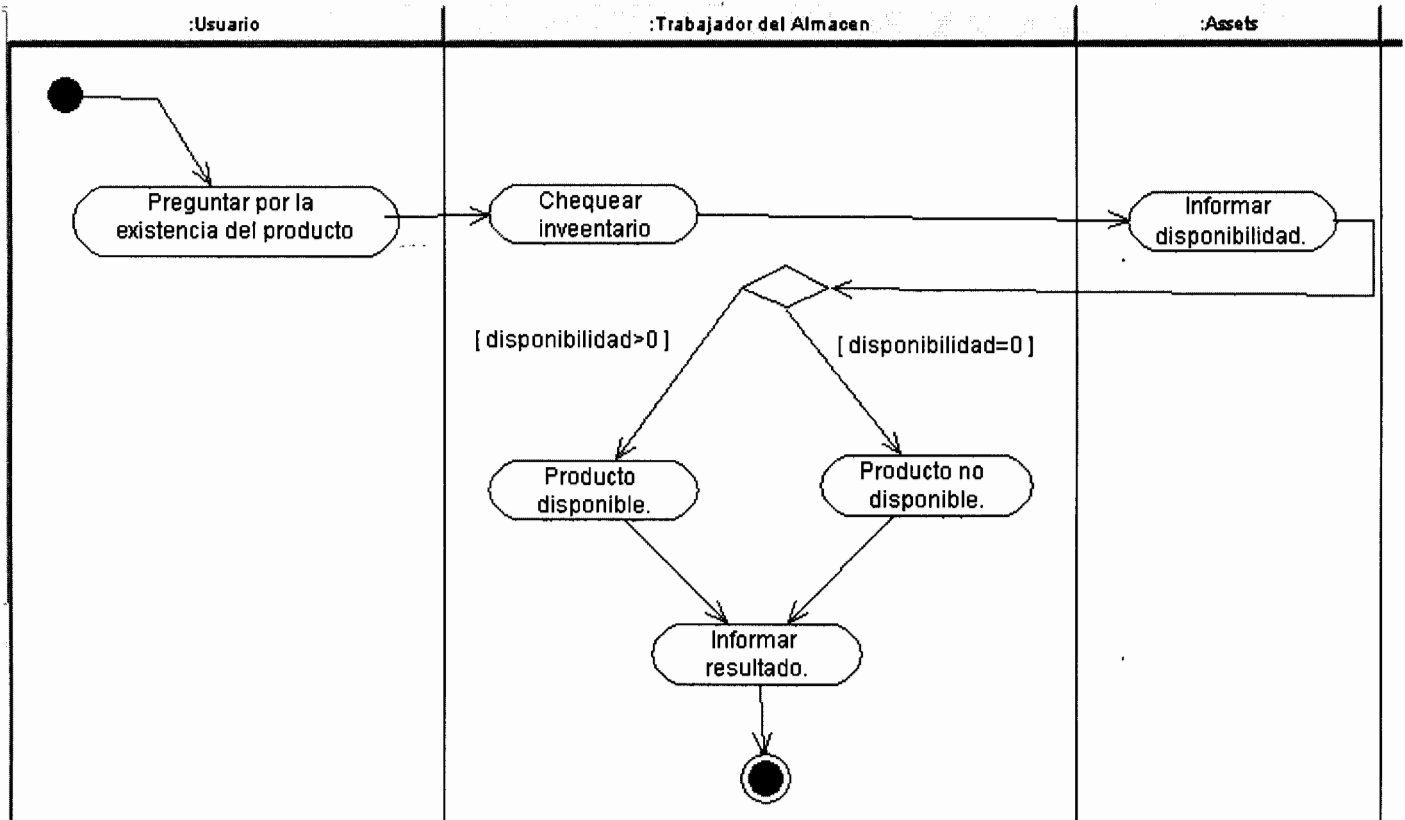


Figura 33: Diagrama de actividades para el caso de uso Consultar Existencia de Producto.

De aquí también se puede generar el modelo de objetos del caso de uso:

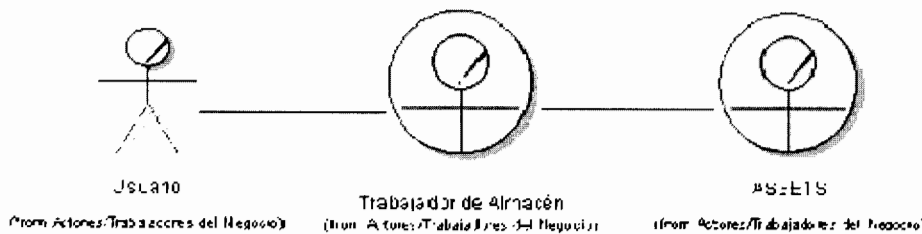


Figura 34: Modelo de Objetos del negocio para el CU Consultar Existencia de Producto.

A la hora de continuar llevando este caso de uso a UML se debe tener en cuenta si pertenece a un caso de uso del sistema o no. Cómo lo que se está haciendo es simplemente mostrando la metodología y

no desarrollando un caso en específico se dejará este ejemplo hasta aquí, se mostrará en ejemplos posteriores cómo se podría seguir hacia los diagramas de casos de uso del sistema.

4.2.2 Solicitud de producto.

Lo primero que se realiza es solicitar un producto determinado, esto se hace mediante la confección de un vale de solicitud de materiales del almacén por parte del cliente (este proceso incluye además el proceso de aprobar solicitud, ya que el vale debe contar con una firma acreditada y válida para la organización en funciones). Una vez que se aprueba y receptiona la solicitud de vale, el trabajador del almacén procede a entregarle los productos al cliente, posteriormente se marcaría la solicitud dada como que ya se despachó y después se crearía un vale de salida en formato digital utilizando para ello el sistema informático ASSETS que es el que se está utilizando en el departamento de contabilidad, el cuál sería almacenado en una base de datos. Posteriormente se confirmaría el vale por parte del trabajador del almacén para poder entonces actualizar los inventarios por parte de ASSETS y tomar la decisión de imprimir o no el modelo de vale. En caso de no imprimirse, se procedería a finalizar la solicitud. Si por el contrario, se decide imprimir el vale, se ordenaría la impresión del mismo por parte del trabajador del almacén, y el vale sería impreso por el sistema ASSETS, de las copias se le entregaría una al cliente y finalizaría la solicitud. Esta sería la descripción del siguiente diagrama:

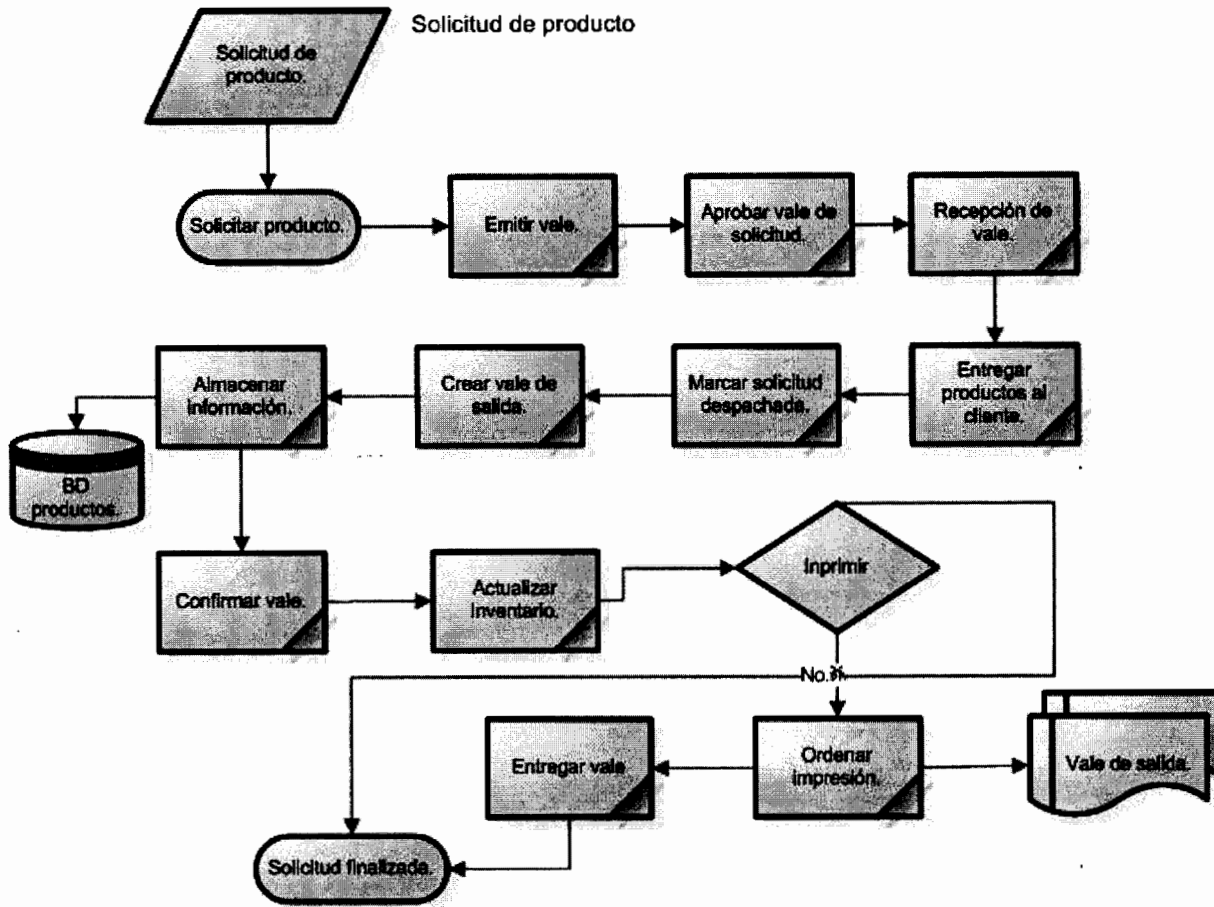


Figura 35: Diagrama de flujo para el proceso Solicitud de producto.

Este proceso llevado a caso de uso del negocio quedaría:

CU Solicitar Productos al Almacén: Inicia cuando un Cliente de Almacén emite un Vale de Solicitud de Materiales (incluye el CU Aprobar Solicitud, ya que dicho vale debe aprobarse por una persona con firma autorizada). Una vez aprobada la solicitud, el Cliente la envía al trabajador del almacén que le entrega los productos. Luego el trabajador del almacén introduce la información requerida en ASSETS para crear un vale de salida, lo confirma, con lo cual se actualiza el inventario digital mantenido por dicho sistema, y lo imprime si desea, para entregar una copia al cliente o para otro fin.

De aquí se extrae el diagrama de actividades con marcos de responsabilidad, apoyándose para esto en el trabajo realizado por los ingenieros industriales:

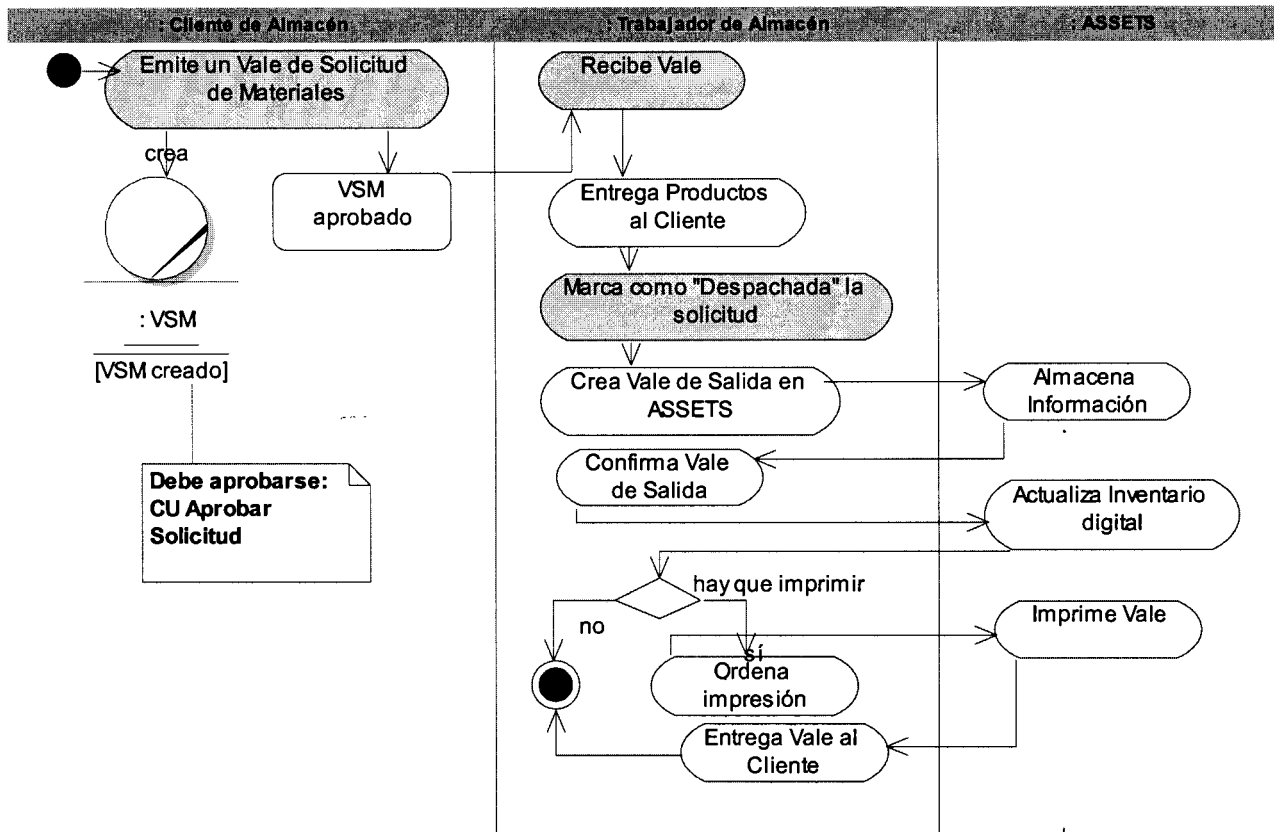


Figura 36: Diagrama de actividades para el caso de uso Solicitar Productos al Almacén.

El modelo de objetos del caso de uso quedaría:

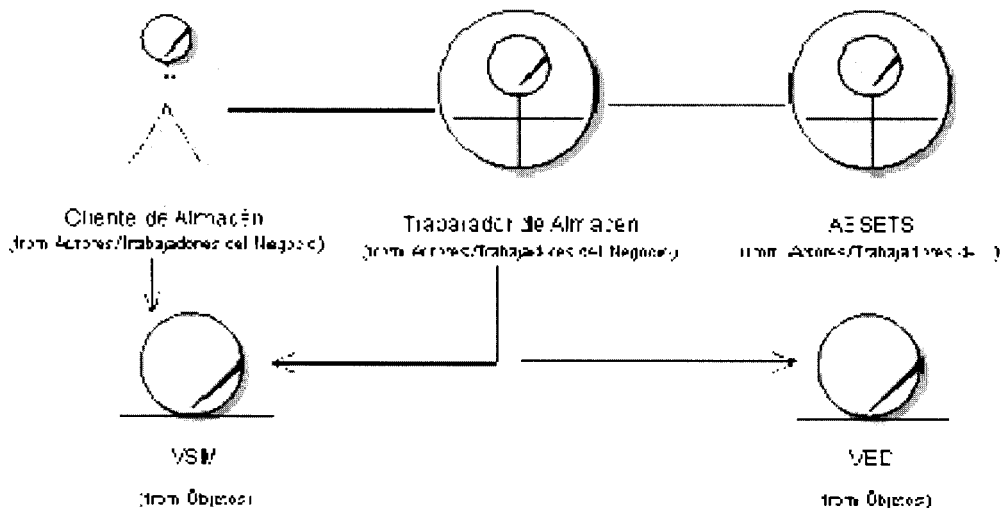


Figura 37: Modelo de objetos para el caso de uso Solicitar Productos al Almacén.

4.2.3 Aprobación de solicitud.

Este es un proceso interno, o sea él no se inicia por sí mismo, esto es debido a que forma parte de algún otro proceso. El mismo se inicia cuando llega una solicitud de aprobación con el objetivo de verificarla y decidirse si se aprueba o no. Se le entrega el vale por aprobar a la vicerrectora en funciones la cuál procede a consultar el material que se desea comprar. En caso de considerarse adecuada la solicitud, se procedería a enviar la confirmación de aprobación de dicha solicitud de compra y posteriormente se le informarían los resultados al cliente del almacén, después de esto habría finalizado la solicitud de aprobación. En caso de no considerarse adecuada, se denegaría el vale de solicitud y se le informaría al cliente del almacén los motivos por los cuáles se denegó la solicitud, dando unas recomendaciones pertinentes para tener en cuenta a la hora de solicitar una nueva aprobación.

Esta descripción se corresponde con el diagrama que aparece a continuación:

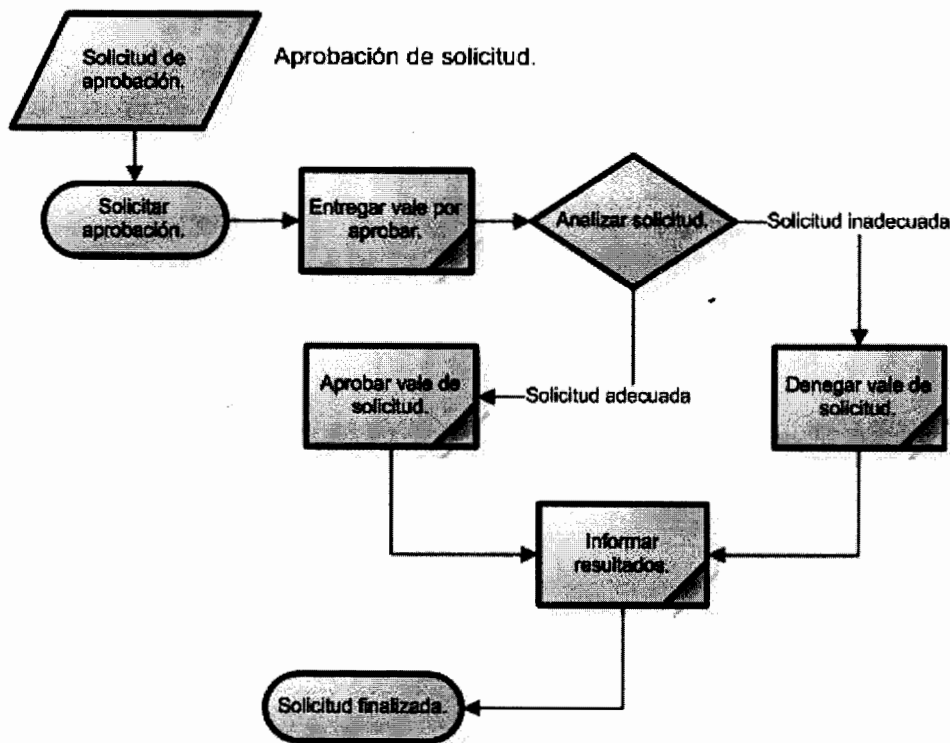


Figura 38: Diagrama de flujo para el proceso de Aprobación de solicitud.

Con esta información en las manos, se haría la conversión del mismo a CUN como bien se mostró anteriormente, quedando como descripción:

CU Aprobar Solicitud: Inicia cuando, por ser necesario, un Cliente de Almacén, entrega una solicitud a la persona con firma autorizada que debe aprobar la solicitud para que esta sea válida. La persona analiza la solicitud y decide aprobarla o no, en cualquier caso le informa del resultado al solicitante.

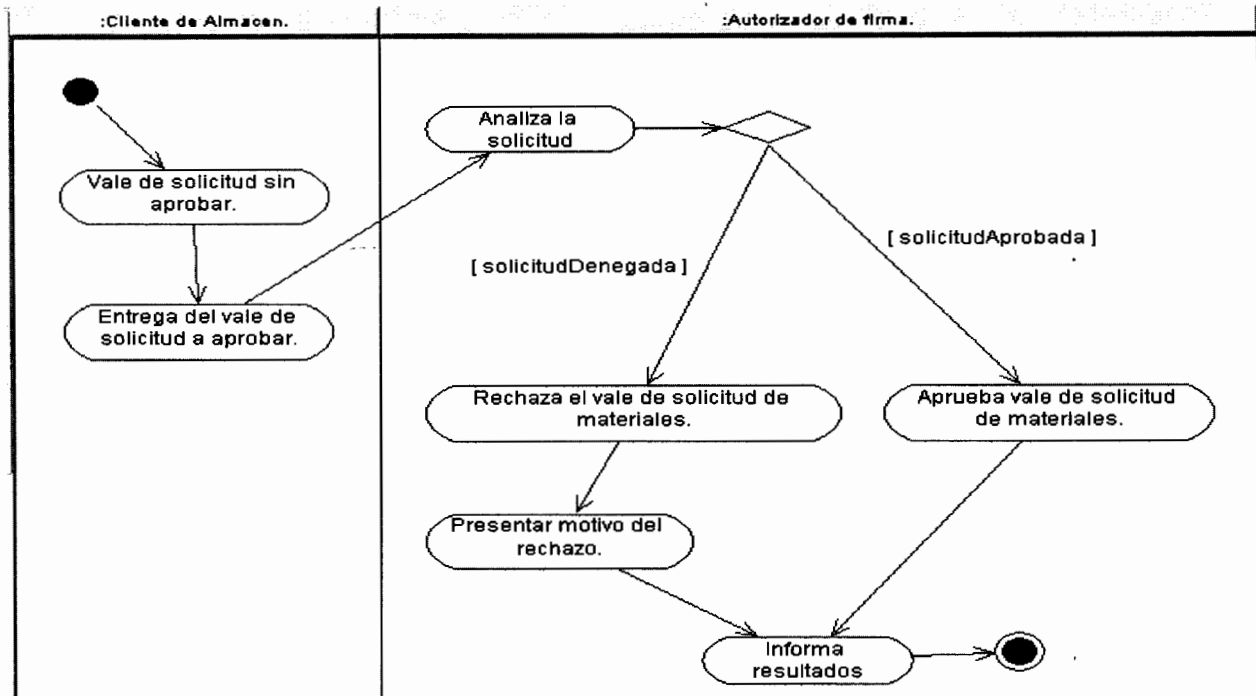


Figura 39: Diagrama de actividades para el caso de uso aprobar Solicitud.

De aquí se obtiene el modelo de objetos para este caso de uso:

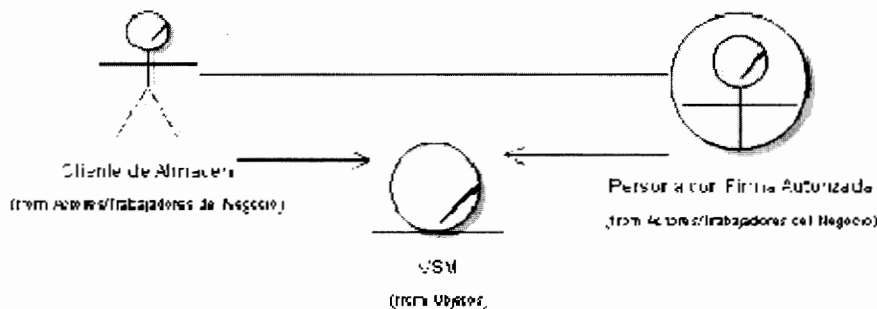


Figura 40: Modelo de objetos del negocio para el caso de uso aprobar Solicitud.

4.2.4 Comprar productos básicos.

Lo primero que se realiza en este proceso es reportar al jefe de compras y almacén de la escasez de los productos básicos, por lo que se decide ordenar la compra, posteriormente el especialista en compras

lleva a cabo un estudio de mercado que consiste en verificar posibles ofertas de compra y de vendedores, para tomar, de acuerdo al estándar de calidad requerido, los productos más asequibles. Después dicho especialista envía sus resultados al comité de compras el que tiene como función aprobar la compra. Este proceso concluiría en cuanto se compren los productos básicos necesarios.

El diagrama de flujo de este proceso queda:

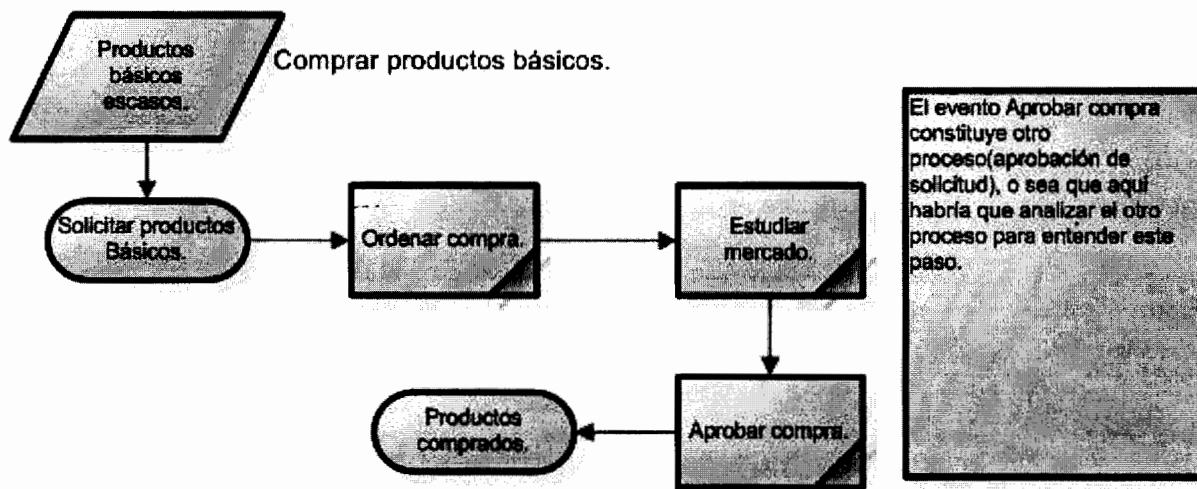


Figura 41: Diagrama de flujo del proceso comprar productos básicos.

Aplicando la metodología explicada aquí quedaría un caso de uso del negocio como sigue:

CU Ordenar Compra de Productos Básicos: Este proceso inicia cuando al Jefe de Compras del Almacén conoce de la existencia por debajo del límite de uno o más productos básicos y ordena la compra de los mismos. El especialista de compras hace estudio de mercado y finalmente se aprueba la compra en el Comité de Compras.

Posteriormente se obtiene el diagrama de actividades y el modelo de objetos del caso de uso en función:



Figura 42: Diagrama de actividades para el caso de uso ordenar compra de productos básicos.

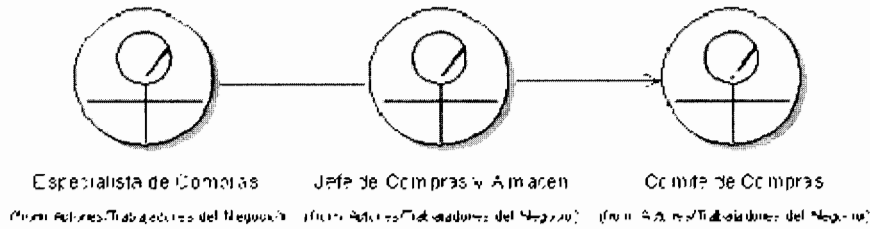


Figura 43: Modelo de objetos para el caso de uso ordenar compra de productos básicos.

Después de haber visto los ejemplos anteriores, ya es hora de llevar uno de estos casos de uso del negocio a casos de uso del sistema para ejemplificar este proceso. Ya esto depende del caso en específico que se desee informatizar. En este caso de uso que se está tratando, suponiendo que se deseara desarrollarlo, se podrían sacar los siguientes requisitos funcionales:

1. Seleccionar el o los producto(s) que debe(n) ser comprado(s) y su(s) cantidad(es).
2. Modificar la lista de los productos seleccionados hasta el momento.

Puede darse el caso que de este caso de uso se puedan obtener más requisitos de tipo funcionales, pero como esto se realiza en las fases iniciales del ciclo de desarrollo, no se quiere entrar en muchos detalles. El formato expandido de este caso de uso del sistema sería:

Caso de Uso		Ordenar Compra de Productos Básicos	
Actores		Jefe de Compras(inicia) y Almacén	
Resumen		Se da la posibilidad de seleccionar los productos básicos que hay que comprar, sus cantidades y de ordenar una compra.	
Propósito		Reaprovisionamiento del almacén.	
Referencias		R1, R2.	
Precondiciones		El actor escogió la opción de Ordenar Compra de Productos Básicos.	
Flujo Normal de Eventos			
Acción del Actor		Respuesta del Sistema	
		1. Muestra la pantalla dividida en dos paneles. En el izquierdo, una lista de los productos básicos, con sus datos (código de ASSETS, nombre, existencia, disponibilidad, límite mínimo). El derecho aparece vacío. Este panel está destinado a mostrar los productos que va seleccionando el actor, con sus cantidades editables en todo momento. (Al igual que en un carrito de compras se puede eliminar un producto o vaciar completamente).	
2. Selecciona un producto de la lista del panel izquierdo o, si ya su lista está completa, ordena la compra. Ir a paso 5.		3. Muestra el producto en el panel derecho con la cantidad en cero y editable.	
4. Edita la cantidad. Vuelve al paso 2.		5. Muestra un mensaje de confirmación.	
6. Confirma la orden (ver CA4). Termina el caso de uso.			
Cursos Alternos			
CA1 Selecciona la opción de eliminar un producto de la lista a solicitar.		Muestra la lista actualizada.	

CA2 Selecciona la opción de vaciar la lista a solicitar.	Muestra la lista actualizada (vacía).
CA3 El usuario cancela la operación antes de concluir el caso de uso.	Se retorna a la pantalla anterior sin guardar los cambios.
CA4 El actor no confirma la orden.	Se vuelve al paso 2.
Poscondiciones	Se ha ordenado una compra de productos básicos.

Tabla 4: Descripción de CU. Ordenar compra de PB

Para un mayor estudio de este caso de uso consultar el diagrama de Clases Web en el anexo1.

4.2.5 Entrega del mobiliario.

Los ingenieros industriales que laboran con los diagramas de flujo de procesos también han realizado diagramas a procesos individuales que se podrían adoptar posteriormente por algún desarrollo de software que lo contenga. Tal es el caso del proceso de entrega del mobiliario en la UCI. A continuación se muestra la misma metodología seguida anteriormente, comenzando por la descripción y el diagrama del proceso en función.

Lo primero que se realiza en para llevar a cabo este proceso es la solicitud de mobiliario por parte de los usuarios, posteriormente una persona técnica capacitada se encarga de verificar que el solicitante no haya recibido todavía su mobiliario. En caso de ya haberlo recibido, se informaría el resultado concluyendo así este proceso. En caso de no haberlo cogido, dicho especialista le entregaría al usuario su respectivo módulo, registrándose la entrega del mismo. Después de esto, el usuario procedería a firmar como que él recibió el mobiliario y se actualizaría el inventario del depósito, concluyéndose así el proceso de entrega del mobiliario.

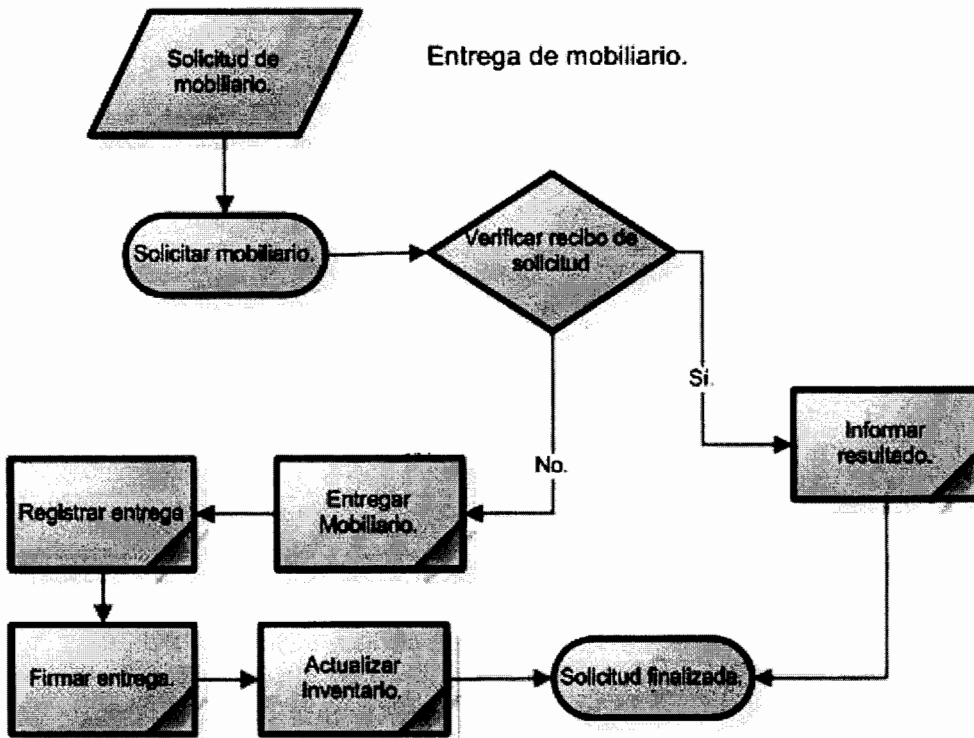


Figura 44: Diagrama de flujo para el proceso de entrega de mobiliario.

De lo anteriormente planteado se obtendría un caso de uso del negocio como sigue:

CU Entregar Mobiliario: Este proceso inicia cuando una persona solicita a un Trabajador de Depósito que le entregue el mobiliario que le corresponde. El Trabajador de Depósito verifica que la persona presente no ha recibido con anterioridad los productos durante la entrega en curso. En caso de que no haya sido recibido, le entrega los materiales establecidos para dicha entrega y registra la operación (en la Tarjeta de Control de Entrega de Mobiliarios). Posteriormente la persona firma en dicha tarjeta. Para finalizar el Trabajador actualiza el inventario del depósito.

El diagrama de actividades correspondiente a este caso de uso quedaría:

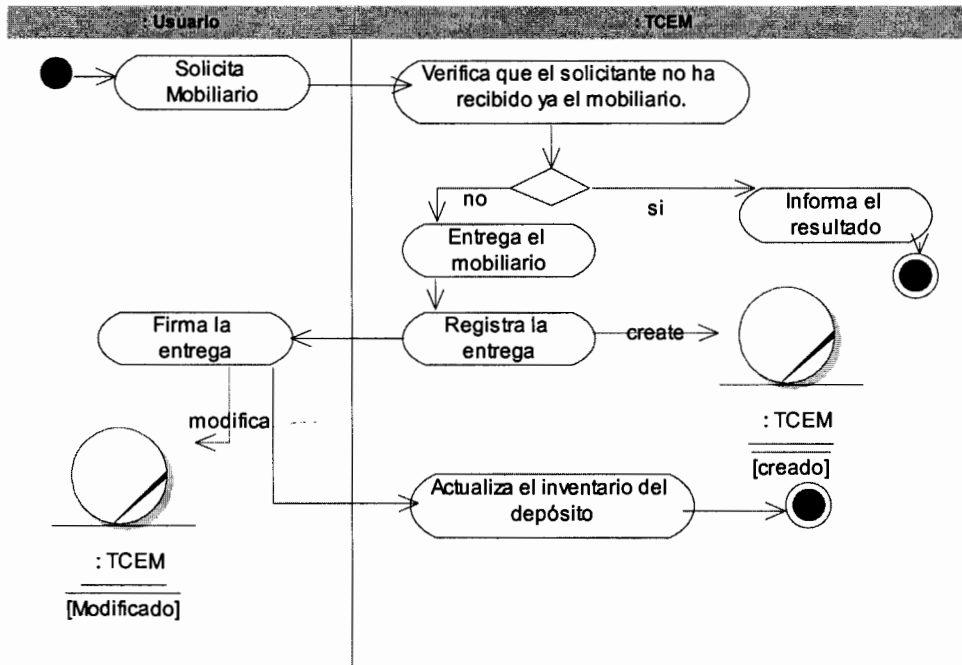


Figura 45: Diagrama de actividades para el caso de uso entregar mobiliario.

Además de esto cómo bien se ha visto anteriormente, se puede realizar el modelo de objetos de dicho caso de uso, el cual quedaría:

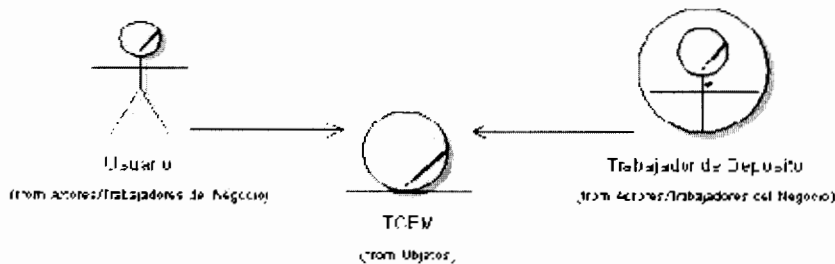


Figura 46: Modelo de objetos para el caso de uso entregar mobiliarios.

Para el desarrollo mediante RUP como metodología de trabajo de este caso de uso, se pueden extraer del mismo los siguientes requisitos funcionales:

1. Verificar que la persona no ha recibido aún el mobiliario que le corresponde.
2. Mostrar el listado de los productos que debe recibir.
3. Imprimir Vale de Constancia.

Este caso de uso suponiendo que fuera primario, o que de alguna forma se necesitara en su formato expandido, el mismo quedaría:

Caso de Uso	Entregar Mobiliario.	
Actores	Trabajador de Depósito	
Resumen	El caso de uso se inicia cuando el Trabajador de Depósito de mobiliarios decide entregar el módulo correspondiente a una persona. Se valida la entrega a la persona mediante el código de barras de su solapín y se muestran los materiales que debe recibir al tiempo que se descuenta del inventario. Se imprime un comprobante.	
Propósito	Registrar una entrega de mobiliario a una persona.	
Referencias	R1, R2, R3	
Precondiciones	Que el TD haya accedido a la opción de entregar materiales.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
	1. Muestra los controles necesarios para la introducción y confirmación del código de barras.	
2. Introduce y confirma el código de barras utilizando el lector de código de barras.	3.1 Verifica que la persona no haya recibido ya el mobiliario en la distribución actual. En caso de que algún dato enviado esté incorrecto ver CA2 y en caso de que ya lo haya recibido ver CA3. 3.2 Muestra los datos de la persona y los productos que debe recibir. Imprime un comprobante y registra la entrega así como actualiza el inventario del depósito. 3.3 Se regresa al paso 1.	
Cursos Alternos		
CA1: El usuario canceló la operación antes de concluir el caso de uso.	Se retorna a la pantalla anterior sin guardar los cambios.	
CA2: Los códigos de barras no coinciden o no son válidos.	Se muestra un mensaje de identidad no válida. Se retorna al paso 1.	
CA3: La persona ya ha recibido el mobiliario.	Se muestra un aviso y se retorna al paso 1.	
Poscondiciones	Se registró una entrega de mobiliario y se actualizó el inventario de un depósito.	

Tabla 5: Descripción de CU. Entregar Mobiliario.

Para un mayor estudio de este caso de uso consultar el diagrama de Clases Web en el anexo1.

4.3 Conclusiones.

En este capítulo se debatieron varios de los ejemplos de procesos que han sido analizados en la Universidad de las Ciencias Informáticas, los cuáles se han llevado a diagramas de UML siguiendo la metodología que aquí se define. Se puede observar con ellos que se han cumplido los objetivos que se perseguían puesto que se ha conseguido realizar los mismos diagramas que se realizaban bajo una metodología como las que existen actualmente, solo que ahora ha sido mucho más fácil este proceso para el informático.

Capítulo 5. Estudio de Factibilidad.

5.1 Introducción.

El presente capítulo expone los beneficios tangibles e intangibles del uso de esta metodología, además se llevarán dichos beneficios al área que más concierne, explicando como se puede beneficiar la Universidad de las Ciencias Informáticas con la adopción de los diagramas de flujo como un paso inicial y necesario a la hora de desarrollar un proyecto.

5.2 Beneficios Tangibles e Intangibles.

Se tienen casos claros de cómo algunas asunciones se desvanecen a la luz de una correcta experimentación y validación. Tal es el caso de la utilización de los diagramas de flujo para diseñar y documentar programas. En 1977 un estudio experimental realizado por B. Shneiderman demostró que los diagramas de flujo no ayudaban a entender mejor los programas que el pseudo código. Así, se dejaron de utilizar esos diagramas gráficos. Sin embargo, en 1989 D. Scanlan demostró exactamente lo contrario, y expuso varios fallos cometidos por B. Shneiderman en el diseño del experimento. A continuación se mostrarán algunos de los beneficios que traerán dichos diagramas para el desarrollo de software.

Al llevar el análisis teórico de los proyectos de desarrollo a la práctica es necesario concretar algunos conceptos. En este análisis concreto resulta de mucha utilidad el uso de diagramas de flujo, los cuales son una representación gráfica del proyecto. En esencia estos diagramas muestran el difícil equilibrio entre costos y beneficios, y en general permiten captar a las cinco variables a ser consideradas en toda evaluación de proyectos:

- I. Los costos, o inversión propiamente dicha, que suelen ser subdivididos en inversión inicial y costos operativos.
- II. Los beneficios o ingresos del proyecto, que igualmente se subdividen en ingresos operativos y valor de salvamento.
- III. El tiempo de duración de la inversión u horizonte temporal del proyecto.
- IV. La rentabilidad de la inversión, que resulta de una comparación de los ingresos frente a los costos.
- V. El riesgo asociado a la rentabilidad, toda vez que existe una indisoluble y directa relación entre rentabilidad y riesgo: a mayor rentabilidad esperada, mayor riesgo.

El gráfico siguiente muestra en detalle estos cinco elementos clave presentes en todo proyecto.

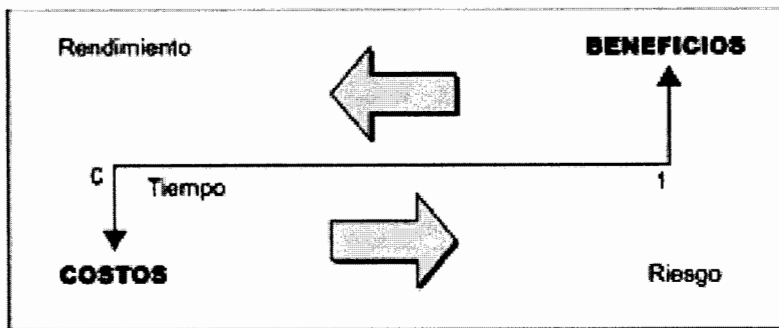


Figura 47: Elementos claves de cada proyecto.

El diagrama de flujo es un instrumento que, además, permite visualizar el efecto del financiamiento externo sobre la rentabilidad del proyecto, dándose aquí la posibilidad de ver las probabilidades reales del cambio. Son una herramienta valiosa para la mejora de los procesos, permiten detectar las actividades que agregan valor y aquellas que son redundantes o innecesarias, para la posterior informatización.

También son de gran utilidad durante el desarrollo de la documentación de los Sistemas de Gestión, pues proveen una descripción de los procesos y un detalle de las operaciones mucho más amigable que los procedimientos e instructivos basados en texto que se hacen siguiendo algunas metodologías como RUP. Contribuyen a resolver uno de los principales problemas, que es la resistencia del personal a emplear los documentos como referencias para el desempeño de las tareas. Una copia ampliada del diagrama de flujo al alcance de los desarrolladores de software facilita la consulta y promueve la creatividad. Por otra parte la adopción de esta metodología de trabajo favorece la comprensión del proceso ya que lo muestra como un dibujo. El cerebro humano reconoce fácilmente los dibujos. Un buen diagrama de flujo reemplaza varias páginas de texto. Aunque esta metodología de trabajo no tiene fines comerciales, su utilización reporta una reducción considerable del tiempo de desarrollo.

5.3 Factibilidad para el área de informatización.

Esta metodología traerá consigo una gran cantidad de mejoras significativas en la forma de funcionar y en la rapidez con las que se darán respuesta a las tareas que le conciernen al área de informatización. Entre estas facilidades se destacan que permitirá identificar los problemas y las oportunidades de mejora de los procesos en un menor plazo. Se identificarán los pasos redundantes, los flujos de los reprocesos, los conflictos de autoridad, las responsabilidades, los cuellos de botella, y los puntos de decisión necesarios a la hora de regirse por UML. Se mostrarán cada uno de los procesos que conciernen a un proyecto dado, facilitando a los desarrolladores el análisis de los mismos. Los diagramas de flujo que se implementarán

inicialmente serán una excelente herramienta para capacitar a los nuevos empleados y también a los que desarrollan la tarea cuando se realizan mejoras en los procesos iterativos.

Por otra parte como el área que cubre el departamento de informatización es tan abarcadora, las personas que no están directamente involucradas en los procesos de realización del software, tienen imágenes idealizadas de los mismos, que pocas veces coinciden con la realidad. De aquí que la utilización de los diagrama de flujo de procesos será una actividad que agregará valor, pues el proceso que representa está ahora disponible para ser analizado, no sólo por quienes lo llevan a cabo, sino también por todas las partes interesadas que aportarán nuevas ideas para cambiarlo y mejorarlo.

Además de todo lo antes expuesto, como se vio en los capítulos anteriores, es evidente que después de comprender la forma de conversión de los diagramas de flujo procesos en diagramas UML, ya no quedará problema alguno para la realización de la ingeniería de software pues no solo se podrá entrevistar a los clientes y usuarios en caso de algo que no quede claro, sino que ahora también se tendrá una forma más elocuente de enseñarles lo que se está haciendo.

Igualmente se verán otras facilidades tales como las grandes mejorías que se darán en cuanto al tiempo de desarrollo del producto, puesto que como bien se abordó anteriormente, al trabajar con las definiciones y diagramas de flujo de procesos realizados por los ingenieros industriales ya se está haciendo realidad que se ganará en rapidez puesto que es mucho más fácil comprender las áreas a informatizar mediante los diagramas de flujo que por plena voz de los clientes, además, en numerosas ocasiones no se logra extraer toda la información que los clientes pueden dar. Asimismo se ganará en eficiencia y calidad del software en construcción ya que los diagramas de procesos que desarrollan los ingenieros industriales cumplen con los estándares de diseño de diagramas recogidas en las normas ISO vigentes. También es mucho más factible medir la calidad en cuanto a diagramas que mediante documentos. Aquí se puede incorporar el hecho de que debido a que las tareas iniciales serán llevadas a cabo por un personal que se encontrará ampliamente capacitado para esta tarea, y que no tendrá más ninguna actividad colateral que realizar, se podrá decir que será un experto en este sentido, lo cuál además le posibilitará al grupo de trabajo tener una mayor cantidad de personal disponible para otras tareas, al mismo tiempo que puede ayudar a los desarrolladores en la documentación, diseño y transmisión de sistemas complejos de una manera clara para facilitar la colaboración y una toma de acciones más efectiva.

También se puede decir que se requiere tiempo y, por tanto dinero para seguir un proceso de desarrollo y realizar el análisis y diseño orientado a objetos. A continuación se exponen las razones que permitirán agilizar este proceso mediante esta metodología:

1. Como indican las estadísticas existentes, los proyectos son una actividad riesgosa, y de ahí que el motivo principal sea aminorar el riesgo, esto significa aumentar las posibilidades de desarrollar un sistema eficiente. Una estrategia para conseguirlo consiste en examinar los requerimientos y elaborar un diseño formal, y mediante esta metodología se puede agilizar esta tarea.
2. Conviene crear un proceso que puedan reproducir los individuos, y sobre todo los equipos. Por ende esta metodología sería ideal bajo esta premisa.
3. Es más barato y fácil efectuar cambios durante las etapas de análisis y diseño que en la fase de construcción, y con esta metodología se podrían analizar mejor los sistemas para evitar cambios posteriores.
4. Mediante esta metodología se puede atenuar y manejar la abrumadora complejidad con solo modelar los sistemas y hacer abstracciones para detectar los detalles esenciales.
5. En el proceso de desarrollo del software conviene crear sistemas robustos, susceptibles de mantenimiento y capaces de soportar una mayor reutilización del software. Y como se ve en las características esenciales de los diagramas de flujo, estos se caracterizan por lograr trascender en el tiempo, a no ser que sea necesaria una reingeniería de procesos.

Puede ser que haya desarrolladores que se resistan a la idea del cambio, pero no hay más que ver las estadísticas, las cuáles no mienten:

Con frecuencia se oyen verdaderas historias de terror acerca de los fracasos de proyectos de software que han costado millones de dólares: sistemas fiscales, sistemas de defensa, entre otros. La conclusión salta a la vista: el desarrollo de software es un negocio riesgoso. Un estudio reveló que el 31% de los proyectos no se concluyen, el 53% de ellos rebasa casi el 200% del costo estimado, que en 1995 las compañías estadounidenses y las dependencias gubernamentales invirtieron 81 millones de dólares en proyectos de software que fueron cancelados. Con solo escuchar cosas como estas se puede verificar que no se están obteniendo los resultados deseados.

Es por todo lo anteriormente planteado en este documento que se considera deben experimentarse otras áreas para buscar la forma de lograr una gran mejoría, más aún en la UCI, centro que tiene entre sus objetivos ser vanguardia en la industria del software.

5.4 Conclusiones.

En este capítulo se efectuó el estudio de factibilidad correspondiente al desarrollo de los proyectos de informatización de la universidad siguiendo esta metodología, así como todos los aspectos que se verán beneficiados por su adopción. El mismo permitió llegar a la conclusión que resultará factible esta

metodología, ya que aunque existe cierto costo total debido fundamentalmente al desconocimiento que pueda haber hacia este tema, y a que muchos desarrolladores se resistan al cambio, los beneficios que se alcanzarán son considerables.

Conclusiones Generales.

A lo largo de este trabajo se evidenció una nueva variante a incluir al proceso de informatización. Mediante ejemplos abordados en la misma se puede concluir que si se utiliza debidamente como una herramienta de ayuda más se podrán llevar a cabo mejores proyectos sin temor a no culminarlos ya que esta metodología de trabajo ayuda entre otras cosas a prevenir la envergadura del proyecto en función, además de que conlleva a una correcta comprensión del negocio mucho antes de realizar las entrevistas al personal capacitado.

Con las investigaciones realizadas que culminan con la explicación cabal de la metodología para unificar los diagramas de flujo de procesos al desarrollo del software se da cumplimiento a los objetivos de este trabajo, pues se obtuvieron una serie de pasos a seguir para el desarrollo de un mejor software. Los logros más relevantes se mencionan a continuación:

1. Los diagramas de flujo de procesos pueden ayudar a aliviar en una medida considerable el difícil trabajo de captura de requisitos debido a deficiencias en la forma de controlar las entrevistas a los clientes.
2. Mediante el trabajo sobre los diagramas de flujo, en vez de hacerlo directamente sobre el funcionamiento del negocio como tal, se logran desarrollar productos de un mayor ciclo de vida, ya que dichos diagramas se encontrarán automatizados.
3. Con la unificación del trabajo realizado por parte de los ingenieros industriales y los desarrolladores de software se lograría una mejor comprensión por parte de estos últimos de la lógica de funcionamiento del negocio en función.
4. La entidad que emplee esta metodología de trabajo percibirá grandes mejoras en la velocidad y en la facilidad de trabajo.
5. Al utilizar esta metodología, se puede lograr involucrar de forma más viable lo mismo a los clientes, que a nuevos desarrolladores que se hayan acabado de incorporar al desarrollo del software, ya que los diagramas de flujo son bastante explicativos sobre los procesos de negocio existentes.

Por todo lo anteriormente planteado, se espera que esta metodología de trabajo sea utilizada solo con fines de ayuda al proceso de desarrollo del software, y no como un recurso por separado para el desarrollo de aplicaciones.

Recomendaciones.

Tomando como base las investigaciones realizadas y la experiencia acumulada durante la realización de este trabajo, se proponen las siguientes recomendaciones:

1. Integrar esta metodología de trabajo a todas las demás áreas de la universidad, o al menos hacerlo paulatinamente a medida que se vaya probando su nivel de ayuda.
2. Capacitar a los ingenieros industriales que laboran en la tarea de realizar los diagramas de flujo de procesos, para que sean capaces de entregarle a los ingenieros industriales, en vez de diagramas de flujo de procesos, los casos de uso del negocio, con su respectivo diagrama de actividades y el modelo de objetos.
3. Utilizar otra herramienta más específica para la tarea de realizar los diagramas de flujo de procesos, posiblemente una de libre acceso como las abordadas anteriormente.

Bibliografía.

1. [BER04] Bernier Villamor, José Luis "Fundamentos de la programación 1" Marzo, 2004.
2. [Beck94] Beck, K. 1994. Patterns and Software Development. Dr. Dobbs Journal. Febrero 1994.
3. [BJR97] Booch, G., Jacobson, I. y Rumbaugh, J. 1997. The UML specification documents. Santa Clara, CA: Rational Software Corp.
4. [Booch94] Booch, G., 1994. Object Oriented Analysis and Design. Redwood City CA: Benjamin/Cummings.
5. [Coad95] Coad, P. 1995. Object Models: Strategies Patterns and Applications. Englewood Cliffs, NJ.: Prentice-Hall.
6. [GHJV95] Gamma, E., Helm, R., Johnson, R., Vlissides, J. 1995. Design Patterns. Reading, MA: Addison-Wesley.
7. [Zhao03]Zhao, Jie.2003. Comparación de Herramientas de modelado UML. Septiembre, 2003.
8. [Horn00]Horn, H.2000. Diagramas de flujo.Abril, 2000.
< <http://www.calidad.org/s/flujo.pdf>>
9. [Larman04] Larman, Craig 2004. UML y Patrones. Introducción al análisis y diseño orientado a objetos. Editorial Félix Varela. La Habana, 2004.
10. [JBR04]Jacobson, Ivar; Booch, Grady y Rumbaugh, James. El Proceso Unificado de Desarrollo de Software. Editorial Félix Varela, La Habana, 2004.
11. [Zaratiegui99]Zaratiegui, J. R. 1999. La Gestión por procesos: Su papel e importancia.
12. Hernández Calderín, Ernesto E. ¿Cómo elaborar un diseño de investigación en Gráfica de Ingeniería? Trabajo presentado en el evento GRAFIAR 2000.
13. [BRJ00]Booch, Grady, Rumbaugh, James, Jacobson, Ivar. El Lenguaje Unificado de Modelado. 2000. Addison Wesley.

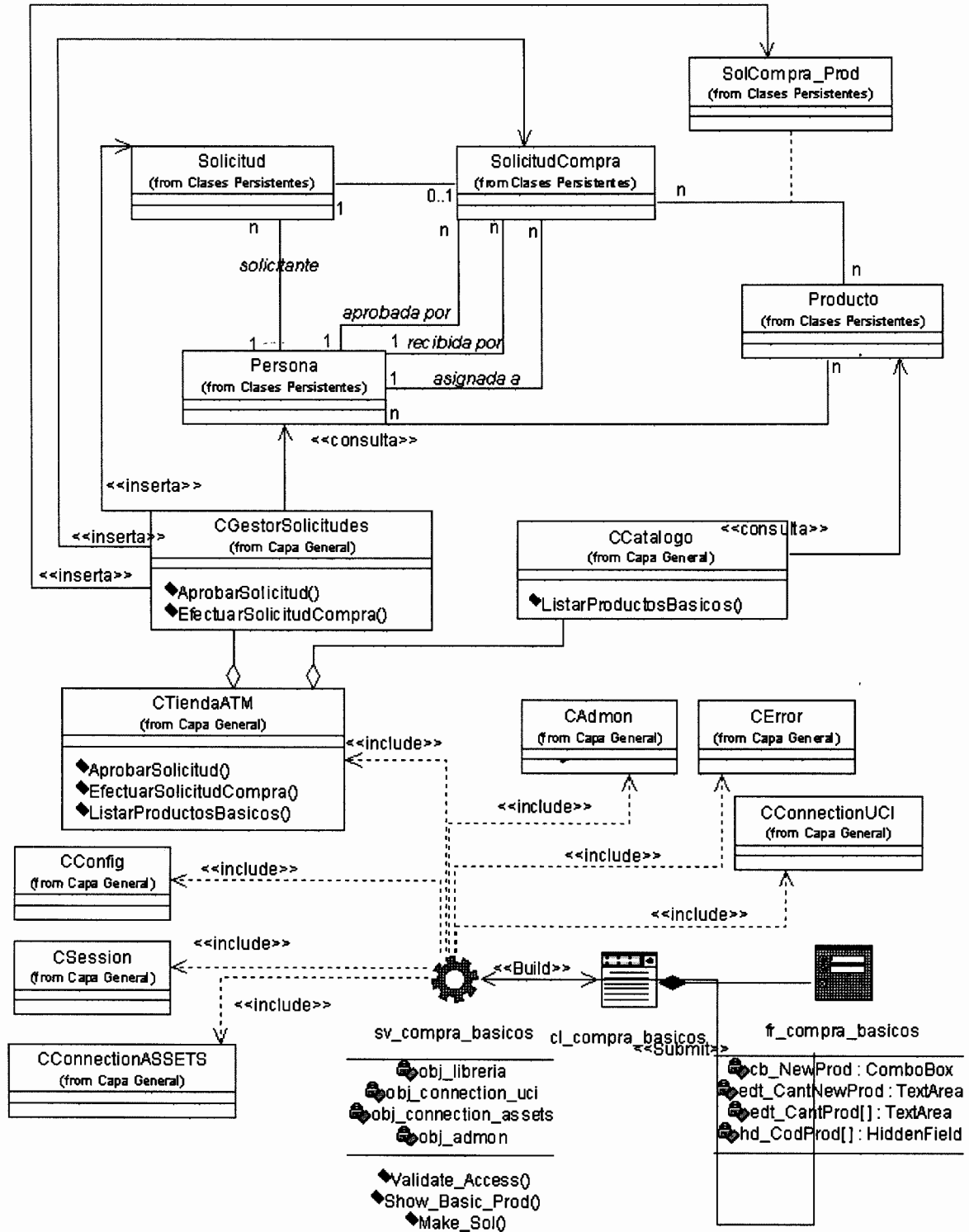
14. [Pressman00]Pressman, Roger. Ingeniería de Software. Un enfoque práctico. 2000. McGraw-Hill/Interamericana de España.
15. [Encarta04]Biblioteca de Consulta Microsoft ® Encarta ® 2004. Flujo de Procesos.
16. [Schmuller99]Schmuller, Joseph. Aprendiendo UML en 24 horas. 1999.
17. Sitio Web de Modelado de sistemas con UML.
<<http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiple-html/index.html>> (22/11/05)
18. Sitio Web de Concept Draw.
<http://www.conceptdraw.com/sp/products/cd5/ap_flowchart.php> (13/11/05)
19. Sitio Web de ArchivosPC.com.
<<http://www.archivospc.com/programas/categorias/Diagramas%20de%20Flujo.php>> (19/11/05)
20. Sitio Web de Microsoft. <<http://www.microsoft.com/latam/office/prodinfo.mspx>> (10/11/05)
21. Sitio Web de Microsoft Office System Product Information.
<<http://www.microsoft.com/latam/office/visio/prodinfo/default.mspx>> (13/11/05)
22. Sitio Web Adictos al Trabajo.
<<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=umlintro>> (22/11/05)
23. Sitio Web de Comparativa de Herramientas UML de Libre Distribución.
<<http://www.diatel.upm.es/malvarez/UML/Comparativa.html>> (28/11/05)
24. Sitio Web de Fundamentos de Rational Rose.

<<http://www.rational.com.ar/cursos/fundrose.html>> (10/11/05)
25. Sitio Web de productos IBM Racional Rose.
<<http://www.rational.com.ar/herramientas/rose-xde.html>> (24/11/05)
26. Sitio Web de Remis. <<http://www.sc.ehu.es/jiwdocoj/remis/remis.htm>> (23/11/05)
27. Sitio Web de Herramientas Organizacionales.
<<http://www.estrucplan.com.ar/Producciones/Entrega.asp?identrega=526>> (20/11/05)
28. Sitio Web de Diagramas de Flujo.
<<http://www.centrosdeexcelencia.com/entidades/herram/flujo.htm>> (10/11/05)
29. Sitio Web Diagramación. <http://html.rincondelvago.com/diagramas-de-flujo_1.html> (22/06/05)

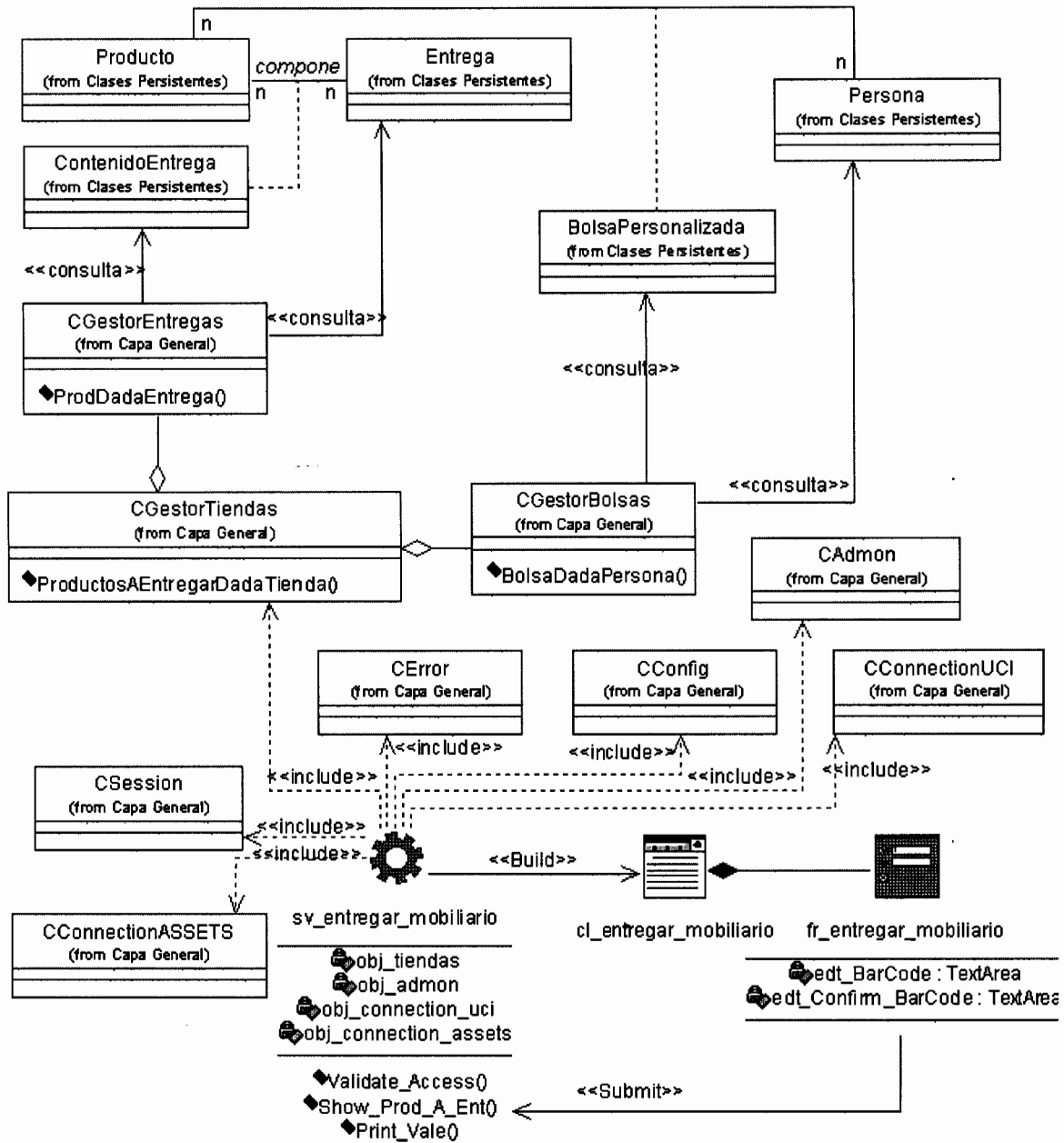
A continuación se muestra, en orden alfabético, el significado de algunos términos usados en este documento cuyo uso no es común y que pueden dificultar la comprensión del mismo:

1. **Ad-hoc:** Se refiere a algo destinado a un uso y situación concretos.
2. **CASE:** Computer Aided Software Engineering. Las herramienta CASE están dirigidas a los programadores. Proporcionan automatización en fases del desarrollo de programas o facilidades de documentación y comprensión.
3. **Cliparts:** Se refiere a imágenes prediseñadas.
4. **Diagramas de flujo:** Se basan en la utilización de diversos símbolos para representar operaciones específicas. Se les llama diagramas de flujo porque los símbolos utilizados se conectan por medio de flechas para indicar la secuencia de operación.
5. **GDI:** Graphics Device Interface. Es el responsable en el Sistema Operativo Windows de la representación gráfica, tanto dibujos, como textos, como en impresiones.
6. **Grasp:** Grupo de patrones de diseño de asignación de responsabilidades.
7. **Gof:** Del inglés pandilla de los cuatro.
8. **IDE:** Integrated Device Electronic también conocido como interface de bus AT y fue la primera que se utilizó para discos duros en microordenadores y tiene capacidad para 2 unidades de almacenamiento, con un máximo en discos duros de 528 megas. Fue originario de la empresa Conner y evoluciona en la norma comúnmente utilizado en la actualidad, que es EIDE.
9. **ISO:** International Organization for Standardization. Fundada en 1946, es una federación internacional que unifica normas en unos cien países. Una de ellas es la norma OSI, modelo de referencia universal para protocolos de comunicación.
10. **JDK:** Java Development Kit (JDK(TM)) es un software distribuido por Sun Microsystems, Inc. para los desarrolladores de Java.
11. **ODBC:** Open Data Base Connectivity. Ha sido la base de Windows en sistemas abiertos, es decir que permiten una conectividad entre distintos lenguajes de programación con distintas bases de datos.

12. **OLE:** Object Linking and Embedding. Es un concepto complejo del entorno de programación en Windows y que abarca campos muy amplios. Podría decirse que son conjuntos de librerías, que permiten crear conexiones entre aplicaciones distintas, como dirigirse desde un programa determinado a una hoja de cálculo o un procesador de textos. Pero así expresado es muy rudimentario, porque también permite la interacción entre el Sistema Operativo y otros programas, documentos compuestos que alberguen formatos muy distintos desde utilidades y aplicaciones muy variadas.
13. **OMG:** Debido a sus siglas en inglés Object Management Group.
14. **RUP:** Rational Unified Process, es una metodología para el análisis, diseño y desarrollo de sistemas.
15. **Six Sigma:** Metodología normal de la industria que las compañías utilizan para señalar los problemas y mejorar la eficiencia.
16. **SVG:** Del inglés Scalable Vector Graphic.
17. **Tarjeta de Control de Entrega de Mobiliarios (TCEM):** Tarjeta donde se registran las entregas de mobiliarios para cada persona.
18. **Tarjeta de Estiba (TE):** Tarjeta que contiene la existencia de un producto. Esta existencia se incrementa cuando se recibe una cantidad de dicho producto y se decrementa cuando se extrae una cantidad del almacén.
19. **Top-down:** Del inglés de arriba hacia abajo.
20. **UML:** Se le denomina así al Lenguaje Unificado de Modelado, basados en los primeros métodos de Programación Orientada a Objetos (POO) y está pensado para realizar análisis completos para desarrollo de aplicaciones de unas dimensiones amplias.
21. **Vale de Entrega o Devolución (VED):** Se emite al sacar los productos del almacén o al devolverlos a este, se le pasa copia a Economía.
22. **Vale de Solicitud de Materiales (VSM):** Se emite para solicitar productos del almacén.



Anexo 1: Diagrama de Clases Web. CU Ordenar Compra de Productos Básicos.



Anexo 2: Diagrama de Clases Web. CU Entregar Mobiliario.