



03053 I
7

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

**APORTACIONES AL PROCESO UNIFICADO
MEDIANTE EL ANÁLISIS DEL USUARIO**

T E S I S

QUE PARA OBTENER EL GRADO DE

MAESTRO EN CIENCIAS *de*

Ingeniería de la computación.

P R E S E N T A:

LIC. ALEJANDRO LÓPEZ KOLKOVSKY

DIRECTOR: DR. FERNANDO GAMBOA RODRÍGUEZ

MÉXICO, D. F.

MARZO 2003

TESIS CON
FALLA DE ORIGEN



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Autorizo a la Dirección General de Bibliotecas de la UNAM a difundir en formato electrónico e impreso el contenido de mi trabajo recepcional.

NOMBRE: ALEJANDRO
LOPEZ KOLKOLEKY

FECHA: 18 MARZO 03

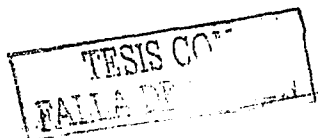
FIRMA: ALopez

Índice general

1. Introducción	1
1.1. La Ingeniería de Software y el Proceso Unificado	3
1.1.1. El Proceso Unificado	4
1.1.2. El flujo de trabajo de captura de requisitos del PU	6
1.1.3. El problema de la captura de requisitos	10
1.2. El Diseño centrado en el usuario y el Análisis de la Tarea	11
1.2.1. El Análisis de la Tarea	13
1.2.2. El Análisis Jerárquico de la Tarea	14
1.3. Enfoque del trabajo	17
2. El formalismo MAD STAR	19
2.1. El Análisis de la Tarea empleando MAD STAR	19
2.1.1. Actividad 1: Elaboración del árbol de la tarea del usuario	20
2.1.2. Actividad 2: Elaboración de las fichas de descripción de las tareas	20
2.1.3. Actividad 3: Poner las condiciones de las fichas en términos de notación orientada a objetos	23
2.2. Validación de los productos del análisis de la tarea	24
2.3. Ejemplo de un Análisis de la Tarea con MAD*	25
3. Encontrando el puente entre el AT y el PU	39
3.1. El análisis de la tarea iterativo e incremental	39
3.2. La relación entre el Árbol de la Tarea y los Casos de Uso	41
3.3. Paso del Árbol de la Tarea a Casos de Uso	45
3.3.1. El árbol de la tarea modificada	46
3.3.2. Casos de uso preliminares	47
3.4. Resumen	49
4. Integración del Análisis de la Tarea del Usuario al Proceso Unificado	51
4.1. Descripción rápida	51
4.2. Nuevos artefactos	52
4.2.1. Modelo MAD* de la tarea del usuario	52
4.2.2. Modelo MAD* de la tarea modificada	55
4.2.3. Modelo de casos de uso preliminares	55
4.2.4. Documentación de las entrevistas	55
4.3. Nuevo trabajador: Analista de la tarea	56
4.4. Nuevas actividades	57
4.4.1. Realizar análisis de la tarea del usuario	57

TESIS CON
FALLA DE ORIGEN

4.4.2.	Obtener árbol de la tarea modificada	59
4.4.3.	Elaborar diagramas de casos de uso preliminares	60
4.5.	Actividades del PU que se apoyan con la integración	62
4.5.1.	Encontrar actores y casos de uso	62
4.5.2.	Detallar un caso de uso	63
4.5.3.	Prototipar la interfaz del usuario	63
4.6.	Resumen	64
5.	Desarrollo de un caso de estudio	65
5.1.	Objetivo	65
5.2.	Planteamiento del problema	66
5.3.	Desarrollo del proceso	66
5.3.1.	Análisis de la tarea del usuario	68
5.3.2.	Requisitos candidatos	68
5.3.3.	Glosario de términos	70
5.3.4.	Obtener árbol de la tarea modificada	73
5.3.5.	Mapco de la tarea a casos de uso preliminares alrededor de objetivos fundamentales	76
5.3.6.	Mapco de la tarea a casos de uso preliminares alrededor de objetos de la tarea	78
5.3.7.	Mapco de la tarea a casos de uso preliminares alrededor de la tarea como la realiza el usuario	80
5.3.8.	Mapco de la tarea a casos de uso preliminares alrededor de acciones ejecutadas en la tarea	82
5.4.	Discusión	84
6.	Conclusiones	87
6.1.	El uso de teorías que fundamenten las metodologías	87
6.2.	La aportación al Proceso Unificado	88
6.3.	Resultados en el primer caso de estudio	89
6.4.	Los siguientes pasos	89
A.	Fichas del árbol de la tarea del usuario	91
B.	Fichas con objetos del árbol de la tarea del usuario	107



Capítulo 1

Introducción

En las últimas décadas, la industria de la computación ha tenido una tasa de crecimiento impresionante. Actualmente, las computadoras se pueden encontrar en casi cualquier lugar, comenzando desde los más comunes como televisiones y agendas electrónicas, hasta llegar a situaciones de alto riesgo como estaciones eléctricas y plantas nucleares. Muchas de las actividades que se realizan en los ambientes escolares y de oficina se basan en que, con la ayuda de una computadora, se podrán hacer tareas de forma más rápida, precisa y segura. Como consecuencia de esta proliferación, el software con el cual las computadoras son controladas es muy versátil y casi imposible de clasificar.

Esta variedad y versatilidad del software ha hecho que el desarrollo de sistemas sea una actividad compleja, en donde se requiere un amplio conocimiento del dominio en donde éste operará, de las tecnologías involucradas, del software existente en el mercado, de técnicas de gestión del proyecto, etc. Todas estas variables permiten explicar porqué, en la actualidad todavía se tiene un gran nivel de incertidumbre al empezar un proyecto de software; y en particular, se tiene poca seguridad sobre el tiempo y recursos que necesitará, si cumplirá con todos los requerimientos del cliente, si éste podrá adaptarse a otros sistemas, si será robusto, etc.

Cuando se presupuesta el costo y tiempo de un proyecto de software se emplea la experiencia sobre los recursos que se usaron en proyectos similares, esperando que los problemas que se presenten puedan resolverse dentro del tiempo y recursos calculados. Sin embargo, es común que algunos de ellos requieran de significativamente más trabajo, y que el costo y tiempo de desarrollo sobrepase por mucho lo presupuestado. De forma similar, se calcula el trabajo que se necesitará para implementar cada función al sistema, pero durante el desarrollo es posible encontrar algunas que requieren de significativamente más esfuerzo e inclusive, que no se llegan a implementar. Por si esto fuera poco, cuando se termina el proyecto siempre se tiene la inseguridad de si le agrada al cliente y de si satisfará las necesidades de su negocio.

Actualmente el sentimiento de los desarrolladores es, como lo indica Boris Beizer [6], que el software es un producto diferente a cualquier otro en la vida cotidiana, pues es difícil pronosticar su complejidad, el número de líneas que requerirá para completarse, los defectos o carencias que tendrá, el tiempo que funcionará sin requerir de servicio, etc.; y aún acabado el desarrollo, no es simple comprobar que éste funcione adecuadamente o cumpla con los requerimientos.

Los problemas en el desarrollo van por supuesto ligados a un aumento del costo del nuevo sistema, pues implican presupuestos entre 50 % y 100 % más altos, en donde los pro-

TESIS CON
FALLA DE ORIGEN

gramadores se protegen contra los posibles imprevistos y problemas que puedan encontrar; y aún así, no es poco común que la entrega del producto se exceda en tiempo y en costo, no cumpliendo con las expectativas del cliente.

Esto se pone en contraste con que el cliente espera costos proporcionales a los beneficios que obtendrá del nuevo producto, y un ciclo de vida similar al de cualquier otro producto, como por ejemplo un automóvil, en donde a priori se sabe qué precio tendrá de acuerdo a su equipamiento, su tiempo de entrega, el mantenimiento que requerirá y el momento en que será oportuno reemplazarlo.

Es por eso que los problemas relacionados al desarrollo de software han sido atacados por las diversas comunidades académicas y del mundo de desarrollo, quienes buscan técnicas y teorías que ayuden a convertir el desarrollo en una actividad predecible y controlable.

En particular, de todos los problemas involucrados con el desarrollo de software, dos de gran importancia son:

- *Calidad en el desarrollo del software:* Los problemas de desarrollo antes mencionados llevan a preguntarse qué métodos y prácticas deben de emplearse para alcanzar los requerimientos y objetivos establecidos tanto para el proyecto de desarrollo como para los productos obtenidos, de tal forma que la construcción de sistemas sea una actividad exitosa repetible: Cuando un proyecto termina exitosamente, no se tiene mucha certeza sobre el resultado que se tendrá en la realización de otro similar.
- *Usabilidad del software:* Las computadoras son herramientas, y por tanto su diseño afecta la forma en que son usadas. Dado que el software es generalmente la *interfaz* a la computadora, surge la duda de cómo diseñarlo de tal forma que éste aproveche todas las habilidades de quien lo usa, tomando en cuenta sus limitaciones y preferencias. En la actualidad es frecuente que cada vez que se reemplaza un sistema existente, o se libera un nuevo producto, los usuarios tienen que pasar por semanas de adaptación y capacitación, disminuyendo el desempeño de los usuarios e implicando costos adicionales al producto.

Los dos problemas anteriores se resuelven mediante la aplicación de teorías provenientes de distintas disciplinas, por lo que las soluciones que se han propuesto son parciales. Sin embargo, como el desarrollo de un sistema es una actividad única, es necesario tener una metodología que resuelva ambos problemas.

La comunidad de *Ingeniería de Software* se ha dedicado a estudiar el problema de los procesos de desarrollos controlados y de los productos con calidad. Sus propuestas de procesos para desarrollo de software contemplan la mayoría de las actividades involucradas en él, especificando qué características debe cumplir cada actividad y no tanto cómo debe de realizarse. Como consecuencia, el que se tome en cuenta las características del usuario y su ambiente actual depende más del desarrollador que del proceso. En efecto, aunque los ingenieros de software han tratado de integrar las características, intereses y objetivos del usuario a sus procesos mediante la especificación de actividades más concretas, no han podido dar una solución al problema de la usabilidad del software, ya que no cuentan con los elementos necesarios, de manera clara, en su disciplina (ver por ejemplo Insfran [15] o Sánchez [16]).

Por otro lado, la gente de *Interacción Humano-Computadora* ha propuesto varios enfoques para realizar un desarrollo *Centrado en el usuario*, el cual toma en cuenta las habilidades, limitaciones y preferencias de quien lo usa, pero éstos han fracasado debido a que no se han integrado a un proceso de desarrollo integral que contemple todas las actividades

y aceptado por la comunidad de Ingeniería de Software. Así, diversos intentos que se han hecho van en la dirección de establecer un proceso de desarrollo a partir de las técnicas de diseño de interfaces humano-computadora, lo cual no ha dado muchos frutos debido a que:

1. Los desarrolladores no están familiarizados con el diseño centrado en el usuario y emigrar a estas metodologías constituye un gran cambio.
2. Las actividades básicas del diseño centrado en el usuario, tales como el *Análisis de la Tarea del Usuario*, o las pruebas con usuarios, llevan un tiempo considerable y requieren de gente calificada.
3. Las propuestas no son completamente compatibles con los enfoques orientados a objetos que actualmente se usan en la industria de software.
4. Se ha logrado efectuar ejercicios académicos, pero no se tienen proyectos reales del tamaño suficiente como para comprobar los beneficios de las propuestas.

El enfoque que se propone en este trabajo es tomar al *Proceso Unificado* - la metodología de ingeniería de software más aceptada por los desarrolladores y en general por los académicos - e integrar en ella al análisis del usuario y su tarea, uno de los elementos formales en el diseño centrado en el usuario. El reto será efectuar dicha integración cuidando que el enfoque y metodología del Proceso Unificado no cambie en su esencia y filosofía, y que los productos de las nuevas actividades se puedan especificar en términos del lenguaje *UML*. Creemos que esto podría tener éxito ya que los desarrolladores familiarizados con el Proceso Unificado y *UML* emplearán básicamente el mismo proceso y la misma notación, y no tendrán que aprender o familiarizarse a nuevas formas de trabajo. Como resultado, se tendrá un proceso de desarrollo de probada eficiencia, más completo, y que permita crear sistemas centrados en el usuario.

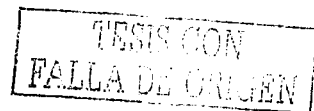
En lo que resta de este capítulo se analizarán en detalle los puntos relevantes del Proceso Unificado y del Análisis de la Tarea, mostrando sus inconvenientes y resaltando aquellas características que nos llevan a pensar en la viabilidad de una integración.

1.1. La Ingeniería de Software y el Proceso Unificado

La *Ingeniería de Software* (IS) estudia el proceso de desarrollo de software —también llamado proceso de software— desde varios aspectos, con el objetivo de establecer técnicas que permitan efectuar dicho proceso de forma controlada y obteniendo resultados predecibles. La experiencia que se ha adquirido en los últimos veinte años muestra que hay muchos factores en un proyecto que pueden ocasionar retrasos, costos elevados, productos defectuosos e inclusive el fracaso total.

El objetivo central de la IS es asegurar la calidad del software. La noción de calidad es algo distinta a la del sentido común, ya consiste en satisfacer los requerimientos del cliente (como lo son funcionalidad y robustez de los productos), además de satisfacer las expectativas de tiempo y costo tanto del cliente como del desarrollador (ver [29]).

Para lograr esto, la Ingeniería de Software propone aplicar distintas metodologías en el proceso de software, de modo a controlar aquellos aspectos que tienen un impacto en la posibilidad de terminar el proceso de forma exitosa y predecible, y que permitan obtener productos que cumplan con las expectativas de los desarrolladores y de los clientes. Entre estos aspectos podemos encontrar la planeación, la administración, las pruebas, la verificación, la validación y, lo que es de nuestro interés, el desarrollo.



Hay una gran variedad de procesos de software que se han propuesto en la IS: están desde los más generales, que indican cómo desarrollar un proyecto de software, hasta los más particulares que proponen cómo tratar un aspecto en específico; también podemos encontrar que mientras algunas metodologías indican solamente las características que debe de cumplir el proceso, hay otras que establecen puntualmente cómo efectuar cada actividad del proceso de software.

1.1.1. El Proceso Unificado

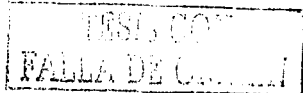
Con los impresionantes avances tecnológicos de hoy en día, podemos encontrar aplicaciones que realizan tareas muy complejas en poco tiempo, y cuya complejidad implica que su desarrollo sea una actividad que requiera de la participación de muchas personas con distintos roles. El riesgo consecuente es que una mala organización en el grupo de trabajo, o la falta de metodologías para el desarrollo, pueden implicar tiempos y costos excesivos en el desarrollo, productos con un gran número de errores, productos que no cubren las necesidades planteadas por el cliente, etc. En la Ingeniería de Software hay varios esquemas y metodologías de desarrollo que han sido probados en la industria y mejorados a lo largo del tiempo. El que se presenta a continuación, fue propuesto recientemente (1998) y ha tenido éxito debido a que integra la experiencia en el desarrollo que se tiene en IS, es de fácil seguimiento y comprensible por los desarrolladores, emplea una notación estándar, y propone un esquema que permite integrar el proceso a desarrollos orientados a objetos.

Esta metodología es la del *Proceso Unificado* [3] (PU). En ella, el desarrollo es visto como una serie de ciclos similares de trabajo, que permiten administrar el proceso dividiendo el problema y estableciendo metas para cada ciclo. De esta forma, las etapas que se consideran generalmente en un proceso de desarrollo de software (inicio, elaboración, construcción, transición) están ahora conformadas por ciclos con distintos objetivos principales.

Cada ciclo tiene cinco flujos de trabajo fundamentales:

- Captura de requisitos
- Análisis
- Diseño
- Implementación
- Prueba

Luego, dependiendo de los objetivos del ciclo se dedicará distinta cantidad de trabajo a cada flujo. Cuando se está en un ciclo de la etapa de inicio, el objetivo es entender el dominio del problema y los requerimientos, por lo que se hace principalmente "Captura de requisitos", y un poco de "Análisis" (ver figura 1.1). En la etapa de elaboración, en los primeros ciclos se hace principalmente "Análisis" junto con algo de "Captura de requisitos" y "Diseño"; luego primordialmente "Diseño" con algo de "Captura de requisitos" y "Análisis"; y en los últimos ciclos de la elaboración se hace "Diseño" y algo de "Implementación". En la etapa de construcción el flujo de trabajo principal es el de "Implementación", aunque debemos notar que también se efectúan algunos otros debido a que constantemente se modifican algunos requisitos y se hace análisis y diseño. También es importante observar que durante todo el ciclo de desarrollo se efectúan pruebas, debido a que hay que validar los distintos productos del desarrollo (requisitos, esbozos de interfaz, diagramas del sistema, diagramas de la instalación, etc.).



Esta forma de ver el proceso de desarrollo es consistente con la realidad, ya que en general nunca se realiza una actividad de manera aislada. Por ejemplo, cuando se hace análisis del problema también se hace adicionalmente algo de la captura de requisitos o del diseño, por lo que al final del ciclo no sólo se obtienen productos del análisis, sino también de los otros flujos de trabajo.

Es por esto que el Proceso Unificado ve al desarrollo de software como un solo ciclo iterativo e incremental, en donde se realizan todos los flujos de trabajo en distintas medidas y los productos obtenidos son versiones más detalladas de los productos de entrada. Cada etapa del proceso de desarrollo se puede ver como una serie de iteraciones de un mismo ciclo que está compuesto por los cinco flujos de trabajo.

En la figura 1.1 se resume el enfoque del PU y se muestra la cantidad de trabajo que se emplea generalmente en cada flujo de trabajo, dependiendo de la iteración y fase en que está el proyecto.

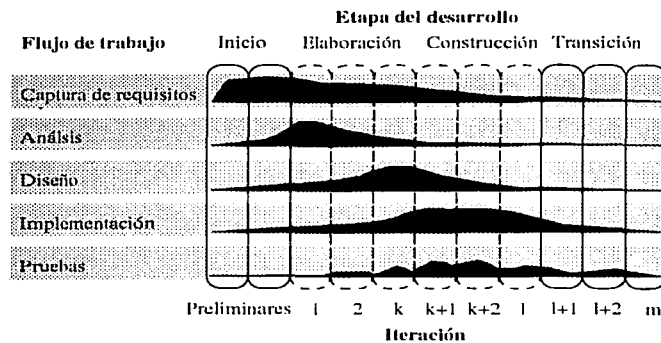


Figura 1.1: Carga de trabajo típica para cada flujo y por etapa. Ilustración tomada de *El Proceso Unificado de Desarrollo de Software* ([3]).

Para cada flujo de trabajo, el Proceso Unificado especifica las *actividades* que se deben llevar a cabo para lograr los objetivos, así como las personas a las que deben de ser asignadas (*roles*), los productos que se obtendrán y la forma de documentarlos (*artefactos*), todo con el fin de llevar el proceso de desarrollo de manera controlada.

Uno de los motivos del éxito del PU es la forma en que divide el trabajo. Los objetivos de cada iteración del desarrollo no se asignan de manera arbitraria, sino que se asignan de acuerdo a las prioridades de entidades llamadas *casos de uso*, que resumen la funcionalidad que deberá tener el sistema. Estas entidades, y en general toda la documentación de los flujos de trabajo, se expresa en términos del *Lenguaje Unificado de Modelado* (UML del inglés *Unified Modeling Language*) que por su similitud con el modelado orientado objetos, permite efectuar la implementación y pruebas del sistema de manera casi directa a partir de la documentación generada en los otros flujos de trabajo. La notación por casos de uso permite resumir la funcionalidad que tendrá el sistema en términos del lenguaje del cliente; también expresa sus relaciones y dependencias, por lo que los casos de uso resultan muy convenientes para dividir y priorizar el trabajo de desarrollo.

El prototipado de la interfaz de usuario es una actividad que se contempla dentro del flujo de trabajo de captura de requisitos, ya que por un lado ayuda a entender y establecer los requerimientos del sistema y por otro permite corroborarlos con el cliente antes de empezar en forma el proceso de desarrollo¹. Es por eso que a continuación se explicará en detalle el flujo de trabajo de captura de requisitos y cómo se emplean sus productos en los demás flujos de trabajo.

Es importante resaltar que el PU no especifica cuándo los productos del prototipado de interfaces de usuario son retomados y convertidos en la interfaz de usuario. Probablemente ésto se hace en alguna actividad del diseño o de la implementación.

1.1.2. El flujo de trabajo de captura de requisitos del PU

En este flujo del PU, el objetivo es averiguar las características del sistema que se debe de construir, lo cual comprende determinar los requerimientos del cliente, la funcionalidad que debe proporcionar el nuevo sistema, sus propiedades, así como el contexto en el cual trabajará. Dado que este flujo del desarrollo se ejecuta antes de firmar un contrato con el cliente, también se crean esbozos de la interfaz con el propósito de acordar con él la funcionalidad y apariencia que tendrá el sistema terminado. Al igual que los demás flujos del Proceso Unificado, el de captura de requisitos es también iterativo e incremental, por lo que sus actividades se pueden repetir varias veces, generando artefactos cada vez más complejos y completos.

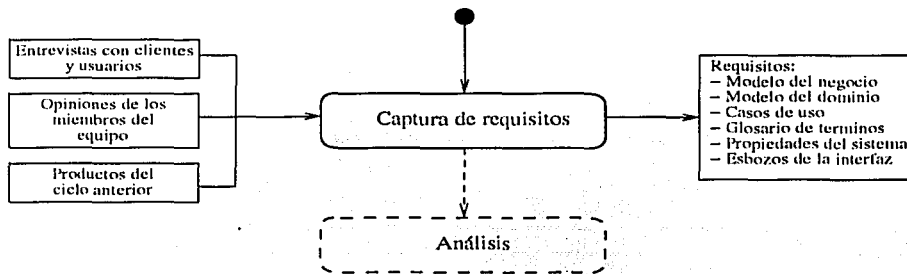


Figura 1.2: Diagrama de actividades de UML ([27]) que muestra los objetos requeridos para este flujo, así como los objetos obtenidos.

Entonces, de acuerdo al Proceso Unificado, la captura de requisitos es el primer flujo de trabajo que se ejecuta en cada ciclo (figura 1.2), y después de éste sigue el análisis, el diseño, etc.

El flujo de captura de requisitos comprende cuatro pasos: “Enumerar requisitos candidatos”, “Comprender contexto del sistema”, “Capturar requisitos funcionales” y “Capturar requisitos no funcionales”. Si bien el PU no especifica en qué secuencia deben de efectuarse, o si se ejecutan en paralelo, las presenta en ese orden y así será respetado en este trabajo.

¹De la figura figura 1.1 se puede observar que las iteraciones en la fase de inicio, en los cuales generalmente se hará la mayor parte de la captura de requisitos, se consideran como iteraciones preliminares.

Enseguida se detalla cada una de estos pasos, que en su conjunto forman el flujo de trabajo de "Captura de requisitos" tal y como es presentado en el Proceso.

Paso 1: "Enumerar requisitos candidatos"

En este paso se enumeran los requisitos propuestos por los clientes, usuarios y desarrolladores, con el fin de mantener una lista de requisitos que ayude a estimar costos de desarrollo y a dividir el proceso en iteraciones. Cada requisito propuesto tendrá un número que lo identifique, un estado (propuesto, aprobado, validado), un costo estimado de implementación (horas.hombre/máquina), una prioridad (crítico, importante, secundario) y un nivel de riesgo asociado a la implementación (crítico, significativo, ordinario). A partir de los atributos de los requisitos propuestos se decide cuáles requisitos serán considerados en la iteración y cuáles se dejarán para una posterior. El diagrama que resume esta actividad es el siguiente:

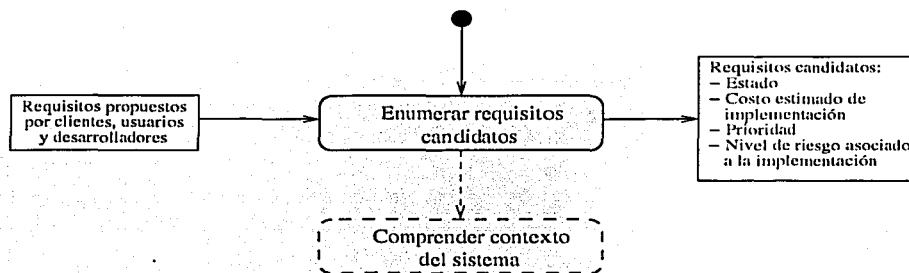


Figura 1.3: Diagrama del paso de enumerar requisitos candidatos, basado en la descripción y sugerencias que da el PU sobre el mismo.

Paso 2: "Comprender contexto del sistema"

En esta paso, tal y como lo indica su nombre, se tiene que comprender el contexto del sistema, lo cual comprende desde conocer las reglas del negocio, el ámbito de trabajo, técnicas y procedimientos de los usuarios para realizar su trabajo, su experiencia, etc. hasta sus gustos, molestias y capacidades físicas. De este conocimiento se desprenderá la funcionalidad que deberá tener el sistema de software, las restricciones que tendrá, los requerimientos mínimos que debe de cumplir, etc.

La captura de requisitos depende totalmente de esta paso, ya que si no se ha comprendido bien el contexto en donde trabajará el sistema, es probable que los requerimientos encontrados no sean correctos o estén incompletos, lo cual lleva al desarrollo de un software que no le servirá a los usuarios.

Este es una de los pasos más complejos del flujo de trabajo ya que para realizarlo con éxito no sólo se necesita aplicar una metodología de investigación, sino también tener ciertas aptitudes por parte de los desarrolladores en su papel de analistas.

El principal problema es que generalmente el desarrollador cree que con sólo preguntar al cliente o usuario los aspectos básicos del negocio podrá construir el sistema de software,

resolviendo sus dudas mediante supuestos que a él le parecen lógicos. En estos casos, los sistemas resultantes tienen alta probabilidad de fracasar por no cumplir con la funcionalidad que requieren los usuarios, por ser de difícil uso o por entorpecer su trabajo, y la razón es que la "lógica" cambia de negocio a negocio (incluso de persona a persona), y los supuestos que para el desarrollador son lógicos para el usuario pueden ser descabellados. Viceversa, hay aspectos que para el usuario son evidentes y que los tiende a omitir en las entrevistas con los desarrolladores. El panorama empeora cuando el sistema a desarrollar es nuevo, ya que inclusive ni el usuario ni el cliente pueden saber qué es lo que quieren con exactitud.

Aunque el Proceso Unificado identifica estos problemas, no propone una forma de solucionarlos. Sólo indica que el desarrollador (analista) debe adaptar sus técnicas en cada situación y debe de poner especial cuidado en rectificar sus supuestos. Sin embargo, pro-

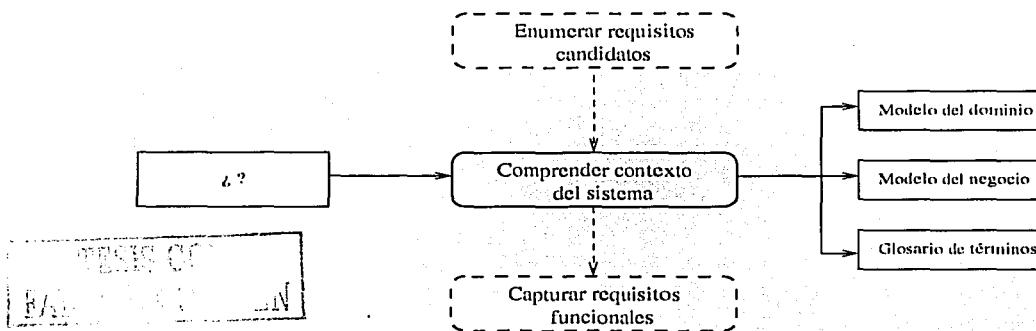


Figura 1.4: Diagrama del paso de comprender contexto del sistema. El PU no especifica ni cómo efectuarlo ni qué artefactos necesitará.

pone dos formas de representar el conocimiento adquirido: los diagramas del modelo del dominio y del modelo del negocio.

En los diagramas de modelo del dominio se identifican todos los objetos del contexto en donde trabajará el sistema, así como sus relaciones. En los diagramas de modelo del negocio se establecen los procesos que intervienen en el negocio, quién los ejecuta y bajo qué circunstancias, las operaciones que realizan, etc. El modelo del dominio es particularmente importante, ya que sus objetos y relaciones ayudan a identificar clases y jerarquías en los flujos de análisis y de diseño del Proceso Unificado. A partir de los modelos de dominio y de negocio se debe obtener un glosario de términos que ayude a relacionar los objetos del sistema de software con los objetos reales de los usuarios. Como ya se mencionó antes, el PU no menciona de qué forma se pueden obtener los modelos del dominio y del negocio.

Paso 3: "Capturar requisitos funcionales"

En este paso se define la funcionalidad y características que tendrá el sistema de software, a lo cual en conjunto se le llama *requisitos funcionales*. Como es importante que todo el grupo de desarrollo tenga la misma noción sobre los requisitos, éstos son representados utilizando los *casos de uso* de la notación UML. En ellos se especifican los actores que

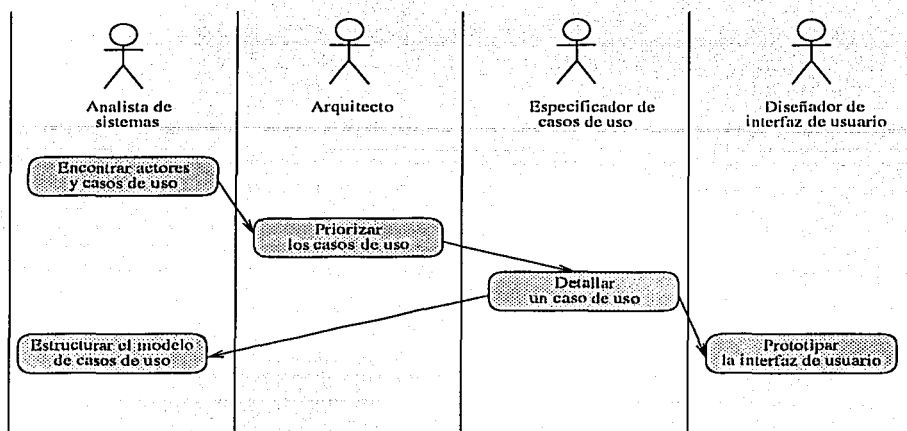


Figura 1.5: División en actividades de la captura de requisitos funcionales. Fuente: *El Proceso Unificado de Desarrollo de Software*[3].

interactuarán con el sistema, las funciones que accesarán, las relaciones entre éstas, etc.

El uso de los casos de uso no sólo resume la funcionalidad de manera conveniente, sino que permite efectuar una conversión casi directa de casos de uso a objetos y procedimientos en la aplicación. Es por esto que las demás actividades en el desarrollo dividirán sus ciclos de trabajo sobre los casos de uso.

A partir del conocimiento que se tiene sobre el contexto del sistema, se debe definir su alcance y la funcionalidad que debe de tener para lograrlo, expresando cada función en forma de un caso de uso. Posteriormente, en el flujo de trabajo de análisis, cada uno de los casos de uso será analizado y se encontrarán los sub-casos de uso que se requieren para llevarlo a cabo, y así sucesivamente hasta tener una descomposición de bajo nivel de abstracción de los requerimientos más complejos. Esta descomposición será representada en forma de un *diagrama de casos de uso* y servirá como "esqueleto" de la estructura lógica del sistema.

También durante la captura de requisitos funcionales se crean los esbozos de la interfaz y un glosario de términos más específico. Con los esbozos de la interfaz se puede discutir la funcionalidad del sistema antes de comenzar su desarrollo, e inclusive antes de firmar un contrato. El propósito del glosario de términos actualizados es relacionar los objetos que componen los casos de uso con los requisitos funcionales que se enumeraron en el primer paso. Este mismo glosario será utilizado también para nombrar los objetos, clases y procedimientos en las siguientes etapas del desarrollo.

Dado que la captura de requisitos funcionales es un proceso no trivial, el PU propone una forma de trabajo para implementarlo, especificando las distintas actividades que se tienen que desarrollar y los roles correspondientes. El esquema de trabajo se muestra en la figura 1.5.

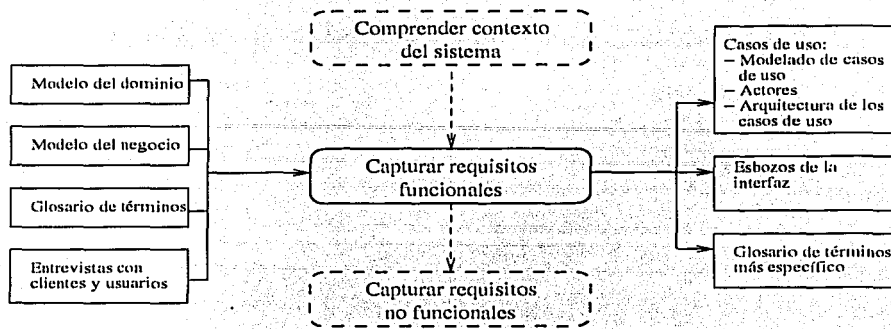


Figura 1.6: Diagrama del paso de "capturar requisitos funcionales".

Paso 4: "Capturar requisitos no funcionales"

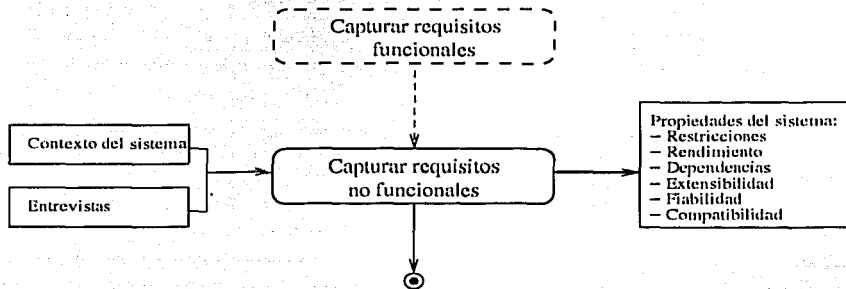


Figura 1.7: Diagrama del paso de "capturar requisitos no funcionales".

En este paso se determinan los requisitos adicionales del sistema, tales como rendimiento, compatibilidad, etc. Éstos se encuentran también a partir de las entrevistas con clientes y usuarios y de los modelos del contexto del sistema (figura 1.7). Dado que en el proceso de desarrollo los requisitos no funcionales influyen sólo en las decisiones del flujo de trabajo de diseño, el PU propone tenerlos como una simple lista y conservarlos hasta ese flujo.

1.1.3. El problema de la captura de requisitos

Como se puede observar en lo hasta aquí expuesto, aún cuando se use el Proceso de Desarrollo Unificado al pie de la letra, el problema de la captura de requisitos no es solucionado en su totalidad ya que, como antes se explicó, no se propone cómo realizar cada una de los pasos que intervienen en dicho análisis. En vez de eso, se hace uso de ejemplos que ilustran

la forma de desarrollarlas. El inconveniente consiste en que la captura de requisitos sigue dependiendo en su totalidad de la experiencia y habilidad del desarrollador.

A este respecto es muy ilustrador el trabajo de Hammond, Jourgensen, et al ([30]), quienes constatan que, contrario a una idea generalmente aceptada, los diseñadores de aplicaciones interactivas toman muy en serio la definición y formulación de la tarea del usuario en el momento de la especificación de requerimientos. Desafortunadamente, estos esfuerzos dan los frutos esperados en muy pocas ocasiones, pues los diseñadores, ignorantes de las técnicas de análisis de la tarea y de los modelos adecuados, terminan prisioneros en su propio análisis lógico y de su intuición acerca de las posibles necesidades del usuario.

La forma de validar los resultados de la captura de requisitos, mediante revisiones con el cliente de documentos escritos y esbozos de interfaz, también tiene importantes inconvenientes. En los primeros ciclos de desarrollo, en los cuales sólo se tienen los productos de la captura de requisitos, es común que el cliente acepte requisitos incompletos, ambiguos y/o incorrectos en base a los esbozos que se le proporcionaron. Para cuando se tiene la interfaz implementada en el sistema, los cambios propuestos por los usuarios durante las pruebas o demostraciones pueden tener repercusiones importantes en el proyecto, debido a que generalmente el sistema está casi terminado o por entregarse. El origen de esta problemática es que, para que los usuarios identifiquen las características que debe tener el nuevo sistema, es necesario que utilicen la nueva interfaz para realizar sus tareas cotidianas. Es poco común que imaginen el mundo de posibilidades cuando miran los esbozos en papel y hagan todas las sugerencias pertinentes antes de que comience el desarrollo en forma del sistema. Además, estos primeros esbozos son generalmente presentados no a los usuarios sino a los clientes y dueños del producto, lo cual no es adecuado pues no necesariamente conocen los detalles de las actividades que se realizarán en el sistema.

Don Norman [9] de Apple Computer, resume muy bien el sentir general de los desarrolladores respecto a la captura y análisis de requisitos (traducción libre):

Mi experiencia es que ... las especificaciones iniciales ... están generalmente incorrectas, ambiguas o incompletas. En parte esto se debe a que son capturadas por personas que no entienden los problemas reales encarados por usuarios eventuales... Aún peor, los usuarios pueden no saber lo que quieren, así que tenerlos en el equipo de desarrollo no es una solución. De hecho, capturar especificaciones correctas puede no ser factible, pues ... el verdadero entendimiento de una herramienta viene sólo con el uso, en parte porque las nuevas herramientas cambian al sistema, y por tanto las necesidades y los requerimientos.

Es por los problemas de la captura de requisitos antes mencionados que en este trabajo se propone el uso del *Análisis de la Tarea*, que se explica a continuación, como un medio para obtener los requerimientos del nuevo sistema a partir de las actividades y características de los usuarios, y comenzar así el desarrollo con requisitos más ciertos.

1.2. El Diseño centrado en el usuario y el Análisis de la Tarea

Para los usuarios de las computadoras, la facilidad de uso se ha convertido en un factor importante para decidir qué software o qué equipo comprar. Hoy en día el tiempo es oro y nadie quiere perderlo leyendo manuales; los usuarios quieren invertir su tiempo en lograr objetivos y no en aprender cómo usar un sistema. Cualquier sistema que haga perder tiempo al usuario o que le cause frustración es, para el usuario, un sistema malo o que no puede

FALLA DE ORIGEN

hacer lo que se requiere. La usabilidad se ha convertido en un requisito indispensable en cualquier software comercial, y para los desarrolladores y sus sistemas es un tema de vida o muerte.

Actualmente la mayoría de las organizaciones que producen software tienen a la usabilidad como un requisito en sus productos, por lo que es tomada muy en serio en los procesos de desarrollo. Sin embargo hay muchas formas de mejorar la facilidad de uso de un sistema, cada una con sus ventajas y desventajas. La que ha sido empleada en el desarrollo de la mayor parte del software comercial consiste en desarrollar productos, probarlos en el mercado, y luego trabajar sobre nuevas versiones o actualizaciones. La ventaja de este esquema es que casi no se invierte tiempo en las etapas de captura de requisitos y de análisis, por lo que se pueden desarrollar sistemas en poco tiempo y sacarlos al mercado. Con la actual guerra de versiones en el mercado, este ha sido un factor de mucho peso por el cual las empresas seleccionan esta estrategia. La gran desventaja es que los productos obtenidos en un ciclo de desarrollo generalmente no cumplen con las expectativas de sus consumidores, y éstos tienen que esperar a las siguientes versiones para tener un producto más cercano a sus necesidades. Para que el usuario tenga el producto que requiere, debe esperar varios ciclos de desarrollo-venta-retroalimentación-corrección, lo cual le implica altos gastos y desconfianza en la empresa.

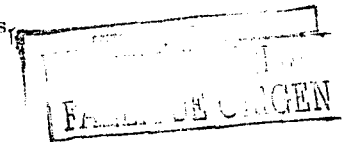
La alternativa a esta forma de producir es desarrollar software basado en *diseño centrado en el usuario*. En este esquema, antes de comenzar en forma el desarrollo del producto se hace un estudio sobre el usuario, sus actividades y su ambiente, de manera que se tengan mejor identificados sus requerimientos y se pueda elaborar la aplicación en base a ellos. Cuando ésta está terminada se tiene más certeza de que cumple con las necesidades del usuario que en el caso anterior y, aunque probablemente se necesiten elaborar nuevas versiones del producto, se requieren significativamente menos repeticiones del proceso de desarrollo para entregarle al usuario el sistema que él espera.

La aparente desventaja del diseño centrado en el usuario es que en las primeras versiones del producto se emplea mucho tiempo en las etapas de captura de requerimientos y de análisis, y no es posible desarrollar software a la velocidad con que lo hace el mercado actual. En realidad, el trabajo invertido da como resultado sistemas que se encuentran más cerca de las necesidades de los usuarios y por lo tanto, sistemas que necesitaran menos desarrollo posterior (ver figura 1.8).

De esta forma, se puede ver que utilizando el esquema de desarrollo clásico se requieren de más recursos para obtener el producto requerido que con el desarrollo empleando el diseño centrado en el usuario.

El diseño centrado en el usuario aporta:

- Menos problemas con la comprensión y validación de requisitos.
- Menores costos de capacitación.
- Incremento en la productividad.
- Menores tasas de error del usuario.



A continuación se presenta la técnica más ampliamente usada en el diseño centrado en el usuario para capturar las características de los usuarios, sus tareas y su entorno: El Análisis de la Tarea. Una variante de éste será el que se propondrá integrar al Proceso Unificado.

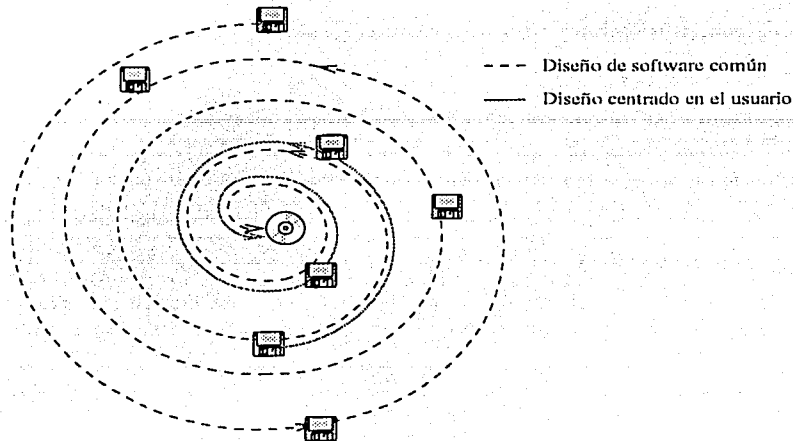


Figura 1.8: Si se ve el desarrollo de software como una espiral, en donde el desarrollo sucesivo de nuevas versiones lleva a productos más apegados a las necesidades del usuario, se tiene que el uso del diseño centrado en el usuario puede ahorrar varias repeticiones del proceso de desarrollo.

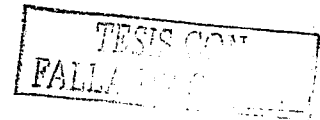
1.2.1. El Análisis de la Tarea

Los estudios que ha realizado la gente de Interacción Humano Computadora (IHC), y la experiencia, indican que si se llevan a cabo entrevistas y pruebas con los usuarios para conocer sus capacidades perceptivas, la percepción de su tarea y de su entorno, sus habilidades y experiencia, y sus preferencias, se tienen los elementos necesarios para hacer un diseño de interfaz exitoso. Esto permite concluir que la frase clave para el diseño de interfaz es “*conoce a tu usuario*” (y conocer al usuario significa conocer los aspectos mencionados), y gran parte de esta tarea se ha concentrado en un subcampo de IHC conocido como *Análisis de la Tarea (AT)*.

Uno de los primeros aspectos importantes es que el Análisis de la Tarea se efectúa sobre los usuarios potenciales del sistema de software, y no sobre los clientes que lo financiarán. Esto es porque el objetivo no es sólo conocer el modelo del negocio, como en el PU, sino conocer la forma en que cada usuario efectúa sus tareas, y tomar provecho de esa información en la elaboración de un modelo del usuario y su tarea.

El AT no sólo apoya la captura de requisitos, también ataca los siguientes temas:

- Diseño de interfaces
- Especificaciones de trabajo.
- Organización de personal y de trabajo.
- Diseño de entrenamiento y capacitación.



- Desarrollo de manuales y guías.
- Administración de seguridad.
- Revisiones de desempeño.
- Distribución de funciones, ya sea entre personas en una organización o entre hombres y máquinas.

Dependiendo de la notación, un Análisis de Tarea captura distintos tipos de información sobre el usuario y su tarea, por lo que éste se puede clasificar en (ver [12], [5] y [7]):

- *Análisis Jerárquico de la Tarea*: Útil para describir, mediante una representación gráfica, la actividad del usuario por medio de la descomposición de su tarea en tareas más simples, hasta llegar a acciones concretas. El proceso consiste en identificar las tareas del usuario y sus objetivos, categorizarlas, descomponerlas en tareas más simples y luego revisar la exactitud de la descomposición y del esquema. Dado su carácter descriptivo, este tipo de análisis es particularmente útil en situaciones donde lo que se quiere es capturar qué se hace, cómo se hace y con qué se hace; información invaluable cuando se trata de estipular las características de un sistema informático que habrá de apoyar al usuario en sus actividades de manera eficiente.
- *Análisis Cognitivo de la Tarea*: Se basa en técnicas que capturan con alguna forma de representación el conocimiento que la gente tiene, o que requiere, para completar una tarea. Su objetivo es el de proponer marcos de trabajo que modelen el comportamiento cognitivo humano, de forma tal que sea posible analizar y predecir el comportamiento en distintas situaciones.
- *GOMS*: Es una técnica de modelado de la tarea que ve al usuario como un procesador de información, con distintos tipos de memoria, entradas y salidas. Esta arquitectura de procesador puede ser usada para simular y predecir el desempeño de un usuario al realizar su tarea con un sistema dado. Esta interacción entre el usuario y su interfaz es modelada a cuatro distintos niveles de abstracción: objetivos, operadores, métodos y reglas de selección. Sin embargo, las distintas versiones de GOMS requieren de comenzar con una lista de objetivos, la cual tiene que obtenerse por otros análisis de la tarea.
- *Análisis del Conocimiento de la Tarea*: Es un método de Análisis de la Tarea que se desarrolla desde un punto de vista de la teoría de las Estructuras de Conocimiento de la Tarea (Task Knowledge Structures - TKS). En él se efectúa un análisis sobre los documentos, entrevistas y sesiones de usuario que están relacionados con una tarea en particular, con el fin de determinar similitudes entre las distintas formas de realizar una misma tarea, y predecir cómo la gente suele realizar una tarea en particular.

1.2.2. El Análisis Jerárquico de la Tarea

Para la captura de los requisitos del usuario, la metodología más utilizada es la del Análisis Jerárquico de la Tarea, ya que es el modelo que proporciona las herramientas para representar el conocimiento del usuario en términos de sus objetivos, tareas y acciones; esto es: ¿Qué hace?, ¿Cómo lo hace?, ¿Qué necesita?, ¿En qué momento?, etcétera. En lo que sigue, se presentará con mayor detalle el Análisis Jerárquico y las variantes que presentan

sus distintos formalismos. Más información sobre los “Análisis Cognitivos”, “Análisis del Conocimiento” y de “GOMS” se pueden encontrar en Preece [12], Whitefield [7] y Scheffer [5].

El Análisis de la Tarea del Usuario comienza desde los niveles más altos de abstracción de la tarea hasta llegar a los más bajos o concretos: Primero se determina la *tarea principal* del usuario; luego se encuentran las *tareas* que tiene que efectuar y que son necesarias para cumplir el *objetivo* de su tarea principal; después cada tarea es descompuesta en sus *subtareas*; y finalmente se encuentran las *acciones* que se efectúan en cada subtarea. Como resultado se obtiene un *árbol jerárquico* de la tarea, el cual tiene en su raíz la tarea principal del usuario, en sus ramas las tareas con las cuales se logra, y en sus hojas las acciones concretas requeridas.

Un ejemplo de un árbol jerárquico para la tarea de un empleado de mostrador en un video centro, aplicando el procedimiento del análisis jerárquico de la tarea, se presenta en las figuras 1.9 y 1.10.

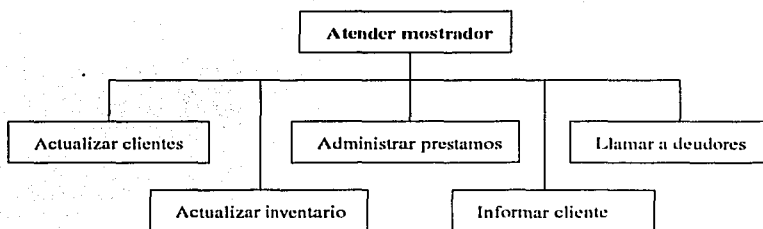


Figura 1.9: Descomposición de la tarea de un empleado de mostrador en un video centro. El análisis comienza descomponiendo la tarea principal en tareas, y luego cada tarea en subtareas hasta que cada tarea quede en términos de acciones.

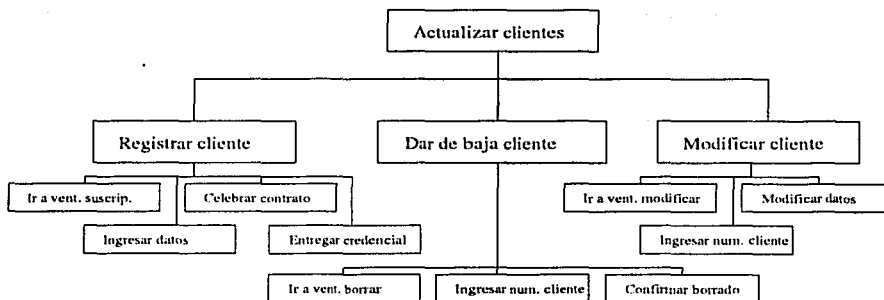
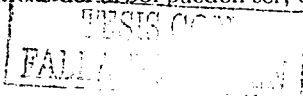


Figura 1.10: Detalle de una subtarea del árbol de la tarea.

Nótese que las tareas del árbol pueden ser, o no, asistidas por una computadora, ya que



éste representa la forma en que el usuario realiza sus objetivos con las herramientas con que dispone.

J. Proece [12] describe el Análisis de la Tarea de forma procedural como sigue:

1. Iniciar el análisis:
 - a) Especificar la tarea principal.
 - b) Descomponer la tarea principal en subtareas.
 - c) Asegurarse que la descomposición en subtareas es lógica, técnicamente correcta, y que ninguna falta.
 - d) Validar la descomposición con el usuario, mediante simulaciones y entrevistas².
2. Desarrollar el análisis:
 - a) Decidir el nivel de detalle y de descomposición.
 - b) Decidir la estrategia del análisis (desarrollar por ramas del árbol o por niveles).
 - c) Continuar el análisis a partir de la parte del árbol que ya se tiene.
3. Finalizar el análisis:
 - a) Revisar el análisis para verificar que las descomposiciones y el número de subtareas es consistente.
 - b) Presentar el árbol de la tarea a alguien que no estuvo presente en el análisis pero que conoce la tarea suficientemente bien.

El análisis jerárquico puede representar distintos aspectos de la tarea, dependiendo de qué notación o formalismo se esté empleando. Entre ellos están:

- Secuencias en los comportamientos del sistema: El detalle en la secuencia de los comportamientos que se dan dentro del sistema depende de cuál es el objetivo del análisis de la tarea.
- Objetivos de la tarea: Estado de las cosas que el usuario desea, y por el cual realiza su tarea.
- Descomposición: El ejemplo de la tarea de empleado de mostrador de video centro muestra una descomposición de comportamientos (o de tareas), pero también se puede realizar una descomposición de los objetivos. Cuál de estas descomposiciones emplear depende del formalismo empleado y de los objetivos del AT.
- Objetos del sistema de trabajo y sus atributos: Las tareas se realizan con el objetivo de cambiar algo en el entorno del usuario, por lo que es importante conocer sobre qué objetos y de qué forma se harán.
- Estructuras físicas y abstractas en el sistema interactivo de trabajo: Un ejemplo de estructura abstracta es el conjunto de requisitos que una persona necesita cumplir para ser socio de un video centro. Ejemplo de una estructura física son las componentes de la computadora con la cual trabajará el empleado de mostrador (teclado, ratón, impresora, pantalla, etc.).
- Desempeño: Una descripción del desempeño de una tarea puede consistir de la calidad del producto deseado y del obtenido en términos de los recursos empleados.

²Esta actividad no está contemplada en [12], pero es conveniente realizarla para validar el análisis hecho.

1.3. Enfoque del trabajo

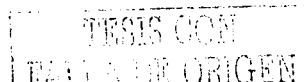
Como se ha descrito anteriormente, dentro de la Ingeniería de Software, la fase de análisis de requisitos es señalada como de particular importancia; primero, porque en ella se deben definir de manera precisa, completa y sin ambigüedades, las características y requerimientos que deberá satisfacer el sistema futuro; segundo, porque es la etapa en que las correcciones se pueden realizar a un costo moderado, comparado con el costo que implica corregir la aplicación en fases más avanzadas de su desarrollo. Sin embargo, hemos visto que realizar de manera adecuada este análisis se ha revelado como un proceso complicado, caro en tiempo y difícilmente libre de errores. Un indicio de las dificultades que este análisis encierra se puede observar en la poca información que brindan al respecto los procesos más reconocidos y mejor establecidos en la IS como son el Proceso Unificado, el Team Software Process, o el eXtreme Programming. Esta falta de información en los procesos paradigma muestra que integrar a un proceso de software qué quiere hacer el usuario, cómo debe hacerlo, qué herramientas e informaciones necesita, etc. es un problema que no ha sido solucionado satisfactoriamente.

Por otro lado, como se vio el problema del análisis del usuario y sus objetivos ha sido objeto de trabajo de los estudiosos de la Interacción Humano-Computadora. En efecto, el objetivo principal de esta comunidad es desarrollar sistemas que permitan apoyar a los usuarios de manera eficiente en sus tareas, y no retardarlos con aplicaciones complejas, de organización caprichosa y funcionamiento poco comprensible. Esto puntos, es evidente, pasan por una correcta interpretación de las necesidades del usuario y de sus objetivos. Así, la comunidad de IHC ha estudiado y desarrollado diversas metodologías para el desarrollo de aplicaciones que denominan *Centradas en el usuario*, como una manera de expresar que son aplicaciones en las que los objetivos y procedimientos del usuario tienen prioridad sobre los aspectos técnicos. Desafortunadamente estas metodologías no han logrado impactar el mundo de la Ingeniería de Software por varios motivos, como se explicó anteriormente.

En este trabajo se parte de la hipótesis de que la dificultad principal se encuentra en la falta de un lenguaje común, entre la Ingeniería de Software y la Interacción Humano Computadora, que permita a ambas comunidades entender, utilizar y aprovechar los trabajos desarrollados por la otra. Es por eso que, para encontrar un lenguaje común entre las técnicas del Análisis de la Tarea y el Proceso Unificado de Desarrollo, es necesario analizar e identificar los puntos de coincidencia, establecer equivalencias, y fijar los métodos que permitan traducir e integrar de manera formal y eficiente la información.

El resultado esperado de integrar el AT al PU será un Proceso en el que el que el análisis de requisitos sea especificado y validado de una manera más rigurosa, que contenga los elementos que permitan hacer desarrollos "centrados en el usuario", y que conlleve al desarrollo de aplicaciones de calidad, que apoyen al usuario en sus tareas, y que minimicen los cambios a realizar en fases finales del desarrollo.

Un punto de especial interés para el medio de desarrollo de software será la posibilidad de obtener los primeros diagramas de casos de uso a partir del análisis de la tarea, pues como se mencionó anteriormente, el Proceso Unificado carece de una forma procedural de generar dichos diagramas. Esto deberá ayudar a disminuir los casos en que desarrollos de sistemas ya en fases finales requieren de costosas modificaciones, adhesiones o enmiendas (en ocasiones muy costosas). De hecho, es de esperarse que el tiempo adicional que se empleará para efectuar el análisis de tarea, pueda verse compensado con una reducción en los tiempos de Validación y Verificación del producto, pues con el árbol de tareas, los casos de uso, sus tareas asociadas y los objetivos fundamentales, podría ser más sencillo revisar



que el producto que se desarrolla cumple con las expectativas y requerimientos del cliente y usuarios.

En el siguiente capítulo se abordará un modelo de Análisis de Tarea que presenta características de interés para el enfoque del trabajo. Posteriormente, se analizarán los temas relacionados con la integración del AT al PU, para luego agregar al PU un número limitado y ponderado de actividades, roles y artefactos, todos ellos documentados como se hace en el PU. Finalmente, en el Capítulo 5, se aplicará el proceso propuesto a un ejemplo real para evaluar los resultados y medir los beneficios.)



Capítulo 2

El formalismo MAD STAR

2.1. El Análisis de la Tarea empleando MAD STAR

En este trabajo se empleará el formalismo MAD* (*Méthode Analytique de Description tâcheS orientÉ spécifiqueAtion d'interfAcés*) para modelar la tarea del usuario. MAD* tiene como meta representar los requerimientos del usuario en un modelo que sea comprensible tanto por ergónomos, como por desarrolladores, y que represente la forma en que el usuario concibe a su tarea, poniendo así las bases para un diseño centrado en él.

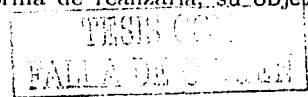
MAD* fue concebido por Hammouche ([31]) y revisado por Gambon ([1]). Ambos trabajos se basan en una primera versión llamada MAD, desarrollada por Scapin y Pierret-Golbreich ([32]) como una forma de tomar en cuenta al usuario y su tarea en los ciclos de diseño. El objetivo era proveer a los equipos de desarrollo una herramienta metodológica que les permitiera abstraer la información de la tarea del usuario necesaria para el diseño de interfaces, de una forma suficientemente formal que permitiera su integración en los procesos de desarrollo. Así, se trata de un lenguaje de trabajo que facilita el diálogo e intercambio de información entre los participantes de un equipo de diseñadores.

MAD* fue creado de acuerdo a los siguientes criterios:

- Ser riguroso y estable (dejando poca posibilidad de interpretación);
- mejorar la calidad del trabajo de análisis, en particular del punto de vista de coherencia y completitud;
- permitir la descripción tanto desde un punto de vista procedural (secuencia de estados) como declarativo (información necesaria para la tarea);
- permitir el modelado de acciones paralelas, y no sólo secuenciales;
- representar los conceptos de una forma jerárquica y uniforme.

MAD STAR se basa en el paradigma de la estructura jerárquica, ampliando la noción de elemento jerárquico con los aspectos de orden (antes, durante, después) y del estado del mundo (información necesaria para la ejecución de la tarea). El método está asociado a una técnica de captura de información basado en entrevistas *dirigidas* y *no-dirigidas*, así como a procedimientos de observación para la validación de las descripciones ([25]).

Así, para cada nodo del árbol MAD* modela en una ficha descriptiva el nombre de la tarea, su posición en la estructura jerárquica, la forma de realizarla, su objetivo, su



importancia, los eventos que la disparan y la detienen, y los resultados de su ejecución, entre otros. Cada uno de estos campos es un atributo con un nombre y valores posibles, como se verá más adelante, y para llenar esta información los analistas requieren de un conocimiento profundo de la tarea del usuario. Las técnicas de entrevista, construcción del árbol y llenado de sus fichas constituyen en sí una metodología explícita y procedural para adquirir este conocimiento.

Es por lo anterior que en este trabajo se escogió MAD* como el formalismo para efectuar el Análisis de la Tarea, pues contiene toda la información necesaria para tener un levantamiento de requerimientos exitoso:

- ¿Qué se hace?
- ¿Porqué se hace?
- ¿Cómo se hace?
- ¿Con qué se hace?
- ¿Quién lo hace?
- ¿Cuándo lo hace?

Finalmente, MAD* también puede sernos de utilidad para la especificación de interfaces (como F. Gamboa y D. L. Scapin ([1]) lo indican, el formalismo aporta toda la información necesaria para crear interfaces reactivas basadas en el modelo de la tarea del usuario).

Si el lector desea ver otras metodologías basadas en el AT, que pueden traer otro tipo de beneficios, refiérase a DIANE+ ([17], [18], [19]), ADEPT ([20]), Trident ([21]), ATOM ([33]) o GOMS ([34]).

A continuación se describe en detalle los pasos (actividades) para realizar un Análisis de la Tarea empleando MAD*.

2.1.1. Actividad 1: Elaboración del árbol de la tarea del usuario

El análisis de tarea con MAD* comienza con *entrevistas no dirigidas* al usuario, en las cuales a partir de preguntas como "¿Y usted qué hace?" y "¿Podría dar más detalles?", se deja al usuario que describa de forma libre su actividad. Con la información obtenida se crea una primera versión del árbol de la tarea, el cual se detallará y refinará en subsiguientes *entrevistas dirigidas*, en las que se cuestiona al usuario sobre puntos específicos de su actividad, necesarios para completar el árbol. A cada nodo (tarea) del árbol se le asignará un número que especifique su posición en el árbol, de tal forma que, cuando se elabore la ficha que detalla sus características, sea fácil de identificar. Un ejemplo de un árbol de tarea MAD* se muestra en la figura 2.1.

2.1.2. Actividad 2: Elaboración de las fichas de descripción de las tareas

Ya que se tiene el árbol MAD* de la tarea del usuario, se elaboran las fichas de cada uno de sus nodos. Las fichas detallan el comportamiento de las tareas, especificando el objetivo, la persona que las realiza, cuándo se realizan, qué requieren, etc. Cada ficha se puede ver formada por dos partes: la primera, el *núcleo*, que describe la tarea, su importancia, cómo y por quién es ejecutada; y la segunda, las *condiciones*, que establecen en qué momento es ejecutada, en qué condiciones, y cuándo ésta concluye. Así pues, los campos contenidos en cada ficha son los siguientes:

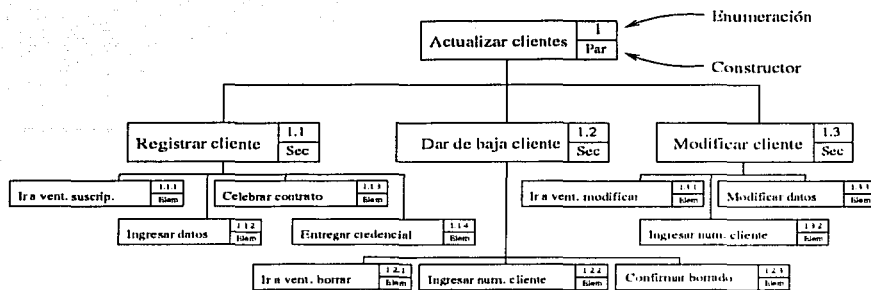


Figura 2.1: Subárbol MAD* para la tarea “Actualizar clientes” del empleado de mostrador de video centro.

- En el núcleo:
 - Número: Numérico, representando la posición de la tarea en el árbol.
 - Nombre: Alfanumérico, nombre de la tarea.
 - Objetivo: Alfanumérico, descripción del objetivo de esa tarea particular.
 - Prioridad: Entero, permite resolver conflictos entre tareas que quieren ejecutarse simultáneamente.
 - Facultativa: Booleano, indicando si la tarea es obligatoria o se puede omitir su ejecución.
 - Interruptionable: Entero. 0 indica no interruptible, 1 interruptible con posibilidad de continuar la tarea, 2 indica interruptible y al regresar se reinicia la tarea, y 3 indica interruptible y no se regresa a la tarea.
 - Tipo: Alfanumérico, indicando si es sensorimotriz o cognitiva.
 - Modo: Alfanumérico, indicando si es manual, automática o interactiva.
 - Importancia: Alfanumérico, describiendo la importancia de la tarea.
 - Papel del usuario: Alfanumérico. Descripción del papel del usuario en la tarea, así como su experiencia, preferencias, limitaciones, etc.
 - Constructor, uno de los siete siguientes:
 - Secuencial: Las subtareas se ejecutan en el orden especificado. La ejecución termina al haber ejecutado todas las subtareas obligatorias, y las optativas han sido ejecutadas o han sido omitidas.
 - Paralelo: Las subtareas se ejecutan en cualquier orden y sin compartir datos. La ejecución termina al haber ejecutado todas las subtareas obligatorias, y las optativas han sido ejecutadas o han sido omitidas.
 - Simultáneo: Las subtareas se ejecutan de forma simultánea y por distintos usuarios o entidades. La ejecución termina al haberse ejecutado todas las subtareas obligatorias, y las optativas han sido ejecutadas o han sido omitidas.

- Y: Las subtareas se ejecutan en cualquier orden y compartiendo datos. La ejecución termina al haberse ejecutado todas las subtareas obligatorias, y las optativas han sido ejecutadas o han sido omitidas.
 - O: Las subtareas se ejecutan en cualquier orden y comparten datos. La ejecución termina cuando al menos una de las subtareas obligatorias se ha completado.
 - Alternativa: Una y sólo una de las subtareas es ejecutada.
 - Elemental: La tarea no tiene subtareas. Este constructor indica que el nodo es una tarea simple o una acción que se ejecuta directamente.
- En las condiciones, todas las siguientes son requeridas, a excepción de las marcadas como opcionales:
- Estado del mundo: Objetos del mundo empleados en la tarea y de los cuales dependen las pre- y post-condiciones.
 - Pre-condiciones:
 - Condición de inicio: Estado del mundo requerido para que se puede ejecutar la tarea.
 - Condición de arranque (opcional): Estado del mundo que demanda la ejecución de la tarea. El estado de inicio debe de cumplirse para que la tarea arranque.
 - Condición de paro (opcional): Estado del mundo que indica cuándo detener la ejecución de la tarea (y de sus subtareas). Este atributo es útil si se quiere que la ejecución de la tarea se repita hasta lograr una condición.
 - Post-condición: Estado del mundo después de que la tarea terminó su ejecución.

Así, por ejemplo, la ficha para la tarea 1.1 del árbol MAD*, en la tarea del empleado de mostrador de video centro, es:

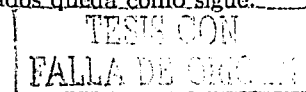


<i>Núcleo:</i>	
Número	1.1
Nombre	Registrar cliente
Objetivo	Registrar los datos de un nuevo cliente y efectuar el contrato
Prioridad	5
Facultativa	Falso
Interrumpible	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Pedir datos al cliente e introducirlos al sistema. Generar contrato y credencial. El usuario no necesariamente tiene mucha experiencia en esta actividad.
Constructor	Secuencial
<i>Pre y post-condiciones:</i>	
Inicio	Hay clientes no registrados
Arranque	Llegó un nuevo cliente y éste desea asociarse al video club
Paro	Se registraron los datos del cliente y se efectuó el contrato
Postcondiciones	Se tienen los datos del nuevo cliente
Operador	Empleado

2.1.3. Actividad 3: Poner las condiciones de las fichas en términos de notación orientada a objetos

Una vez elaboradas las fichas del árbol se procede a identificar los objetos que participan en la tarea del usuario, así como sus atributos: Se revisan las pre y post-condiciones de todas las fichas de las tareas haciendo un listado de los objetos candidatos y sus atributos, para después procesar esta lista y obtener los objetos de la tarea. Posteriormente, las pre y post-condiciones de cada ficha serán puestas en términos de estos objetos, de tal forma que las modificaciones que hacen las tareas a los estados de sus objetos representen la forma en que la tarea del usuario modifica su entorno. Los objetos y atributos encontrados son correctos si permiten describir adecuadamente una tarea al poner su ficha en términos de objetos.

La ficha de la tarea 1.1 en términos de los objetos encontrados queda como sigue:

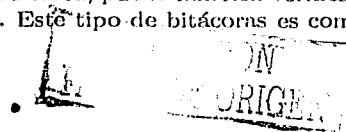


<i>Núcleo:</i>	
Número	1.1
Nombre	Registrar cliente
Objetivo	Registrar los datos de un nuevo cliente y efectuar el contrato
Prioridad	5
Facultativa	Falso
Interrumpible	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Pedir datos al cliente e introducirlos al sistema. Generar contrato y credencial. El usuario no necesariamente tiene mucha experiencia en esta actividad.
Constructor	Secuencial
<i>Pre y post-condiciones:</i>	
Inicio	Cliente.Registrado?()==false
Arranque	Cliente.Esperando?()==true AND Cliente.Atendido?()==false AND Cliente.Registrado?()==false
Paro	Cliente.Registrado?()==true AND Contrato.Celebrado?()==true
Postcondiciones	Registro_Clientes.Actualizado?()==true
Operador	Empleado

2.2. Validación de los productos del análisis de la tarea

Los resultados del uso de MAD*, al igual que los de cualquier otra metodología de análisis, deben de ser probados para verificar su validez, ya que es el analista de la tarea el que interpreta el contenido de las entrevistas y modifica el árbol de la tarea, actualizando sus fichas y objetos. Hay varias técnicas para efectuar esta validación, de acuerdo al alcance deseado y los recursos con que se cuenta:

- Observación del usuario *In Situ*: Se graba al usuario en una sesión normal de trabajo, con el objetivo de comprobar que la secuencia de acciones y eventos reales correspondan a los flujos modelados en el modelo MAD*. Como se ha de recordar, cada subtarea del árbol MAD* tiene varias pre-condiciones y una post-condición, las cuales, con ayuda de los constructores, modelan una situación real y establecen claramente cuándo debe de ejecutarse cada tarea. Si el árbol y las descripciones de sus nodos son correctos, todas las actividades realizadas por el usuario en la grabación deben de caber en un flujo de tareas en el árbol.
- Análisis de Trazas: Cuando quedan registros escritos de la actividad del usuario, que permitan comparar dicha actividad con el árbol de su tarea, puede también verificarse que sus acciones estén contempladas en el modelo. Este tipo de bitácoras es común hacerlas en puestos de alto riesgo.



- Prueba de la tarea MAD* modelada: En una sesión de prueba con el usuario, se le plantean situaciones hipotéticas modeladas en el árbol MAD*, para analizar sus respuestas y verificar que las acciones que realizaría queden también en el modelo de su tarea.
- Prueba con interfaces obtenidas a partir del modelo MAD*: Gamboa y Scapin ([1]) proponen generar interfaces reactivas de forma rápida a partir del árbol de la tarea y sus fichas; posteriormente, en sesiones de prueba con los usuarios, se les pide que realicen actividades cotidianas en las interfaces de prueba, y se analiza si el flujo de ejecución de las tareas modelado en el MAD*, representado de distintas formas en la interfaces, permite que el usuario efectúe lo pedido de una forma intuitiva y sencilla para él. Aunque este tipo de prueba requiere de más trabajo por parte del equipo de desarrolladores, e involucra otras metodologías para la generación de interfaces reactivas a partir del modelo de la tarea, aporta más información sobre la interacción del usuario y la interfaz propuesta.

2.3. Ejemplo de un Análisis de la Tarea con MAD*

El análisis de tarea que se presenta a continuación es el de un empleado de video centro encargado del mostrador. Éste se efectuó usando el formalismo MAD* en el desarrollo de un sistema para un video centro, con el objetivo de tener un mejor entendimiento de los requerimientos y desarrollar un sistema centrado en el usuario. El objetivo del nuevo sistema era mejorar el desempeño de los usuarios en sus actividades de mostrador, y eliminar los problemas técnicos existentes.

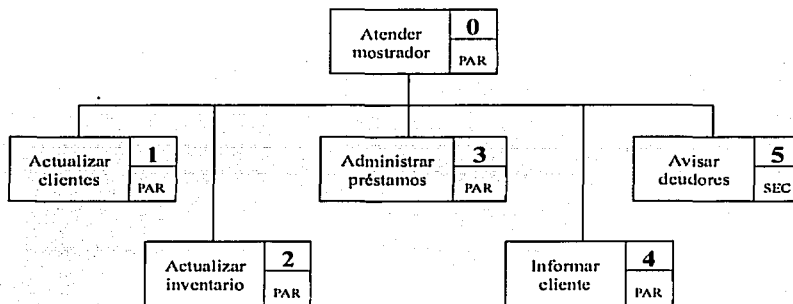


Figura 2.2: Tarea de "Atender mostrador".

Se entrevistaron a dos usuarios que laboraban en un video centro, para obtener un modelo de su tarea, y luego validar los productos con pruebas con interfaces reactivas. Todas las entrevistas fueron grabadas en cinta de audio y se tomaron las notas pertinentes.

En la primera entrevista (no dirigida), los usuarios explicaron el tipo de actividades que realizaban, los procedimientos que seguían a grandes rasgos, las herramientas con las que contaban, y los problemas o contratiempos que encontraban. En este caso, ya contaban con un sistema desarrollado a la medida que contaba con: Control de inventario de películas,



Manejo de clientes y Sistema de caja. La entrevista fue analizada por un equipo de tres analistas, obteniendo rápidamente una primera versión del árbol de la tarea (figura 2.2), así como numerosas dudas.

Como se puede observar, su tarea de "Atender Mostrador" consiste de cinco subtareas: Altas y bajas de clientes (tarea 1); altas y modificaciones en el inventario de películas (tarea 2); préstamo, cobro y recepción de películas (t. 3); dar información a clientes sobre películas y estrenos (t. 4); y llamadas de atención a clientes deudores (t. 5). La tarea 0 tiene un constructor paralelo, pues el empleado de mostrador puede, en cualquier momento, realizar cualquiera de las tareas 1 a 5. La prioridad de atender clientes sobre cualquier otra tarea será modelada con el campo de prioridad en la ficha que describa a la tarea 0, como se verá más adelante.

En dos entrevistas posteriores (dirigidas) se resolvieron las dudas y se obtuvieron descripciones muy detalladas de cada actividad, con lo cual fue posible terminar el árbol de la tarea. Es importante mencionar que en la descripción de la tarea quedó involucrado el sistema existente, pues los usuarios realizaban todas sus actividades a través de él.

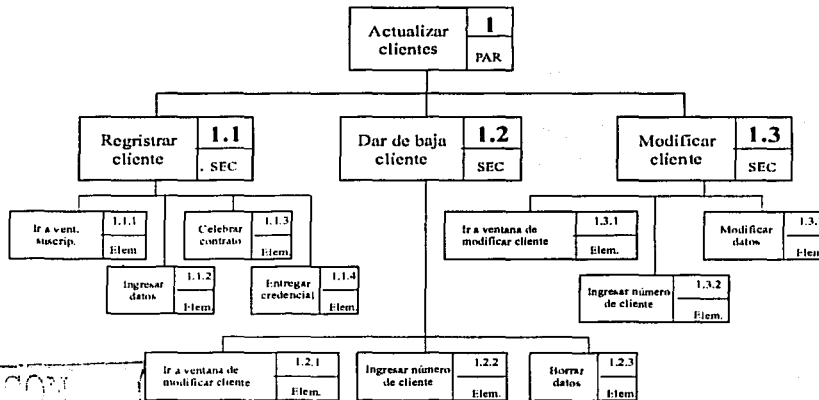


Figura 2.3: Tarea 1 de "Actualizar clientes".

En la figura 2.3 se muestra, la tarea 1 "Actualizar clientes" en detalle, con una descomposición de subtareas hasta el nivel de interacción con la interfaz del sistema actual. El nivel de detalle (granularidad) de la descripción de la tarea es decisión del analista, pero si se pretende realizar el diseño de la interfaz del nuevo sistema, es importante tener la información de la interacción del usuario con el sistema actual, y así realizar un diseño que tome en cuenta los cambios a los que los usuarios tendrán que adaptarse.

De las subtareas de "Actualizar clientes" la de más prioridad es "Registrar cliente", pues se ejecuta cuando hay un nuevo cliente al mostrador, y puede implicar que el empleado deje su actividad para atender al cliente. Esto comportamiento se modela mediante los atributos de prioridad e interruptibilidad de cada tarea, y el evento que dispara a una tarea e interrumpe a otra se tiene en la pre-condición de arranque.

En este caso, el equipo de analistas decidió también agregar etiquetas de "interruptible"

y “facultativa” al árbol de la tarea para denotar qué tareas eran del tipo interruptible 1 (se interrumpe con posibilidad de continuar la tarea) y facultativas (el usuario decide si ejecutarlas o no), respectivamente.

En las siguientes páginas, se muestra el resto de las tareas del árbol de la tarea del empleo de mostrador de video centro, obtenido a partir de la aplicación de la metodología MAD*.

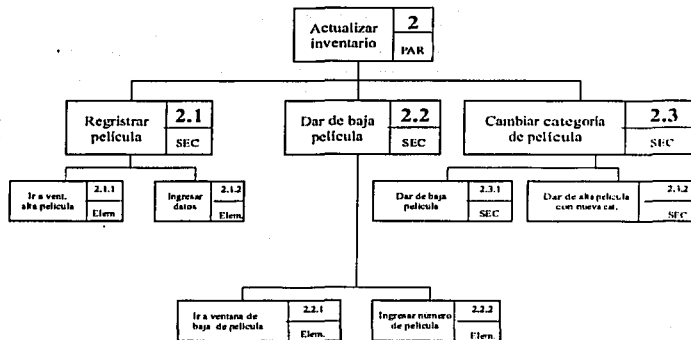


Figura 2.4: Tarea 2 de “Actualizar inventario”.

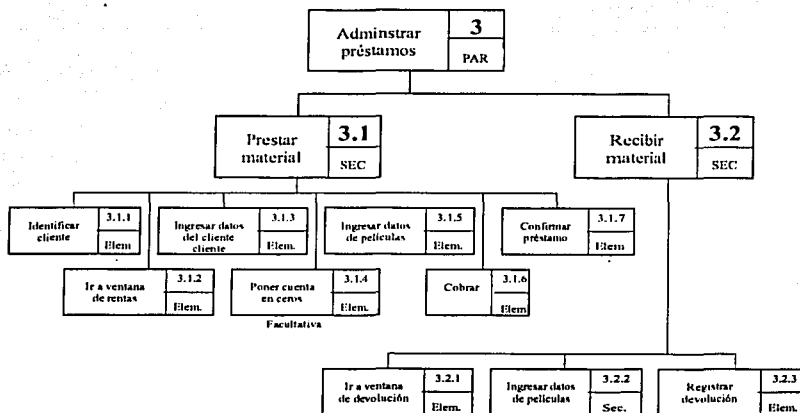


Figura 2.5: Tarea 3 de “Administrar préstamos”.

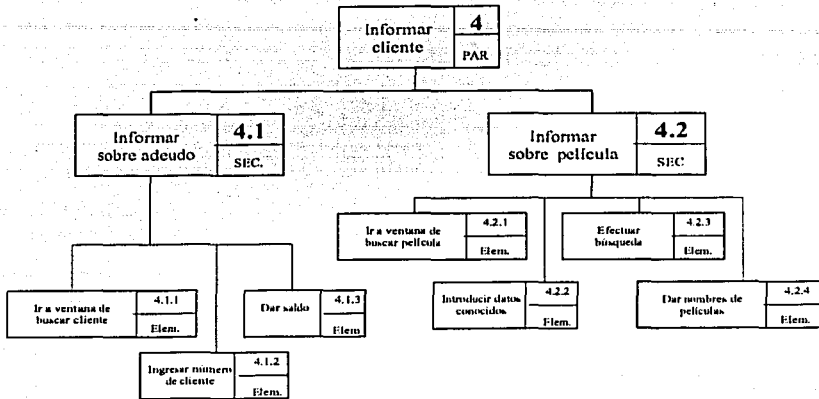


Figura 2.6: Tarea 4 de "Informar cliente".

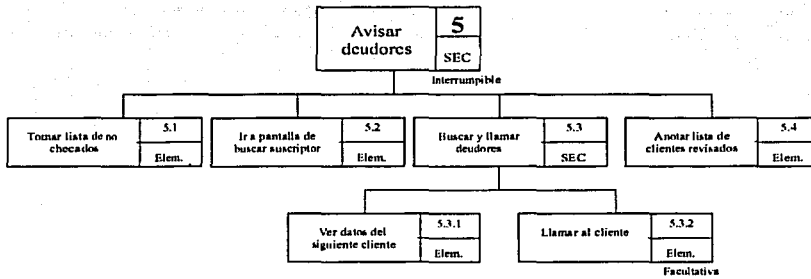


Figura 2.7: Tarea 5 de "Avisar deudores".

El árbol de la tarea modela la descomposición jerárquica de una tarea, así como los flujos en los que ésta puede ejecutarse; y en las fichas de los nodos es donde está la información del objetivo de la tarea, su prioridad, las condiciones que requiere para ejecutarse, el evento que la dispara, y los resultados de su ejecución. Por eso, es durante la elaboración de las fichas cuando el analista puede encontrar numerosas dudas que sólo se resuelven mediante nuevas entrevistas con los usuarios. Esta es una gran ventaja del modelo, pues empuja a los analistas a considerar y cuestionar cada nodo del árbol, por pequeño o simple que este sea. Esto deriva en un amplio conocimiento de la tarea del usuario y un formalismo que lo documenta.

A continuación se mostrarán y explicarán únicamente las fichas de la tarea 1 “Actualizar clientes” y sus subtareas (incluir en este trabajo las casi sesenta fichas del árbol sería ocioso).

En la ficha de la tarea 1 “Actualizar Clientes” está la información necesaria para ejecutar la tarea principal “Atender Mostrador”. Primero, se tienen los datos necesarios para ubicar la tarea en el árbol, como lo son su número (posición en el árbol) y su nombre. A continuación está el constructor, que es de tipo *Alternativo*, lo cual significa que sólo una de sus subtareas será ejecutada (cuál de ellas será dependerá de sus precondiciones y de sus prioridades).

Número	1
Nombre	Actualizar clientes
Constructor	ALT
Objetivo	Mantener el registro de clientes actualizado.
Inicio	El registro de clientes no está actualizado.
Arranque	Llegó un nuevo cliente, se tienen datos actualizados de alguno existente o se le tiene que dar de baja.
Paro	Se efectuaron todas las altas, modificaciones y bajas de clientes que se requerían.
Post-condiciones	El registro de clientes está al día.
Prioridad	5
Interruptible	0
Facultativa	0
Operador	Empleado

La prioridad de esta tarea es de 5, más alta que la de las tareas 2, 4 y 5, e igual a la de la tarea 3; esto se debe a que estas dos tareas son más importantes pues se dan cuando hay un cliente en el mostrador y no se le puede dejar esperando. Esto además puede ser corroborado con su condición de arranque. A continuación el objetivo de la tarea, es “Mantener el registro de clientes actualizado”. Para que esta tarea se pueda ejecutar, es necesario que “El registro de clientes esté al día” (condición de inicio), y se cumpla la condición de arranque: “Llegó un nuevo cliente, se tienen datos actualizados de alguno existente o se le tiene que dar de baja”. Cuando se cumple la condición de paro (“Se efectuaron todas las altas, modificaciones y bajas de clientes que se requerían”) la tarea termina y se tiene la post-condición “El registro de clientes está al día”. Esta tarea no es interruptible ni facultativa, lo cual significa que el empleado tendrá que ejecutarla hasta terminarla y no puede decidir si omitir su ejecución.

Como se puede ver en el árbol, la ejecución de la tarea 1 consiste en la realización de las subtareas 1.1, 1.2 y 1.3, cuyas fichas se muestran a continuación.

TRABAJOS CON
FALLA EN ORIGEN

Número	1.1
Nombre	Registrar cliente
Constructor	SEC
Objetivo	Registrar los datos de un nuevo cliente y efectuar el contrato.
Inicio	Hay clientes no registrados.
Arranque	Llegó un nuevo cliente y este desea asociarse al videoclub.
Paro	Se registraron los datos del cliente y se efectuó el contrato.
Post-condiciones	Se tienen los datos del nuevo cliente.
Prioridad	5
Interruptionable	0
Facultativa	0
Operador	Empleado

Número	1.2
Nombre	Dar de baja a cliente.
Constructor	SEC
Objetivo	Eliminar datos de clientes que ya no son suscriptores.
Inicio	Existen datos de clientes que no son suscriptores.
Arranque	El empleado determina que hay datos de personas que ya no son suscriptores.
Paro	Se eliminaron datos del cliente.
Post-condiciones	El registro del sistema contiene sólo datos de suscriptores.
Prioridad	5
Interruptionable	0
Facultativa	0
Operador	Empleado

REGISTRO
 DE
 OPERACIONES

Número	1.3
Nombre	Modificar cliente.
Constructor	SEC
Objetivo	Actualizar datos que se tienen de un cliente.
Inicio	El sistema tiene datos no actualizados de un cliente.
Arranque	Un cliente pide actualizar sus datos o el empleado tiene datos actualizados sobre algún cliente.
Paro	Los datos del cliente son actuales.
Post-condiciones	El sistema contiene datos actualizados de clientes.
Prioridad	5
Interruptionable	0
Facultativa	0
Operador	Empleado

Como la tarea 1 es de tipo *Alternativa*, sólo una las subtarecas 1.1, 1.2 y 1.3 será ejecutada: Se realizará aquella cuya condición inicial se satisfaga y se dé su condición de arranque; si hay dos tareas que se pueden ejecutar bajo estos términos, el siguiente recurso sería usar sus prioridades para decidir la secuencia de ejecución. Las tres tareas son de tipo *Secuencial*, lo que indica que en su ejecución se realizarán sus tareas en orden estricto: En la tarea 1.1 "Registrar cliente" se ejecutará primero la 1.1.1, luego la 1.1.2, 1.1.3 y finalmente la 1.1.4.

Las descripciones de las subtarecas de las tareas 1.1, 1.2 y 1.3 se muestran a continuación. Es importante notar que todas son de tipo *Elemental* pues ya no se descomponen en más subtarecas; este es el nivel de detalle máximo que se fijó por el grupo de analistas, en el cual se tienen acciones del tipo "Ir a ventana de suscripción", "Entregar credencial", "Ingresar número de cliente", etc.

Número	1.1.1
Nombre	Ir a ventana de suscripción.
Constructor	Elemental
Objetivo	Llegar a la ventana con la forma de inscripción.
Inicio	El sistema está en el menú principal.
Arranque	
Paro	
Post-condiciones	Se tiene la forma de inscripción en la pantalla.
Prioridad	5
Interruptionable	0
Facultativa	0
Operador	Empleado

TESIS CON
FALLA DE ORIGEN

Número	1.1.2
Nombre	Ingresar datos.
Constructor	Elemental
Objetivo	Llenar la forma con los datos del cliente.
Inicio	El sistema no tiene los datos del nuevo cliente.
Arranque	
Paro	
Post-condiciones	El sistema cuenta con todos los datos del cliente.
Prioridad	5
Interrumpible	0
Facultativa	0
Operador	Empleado

Número	1.1.3
Nombre	Celebrar contrato.
Constructor	Elemental
Objetivo	Contar con un contrato suscriptor-video centro.
Inicio	Se tienen los datos del cliente pero no el contrato.
Arranque	
Paro	
Post-condiciones	El sistema tiene los datos del cliente y el empleado el contrato.
Prioridad	5
Interrumpible	0
Facultativa	0
Operador	Empleado

Número	1.1.4
Nombre	Entregar credencial.
Constructor	Elemental
Objetivo	Que el cliente tenga una identificación como suscriptor.
Inicio	Se tienen datos de cliente y contrato, pero el cliente no tiene credencial para rentar.
Arranque	
Paro	
Post-condiciones	El cliente cuenta con credencial de suscriptor.
Prioridad	5
Interrumpible	0
Facultativa	0
Operador	Empleado

TESIS CON
 [Illegible stamp]

Número	1.2.1
Nombre	Ir a ventana de modificar cliente.
Constructor	Elemental.
Objetivo	Llegar a la ventana del sistema que contiene los datos del cliente.
Inicio	El sistema está en el menú principal.
Arranque	
Paro	
Post-condiciones	El sistema está en la ventana de modificar cliente.
Prioridad	5
Interrumpible	0
Facultativa	0
Operador	Empleado

Número	1.2.2
Nombre	Ingresar número de cliente.
Constructor	Elemental.
Objetivo	Tener en la forma los datos del cliente que se desca eliminar.
Inicio	El sistema tiene la forma en blanco.
Arranque	
Paro	
Post-condiciones	La forma contiene los datos del cliente a dar de baja.
Prioridad	5
Interrumpible	0
Facultativa	0
Operador	Empleado

Número	1.2.3
Nombre	Borrar datos.
Constructor	Elemental.
Objetivo	Borrar los datos del cliente.
Inicio	El sistema tiene los datos del cliente.
Arranque	
Paro	
Post-condiciones	El sistema no tiene datos sobre el cliente que se dio de baja.
Prioridad	5
Interrumpible	0
Facultativa	0
Operador	Empleado

TESIS CON
FALLA DE ORIGEN

Número	1.3.1
Nombre	Ir a ventana de modificar cliente.
Constructor	Elemental.
Objetivo	Llegar a la ventana del sistema que contiene los datos del cliente.
Inicio	El sistema está en el menú principal.
Arranque	
Paro	
Post-condiciones	El sistema está en la ventana de modificar cliente.
Prioridad	5
Interruptionable	0
Facultativa	0
Operador	Empleado

Número	1.3.2
Nombre	Ingresar número de cliente.
Constructor	Elemental.
Objetivo	Tener en la forma los datos del cliente que se desca actualizar.
Inicio	El sistema tiene la forma en blanco.
Arranque	
Paro	
Post-condiciones	La forma contiene los datos del cliente con datos no actualizados.
Prioridad	5
Interruptionable	0
Facultativa	0
Operador	Empleado

Número	1.3.3
Nombre	Modificar datos.
Constructor	Elemental.
Objetivo	Reemplazar los datos viejos con los nuevos.
Inicio	El sistema tiene los datos no actualizados.
Arranque	
Paro	
Post-condiciones	El sistema tiene los datos actualizados del cliente.
Prioridad	1
Interruptionable	0
Facultativa	0
Operador	Empleado

Como último paso, se revisan las fichas de las tareas de todo el árbol con el objetivo de encontrar y especificar los objetos del mundo y sus estados que intervienen en las pre y post condiciones. Esta lista de objetos y atributos será usada para poner las pre y post

condiciones de las fichas en un formato que reduzca la posibilidad de que sean interpretadas por los analistas.

- *Class Empleado*
 - *Boolean* Tareas Pendientes
 - *Boolean* Se percató
- *Class Cliente*
 - *Boolean* Registrado
 - *Boolean* Actualizado
 - *Int* Número
 - *String* Datos
 - *Boolean* Credencial
 - *Boolean* Deudor
 - *Int* Monto deuda
 - *Boolean* Informado sobre películas
 - *Boolean* Informado sobre deuda
 - *Boolean* Revisado
 - *Boolean* Atendido
 - *Boolean* Con rentas
 - *Boolean* Con película
 - *Boolean* Conocido
- *Class Deudor*
 - *Boolean* Avisado
- *Class Cobro*
 - *Boolean* Efectuado
- *Class Mostrador*
 - *Boolean* Atendido
- *Class Contrato*
 - *Boolean* Celebrado
- *Class Película*
 - *Boolean* Registrada
 - *Boolean* En préstamo
 - *Int* Clave

TESIS CON
FALLA DE ORIGEN

- *Char* Categoría
- *String* Datos
- *Boolean* Actualizada
- *Class* Registro de clientes
 - *Boolean* Actualizado
- *Class* Inventario de películas
 - *Boolean* Actualizado
- *Class* Registro de préstamos
 - *Boolean* Actualizado
- *Class* Lista de clientes revisados
 - *Boolean* Actualizada
- *Class* Sistema
 - *String* Estado
 - *Object* Forma con datos

Objetos y sus atributos para la tarea del empleado de mostrador.

En las siguientes páginas se muestran algunas de las fichas de los nodos del árbol de la tarea del empleado de mostrador, con sus pre y post condiciones puestos ahora en términos de los objetos que intervienen en la misma.

Número	1
Nombre	Actualizar clientes
Constructor	ALT
Objetivo	Mantener el registro de clientes actualizado.
Inicio	Registro de clientes.Actualizado=false
Arranque	Cliente.Registrado=false OR
Paro	Cliente.Actualizado=false
Post-condiciones	Cliente.Registrado=true AND
Prioridad	5
Interrumpible	0
Facultativa	0
Operador	Empleado

Número	1.1
Nombre	Registrar cliente
Constructor	SEC
Objetivo	Registrar los datos de un nuevo cliente y efectuar el contrato.
Inicio	Cliente.Registrado=false
Arranque	Cliente.Atendido=false AND
Paro	Cliente.Registrado=false
Post-condiciones	Cliente.Registrado=true AND
Prioridad	5
Interrumpible	0
Facultativa	0
Operador	Empleado

Número	1.2
Nombre	Dar de baja a cliente.
Constructor	SEC
Objetivo	Eliminar datos de clientes que ya no son suscriptores.
Inicio	Registro de clientes.Actualizado=false
Arranque	Empleado.Se percató=true
Post-condiciones	Cliente.Registrado=false
Prioridad	5
Interrumpible	0
Facultativa	0
Operador	Empleado

Número	1.3
Nombre	Modificar cliente.
Constructor	SEC
Objetivo	Actualizar datos que se tienen de un cliente.
Inicio	Registro de clientes.Actualizado=false
Arranque	Cliente.Actualizado=false
Paro	Cliente.Actualizado=true
Post-condiciones	Registro de clientes.Actualizado=true
Prioridad	5
Interrumpible	0
Facultativa	0
Operador	Empleado

Así pues, se ha descrito el modelo del Análisis de la Tarea con MAD*, resaltando sus ventajas y capacidades, y se han reafirmado los conceptos con el ejemplo de un caso de uso. En el siguiente Capítulo, ya que se conoce la técnica, se harán los análisis y planteamientos necesarios para determinar las diferencias, y sobre todo, las similitudes entre el AT y el PU que puedan servir como puntos de unión que nos lleven a tener un Proceso de Desarrollo más completo.

TESIS CON
FALLA DE ORIGEN

TESIS
MAD STAR

Capítulo 3

Encontrando el puente entre el AT y el PU

Para integrar el AT al PU se requiere indefectiblemente encontrar un “lenguaje común”, que permita a los usuarios efectuar las actividades del Proceso Unificado y del Análisis de la Tarea indistintamente, dentro de flujos de trabajo iguales a los del PU.

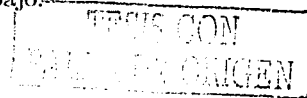
Para lograr esto, es primero necesario comprobar que hay una afinidad de enfoques, de modo que el Análisis de la Tarea pueda efectuarse de forma iterativa e incremental, como lo demanda el PU. En la literatura se plantea al AT como un proceso en cascada, efectuando los pasos establecidos de forma consecutiva hasta terminarlo, por lo que se debe tener cuidado en que las modificaciones necesarias para su integración al PU sean mínimas y que permitan conservar su esencia.

La funcionalidad del nuevo sistema (casos de uso) se obtiene a partir de los requerimientos del usuario, los cuales se encuentran de forma explícita en el AT. Esto sugiere, en primera aproximación, hacer una traducción de las tareas a casos de uso del sistema. Encontrar los detalles de esta similitud entre los elementos centrales de cada notación (Tareas y Casos de Uso), también es parte del lenguaje común que es necesario encontrar para obtener beneficios adicionales a los de la simple realización de cada metodología en paralelo. Finalmente, sería de gran utilidad encontrar alguna técnica que permitiera obtener los casos de uso a partir del árbol de la tarea, y atacar así el problema de la captura de requisitos en el PU; lo cual de antemano sabemos que no será simple, pues es en el análisis de requisitos en donde entra en juego la creatividad para proponer casos de uso que modelen un sistema que cumpla con los requerimientos.

Una vez que se tenga la forma de conciliar los enfoques del Análisis de la Tarea y del Proceso Unificado (sección 3.1), y la relación que tienen sus lenguajes (sec. 3.2), será posible proponer una traducción entre las notaciones (sec. 3.3), para finalmente proponer las actividades, roles y artefactos que deben agregarse al Proceso Unificado (Cap. 4).

3.1. El análisis de la tarea iterativo e incremental

Una de las características más importantes del Proceso Unificado es que trabaja de forma iterativa e incremental, con lo cual se obtiene un desarrollo mejor controlado. Si se piensa en integrar el Análisis de la Tarea a esta forma de trabajo, es importante hacer también el AT de forma iterativa e incremental, para no introducir actividades que rompan con el esquema de ciclos de desarrollo compuestos por flujos de trabajo.



Revisando la literatura que documenta la forma de realizar un Análisis de Tarea ([12]), parecería que es una actividad que se realiza en pasos consecutivos y en un solo ciclo, sugiriendo que no es iterativa e incremental en su versión original (ver sección 1.2.1). Sin embargo, cualquier persona que realice un análisis de la tarea rápidamente se dará cuenta que por el contrario, es totalmente iterativo e incremental, con ciclos de trabajo muy cortos.

En efecto, se comienza realizando una entrevista al usuario, la cual posteriormente es analizada para crear el primer bosquejo del árbol de la tarea; de este primer bosquejo invariablemente surgen dudas que tienen que resolverse en nuevas entrevistas. Trabajando de esta forma —entrevista, análisis, desarrollo del árbol y nuevamente entrevistas— se llega a un árbol de la tarea cuya estructura refleja de manera fiel la percepción del usuario de su tarea.

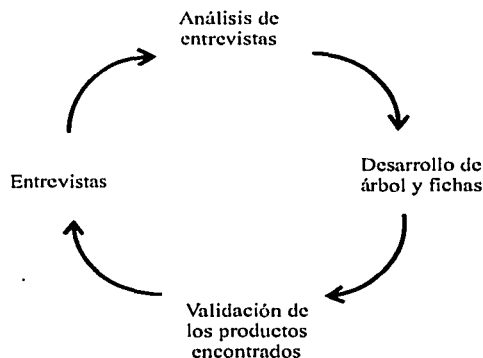


Figura 3.1: Análisis de la Tarea iterativo e incremental.

El siguiente paso consiste en crear las fichas de cada tarea, encontrar los objetos de la tarea y poner las fichas en términos de estos objetos. Nuevamente, al realizar estas actividades nos damos cuenta que surgen dudas y es necesario acudir con el usuario a realizar entrevistas dirigidas que las aclaren, por lo que nuevamente se entra en el ciclo de entrevista, análisis y desarrollo.

Finalmente, cuando se tienen listos todos los productos del AT, es necesario validarlos con el usuario, mediante distintos tipos de pruebas. Estas pruebas arrojan correcciones que dan lugar a nuevas dudas, entrevistas, análisis y desarrollo; por lo que el Análisis de la Tarea siempre se está realizando de forma iterativa e incremental, en un ciclo que se ilustra en la figura 3.1.

Como el Análisis de la Tarea es de hecho iterativo e incremental, podemos proponer integrarlo al ciclo de desarrollo del Proceso Unificado sin mayores modificaciones ni problemas. Para esto, tendremos que identificar las actividades necesarias para realizar el AT, en qué flujos de trabajo deberán ir y quién deberá realizarlas. En este sentido, según el nivel de detalle se pueden proponer pocas actividades, como “Entrevistar”, “Analizar” y “Probar”; o muchas como “Realizar entrevistas no dirigidas”, “Realizar entrevistas dirigidas”, “Diseñar árbol de la tarea”, “Implementar fichas de tareas”, “Encontrar objetos de las tareas”, etc.

Sin embargo, se debe de notar que un ciclo de trabajo para el AT de “Entrevistas -

Análisis - Desarrollo de artefactos - Validación” puede ser bastante rápido, digamos de un par de días¹, por lo que ver el AT como compuesto de muchas actividades puede ocasionar trabajo administrativo para un solo ciclo. Es por esto que en este trabajo hemos optado por descomponer el AT en pocas actividades (figura 3.2) que permitan a una administración más simple; y que su nivel de abstracción permita rápidamente identificar a qué flujo de trabajo del PU pertenecen.

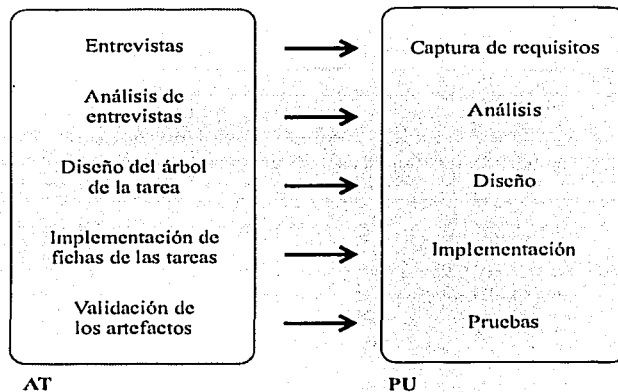


Figura 3.2: Actividades del Análisis de la Tarea propuestas, y flujos de trabajo del Proceso Unificado a los que serán asignadas.

El nivel de abstracción de las actividades propuestas en la figura 3.2 proporciona también flexibilidad al modelo, permitiendo utilizar distintas técnicas de trabajo en cada actividad sin tener que modificarlo, contrario a como sería el caso de muchas actividades con un gran nivel de detalle.

3.2. La relación entre el Árbol de la Tarea y los Casos de Uso

Para integrar el Análisis de la Tarea al Proceso Unificado es necesario recordar qué ventajas nos ofrece cada metodología, para luego proponer una integración que ofrezca ventajas adicionales a la ejecución en paralelo de ambas. Recapitulando, la actividad de “Captura de requisitos” del PU determina la funcionalidad que debe tener el nuevo sistema y la pone en términos de casos de uso; mientras que el AT con MAD* modela la actividad del usuario en términos de árboles de tareas. Luego, hay una diferencia sutil pero con consecuencias importantes:

Los casos de uso especifican cómo se efectuará la tarea en el sistema, mientras que el árbol MAD, y sus fichas, especifican cómo se hace actualmente (sin la*

¹Dos horas de entrevista, cuatro horas para escucharla y analizarla, cuatro horas más para hacer cambios en el árbol y elaborar nuevas fichas, obteniendo dudas y preguntas para la próxima entrevista.

nueva aplicación).

Inmediatamente surge la pregunta de si será posible prescindir de la semántica y notación de alguna de las dos metodologías dentro del nuevo proceso integrado, pero hay diferencias importantes, como se explica en los siguientes párrafos, que impiden que podamos sustituir el árbol de la tarea y sus fichas por casos de uso, y viceversa, las tareas por casos de uso.

En primer lugar, no es recomendable sustituir el árbol de tareas y sus fichas por casos de uso ya que éstos carecen de la notación y abstracción necesaria para especificar la forma en que se ejecuta una tarea (las fichas MAD* son las que contienen toda esta información). Además, el árbol MAD* puede ser usado como una herramienta para control de cambios en los requerimientos. Si se recuerda, en los procesos de desarrollo de software se capturan los requisitos mediante entrevistas, a partir de las cuales se divide el trabajo en ciclos. Sin embargo la interpretación que se da a los requerimientos puede cambiar a lo largo de los ciclos de desarrollo, y la funcionalidad del sistema puede alejarse considerablemente de lo que espera el cliente y los usuarios. Si se tiene el árbol MAD* se puede emplear para cuidar que el sistema que se está desarrollando cumpla con las expectativas planteadas en las entrevistas.

Por otro lado, el árbol de tareas y sus fichas no pueden sustituir a los casos de uso del Proceso de Desarrollo Unificado, ya que no capturan las relaciones de generalización, particularización, extensión, inclusión, etc. que hay entre las distintas funcionalidades que tendrá el sistema.

El segundo aspecto que es necesario considerar con respecto a prescindir de la semántica y notación de alguna de las metodologías es que no es claro cómo traducir el árbol de tareas a un modelo similar al de casos de uso, que permita efectuar la división de trabajo en ciclos de desarrollo. La división del árbol en subtareas depende de cómo el usuario percibe su tarea, y no refleja como éstas pueden ser agrupadas por funcionalidad. Por ejemplo, puede haber tareas a distintos niveles de abstracción y en diferentes ramas del árbol que se realizan con el mismo procedimiento. Éstas deberían desarrollarse en el mismo ciclo, pero el modelo MAD* no lo indica.

De lo anterior, se concluye que no se debe prescindir ni de los casos de uso ni del árbol de tareas. Esto indica que en la integración de los dos enfoques se tendrá que encontrar la manera de relacionar la Tarea con los Casos de Uso. La forma de hacerlo no es evidente, ya que:

- En un nuevo sistema no todas las subtareas del árbol de la tarea del usuario se incluyen, debido a la naturaleza o complejidad de las mismas, como por ejemplo tareas manuales que no se pueden llevar a la computadora (“Contestar teléfono”);
- En el análisis y diseño del nuevo sistema, un caso de uso puede realizar más de una subtarea de la tarea del usuario (El caso de uso “Enviar mensajes” del sistema puede efectuar “Enviar fax” y “Enviar correo electrónico” de la tarea del usuario);
- En el nuevo sistema aparecen casos de uso que no se realizan en ninguna tarea del árbol MAD* del usuario (ej. “Salir del sistema”, “Validar datos”).

Para responder a estas dudas es necesario considerar, en primer lugar, que el Árbol de la Tarea y el Caso de Uso tienen significado distinto y por tanto debemos emplearlos para distintos objetivos:

- El gran contenido semántico que tiene un árbol de la tarea junto con sus fichas, sugiere emplearlo para modelar esquemas complejos donde se requiere un gran control sobre la

secuencia de eventos y cómo éstos afectan a los objetos del mundo. Al modelar la tarea del usuario también se captura gran parte de la información referente a la lógica del negocio, sus escenarios, y la interfaz del sistema que emplea el usuario de una forma mucho más rica que si se emplearan los casos de uso. Aparte de las ventajas que nos da la notación MAD*, con el AT tenemos una forma procedural de efectuar el análisis, a diferencia del PU, en el cual no se especifica cómo se obtendrán los productos de sus actividades.

- Por otro lado, los casos de uso proveen de toda la abstracción necesaria para modelar el nuevo sistema desde un punto de vista orientado a objetos, pues tienen la ya muy probada y trabajada notación de UML.

Así, lo que se busca es efectuar el flujo de trabajo de "Captura de requisitos" con ayuda del Análisis de la Tarea, y traducir los resultados en diagramas de casos de uso de modo que dicha información pueda ser utilizada de manera eficiente en los otros flujos de trabajo. Para poder realizar esto se requiere comprender la relación que guardan los casos de uso y la tarea del usuario, por lo cual se compararán sus definiciones y se propondrá una relación entre ambos esquemas.

- **Definición de la Tarea y su árbol**

Remontándose a los primeros planteamientos acerca de la tarea y su representación, encontramos que Fikes ([22], 1981) sugiere modelarla utilizando el paradigma de planificación jerárquica establecido por Sacerdoti ([23]), en el que cada etapa particular se considera como una tarea con sus propias entradas, condiciones, objetivos, etc. Así, cada esquema de planificación representa el desarrollo de la acción. Las representaciones jerárquicas además permiten establecer jerarquías en dicho desarrollo: se tienen en la parte superior una simplificación o una abstracción del plan de trabajo, y en el nivel más bajo el detalle de dicho plan. Posteriormente Sperandio ([24], 1988) define la tarea como un conjunto de condiciones de ejecución de un trabajo determinado, para un objetivo dado y según un conjunto de exigencias dadas. En 1994 Sebillote ([25]) define la tarea como el conjunto de objetivos que los operadores buscan alcanzar, así como los procedimientos que les permiten hacerlo.

La definición que reúne los conceptos anteriores es la de Alonso ([14], 1996), en donde una tarea se describe de manera genérica por un objetivo (el qué), y una lista de procedimientos (el cómo). Según la autora, la tarea es la entidad que permite agrupar al mismo tiempo los conocimientos tanto declarativos (el objetivo y el dominio) como procedurales (su desarrollo en términos de acciones concretas), relativas a un objetivo establecido.

- **Definición de un Caso de Uso**

Por otro lado, en UML ([27], pág. 488), se define a un caso de uso como una unidad coherente de funcionalidad externa visible que provee un sistema, y se expresa por secuencias de mensajes intercambiados entre el sistema y uno o más actores² del sistema. También hacen hincapié en que la definición incluye al comportamiento del sistema, tal como las secuencias principales, variaciones del comportamiento normal y la respuesta descada. Posteriormente, en el PU ([3], pág. 39), los autores replantean

²UML define a un actor como una idealización de una persona externa, proceso u otra entidad que interactúa con el sistema.

a un caso de uso como una descripción de un conjunto de secuencias de acciones, incluyendo variaciones, que un sistema lleva a cabo y que conduce a un resultado observable de interés para un actor determinado.

Analizando las definiciones anteriores encontramos que, mientras que la tarea modela los procedimientos que tiene que realizar una persona para llegar a su objetivo, los casos de uso modelan la secuencia de acciones que un sistema lleva a cabo para proporcionar el resultado deseado.

En el marco de interacción humano-computadora, estas definiciones tienen una equivalencia en el sentido de que las funciones del sistema deben de apoyar al usuario en la realización de su tarea:

La funcionalidad del sistema, expresada en casos de uso, se obtiene como consecuencia de los servicios que el sistema tiene que proporcionar al usuario, para que éste pueda realizar su tarea.

Esto es, cada caso de uso del sistema justifica su presencia si apoya en la realización de alguna tarea del usuario.

En base a lo anterior, una primera aproximación para obtener los casos de uso del AT, sería obtener una primera versión de los diagramas de casos de uso a partir del árbol de tareas, transformando cada subtarea del árbol en un caso de uso que lo realice. Incluso se podría usar la misma jerarquía del árbol de tareas en los casos de uso para dividirlos en casos y sub-casos de uso. El trabajo restante consistiría en procesar y re-estructurar estos diagramas, para encontrar las relaciones y factorizar los casos de uso.

En la siguiente sección, se hace el planteamiento de la parte más importante de la aportación: Cómo establecer el puente entre el AT y el PU, de tal forma que podamos mapear el árbol de la tarea a diagramas de casos de uso. Posteriormente, se propondrán técnicas que permitan generar los demás artefactos del PU con la información del modelo MAD* de la tarea y otros elementos como los *objetivos fundamentales del usuario*. Los objetivos fundamentales del usuario implican un valor agregado a la propuesta, pero su identificación puede resultar compleja. A continuación se explica en qué consisten.

Los Objetivos Fundamentales y la Tarea del Usuario

La tarea del usuario tiene generalmente una afinidad con los propósitos de la empresa, negocio o entidad en la que éste labora. Incluso, esta es la razón por la que la tarea del usuario y los procesos del negocio del cliente pueden parecer tan similares, y ser confundidos por los miembros del equipo de desarrollo.

La relación entre el usuario y su negocio consiste en que hay objetivos fundamentales que el usuario debe de alcanzar mediante la realización de su tarea para lograr las metas del negocio; las actividades del usuario no tendrían cabida en el negocio si no apoyaran estos objetivos. Así, los objetivos fundamentales para la tarea del empleado de mostrador del capítulo 2 podrían ser, por ejemplo, "prestar películas", "cobrar por el servicio" y "recuperar el material".

Como lo indica D. Norman [10], cualquier interacción entre un hombre y una herramienta tiene varios niveles de abstracción: El léxico, el sintáctico, el semántico y el de objetivo. El nivel de objetivo se refiere a los objetivos fundamentales; el nivel semántico contiene el conocimiento del usuario sobre cómo hacer su tarea; el nivel sintáctico encierra el conocimiento de cómo usar la herramienta para resolver un problema; y el nivel léxico trata sobre qué acciones precisas se requieren ejecutar para realizar una tarea con la herramienta.

Como ejemplo, para una interacción humano-computadora podemos tener que en el nivel de objetivo fundamental se tiene "Enviar carta a clientes distinguidos"; el nivel semántico consiste de "qué es mandar", "qué es carta", "qué es cliente", "qué es cliente distinguido"; el nivel sintáctico comprende cómo se puede usar la computadora para apoyar el objetivo fundamental, y el nivel léxico son las acciones paso a paso: "Abrir word", "Crear nuevo documento", etc.

El Análisis de la Tarea captura la información de los niveles léxico, sintáctico, y quizás una parte del semántico, pero no captura el objetivo fundamental del usuario.

El conocimiento de los objetivos fundamentales de la tarea del usuario puede ser vital para un desarrollo: Si el planteamiento del nuevo sistema no apoya al usuario en la realización de sus objetivos fundamentales, probablemente tampoco cumpla con los requerimientos del cliente y del usuario.

En este trabajo, el conocimiento de los objetivos fundamentales implica un valor agregado al proceso de desarrollo, pues permitirá por un lado asignar, en base a su relación con estos objetivos, niveles de importancia a las tareas del usuario; y por otro clasificar las subtareas del árbol de acuerdo al objetivo fundamental que apoyan.

Sin embargo, la obtención de éstos puede ser algo no trivial, pues requiere de un verdadero conocimiento de la tarea del usuario y del negocio. Los objetivos fundamentales pueden estar fuertemente vinculados a la misión y metas del negocio del cliente, cuyo análisis está fuera del alcance de este trabajo.

En este aspecto, el enfoque será el de no considerar a los objetivos fundamentales como una parte esencial de la propuesta de integración, para evitar las dificultades que se pueden encontrar al introducirlo como una actividad obligatoria.

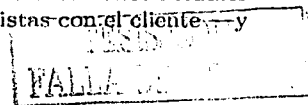
En algunas actividades, el conocimiento de éstos podrá tener beneficios interesantes, como la posibilidad de asociar cada caso de uso con el objetivo fundamental que apoyará, como se verá más adelante.

3.3. Paso del Árbol de la Tarea a Casos de Uso

Ya se tienen las principales actividades del Análisis de la Tarea identificadas, así como los flujos de trabajo en los que irán. Lo que falta, es una parte muy importante de la integración del AT al PU, y es cómo se obtendrán los casos de uso a partir del análisis de la tarea efectuado. Esta parte medular será tratada en las dos subsecciones que siguen.

En primera aproximación, se buscó un proceso con el cual se transformara el Árbol de la Tarea en Casos de Uso; sin embargo, pronto nos dimos cuenta que esto no es posible: El árbol de la tarea modela la percepción actual del usuario, sin el nuevo sistema. En la gran mayoría de los desarrollos, se busca construir un sistema que mejore en algún sentido las actividades que realiza el usuario, lo cual implica necesariamente un cambio en su tarea, que puede consistir en la automatización o modificación de sus actividades, para que se realicen de forma más sencilla, rápida y/o segura. Luego, el Análisis de la Tarea no nos puede dar la funcionalidad exacta del nuevo sistema —pues modela al viejo—, pero contiene toda la información del usuario, su tarea y su entorno, y en particular las actividades que son de importancia para él.

Sin embargo, es posible proponer los casos de uso de forma fundamentada, basándose en el contenido del árbol de la tarea del usuario, en contraste con muchos desarrollos actuales en los cuales se propone la funcionalidad a partir de un par de entrevistas con el cliente— y no con el usuario— y a partir de meras suposiciones.



Debemos de notar que no hay un método automatizado para obtener los casos de uso a partir del árbol; para esto se requiere el análisis y comprensión del problema, así como una buena dosis de creatividad del grupo de desarrollo. Lo que sí se puede hacer es procesar el árbol de la tarea y generar *casos de uso preliminares*, con los cuales los analistas de sistemas puedan generar los casos de uso del nuevo sistema de forma más rápida y sencilla.

3.3.1. El árbol de la tarea modificada

El primer paso para obtener “casos de uso preliminares” a partir del árbol de la tarea y sus fichas, consiste en seleccionar la información del AT que es de utilidad, para obtener un nuevo árbol de tarea que contenga sólo la información necesaria para la construcción de los casos de uso. A este árbol le llamaremos *árbol de la tarea modificada*. Se considera que este es un nombre apropiado pues se obtiene un árbol de tarea que no es ni de la tarea actual del usuario ni de su tarea con el nuevo sistema, sino un árbol que servirá de intermediario entre el árbol original y los casos de uso preliminares.

La forma de construir el árbol de la tarea modificada consistirá en eliminar del árbol de la tarea del usuario la información que no es de utilidad para nuestro propósito y modificar las tareas restantes de acuerdo al conocimiento adquirido del problema, los alcances del proyecto de desarrollo, requerimientos adicionales, etc.

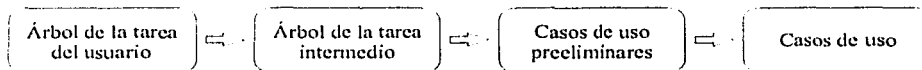


Figura 3.3: Pasos para elaborar los casos de uso a partir del árbol de la tarea del usuario; en cada uno de ellos interviene la experiencia y creatividad del analista.

A priori, en el árbol de la tarea modificada se pueden eliminar todas las tareas que estén relacionadas con el uso de la interfaz del sistema anterior u otras herramientas, pues estas acciones —tales como “dar click en abrir archivo nuevo”— no representan casos de uso. El hecho de eliminarlas del árbol de la tarea modificada no quiere decir que la interfaz del nuevo sistema no tomará en cuenta esta interacción para dar lugar a un diseño de interfaz centrado en el usuario; el diseño de la interfaz del nuevo sistema debe ser a partir del árbol de la tarea del usuario y es hasta cierto punto independiente de los casos de uso.

Todas las tareas que queden en el árbol de la tarea modificada son candidatas para convertirse en casos de uso del nuevo sistema. Ahora es necesario evaluar, en base al criterio del grupo de desarrollo, cuáles tareas se conservarán y pasarán a los casos de uso propuestos, y cuáles serán descartadas. Para esto, se propone el siguiente criterio:

Una tarea del árbol de la tarea del usuario es descartada del árbol de la tarea modificada si:

- Ya no se realiza debido a la introducción del nuevo sistema (ej. “Abrir archivero”, “Buscar carpeta”, “Depositar sobre”, etc.).
- No se incluirá en el nuevo sistema debido al alcance del proyecto (ej. “Clasificar correo”. “Efectuar cobro en línea”, “Búsqueda avanzada”, etc.).

- No se puede realizar con el sistema (ej. "Contestar el teléfono", "Abrir puerta").

Aplicando el criterio anterior se obtiene un árbol de la tarea modificada que contiene primordialmente tareas cuya ejecución será apoyada por el nuevo sistema, y que son candidatas a convertirse en casos de uso para el proceso de desarrollo.

3.3.2. Casos de uso preliminares

El árbol de la tarea modificada contiene de la tarea del usuario sólo las actividades que serán apoyadas por el nuevo sistema. Cada nodo de este árbol se convertirá en un caso de uso preliminar: El "nombre" de la tarea será el nombre del caso de uso, y el "objetivo" y el "papel del usuario" servirán como descripción del caso de uso (ver figura 3.4); otros atributos de la tarea, como el "tipo", el "modo", la "importancia" y el "operador" pueden también incluirse en la descripción del caso de uso para proporcionar información adicional.

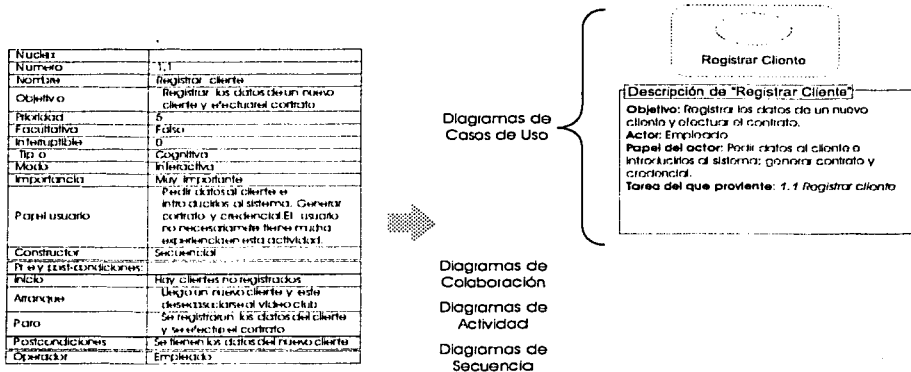


Figura 3.4: Conversión de una tarea a un caso de uso.

La información que captura el modelo MAD* de la tarea del usuario y la interacción de éste con su medio es tan detallada, que inclusive se pueden obtener los diagramas de colaboración, de secuencia y de actividad de UML para modelar el aspecto dinámico del comportamiento del sistema y su interacción con el usuario. Si bien las fichas de las tareas modelan la interacción del usuario con su ambiente actual y no con el nuevo sistema, es posible, al igual que se hizo para la relación tarea - caso de uso, mapear estos objetos a interfaces del sistema que den lugar a especificaciones del comportamiento dinámico del usuario con el nuevo sistema.

La simple traducción de las tareas del árbol de la tarea modificada daría como resultado una gran lista con todos los casos de uso propuestos, lo cual, si bien es un avance en la obtención de casos de uso, requiere todavía adicionar una cantidad de trabajo para obtener diagramas en donde se indiquen las asociaciones entre los casos de uso. Para obtener una buena parte de estas asociaciones a partir de la información adicional que proveen las tareas,

se requiere de plantear y probar técnicas multidisciplinarias que den lugar a diagramas de casos de uso detallados, lo cual constituye en sí otro trabajo.

Es por eso que como alternativa se propone a continuación una heurística para determinar si hay asociaciones entre los casos de uso, a partir de la información que se puede obtener directamente de las fichas de sus tareas. Ésta no nos dará información detallada, como por ejemplo el tipo de asociación (de generalización, de extensión, de inclusión, etc.), pero permitirá obtener diagramas preliminares en vez de listados de casos de uso.

Si se revisa la estructura del árbol y las fichas MAD*, se encontrará que éstos capturan:

- Los objetos que afecta cada tarea.
- Las acciones que realiza cada tarea.
- El objetivo fundamental que cada tarea apoya.
- La descomposición jerárquica de la tarea del usuario.

Esta información provee, de forma implícita, las relaciones de asociación entre los casos de uso, pues permite agruparlos con un criterio y establecer asociaciones de grupo. Así, por ejemplo, si se agrupan los casos de uso de acuerdo a los objetos que afectan, dos casos de uso estarán asociados si están en el mismo grupo, o lo que es equivalente, si afectan al mismo objeto.

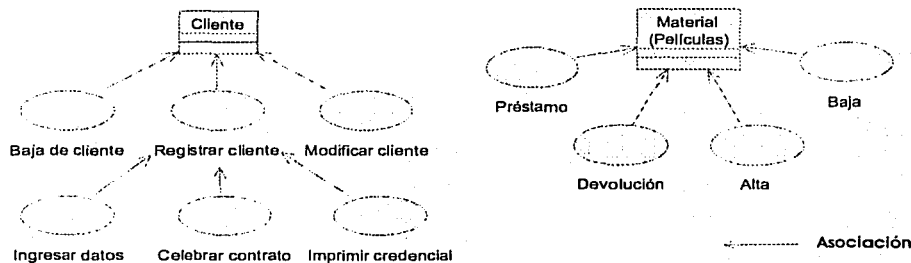


Figura 3.5: Ejemplo de agrupación de casos de uso preliminares alrededor de los objetos que afectan en sus fichas. En este caso, se utilizó la jerarquía de las tareas en el árbol para obtener niveles de detalle para cada grupo.

Aunque la heurística da lugar a interpretación por parte de los analistas, constituye una forma rápida de obtener varios diagramas de casos de uso preliminares candidatos a ser refinados; además, una parte importante (los casos de uso) está basada en información capturada de forma analítica y procedural (el árbol MAD* y sus tareas).

Así pues, recapitulando, el árbol de la tarea modificada contiene la información necesaria para la creación de los casos de uso preliminares; cada tarea (nodo) del árbol se convertirá en un caso de uso, y las asociaciones entre ellos se obtendrán de las propiedades de las tareas. Una vez obtenido el diagrama de los casos de uso preliminares, restaría determinar, para cada asociación entre dos casos de uso, si ésta es de generalización, extensión, inclusión, comportamiento o realización.

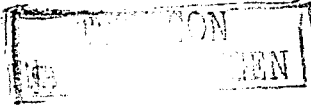
3.4. Resumen

Los enfoques del Análisis de la Tarea y del Proceso Unificado tienen, contrario a lo supuesto inicialmente, varios puntos de cohesión que permiten presuponer una integración del AT al PU concisa y consistente. El AT y el PU tienen formas de trabajo similares, e inclusive flujos de trabajo equivalentes.

La integración del Análisis de la Tarea al Proceso Unificado de Desarrollo implica la introducción de una nueva figura: La tarea modificada. Ésta sólo contiene las tareas que son de interés para el desarrollo de los casos de uso preliminares, pues la tarea del usuario puede contener información que no es de utilidad para este fin, como la interacción del usuario con sistemas previos, o sub-tareas que por el alcance del proyecto no es posible implementar en el nuevo sistema.

El punto de unión entre las dos metodologías es ahora claro: El usuario y su sistema. Mientras que el AT estudia la interacción que tiene el usuario con su medio y en particular con sus herramientas durante la realización de su tarea, el PU busca el desarrollo de sistemas de software que apoyen al usuario en su tarea. Luego, *la funcionalidad del sistema, expresada en casos de uso, se obtiene como consecuencia de los servicios que el sistema tiene que proporcionar al usuario, para que éste pueda realizar su tarea.* La oración anterior es de importancia pues establece una relación entre la tarea y el caso de uso, que permite tener una primera versión de los casos de uso a partir del análisis de la tarea realizado. Y no sólo eso, la descripción y detalle de los casos de uso puede también obtenerse del AT.

TESIS CON
FALLA DE ORIGEN



Capítulo 4

Integración del Análisis de la Tarea del Usuario al Proceso Unificado

En este capítulo se propondrán los cambios necesarios para integrar el Análisis de la Tarea al Proceso Unificado de Desarrollo de Software, considerando el uso de los productos del AT como una fuente de información para actividades del PU.

En resumen, la integración consiste de la adición de un nuevo rol en el equipo de desarrollo que realizará las actividades del análisis de la tarea y de su adecuación al Proceso; se propondrá también cambios a varias actividades de la “Captura de requisitos” para que éstas se realicen de forma más sencilla a partir de los resultados del AT.

La forma de presentar la propuesta será tal y como el PU ([3]) describe a sus flujos de trabajo fundamentales. La propuesta de integración sólo agrega roles, actividades y productos al flujo de “Captura de requisitos”, por lo que el enfoque será describir las adiciones y cambios a este flujo.

Es importante enfatizar que sólo se describirán las adiciones y modificaciones al PU; todos los roles, artefactos y actividades que no se describan se conservarán tal y como están en el PU.

4.1. Descripción rápida

La integración del Análisis de la Tarea al Proceso Unificado apoya la mayoría de las actividades del flujo de trabajo de “Captura de Requisitos”, basándose en dos pasos fundamentales:

1. Realización del Análisis de la Tarea
2. Obtención del árbol de la tarea modificada, a partir del AT.

A partir de los productos de estas dos actividades es que se pueden generar diagramas preliminares de casos de uso, descripciones de los casos de uso, diagramas de actividad, prototipos de interfaz, el glosario de términos, etc.

Tomando como referencia la figura 7.1 del Proceso Unificado de Desarrollo de Software ([3], pág. 127), se presenta a continuación la figura que ilustra los trabajadores y artefactos para el flujo de “Captura de Requisitos” que integra el AT:

En la figura se puede destacar la adición de un nuevo rol —“Analista de la tarea”— y tres nuevos artefactos, resultado del AT y de su integración. Estos nuevos artefactos son

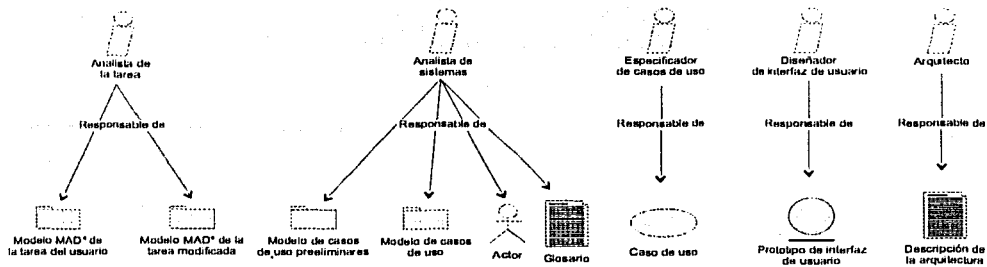


Figura 4.1: Trabajadores y artefactos para el flujo de "Captura de Requisitos" que integra al Análisis de la Tarea.

resultado de tres nuevas actividades, que serán descritas más adelante en la sección 4.4 "Nuevas actividades".

A continuación se describen los nuevos artefactos y roles involucrados en la integración del AT al PU.

4.2. Nuevos artefactos

4.2.1. Modelo MAD* de la tarea del usuario

El modelo de la tarea del usuario está basado en el formalismo MAD* del Análisis de la Tarea, y consiste de dos partes: el árbol de la tarea y las fichas de las tareas. Como antes se ha explicado (Capítulo 2), MAD* permite capturar los aspectos de orden y de estado del mundo de la tarea del usuario, mediante el uso de una representación jerárquica de la tarea (árbol de la tarea) y fichas descriptivas de cada nodo.

La estructura del *árbol de la tarea* es la siguiente (figura 4.2):

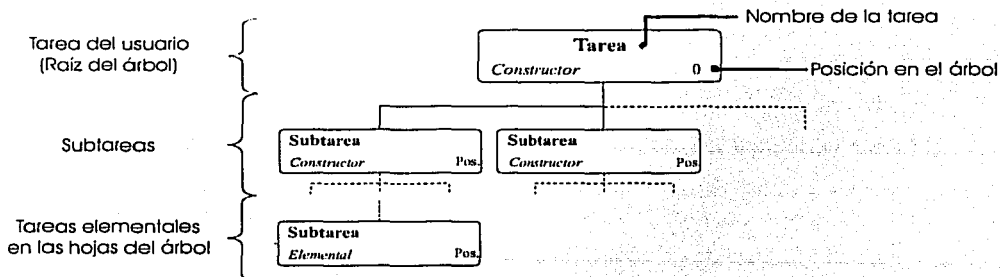


Figura 4.2: Estructura del árbol de la tarea del usuario.

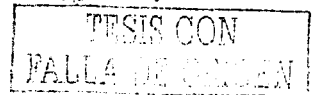
- La raíz del árbol representa a la tarea del usuario.
- Cada nodo en el árbol tiene:
 1. Nombre que contiene un verbo
 2. Constructor
 3. Número que indica posición
 4. Ficha que lo describe
- Las hojas del árbol se llaman "Tareas elementales", y no requieren de más descomposición para entenderse.

El árbol de la tarea provee, en una sola figura, toda la información referente a la secuencia y estructura de la tarea del usuario, mientras que los datos de estado del mundo, antes y después de la ejecución de la tarea y sus subtareas, se encuentran en la fichas que las describen.

Las fichas detallan el comportamiento de las tareas, especificando el objetivo, la persona que las realiza, cuándo se realizan, qué requieren, etc. Cada ficha se puede ver formada por dos partes: la primera, el *núcleo*, que describe la tarea, su importancia, cómo y por quién es ejecutada; y la segunda, las *condiciones*, que establecen en qué momento es ejecutada, en qué condiciones, y cuándo ésta concluye. Numerosos ejemplos y explicaciones se pueden encontrar en el Capítulo 2.

Los campos contenidos en cada ficha son los siguientes:

- En el núcleo:
 - Número: Numérico, representando la posición de la tarea en el árbol.
 - Nombre: Alfanumérico, nombre de la tarea.
 - Objetivo: Alfanumérico, descripción del objetivo de esa tarea particular.
 - Prioridad: Entero, permite resolver conflictos entre tareas que quieren ejecutarse simultáneamente.
 - Facultativa: Booleano, indicando si la tarea es obligatoria o se puede omitir su ejecución.
 - Interruptible: Entero. 0 indica no interruptible, 1 interruptible con posibilidad de continuar la tarea, 2 indica interruptible y al regresar se reinicia la tarea, y 3 indica interruptible y no se regresa a la tarea.
 - Tipo: Alfanumérico, indicando si es sensorimotriz o cognitiva.
 - Modo: Alfanumérico, indicando si es manual, automática o interactiva.
 - Importancia: Alfanumérico, describiendo la importancia de la tarea.
 - Papel del usuario: Alfanumérico. Descripción del papel del usuario en la tarea, así como su experiencia, preferencias, limitaciones, etc.
 - Constructor, uno de los siete siguientes:
 - Secuencial: Las subtareas se ejecutan en el orden especificado. La ejecución termina al haber ejecutado todas las subtareas obligatorias, y las optativas han sido ejecutadas o han sido omitidas.



- **Paralelo:** Las subtareas se ejecutan en cualquier orden y sin compartir datos. La ejecución termina al haber ejecutado todas las subtareas obligatorias, y las optativas han sido ejecutadas o han sido omitidas.
 - **Simultáneo:** Las subtareas se ejecutan de forma simultánea y por distintos usuarios o entidades. La ejecución termina al haberse ejecutado todas las subtareas obligatorias, y las optativas han sido ejecutadas o han sido omitidas.
 - **Y:** Las subtareas se ejecutan en cualquier orden y compartiendo datos. La ejecución termina al haberse ejecutado todas las subtareas obligatorias, y las optativas han sido ejecutadas o han sido omitidas.
 - **O:** Las subtareas se ejecutan en cualquier orden y comparten datos. La ejecución termina cuando al menos una de las subtareas obligatorias se ha completado.
 - **Alternativa:** Una y sólo una de las subtareas es ejecutada.
 - **Elemental:** La tarea no tiene subtareas. Este constructor indica que el nodo es una tarea simple o una acción que se ejecuta directamente.
- En las condiciones, todas las siguientes son requeridas, a excepción de las marcadas como opcionales:
 - **Estado del mundo:** Objetos del mundo empleados en la tarea y de los cuales dependen las pre- y post-condiciones.
 - **Pre-condiciones:**
 - **Condición de inicio:** Estado del mundo requerido para que se puede ejecutar la tarea.
 - **Condición de arranque (opcional):** Estado del mundo que demanda la ejecución de la tarea. El estado de inicio debe de cumplirse para que la tarea arranque.
 - **Condición de paro (opcional):** Estado del mundo que indica cuándo detener la ejecución de la tarea (y de sus subtareas). Este atributo es útil si se quiere que la ejecución de la tarea se repita hasta lograr una condición.
 - **Post-condición:** Estado del mundo después de que la tarea terminó su ejecución.

Esta información puede ser presentada en varios formatos, como el siguiente:

<i>Núcleo:</i>	
Número	
Nombre	
Objetivo	
Prioridad	
Facultativa	
Interrumpible	
Tipo	
Modo	
Importancia	
Papel usuario	
Constructor	
Operador	
<i>Pre y post-condiciones:</i>	
Inicio	
Arranque	
Paro	
Postcondiciones	

4.2.2. Modelo MAD* de la tarea modificada

El modelo MAD* de la tarea modificada se documenta de la misma forma que la tarea del usuario, mas sin embargo, tiene un significado semántico distinto, pues ya no representa la forma en que el usuario realiza su tarea actualmente; la tarea modificada captura las actividades de la tarea del usuario que el analista de la tarea quiere apoyar con el nuevo sistema. Por ese motivo, el árbol no contiene tareas que:

- Se relacionan con la interfaz del viejo sistema o herramienta.
- Ya no se realizan debido a la introducción del nuevo sistema.
- Quedan fuera del alcance del proyecto.
- No se pueden realizar con apoyo del sistema

4.2.3. Modelo de casos de uso preliminares

El modelo de casos de uso preliminares es el mismo que el modelo para los casos de uso de UML. Las relaciones entre los casos de uso son sólo de asociación, ya que no es posible obtener más información de la relación de la tarea del usuario de forma directa. Esta característica permite crear de forma rápida varios diagramas preliminares de casos de uso, de manera que el analista de sistemas pueda escoger el más adecuado como punto de partida para el modelo de casos de uso definitivo.

4.2.4. Documentación de las entrevistas

La documentación de cada entrevista a los usuarios es de gran importancia en el análisis de la tarea, pues el analista de la tarea requiere frecuentemente revisar repetidamente la entrevista para llegar a un modelo de la tarea consistente.

Lo recomendable es grabar en audio o en video cada entrevista, y conservar las cintas como parte de la documentación del proyecto. Otra documentación de las entrevistas que también es importante conservar es:

- Protocolo de introducción, en donde se explica claramente el propósito de la entrevista, su duración, el tipo de tareas a observar, qué materiales se requerirá que proporcione el entrevistado, etc.
- Cuestionarios demográficos que se les da a los entrevistados para clasificarlos.
- Notas tomadas por el entrevistador.
- Cartas de confidencialidad.

Se pueden consultar ejemplos y guías de estos documentos en el libro de J. T. Hackos [2], páginas 200-215.

4.3. Nuevo trabajador: Analista de la tarea

El analista de la tarea es responsable de llevar a cabo el análisis MAD* de la tarea del usuario y luego proponer el modelo MAD* de la tarea modificada. Esto implica una participación muy importante del analista de la tarea, pues lo involucra en la planeación, coordinación y realización de las entrevistas a los usuarios. También decidirá, con apoyo del analista de sistemas, el arquitecto de sistemas y los administradores del proyecto, qué tareas quedarán en el árbol de la tarea modificada y que por tanto, son candidatas a implementarse en el nuevo sistema.

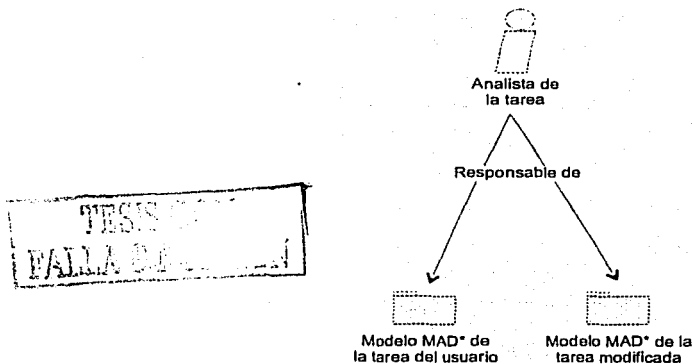


Figura 4.3: Responsabilidades del analista de la tarea durante el flujo de trabajo de captura de requisitos.

El analista de la tarea tiene que tomar decisiones sobre (ver páginas 155-192 de J. T. Hackos [2]):

- Objetivos y dudas a tratar en las entrevistas: Aparte de la realización del análisis de la tarea, qué objetivos es necesario cumplir, y qué preguntas es necesario responder, basándose en la información que ya tiene el grupo de desarrollo, el cliente y sus usuarios sobre el viejo sistema y la problemática del negocio.
- Entrevistados: ¿Qué usuarios serán entrevistados?. Es importante notar que, por razones de tiempo y presupuesto, no es posible entrevistar a todos los usuarios. Se tiene que decidir a cuántos participantes es posible entrevistar, cuáles son los más representativos, etc.
- Lugar de la entrevista: ¿En dónde se van a realizar las entrevistas?, ¿Se requiere tramitar el préstamo de las instalaciones?.
- Agenda de las entrevistas: ¿Cuándo se realizarán las entrevistas?, ¿Están los usuarios disponibles?, ¿Cuánto tiempo durarán las entrevistas?.
- Entrevistadores: ¿Quiénes serán los que realizarán las entrevistas?, ¿Requieren de algún tipo de capacitación?.
- Materiales que se usaran para las entrevistas.
- Contenido de los reportes que se entregarán después de cada entrevista.

El perfil del analista de la tarea es el de un ergónomo, pues requiere tener conocimiento y experiencia sobre la realización del análisis de la tarea, encontrar las metas de los usuarios, y sobre todo, conocimiento de la ideología del diseño centrandolo en el usuario (el sistema es el que se tiene que adaptar al usuario, no al contrario).

4.4. Nuevas actividades

Como se comentó anteriormente, la integración del Análisis de la Tarea implica tres nuevas actividades, "Realizar análisis de la tarea del usuario", "Obtener árbol de la tarea modificada" y "Obtener casos de uso preliminares", cuyos resultados dan lugar a dos nuevos artefactos (figura 4.4).

4.4.1. Realizar análisis de la tarea del usuario

Esta actividad consiste en realizar el análisis de la tarea empleando el modelo MAD*, de forma muy similar a como se describió en el capítulo 2. Las modificaciones consisten en realizar la actividad dentro de un flujo de trabajo iterativo e incremental, y están basadas en el planteamiento que se hizo en la sección 3.1.

La planeación de las entrevistas es de suma importancia, pues constituyen la "materia prima" para el Análisis de la Tarea. Aunque es el analista quien propone la estructura del árbol de la tarea, los atributos de las fichas deben estar justificados por lo planteado por el usuario en las entrevistas. Por lo tanto, la documentación de las entrevistas es imprescindible, ya que un solo analista puede trabajar con varios usuarios en la elaboración del modelo de su tarea, y olvidar algunos detalles de las entrevistas.

La planeación de las entrevistas requiere de considerar las personas o equipos que realizarán las entrevistas, qué usuarios serán entrevistados, los materiales que se requieren, los medios por los que se documentarán las entrevistas, las reuniones de equipo necesarias para

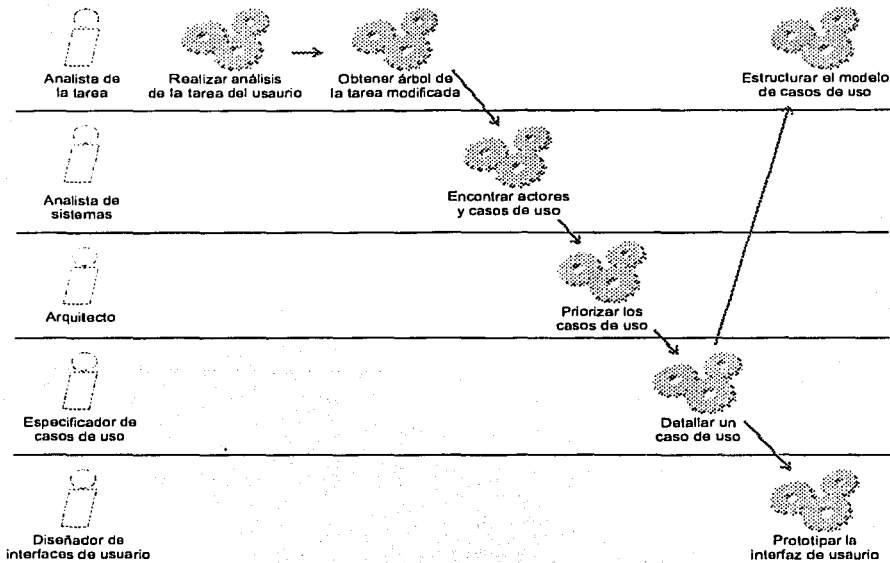


Figura 4.4: Nuevo flujo de trabajo para la captura de requisitos. Diagrama basado en la figura 7.10 del PU ([3], pág. 136).

intercambio de información, así como la fecha de entrega del árbol y sus fichas. En este plan es importante tomar en cuenta la disponibilidad de los usuarios para las entrevistas.

El Análisis de la Tarea del Usuario se realizará de forma iterativa e incremental con los siguientes pasos:

- **Realizar entrevistas:** En base al calendario de entrevistas que se tenga con el usuario, se tendrán reuniones en un principio para entrevistas no dirigidas, y posteriormente para entrevistas dirigidas, en las cuales el analista planteará al usuario sus dudas y solicitará cualquier material que pueda servir de apoyo en la comprensión de la tarea del usuario (manuales operativos, del sistema, bitácoras, etc.).
- **Análisis de las entrevistas:** La documentación de las entrevistas debe de ser analizada una y otra vez para estar certero de que se entendieron los planteamientos del usuario. Es importante anotar cualquier duda que se tenga sobre ellos, para resolverla en la siguiente entrevista.
- **Desarrollo del árbol:** A partir del conocimiento adquirido en el paso anterior, se procede a diseñar el árbol MAD* de la tarea del usuario que capture la forma en que el usuario realiza su tarea. Cuando se entrevista a varios usuarios, debe construirse un árbol que concilie los distintos enfoques.

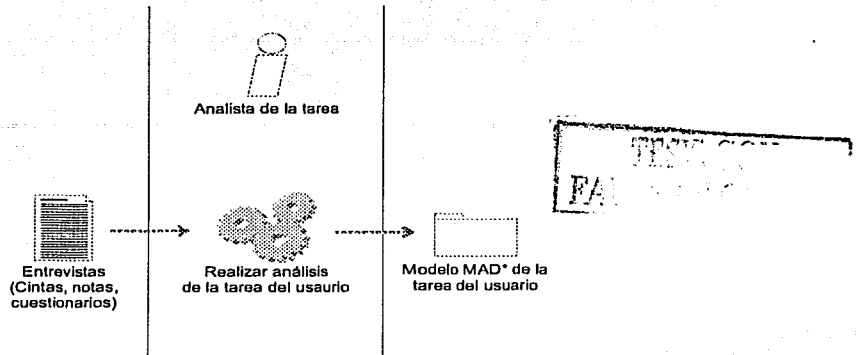


Figura 4.5: Las entradas y los resultados de “Realizar análisis de la tarea del usuario”.

- Creación de las fichas de las tareas: Cada nodo del árbol MAD* debe tener una ficha que lo describa. En esta paso los analistas frecuentemente encuentran varias preguntas que el usuario tendrá que responder en las siguientes entrevistas.
- Validación de los productos: Como se explicó en el Capítulo 2, el árbol y sus fichas se pueden validar mediante “Observación del usuario *In Situ*”, “Análisis de Trazas”, “Prueba de la tarea MAD* modelada” y “Prueba con interfaces obtenidas a partir del modelo MAD*” entre otros; pero cuando aún no se termina el AT algo que da buenos resultados es que el analista le plantee al usuario el entendimiento que tiene de su tarea y la estructura que le dio en el modelo MAD* para obtener comentarios.

4.4.2. Obtener árbol de la tarea modificada

Antes de obtener el árbol de la tarea modificada a partir del modelo MAD* de la tarea del usuario, es necesario que el analista de la tarea se reúna junto con los miembros del equipo de desarrollo y los encargados del proyecto, para establecer el alcance y los recursos del proyecto y poder así decidir qué subtareas pueden incluirse dentro de las tareas modificada.

Para obtener el árbol de la tarea modificada a partir del árbol de la tarea del usuario, se puede usar el siguiente procedimiento, en donde se recorren los nodos del árbol de la tarea del usuario determinando si pasarán al de la tarea modificada:

Para cada nodo X en el árbol, comenzando por la raíz (Tarea 0):

- Si X es una tarea no elemental:
 1. Aplicar procedimiento a cada una de sus subtareas.
 2. Si la complejidad de las subtareas en su conjunto es muy alta para incluir la tarea X en el nuevo sistema, descartar la tarea.
 3. Si todas las subtareas ya no se realizan con el nuevo sistema, entonces la tarea tampoco se realiza con el sistema y se descarta.

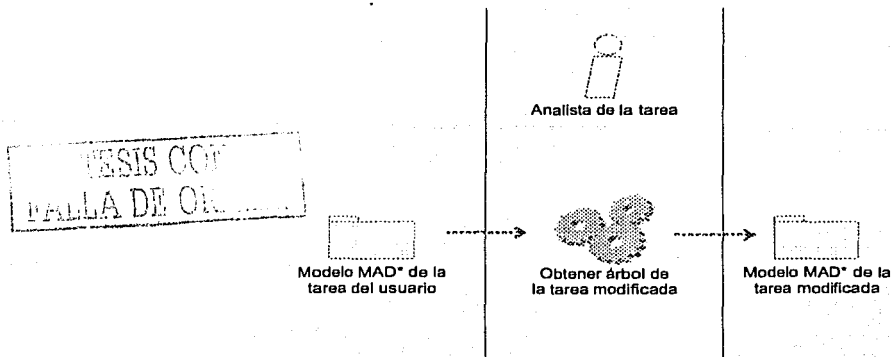


Figura 4.6: Las entradas y los resultados de "Obtener árbol de la tarea modificada".

4. Si, de igual forma, todas las subtareas de la tarea X se descartan debido a su complejidad, entonces X también se descarta.
 5. Si hay subtareas que se incluyen en el árbol de la tarea modificada, entonces la tarea también se incluye.
- Si la tarea es elemental, decidir si:
 1. Se descarta: La actividad ya no se realiza debido a la introducción del nuevo sistema, el alcance del proyecto no contempla esta actividad o la actividad no se puede realizar con el sistema.
 2. Se incluye en el árbol de la tarea modificada: La actividad se realizará con ayuda del sistema.

4.4.3. Elaborar diagramas de casos de uso preliminares

Como ya se ha mencionado, los diagramas de casos de uso preliminares son una primera versión de los casos de uso para el sistema, que requieren de menor trabajo para ser terminados. Éstos se obtienen del modelo de la tarea modificada mediante diversos criterios, como son por la relación con los objetos de la tarea, con los objetivos fundamentales, con las acciones que las tareas realizan, o con la posición de las tareas en el árbol (consultar sección 3.3.2).

Estos criterios fueron propuestos teniendo en mente el crear los diagramas de casos de uso preliminares de forma rápida que permita usar todos los criterios para un mismo proyecto, y tener así la posibilidad de escoger el diagrama que mejores características provea.

El procedimiento para obtener un diagrama de casos de uso preliminar es simple. En un diagrama de casos de uso se ponen como inicio las entidades en las que se basa el criterio usado (ej. objetos de las tareas, objetivos fundamentales, acciones de las tareas, etc.). Luego, para cada nodo del árbol de la tarea modificada, se decide con qué entidad Z la tarea está más fuertemente relacionada; si el padre Y de la tarea X está asociado a

la misma entidad Z, entonces la tarea X se coloca justo debajo de Y; de lo contrario X se asocia directamente a la entidad Z. Este proceso se aplica de forma recursiva a todas las tareas del árbol de la tarea modificada, para obtener al final un diagrama de casos de uso preliminar.

A continuación, se proponen formalmente los procedimientos para obtener diagramas de casos de uso preliminares a partir de la tarea modificada, con los criterios mencionados.

Procedimiento para obtener el diagrama de casos de uso preliminar, agrupando alrededor de objetos.

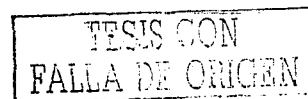
1. En el diagrama inicial, poner todos los objetos encontrados en las post-condiciones de las fichas de las tareas del árbol de la tarea modificada.
2. Comenzando por la tarea 1 del árbol de la tarea modificada, para cada tarea X con padre Y:
 - a) Determinar con qué objeto Z del diagrama X está más fuertemente relacionado, usando como referencia el objeto del mundo que es modificado por la tarea X, encontrado en su post-condición.
 - b) Si la tarea Y está asociada al mismo objeto Z que X, poner una relación del tipo de asociación entre X y Y. Si Y no está asociada al objeto Z, poner una relación de asociación entre X y Z directamente.
 - c) Aplicar el mismo procedimiento a todas las subtareas de X, si es que tiene.

Procedimiento para obtener el diagrama de casos de uso preliminar, agrupando alrededor de objetivos fundamentales.

1. En el diagrama inicial, poner como caso de uso a cada uno de los objetivos fundamentales encontrados posteriormente al análisis de la tarea del usuario.
2. Comenzando por la tarea 1 del árbol de la tarea modificada, para cada tarea X con padre Y:
 - a) Determinar con qué objetivo fundamental Z del diagrama X está más fuertemente relacionado, usando como referencia el objetivo y en general la descripción de la tarea X.
 - b) Si la tarea Y está asociada al mismo objetivo Z que X, poner una relación del tipo de asociación entre X y Y. Si Y no está asociada al objetivo Z, poner una relación de asociación entre X y Z directamente.
 - c) Aplicar el mismo procedimiento a todas las subtareas de X, si es que tiene.

Procedimiento para obtener el diagrama de casos de uso preliminar, agrupando alrededor de acciones.

1. En el diagrama inicial, poner como caso de uso a cada una de las acciones encontradas en las fichas de las tareas del árbol de la tarea modificada (ej. buscar, actualizar, guardar, etc.)
2. Comenzando por la tarea 1 del árbol de la tarea modificada, para cada tarea X con padre Y:



- a) Determinar con qué acción Z del diagrama X está más fuertemente relacionada, usando como referencia la descripción de la tarea X.
- b) Si la tarea Y está asociada a la misma acción Z que X, poner una relación del tipo de asociación entre X y Y. Si Y no está asociada a la acción Z, poner una relación de asociación entre X y Z directamente.
- c) Aplicar el mismo procedimiento a todas las subtareas de X, si es que tiene.

Procedimiento para obtener el diagrama de casos de uso preliminar, tal y como el usuario ve su tarea.

Para el caso en el que se quiera pasar el árbol de la tarea modificada directamente a un diagrama de casos de uso preliminar, conservando la estructura jerárquica, el enfoque es distinto pero mucho más sencillo: Las tareas se ponen directamente en el diagrama de casos de uso; cuando una tarea tenga subtareas, se creará un nuevo diagrama de casos de uso que especifique este detalle de la tarea. Así, la estructura jerárquica del árbol se convierte en una estructura de sub-diagramas de casos de uso.

Comenzando por la tarea 1 del árbol de la tarea modificada, para cada tarea X con padre Y:

1. Poner a X en el diagrama de casos de uso, asociado a su padre Y.
2. Si X tiene subtareas, crear un nuevo diagrama de casos de uso que describa a X, y aplicar el procedimiento para las subtareas de X y el sub-diagrama creado.

4.5. Actividades del PU que se apoyan con la integración

Con la integración del Análisis de la Tarea al Proceso Unificado de Desarrollo se apoya la realización de varias actividades existentes. Ahora se tiene, como se verá a continuación, heurísticas con las cuales buena parte de los artefactos pueden obtenerse usando la información del AT.

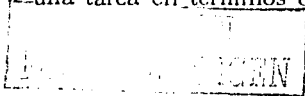
4.5.1. Encontrar actores y casos de uso

La actividad que tiene mayor apoyo es, por supuesto, "Encontrar actores y casos de uso", pues la elaboración de los diagramas de casos de uso puede hacerse ahora usando como punto de inicio los diagramas de casos de uso preliminares. Aunque en este trabajo no se propone un procedimiento con el cual obtener los actores y los casos de uso de forma directa, el trabajo que tiene que efectuarse para obtener un diagrama de casos de uso terminado es mucho menor.

A continuación, se proponen una heurísticas para elaborar el glosario de términos con ayuda de la información que aporta el AT y el modelo MAD* de la tarea.

Elaboración de glosario

El glosario de términos puede elaborarse usando como punto de partida la lista de objetos empleados en la tarea. Como se ha de recordar, es parte del AT con MAD* hacer un listado de todos los objetos que se usan en las pre y post-condiciones de las tareas del árbol, y proponer atributos a los objetos que permitan especificar los resultados de la ejecución de una tarea en términos de las modificaciones a los objetos del mundo del usuario. El AT



carece de una descripción de los objetos de la tarea, por lo que el analista tendrá proponer cada definición; mas sin embargo, el listado de objetos sirve muy bien para identificar e incluir todos los términos importantes, pues estos están directamente involucrados en las condiciones de las tareas.

La lista de objetos de las tareas, y sus atributos, puede usarse como candidatos a entidades a especificarse en el glosario de términos. Para cada objeto o atributo, se requiere ponderar si su significado es de sentido común o tiene significado especial en el contexto del negocio. Si el término no es de sentido común, o tiene significado especial en el contexto, entonces requiere describirse en el glosario de términos.

Esto es, una entidad de la lista de lista objetos de la tarea tiene que estar en el glosario de términos si su significado no es del sentido común o depende del contexto. El analista tiene la sensibilidad para decidir si requiere definirse o no debido a que tiene los productos del AT, que le aportan un conocimiento “desde adentro” del mundo del usuario. Por supuesto, se deberá evaluar la posibilidad de agregar al glosario cualquier nuevo término que surja durante la descripción de una entidad, y cuyo significado no sea claro.

4.5.2. Detallar un caso de uso

El detalle de los casos de uso se obtiene de forma casi directa del AT. Si el caso de uso proviene un diagrama preliminar de casos de uso, entonces fue creado a partir de una tarea que tiene una ficha que la describe. De esta ficha, se pueden usar los campos de “Objetivo”, “Actor” y “Papel del actor” para elaborar la descripción del caso de uso, tal y como se ilustró en la figura 3.4.

El “flujo de sucesos” y los “caminos alternativos” de la descripción del casos de uso pueden obtenerse directamente del árbol de la tarea modificada; los diagramas de casos de uso preliminares pueden aportar también información de los niveles de detalle de la composición de un caso de uso, que puede ser de utilidad en la descripción del caso de uso.

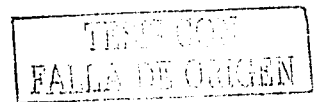
4.5.3. Prototipar la interfaz del usuario

La creación de prototipos de interfaz a partir del análisis de la tarea con MAD* también es posible, ya que este captura la secuencia detallada de los eventos durante la ejecución de una tarea.

De igual forma, los constructores de las tareas sugieren el tipo de control que debe proporcionar la interfaz en un cierto punto de la ejecución de la tarea: “paralelo” implica que el usuario debe poder efectuar una de varias subtarecas en la misma interfaz; “secuencial” sugiere un flujo de trabajo tipo wizard, en donde la interfaz lleva al usuario por los diferentes pasos de la tarea; un constructor “Y” implica que el usuario debe de ser capaz de realizar dos tareas en simultáneo, compartiendo información entre ellas; etc.

Se puede verificar también si la información que aparece en una pantalla es adecuada, revisando los objetos involucrados en la tarea que apoya la interfaz. Comúnmente se requiere tener en pantalla la información de los objetos del mundo real que la tarea afectará con su ejecución.

Una propuesta de un modelo, y un aplicativo de software, que permite crear de forma rápida prototipos de interfaz reactivos a partir un árbol de tarea MAD* puede encontrarse en F. Gamboa, D. L. Scapin[1].



4.6. Resumen

En suma, la integración del Análisis de la Tarea al Proceso Unificado implica la introducción de un nuevo rol de trabajo y dos nuevas actividades. Como se había buscado, las actividades del flujo de captura de requisitos del PU se mantuvieron intactas, y inclusive se propusieron heurísticas con las cuales se pueden desarrollar de forma más sencilla y rápida.

Entre los beneficios de la integración, podemos destacar el uso del AT como un método de entrevistado y de captura de información del usuario, así como la obtención de los casos de uso, sus diagramas y descripciones, a partir de un método analítico y procedural. El análisis de la tarea es, como se quería, una herramienta que apoya la captura de requisitos.

El nuevo artefacto, el modelo MAD* de la tarea, documenta muy bien al usuario y su tarea, y sirve como referencia no sólo para el flujo de captura de requisitos, sino para todo el proyecto. Sería interesante, y de gran utilidad, el plantear los beneficios para los otros flujos de trabajo de haber incorporado el AT con MAD* al PU.

TEXTO CON
FALLA DE ORIGEN

Capítulo 5

Desarrollo de un caso de estudio

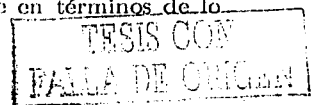
5.1. Objetivo

Lo que se espera del proceso de desarrollo propuesto en esta tesis, es que los desarrollos tengan requerimientos más apegados a los objetivos, procedimientos y formas de pensar del usuario; y que además, la obtención de estos requerimientos sea una actividad con un sentido procedural y analítico, en comparación a la situación de un desarrollo con el PU usual.

Además, como se mencionó en la introducción, la propuesta no sólo consiste en aplicar el AT de forma aislada y/o paralela al PU, sino que se han introducido las actividades necesarias para obtener un proceso integrado. De esta integración se espera también una mejora en la implementación del proceso de desarrollo en función de la facilidad, esfuerzo y tiempo requeridos, por lo que la evaluación del nuevo proceso tendrá que abarcar varios aspectos. Entre ellos podemos identificar:

- El nuevo enfoque propone nuevas actividades, como las de “Realizar análisis de la tarea del usuario”, “Obtener árbol de la tarea modificada”, “Elaborar diagramas de casos de uso preliminares”, etc. que requieren de nuevos roles en el equipo de desarrollo, y cuyo objetivo es facilitar y disminuir la complejidad de las actividades del PU. Es necesario evaluar los cambios en tiempo y carga de trabajo en la propuesta de desarrollo.
- El modelo MAD* de la tarea del usuario y de la tarea modificada son artefactos que se han introducido al proceso de desarrollo, buscando documentar de forma adecuada la tarea del usuario, y apoyar la documentación de las reglas y conocimiento del negocio. Es necesario evaluar en casos prácticos cuánto aportan estos nuevos artefactos y que tanto facilitan la comprensión del problema a un nuevo integrante del grupo de desarrollo.
- Los casos de uso y sus diagramas son una parte medular en el PU, y el proponer una forma de obtenerlos a partir del AT sugiere que debemos de evaluar también la forma en que esto afecta al proceso de desarrollo. De primera vista, podría esperarse que casos de uso mejor estructurados nos llevan a un proceso mejor administrado.

En este capítulo se aplicará la propuesta de desarrollo a un caso de la vida real con el objetivo de obtener una primera evaluación del nuevo enfoque en términos de lo anteriormente planteado.



5.2. Planteamiento del problema

El alcance de este trabajo no permite efectuar el desarrollo completo de un sistema real, en el cual fuera posible evaluar todos los puntos planteados anteriormente. En las opciones están efectuar sólo una parte del proceso de desarrollo de un sistema complejo, o realizar el desarrollo completo de un sistema modesto.

En el caso de desarrollar de forma completa un sistema pequeño, el inconveniente es que una gran parte del tiempo invertido se dedicaría a actividades de diseño, implementación y pruebas, las cuales quedan fuera del interés de nuestro trabajo. La ventaja sería evaluar en su totalidad el desempeño del sistema en un ambiente de producción real.

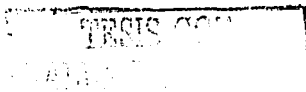
Además, aún si se efectuara un desarrollo completo, no se podrían obtener resultados concluyentes, puesto que para eso se requiere una aplicación repetida a varios casos reales de diversa índole y complejidad.

La propuesta de desarrollo dejó al AT intacto, por lo que más bien se requiere evaluar las modificaciones que se hicieron al PU. Es por eso que consideramos que es de mayor utilidad aplicar el proceso propuesto a la etapa de inicio de un sistema de mayores dimensiones, en donde se pueda apreciar al AT, integrado al PU, como una herramienta de captura de requisitos.

Con este propósito se eligió el realizar la etapa de inicio de un proceso de desarrollo para un sistema que apoyara las actividades de la contadora Isabel Andrade Loredo, del Centro de Ciencias Aplicadas y Desarrollo Tecnológico. Cabe mencionar que el tipo de contabilidad que se realiza en las dependencias de la UNAM es totalmente distinta a la contabilidad comercial, por lo que éste resulta un buen ejemplo, pues es de grandes dimensiones y se tiene total desconocimiento del dominio del problema.

5.3. Desarrollo del proceso

Se comenzó el proceso de desarrollo con una entrevista con la contadora Andrade Loredo, para conocer el estado actual de sus actividades y tener una primera impresión del tipo de problemas que enfrentaba. El documento que resume la entrevista se muestra en la figura 5.1.



Primer contacto con el usuario

Corno encargada del fondo fijo del Centro de Ciencias Aplicadas y Desarrollo Tecnológico es responsable de llevar toda la contabilidad de éste. La actividad que le demanda más tiempo y esfuerzo es la de revisar todos sus documentos para encontrar incongruencias.

Aunque ya cuenta con sistema de computadora para apoyarla en su actividad, la revisión de documentos la hace de forma manual a partir de las impresiones que le proporciona el sistema.

Los requerimientos para el nuevo sistema, planteados por ella, son:

1. Ayuda en la revisión de registros.
2. Formas de captura que reduzcan la posibilidad de error.

Cabe mencionar que ella no tenía muy clara la forma de lograr estos requerimientos e inclusive si fuera posible hacerlo, por lo que no se sentía muy optimista al respecto.

Figura 5.1: Documento que resume la información que se obtuvo en la primera cita y en donde se describe el problema.

TESIS CON
FALLA DE ORIGEN

5.3.1. Análisis de la tarea del usuario

El análisis de la tarea se realizó tal y como se indicó en la propuesta, comenzando por las entrevistas no dirigidas, de las cuales se obtuvo un bosquejo de lo que sería la tarea del usuario; y posteriormente se realizaron entrevistas dirigidas para completar el árbol de la tarea.

Sólo fueron necesarias dos entrevistas no dirigidas, la primera de cincuenta minutos y la segunda de una hora. Posteriormente se efectuaron tres entrevistas dirigidas (45, 11 y 15 minutos) con las cuales se obtuvo el árbol final de la tarea del usuario, que se muestra en la figura 5.2. Todas las entrevistas fueron grabadas en cinta de audio como medio de documentación.

Una vez que se llegó a un árbol completo y estable, se elaboraron sus fichas, se identificaron los objetos utilizados en cada tarea y se elaboraron posteriormente las fichas con objetos. En el Apéndice A se pueden consultar las fichas para el árbol de tarea de la contadora, y en el Apéndice B sus objetos y fichas con objetos.

Con el conocimiento adquirido del Análisis de la Tarea, y siguiendo las guías propuestas en el capítulo 4, se encontraron los requisitos candidatos (figura 5.3) y el glosario de términos (figura 5.4), como se explica a continuación.

5.3.2. Requisitos candidatos

Siguiendo las sugerencias del PU, se realizó la enumeración de los requisitos candidatos, basándose en la información aportada por el Análisis de la Tarea. Aunque la obtención de los requisitos candidatos no es una actividad que se apoya en esta propuesta, es importante notar que es bueno encontrar la justificación de éstos a partir de lo expresado en el modelo de la tarea del usuario.

En las entrevistas, el usuario planteó cómo el sistema actual fue evolucionando, tratando de adaptarse a sus necesidades. De aquí se concluye que el usuario y el cliente esperan que el nuevo sistema los apoye al menos en las mismas actividades (Requerimiento 1).

De las fichas de las tareas se concluye que la tarea "2.4 Conciliar datos con estado de cuenta", que es muy demandante de tiempo, se realiza debido a que se introducen errores humanos al momento de la captura de los documentos, ocasionando que el saldo presentado por el sistema no coincida con el de la cuenta bancaria. Por eso, se pone como requisito (Requisito 2) el contar con interfaces que ayuden a reducir los errores de captura.

De la tarea "3 Revisar registros" y las fichas de sus subtareas, se encuentra que una actividad importante es la de dar seguimiento al estado de los préstamos. Sin embargo, de las tareas 2.1, 2.2 y 2.3, así como de las pantallas del sistema contable actual que proporcionó el usuario, se puede ver que el sistema actual no ayuda mucho a relacionar préstamos con comprobaciones y reembolsos. Es por eso que se propone (Requisito 3) implementar un esquema en el cual ese sistema relacione de forma clara los movimientos que pertenecen a un mismo proceso de préstamo.

Por otra parte, sabemos que aunque se proponga un mecanismo para reducir los errores de captura en el requisito 2, éstos seguirán ocurriendo. De la descripción de la tarea "2.4 Conciliar datos con estados de cuenta" encontramos que la contadora tiene que revisar registro por registro para encontrar inconsistencias. Lo que se propone (Requerimiento 4) es apoyar a la contadora mediante la generación de estados de cuenta del sistema similares a los del banco, imprimiendo los movimientos por fecha y agrupados por depósito o retiro, de tal forma que rápidamente se pueda encontrar en qué registro está la inconsistencia, para

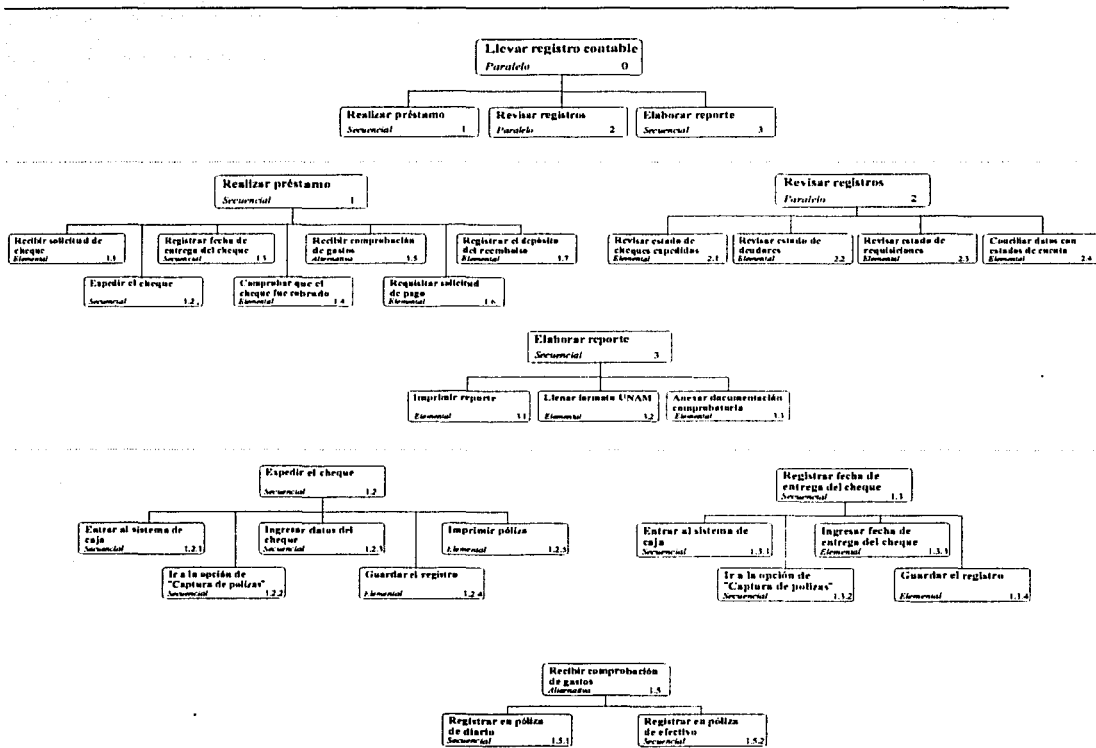


Figura 5.2: Árbol de la tarea de la contadora.

TESIS CON
FALLA DE ORIGEN

Requisitos candidatos

1. Desarrollar un sistema que tenga toda la funcionalidad del sistema actual.
Estado: Validado, Costo: 1920 horas/hombre, Prioridad: Crítico, Riesgo: Significativo.
2. Las interfaces del nuevo sistema deben de estar encaminadas a reducir los errores humanos.
Estado: Aprobado, Costo: 160 horas/hombre, Prioridad: Crítico, Riesgo: Significativo.
3. El nuevo sistema debe de asistir al usuario en la revisión de movimientos pendientes.
Estado: Propuesto, Costo: 80 horas/hombre, Prioridad: Importante, Riesgo: Ordinario.
4. El sistema debe asistir al usuario en la revisión de inconsistencias entre resultados del sistema y los estados de cuenta.
Estado: Propuesto, Costo: 80 horas/hombre, Prioridad: Importante, Riesgo: Ordinario.
5. Creación de manuales para los usuarios especialmente diseñados para disminuir tiempos de capacitación.
Estado: Aprobado, Costo: 160 horas/hombre, Prioridad: Secundario, Riesgo: Ordinario.

Figura 5.3: Enumeración de requisitos candidatos, propuestos por el usuario y los analistas.

luego proceder a revisarlo.

En la entrevista el usuario planteó que se requería de una capacitación extensiva para que un nuevo usuario pudiera operar el sistema contable actual. Dada la complejidad de la tarea del usuario, sabemos que aunque se diseñen interfaces que cumplan con los criterios ergonómicos ([?]), aún será necesario proveer manuales orientados a disminuir los tiempos necesarios para familiarizarse con el nuevo aplicativo (Requerimiento 5). Sin embargo, es de esperarse que una persona con experiencia en la administración universitaria sea capaz de utilizar el sistema en tiempos cortos.



5.3.3. Glosario de términos

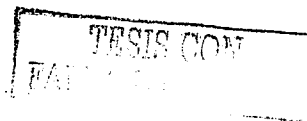
Usando la guía propuesta en la sección 4.5.1, se toma como referencia la lista de objetos modificados por la tarea del usuario y sus atributos (ver apéndice B), de los cuales tenemos como entidades importantes a:

Sistema contable	Dependencia
Deudor	Préstamo
Cheque	Póliza
Documentación comprobatoria	Requisición
Reembolso	Fondo fijo
Solicitud de préstamo	Contador
Reporte	

Para no definir a todos los términos, muchos de los cuales se conocen del sentido común, se definen aquellos que requieran de más explicación, como se justifica en la siguiente tabla:

Término	Razón por la que se define
Deudor	Explicar papel del deudor en la dependencia.
Préstamo	Explicar papel del préstamo en la dependencia.
Fondo fijo	Explicar cuál es el fin del fondo fijo.
Documento comprobatorio	Explicar su significado y tipos.
Solicitud de préstamo	Explicar su papel en la dependencia y sus elementos.
Requisición	Explicar término y papel en el contexto.
Contador	Definir lo que es en el contexto de la dependencia.
Reembolso	Explicarlo en el contexto de la dependencia y sus préstamos.
Reporte	Hay muchos tipos de reporte, especificar para dar más claridad.

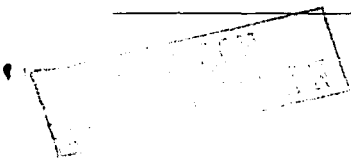
Explicando cada término, como se muestra en la figura 5.4, se encuentra que para definir "Fondo fijo" es necesario también definir lo que es "Partida presupuestal".



Glosario de términos

- *Contador(a)*: Persona encargada de llevar el seguimiento de todas las entradas y salidas de dinero de una institución de la UNAM, poniendo especial interés de que cada trámite de dinero termine adecuadamente. Se encarga también de generar, documentar y entregar los estados financieros de su institución.
- *Fondo fijo*: Cuenta bancaria empleada para efectuar préstamos que posteriormente se cobrarán a las partidas presupuestales. Al comienzo de cada ejercicio se asigna cierta cantidad de dinero y al fin del mismo debe de haber la misma cantidad.
- *Partida presupuestal*: Fuente de dinero para una institución. Entre ellas están *Ingresos presupuestales* e *Ingresos extraordinarios*. Cada partida tiene personal encargado de su administración, de forma similar a como la institución tiene su contador(a).
- *Préstamo*: Dinero que sale del *fondo fijo* y que, mediante documentos que comprueben su gasto, será recuperado de una partida presupuestal.
- *Documento comprobatorio*: La nota o factura de un gasto efectuado con el dinero del préstamo. En caso de que no se haga gastado el total del préstamo, el dinero restante es considerado documento comprobatorio.
- *Deudor*: Persona a la que se le otorga un préstamo por algún concepto y que está obligada a efectuar gastos sólo por ese concepto y entregar los documentos comprobatorios a la contadora.
- *Reembolso*: Pago de un préstamo.
- *Requisición*: Proceso mediante el cual la contadora emplea los documentos comprobatorios para solicitar a una partida presupuestal el reembolso de un préstamo.
- *Solicitud de préstamo*: Documento mediante el cual el subdirector de la dependencia solicita a la contadora efectúe un préstamo a alguna persona.
- *Reporte*: Documentos que indican la situación actual del fondo fijo, incluyendo los préstamos en trámite, sus comprobaciones y los reembolsos. Estos documentos incluyen los estados de cuenta bancarios, el reporte impreso generado por el sistema y fichas de depósito de los reembolsos.

Figura 5.4: Glosario de términos



5.3.4. Obtener árbol de la tarea modificada

El procedimiento empleado para la obtención del modelo de la tarea modificada es el propuesto en la sección 4.4.2.

A partir del árbol de la tarea y tomando en cuenta el alcance y recursos del proyecto, se elabora el árbol de la tarea del nuevo sistema (figura 5.7), en el cual las tareas que no se incluirán se marcan con una "X".

Para decidir qué tareas no quedarán en el árbol intermedio, comenzamos analizando las subtareas de cada tarea: Para decidir si la tarea 0 quedará, tenemos primero que revisar 1, 2 y 3. Para decidir sobre 1, hay que analizar de la 1.1 a la 1.7, y así sucesivamente.

Para la tarea 1 se tiene:

- 1.1 no queda en la tarea intermedia, pues "Recibir solicitud de cheque" se realiza de forma totalmente manual y no es posible automatizarla con el sistema.
- La tarea "1.2 Expedir el cheque" se queda pues tiene tareas que se conservan:
 - "1.2.1 Entrar al sistema de caja" se elimina ya que se refiere a la forma de interactuar con la interfaz y ésta cambiará en el nuevo sistema.
 - De forma similar, "1.2.2 Ir a la opción de captura de pólizas" y "1.2.4 Guardar registro" no estarán en el árbol intermedio pues se refieren a interacción con la interfaz.
 - Las tareas de "1.2.3 Ingresar datos del cheque" y "1.2.5 Imprimir póliza" se conservan ya que sí quedan en el alcance del proyecto.

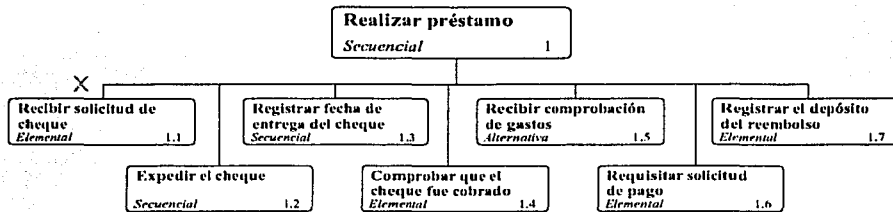


Figura 5.5: Tarea 1 de la tarea modificada

- También "1.3 Registrar fecha de entrega del cheque" se queda pues tiene una subtarea:
 - Las tareas "1.3.1 Entrar al sistema de caja", "1.3.2 Ir a la opción de captura de pólizas" y "1.3.4 Guardar el registro" no aparecen en la tarea modificada pues se refieren a interacción con el sistema.
 - "1.3.3 Ingresar fecha de entrega del cheque" se mantiene pues es necesaria en la tarea y queda en el alcance del proyecto.
- "1.4 Comprobar que el cheque fue cobrado" se incluye en el nuevo sistema, debido a la importancia para el usuario de tener el registro de cheques cobrados.

- “1.5 Recibir comprobación de gastos” se queda pues tiene dos subtareas interactivas:
 - “1.5.1 Registrar en póliza de diario” y
 - “1.5.2 Registrar en póliza de efectivo” se quedan pues son necesarias en la tarea y quedan en el alcance del desarrollo.
- “1.6 Requisitar solicitud de pago” se incluye en el nuevo sistema.
- “1.7 Registrar el depósito del reembolso” se queda.

De esta forma, la tarea “1 Realizar préstamo” se conserva ya que tiene subtareas que quedan en el alcance del proyecto.

Procediendo de la misma forma con “2 Revisar registros”, encontramos que se conserva en la tarea modificada:

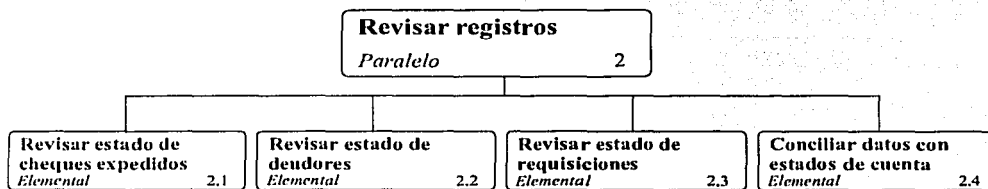


Figura 5.6: Tarea 2 de la tarea modificada

- “2.1 Revisar estado de cheques expedidos”, “2.2 Revisar estado de deudores” y “2.3 Revisar estado de comprobaciones” se mantienen pues son importantes en la tarea del usuario y quedan en el alcance del proyecto.
- “2.4 Conciliar datos con estados de cuenta” se incluye en la tarea modificada, en el sentido de proporcionar un medio para conciliar registros con estados de cuenta.

Finalmente, “3 Elaborar reporte” se conserva pues tiene una única subtarea:

- “3.1 Imprimir reporte” queda dentro del alcance del proyecto.
- “3.2 Llenar formato de la UNAM” no se incluirá pues queda fuera de este proyecto el automatizarla.
- “3.3 Anexar documentación comprobatoria” es una tarea manual y queda fuera de la tarea modificada.

De lo anterior, tenemos que las tareas “1 Realizar préstamo”, “2 Revisar registros” y “3 Elaborar reporte” quedan en la tarea intermedia; 1 y 2 son tareas interactivas mientras que 3 es automática.

Luego, el árbol de la tarea intermedia es mapeado a casos de uso preliminares. Como se planteó en el capítulo anterior, el mapeo puede ser alrededor de objetivos fundamentales (subsección 5.3.5), de los objetos de la tarea (subsección 5.3.6), como la ve el usuario (subsección 5.3.7) o de las acciones que el usuario realiza (subsección 5.3.8).

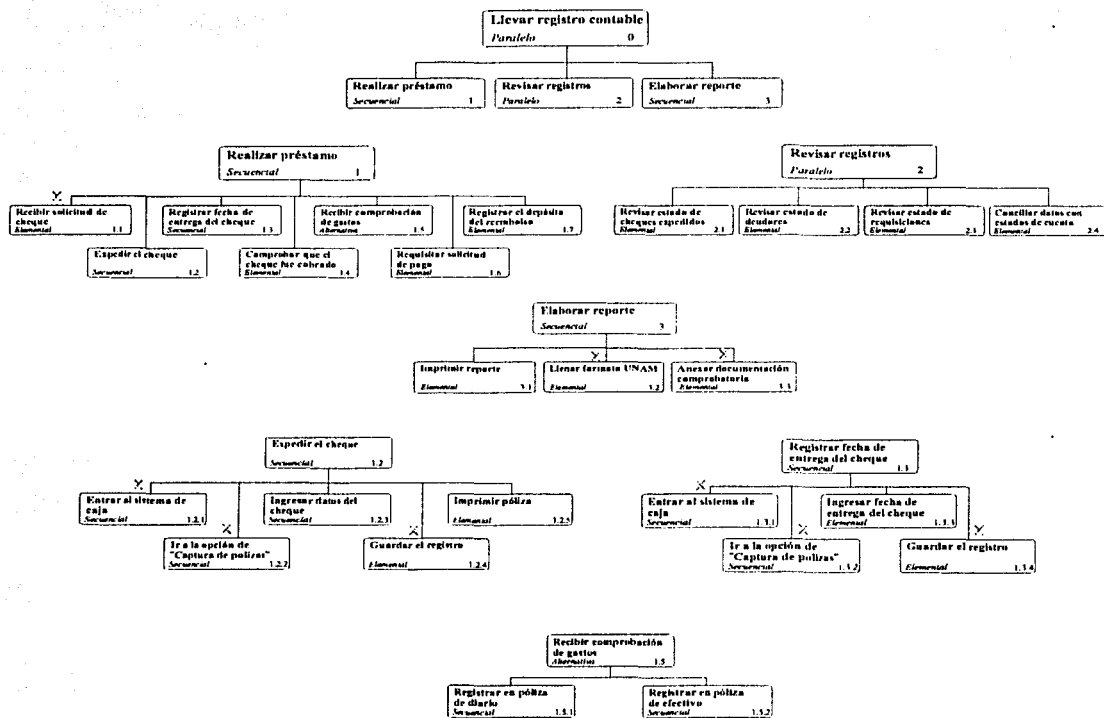


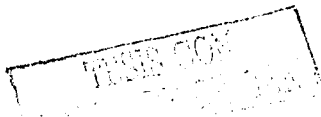
Figura 5.7: Árbol de la tarea modificada. Las tareas marcadas con una "X" no se incluyen.

TESIS CON
FALLA DE CALIDAD

5.3.5. Mapeo de la tarea a casos de uso preliminares alrededor de objetivos fundamentales

Siguiendo el procedimiento propuesto en el capítulo 4, comenzamos tomando a los objetivos fundamentales de la tarea del usuario, "*Prestar dinero*" y "*No quedarse sin dinero*", como los dos casos de uso primordiales. Recorriendo el árbol intermedio por ramas, cada tarea pasará al diagrama preliminar como un caso de uso relacionado a un objetivo fundamental:

- La tarea "*1 Realizar préstamo*" es asociada directamente con el objetivo fundamental de "*Prestar dinero*" dada su similitud.
- "*1.2 Expedir el cheque*" tiene que ver con "*Prestar dinero*", por lo que irá asociada con este objetivo fundamental. Como la tarea *1* ya está también, asociamos a *1.2* con *1*.
- "*1.2.3 Ingresar datos del cheque*" va en "*Prestar dinero*" pues es parte del proceso para otorgar un préstamo por medio de un cheque. De igual forma "*1.2.5 Imprimir póliza*" va en el mismo objetivo fundamental.
- "*1.3 Registrar fecha de entrega del cheque*" va en "*Prestar dinero*" pues tiene que servir para demostrar que el préstamo ya se otorgó a la persona indicada.
- "*1.4 Comprobar que el cheque fue cobrado*" es relacionada con "*No quedarse sin dinero*", pues su objetivo es que el contador sepa en todo momento cuántos cheques faltan por cobrar y a cuántos préstamos hay que comenzarles el proceso de seguimiento.
- "*1.5 Recibir comprobación de gasto*" va en "*No quedarse sin dinero*" pues forma parte del conjunto de tareas cuyo objetivo es dar seguimiento al proceso de préstamo para que éste sea recuperado. De igual forma, sus subtareas "*1.5.1 Registrar en póliza de diario*" y "*1.5.2 Registrar en póliza de efectivo*", así como las tareas "*1.6 Requirir solicitud de pago*" y "*1.7 Registrar el depósito del reembolso*", quedan también en este objetivo fundamental.
- "*2 Revisar registros*" se hace con el objetivo de encontrar tanto los movimientos que están pendientes, como los que han excedido de tiempo; y también para encontrar errores en los datos del sistema. Esto lo hace el contador para recuperar el dinero prestado cuando antes posible e íntegro, por lo que claramente esta tarea se asociará con "*No quedarse sin dinero*".
- Prosiguiendo el análisis con las subtareas de "*2*", es evidente que "*2.1 Revisar estado de cheques expedidos*", "*2.2 Revisar estado de deudores*" y "*2.3 Revisar estado de requisiciones*" tienen como propósito percatarse al contador del estado actual de cada préstamo, con el fin de que éste intervenga en aquellos que han excedido el tiempo esperado y consiga el reembolso del préstamo a tiempo. De esta forma, estas tareas también caen dentro del O. F. "*No quedarse sin dinero*".



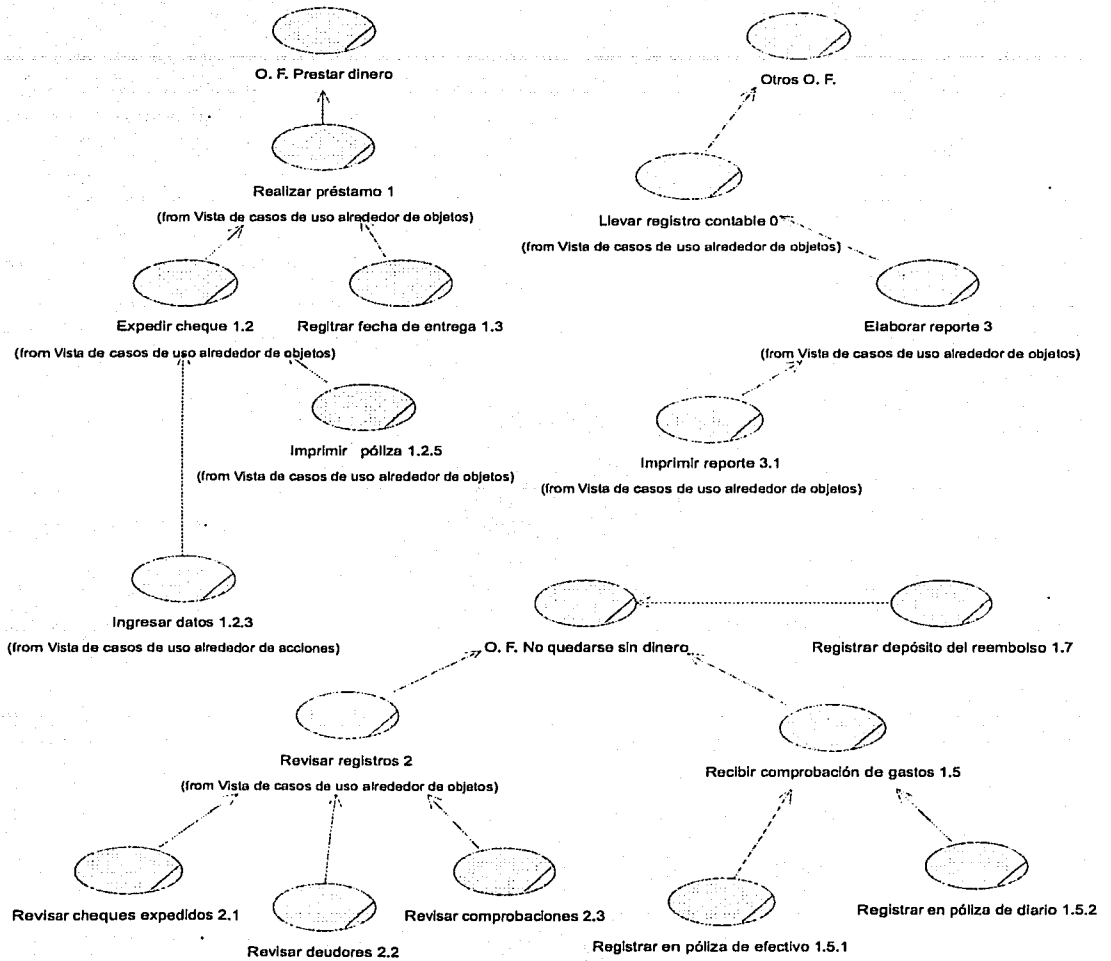
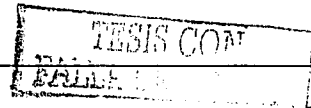


Figura 5.8: Casos de uso preliminares alrededor de los objetivos fundamentales de la tarea del usuario.



- *"2.4 Conciliar datos con estados de cuenta"* se hace con el objetivo de encontrar los inconsistencias de los datos del sistema con los de las cuentas bancarias, y así evitar principalmente la pérdida de dinero. Es por eso que *"2.4"* es relacionada con el O. F. de *"No quedarse sin dinero"*.
- Al analizar *"3 Elaborar reporte"* nos damos cuenta de que no se identificaron todos los objetivos fundamentales de la tarea del usuario, pues esta tarea no encaja en los dos O. F. identificados. Para propósitos de la traducción de tareas a casos de uso preliminares, se crea otro caso de uso para asignarle *"3"*. Lo mismo ocurre con *"3.1 Imprimir reporte"* y queda relacionada con *"3"*, dentro del objetivo fundamental no encontrado.

El resultado de este procedimiento puede verse en la figura 5.8.

5.3.6. Mapeo de la tarea a casos de uso preliminares alrededor de objetos de la tarea

El proceso de mapeo comienza seleccionando los objetos de la tarea que serán utilizados para agrupar los casos de uso. Analizando la lista de objetos de la tarea del usuario se seleccionan

Registro del sistema	Dependencia
Deudor	Préstamo
Comprobación	Reembolso
Cheque	Reporte

pues son las entidades indispensables para que la tarea se pueda llevar a cabo. Asignando cada subtarea de cada rama del árbol a un objeto:

- *"0 Llevar registro contable"*, aunque involucra a todos los objetos propuestos por ser la raíz del árbol de la tarea, tiene una relación más fuerte con *"Centro de Ciencias Aplicadas y Desarrollo Tecnológico"* pues en diagrama jerárquico, al igual que en el árbol de la tarea, sería el que contiene a todos los demás objetos.
- *"1 Realizar préstamo"* tiene relación con *"Cheque"*, *"Deudor"*, *"Comprobación"*, *"Centro de Ciencias Aplicadas y Desarrollo Tecnológico"* y *"Reembolso"*, pero la relación queda con *"Préstamo"* pues es más general y contiene a los anteriores.
- *"1.2 Expedir el cheque"* se relaciona directamente con *"Cheque"* pues éste el objeto en el que trabaja, y con el mismo razonamiento *"1.2.3 Ingresar datos el cheque"*, *"1.2.5 Imprimir póliza"*, *"1.3 Registrar fecha del cheque"*, *"1.3.3 Ingresar fecha de entrega del cheque"* y *"1.4 Comprobar que el cheque fue cobrado"* quedan también asociados con este objeto. Nótese que *"1.2.3"* y *"1.3"* podrían relacionarse con *"Deudor"*; y por supuesto, también todas las tareas anteriores se relacionan con *"Préstamo"*; pero se consideró que todas estas tareas tienen una relación más fuerte con *"Cheque"* ya que el usuario las efectúa más bien para tener el cheque del préstamo y no el préstamo en sí. Los préstamos se pueden otorgar de muchas otras formas que no son por cheque.
- *"1.5 Recibir comprobación de gastos"*, *"1.5.1 Registrar en póliza de diario"* y *"1.5.2 Registrar en póliza de efectivo"* se relacionan en el diagrama preliminar de casos de uso con *"Comprobación"*, pues las tareas van encaminadas a controlar la entrada de comprobaciones en primera instancia.

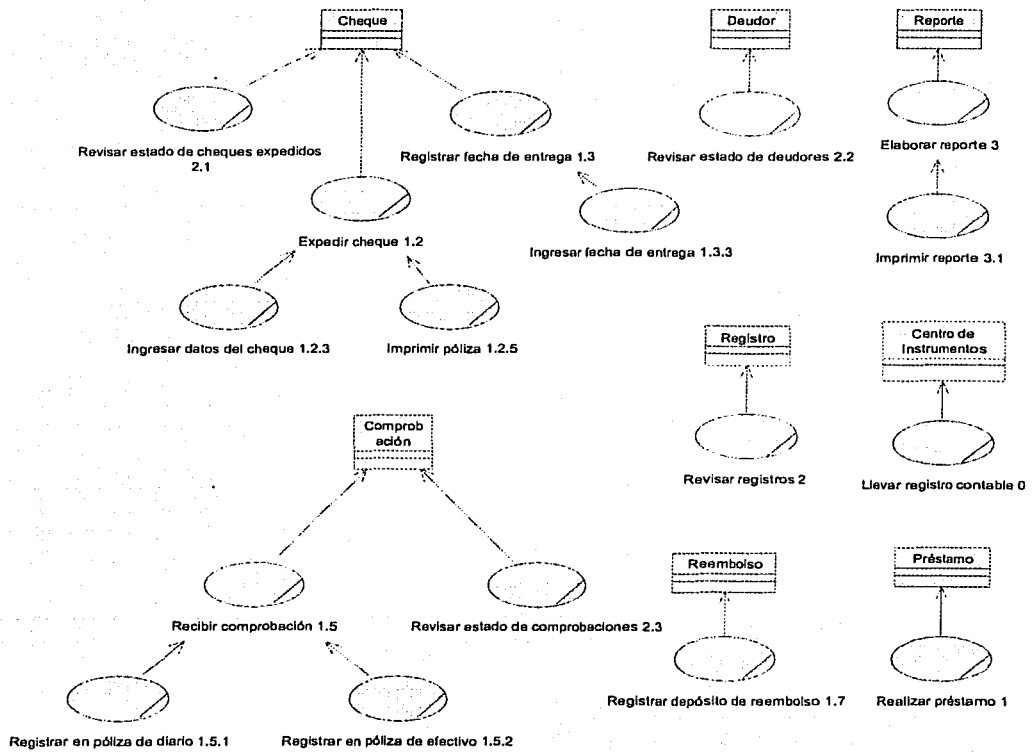


Figura 5.9: Casos de uso preliminares alrededor de los objetos empleados en la tarea del usuario.

TESIS CON
FALLA DE ORIGEN

- “1.6 Requisar solicitud de pago” y “1.7 Registrar el depósito del reembolso” se asignan a “Reembolso”, ya que ambas administran directamente a esta entidad.
- “2 Revisar registros” se asocia a “Registro”.
- Aunque “2” se asoció a “Registro”, “2.1 Revisar estado de cheques expedidos” se asocia a “Cheque”, “2.2 Revisar estado de deudores” a “Deudor” y “2.3 Revisar estado de requisiciones” a “Comprobaciones”, pues las tareas trabajan sobre esos objetos principalmente.
- “2.4 Conciliar datos con estados de cuenta” se asocia a “Registro” pues es de todos los objetos con el que guarda más relación.
- “3 Elaborar reporte” y “3.1 Imprimir reporte” se asignan a “Reporte”.

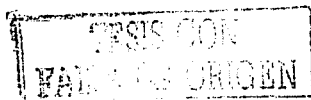
El resultado de este procedimiento puede verse en la figura 5.9.

5.3.7. Mapeo de la tarea a casos de uso preliminares alrededor de la tarea como la realiza el usuario

Para este caso el mapeo es simple. Sin embargo, las relaciones no se pueden expresar en un solo diagrama. Se requiere, para tarea que pasa a caso de uso, elaborar un subdiagrama con las relaciones que tiene con sus subtareas:

- Comenzando por la raíz del árbol (“0 Llevar registro contable”), creamos un diagrama en donde se encuentre ésta y sus subtareas (“1 Realizar préstamo”, “2 Revisar registros” y “3 Elaborar reporte”).
- Luego, para el mapeo de “1”, creamos un nuevo diagrama en donde lo relacionamos con todas sus subtareas (“1.2”, “1.3”, “1.4”, “1.5”, “1.6” y “1.7”).
- Para “1.2 Expedir cheque” se hace un diagrama que lo relaciona con sus subcasos de uso “1.2.3” y “1.2.5”.
- Se crea un diagrama, que corresponde al detalle de “1.3 Registrar fecha de entrega del cheque”, en donde se le relaciona “1.3.3 Ingresar fecha de entrega del cheque”.
- De igual forma, en el detalle de “1.5 Recibir comprobación de gastos” se incluyen “1.5.1” y “1.5.2”.
- Continuando el desarrollo por ramas del árbol de la tarea, para el detalle de casos de uso de “2 Revisar registros” se crea un diagrama de casos de uso en donde se ponen como subcasos de “2” a las tareas “2.1”, “2.2”, “2.3” y “2.4”.
- Finalmente, se describe “3 Elaborar reporte” con su único subcaso de uso “3.1 Imprimir reporte”.

El resultado de este procedimiento puede verse en la figura 5.10.



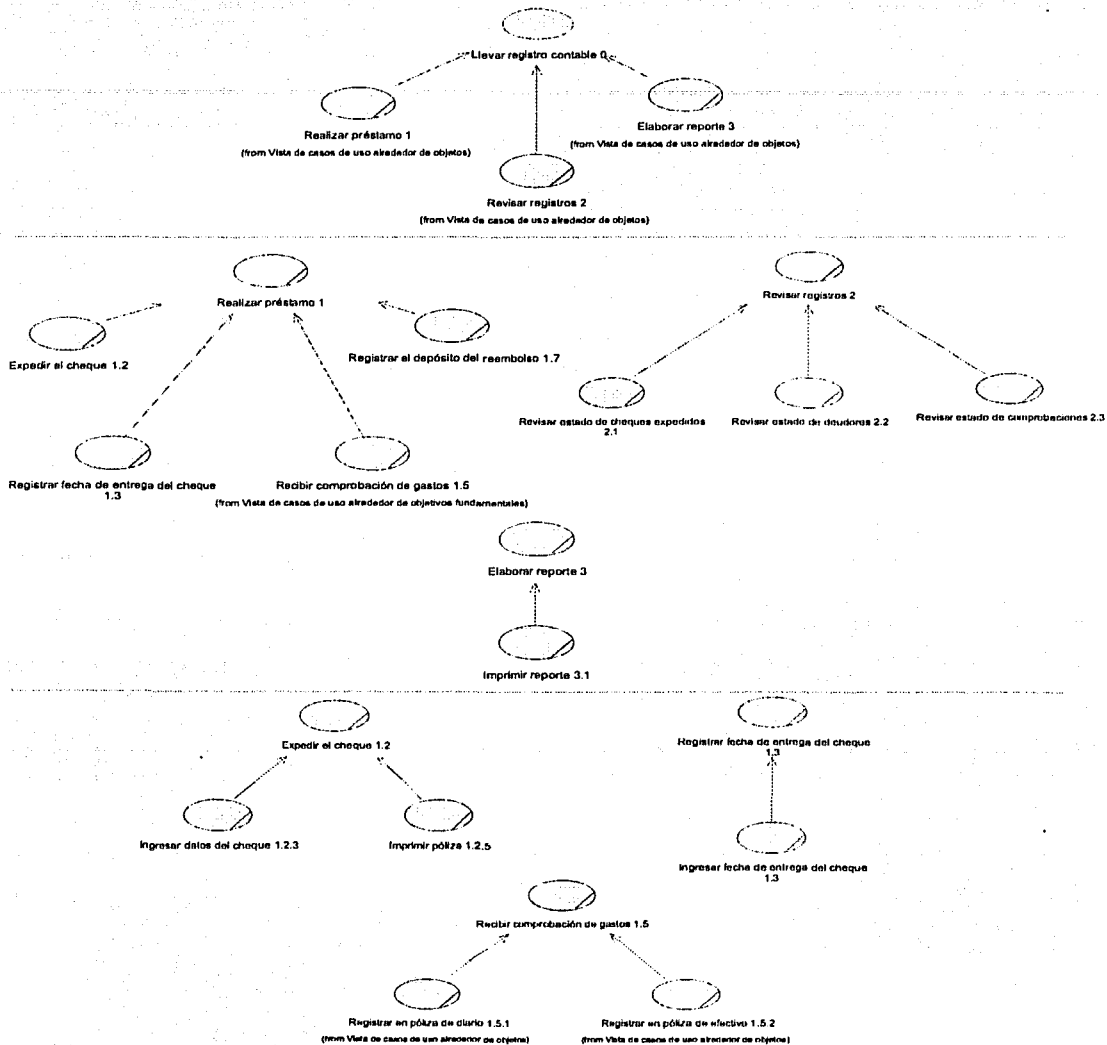
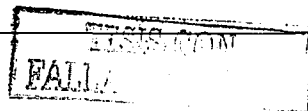


Figura 5.10: Casos de uso preliminares tal y como el usuario realiza su tarea.



5.3.8. Mapeo de la tarea a casos de uso preliminares alrededor de acciones ejecutadas en la tarea

En la tarea del usuario se identifican tres acciones que se realizan en diferentes subtareas y bajo diferentes circunstancias:

1. “*Elaborar*”, en donde se crea algo desde el principio.
2. “*Revisar*”, con la cual se corrobora la veracidad de lo hecho.
3. “*Registrar*”, en donde se registra algún evento.

Se analizó también la opción de proponer acciones más específicas, pero se encontró que si no se tomaban las acciones tal y como están en el árbol de la tarea, en cuyo caso se obtiene un diagrama de casos de uso con poca estructura, se tendrían tareas que no se podrían asociar a ninguna acción. Es por eso que se optó por usar acciones más generales y que puedan abarcar a varias tareas.

Tratando de categorizar el árbol por acciones tenemos:

- “*1 Realizar préstamo*” corresponde a “*Elaborar*”, pues se refiere a iniciar o crear un nuevo proceso de préstamo.
- “*1.2 Expedir cheque*”, “*1.2.3 Ingresar datos del cheque*” y “*1.2.5 Imprimir póliza*” se asocian a “*Elaborar*”, aunque las tareas —sobre todo “*1.2.3*”— no están de forma clara relacionadas con esta acción.
- “*1.3 Registrar fecha de entrega del cheque*” y “*1.3.3 Ingresar fecha de entrega del cheque*” se asocian a “*Registrar*”.
- “*1.4 Comprobar que el cheque fue cobrado*” se asocia a “*Revisar*”.
- “*1.5 Recibir comprobación de gastos*” se asocia a “*Registrar*”, al igual que sus subtareas “*1.5.1 Registrar en póliza de diario*” y “*1.5.2 Registrar en póliza de efectivo*”.
- “*1.6 Requisar solicitud de pago*” es asociada con “*Elaborar*”, aunque la relación no es tan fuerte: La requisición involucra la elaboración de un documento.
- “*1.7 Registrar el depósito del reembolso*” se asocia con “*Registrar*”.
- “*2 Revisar registros*” está asociada claramente con “*Revisar*”. De igual forma, todas sus subtareas son de revisión primordialmente: “*2.1 Revisar estado de cheques expedidos*”, “*2.2 Revisar estado de deudores*”, “*2.3 Revisar estado de requisiciones*” y “*2.4 Conciliar datos con estado de cuenta*”.
- “*3 Elaborar reporte*” se asocia a “*Elaborar*”, al igual que su subtask “*3.1 Imprimir reporte*”.

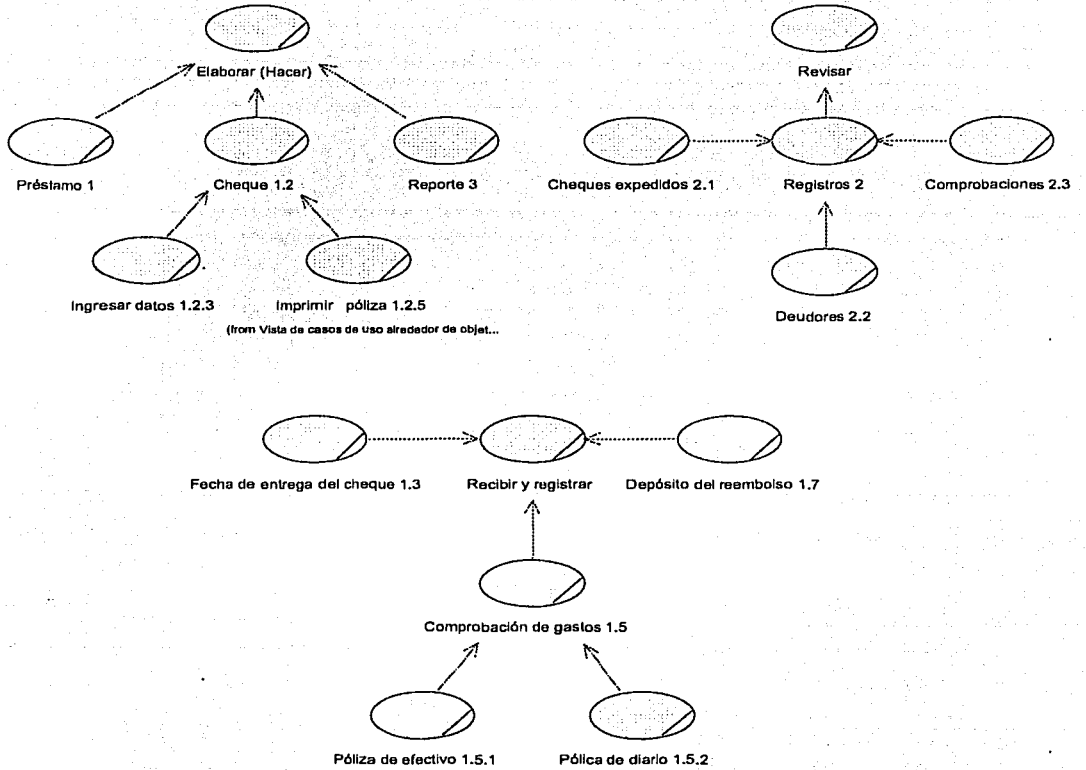


Figura 5.11: Casos de uso preliminares alrededor de las acciones ejecutadas en la tarea del usuario.

TESIS CON
FALLA DE ORDEN

5.4. Discusión

El ejercicio realizado en este Capítulo nos permite evaluar la propuesta de integración del AT al PU, así como los beneficios obtenidos. La nueva actividad, "Realizar análisis de la tarea del usuario" tomó tan sólo cinco entrevistas que suman no más de cuatro horas, recavándose suficiente información para elaborar el árbol de la tarea, sus fichas y encontrar sus objetos. En conjunto, las entrevistas y el AT contenían tanta información que no fue necesario volver realizar nuevas entrevistas para el resto de las actividades.

La lista de requisitos candidatos se tuvo de forma casi directa, pues el AT y las entrevistas proporcionaron el conocimiento necesario sobre el problema del cliente y de su ámbito. De forma similar, el glosario de términos se obtuvo sin dificultad pues, dado que en el árbol de la tarea se emplean las expresiones y términos del usuario, el vocabulario a documentar resultaba familiar.

Sin embargo, la documentación del árbol de la tarea y del árbol de la tarea intermedia resultó algo tediosa debido a la falta de herramientas para procesar este tipo de datos. Para el propósito se tuvo que emplear una herramienta de graficación con la cual había que dibujar cada tarea y colocar sus propiedades manualmente. Es por eso que la transición del árbol de la tarea al árbol intermedio no fue inmediata, se empleó más tiempo resolviendo problemas técnicos que analizando la información.

Haciendo a un lado el problema de la documentación de los árboles de la tarea, el paso del árbol de la tarea del usuario al del sistema con el método planteado en el capítulo 4 fue sencillo. Además, nuevamente, el conocimiento adquirido con el AT facilitó la toma de decisiones respecto a qué tareas no incluir, pues de antemano se tenía conocimiento de la complejidad de cada tarea.

El punto más interesante surgió en la elaboración de los casos de uso preelminares a partir de la tarea modificada. Para cada enfoque se obtuvieron ventajas, pero también se encontraron dificultades.

En el caso de agrupar los casos de uso alrededor de los objetivos fundamentales, se identificaron los siguientes inconvenientes:

- No es fácil encontrar todos los objetivos fundamentales, y si no se incluyen todos, pueden quedar tareas (casos de uso) sin asignar.
- El diagrama preliminar de casos de uso no es necesariamente correcto, pues el usuario puede no identificarse con objetivos fundamentales demasiado abstractos.

A favor, la agrupación alrededor de objetivos fundamentales tiene que los casos de uso quedan muy bien identificados respecto de qué propósito tienen en el sistema y cómo apoyarán a la tarea del usuario.

Para la agrupación de casos de uso alrededor de objetos, se encontraron las siguientes dificultades:

- Los objetos encontrados están a distintos niveles jerárquicos, tal como "Cheque" y "Préstamo", en donde una entidad contiene a la otra (el cheque forma parte de un préstamo); por lo que una tarea (caso de uso) puede ser asociada a varios objetos. El analista tiene que decidir cuál es el "mejor" lugar para ésta.
- Tareas que corresponden a una misma rama del árbol pueden quedar en agrupaciones distintas. Inclusive, una tarea y sus subtareas pueden ser separadas y perder la estructura jerárquica. Sería necesario hacer más aplicaciones a casos de estudio para evaluar el impacto de esto.

TESIS CON
FALLA DE ORIGEN

Resumen		
Mapeo alrededor de	Ventajas	Desventajas
Objetivos fundamentales	- Los casos de uso quedan relacionados de forma clara con el objetivo de negocio que apoyan.	- No es trivial encontrar los objetivos fundamentales. - El usuario puede no identificarse con objetivos fundamentales demasiado abstractos.
Objetos de la tarea	- Agrupación por objetos, usada ampliamente.	- Objetos a distintos niveles jerárquicos. - Se pierde la estructura jerárquica dada por el AT.
Tarea como la ve el usuario	- Conversión directa.	- La organización no es quizás la más adecuada.
Acciones de la tarea		- Decisión arbitraria de acciones. - Decisiones difíciles sobre las asociaciones. - Pueden quedar tareas sin asociación.

Figura 5.12: Resumen de los resultados en la aplicación de cada enfoque para obtener los casos de uso preliminares.

Como ventaja se tiene que todas las tareas quedan asociadas a un objeto, y además, la agrupación de casos de uso por objetos obtenida es muy común y no da malos resultados.

Cuando se agrupan los casos de uso de acuerdo a la forma en que el usuario ve su tarea, la conversión del árbol intermedio al diagrama preliminar de casos de uso es inmediato, más sin embargo, presenta el inconveniente de que la forma en que el usuario percibe su tarea no es necesariamente la mejor forma de documentar la funcionalidad de un sistema.

Finalmente, para el diagrama preliminar de casos de uso agrupados alrededor de acciones, en este ejemplo sólo se encontraron puntos en contra:

- El analista decide arbitrariamente qué acciones deben de usarse.
- Una tarea puede requerir de estar en varias acciones, y el analista tiene que confinarla sólo a una.
- Si las acciones no son propuestas adecuadamente, las tareas pueden quedar fuera, o en una acción que no corresponde del todo.

Los diversos inconvenientes encontrados en cada uno de las propuestas para obtener casos de uso preliminares sugieren que las "recetas" para mapear el árbol de la tarea modificada a diagramas de casos de uso "óptimos" es algo que no es posible, ya que la estructura y enfoque de la tarea cambia de usuario a usuario, de analista a analista, y de problema a problema. Aún con este inconveniente, el avance es importante: Ya se tiene el análisis de la tarea en casos de uso.

1985
FALLA DE ORIGEN

La iniciativa que se debe de tomar es la de estudiar y refinar estas metodologías para obtener diagramas preliminares de casos de uso cada vez más certeros, y que requieran de menor trabajo para obtener los casos de uso finales.

YESSE CON
FALLA DE RESPUESTA

Capítulo 6

Conclusiones

6.1. El uso de teorías que fundamenten las metodologías

El éxito de una metodología en casos reales depende, en su mayor parte, de que los que la usan conozcan y apliquen los principios que la sustentan; la forma de aplicar una técnica debe provenir no de la experiencia, sino de la comprensión de sus fundamentos.

En el Proceso Unificado, las actividades de los flujos de trabajo de Análisis, Diseño, Implementación y Pruebas están muy bien fundamentadas, basándose en flujos de trabajo iterativos e incrementales, y empleando al Caso de uso como la entidad fundamental para la división y administración del trabajo en el proceso de desarrollo. En el PU ([3]), estos principios están claramente explicados e integrados a la metodología de trabajo, por lo que la aplicación a los casos reales ha dado muy buenos resultados.

Sin embargo, el Proceso Unificado basa el flujo de trabajo de “Captura de requisitos” en la experiencia de los analistas, careciendo de una metodología fundamentada. En efecto, se puede constatar (revisar [3], capítulo 6) que el planteamiento de las actividades para este flujo se basa en ejemplos y guías, los cuales los trabajadores pueden malinterpretar. Como se analizó en la sección 1.1.3, el PU no propone un método analítico y procedural para la obtención de los requisitos del cliente y sus usuarios; en sus pasos fundamentales (ver sección 1.1.2), ni siquiera se plantea de qué artefactos se obtendrá la información más importante: los requisitos.

Como resultado, se tiene un proceso de desarrollo que funciona muy bien en lo que respecta a las actividades que típicamente se asocian al desarrollo de un sistema, pero cuya aplicación a proyectos reales presenta problemas no triviales debido a un levantamiento de requerimientos pobre.

En contraste, el Análisis de la Tarea con MAD* funciona muy bien en la captura de requisitos, pues fue concebido pensando en el análisis de los usuarios y de sus requerimientos. Su fundamento, “conoce a tu usuario”, está basado en la no presuposición de requerimientos y en el uso de un modelo, analítico y procedural, que captura información a partir de lo explícitamente planteado y hecho por el usuario.

Sin embargo, el AT es una metodología que no contempla todas las actividades de un proceso de desarrollo. Los distintos esfuerzos que se han hecho para considerarlo como tal, al ser propuestos por gente de Interacción Humano-Computadora, carecen de fundamentos de Ingeniería de Software, por lo que en la práctica no han dado buenos resultados. El problema del uso del Análisis de la Tarea en proyectos de software reales comienza desde su planteamiento como un proceso en cascada, incompatible con los enfoques contemporáneos.

6.2. La aportación al Proceso Unificado

El Proceso Unificado y el Análisis de la Tarea se complementan muy bien, tanto en sus ventajas, como en las teorías que los respaldan, lo cual hace que la propuesta de integración sea potencialmente exitosa en los resultados de su aplicación a proyectos reales.

El Análisis de la Tarea en realidad no se realiza como un proceso en cascada, sino como un conjunto de actividades que se realizan de forma iterativa, trabajando sobre el árbol de la tarea incrementalmente. En efecto, los ergónomos no obtienen el modelo de la tarea del usuario en una sola iteración, sino que es por medio de repetidas entrevistas, análisis, desarrollos y pruebas como se va construyendo el modelo MAD* de la tarea del usuario (ver figura 3.1). Esto permite incluir al AT dentro del PU sin necesidad de modificaciones.

Los Casos de Uso y la Tarea del usuario, entidades fundamentales del PU y del AT respectivamente, no son tan distintos como se pensaba originalmente. De hecho, estos enfoques tienen un punto de cohesión que permite construir un puente entre ellos: El usuario y la interacción que tiene con su sistema. Mientras que el AT estudia la interacción que tiene el usuario con su medio y en particular con sus herramientas durante la realización de su tarea, el PU busca el desarrollo de sistemas de software que apoyen al usuario en su tarea.

Luego, *la funcionalidad del sistema, expresada en casos de uso, se obtiene como consecuencia de los servicios que el sistema tiene que proporcionar al usuario, para que éste pueda realizar su tarea.* Esta relación permite tener una primera versión de los casos de uso a partir del resultado del Análisis de la Tarea, lo cual consideramos es una gran aportación en la resolución del problema de captura de requisitos en el Proceso Unificado.

La adecuación e integración del Análisis de la Tarea al Proceso Unificado, evita los problemas que se han encontrado al tomar el enfoque inverso de extender el AT con un proceso de desarrollo, y aprovecha el éxito que ha tenido el PU en el medio académico y de la comunidad de desarrollo de software.

El empleo del AT en el PU da como resultado proyectos en donde los requerimientos son más claros debido a la comprensión aportada por el modelo de la tarea, y están justificados con argumentos dados explícitamente por el usuario. Los artefactos del Análisis constituyen además una forma muy buena de documentación de las entrevistas y del estado del medio del usuario antes del nuevo sistema.

El valor agregado en la integración, es que ahora se tiene una metodología analítica y procedural para realizar esas actividades del flujo de trabajo de "Captura de Requisitos" que no especificaban cómo generar sus artefactos explícitamente.

La propuesta también considera la situación real de proyectos en donde el usuario plantea requerimientos que no se pueden cubrir debido al presupuesto de tiempo y dinero: Mediante el modelo de la tarea modificada, los miembros del equipo de desarrollo pueden decidir, basados en la complejidad real de la tarea del usuario, qué partes serán apoyadas por el sistema, y cuales no de acuerdo al alcance del proyecto.

El Proceso Unificado de Desarrollo permanece básicamente igual pues no se modificaron ni removieron sus actividades; sólo se agregaron tres nuevas actividades que se realizan de manera previa a las que contiene el PU. De esta forma, los desarrolladores que ya usaban el PU en sus proyectos no tienen que cambiar su forma de trabajo, ni descartar su experiencia, que era lo que se buscaba en los objetivos de este trabajo.

ANÁLISIS CON
DE ORIGEN

6.3. Resultados en el primer caso de estudio

La aplicación de la propuesta al caso de estudio dio resultados satisfactorios, pues la captura de requerimientos se efectuó de forma más rápida y ordenada que en otras situaciones en donde se ha usado el PU solamente. En efecto, aún cuando el analista no tenía ningún conocimiento en contabilidad, las nuevas actividades se realizaron en poco tiempo, dejando un conocimiento profundo y documentado de la tarea del usuario. Esto contrastó con experiencias de proyectos para sistemas que apoyaban tareas más simples, pero en donde la sola realización de las entrevistas tomó varios meses, debido a que se cayó en preguntas recurrentes y a situaciones en donde en etapas avanzadas del proyecto se encontraban dudas importantes.

La realización del análisis de la tarea para un caso con complejidad similar es una actividad que toma menos de una semana. Una vez que se ha comprendido la tarea del usuario, la obtención del árbol de la tarea modificada, y sus subsecuentes diagramas de casos de uso preliminares, puede tomar menos de dos o tres días. Esto muestra que, contrario a una idea generalizada, los tiempos de la captura de requisitos con el Proceso Unificado aumentado pueden ser muy similares a los del PU típico, pero se tienen requisitos más certeros y fundamentados.

En el caso de estudio el analista, después de haber efectuado la captura de requerimientos como en la propuesta, pudo corroborar su conocimiento al intercambiar información con la contadora Isabel Andrade sobre su tarea y necesidades. También se pudo comprobar la eficiencia del modelo MAD* como medio de documentación, al exponerlo en diversos seminarios y observar el rápido entendimiento de los asistentes sobre la tarea del usuario.

Cuando se realizó el primer diagrama de casos de uso, a partir de los diagramas preliminares, cada caso de uso agregado o removido pudo justificarse por lo establecido explícitamente por el usuario.

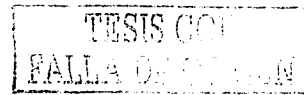
Sin embargo, la desventaja principal en la realización del caso de estudio fue la falta de un sistema que apoyara la elaboración y mantenimiento de los productos del Análisis de la Tarea. La elaboración de los árboles de la tarea del usuario y de la tarea modificada se realizó en aplicaciones de diseño gráfico que hacían difícil la manipulación de los elementos. De igual forma, las fichas de las tareas se mantuvieron con un procesador de texto, lo cual dejaba funcionalidad que desear.

6.4. Los siguientes pasos

Los puntos de cohesión entre el AT y el PU permiten no sólo obtener los casos de uso con ayuda del modelo de la tarea. Los diagramas de interacción y de actividades, así como las interfaces del sistema, pueden obtenerse también de los productos del Análisis de la Tarea, con un enfoque similar al propuesto en este trabajo. Es necesario estudiar estos puntos e integrarlos al PU para apoyar la generación de artefactos como éstos, que típicamente no se realizan en los primeros ciclos del desarrollo debido a la carga de trabajo que implican.

Por supuesto, una evaluación más completa del Proceso aumentado requiere de aplicarlo a varios problemas de la vida real, y comparar los resultados con los de otros sistemas desarrollados bajo el enfoque del PU únicamente, ya que la introducción del Análisis de la Tarea al Proceso Unificado implica cambios en la forma de trabajar, en el tiempo y esfuerzo de las actividades, y en los productos que se obtienen.

La aplicación de la propuesta a varios casos de estudio dará también como resultado



una refinado de los detalles de la integración del AT al Proceso, los cuales no se pueden apreciar con su uso en un solo desarrollo.

La presente propuesta puede aplicarse, de forma recursiva, en el desarrollo de herramientas que soporten el nuevo proceso de desarrollo, de tal forma que se apoye a los desarrolladores en la creación, manejo y mantenimiento de los artefactos introducidos (modelo MAD* de la tarea y diagramas de casos uso preliminares) y facilitar así su tarea.

TESIS CON
FALLA EN

Apéndice A

Fichas del árbol de la tarea del usuario

<i>Núcleo:</i>	
Número	0
Nombre	Llevar registro contable
Objetivo	Evitar pérdidas monetarias y conocer todas las entradas y salidas de dinero.
Prioridad	10
Facultativa	Falso
Interruptionable	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Jefa del área de contabilidad. Encargada de la administración del fondo fijo.
Constructor	Paralelo
<i>Pre y post-condiciones:</i>	
Inicio	
Arranque	
Paro	
Postcondiciones	

TESIS CON
FALTA DE ORDEN

<i>Núcleo:</i>	
Número	1
Nombre	Realizar préstamo
Objetivo	Hacer préstamos monetarios y lograr recibir los pagos a tiempo.
Prioridad	10
Facultativa	Falso
Interruptionable	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Llevar el control del proceso de préstamo, llevando un registro de los eventos importantes.
Constructor	Secuencial
<i>Pre y post-condiciones:</i>	
Inicio	Se tienen recursos en el fondo fijo y se puede hacer uso de ellos
Arranque	Llegó una solicitud de préstamo.
Paro	Se recuperó el dinero prestado.
Postcondiciones	El fondo fijo ha servido a su propósito.

<i>Núcleo:</i>	
Número	2
Nombre	Revisar registros
Objetivo	Encontrar irregularidades en los movimientos de entrada y salida de la institución.
Prioridad	8
Facultativa	Verdadero
Interruptionable	1
Tipo	Cognitiva
Modo	Interactiva
Importancia	Importante
Papel usuario	Revisar la consistencia de los movimientos de su cuenta y de los de las áreas generadoras.
Constructor	Paralelo
<i>Pre y post-condiciones:</i>	
Inicio	No se tiene conocimiento de todos los movimientos pendientes o inconsistentes.
Arranque	
Paro	Se revisaron todos los trámites pendientes y movimientos recientes.
Postcondiciones	Se tiene conocimiento de los movimientos pendientes o inconsistentes.

TESIS COM

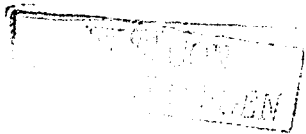
<i>Núcleo:</i>	
Número	3
Nombre	Elaborar reporte
Objetivo	Informar a otra entidad sobre el estado contable actual de la dependencia o del fondo fijo.
Prioridad	10
Facultativa	Falso
Interrumpible	1
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Elaborar y documentar el reporte.
Constructor	Secuencial
<i>Pre y post-condiciones:</i>	
Inicio	No hay un reporte reciente del estado contable.
Arranque	Se le solicitó que elabore un reporte de la contabilidad.
Paro	El reporte está completo y documentado.
Postcondiciones	Se tiene un reporte reciente.

<i>Núcleo:</i>	
Número	1.1
Nombre	Recibir solicitud de cheque
Objetivo	Tener documentación que justifique un préstamo.
Prioridad	10
Facultativa	Falso
Interrumpible	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Recibir la solicitud y verificar que la firma sea autorizada.
Constructor	Elemental
<i>Pre y post-condiciones:</i>	
Inicio	No hay documentación que justifique un nuevo préstamo.
Arranque	Le entregan una solicitud de cheque.
Paro	Se comprobó la validez de la solicitud.
Postcondiciones	Se tienen documentos que justifican la emisión de un préstamo.

TESIS CON
FALLA EN ORIGEN

<i>Núcleo:</i>	
Número	1.2
Nombre	Expedir el cheque
Objetivo	Entregar el cheque del préstamo y tener su póliza
Prioridad	10
Facultativa	Falso
Interrumpible	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Generar el cheque y su póliza y registrarlos en el sistema.
Constructor	Secuencial
<i>Pre y post-condiciones:</i>	
Inicio	Se tienen documentos que justifican la emisión de un préstamo, pero éste no se ha hecho.
Arranque	Se tiene una solicitud de préstamo validada.
Paro	Se ha elaborado el cheque y su póliza.
Postcondiciones	Se tienen los documentos necesarios para efectuar el préstamo.

<i>Núcleo:</i>	
Número	1.2.1
Nombre	Entrar al sistema de caja
Objetivo	Tener en el sistema las herramientas necesarias para registrar movimientos de caja.
Prioridad	10
Facultativa	Falso
Interrumpible	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Usuario de sistema
Constructor	Secuencial
<i>Pre y post-condiciones:</i>	
Inicio	No se pueden registrar movimientos de caja.
Arranque	Se tiene una solicitud de cheque válida.
Paro	Se está en el sistema de caja.
Postcondiciones	Se pueden registrar movimientos de caja.



<i>Núcleo:</i>	
Número	1.2.2
Nombre	Ir a la opción de "Captura de pólizas"
Objetivo	Tener en pantalla la ventana para capturar pólizas.
Prioridad	10
Facultativa	Falso
Interrumpible	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Usuario de sistema
Constructor	Secuencial
<i>Pre y post-condiciones:</i>	
Inicio	Se está en el sistema de caja, pero no en la ventana para capturar pólizas.
Arranque	Se tiene una solicitud de cheque válida.
Paro	Se está en la ventana de captura de pólizas.
Postcondiciones	Se está en el sistema de caja y en la ventana para capturar pólizas.

<i>Núcleo:</i>	
Número	1.2.3
Nombre	Ingresar datos del cheque
Objetivo	Tener en pantalla los datos del cheque y su póliza para luego imprimirlos.
Prioridad	10
Facultativa	Falso
Interrumpible	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Usuario de sistema
Constructor	Secuencial
<i>Pre y post-condiciones:</i>	
Inicio	Se está en la ventana para capturar pólizas.
Arranque	Se tiene una solicitud de cheque que hay que capturar.
Paro	La ventana tiene los datos del cheque, la póliza y el deudor.
Postcondiciones	La ventana de captura tiene todos los campos requeridos para elaborar la póliza.

TESIS CON
FALLA DE ORIGEN

<i>Núcleo:</i>	
Número	1.2.4
Nombre	Guardar el registro
Objetivo	Tener en el sistema los datos de la póliza que se expedirá.
Prioridad	10
Facultativa	Falso
Interruptionable	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Usuario de sistema
Constructor	Elemental
<i>Pre y post-condiciones:</i>	
Inicio	Se está en la ventana para capturar pólizas y no se ha guardado la nueva póliza.
Arranque	Se quiere guardar la póliza en el sistema.
Paro	Se han guardado los datos de la póliza.
Postcondiciones	Se está en la ventana para capturar pólizas y se ha guardado la nueva póliza.

<i>Núcleo:</i>	
Número	1.2.5
Nombre	Imprimir póliza
Objetivo	Tener la póliza y el cheque en papel para ser firmados y entregados.
Prioridad	10
Facultativa	Falso
Interruptionable	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Usuario de sistema
Constructor	Elemental
<i>Pre y post-condiciones:</i>	
Inicio	Se está en la ventana para capturar pólizas y se ha guardado la nueva póliza.
Arranque	Se quiere imprimir la póliza y su cheque.
Paro	Se ha impreso la póliza y su cheque.
Postcondiciones	Se ha impreso la póliza y el cheque del nuevo registro.

IMPRESOS CON
MÁQUINA DE GARCÍA

<i>Núcleo:</i>	
Número	1.3
Nombre	Registrar fecha de entrega del cheque
Objetivo	Tener el comprobante de que el cheque fue recibido
Prioridad	10
Facultativa	Falso
Interruptible	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Recibir la póliza firmada y registrar su fecha.
Constructor	Secuencial
<i>Pre y post-condiciones:</i>	
Inicio	Se ha expedido el cheque y su póliza, pero aún no se tiene conocimiento de cuándo fue recibido.
Arranque	Se recibe la póliza de cheque firmada.
Paro	Se almacenó la póliza firmada y se registró la fecha de recepción del cheque.
Postcondiciones	Se ha expedido el cheque y su póliza, y se tiene la fecha de recepción y el documento comprobatorio.

<i>Núcleo:</i>	
Número	1.3.1
Nombre	Entrar al sistema de caja
Objetivo	Tener en el sistema las herramientas necesarias para registrar movimientos de caja.
Prioridad	10
Facultativa	Falso
Interruptible	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Usuario de sistema
Constructor	Secuencial
<i>Pre y post-condiciones:</i>	
Inicio	No se pueden registrar movimientos de caja.
Arranque	Se tiene la fecha en que se recibió el cheque
Paro	Se está en el sistema de caja.
Postcondiciones	Se pueden registrar movimientos de caja.

TESIS CON
FOLIA DE ORIGEN

<i>Núcleo:</i>	
Número	1.3.2
Nombre	Ir a la opción de "Captura de pólizas"
Objetivo	Tener en pantalla la ventana para capturar y modificar pólizas.
Prioridad	10
Facultativa	Falso
Interruptible	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Usuario de sistema
Constructor	Secuencial
<i>Pre y post-condiciones:</i>	
Inicio	Se está en el sistema de caja, pero no en la ventana para capturar pólizas.
Arranque	Se tiene la fecha en que se recibió el cheque
Paro	Se está en la ventana de captura de pólizas.
Postcondiciones	Se está en el sistema de caja y en la ventana para capturar y modificar pólizas.

<i>Núcleo:</i>	
Número	1.3.3
Nombre	Ingresar fecha de entrega del cheque
Objetivo	Tener en pantalla la fecha de recepción del cheque para luego salvarla.
Prioridad	10
Facultativa	Falso
Interruptible	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Usuario de sistema
Constructor	Secuencial
<i>Pre y post-condiciones:</i>	
Inicio	Se está en la ventana para capturar y modificar pólizas.
Arranque	Se tiene una fecha de entrega de cheque que hay que capturar.
Paro	La ventana tiene los datos de la entrega del cheque.
Postcondiciones	La ventana de captura tiene los datos de la póliza y la fecha en que ésta fue entregada.

<i>Núcleo:</i>	
Número	1.3.4
Nombre	Guardar el registro
Objetivo	Tener en el sistema los datos actualizados de la póliza, incluyendo la fecha de entrega del cheque.
Prioridad	10
Facultativa	Falso
Interruptionable	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Usuario de sistema
Constructor	Elemental
<i>Pre y post-condiciones:</i>	
Inicio	Se está en la ventana para capturar pólizas y no se ha guardado la póliza actualizada.
Arranque	Se quiere guardar la póliza en el sistema.
Paro	Se han guardado los datos de la póliza.
Postcondiciones	Se está en la ventana para capturar pólizas y se ha guardado la póliza actualizada.

<i>Núcleo:</i>	
Número	1.4
Nombre	Comprobar que el cheque fue cobrado
Objetivo	Tener conocimiento de cuándo ya fue cobrado.
Prioridad	10
Facultativa	Falso
Interruptionable	2
Tipo	Cognitiva
Modo	Manual
Importancia	Importante
Papel usuario	Revisar el estado de cuenta y anotar cheques cobrados.
Constructor	Elemental
<i>Pre y post-condiciones:</i>	
Inicio	Se expedió un cheque y el deudor ya firmó de recibido.
Arranque	No se sabe si el cheque ya fue cobrado.
Paro	Se revisó qué cheques ya fueron cobrados.
Postcondiciones	Se tiene conocimiento de si el cheque ha sido cobrado o no.

ESTAS CON
FALLA DE SERVICIO

<i>Núcleo:</i>	
Número	1.5
Nombre	Recibir comprobación de gastos
Objetivo	Tener los documentos necesarios para solicitar el pago del préstamo.
Prioridad	10
Facultativa	Falso
Interruptionable	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Recibir los documentos y comprobar que cubran el total del préstamo.
Constructor	Alternativo
<i>Pre y post-condiciones:</i>	
Inicio	El deudor no ha comprobado los gastos relacionados a su préstamo.
Arranque	Se recibe documento que comprueba un gasto.
Paro	Los documentos recibidos cubren el total del préstamo.
Postcondiciones	El deudor a comprobado los gastos relacionados a su préstamo y es posible solicitar el pago del mismo.

<i>Núcleo:</i>	
Número	1.5.1
Nombre	Registrar en póliza de diario
Objetivo	Registrar que se ha recibido documentación que comprueba un gasto.
Prioridad	10
Facultativa	Falso
Interruptionable	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Usuario de sistema
Constructor	Elemental
<i>Pre y post-condiciones:</i>	
Inicio	Existe un préstamo que no ha sido comprobado
Arranque	Hay documentos que comprueban gastos y que no han sido registrados en el sistema
Paro	Se han registrado los documentos comprobatorios y el préstamo fue comprobado en su totalidad.
Postcondiciones	El préstamo ha sido comprobado.

TRABAJO CON
MÁS DE UNOS

<i>Núcleo:</i>	
Número	1.5.2
Nombre	Registrar en póliza de efectivo
Objetivo	Registrar que se ha recibido efectivo como pago de un préstamo.
Prioridad	10
Facultativa	Falso
Interrumpible	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Usuario de sistema
Constructor	Elemental
<i>Pre y post-condiciones:</i>	
Inicio	Existe un préstamo que no ha sido comprobado
Arranque	Se ha recibido efectivo como pago de un préstamo.
Paro	Se ha registrado la recepción del efectivo y el préstamo fue comprobado en su totalidad.
Postcondiciones	El préstamo ha sido comprobado.

<i>Núcleo:</i>	
Número	1.6
Nombre	Requisitar solicitud de pago
Objetivo	Solicitar el pago del préstamo a la entidad correspondiente.
Prioridad	10
Facultativa	Falso
Interrumpible	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Solicitar verbalmente el reembolso del préstamo y entregar la documentación comprobatoria.
Constructor	Elemental
<i>Pre y post-condiciones:</i>	
Inicio	El préstamo fue comprobado con documentos que no son pagos en efectivo, y aún no se ha hecho el pago del préstamo.
Arranque	Se cuenta con todos los documentos que comprueban el gasto.
Paro	Se han entregado los documentos que requisitan la solicitud.
Postcondiciones	Se ha requisitado la solicitud de pago y se está en espera del mismo.

COPIA DE ORIGEN

Núcleo:	
Número	1.7
Nombre	Registrar el depósito del reembolso.
Objetivo	Tener conocimiento de que se ha recibido el pago.
Prioridad	10
Facultativa	Falso
Interruptionable	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Recibir el comprobante y revisar el movimiento bancario.
Constructor	Elemental
Pre y post-condiciones:	
Inicio	El préstamo fue comprobado con documentos que no son pagos en efectivo, y se elaboró y envió la solicitud de pago.
Arranque	Se recibe la ficha de depósito.
Paro	Se ha comprobado que el pago ha sido efectuado.
Postcondiciones	Se ha recuperado el total del préstamo.

Núcleo:	
Número	2.1
Nombre	Revisar estado de cheques expedidos
Objetivo	Tener conocimiento de qué cheques han sido cobrados y cuáles están pendientes.
Prioridad	8
Facultativa	Verdadero
Interruptionable	2
Tipo	Cognitiva
Modo	Interactiva
Importancia	Importante
Papel usuario	Revisar el estado de cuenta y buscar qué nuevos cobros de cheques ha habido.
Constructor	Elemental
Pre y post-condiciones:	
Inicio	No se sabe qué nuevos cobros de cheques ha habido.
Arranque	Se quiere saber cuál es el monto de cheques por cobrar, y de qué préstamos esperar comprobantes.
Paro	Se revisó a qué cheques corresponden los nuevos cobros.
Postcondiciones	Se sabe qué cheques ya fueron cobrados y cuáles están aún pendientes.

TESIS CON
PALLA DE ORO

<i>Núcleo:</i>	
Número	2.2
Nombre	Revisar estado de deudores
Objetivo	Saber el estado de los préstamos actuales.
Prioridad	8
Facultativa	Verdadero
Interrumpible	1
Tipo	Cognitiva
Modo	Interactiva
Importancia	Importante
Papel usuario	Revisar préstamo por préstamo y ver en qué estado está su comprobación.
Constructor	Elemental
<i>Pre y post-condiciones:</i>	
Inicio	Hay préstamos de los cuales no se sabe qué monto se ha comprobado.
Arranque	Se quiere saber qué préstamos ya se pueden requisitar para reembolso, y a qué dueños es necesario contactar para que entreguen comprobantes.
Paro	Se revisaron todos los préstamos y sus respectivas comprobaciones.
Postcondiciones	Se sabe en qué estado se encuentra cada préstamo.

<i>Núcleo:</i>	
Número	2.3
Nombre	Revisar estado de requisiciones
Objetivo	Saber el estado de las requisiciones en curso.
Prioridad	8
Facultativa	Verdadero
Interrumpible	1
Tipo	Cognitiva
Modo	Interactiva
Importancia	Importante
Papel usuario	Revisar requisición por requisición y ver cuáles ya fueron pagas y cuáles faltan.
Constructor	Elemental
<i>Pre y post-condiciones:</i>	
Inicio	Hay requisiciones de las cuales no se sabe cuándo van a ser reembolsadas.
Arranque	Se quiere saber para cada requisición cuándo fue entregada y si ya fue reembolsado el monto.
Paro	Se revisaron todas las requisiciones y sus respectivos reembolsos.
Postcondiciones	Se sabe qué requisiciones faltan por reembolsar y su estado y cuáles ya fueron pagadas y cuándo.

CON
 ALA DE ORIGEN

<i>Núcleo:</i>	
Número	2.4
Nombre	Conciliar datos con estados de cuenta
Objetivo	Comprobar que los movimientos realizados concuerden con los mostrados en el estado de cuenta y lo reportado por el sistema.
Prioridad	8
Facultativa	Verdadero
Interruptionable	1
Tipo	Cognitiva
Modo	Interactiva
Importancia	Importante
Papel usuario	Comparar los reportes del sistema con los estados de cuenta, y en donde hay diferencia comparar con el comprobante de depósito o retiro.
Constructor	Elemental
<i>Pre y post-condiciones:</i>	
Inicio	No se sabe si los movimientos en la cuenta bancaria concuerdan con lo reportado por el sistema y lo que se tiene en comprobantes.
Arranque	Se tiene el estado de cuenta y se desea saber si todos los movimientos concuerdan con los archivado y capturado.
Paro	Se hizo la comparación, y todas las discrepancias en lo capturado y en lo documentado fueron encontradas y corregidas.
Postcondiciones	Se sabe que el estado de cuenta concuerda con lo que se tiene en el sistema y con lo documentado.

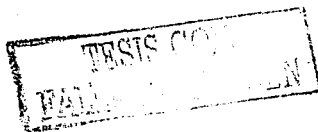
<i>Núcleo:</i>	
Número	3.1
Nombre	Imprimir reporte
Objetivo	Imprimir el reporte de la contabilidad.
Prioridad	10
Facultativa	Falso
Interruptionable	1
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Ir a opciones de reportes y seleccionar la impresión del adecuado.
Constructor	Elemental
<i>Pre y post-condiciones:</i>	
Inicio	No se tiene un reporte impreso.
Arranque	Se requiere un reporte de la contabilidad y no se tiene uno impreso.
Paro	Se tiene el reporte actual impreso.
Postcondiciones	Se tiene el reporte actual impreso.

ANÁLISIS CON
ORIGEN

<i>Núcleo:</i>	
Número	3.2
Nombre	Llenar formato UNAM
Objetivo	Llenar el formato para reportes de la UNAM.
Prioridad	10
Facultativa	Falso
Interruptionable	1
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Llenar a mano el reporte de la UNAM, con los datos obtenidos del reporte impreso.
Constructor	Elemental
<i>Pre y post-condiciones:</i>	
Inicio	Se tiene el reporte impreso pero no el formato lleno.
Arranque	Se requiere un reporte de la contabilidad y no se tiene la forma llena.
Paro	Se tiene el formato de la UNAM llenado.
Postcondiciones	Se tiene el reporte impreso y la forma de la UNAM.

<i>Núcleo:</i>	
Número	3.3
Nombre	Anexar documentación comprobatoria
Objetivo	Incluir en el reporte toda la documentación (estados de cuenta bancarios, recibos, préstamos) que respalda al reporte de contabilidad.
Prioridad	10
Facultativa	Falso
Interruptionable	1
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Para cada uno de los movimientos en el reporte impreso, anexar los documentos que lo respaldan.
Constructor	Elemental
<i>Pre y post-condiciones:</i>	
Inicio	Se tienen el reporte impreso y su formato, pero no los documentos comprobatorios.
Arranque	Se requiere un reporte de la contabilidad y no se tiene a la mano la documentación que respalda el reporte de contabilidad.
Paro	Se tiene la documentación comprobatoria ordenada de acuerdo al reporte.
Postcondiciones	Se tiene listo el reporte de contabilidad.

TESIS CON
FALLA DE URGEN



Apéndice B

Fichas con objetos del árbol de la tarea del usuario

Objetos de la tarea y sus atributos:

- Registro del sistema
 - Irregular
 - Revisado
 - Actualizado
 - Póliza de diario
 - Póliza de efectivo
 - Dependencia
 - Cheques por cobrar
 - Préstamos por comprobar
 - Requisiciones por reembolsar
 - Contabilidad conciliada
 - Deudor
 - Préstamo
 - Nuevo
 - Reembolsado
 - Cheque
 - Datos
 - Firma
 - Monto
 - Impreso
 - Recibido
 - Cobrado
 - Póliza
 - ◊ Firma
 - ◊ Fecha de entrega
 - ◊ Guardada
 - ◊ Impresa
 - Documentación comprobatoria
 - Nueva
- Documentos
- Efectivo
- Total
- Solicitud de pago
 - Requisitado
- Ficha de depósito
- Reembolsado
- Fondo fijo
 - Fondos partida X
- Solicitud de cheque
 - Válida
 - Nueva
- Sistema de caja
 - Ventana de "Captura de póliza"
 - Datos de cheque
 - Datos de póliza
 - Fecha de entrega del cheque
- Reporte
 - Actual
 - Requerido
 - Documentado
 - Impreso
 - Formato UNAM
-



<i>Núcleo:</i>	
Número	0
Nombre	Llevar registro contable
Objetivo	Evitar pérdidas monetarias y conocer todas las entradas y salidas de dinero.
Prioridad	10
Facultativa	Falso
Interrumpible	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Jefa del área de contabilidad. Encargada de la administración del fondo fijo.
Constructor	Paralelo
<i>Pre y post-condiciones:</i>	
Inicio	
Arranque	
Paro	
Postcondiciones	

TESIS CON
FALLA DE ORIGEN

<i>Núcleo:</i>	
Número	1
Nombre	Realizar préstamo
Objetivo	Hacer préstamos monetarios y lograr recibir los pagos a tiempo.
Prioridad	10
Facultativa	Falso
Interrumpible	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Llevar el control del proceso de préstamo, llevando un registro de los eventos importantes.
Constructor	Secuencial
<i>Pre y post-condiciones:</i>	
Inicio	Fondo fijo.Fondos partida X≠0
Arranque	Solicitud de cheque.Nueva=true
Paro	Préstamo.Reembolsado=true
Postcondiciones	Fondo fijo.Fondos partida X≠0

<i>Núcleo:</i>	
Número	2
Nombre	Revisar registros
Objetivo	Encontrar irregularidades en los movimientos de entrada y salida de la institución.
Prioridad	8
Facultativa	Verdadero
Interruptionable	1
Tipo	Cognitiva
Modo	Interactiva
Importancia	Importante
Papel usuario	Revisar la consistencia de los movimientos de su cuenta y de los de las áreas generadoras.
Constructor	Paralelo
<i>Pre y post-condiciones:</i>	
Inicio	Registro del sistema.Revisado = false
Arranque	
Paro	Registro del sistema.Actualizado = true
Postcondiciones	Registro del sistema.Irregular = false Registro del sistema.Revisado = true

<i>Núcleo:</i>	
Número	3
Nombre	Elaborar reporte
Objetivo	Informar a otra entidad sobre el estado contable actual de la dependencia o del fondo fijo.
Prioridad	10
Facultativa	Falso
Interruptionable	1
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Elaborar y documentar el reporte.
Constructor	Secuencial
<i>Pre y post-condiciones:</i>	
Inicio	Reporte.Actual=false
Arranque	Reporte.Requerido=true
Paro	Reporte.Documentado=true
Postcondiciones	Reporte.Actual=true

TESIS CON
FALLA DE ORIGEN

<i>Núcleo:</i>	
Número	1.1
Nombre	Recibir solicitud de cheque
Objetivo	Tener documentación que justifique un préstamo.
Prioridad	10
Facultativa	Falso
Interruptionable	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Recibir la solicitud y verificar que la firma sea autorizada.
Constructor	Elemental
<i>Pre y post-condiciones:</i>	
Inicio	Préstamo.Nuevo=false
Arranque	Solicitud de cheque.Nueva=true
Paro	Solicitud de cheque.Válida=true
Postcondiciones	Préstamo.Nuevo=true

TESIS CO
 VALLE DE
 CAJAMARCA

<i>Núcleo:</i>	
Número	1.2
Nombre	Expedir el cheque
Objetivo	Entregar el cheque del préstamo y tener su póliza
Prioridad	10
Facultativa	Falso
Interruptionable	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Generar el cheque y su póliza y registrarlos en el sistema.
Constructor	Secuencial
<i>Pre y post-condiciones:</i>	
Inicio	Préstamo.Nuevo=true
Arranque	Solicitud de cheque.Válida=true
Paro	Préstamo.Cheque.Impreso=true Préstamo.Cheque.Póliza.Impresa=true
Postcondiciones	Préstamo.Cheque.Recibido=false Préstamo.Cheque.Impreso=true Préstamo.Cheque.Póliza.Impresa=true

<i>Núcleo:</i>	
Número	1.2.1
Nombre	Entrar al sistema de caja
Objetivo	Tener en el sistema las herramientas necesarias para registrar movimientos de caja.
Prioridad	10
Facultativa	Falso
Interruptionable	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Usuario de sistema
Constructor	Secuencial
<i>Pre y post-condiciones:</i>	
Inicio	Sistema de caja=false
Arranque	Solicitud de cheque.Válida=true
Paro	Sistema de caja=true
Postcondiciones	Sistema de caja=true

TESTS CON
FALLA DE ORIGEN

<i>Núcleo:</i>	
Número	1.2.2
Nombre	Ir a la opción de "Captura de pólizas"
Objetivo	Tener en pantalla la ventana para capturar pólizas.
Prioridad	10
Facultativa	Falso
Interruptionable	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Usuario de sistema
Constructor	Secuencial
<i>Pre y post-condiciones:</i>	
Inicio	Sistema de caja=true Sistema de caja.Ventana de Captura de Póliza=false
Arranque	Solicitud de cheque.Válida=true
Paro	Ventana de Captura de Póliza=true
Postcondiciones	Sistema de caja=true Sistema de caja.Ventana de Captura de Póliza=true

<i>Núcleo:</i>	
Número	1.2.3
Nombre	Ingresar datos del cheque
Objetivo	Tener en pantalla los datos del cheque y su póliza para luego imprimirlos.
Prioridad	10
Facultativa	Falso
Interruptionable	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Usuario de sistema
Constructor	Secuencial
<i>Pre y post-condiciones:</i>	
Inicio	Sistema de caja.Ventana de Captura de Póliza=true
Arranque	Solicitud de Cheque.Válida=true
Paro	Sistema de caja.Ventana de Captura de Póliza. Datos de cheque=true Sistema de caja.Ventana de Captura de Póliza. Datos de póliza=true
Postcondiciones	Sistema de caja.Ventana de Captura de Póliza. Datos de cheque=true Sistema de caja.Ventana de Captura de Póliza. Datos de póliza=true

<i>Núcleo:</i>	
Número	1.2.4
Nombre	Guardar el registro
Objetivo	Tener en el sistema los datos de la póliza que se expedirá.
Prioridad	10
Facultativa	Falso
Interruptionable	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Usuario de sistema
Constructor	Elemental
<i>Pre y post-condiciones:</i>	
Inicio	Sistema de caja.Ventana de Captura de Póliza. Datos de cheque=true Sistema de caja.Ventana de Captura de Póliza. Datos de póliza=true Préstamo.Cheque.Póliza.Guardada=false
Arranque	Sistema de caja.Ventana de Captura de Póliza. Datos de cheque=true Sistema de caja.Ventana de Captura de Póliza. Datos de póliza=true Préstamo.Cheque.Póliza.Guardada=false
Paro	Préstamo.Cheque.Póliza.Guardada=true
Postcondiciones	Sistema de caja.Ventana de Captura de Póliza=true Préstamo.Cheque.Póliza.Guardada=true

<i>Núcleo:</i>	
Número	1.2.5
Nombre	Imprimir póliza
Objetivo	Tener la póliza y el cheque en papel para ser firmados y entregados.
Prioridad	10
Facultativa	Falso
Interruptionable	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Usuario de sistema
Constructor	Elemental
<i>Pre y post-condiciones:</i>	
Inicio	Sistema de caja.Ventana de Captura de Póliza =true Préstamo.Cheque.Póliza.Guardada=true
Arranque	Sistema de caja.Ventana de Captura de Póliza =true Préstamo.Cheque.Póliza.Guardada=true
Paro	Préstamo.Cheque.Impreso=true Préstamo.Cheque.Póliza.Impresa=true
Postcondiciones	Préstamo.Cheque.Impreso=true Préstamo.Cheque.Póliza.Impresa=true Préstamo.Cheque.Póliza.Guardada=true

TRIS CON
FALSO DE ORIGEN

<i>Núcleo:</i>	
Número	1.3
Nombre	Registrar fecha de entrega del cheque
Objetivo	Tener el comprobante de que el cheque fue recibido
Prioridad	10
Facultativa	Falso
Interruptionable	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Recibir la póliza firmada y registrar su fecha.
Constructor	Secuencial
<i>Pre y post-condiciones:</i>	
Inicio	Préstamo.Cheque.Recibido=true Préstamo.Cheque.Póliza.Fecha de entrega=null
Arranque	Préstamo.Cheque.Póliza.Firmada=true
Paro	Préstamo.Cheque.Póliza.Fecha de entrega≠null
Postcondiciones	Préstamo.Cheque.Recibido=true Préstamo.Cheque.Póliza.Fecha de entrega ≠ null

<i>Núcleo:</i>	
Número	1.3.1
Nombre	Entrar al sistema de caja
Objetivo	Tener en el sistema las herramientas necesarias para registrar movimientos de caja.
Prioridad	10
Facultativa	Falso
Interruptible	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Usuario de sistema
Constructor	Secuencial
<i>Pre y post-condiciones:</i>	
Inicio	Sistema de caja=false
Arranque	Préstamo.Cheque.Recibido=true
Paro	Sistema de caja=true
Postcondiciones	Sistema de caja=true

TESIS CON
FALLA DE ORIGEN

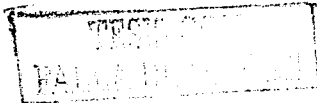
<i>Núcleo:</i>	
Número	1.3.2
Nombre	Ir a la opción de "Captura de pólizas"
Objetivo	Tener en pantalla la ventana para capturar y modificar pólizas.
Prioridad	10
Facultativa	Falso
Interruptible	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Usuario de sistema
Constructor	Secuencial
<i>Pre y post-condiciones:</i>	
Inicio	Sistema de caja=true Sistema de caja.Ventana de Captura de Póliza=false
Arranque	Préstamo.Cheque.Recibido=true
Paro	Ventana de Captura de Póliza=true
Postcondiciones	Sistema de caja=true Sistema de caja.Ventana de Captura de Póliza=true

<i>Núcleo:</i>	
Número	1.3.3
Nombre	Ingresar fecha de entrega del cheque
Objetivo	Tener en pantalla la fecha de recepción del cheque para luego salvarla.
Prioridad	10
Facultativa	Falso
Interrumpible	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Usuario de sistema
Constructor	Secuencial
<i>Pre y post-condiciones:</i>	
Inicio	Sistema de caja.Ventana de Captura de Póliza =true Sistema de caja.Ventana de Captura de Póliza. Fecha de entrega del cheque= null
Arranque	Préstamo.Cheque.Recibido =true
Paro	Sistema de caja.Ventana de Captura de Póliza. Fecha de entrega del cheque / null
Postcondiciones	Sistema de caja.Ventana de Captura de Póliza =true Sistema de caja.Ventana de Captura de Póliza. Fecha de entrega del cheque / null Préstamo.Cheque.Póliza.Guardada = false

<i>Núcleo:</i>	
Número	1.3.4
Nombre	Guardar el registro
Objetivo	Tener en el sistema los datos actualizados de la póliza, incluyendo la fecha de entrega del cheque.
Prioridad	10
Facultativa	Falso
Interrumpible	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Usuario de sistema
Constructor	Elemental
<i>Pre y post-condiciones:</i>	
Inicio	Sistema de caja.Ventana de Captura de Póliza. Fecha de entrega del cheque / null Préstamo.Cheque.Póliza.Guardada =false
Arranque	Sistema de caja.Ventana de Captura de Póliza. Fecha de entrega del cheque / null Préstamo.Cheque.Póliza.Guardada =false
Paro	Préstamo.Cheque.Póliza.Guardada =true
Postcondiciones	Sistema de caja.Ventana de Captura de Póliza. Fecha de entrega del cheque / null Préstamo.Cheque.Póliza.Guardada =true

TESIS CON
FALLA DE OMBEN

<i>Núcleo:</i>	
Número	1.4
Nombre	Comprobar que el cheque fue cobrado
Objetivo	Tener conocimiento de cuándo ya fue cobrado.
Prioridad	10
Facultativa	Falso
Interruptionable	2
Tipo	Cognitiva
Modo	Manual
Importancia	Importante
Papel usuario	Revisar el estado de cuenta y anotar cheques cobrados.
Constructor	Elemental
<i>Pre y post-condiciones:</i>	
Inicio	Registro del sistema.Actualizado=false
Arranque	Préstamo.Cheque.Recibido=true Préstamo.Cheque.Cobrado=null
Paro	Préstamo.Cheque.Cobrado=true o Préstamo.Cheque.Cobrado=false
Postcondiciones	Registro del sistema.Actualizado=true



<i>Núcleo:</i>	
Número	1.5
Nombre	Recibir comprobación de gastos
Objetivo	Tener los documentos necesarios para solicitar el pago del préstamo.
Prioridad	10
Facultativa	Falso
Interruptionable	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Recibir los documentos y comprobar que cubran el total del préstamo.
Constructor	Alternativo
<i>Pre y post-condiciones:</i>	
Inicio	Préstamo.Documentación comprobatoria=false
Arranque	Préstamo.Documentación comprobatoria.Nueva=true
Paro	Préstamo.Documentación comprobatoria.Total= Préstamo.Cheque.Monto
Postcondiciones	Préstamo.Documentación comprobatoria=true

<i>Núcleo:</i>	
Número	1.5.1
Nombre	Registrar en póliza de diario
Objetivo	Registrar que se ha recibido documentación que comprueba un gasto.
Prioridad	10
Facultativa	Falso
Interrumpible	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Usuario de sistema
Constructor	Elemental
<i>Pre y post-condiciones:</i>	
Inicio	Préstamo.Documentación comprobatoria=false
Arranque	Préstamo.Documentación comprobatoria.Nueva=true Préstamo.Documentación comprobatoria.Documentos=true
Paro	Préstamo.Documentación comprobatoria.Total= Préstamo.Cheque.Monto
Postcondiciones	Préstamo.Documentación comprobatoria=true

TESIS CON
FALLA DE ORIGEN

<i>Núcleo:</i>	
Número	1.5.2
Nombre	Registrar en póliza de efectivo
Objetivo	Registrar que se ha recibido efectivo como pago de un préstamo.
Prioridad	10
Facultativa	Falso
Interrumpible	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Usuario de sistema
Constructor	Elemental
<i>Pre y post-condiciones:</i>	
Inicio	Préstamo.Documentación comprobatoria=false
Arranque	Préstamo.Documentación comprobatoria.Nueva=true Préstamo.Documentación comprobatoria.Efectivo=true
Paro	Préstamo.Documentación comprobatoria.Total= Préstamo.Cheque.Monto
Postcondiciones	Préstamo.Documentación comprobatoria=true

<i>Núcleo:</i>	
Número	1.6
Nombre	Requisitar solicitud de pago
Objetivo	Solicitar el pago del préstamo a la entidad correspondiente.
Prioridad	10
Facultativa	Falso
Interruptionable	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Solicitar verbalmente el reembolso del préstamo y entregar la documentación comprobatoria.
Constructor	Elemental
<i>Pre y post-condiciones:</i>	
Inicio	Préstamo.Reembolsado=false Préstamo.Solicitud de pago=false
Arranque	Préstamo.Documentación comprobatoria.Documentos=true
Paro	Préstamo.Solicitud de pago.Requisitada=true
Postcondiciones	Préstamo.Reembolsado=false Préstamo.Solicitud de pago=true

<i>Núcleo:</i>	
Número	1.7
Nombre	Registrar el depósito del reembolso.
Objetivo	Tener conocimiento de que se ha recibido el pago.
Prioridad	10
Facultativa	Falso
Interruptionable	0
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Recibir el comprobante y revisar el movimiento bancario.
Constructor	Elemental
<i>Pre y post-condiciones:</i>	
Inicio	Préstamo.Reembolsado=false Préstamo.Solicitud de pago=true
Arranque	Préstamo.Ficha de depósito=true
Paro	Préstamo.Reembolsado=true
Postcondiciones	Fondo fijo.Fondos partida X≠0

<i>Núcleo:</i>	
Número	2.1
Nombre	Revisar estado de cheques expedidos
Objetivo	Tener conocimiento de qué cheques han sido cobrados y cuáles están pendientes.
Prioridad	8
Facultativa	Verdadero
Interrumpible	2
Tipo	Cognitiva
Modo	Interactiva
Importancia	Importante
Papel usuario	Revisar el estado de cuenta y buscar qué nuevos cobros de cheques ha habido.
Constructor	Elemental
<i>Pre y post-condiciones:</i>	
Inicio	Registro del sistema.Revisado = false
Arranque	Dependencia.Cheques por cobrar = ?
Paro	Dependencia.Cheques por cobrar / ? Registro del sistema.Actualizado = true
Postcondiciones	Registro del sistema.Revisado = true

TEST DE
FALLA DE CHEQUE

<i>Núcleo:</i>	
Número	2.2
Nombre	Revisar estado de deudores
Objetivo	Saber el estado de los préstamos actuales.
Prioridad	8
Facultativa	Verdadero
Interrumpible	1
Tipo	Cognitiva
Modo	Interactiva
Importancia	Importante
Papel usuario	Revisar préstamo por préstamo y ver en qué estado está su comprobación.
Constructor	Elemental
<i>Pre y post-condiciones:</i>	
Inicio	Registro del sistema.Revisado=false
Arranque	Dependencia.Préstamos por comprobar = ?
Paro	Dependencia.Préstamos por comprobar ≠ ? Registro del sistema.Actualizado=true
Postcondiciones	Registro del sistema.Revisado=true

<i>Núcleo:</i>	
Número	2.3
Nombre	Revisar estado de requisiciones
Objetivo	Saber el estado de las requisiciones en curso.
Prioridad	8
Facultativa	Verdadero
Interruptionable	1
Tipo	Cognitiva
Modo	Interactiva
Importancia	Importante
Papel usuario	Revisar requisición por requisición y ver cuáles ya fueron pagas y cuáles faltan.
Constructor	Elemental
<i>Pre y post-condiciones:</i>	
Inicio	Registro del sistema.Revisado=false
Arranque	Dependencia.Requisiciones por reembolsar=?
Paro	Dependencia.Requisiciones por reembolsar ≠ ?
Postcondiciones	Registro del sistema.Actualizado=true

<i>Núcleo:</i>	
Número	2.4
Nombre	Conciliar datos con estados de cuenta
Objetivo	Comprobar que los movimientos realizados concuerden con los mostrados en el estado de cuenta y lo reportado por el sistema.
Prioridad	8
Facultativa	Verdadero
Interruptionable	1
Tipo	Cognitiva
Modo	Interactiva
Importancia	Importante
Papel usuario	Comparar los reportes del sistema con los estados de cuenta, y en donde hay diferencia comparar con el comprobante de depósito o retiro.
Constructor	Elemental
<i>Pre y post-condiciones:</i>	
Inicio	Registro del sistema.Revisado=false
Arranque	Dependencia.Contabilidad conciliada=false
Paro	Dependencia.Contabilidad conciliada=true
Postcondiciones	Registro del sistema.Actualizado=true

<i>Núcleo:</i>	
Número	3.1
Nombre	Imprimir reporte
Objetivo	Imprimir el reporte de la contabilidad.
Prioridad	10
Facultativa	Falso
Interruptionable	1
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Ir a opciones de reportes y seleccionar la impresión del adecuado.
Constructor	Elemental
<i>Pre y post-condiciones:</i>	
Inicio	Reporte.Actual = false
Arranque	Reporte.Requerido = true
Paro	Reporte.Impreso = true
Postcondiciones	Reporte.Impreso = true

SE PUEDE CON
Y PARA INFORMACION

<i>Núcleo:</i>	
Número	3.2
Nombre	Llenar formato UNAM
Objetivo	Llenar el formato para reportes de la UNAM.
Prioridad	10
Facultativa	Falso
Interruptionable	1
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Llenar a mano el reporte de la UNAM, con los datos obtenidos del reporte impreso.
Constructor	Elemental
<i>Pre y post-condiciones:</i>	
Inicio	Reporte.Actual = false
Arranque	Reporte.Impreso = true
Paro	Reporte.Formato UNAM = true
Postcondiciones	Reporte.Actual = true Reporte.Documentado = true

<i>Núcleo:</i>	
Número	3.3
Nombre	Anexar documentación comprobatoria
Objetivo	Incluir en el reporte toda la documentación (estados de cuenta bancarios, reembolsos, préstamos) que respalda al reporte de contabilidad.
Prioridad	10
Facultativa	Falso
Interruptionable	1
Tipo	Cognitiva
Modo	Interactiva
Importancia	Muy importante
Papel usuario	Para cada uno de los movimientos en el reporte impreso, anexar los documentos que lo respaldan.
Constructor	Elemental
<i>Pre y post-condiciones:</i>	
Inicio	Se tienen el reporte impreso y su formato, pero no los documentos comprobatorios.
Arranque	Se requiere un reporte de la contabilidad y no se tiene a la mano la documentación que respalda el reporte de contabilidad.
Paro	Se tiene la documentación comprobatoria ordenada de acuerdo al reporte.
Postcondiciones	Se tiene listo el reporte de contabilidad.

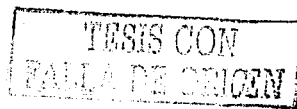


Bibliografía

- [1] F. Gamboa Rodríguez, D. L. Scapin, *Editing MAD* task descriptions for specifying user interfaces, at both semantic and presentation levels*, Institut National de Recherche en Informatique et en Automatique Domaine de Voluceau - Rocquencourt - B. P. 105, 78153 Le Chesnay Cedex, France.
- [2] JoAnn T. Hackos, Janice C. Redish, *User and Task Analysis for Interface Design*, Wiley Computer Publishing, John Wiley & Sons, Inc. New York, 1998.
- [3] Ivar Jacobson, Grady Booch, James Rumbaugh, *El Proceso Unificado de Desarrollo de Software*, Pearson Educación, S. A., Madrid, 2000.
- [4] Scott W. Ambler & Larry L. Constantine compiling editors, *The Unified Process Inception Phase*, CMP Books, Lawrence, Kansas.
- [5] Renardus M. M. Scheffer, *Task Analysis for Interactive System and Service Design*, KPN Research, Department of Service Development & Support, Groningen, 1995.
- [6] Boris Beizer, *Software is Different*, SQP 1, 1998, ASQ.
- [7] Andy Whitefield, Becky Hill, *Comparative Analysis of Task Analysis Products*, Interacting with Computers, Vol. 6, No. 3, 1994.
- [8] Brad A. Myers, *Why are Human-Computer Interfaces Difficult to Design and Implement?*, Computer Science Department, Carnegie Mellon University, Pittsburgh, 1993.
- [9] Don Norman, Apple Computer Inc., Electronic bulletin board posting, 1993.
- [10] Don Norman, *Cognitive engineering*, User centered system design: new perspectives on human-computer interaction, Erlbaum, 1996.
- [11] Kevin Mullet, Darrell Sano, *Designing visual interfaces*, SunSoft Press, Prentice Hall.
- [12] Jenny Prece, *Human-Computer Interaction*, Addison-Wesley, 1994.
- [13] J. M. Christian Bastien, Dominique L. Scapin, *Ergonomic Criteria for the Evaluation of Human-Computer Interfaces*, Technical report 156, Institut National de Recherche en Informatique et en Automatique Domaine de Voluceau - Rocquencourt - B. P. 105, 78153 Le Chesnay Cedex, France.
- [14] B. Alonso, *Analyse des tâches entre contrôleurs aérien par le formalisme MAD pour la spécification ergonomique de l'interface*, Thèse d'Ergonomie, Université Paris 5, France.

- [15] Emilio Insfrán, Pedro J. Molina, Soffa Martí, Vicente Pelechano, *Ingeniería de Requisitos aplicada al modelado conceptual de interfaz de usuario*, Dpto. de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, Valencia, España.
- [16] Juan Sánchez Díaz, Rafael Vañó Pérez, Margarita Burbano, *Una Aproximación Metodológica para Obtener Prototipos de Interface de Usuario a partir de Modelos de Requisitos*, Departamentos de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, Valencia, España.
- [17] Jean-Claude Tarby, *The DIANE+ Method*, Computer-Aided Design of User Interfaces, Proceedings of the 2nd International Workshop on Computer-Aided Design of User Interfaces CADUP'96, Presses Universitaires de Namur, Namur, 1996.
- [18] Jean-Claude Tarby, Marie France Barthet, *From Tasks to Objects: an example with Diane+*, TRIGONE Laboratory CUEEP Institute University Lille 1 59655 Villeneuve d'Ascq Cedex France.
- [19] Jean-Claude Tarby, *Merge Tasks and Objects: Dream or Reality ?*, TRIGONE Laboratory CUEEP Institute University Lille 1 59655 Villeneuve d'Ascq Cedex France.
- [20] Peter Johnson, *A DEPT Advanced Design Environment for Prototyping with Task Models*, Proc. ACM Conf. Human Factors in Computing Systems, INTERCHI, 1993.
- [21] F. Bodart, A.-M. Hennebert, J.-M. Leheureux, I. Provot, B. Sacre, J. Vanderdonckt, *Towards a Systematic Building of Software Architectures: the Trident methodological guide*, Proc. of 2nd Eurographics Workshop on Design, Specification, Verification of Interactive Systems DSV-IS'95, Springer-Verlag, 1995.
- [22] Fikes, 1981.
- [23] E. D. Sacerdoti, *A structure for plans and behavior*, Amsterdam: Elsevier Computer Science Library.
- [24] J. C. Sperandio, *Lérgonomie du travail mental*, Paris: Masson, 1988.
- [25] S. Sebillote, *Task analysis and formalization according to MAD: Hierarchical task analysis, method of data gathering and examples of task description*, Amsterdam.
- [26] S. Sebillote, *Méthodologie pour fournir de éléments de spécification d'interfaces à partir de l'analyse des tâches des opérateurs humains*, rapport technique no. 163, Institut National de Recherche en Informatique et en Automatique.
- [27] James Rumbaugh, Ivar Jacobson, Grady Booch. *The Unified Modeling Language Reference Manual*, Addison-Wesley, 1999.
- [28] Jeffrey Rubin, *Handbook of Usability Testing*, Wiley Technical Communication Library, John Wiley & Sons, Inc., 1994.
- [29] ISO/IEC 15504 (SPICE), *An Emerging Standard on Software Process Assessment*.
- [30] Hammond.
- [31] H. Hammouche, *De la modélisation des tâches utilisateurs à la spécification conceptuelle d'interfaces Homme-Machine*, Thèse d'Informatique, Paris VI, France.

- [32] D. L. Scapin, C. et Pierret-Golbreich, *Towards a method for task description: MAD*, Work with Display Unit '89, p. 371-380, Amsterdam, The Netherlands: Elsevier Science Publishers, 1988.
- [33] P. A. Walsh, *Analysis for Task Object Modelling*, Task Analysis for human-computer interaction, 186-209, Ellis Horwood Limited.
- [34] Bonnie E. John, David E. Kieras, *The GOMS Family of User Interface Analysis Techniques*, ACM ToCHI, 1996.
- [35] William M. Newman, Michael G. Lamming, Michael Lamming, *Interactive System Design*, Addison-Wesley, 1995.



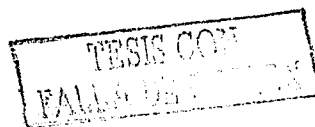




Índice de figuras

1.1. Carga de trabajo típica para cada flujo y por etapa. Ilustración tomada de <i>El Proceso Unificado de Desarrollo de Software</i> ([3]).	5
1.2. Diagrama de actividades de UML ([27]) que muestra los objetos requeridos para este flujo, así como los objetos obtenidos.	6
1.3. Diagrama del paso de enumerar requisitos candidatos, basado en la descripción y sugerencias que da el PU sobre el mismo.	7
1.4. Diagrama del paso de comprender contexto del sistema. El PU no especifica ni cómo efectuarlo ni qué artefactos necesitará.	8
1.5. División en actividades de la captura de requisitos funcionales. Fuente: <i>El Proceso Unificado de Desarrollo de Software</i> [3].	9
1.6. Diagrama del paso de "capturar requisitos funcionales".	10
1.7. Diagrama del paso de "capturar requisitos no funcionales".	10
1.8. Si se ve el desarrollo de software como una espiral, en donde el desarrollo sucesivo de nuevas versiones lleva a productos más apogados a las necesidades del usuario, se tiene que el uso del diseño centrado en el usuario puede ahorrar varias repeticiones del proceso de desarrollo.	13
1.9. Descomposición de la tarea de un empleado de mostrador en un video centro. El análisis comienza descomponiendo la tarea principal en tareas, y luego cada tarea en subtareas hasta que cada tarea quede en términos de acciones.	15
1.10. Detalle de una subtarea del árbol de la tarea.	15
2.1. Subárbol MAD* para la tarea "Actualizar clientes" del empleado de mostrador de video centro.	21
2.2. Tarea de "Atender mostrador".	25
2.3. Tarea 1 de "Actualizar clientes".	26
2.4. Tarea 2 de "Actualizar inventario".	27
2.5. Tarea 3 de "Administrar préstamos".	27
2.6. Tarea 4 de "Informar cliente".	28
2.7. Tarea 5 de "Avisar deudores".	28
3.1. Análisis de la Tarea iterativo e incremental.	40
3.2. Actividades del Análisis de la Tarea propuestas, y flujos de trabajo del Proceso Unificado a los que serán asignadas.	41
3.3. Pasos para elaborar los casos de uso a partir del árbol de la tarea del usuario; en cada uno de ellos interviene la experiencia y creatividad del analista.	46
3.4. Conversión de una tarea a un caso de uso.	47

3.5. Ejemplo de agrupación de casos de uso preliminares alrededor de los objetos que afectan en sus fichas. En este caso, se utilizó la jerarquía de las tareas en el árbol para obtener niveles de detalle para cada grupo.	48
4.1. Trabajadores y artefactos para el flujo de "Captura de Requisitos" que integra al Análisis de la Tarea.	52
4.2. Estructura del árbol de la tarea del usuario.	52
4.3. Responsabilidades del analista de la tarea durante el flujo de trabajo de captura de requisitos.	56
4.4. Nuevo flujo de trabajo para la captura de requisitos. Diagrama basado en la figura 7.10 del PU ([3], pág. 136).	58
4.5. Las entradas y los resultados de "Realizar análisis de la tarea del usuario".	59
4.6. Las entradas y los resultados de "Obtener árbol de la tarea modificada".	60
5.1. Documento que resume la información que se obtuvo en la primera cita y en donde se describe el problema.	67
5.2. Árbol de la tarea de la contadora.	69
5.3. Enumeración de requisitos candidatos, propuestos por el usuario y los analistas.	70
5.4. Glosario de términos	72
5.5. Tarea 1 de la tarea modificada	73
5.6. Tarea 2 de la tarea modificada	74
5.7. Árbol de la tarea modificada. Las tareas marcadas con una "X" no se incluyen.	75
5.8. Casos de uso preliminares alrededor de los objetivos fundamentales de la tarea del usuario.	77
5.9. Casos de uso preliminares alrededor de los objetos empleados en la tarea del usuario.	79
5.10. Casos de uso preliminares tal y como el usuario realiza su tarea.	81
5.11. Casos de uso preliminares alrededor de las acciones ejecutadas en la tarea del usuario.	83
5.12. Resumen de los resultados en la aplicación de cada enfoque para obtener los casos de uso preliminares.	85



Índice alfabético

- Árbol de la Tarea, 43, 52
 - Modificada, 46, 55, 59
- Actividad, 5
- Actor, 43
- Análisis de la Tarea, 3, 13, 16
 - Análisis del Conocimiento de la Tarea, 14
 - Cognitivo, 14
 - GOMS, 14
 - Integrado al Proceso Unificado, 57
 - Iterativo e incremental, 39
 - Jerárquico, 14
- Analista de la Tarea, 56
- Artefacto, 5
- Calidad en el desarrollo de software, 2, 3
- Captura de Requisitos, 6
 - con el Análisis de la Tarea, 51
- Casos de Uso, 5, 43
 - Preliminares, 46, 55, 60
 - Alrededor de Acciones, 61
 - Alrededor de Objetivos Fundamentales, 61
 - Alrededor de objetos, 61
 - Como el Usuario ve su Tarea, 62
- Ciclo iterativo e incremental, 5
- Comprender contexto del sistema, 7
- Condiciones de la tarea, 20
- Conoce a tu usuario, 13
- Constructor de la tarea, 21
- Desarrollo centrado en el usuario, 2
- Diagramas de modelo del dominio, 8
- Diagramas de modelo del negocio, 8
- Dificultades en un desarrollo, 1
- Diseño centrado en el usuario, 12
- El software es un producto diferente, 1
- Entrevistas
 - Dirigidas, 20
 - Documentación, 55
 - No dirigidas, 20
- Enumerar requisitos candidatos, 7
- Esbozos de la interfaz, 9
- Fichas de las tareas, 20, 52
- Glosario de términos, 9
- IHC, 2
- Ingeniería de Software, 2, 3
- Integración del Análisis de la Tarea al Proceso Unificado, 51
- Interacción Humano-Computadora, 2
- Interfaz de sistema, 2
 - IS, 2, 3
- Lenguaje Unificado de Modelado, 5
- MAD*, 19, 52
- Núcleo de la tarea, 20
- Proceso Unificado, 3, 4
 - Flujos de trabajo fundamentales, 4
- Requisitos funcionales, 8
- Rol, 5
- Tarea
 - Elemental, 53
 - Modificada, 55
- Tarea del usuario, 15, 43
- UML, 3
- Usabilidad, 2, 12

