



UNIVERSIDAD NACIONAL DE SANTIAGO DEL ESTERO

FACULTAD DE CIENCIAS EXACTAS Y TECNOLOGÍAS



LICENCIATURA EN SISTEMAS DE INFORMACIÓN

“TRABAJO FINAL DE GRADUACIÓN”

**MÉTODO DE ANÁLISIS DERIVADO
DE UNA ESPECIFICACIÓN TEXTUAL.
SU VALIDACIÓN EMPÍRICA**

Autor:

IVANNA PAOLA DEL VALLE MALDONADO WEKID

Profesor Guía:

SILVIA TERESITA ACUÑA CASTILLO

Profesor Asesor:

MABEL DEL VALLE SOSA DE GOLDAR

Julio de 2007

TRABAJO FINAL DE GRADUACIÓN DE LA LICENCIATURA EN SISTEMAS DE INFORMACIÓN

**MÉTODO DE ANÁLISIS DERIVADO DE
UNA ESPECIFICACIÓN TEXTUAL.
SU VALIDACIÓN EMPÍRICA**

Autor:

.....
Ivanna Paola del Valle Maldonado Wekid

Profesor Guía:

.....
Silvia Teresita Acuña Castillo

Profesor Asesor:

.....
Mabel del Valle Sosa de Goldar

* _____ *

Aprobado el día del mes de del año 200..... por
el Tribunal integrado por:

.....
(firma) (firma) (firma)

.....
(aclaración) (aclaración) (aclaración)

PARA KANELA

A DIOS POR SOBRE TODO

*A MIS PADRES Y TÍA POR ESTAR A MI
LADO SIEMPRE*

*A MI DIRECTORA POR TODO LO
APRENDIDO, UN TESORO INCALCULABLE*

*A MI ASESORA POR SU PACIENCIA Y EL
ALIENTO EN CADA PASO*

*A LA ING. BARCHINI POR SU AYUDA
INCONDICIONAL E INVALORABLE*

*A TODOS AQUELLOS QUE ME AYUDARON A
CONCRETARLO*

MUCHAS GRACIAS!!!

CONTENIDO

RESUMEN	xiii
INTRODUCCIÓN	xv
CAPÍTULO I. PROBLEMA, OBJETIVOS E HIPÓTESIS	1
I.1. OBSERVACIÓN Y DELIMITACIÓN DEL FENÓMENO	3
I.2. PLANTEAMIENTO DEL PROBLEMA	8
I.3. HIPÓTESIS	11
I.4. OBJETIVOS	14
CAPÍTULO II. MARCOS REFERENCIALES	17
II.1. MARCO TEÓRICO	19
II.1.1. MODELOS DE PROCESO SOFTWARE MULTINIVEL.....	19
II.1.2. NIVEL DE INGENIERÍA DEL MODELO DE PROCESO MULTINIVEL	21
II.2. MARCO METODOLÓGICO	28
II.2.1. MODELOS CONCEPTUALES OBTENIDOS POR MATE	28
II.2.2. ELEMENTOS DE LOS MODELOS CONCEPTUALES UTILIZADOS POR FORMOL.....	31
II.2.2.1. PATRONES CONCEPTUALES ESTÁTICOS	32
II.2.2.2. PATRONES CONCEPTUALES DINÁMICOS	33
II.2.3. LENGUAJE NATURAL COMO ENTRADA A MATE.....	34
II.2.4. SUBLEGUAJES UTILIZADOS POR FORMOL.....	37
II.2.4.1. LENGUAJE DE UTILIDAD ESTÁTICO	37
II.2.4.2. LENGUAJE DE UTILIDAD DINÁMICO	39
II.2.5. MÉTODO DE ANÁLISIS DERIVADO DE UNA ESPECIFICACIÓN TEXTUAL.....	41
II.2.5.1. ETAPA 1: EXTRACCIÓN DE LA INFORMACIÓN ESENCIAL.....	44
II.2.5.2. ETAPA 2: IDENTIFICACIÓN DE SINÓNIMOS Y POLISEMAS	47
II.2.5.3. ETAPA 3: SEPARACIÓN DE LA INFORMACIÓN ESTÁTICA Y DINÁMICA.....	48

II.2.5.4. ETAPA 4: ESTRUCTURACIÓN DE LOS REQUISITOS ESTÁTICOS.....	51
II.2.5.5. ETAPA 5: ESTRUCTURACIÓN DE LOS REQUISITOS DINÁMICOS.....	53
II.2.5.6. ETAPA 6: CONSTRUCCIÓN DEL MODELO DE OBJETOS.....	55
II.2.5.7. ETAPA 7: CONSTRUCCIÓN DEL MODELO DE COMPORTAMIENTO	57
II.2.5.8. ETAPA 8: INTEGRACIÓN DEL MODELO DE OBJETOS Y DEL MODELO DE COMPORTAMIENTO	59
II.2.5.9. ETAPA 9: VERIFICACIÓN DE LOS MODELOS DE OBJETOS Y DE COMPORTAMIENTO.....	62
II.2.5.10. RESUMEN DE TODAS LAS ETAPAS DEL MÉTODO MATE.....	62
II.2.6. PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE.....	64
II.2.7. FASE DE ANÁLISIS	74
II.2.8. CONCLUSIONES SOBRE EL PROCESO UNIFICADO	82
CAPÍTULO III. FORMALIZACIÓN DEL MODELO DE PROCESO MULTINIVEL	83
III.1. INTRODUCCIÓN	85
III.1.1. EXTRACCIÓN DE LA INFORMACIÓN ESENCIAL	85
III.1.2. IDENTIFICACIÓN DE SINÓNIMOS Y POLISEMAS	88
III.1.3. SEPARACIÓN DE LA INFORMACIÓN ESTÁTICA Y DINÁMICAS.....	89
III.1.4. ESTRUCTURACIÓN DE LOS REQUISITOS ESTÁTICOS.....	89
III.1.5. ESTRUCTURACIÓN DE LOS REQUISITOS DINÁMICOS.....	95
III.1.6. CONSTRUCCIÓN DEL MODELO DE OBJETOS	101
III.1.7. CONSTRUCCIÓN DEL MODELO DE COMPORTAMIENTO	104
III.1.8. INTEGRACIÓN DEL MODELO DE OBJETOS Y DEL MODELO DE COMPORTAMIENTO	104
III.1.9. VERIFICACIÓN DEL MODELO DE OBJETOS Y DEL MODELO DE COMPORTAMIENTO	107
CAPÍTULO IV. ANÁLISIS E INTERPRETACIÓN DE LOS RESULTADOS.....	109
IV.1. CONSIDERACIONES PREVIAS.....	111
IV.2. ANÁLISIS Y EVALUACIÓN DE LAS VARIABLES RESPUESTAS	113
IV.2.1. VARIABLE RESPUESTA 1: FACILIDAD DE APRENDIZAJE.....	114
IV.2.2. VARIABLE RESPUESTA 2: FACILIDAD DE REPRESENTACIÓN	115
IV.2.3. VARIABLE RESPUESTA 3: IDENTIFICACIÓN DE LOS ELEMENTOS	116
IV.2.4. VARIABLE RESPUESTA 4: TRATAMIENTO DE LA INFORMACIÓN	117

IV.2.5. VARIABLE RESPUESTA 5: UNIFORMIDAD DE LOS MODELOS	118
IV.2.6. VARIABLE RESPUESTA 6: CONSTRUCCIÓN DE LOS MODELOS	119
IV.2.7. VARIABLE RESPUESTA 7: DEFECTOS DETECTADOS	121
IV.2.8. VARIABLE RESPUESTA 8: INTUICIÓN VERSUS GUÍAS	121
IV.2.9. VARIABLE RESPUESTA 9: REPRESENTACIÓN DE LA REALIDAD	123
IV.2.10. VARIABLE RESPUESTA 10: ELECCIÓN DEL MÉTODO.....	123
IV.2.11. SÍNTESIS DE RESULTADOS	125
CONCLUSIONES.....	127
BIBLIOGRAFÍA.....	133
ANEXO A	141
ANEXO B	147
ANEXO C	151
ANEXO D	165

RESUMEN

La importancia de la modelización del proceso software es actualmente aceptada por la comunidad de ingenieros del software. Uno de los principales objetivos de la modelización del proceso es la obtención de modelos conceptuales y formales a partir de las necesidades del usuario del proceso.

En éste trabajo se propone validar empíricamente el Método de Análisis derivado de una Especificación Textual, MATE. Cuya aplicación da como resultado un Modelo de Objetos (estructura del proceso) y un Modelo de Comportamiento (dinámica del proceso). Este método permite llevar a cabo la tarea de análisis del proceso software de manera totalmente formalizada. Para esto se plantea la modelización conceptual de un Modelo de Proceso Multinivel, MPM, mediante la aplicación de MATE. El MPM incluye aspectos organizacionales (procesos, roles, cultura) y de las personas involucradas en el desarrollo de software, normalmente ausentes en los modelos de proceso software tradicionales. Además de la validación, para corroborar la hipótesis se plantea un diseño experimental mediante la aplicación del método MATE y el modelo de análisis del Proceso Unificado en la modelización de ejemplos claves. Éste análisis comparativo se propone para comprobar la efectividad y eficiencia de ambos métodos de análisis.

La especificación formalizada de los aspectos organizacionales provoca que el proceso refleje la situación real del proceso software de la organización modelizada y que se ejecuten procesos adecuados a las necesidades de la organización. El modelo conceptual resultante asegura el recubrimiento y modelización de todos los elementos influyentes del proceso software mediante un método definido y formal, como lo acredita MATE.

El objetivo conceptual de la Teoría de Sistemas es proporcionar un marco para entender y estudiar el comportamiento de los sistemas, diseñando e implementando herramientas que permitan abordar el análisis de los mismos, y los elementos relacionados para dar soporte instrumental en el proceso de construir modelos que representan a cierta realidad.

PALABRAS CLAVE: Método de análisis formal, validación empírica, proceso software, modelización del proceso, modelo de proceso multinivel, ingeniería del software.

INTRODUCCIÓN

*“Un buen libro es aquel que se abre con
expectación y se cierra con provecho”*
Louise May Alcott

MÉTODO DE ANÁLISIS DERIVADO DE UNA ESPECIFICACIÓN TEXTUAL. SU VALIDACIÓN EMPÍRICA

La informática se considera tanto como la ciencia, que trata el estudio de la información, como la ingeniería, que trata automática y sistemáticamente la información aspirando al diseño, desarrollo y ejecución de sistemas informáticos que impactan al medio, permiten el conocimiento y facilitan la toma de decisiones. La informática como disciplina tecnológica transversal ofrece sus métodos y herramientas a las más variadas disciplinas para brindar soluciones a los problemas que afectan a los sistemas organizacionales.

La Teoría de Sistemas, sugiere la existencia de entidades denominadas sistemas. En un sentido más amplio, se refiere a una colección de conceptos generales, principios, instrumentos, problemas, métodos y técnicas relacionados con los sistemas. El objetivo conceptual de la Teoría de Sistemas es proporcionar un marco para entender y estudiar el comportamiento de los sistemas, diseñando e implementando herramientas que permitan abordar el análisis de los mismos, y los elementos relacionados para dar soporte instrumental en el proceso de construir modelos que representan a cierta realidad. El concepto de sistema ha sido ampliamente usado en numerosos campos de aplicación y podría desarrollarse una Teoría de Sistemas para cada sentido que se le otorgue a este término [ULPGC, 99], y en relación con la información, que se captura, almacena, procesa y distribuye, y las tecnologías con las que se cuenta. Los Sistemas de Información (SI) dentro del contexto de las organizaciones constituyen una solución de organización y administración basada en la tecnología de información a los retos y problemas que surgen del medio ambiente [Laudon, 96], como se muestra en la figura 1.

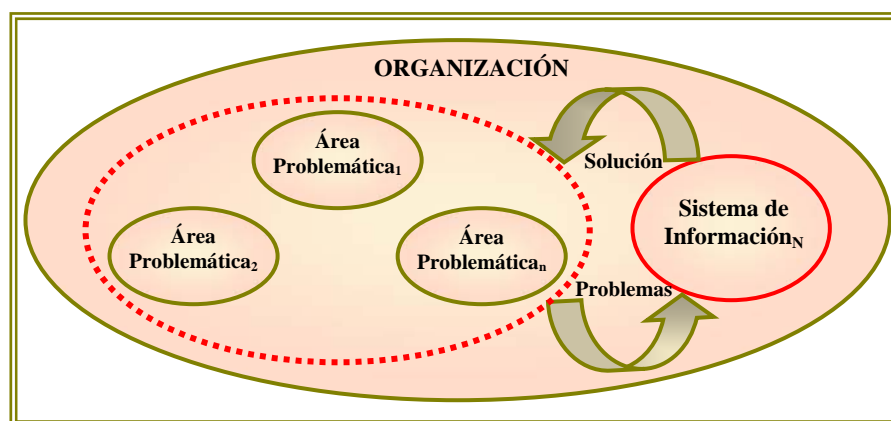


Figura 1. Relación Organización-Sistema de Información

Los SI son parte de las organizaciones [Laudon, 96], cuyos componentes interrelacionados, apoyan la toma de decisiones, el control, análisis y visión de la misma.

El diseño y desarrollo de SI, requiere ser gestionado como un gran proyecto de ingeniería.

Los enfoques sistémicos para el desarrollo y mantenimiento de productos de ingeniería, requieren analizar cuáles son las necesidades de la organización; cómo satisfacer las mismas mediante la aplicación de tecnologías de sistemas software; cómo se van a coordinar todas las personas que van a intervenir en el proceso de producción y cómo controlar el desarrollo en sí mismo, de forma que se obtengan los resultados esperados. Para conducir con éxito este estudio, se debe gestionar un proyecto de construcción de software, el que consistirá en analizar los requerimientos detectados, desarrollar un producto software que los satisfaga y, posteriormente introducir esta solución a la estructura de los procesos del sistema organizacional que lo gesta. A la par que, el proceso software generado se fecunda en una continua relación de reflexividad, ya que tiene por propósito gestionar, transformar y soportar la necesidad de la organización en una solución informatizada que satisface esa necesidad. Estas relaciones se visualizan en la figura 2.



Figura 2. Relación Organización-Proceso Software

En concreto, para solucionar los problemas existentes en cada subsistema, se deben combinar métodos aplicables a cada una de las fases del desarrollo de software, construir herramientas para automatizar estos métodos, utilizar técnicas para garantizar la calidad de los productos desarrollados y coordinar todas las

personas implicadas en el desarrollo de un proyecto, para poder avanzar con eficiencia en la solución de los inconvenientes que soporta el sistema. De ello, se encarga la disciplina llamada Ingeniería del Software (IS), la que ocupa un lugar preponderante en el desarrollo de proyectos de sistemas informáticos de calidad. Las teorías, métodos y herramientas para el desarrollo, administración y evolución de un producto software concierne a esta disciplina, y es en el seno de esta rama de la informática donde se centra este trabajo, y más concretamente, en general, en el proceso de *Análisis de un modelo de proceso* y, en particular, en la *Modelización del proceso software*.

La importancia de la Modelización del proceso software es actualmente aceptada por la comunidad de ingenieros del software. La Modelización del proceso software se encarga de la obtención de modelos del proceso de desarrollo de software conceptuales y formales a partir de las necesidades de la organización. El objetivo último de la modelización del proceso software es guiar, controlar y gestionar las actividades descritas o prescritas por el proceso. Para ello, se hace necesario considerar los entornos organizacionales así como analizar, de modo preciso, las relaciones entre los procesos software modelizados y los entornos organizacionales [Abdel, 91][Seamani, 94][Min, 97]. Esta modelización comienza con el análisis de los requerimientos.

La actividad de Análisis permite que los desarrolladores del software aborden la construcción del sistema, ya que tiene por razón entender y modelizar la necesidad del usuario y su solución. El objetivo primero de la fase de Análisis es la obtención de un modelo conceptual a partir de éstas necesidades.

En el campo de la IS, existen principalmente dos alternativas a aplicar a la hora de realizar el Análisis de un problema en función de la aproximación con que se realiza la representación de los modelos [Meyer 88]. Un ingeniero del software puede centrar su estudio en las funciones que deben realizarse para resolverlo, dando lugar al enfoque estructurado, o en la información que se ha de manejar para resolverlo, dando lugar al enfoque orientado a objetos.

Uno de los modelos conceptuales más flexible y cercano al mundo es el generado mediante la Modelización Orientada a Objetos (MOO). Sin embargo, la falta de formalización en la obtención de dichos modelos supone una importante

limitación. Esta falta de definición es consecuencia de la ausencia métodos formales para la obtención de los componentes de una MOO de un problema, a partir de la especificación de los requisitos del usuario.

En resolución a esta insuficiencia, Moreno [Moreno, 97] diseña un proceso perfectamente definido para realizar el Análisis de la cuestión problemática. La formalización del proceso de Análisis va a permitir determinar la modelización conceptual que representa la situación bajo estudio, de una forma justificada. Privilegiando así, esta modelización frente a las que existirían empíricamente para representar dicho objeto de examen.

Esta tesis plantea la modelización conceptual de un modelo de proceso software [Acuña, 99b]. En 1997, se diseñó el *Modelo de Proceso Multinivel* (MPM) [Acuña, 97], que incorpora aspectos organizacionales (procesos, roles, cultura) y de las personas involucradas en el proceso de desarrollo de software. Este modelo multinivel permite la ingeniería, gestión y realización del proceso software en relación con su ambiente integrado y con una comprensión profunda de que el software resulta de un proceso de comunicación complejo.

El MPM interrelaciona diferentes ambientes: el *ambiente socio-cultural*, el *ambiente científico-tecnológico* y el *ambiente paradigmológico*. El mismo, está formado por los tres ambientes mencionados, por cinco niveles que representan el ciclo del proceso y por las líneas del “qué”, “quién” y “cómo”. Los niveles son: *Nivel de Universo Ecológico*, *Nivel de Ingeniería*, *Nivel de Gestión*, *Nivel de Desarrollo* y *Nivel de Evaluación*. Estos niveles en relación con su ambiente enfatizan el comportamiento humano organizacional, el teológico y decisonal y el de abstracción-modelización, respectivamente. Las líneas del “*qué*”, “*quién*” y “*cómo*” caracterizan los procesos objetos de desarrollo, el perfil de los seres humanos involucrados y el tipo de las herramientas utilizadas.

Específicamente, la finalidad de su Nivel de Ingeniería es, de acuerdo a las necesidades de la organización analizada, diseñar o mejorar modelos de proceso software generales. El mencionado nivel se conforma de dos procesos: el Proceso Organización, que indica cómo modelizar la organización, develar su cultura y determinar las capacidades de sus miembros y el Proceso Ingeniería, que se refiere a cómo representar modelos de procesos software generales que incorporen estos aspectos organizacionales y de las personas involucradas en el desarrollo de software.

Acuña y Barchini [Acuña, 99a][Acuña, 00a] describen en detalle los procesos y sus procedimientos, que permiten la realización del Nivel de Ingeniería del MPM. Para la modelización conceptual, se utiliza el *Método de Análisis derivado de una Especificación Textual* (MATE) [Moreno, 97]. La aplicación de este método permite llevar a cabo la tarea de Análisis del proceso software de manera totalmente formalizada.

El método MATE utiliza técnicas de la Orientación a Objetos (OO) para representar la modelización conceptual del problema, y emplea una base matemática para el estudio de las correspondencias entre los modelos conceptuales generados. La especificación formalizada de los aspectos organizacionales provoca que el proceso refleje la situación real del proceso software de la organización modelizada y que se ejecuten procesos adecuados a las necesidades de la organización. El modelo conceptual resultante asegura el recubrimiento y modelización de todos los elementos influyentes del proceso software mediante un método definido y formal.

La propuesta de este trabajo es presentar y aplicar el método MATE, que proporciona un conjunto de pasos que guían al ingeniero de software durante el proceso de Análisis, desde que éste dispone de la descripción textual de la situación problemática, con las características del problema a resolver, hasta que obtiene los modelos conceptuales de la OO, verificados con dicha descripción textual. Los modelos conceptuales resultantes son el Modelo de Objetos (MO) y el Modelo de Comportamiento (MC), los cuales representan una visión externa del funcionamiento del proceso que se desea modelizar, y un punto de partida para diseñar el proceso software.

Para definir formalmente una correspondencia directa, mediante una relación de equivalencia unívoca, entre el Mundo Lingüístico, en el que están representados los requisitos del usuario, y el Mundo Conceptual, que constituye la salida del proceso de Análisis, se ha empleado un mundo intermedio, el Mundo Matemático¹. De esta forma, se obtienen los componentes clave para una MOO coherente, cuyo modelo formalizado asegure la validez de los objetos y relaciones estudiadas.

¹ Para profundizar sobre el Mundo Matemático que otorga la formalización de MATE como método definido y formal se recomienda la lectura de [Moreno, 97].

En el área de la OO, existen varios métodos pero todos ellos carecen de la madurez necesaria. En 1999, los autores de los tres métodos más usados de la OO: Grady Booch, Ivar Jacobson y Jim Rumbaugh han concebido un método llamado *Proceso de Desarrollo de Software Unificado o Proceso Unificado* (PU) [Jacobson, 00] que está orientado a las personas que realizan las distintas actividades del proceso y tienen una visión tanto técnica como organizacional, de gestión y de soporte del desarrollo de software, de una forma que sea adaptable a un amplio rango de proyectos y organizaciones.

El PU usa UML (del inglés, Unified Modeling Language) [Booch, 99]. UML es el lenguaje que permite modelizar, construir y documentar los elementos que forman un sistema de software orientado a objetos. Este lenguaje de modelización incorpora las principales ventajas de cada uno de los métodos particulares en los que se basa: Booch, OOSE y OMT. De hecho, algunos autores [Acuña, 01][Moreno, 97] afirman que el PU, hoy por hoy es el modelo del proceso de desarrollo de software estándar y más conocido en la comunidad de la IS.

Sobre lo expuesto precedentemente, los objetivos generales se engloban en el siguiente párrafo: *Obtener* modelos conceptuales y formales de un modelo de proceso software coherentes al contexto analizado, *facilitar* su obtención a través de modelos lingüísticos que surjan de la descripción de las necesidades del usuario, para *lograr* una formalización de la OO estático-dinámica conjunta de todos los elementos meritorios de análisis.

Para comprobar la efectividad y eficiencia del método sugerido, MATE, se compara con otro método de la OO como lo es el modelo de análisis del PU [Jacobson, 00]. La experiencia de validación, se realizará con dos muestras, cada una responsables de la evaluación de cada método de análisis que se desea examinar. La evaluación se realizará tomando ejemplos que contienen exigencias sobre las ventajas y desventajas de los mismos. Es decir, se pretende, mediante ejemplos preseleccionados, resaltar particularidades positivas y negativas de cada método, que lo hacen favorable o no dentro de una aplicación. La finalidad de la experimentación que se realiza es comparar, contrastar y examinar ambos métodos respecto a la comprensión y construcción de ejemplos categorizados para su entendimiento y aplicación propuestos, considerando un conjunto de variables respuesta para determinar la bondad de los métodos de análisis utilizados.

Sobre la experimentación propuesta, se refutará o avalará las *hipótesis* de investigación planteadas en esta tesis. La primera, considera si MATE es aplicable a partir de la narración textual de un modelo de proceso software sujeto a observación para la elaboración de los modelos conceptuales. Y la segunda coteja si MATE es altamente poderoso, efectivo y eficiente dentro del proceso de Análisis en relación con el PU.

Finalmente, el MPM formalizado; cuya definición incluye aspectos organizacionales, culturales y de las personas; normalmente, ausentes en los modelos de proceso software actuales, constituye el aporte teórico. La validación del método MATE, como método formal que permite analizar y comprender el comportamiento de los SI garantiza, a partir de su aplicación, modelos conceptuales que reflejan la realidad de manera adecuada. Esta validación constituye el aporte metodológico al campo de los SI. Estas contribuciones, se enfocan con mejor claridad, en la figura 3.

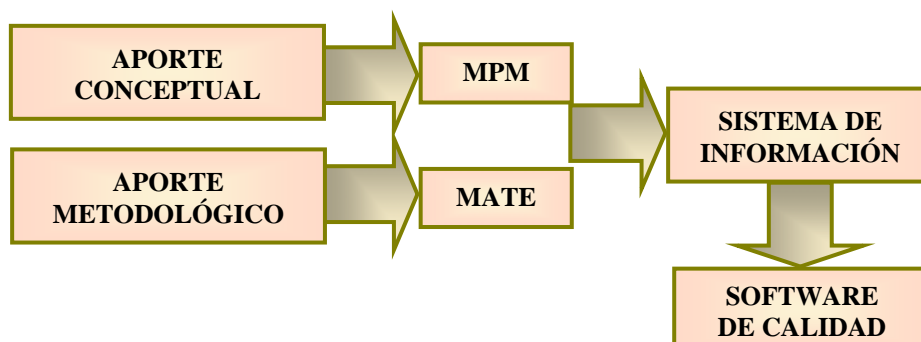


Figura 3. Aportaciones del Trabajo de Graduación

ESTRUCTURACIÓN DEL TRABAJO

El trabajo se estructura de la siguiente forma. En el capítulo primero se indica el problema a solucionar, se plantean los objetivos a alcanzar y las hipótesis a demostrar. En el capítulo segundo se presentan los marcos referenciales. En el tercer capítulo se describe la formalización. En el capítulo cuarto se analizan e interpretan los resultados; y finalmente se publican las conclusiones arribadas a lo largo de todo este trabajo.

CAPÍTULO I. PROBLEMA, OBJETIVOS E HIPÓTESIS

"Solo es útil el conocimiento que nos hace mejores"
Sócrates

I.1. OBSERVACIÓN Y DELIMITACIÓN DEL FENÓMENO

Como ya se ha expresado, esta línea de investigación se encuadra, dentro de la IS, y más concretamente, por un lado, en el proceso de Análisis de un sistema y, por el otro, en la modelización del proceso software. La importancia de la fase de Análisis es crucial en el desarrollo del software, ya que tiene por finalidad entender y modelizar la necesidad del usuario y su solución, a partir de la cual los desarrolladores abordarán la construcción del sistema [Blum, 96].

Dentro de los enfoques más utilizados en IS para abordar la fase de Análisis se consignan: el Análisis Estructurado y el Análisis Orientado a Objetos (AOO). El problema que se pretende abordar en este proyecto, se enmarca en el último, debido a que las técnicas utilizadas por el paradigma de la OO permiten soportar la modelización directa de las entidades del mundo real [Booch, 96]. Esto posibilita que la representación del problema se obtenga de una forma directa y cercana a la realidad, favoreciendo la comunicación con el usuario y facilitando la tarea de Análisis [Sigfried, 96].

La aparición del AOO a comienzos de los años 90 es, sin embargo, reciente. Los principios de la OO se aplicaron progresivamente a los Lenguajes de Programación; a la fase de Diseño; y por último al Análisis, estando todavía en su infancia en esta actividad [Iivari, 91][Honiden, 93][Wang, 97][Northrop, 97]. Durante la fase de AOO se deben plasmar las necesidades de los usuarios y sus requisitos con respecto al producto a desarrollar, en un modelo orientado a objetos. La modelización de un sistema orientado a objetos debe representar el modelo mediante una colección de objetos que exhiben un comportamiento individual y las relaciones entre dichos objetos [Faulk, 97]. La inmadurez del proceso de AOO proviene de la falta de métodos definidos y rigurosos para identificar los componentes anteriores [Argila, 94][Chen, 94][Wang, 97].

Aunque en la actualidad existen métodos que proporcionan un conjunto de pasos definidos para guiar al ingeniero del software durante el Análisis, estos métodos carecen de reglas formales, justificadas, completas y correctas que indiquen cómo obtener los componentes de los modelos conceptuales que representan determinado problema.

Para justificar esta limitación, se pueden revisar brevemente los criterios más relevantes, que siguen los métodos de la OO, para identificar los componentes del sistema. Dentro de la misma, se pueden distinguir dos grupos de métodos en cuanto al proceso de análisis [Moreno, 97]:

- *Métodos Clásicos*, que utilizan guías distintas de la sintaxis y la semántica para identificar los componentes del sistema orientado a objetos, las que distan mucho de constituir procedimientos formalizados y exhaustivos para la obtención de dichos componentes.
- *Métodos Basados en el Estudio de la Descripción Textual*, que identifican los componentes del sistema orientado a objetos basándose en el análisis sintáctico y semántico de las palabras que constituyen la descripción informal del problema a resolver. Los cuales tampoco están formalizados, sino que se limitan a describir heurísticas para identificar estos componentes, o a mostrarlas mediante algún ejemplo.

Dentro de los *Métodos Clásicos*, sirva como muestra de los criterios empleados para la obtención de los componentes de un modelo orientado a objetos, las guías para la obtención de clases y objetos propuestas por sus autores. I. Jacobson [Jacobson, 92] utiliza el comportamiento del sistema como fuente principal de objetos. Así, emplea el concepto de *caso de uso*, que representa cada una de las formas en que puede ser usado el sistema a desarrollar. A partir de los casos de uso se extraen los objetos identificando las entidades que participan en los mismos. S. Shlaer y S. Mellor en su metodología del AOO [Shlaer, 92] proponen que "la mayoría de los objetos caen en una de las siguientes categorías, objetos tangibles, roles, incidentes, interacciones u objetos de especificación que indican reglas, estándares, etc.". E. Yourdon [Yourdon, 95] propone una serie de guías para buscar clases potenciales como son: "procedimientos organizativos, descripciones de productos y servicios, planes, objetivos de negocio, ficheros existentes, aplicaciones existentes, formularios, pantallas, informes, modelos de aplicaciones de áreas de negocio similares, diagramas e información de documentos de requisitos". P. Coad [Coad, 95] propone identificar las clases buscando "cosas tangibles, actores, localizaciones físicas, sucesos recordables, roles jugados,

procedimientos organizacionales y unidades organizativas". G. Booch [Booch, 96] se limita, en este aspecto, a revisar las técnicas que proponen el resto de autores.

Se puede observar que las guías que proporcionan estos métodos distan mucho de constituir procedimientos formalizados y exhaustivos para la obtención de los componentes de una MOO.

Los *Métodos Basados en el Estudio de la Descripción Textual* tampoco están formalizados, sino que se limitan a describir heurísticas para identificar los componentes de un sistema orientado a objetos o a mostrarlas mediante algún ejemplo. Así R. Abbot [Abbot, 83], J. Rumbaugh [Rumbaugh, 91], C. Rolland [Rolland, 92], o E. Buchholz [Buchholz, 96] coinciden en asignar un "nombre común a" una clase; C. Block [Block, 93] y A. Tjoa [Tjoa, 92] relacionan la estructura "es un" con jerarquías de herencia; o F. Tseng [Tseng, 92] y S. Naduri [Naduri, 95] asocian los "verbos con" relaciones entre clases.

En ambos grupos de métodos, generalmente, las asignaciones de los objetos se basan en la intuición, o bien, la experiencia del ingeniero del software y no se define ningún tipo de justificación formal para realizarla. La base intuitiva resulta peligrosa, pues nunca puede asegurarse que los modelos obtenidos sean los que corresponden con las especificaciones de partida. Además, en unas especificaciones de requisitos puede haber mucha información no significativa que debería ser tratada, antes de aplicar los criterios de los autores encontrados en la literatura.

Una minoría de investigadores ha abandonado la intuición para intentar cierta formalización en las asignaciones. Por ejemplo, R. Van de Riet y J.F.M. Burg [Van de Riet, 95][Burg, 97] asignan elementos de un lenguaje de modelización conceptual, a elementos de la modelización conceptual orientada a objetos. Este lenguaje conceptual es próximo al lenguaje natural pero no es directamente lenguaje natural, sino un lenguaje basado en la gramática funcional. La asignación entre ambas representaciones tampoco se justifica formalmente. La aplicación del método está altamente restringida por la transformación de las especificaciones a este lenguaje conceptual y por el uso de un diccionario limitado, denominado Lexicon, que contiene información semántica y sintáctica sobre determinadas estructuras de la lengua inglesa.

Es sustancial mencionar que en los últimos años surgió el PU, una aproximación más formalizada desde la perspectiva de la OO. El PU, si bien, ofrece un conjunto de actividades y roles para llevar a cabo el proceso de modelización del

proceso software, continúa basándose en la apreciación intuitiva para la detección de los elementos claves de la misma.

En resumen, existe un vacío en cuanto a técnicas formales para identificar los componentes de los modelos de la OO de un problema, a partir de unas especificaciones escritas en lenguaje natural. La falta de este tipo de técnicas conlleva una aleatoriedad exacerbada durante el proceso de AOO. Esta ausencia y la necesidad de cubrirla con *un método formalizado y definido*, ha sido resaltado por múltiples autores [Iivari, 91][Honiden, 93][Argila, 94][Chen, 94][Wang, 97].

Estas carencias han llevado a Moreno [Moreno, 97] a la creación del método MATE (definido y formal) para la obtención de los modelos resultantes del Análisis a partir de la información sintáctica y semántica contenida en la declaración informal del problema a analizar, como resultado de su tesis doctoral. Las bondades de éste método se detallan con profundidad en el presente trabajo final de graduación.

Por otra parte, según lo ya manifestado, este estudio se concentra en la modelización del proceso software. El proceso software es un factor crítico en la entrega de sistemas de software de calidad [Lehman, 91][Blum, 96], ya que tiene por finalidad gestionar, transformar y soportar la necesidad del usuario en una solución automatizada que satisface esa necesidad.

Hoy por hoy, un área que juega un papel central en la investigación del proceso software es la modelización del proceso [Tully, 89][Curtis, 92][Finkelstein, 94][McChesney, 95]. La modelización del proceso software se encarga de la creación de modelos del proceso de desarrollo de software y el uso de estos modelos en un IPSE (Entorno Integrado de Soporte del Proyecto, en inglés, Integrated Project Support Environment) [Finkelstein, 94]. Un modelo de proceso software se considera una representación abstracta de la arquitectura, el diseño o la definición del proceso software. Cada una de éstas representaciones describen, a distintos niveles de detalle, una estructura de los elementos del proceso, ya sea de un proceso completado, actual o propuesto. Un modelo de proceso puede analizarse, validarse y, si se ejecuta, se puede simular el proceso modelizado. Los modelos de proceso pueden ser usados para [McChesney, 95]: a) proveer un marco de referencia a fin de comprender, experimentar con, y razonar sobre el proceso; b)

facilitar la automatización del proceso; y c) proporcionar una base para el control del proceso.

Los objetivos específicos y ventajas que presenta la modelización del proceso software son los siguientes [Curtis, 92]:

1. *Facilita la comprensión y la comunicación del proceso.* Para ayudar a comprender y discutir sobre el proceso software se requiere un modelo de proceso, con información suficiente para su representación. Formaliza el proceso, conformando una base para la formación.
2. *Soporta la mejora del proceso.* Para mejorar un proceso software se requiere el reuso de procesos software bien-definidos y efectivos, la comparación de procesos alternativos y el soporte de la evolución del proceso.
3. *Soporta la gestión del proceso.* Para gestionar un proceso software se requiere un proceso software específico del proyecto, sobre el que se producirá seguimiento, gestión y coordinación.
4. *Permite orientaciones automatizadas para la realización del proceso.* En este caso se requiere un entorno de desarrollo de software efectivo, proporcionando orientaciones al usuario, indicaciones y material de referencia.
5. *Permite el soporte de ejecución automatizada.* Para ello se requieren partes automatizadas del proceso, soporte del trabajo cooperativo, recolección de métricas y garantía de la integridad del proceso.

El objetivo último de la modelización del proceso software es orientar, controlar y gestionar las actividades del proceso. Para la orientación, el control y gestión efectiva de las actividades del proceso, los entornos organizacionales deben ser considerados y las relaciones entre los procesos software modelizados y los entornos organizacionales deben ser analizados de modo preciso [Abdel, 91] [Seaman, 94] [Min, 97]. Particularmente, se deben analizar de modo preciso las relaciones entre los procesos software modelizados y los entornos organizacionales.

En esta perspectiva de modelización integral de los procesos software y las propiedades de los entornos de las organizaciones de la IS se enfoca este trabajo.

I.2. PLANTEAMIENTO DEL PROBLEMA

El modelado es una técnica utilizada en todas las ingenierías. Como tal, la IS debe basarse en el modelado como una parte central de todas las actividades que conducen a la producción de software de calidad. Estas actividades varían según la organización y el tipo de sistema, pero en general la gestión del proceso software exige disponer de un modelo. Se construyen modelos para comunicar la arquitectura y el comportamiento deseado en el sistema. Se construyen modelos para visualizar y controlar la arquitectura del sistema y para obtener una mejor comprensión del sistema que se está desarrollando, muchas veces descubriendo posibilidades de simplificación y reutilización. Se construyen modelos para controlar el riesgo [Booch, 99].

Según Sommerville [Sommerville, 95], diferentes clases de modelos de proceso contribuyen de formas diferentes al entendimiento del problema en cuestión. Diferentes modelos de proceso pueden representar diferentes puntos de vista. Por ejemplo, un modelo puede definir los agentes (humano y computarizado) involucrados en cada actividad, mientras que otro puede centrarse en las relaciones entre las actividades y qué productos se intercambian. Cada modelo observa, se focaliza o da prioridad a determinados elementos o aspectos de un mundo tan complejo como la construcción de software [Dowson, 90][Feiler, 93].

Tradicionalmente, los modelos de proceso software [Lonchamp, 90][IEEE, 91][Penedo, 91][Kellner, 91][Bandinelli, 93][Finkelstein, 94][Canals, 94][Engels, 94][Junkermann, 94][Min, 97][Franch, 99] han centrado sus representaciones en tres características elementales del proceso: la actividad, el artefacto y el agente [McChesney, 95], pero empíricamente está demostrada [Boehm, 81][Scacchi, 84][Adelson, 85][Curtis, 85][Hastie, 87][Thamhain, 87][Guindon, 88][Curtis, 88][Carasik, 88][Sherdil, 96][Kawalek, 96] la gran influencia en el proceso de producción de otras características: los roles de los humanos y la organización. En este aspecto, los recursos humanos constituyen el factor menos formalizado en los modelos de proceso software actuales.

Sin embargo, su importancia es obvia: presentan un comportamiento no determinístico y subjetivo que influye de forma decisiva en los resultados de la producción de software que es una actividad básicamente intelectual y social [Sommerville, 95], ver figura 1.1. Además, la falta de especificación de las

personas provoca que el proceso no refleje la situación real de la organización modelizada. Esto también añade el riesgo que se ejecuten procesos no adecuados a la capacidad de los factores humanos de la organización.

Según Jacobson [Jacobson, 00], las personas serán decisivas en el desarrollo de software en el futuro que esperamos. En último término, el contar con recursos humanos adecuados y eficaces es lo que lleva al éxito seguro, permitiéndoles que se dediquen a lo que seres humanos preparados pueden hacer, ejecutar actividades claves conforme a sus aptitudes y capacidades idóneas, de acuerdo a su desempeño, responsabilidad y compromiso para un rol concreto. La gente llega a ocupar diferentes roles en una organización, su preparación para el cargo correcto depende y requiere una formación y entrenamiento preciso, un análisis, seguimiento y supervisión guiadas para una cuidadosa asignación en el puesto de trabajo apropiado. Las personas son decisivas y contribuyen a una conquista sólida dentro del contexto del software.

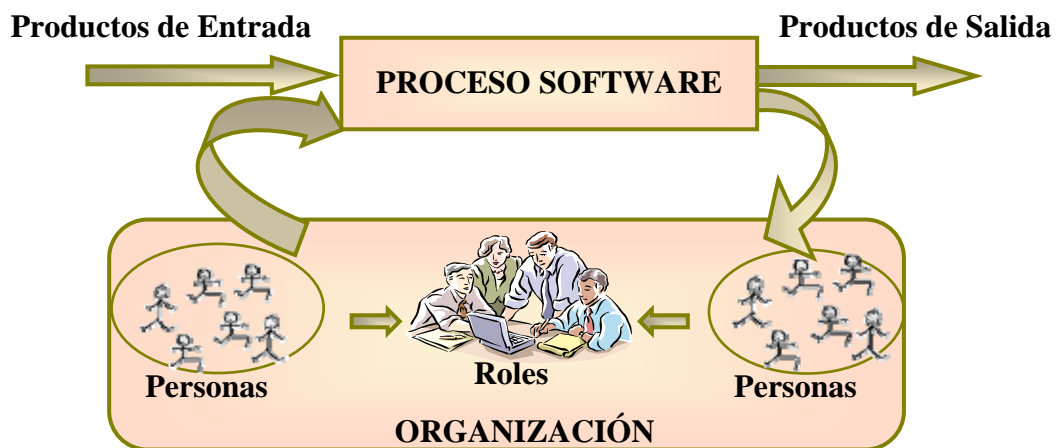


Figura 1.1. Las personas y los roles en los modelos de proceso software existentes

Acuña y Barchini diseñaron el MPM [Acuña, 99a][Acuña, 00a], que considera cuestiones *culturales, sociales, organizacionales* (procesos, roles, cultura) y de las *personas* involucradas en el proceso de desarrollo de software. Este modelo desarrollado se basa en la simbiosis de varios ambientes: el ambiente socio-cultural, el ambiente científico-tecnológico y el ambiente paradigmológico. Es una visión concisa e integrada de niveles de ecología, ingeniería, gestión, desarrollo y evaluación del proceso en relación con sus ambientes. Esta visión incluye la participación y rol de los seres humanos, el tipo de herramientas

intervinientes y permite comprender los productos desarrollados, por cada nivel [Acuña, 97]. Este modelo, aunque integral, es informal y está representado sólo mediante una especificación textual no formalizada.

A pesar de todos los esfuerzos y avances realizados en los últimos años, se detecta todavía una falta de conceptualización y formalización de la incorporación de los humanos y la interacción en la que participan; y la carencia de un método sistemático, disciplinado y formal que permita la modelización conceptual tanto de las partes técnicas como humanas del proceso software [Engels, 94][Seaman, 94][Kawalek, 96][Nguyen, 96][Min, 97]. Este trabajo aporta una modelización conceptual de aspectos organizacionales y de las personas involucradas en el proceso software a través de un método de análisis formal.

Concretamente los problemas detectados en este trabajo son:

- La modelización conceptual de los aspectos organizacionales y de las personas involucradas en el proceso software se encuentran escasamente formalizados.
- La mayoría de los modelos existentes no contemplan ciertos aspectos del componente humano tales como sus roles, capacidades e interacciones con los demás elementos modelizados [Boehm, 81][Scacchi, 84][Adelson, 85][Curtis, 85][Hastie, 87][Thamhain, 87][Guindon, 88][Curtis, 88][Carasik, 88][Sherdil, 96][Kawalek, 96].
- No se cuenta con un método que ayude a los analistas a construir modelos conceptuales y formales de calidad y que cubra los puntos débiles y carencias importantes que poseen los métodos de análisis existentes.
- Los métodos de análisis existentes, que examinan las especificaciones de partida para el análisis y definición de problemas, parten de modelos descriptivo-lingüístico, que se caracterizan por la ambigüedad, reflejo de la calidad de la información con la que se cuenta y por la intuición y / o experiencia atesorada del analista. Este problema puede tener incidencia en la especificación de requisitos al incluir información no significativa [Moreno, 97].

- Basado en el ítem anterior, es inevitable que el analista incluya el factor subjetivo a la hora de decidir cuales serán los elementos a modelizar y directamente grafique determinada información sin un estudio previo.
- Por consiguiente, los modelos resultantes no llegan a ser uniformes y adecuados y puede obtenerse diversas soluciones para un mismo problema de partida, en consecuencia, dichos modelos no reflejan correctamente la situación real analizada.
- Al no contar con un método que esté erigido sobre guías y reglas para obtener los fundamentos claves de los modelos conceptuales del proceso software, así como también, que determine etapas a abordar para llevar a cabo el proceso de Análisis y poder construir los citados modelos, es probable que el problema a resolver y su solución no se representen adecuadamente, aspiración y desafío de todos los participantes que emprenden un desarrollo de software.

I.3. HIPÓTESIS

Este estudio, tiene por objetivo fundamental validar el método de Análisis orientado a objetos MATE.

La hipótesis que se plantea en una investigación es una proposición que permite establecer relaciones entre los hechos y que es puesta a prueba para determinar su validez.

Las hipótesis realizadas para abordar esta finalidad son las siguientes:

Hipótesis 1: *El método MATE es aplicable en la modelización conceptual de los modelos de proceso software a partir de su descripción en lenguaje natural.*

La figura 1.2 representa a la hipótesis 2.

Como ya se refirió, aunque existen métodos que indican un conjunto de pasos definidos para realizar el proceso de Análisis, ninguno de ellos proporciona

además guías formales, justificadas, completas y correctas para construir los modelos conceptuales resultantes de este proceso.



Figura 1.2. Representación de la Hipótesis 1

Por lo tanto, los métodos actuales no proporcionan “la base” para resolver las dificultades no representadas adecuadamente por los analistas del proceso de Análisis. Esta base sería la existencia de un proceso totalmente formalizado para llevar a cabo esta actividad.

Esta insuficiencia en la MOO, y la necesidad de cubrirla ha sido ya resaltada por diversos autores como son: Honiden [Honiden, 93], Argila [Argila, 94], Chen [Chen, 94], Frederiks [Frederiks, 95], Iivari [Iivari, 95], Basili [Basili, 88], Wang [Wang, 97], Northrop [Northrop, 97] y Moreno [Moreno, 97]. Todos ellos hacen hincapié en la inexistencia de criterios rigurosos para identificar los componentes de los modelos de la OO. Indicando que esta carencia hace que al proceso de Análisis continúe en cierta etapa de infancia lo que impide llevar a cabo esta labor de forma efectiva.

Un método de Análisis formal, ha de proporcionar, tal como indica Sutcliffe [Sutcliffe, 97] criterios exhaustivos para identificar los componentes de la modelización conceptual, así como procedimientos concretos que guíen al desarrollador del software durante el proceso de construcción de estos modelos, de manera que representen adecuadamente el proceso y su solución.

MATE constituiría *EL* método de Análisis formal que permita construir modelos conceptuales de manera rigurosa y precisa, evitando así la aleatoriedad existente en los métodos empleados hasta el momento y proporcionando el soporte para solventar las dificultades accidentales e imprevistas que puedan surgir durante el Análisis.

Para alcanzar este fin, el enfoque sugerido, MATE (que en los capítulos siguientes será presentado en detalle) se basa en el estudio de la información que con más probabilidad está disponible al comienzo del desarrollo de software

[Moreira, 94] [Derksen, 96], constituida por un conjunto de sentencias en lenguaje natural que describen las características del problema que se pretende resolver y constituyen la entrada del método propuesto. Esta descripción está formada por palabras con un significado determinado. Estas palabras, a su vez, denotan elementos de una modelización. Estas palabras tienen cierto significado y normalmente su uso en la modelización está relacionado con dicho significado.

En definitiva, para verificar la primera hipótesis presentada, se aplica MATE al MPM a partir de su descripción textual [Acuña, 00b].

Hipótesis 2: *En la modelización conceptual, el método MATE es más eficaz y eficiente que el método de análisis del Proceso Unificado.*

La mejor forma de entender un problema es mediante la construcción de modelos que describen parte del mundo real de una forma no ambigua, no redundante y en un lenguaje común a todos los informáticos, estos modelos reciben el nombre de modelos conceptuales.

El modelo conceptual es una representación equivalente lógica aproximada del proceso considerado, que como tal constituye una abstracción simplificada del mismo y que a continuación se traduce en un modelo apto para su ejecución en una computadora. Asimismo que, la MOO, es la modelización más versátil ya que permite fácilmente el diseño e implementación en cualquier tipo de paradigma. La obtención de modelos conceptuales de la MOO, a partir de los requerimientos del usuario, es uno de los principales propósitos de la modelización del proceso.

Se modelizan las actividades, los documentos de entrada y de salida, las partes humanas, las interacciones entre ellas, sus capacidades y roles junto a los aspectos ya considerados por los modelos de proceso software existentes.

Para que la validación empírica a realizarse sea aceptada por la comunidad científica, se realiza mediante un enfoque orientado a objetos a partir de dos diferentes perspectivas: la de MATE y la del PU, con esta propuesta se pretende proporcionar un modelo totalmente formalizado. Se concentra la investigación en los puntos de confluencia y contradicción de ambos métodos de análisis.

Por ende, es fundamental que el producto final represente adecuadamente y con total éxito la realidad modelada, quedando asegurado el recubrimiento de todas sus entidades: procesos, productos, personas, roles, etc.

En conclusión, el resultado codiciado es asegurar la validez de todos los elementos influyentes, tanto organizacionales como técnicos, su modelización mediante un método adecuado y la obtención de modelos coherentes al problema particular que se está analizando.

I.4. OBJETIVOS

Sobre la base de lo expuesto precedentemente, los objetivos generales y específicos de la tesis son:

- **Objetivos Generales**
 - **Obtener** modelos conceptuales y formales que reflejen aspectos organizacionales (procesos, roles, cultura) y de las personas involucradas en el proceso de desarrollo de software.
 - **Lograr** una modelización estático-dinámico conjunta, orientada a objetos, de las actividades, los procesos, los productos, las personas, los roles, la organización y sus interacciones, presentes en el proceso software.
 - **Facilitar** la obtención de modelos formales a partir de modelos lingüísticos y / o descriptivos de un modelo de proceso software.

- **Objetivos Específicos**
 - **Comprobar** experimentalmente la efectividad y eficiencia de MATE respecto al modelo de análisis del PU.
 - **Eliminar** la ambigüedad existente en los modelos lingüísticos / descriptivos.
 - **Diferenciar e Integrar** los aspectos estáticos y dinámicos de la información.
 - **Reducir** el componente intuitivo y subjetivo en el proceso de modelización.
 - **Facilitar** la representación de los componentes claves de la OO, que formarán parte de los modelos resultantes.

▪ Descripción

La cuestión que se pretende abordar en este trabajo, para cumplir con los objetivos definidos, es realizar la validación empírica del método MATE [Moreno, 97]. El método propuesto será utilizado para modelizar los procesos Organización e Ingeniería y sus procedimientos, que permiten la realización del Nivel de Ingeniería del modelo cíclico a ser presentado en el capítulo 2. La modelización de los mencionados procesos del Nivel de Ingeniería del MPM espera guiar a los ecosistemistas, ingenieros y desarrolladores del proceso en el diseño y construcción de un proceso software útil a la IS.

Para comprobar la hipótesis 1 concebida en este capítulo, se aplica las nueve etapas del método MATE, a ser explicadas en el capítulo siguiente, a las especificaciones textuales del Nivel de Ingeniería del modelo Multinivel. Permitiendo, de esta forma, obtener su modelización conceptual correcta, rigurosa y precisa. Las especificaciones textuales son la información que, con más probabilidad, se dispone al comienzo del desarrollo, [Moreira, 94] [Derksen, 96]. La idea de la investigación, es analizar esta información del MPM desde el punto de vista sintáctico y semántico con el fin de extraer de ella, mediante un procedimiento formal, los componentes de un sistema orientado a objetos representados en un modelo que describe la estructura estática del sistema, el MO y un modelo que describe la parte dinámica del mismo, el MC. Sobre estos modelos se podrá aplicar el diseño y finalmente la codificación. La modelización finalizada servirá para validar la eficacia del método, realizando el análisis formal del modelo de proceso software considerado.

Por otro lado, para verificar la hipótesis 2, se llevará a cabo una experimentación que consistirá en dos muestras; en una muestra se aplicará el método MATE y en la otra muestra se aplicará el modelo de análisis del PU. Los datos que se obtengan en cada muestra estarán sujetos a variables respuesta, que marcan las pautas de análisis e interpretación de los resultados. Esta experimentación servirá para evaluar la efectividad y eficiencia de ambos métodos de análisis, a través de una comparación entre estas diferentes perspectivas de la OO, valoradas para ésta investigación aplicada.

CAPÍTULO II. MARCOS REFERENCIALES

*“Todo método consiste en el orden y disposición
de aquellas cosas hacia las cuales es preciso
dirigir la agudeza de la mente”*
Descartes

II.1. MARCO TEÓRICO

II.1.1 MODELO DE PROCESO SOFTWARE MULTINIVEL

En esta sección se describirá de manera textual el modelo de proceso software cíclico, valorado por sus características exclusivas como es la inclusión del integrante humano. En todo este apartado se transcribe la descripción en lenguaje natural del MPM, entrada al método MATE.

En 1997, se ha diseñado un **Modelo de Proceso Multinivel** (MPM) [Acuña, 97], representado en la figura 2.1, que incorpora aspectos organizacionales en el proceso de desarrollo de software. Este modelo, como ya se ha mencionado, está formado por tres ambientes: el ambiente socio-cultural, el ambiente científico-tecnológico y el ambiente paradigmológico. El *ambiente socio-cultural* es donde principalmente se originan, aplican y evalúan los productos y procesos software. El ambiente *científico-tecnológico* es donde se desarrollan las dimensiones teleológicas, políticas y decisionales. El *ambiente paradigmológico* es donde se produce y desarrolla el ciclo de vida del proceso. Cada ambiente tiene distintos niveles. Estos niveles en relación con su ambiente enfatizan el comportamiento humano y organizacional, el teleológico y decisional y el de abstracción-modelización, respectivamente. Estos niveles representan el ciclo del proceso y son los siguientes: *Nivel de Universo Ecológico*, *Nivel de Ingeniería*, *Nivel de Gestión*, *Nivel de Desarrollo* y *Nivel de Evaluación*; y por las líneas del “*qué*”, “*quién*” y “*cómo*”, que caracterizan los procesos objetos de desarrollo, el perfil de los seres humanos involucrados y el tipo de las herramientas utilizadas. En el *de Nivel de Universo Ecológico*: los eco-sistemistas identifican, comprenden y describen las características socio-culturales y tecnológicas del contexto en el que se inserta la organización, a través de metodologías sistémicas. En el *Nivel de Ingeniería*: los ingenieros del proceso diseñan, construyen y mejoran modelos de proceso software generales mediante el uso de herramientas orientadas al diseño. En el Nivel siguiente denominado *de Gestión*: los gestores del proceso adoptan y personalizan las descripciones desarrolladas en el Nivel de Ingeniería para usarlas en proyectos software específicos, mediante el uso de herramientas orientadas a la gestión. En el *Nivel de Desarrollo*: los desarrolladores del proceso construyen y mejoran el software de aplicación que satisface las necesidades (requisitos) del usuario. En este nivel las herramientas específicas utilizadas por los desarrolladores del proceso

operan sobre las partes del software de aplicación mientras que las herramientas utilizadas en los niveles de Ingeniería y de Gestión operan tanto sobre las partes del proceso como de la organización. Por último, en el *Nivel de Evaluación*: los evaluadores valoran cualitativa y cuantitativamente los procesos involucrados y los productos obtenidos.

Este modelo multinivel permite la ingeniería, gestión y realización del proceso software en relación con su ambiente integrado y con una comprensión profunda de que el software resulta de un proceso de comunicación complejo.

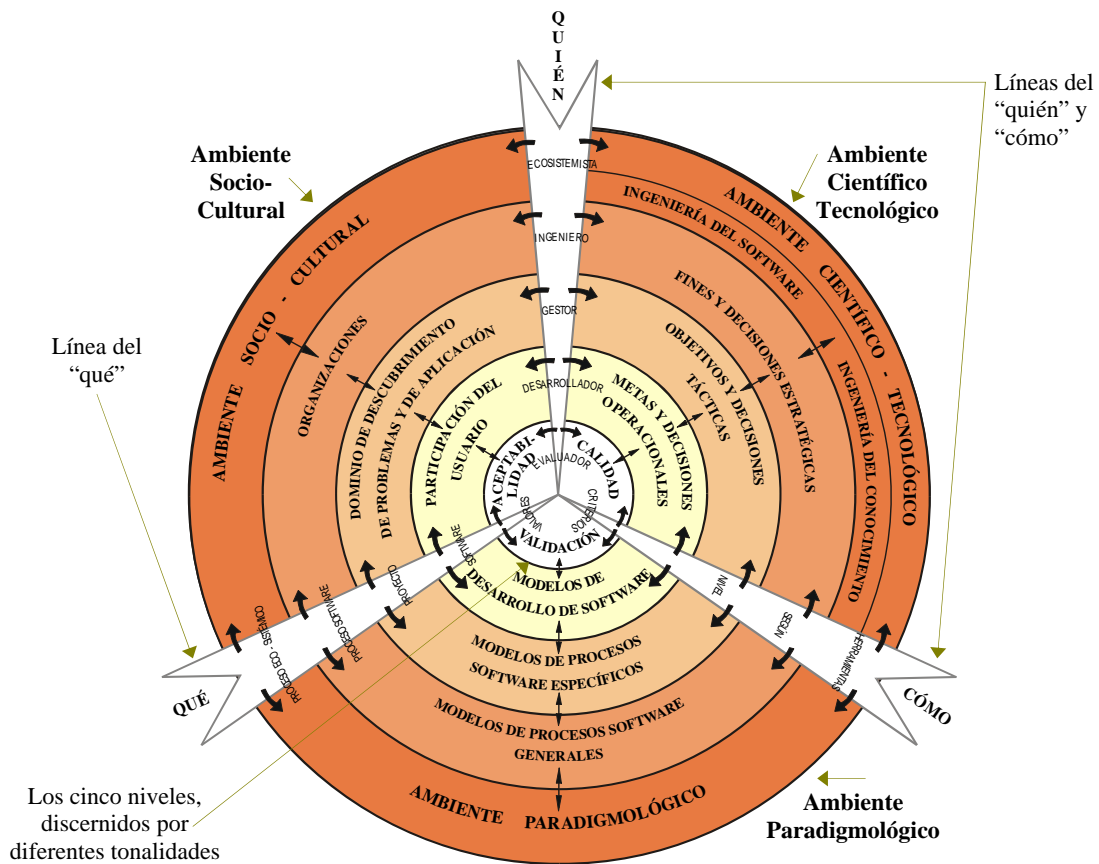


Figura 2.1 Modelo de Proceso Multinivel (MPM)

Acuña y Barchini [Acuña, 99a][Acuña, 00a] describen en detalle los procesos involucrados para el Nivel de Ingeniería de este modelo multinivel. El Nivel de Ingeniería consta de dos procesos: el *Proceso Organización* y el *Proceso Ingeniería*, que consisten, respectivamente, en la modelización de la organización y de los procesos software generales dependientes de la organización.

Estos Procesos desarrollados constan de dos procedimientos: *Procedimiento Cultural* y *Procedimiento de Modelo de Proceso*, que permiten realizar los requisitos asociados con cada proceso. El primer procedimiento indica cómo modelizar la organización, develar su cultura y determinar las capacidades de sus miembros y el segundo procedimiento cómo representar modelos de proceso software generales que incorporen estos aspectos organizacionales y de las personas involucradas en el desarrollo de software.

En el capítulo de desarrollo se modeliza, como experiencia individual, los mencionados procesos y sus procedimientos, que permiten la realización del *Nivel de Ingeniería del MPM*, utilizando el método *MATE* [Acuña, 00b].

II.1.2. NIVEL DE INGENIERÍA DEL MODELO DE PROCESO MULTINIVEL

El **Nivel de Ingeniería**, representado en la figura 2.2, es fundamental para el resto de los niveles del Modelo de Proceso Multinivel, por cuanto su objetivo es diseñar o mejorar los Modelos de Proceso Software Generales (MPSG) adecuados a la organización considerada y Modelos del Ciclo de Vida del Software (MCVS). Los MPSG constituyen el punto de partida para el desarrollo de los proyectos de software. Consta de dos procesos: el Proceso Organización y el Proceso Ingeniería.

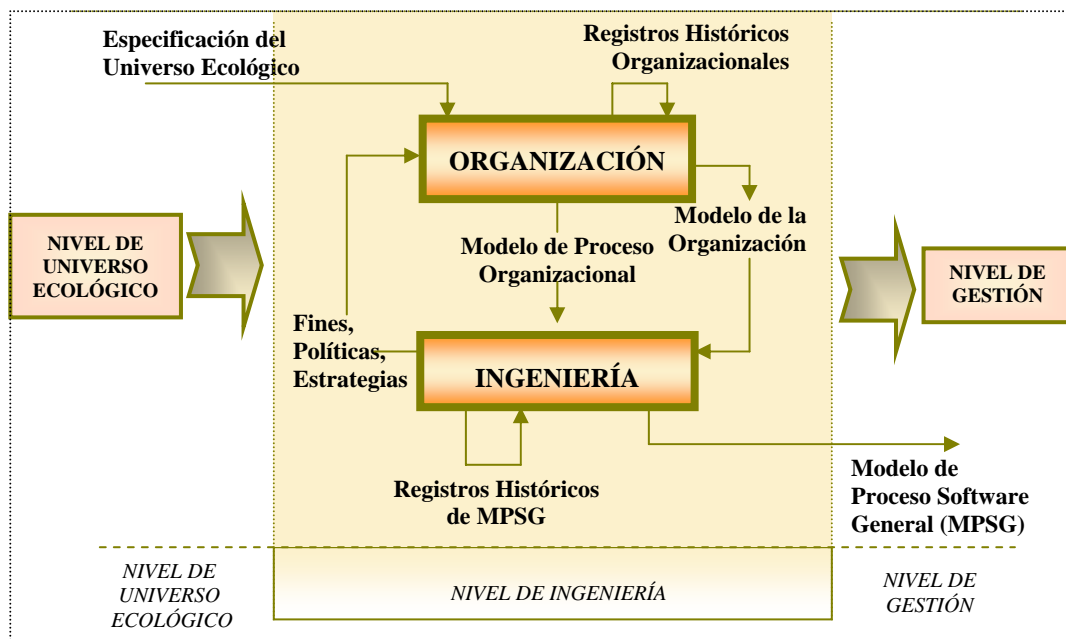


Figura 2.2. Nivel de Ingeniería del MPM

El **Proceso Organización** consiste en el descubrimiento y comprensión de la cultura del entorno de la organización y de la cultura de la propia organización. Se inicia el proceso con una descripción de las características socioculturales y científico-tecnológicas de la organización y de su entorno, en base a los documentos: Especificación del Universo Ecológico y Registros Históricos de la Organización. El análisis de la organización y su entorno permite: develar la cultura y el grado de desarrollo de la organización, formular cambios organizacionales y soluciones potenciales y definir los requisitos de la organización. Se genera, entonces, un **Modelo de la Organización Actual**, que involucra fines y decisiones estratégicas y políticas; un **Modelo de la Organización Futura**, que incluye una estrategia de mejora de la organización; y un **Modelo de Proceso Organizacional**, que abarca los cambios actitudinales, estructurales y procedimentales que deberían realizarse en la organización. Los modelos de la Organización Actual y Futura constituyen el Modelo de la Organización.

El **Proceso Ingeniería** consiste en la modelización conceptual y formal del proceso software, que implica la construcción de una serie de modelos intermedios a diferentes niveles de abstracción. Sobre la base principalmente del Modelo de la Organización Actual, el Modelo de la Organización Futura y el Modelo de Proceso Organizacional, se realiza la construcción o mejora iterativa del modelo requerido. Este proceso permite identificar la idea o necesidad de MPSG, definir esta idea o necesidad, definir los requisitos del modelo a construir, conceptualizar, diseñar y evaluar el modelo y archivar los MPSG. A partir de la realización de estas actividades se obtienen los **MPSG**, reiteradamente mencionados.

Para llevar a cabo el Proceso Organización del Nivel de Ingeniería del Modelo Multinivel se ha diseñado un **Procedimiento Cultural**. Este procedimiento se centra en cómo modelizar la organización, cómo develar su cultura y cómo determinar las capacidades de las personas involucradas en el proceso de desarrollo de software. Las técnicas de Educación proporcionan la *Especificación del Universo Ecológico*, que constituye la entrada del procedimiento y conforman una descripción textual de la organización y su entorno, es decir un conjunto de secciones que expresan cómo es la estructura orgánica, funcional y de comportamiento de la organización, que representan las características de la organización y su entorno a estudiar. La salida de este procedimiento está formada

por una modelización conceptual compuesta por dos modelos: *el Modelo de la Organización* (Actual y Futura) y el *Modelo de Proceso Organizacional*.

El Procedimiento Cultural puede llevarse a cabo de una manera efectiva realizando las etapas siguientes [Acuña, 00a], cuya descripción y documentos de entrada y salida asociados se detallan en la tabla 2.1:

Fase I. Investigación Exploratoria.

Etapas I.1. Extracción de Información Esencial.

Etapas I.2. Estudio de Dominio Organizacional.

Fase II. Investigación Cualitativa.

Etapas II.1. Formulación de los Procesos Relevantes.

Etapas II.2. Determinación de las Capacidades de las Personas, para cada proceso relevante.

Fase III. Investigación Cuantitativa.

Etapas III.1. Construcción de los Modelos Conceptuales, de actividades y de roles para cada proceso relevante.

Etapas III.2. Determinación de la Cultura y la Estrategia de Mejora de la Organización.

Fase IV. Análisis y Evaluación.

Etapas IV.1. Definición de Procesos Factibles Culturalmente.

Etapas IV.2. Validación del Modelo de Proceso Organizacional.

Para llevar a cabo el Proceso Ingeniería del Nivel de Ingeniería del MPM se propone un **Procedimiento de Modelo de Proceso** que consta de las siguientes cinco fases: Exploración, Mejora, Análisis, Construcción y Validación-Verificación. La descripción, objetivos y técnicas de las fases del procedimiento propuesto se presentan en la tabla 2.2.

La modelización del Proceso Multinivel es una tarea difícil que requiere un método definido y formal. Por tal razón se elige el método MATE. La aplicación del método MATE da como resultado un MO y un MC, los cuales representan los modelos conceptuales del funcionamiento estático y dinámico del proceso que se desea modelizar y constituirá un punto de partida para diseñar el proceso software.

Tabla 2.1. Etapas del Procedimiento Cultural del MPM

FASE	DESCRIPCIÓN
<p align="center">ETAPA I.1: Extracción de Información Esencial</p>	<p>Esta etapa parte de la Especificación del Universo Ecológico obtenida durante el proceso de Educación del Universo Ecológico. El objetivo de la misma es descartar de la Especificación del Universo Ecológico la información que no constituya una característica funcional o una característica cultural; y hacer explícita la información implícita necesaria para entender las características funcionales y culturales incluidas en la Especificación del Universo Ecológico. La información de la Especificación de Universo Ecológico, junto con la información implícita constituyen las Características Relevantes, que son la salida de esta etapa.</p>
<p align="center">ETAPA I.2: Estudio de Dominio Organizacional</p>	<p>El objetivo de esta etapa es obtener una especificación global de la estructura, los procesos y el comportamiento de la organización en relación con su entorno y de la propia organización. Se estructuran los elementos que integran situaciones problemáticas en el dominio de la organización. Esta estructuración se realiza a través de dos actividades: una de análisis y otra de síntesis. En la actividad de análisis se considera la información obtenida mediante las técnicas de educación. En la actividad de síntesis se utiliza la técnica del Cuadro Pictográfico (dibujo de libre expresión del dominio organizacional). La salida de esta etapa está constituida por el Diccionario de Conceptos del Dominio de la Organización y el Cuadro Pictográfico, que conforman el Modelo de Problemas.</p>
<p align="center">ETAPA II.1: Formulación de los Procesos Relevantes</p>	<p>El objetivo de esta etapa es definir un modelo conciso que represente tanto la parte estática como la parte dinámica de los procesos relevantes de la organización. A partir del Modelo de Problemas, se identifican posibles procesos “candidatos a problemas” o procesos relevantes y se intenta buscarles “solución”. Dicha solución que implica un cambio de la realidad social de la organización, se expresa a través de la <i>definición básica o definición radical</i>. La definición radical es una especificación textual de lo que es y de cómo deberá ser un proceso relevante. La información descriptiva de la parte estática de la definición radical contiene propiedades estructurales de la información que ha de manejar la organización. La información descriptiva de la parte dinámica de la definición radical especifica interacciones que transforman a la información descrita en la parte anterior. La salida de esta etapa es el Modelo de Procesos.</p>
<p align="center">ETAPA II.2: Determinación de las Capacidades de las Personas</p>	<p>El Modelo de Procesos se utiliza para la construcción del Modelo de Personas Preliminar. Para determinar qué capacidades o competencias conductuales están asociadas con las personas involucradas en cada</p>

Tabla 2.1. Etapas del Procedimiento Cultural del MPM (Continuación)

FASE	DESCRIPCIÓN
	proceso, se utilizan diferentes técnicas, tales como Análisis de las Visiones de las definiciones radicales, Cuestionarios, Test de personalidad, etc. El conjunto de listas que se obtiene tras esta etapa recibe el nombre de Modelo de Personas Preliminar. Este modelo será validado dando lugar al Modelo de Personas.
ETAPA III.1: Construcción de los Modelos Conceptuales	Del Modelo de Procesos se construye el Modelo de Actividades y Roles. Para definir el Modelo de Actividades y Roles es necesario determinar las actividades que deberá realizar el proceso ideal para cumplir los requerimientos de la definición radical y el conjunto de roles asociados. El conjunto de Modelos de Actividades y Roles que se obtiene tras esta etapa recibe también el nombre de Modelo Conceptual. La salida de esta etapa está formada por el Modelo de Actividades y Roles. El Modelo de Personas y el Modelo de Actividades y Roles constituyen el punto de partida para la construcción del Modelo Cultural de la organización.
ETAPA III.2: Determinación de la Cultura y la Estrategia de Mejora de la Organización	A partir de los Modelos de Personas y de Actividades y Roles se realiza la modelización conceptual de la cultura y la estrategia de mejora de la organización. Se devela la cultura de la organización, el grado de desarrollo alcanzado y las propuestas de mejora estratégicas de la organización bajo estudio. La salida es el Modelo Cultural Preliminar. Este modelo es válido si cumple con los requisitos especificados en el Modelo de Procesos, si cumple con las características culturales de factibilidad de la organización. La salida luego de la validación correspondiente es el Modelo Cultural.
ETAPA IV.1: Definición de Procesos Factibles Culturalmente	A partir del Modelo Cultural, se determina qué cambio es posible llevar a cabo en la organización. Se diseña un modelo conceptual participativo que involucra los procesos factibles culturalmente y se definen los cambios dando lugar al Modelo de Proceso Organizacional. Este Modelo incluye también los requisitos estratégico-tácticos de la organización.
ETAPA IV.2: Validación del Modelo de Proceso Organizacional	Se realiza una validación integral de los cambios y requisitos del Modelo de Proceso Organizacional. Para que los cambios puedan ser llevados a cabo, deben satisfacer dos requisitos: a) que sean culturalmente factibles; y b) que sean sistémicamente deseables. La salida estará dada por el Modelo de Proceso Organizacional Validado.

Tabla 2.2. Procedimiento de Modelo de Proceso

Fase	Descripción	Objetivos	Técnicas
I. Exploración	<p>Los requisitos del modelo se desarrollan sobre la base de los siguientes documentos: a) Modelo de la Organización Futura, donde se especifican los requisitos culturales de la organización; b) Modelo de Proceso Organizacional, que establece los requisitos de procesos de gestión, orientados al desarrollo y de soporte factibles culturalmente que se pueden modelizar; y c) Registros Históricos de los modelos de proceso software generales y específicos. Se obtiene el documento Requisitos del Modelo de Proceso, que establece las propiedades genéricas del modelo a construir.</p> <p>Se conforma un equipo de revisión que tiene a su cargo el estudio, análisis y comentarios de este documento. De esta manera, se refinan progresivamente los requisitos que se deben satisfacer. Esta fase reviste un carácter perfectible, iterativo y participativo presente en la especificación y evaluación de requisitos abiertos y dinámicos.</p>	<p>- Producir el documento Requisitos del Modelo de Proceso.</p>	<ul style="list-style-type: none"> - Técnicas de análisis orientadas a procesos, datos y objetos. - Técnicas de análisis de la Ingeniería del Conocimiento. - Técnicas formales de especificación. - Técnicas de prototipación. - Técnicas grupales (Metodología de los Sistemas Blandos, Descubrimiento Estratégico de Hipótesis y Prueba [Flood, 91]).
II. Mejora	<p>Sobre la base del Modelo de la Organización Actual y Futura y a los cambios culturales que necesita la organización (Modelo de Proceso Organizacional), se establecen los objetivos, criterios y factores a considerar de acuerdo a lo que se quiere mejorar en los MPSG de la organización.</p> <p>El ingeniero analiza la “historia clínica” (MPSG archivados) de la organización y considerando los “síntomas” de cada cultura “diagnostica” la situación de necesidad de mejora de la organización y conceptualiza el “tratamiento” (selección, creación de modelos) a llevar a cabo. Determina qué “remedios” (enfoques, procesos, técnicas, etc.) son necesarios e indica la “administración” de los mismos (administración de procesos, técnicas, etc.).</p>	<ul style="list-style-type: none"> - Diagnosticar el “estado de salud” cultural de la organización. - Mejorar la “salud” organizacional. 	<ul style="list-style-type: none"> - Técnicas de comunicación. - “Observación clínica”. - Técnicas de análisis de la Ingeniería del Software y la Ingeniería del Conocimiento. - Técnicas de diagnóstico. - Técnicas de madurez organizacional [Paulk, 95].

Tabla 2.2. Procedimiento de Modelo de Proceso (Continuación)

Fase	Descripción	Objetivos	Técnicas
III. Análisis	Se define el objetivo del modelo de proceso software (descriptivo, prescriptivo, guía, etc.). Se determina en forma preliminar las características del proceso en base a los elementos del Modelo de Proceso Organizacional (procesos, subprocesos, actividades, actores, herramientas, etc. y sus relaciones) y a los conocimientos (estratégicos, tácticos y fácticos) educidos de los expertos del proceso software. Se establece en forma preliminar quién es responsable de qué, qué objetos se requieren por quién y cuándo, qué herramientas son adecuadas para cada actividad, etc. Con un enfoque flexible e incremental se construyen modelos conceptuales estáticos y dinámicos que integren dichos elementos.	<ul style="list-style-type: none"> - Comprender las relaciones relevantes de los elementos del modelo de proceso. - Construir modelos conceptuales generales y comunicables. 	<ul style="list-style-type: none"> - Técnicas de modelización de la Ingeniería del Software y la Ingeniería del Conocimiento. - Diccionarios de conceptos. - Árboles de descomposición. - Modelos de concepto-relación. - Diagrama de flujo de conocimientos.
IV. Construcción	Se realiza una abstracción y refinamiento de los modelos conceptuales generados en la fase anterior y se especifican las interfaces adecuadas. Se diseñan los modelos formales asociados. Esta fase se realiza sucesivas veces recursivamente hasta llegar al nivel de desagregación requerido. La implementación del modelo se realiza en forma análoga a la de un proyecto software siguiendo los procesos del nivel de gestión, desarrollo y evaluación del MPM.	<ul style="list-style-type: none"> - Desarrollar modelos formales flexibles y adaptables 	<ul style="list-style-type: none"> - Técnicas de diseño de la Ingeniería del Software y la Ingeniería del Conocimiento.
V. Validación-Verificación	Se realiza la validación y verificación de los modelos formales de proceso software general. Estos se validan aplicando técnicas de revisión formal e incluyendo en el equipo de revisión a los expertos, a fin de satisfacer los requisitos especificados. Los modelos formales se verifican aplicando técnicas de verificación adecuadas. En caso de discrepancias se itera con la etapa anterior. Esta fase tiene un carácter distintivo ya que constituye una etapa de cada una de las fases anteriores. Se validan modelos de requisitos, de mejora, conceptuales y formales en forma iterativa. Los ajustes se realizan en los modelos correspondientes según los resultados obtenidos de la validación.	<ul style="list-style-type: none"> - Comprobar que los modelos cumplan con los objetivos definidos y con los requisitos especificados. 	<ul style="list-style-type: none"> - Técnicas de verificación. - Técnicas de validación. - Revisiones formales.

II.2. MARCO METODOLÓGICO

II.2.1. MODELOS CONCEPTUALES OBTENIDOS POR MATE

El método **MATE** es un método de Análisis que obtiene un modelo conceptual del cual partir para diseñar el sistema software. Como ya se ha dicho, dependiendo del enfoque usado durante un análisis, pueden obtenerse diversos modelos. En esta sección se muestran los criterios seguidos para determinar los modelos conceptuales que se obtendrán del análisis realizado con MATE. Una vez determinada la salida del método, se analiza la entrada, es decir, el lenguaje natural contenido de la *descripción textual* del problema a resolver, y se verá la necesidad de definir un conjunto de restricciones sobre dicha descripción, dando lugar a un sublenguaje concreto. A su vez, los elementos de los modelos conceptuales descritos y el sublenguaje natural constituyen el marco de trabajo de la Formalización de la Modelización Conceptual a partir del Lenguaje Natural (FORMOL), base del método MATE. Este marco contextual se muestra en la figura 2.3.

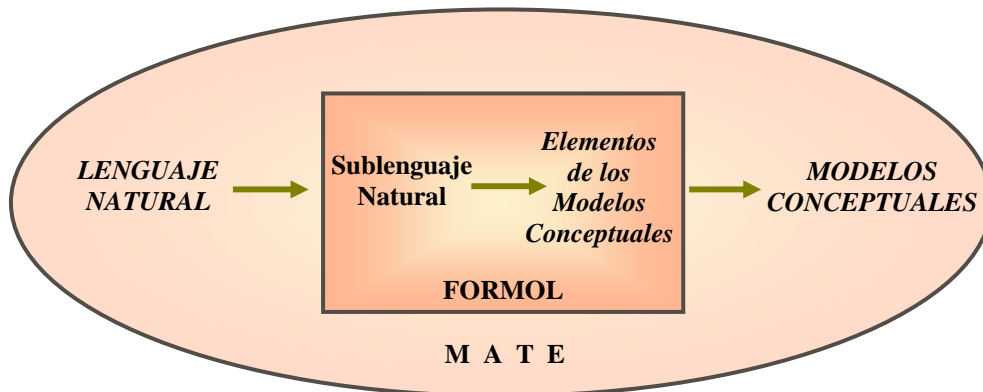


Figura 2.3. Marco Contextual del Método Mate

Se sabe que un modelo capta una parte del universo de discurso de un problema. El éxito de la modelización es, en gran parte, producto de su sencillez [Barbier, 92]. Un buen modelo debe basarse en un conjunto pequeño de conceptos de modelización [Rumbaugh, 96]. Es importante tener en cuenta que puede aumentar la complejidad del proceso de Análisis por el uso de múltiples modelos [Chen, 94]. Por otra parte, la creación de un único modelo integrado que pueda expresar fácilmente todos los aspectos importantes y distintos del documento de requisitos, no sería aconsejable, puesto que este modelo tendría una complejidad elevada, y sería difícil de entender y construir [Wieringa, 96]. Por todo esto, es

importante alcanzar un compromiso entre sencillez, y cobertura de todos los aspectos esenciales de un sistema.

La modelización conceptual de un sistema orientado a objetos debe describir los objetos y las interacciones entre ellos. La modelización conceptual del método MATE consta de un modelo que representa la estructura de la información con la que trabaja el sistema, y otro modelo que representa el comportamiento de dicho sistema, como ya se ha mencionado dichos modelos se denominan **Modelo Estático o de Objetos (MO)** y **Modelo Dinámico o de Comportamiento (MC)**, respectivamente. Este concepto se muestra en el gráfico de la figura 2.4.

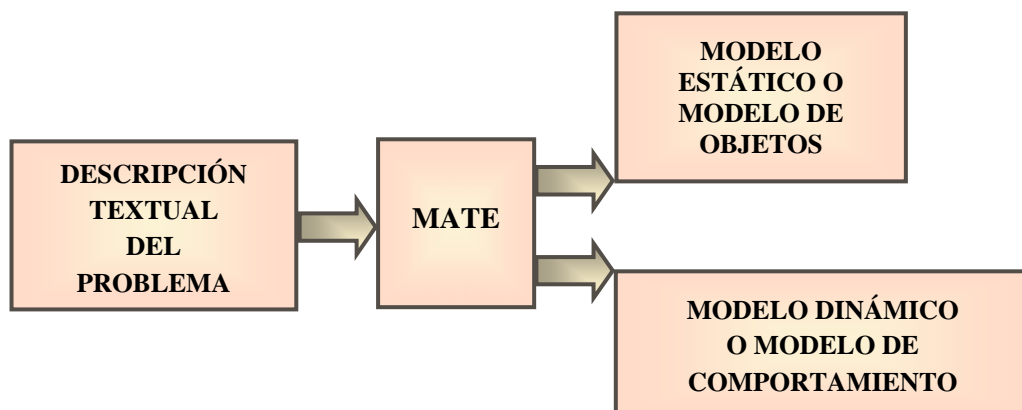


Figura 2.4. Modelos Conceptuales obtenidos por MATE

Los modelos conceptuales que se desarrollan con este método son modelos existentes, ya que es más útil utilizar modelos conocidos y probados, que crear otros nuevos que no estarían validados por el uso [Rumbaugh, 96]. Para representar la estructura de la información, se emplea, básicamente el MO de OMT [Rumbaugh, 91]. Los componentes del MO son las clases con atributos, las relaciones de herencia, de agregación y asociaciones, y las operaciones de los objetos. La notación del MO de dicha metodología se muestra en la figura 2.5.

El método emplea una descripción del comportamiento global del sistema, en términos de eventos y operaciones provocadas por dichos eventos. Más concretamente se usará la notación y los conceptos del MC de [Martin, 92], que se muestran en la figura 2.6.

Los conceptos utilizados por el MC de este método son los siguientes: un evento es un acontecimiento que provoca la realización de una operación. Una operación es una acción realizada por alguno de los objetos del sistema y que puede provocar un cambio en el estado de dicho objeto.

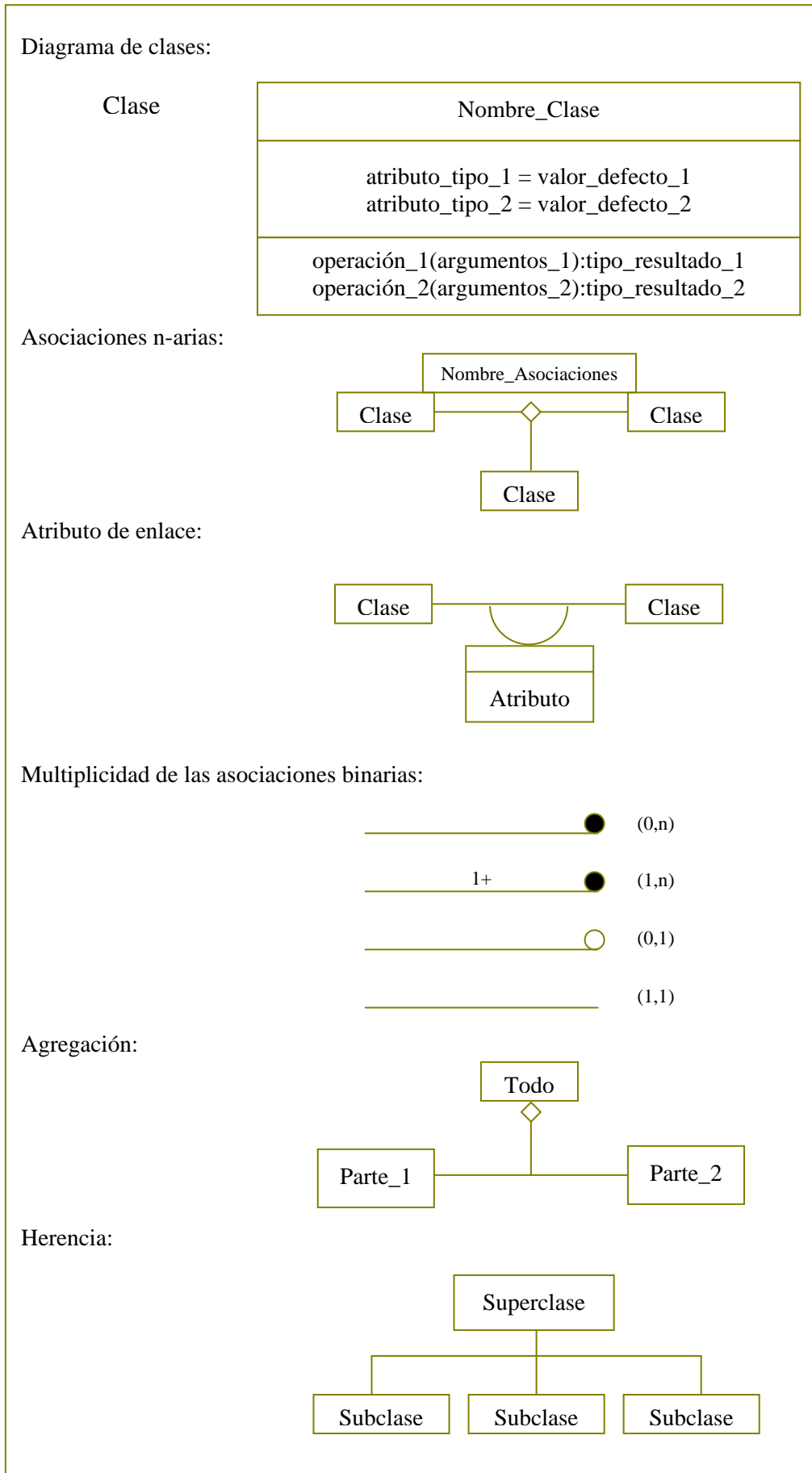


Figura 2.5. Notación del Modelo de Objetos de OMT

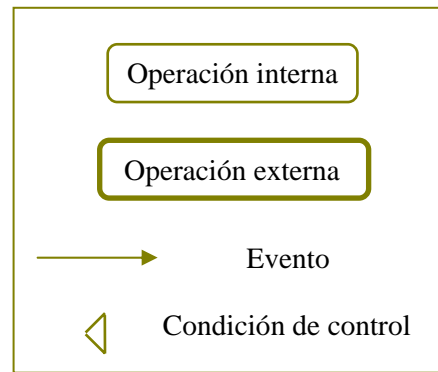


Figura 2.6. Notación del Modelo de Comportamiento

El estado de un objeto está formado por el conjunto de clases a las que el objeto pertenece, el conjunto de relaciones en las que participa, el conjunto de atributos que posee, y la información que pueda generar. Los eventos pueden ser de dos tipos: eventos externos y eventos internos. Un evento externo es el que procede del exterior del sistema que se está modelizando, es decir, es el resultado de una operación externa al sistema. Mientras que un evento interno está provocado por el propio sistema y es el efecto de la ejecución de determinada operación del sistema, operación interna, por el objeto correspondiente. Una operación puede estar provocada por un acontecimiento externo, por la ejecución de otra operación del sistema, o por la combinación de cualquier número de dichos eventos. La relación lógica formada por la conjunción y, o, disyunción entre los eventos que provocan la realización de determinada operación es lo que se conoce como condición de control.

Existen dos razones fundamentales que han llevado al uso de este modelo. En primer lugar, el hecho de que expresar el comportamiento global de un sistema “encaja” muy bien con la forma en que este aspecto se describe en las especificaciones de un problema [Burg, 95], y por otra parte, porque constituye un modelo completo para representar el comportamiento de un sistema [Calles, 93].

II.2.2. ELEMENTOS DE LOS MODELOS CONCEPTUALES UTILIZADOS POR FORMOL

FORMOL va a trabajar con determinadas combinaciones sintácticamente válidas, de los componentes de los modelos conceptuales. A estas combinaciones se las ha denominado **patrones conceptuales**. Estos patrones constituyen las piezas

clave que, combinadas unas con otras, dan lugar a los modelos conceptuales anteriores, el MO y el MC.

Los patrones conceptuales del MO reciben el nombre de *patrones conceptuales estáticos*, y los del MC reciben el nombre de *patrones conceptuales dinámicos*. A continuación, se describen cada uno de ellos.

II.2.2.1. PATRONES CONCEPTUALES ESTÁTICOS

El **MO** representa el aspecto estático del problema a analizar, en términos de clases y las relaciones entre ellas. Los **patrones conceptuales estáticos** representarán el esqueleto de la modelización estática [Meyer, 88], mediante los distintos tipos de relaciones entre los objetos de las clases, y las jerarquías de herencia entre dichas clases. Posteriormente, se completará esta modelización con información adicional como pueden ser las cardinalidades de determinadas relaciones del modelo.

Según el MO que se pretende construir, existen seis tipos de patrones conceptuales estáticos, y son:

c_1 Herencia simple: representa una relación en la que una clase es subclase de una superclase.

c_2 Herencia múltiple: representa una relación en la que una clase es subclase de varias superclases.

c_3 Asociación binaria: representa una relación semántica entre dos clases.

c_4 Agregación: representa un tipo de asociación, con semántica adicional, la relación entre un componente y sus partes.

c_5 Asociación de identificación: representa un tipo de asociación que indica una relación de unicidad entre las clases.

c_6 Asociación n-aria: representa una relación semántica entre un número n de clases mayor que dos.

La representación gráfica de estos patrones es la que se muestra en la figura 2.7. Los patrones c_1 y c_2 muestran respectivamente dos subclases, y dos superclases. Pero se podrían incluir cualquier número de ellas. De forma análoga el patrón c_6 representa una asociación ternaria, no obstante podría darse entre un número mayor de clases.

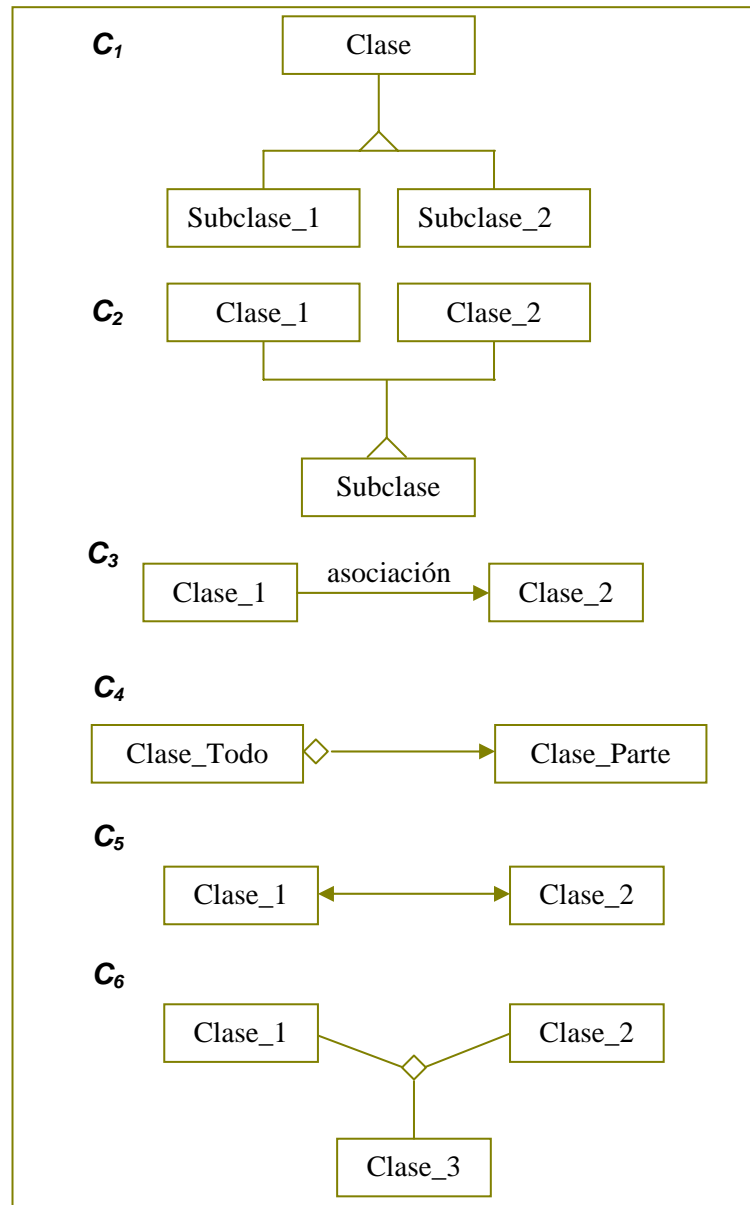


Figura 2.7. Patrones Conceptuales Estáticos

II.2.2.2. PATRONES CONCEPTUALES DINÁMICOS

El **MC** describe el funcionamiento del sistema en términos de las operaciones realizadas por las clases del mismo, y los eventos que provocan dichas operaciones. Esta información se modeliza mediante la combinación de patrones conceptuales como el que se muestra en la figura 2.8, junto con las condiciones de control asociadas a la operación. Estas condiciones de control son la relación lógica formada por la conjunción y, o, disyunción de los eventos $evento_1, \dots, evento_p$. Este **patrón conceptual dinámico** se identificará como c_7 Comportamiento.

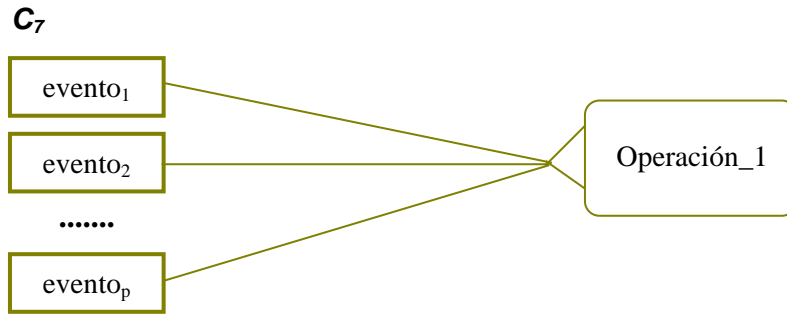


Figura 2.8. Patrón Conceptual Dinámico

II.2.3. LENGUAJE NATURAL COMO ENTRADA A MATE

MATE utiliza la información lingüística contenida en el conjunto de sentencias en lenguaje natural que constituyen la **Descripción Textual** del problema a resolver, con el fin de construir, de la forma más automatizada posible, los modelos conceptuales especificados. Así pues, MATE debe tomar como entrada el *lenguaje natural*. Sin embargo, el problema de usar el lenguaje natural en el proceso de desarrollo de software es su flexibilidad. Así, lo que los lingüistas, escritores y poetas consideran la belleza del lenguaje, es decir, las infinitas posibilidades de usarlo e interpretarlo, su riqueza expresiva, es para los desarrolladores de software y usuarios la principal limitación [Burg, 97]. Tanto es así, que no existe una gramática que describa de forma completa toda la sintaxis de este lenguaje [Chomsky, 89], por lo tanto, el lenguaje natural no es un lenguaje recursivo.

Por otra parte, los aspectos que puede contener la descripción de un problema son [Wieringa, 96]:

1. Requisitos de la empresa: delimitan el alcance de la investigación y proporcionan una justificación para construir el sistema. Consideran la estructura organizativa, los objetivos de la empresa, las actividades, y los productos de la organización. La información que pueden contener se podría englobar en una de estas categorías:
 - Razones para la realización del sistema.
 - Roles, responsabilidades, obligaciones y otras interacciones entre usuarios finales y clientes.

- Actividades productos y procesos de la empresa.
 - Información evaluadora respecto al funcionamiento actual de la empresa y al futuro funcionamiento con el nuevo sistema.
2. Requisitos funcionales: describen *qué* debe hacer el sistema, las capacidades del producto a desarrollar, el comportamiento externo del mismo. El comportamiento del sistema incluye la definición, descripción y organización de los datos con los que trabaja el sistema, y la definición y descripción de su funcionamiento en relación con el entorno.
 3. Requisitos no funcionales: establecen restricciones que el producto debe satisfacer, en términos por ejemplo, de requisitos de rendimiento (estáticos y dinámicos), de interfaces externos (de usuario, hardware, software, comunicaciones), restricciones de diseño, o atributos de calidad (mantenibilidad, seguridad, disponibilidad, etc.).

Esta información puede estar expresada mediante oraciones o frases del lenguaje natural de cualquier tipo, subordinadas, coordinadas, enunciativas, imperativas, etc. No se ha encontrado en la bibliografía información relativa al contenido sintáctico de unas especificaciones, es decir, no existen guías que determinen el tipo de oraciones que ha de incluir este documento, cómo deben estar estructuradas dichas oraciones, el tipo de verbos que se ha de utilizar, etc.

Por todo ello, se puede decir que no existen unas reglas sintácticas para describir la información que debe aparecer en la descripción textual, y que ésta puede contener cualquier expresión del lenguaje natural. Pero, como se ha indicado anteriormente, el lenguaje natural no está delimitado por una gramática bien definida. Por lo tanto, una descripción textual puede contener cualquier tipo de expresión lingüística.

Para poder utilizar formalmente la información lingüística de la descripción textual, es necesario saber con exactitud el formato de la información que puede contener este documento. Como se ha visto, esto no es posible, ya que el lenguaje natural no está delimitado. Por lo tanto, la solución es transformar este lenguaje natural, potencialmente ilimitado en un lenguaje natural restringido. Es decir, el método va a transformar la información contenida en la descripción textual en un subconjunto del lenguaje natural con una sintaxis y semántica definidas. Este

lenguaje restringido contiene un subconjunto manejable y definido de estructuras lingüísticas. La utilización de este tipo de lenguajes conlleva, en primer término, la reducción de la ambigüedad semántica y estructural, de forma que permite que el significado de un texto de especificaciones sea expresado con menor ambigüedad; y, en segundo lugar, proporciona una forma clara de comunicación entre el especialista en el dominio y el desarrollador de software. Estos subconjuntos del lenguaje natural reciben el nombre de lenguaje controlado [Fuchs, 96], lenguaje restringido [Cyre, 95], o lenguaje estructurado [Kristen, 94][Burg, 97]. Las ventajas de usar este enfoque son [Huijsen, 95]:

- Mejora de la legibilidad y entendibilidad, al reducir la ambigüedad léxica y estructural, y al definir las reglas sintácticas de antemano.
- Mejora de la mantenibilidad, lo que es consecuencia directa de la mejora de la legibilidad y entendibilidad.
- Mayor facilidad de proceso automático, debido a la reducción de ambigüedad y a las reglas sintácticas definidas.

El principal inconveniente del uso de estos lenguajes es la imposibilidad de probar que son equivalentes al lenguaje natural. Es decir, no es demostrable que cualquier estructura lingüística del lenguaje natural se pueda representar en el lenguaje controlado definido. Esta comprobación habría que realizarla demostrando la equivalencia entre las gramáticas que constituyen el lenguaje natural y el lenguaje controlado, lo cual no es posible, ya que como se ha indicado anteriormente, el lenguaje natural no es un lenguaje recursivo y, por tanto, carece de gramática que lo defina completamente. Así, este problema de demostrar la equivalencia entre el lenguaje natural y el lenguaje controlado es un problema NP-completo¹ [Frederiks, 95]. No obstante, empíricamente se puede comprobar [Pulman, 96][Dauphin, 96][Adriaens, 95] la equivalencia de estos lenguajes a nivel práctico. Además, las ventajas que proporcionan compensan esta limitación, de ahí su amplio uso en los ámbitos relacionados con el lenguaje natural.

¹ Estos forman parte de los problemas demostrablemente difíciles. Tiene la propiedad de que “si alguno de ellos pudiera resolverse mediante algún algoritmo eficiente, entonces todos los demás problemas de la clase NP, podrían resolverse eficientemente”.

II.2.4. SUBLEGUAJES UTILIZADOS POR FORMOL

Los modelos conceptuales generados por el método MATE, describen la información que manejará el sistema software a desarrollar, mediante el MO, así como el comportamiento de dicho sistema, mediante el MC. Las estructuras lingüísticas que proporcionan la información necesaria para la construcción de cada uno de éstos modelos difieren, ya que cada uno de ellos tiene una semántica muy diferenciada. Por ello, se van a definir dos sublenguajes distintos, uno para representar la información con la que va a trabajar el sistema, y otro para describir el comportamiento del mismo. Estos sublenguajes recibirán el nombre de **Lenguaje de Utilidad Estático** y **Lenguaje de Utilidad Dinámico**, respectivamente.

Estos sublenguajes están constituidos por un conjunto de estructuras lingüísticas denominadas patrones lingüísticos. Estos patrones se especifican en forma de gramática de contexto libre con la siguiente notación:

- < x > : símbolo no terminal
- x : símbolo terminal
- := : operador de definición
- | : alternativa de símbolos
- [x] : elemento opcional
- [x]₀ⁿ : repetitiva del símbolo x

Los símbolos no terminales que representan los verbos, indican cualquier conjugación de los mismos. En concreto el símbolo no terminal <verbo_general> representa los verbos que no están representados por los otros símbolos no terminales que hacen referencia a verbos.

II.2.4.1. LENGUAJE DE UTILIDAD ESTÁTICO

De la posible información que puede contener la descripción textual de un problema, en este momento interesa la que describe, organiza y clasifica los conceptos del universo del discurso relativo al problema a estudiar. En concreto, interesa aquella información sobre relaciones entre los distintos conceptos, y particularmente relaciones de clasificación de conceptos, y relaciones que expresan composición de conceptos en base a otros.

La estructura lingüística que puede representar esta misma información es la oración. Más concretamente, la oración simple. La razón es que una oración simple expresa una relación entre el sujeto y el predicado o, lo que es lo mismo, expresa una relación entre el concepto o conceptos que forman el sujeto, y los conceptos que constituyen los complementos que forman dicho predicado. Esta relación está determinada por el verbo de dicha oración. Según esta interpretación, la información que describe las relaciones entre los conceptos del dominio del problema se puede expresar en oraciones simples. En concreto, las oraciones simples que tienen la forma:

- l_1 Estructuras de clasificación: describen una relación entre el sujeto y los complementos que expresa determinada jerarquía de especialización entre ellos. Se han considerado los tres casos siguientes:
- $l_{1,1}$ Estructuras de clasificación sintéticas, de abajo arriba, o en inglés, “bottom-up”: expresan en el sujeto los subconjuntos en los que se puede descomponer el predicado.
 - $l_{1,2}$ Estructuras de clasificación analíticas, de arriba abajo o en inglés, “top-down”: indica en el predicado, los casos particulares en los que se puede descomponer el sujeto.
 - $l_{1,3}$ Estructuras de clasificación múltiple: los elementos representados por el sujeto poseen las características de los representados en el predicado.
- l_2 Estructuras con enumeración de complementos: representa entre pares de elementos que constituyen el sujeto y el predicado, respectivamente, una relación con carácter permanente. Se pueden identificar dos casos particulares, las estructuras l_3 y l_4 .
- l_3 Estructuras de composición: representan que el sujeto, o el predicado están constituidos por los conceptos que representan el predicado, o el sujeto. En ellas se pueden distinguir:
- $l_{3,1}$ Estructuras de composición del componente: donde el predicado representa los elementos que constituyen el sujeto.
 - $l_{3,2}$ Estructuras de composición del contenido: donde el sujeto representa los elementos que constituyen el predicado.
- l_4 Estructuras de identificación: expresan que el sujeto / predicado representa unívocamente al predicado / sujeto.

l_5 Estructuras con complementos adyacentes: amplían el patrón l_2 introduciendo un tercer elemento (u otros adicionales) definiendo una relación permanente entre todos ellos. Estas estructuras es conveniente expresarlas, cuando sea posible, como varias estructuras de enumeración de complementos, siempre y cuando no se pierda información.

La gramática que describe el **Lenguaje de Utilidad Estático** se muestra en la figura 2.9 y se puede profundizar en sus definiciones en el Anexo A.

II.2.4.2. LENGUAJE DE UTILIDAD DINÁMICO

El **Lenguaje de Utilidad Dinámico** va a representar el comportamiento de un sistema. Este comportamiento está determinado por unas operaciones que se realizan cuando ocurren determinados eventos. Se pueden encontrar múltiples estructuras lingüísticas que podrían expresar este tipo de información. Desde un signo de puntuación como puede ser una coma (“un cliente puede abrir una cuenta en el banco, emitiendo éste la libreta correspondiente”), o dos frases separadas por un punto (“El cliente emite una solicitud de alquiler de un coche. La empresa le asigna un vehículo adecuado para dicha solicitud”), hasta distintos tipos de conjunciones, por ejemplo *por consiguiente* (“un cliente entrega un vehículo, por consiguiente el sistema genera una factura”), o *cuando* (“cuando el proveedor recibe el pedido emite una orden de entrega”), e innumerables estructuras más. Pero no todas las estructuras que contienen, por ejemplo, una coma, ni todas las frases separadas por un punto expresan esta relación entre una acción y sus causas, esto depende del contenido semántico de estas estructuras. De ahí la dificultad de definir criterios objetivos para encontrar en un texto los antecedentes de una operación.

Una de las formas gramaticales más directa para expresar la relación entre los antecedentes de una operación y ésta, son las oraciones subordinadas condicionales ya que, en este tipo de oraciones se hace depender el cumplimiento de lo enunciado en la oración principal, de la realización de la oración subordinada. De esta forma, la información expresada mediante la oración principal contiene la operación a realizar, y la información asociada con la oración subordinada contiene el evento que provoca la ejecución de la operación. En base a lo dicho, se expresará la información que describe el comportamiento del sistema en oraciones subordinadas condicionales.

<Lenguaje de Utilidad Estático> := <I₁ Estructuras de clasificación> | <I₂ Estructuras con enumeración de complementos> | <I₃ Estructuras de composición> | <I₄ Estructuras de identificación> | <I₅ Estructuras con complementos adyacentes>

<I₁ Estructuras de clasificación> := <I_{1,1} Estructuras de clasificación sintéticas> | <I_{1,2} Estructuras de clasificación analíticas> | <I_{1,3} Estructuras de clasificación múltiple>

<I_{1,1} Estructuras de clasificación sintéticas> := Sintagma_Nominal [<Otro_Sintagma_Nominal><Nexo_Coordinante> Sintagma_Nominal] <Verbo_clasificación_sintética> <Complemento>

<Otro_Sintagma_Nominal> := [<Aposición> Sintagma_Nominal]₀ⁿ

<Aposición> := ,

<Nexo_Coordinante> := <Nexo_coordinante_copulativo> | <Nexo_coordinante_disyuntivo>

<Nexo_coordinante_copulativo> := y

<Nexo_coordinante_disyuntivo> := o

<Verbo_clasificación_sintética> := ser | ser un tipo | ser una clase

<Complemento> := Sintagma_Nominal

<I_{1,2} Estructuras de clasificación analíticas> := Sintagma_Nominal <Verbo_clasificación_analítica> <Complemento> [<Aposición> <Complemento>]₀ⁿ <Nexo_coordinante_disyuntivo> <Complemento>

<Verbo_clasificación_analítica> := ser | poder ser

<I_{1,3} Estructuras de clasificación múltiple> := Sintagma_Nominal <Verbo_clasificación_múltiple> <Complemento> [<Aposición> <Complemento>]₀ⁿ <Nexo_coordinante_copulativo> <Complemento>

<Verbo_clasificación_múltiple> := ser | poder ser

<I₂ Estructuras con enumeración de complementos> := Sintagma_Nominal [<Otro_Sintagma_Nominal> <Nexo_Coordinante> Sintagma_Nominal] <Verbo_general> <Complemento> [<Otro_Sintagma_Nominal> <Nexo_Coordinante> Complemento]

<I₃ Estructuras de composición> := <I_{3,1} Estructuras de composición del componente> | <I_{3,2} Estructuras de composición del contenido>

<I_{3,1} Estructuras de composición del componente> := Sintagma_Nominal [<Otro_Sintagma_Nominal> <Nexo_Coordinante> Sintagma_Nominal] <Verbo_composición_componente> <Complemento> [<Otro_Sintagma_Nominal> <Nexo_Coordinante> Complemento]

<Verbo_composición_componente> := constar | contener | tener | poseer | incluir | <Voz_pasiva> formar | <Voz_pasiva> constituir | <Voz_pasiva> dividir

<Voz_pasiva> := ser | estar

<I_{3,2} Estructuras de composición del contenido> := Sintagma_Nominal [<Otro_Sintagma_Nominal> <Nexo_Coordinante> Sintagma_Nominal] <Verbo_composición_contenido> <Complemento> [<Otro_Sintagma_Nominal> <Nexo_Coordinante> Complemento]

<Verbo_composición_contenido> := formar parte | pertenecer | ser parte | ser un componente | <Voz_pasiva> incluir

<I₄ Estructuras de identificación> := Sintagma_Nominal <identificar> <Complemento>

<I₅ Estructuras con complementos adyacentes> := Sintagma_Nominal [<Otro_Sintagma_Nominal> <Nexo_Coordinante> Sintagma_Nominal] <Verbo_general> <Complemento> [<Complemento>]₁^m

Figura 2.9. Gramática del Lenguaje de Utilidad Estático

La gramática que representa el Lenguaje de Utilidad Dinámico es la que se muestra en la figura 2.10, se puede profundizar en sus definiciones en el Anexo A.

```

<Lenguaje de Utilidad Dinámico> := <I6 Estructuras de condición>
<I6 Estructuras de condición> := Si y sólo si <oración_subordinada> entonces
<Oración_principal>
<Oración_subordinada> := <Oración_simple> [[<Nexo_1>
<Oración_simple>]on
<Nexo_2> <Oración_simple>]
<Oración_principal> := <Oración_simple>
<Nexo_1> := <Aposición> | <Nexo_2>
<Aposición> := ,
<Nexo_2> := <Aposición> <Nexo_Coordinante> | <Nexo_Coordinante>
<Nexo_Coordinante> := <Nexo_coordinante_copulativo> |
<Nexo_coordinante_disyuntivo>
<Nexo_coordinante_copulativo> := y
<Nexo_coordinante_disyuntivo> := o
<Oración_simple> := Sintagma_Nominal <verbo_general> [<Complemento>]im

```

Figura 2.10. Gramática del Lenguaje de Utilidad Dinámico

II.2.5. MÉTODO DE ANÁLISIS DERIVADO DE UNA ESPECIFICACIÓN TEXTUAL

En esta sección, se muestra el proceso práctico a seguir para construir los modelos conceptuales estático y dinámico del problema bajo estudio. El método que conduce este proceso constituye un método definido al especificar de forma rigurosa una serie de etapas y un conjunto de productos intermedios obtenidos como resultado de las acciones a realizar en dichas etapas. Durante la aplicación del método se emplea la correspondencia entre los patrones lingüísticos y los patrones conceptuales presentados en los apartados II.2.2. y II.2.4 y sus respectivos ítems.

El método de Análisis que se ha desarrollado por Moreno [Moreno, 97], utiliza como entrada la descripción textual del problema, es decir, un conjunto de oraciones en lenguaje natural que expresan cómo ha de ser el sistema software a desarrollar. La obtención de esta información, es decir, el subproceso de Educación de Requisitos, constituye la primera tarea a realizar en el desarrollo de cualquier sistema software. Las técnicas de obtención de requisitos, en sí, no son parte del proceso de Análisis, así pues no participan en el método aquí propuesto. Las técnicas de Educación deben proporcionar la descripción textual que representan las características del sistema a estudiar.

Según lo especifica Moreno [Moreno, 97], la salida del método está formada por una modelización conceptual formada por dos modelos: el **MO** y el **MC**, que representan una visión externa del funcionamiento del sistema que se desea desarrollar.

MATE puede llevarse a cabo de una manera efectiva realizando las etapas siguientes:

- 1) *Extracción de Información Esencial.*
- 2) *Identificación de Sinónimos y Polisemas.*
- 3) *Separación de la Información Estática y Dinámica.*
- 4) *Estructuración de la Información descriptiva de la Parte Estática.*
- 5) *Estructuración de la Información descriptiva de la Parte Dinámica.*
- 6) *Construcción del Modelo de Objetos Parcial.*
- 7) *Construcción del Modelo de Comportamiento.*
- 8) *Integración del Modelo de Objetos y del Modelo de Comportamiento.*
- 9) *Verificación del Modelo de Objetos y del Modelo de Comportamiento.*

La figura 2.11 representa éstas *etapas*, destacadas con sombreado.

Las flechas entre las etapas del método representan flujos de información entre las mismas. En la figura también se muestran en un óvalo y con trazo fino, los procesos Educación de Requisitos y Validación del Lenguaje de Utilidad. Estos procesos no están englobados en el método propuesto pero han de existir, y se producen iteraciones a, y desde ellos, desde las restantes etapas del método. El proceso de *Educción de Requisitos*, proporciona la descripción del problema a resolver, la descripción textual; y el proceso *Validación del Lenguaje de Utilidad*, permite que el usuario compruebe parte de la información generada durante la aplicación del método.

Los *productos*, en cuadros sin relleno, también se muestran en la figura 2.11.

Tras la Etapa 1: Extracción de la Información Esencial se obtienen los Requisitos Relevantes con la información significativa para el Análisis del problema. Este producto entra en la Etapa 2: Identificación de Sinónimos y Polisemas, y se obtienen los Requisitos Relevantes Depurados, donde se han reemplazado dichos sinónimos y polisemas, que figuran en un Diccionario de Sinónimos y Polisemas creado también, en esta etapa.

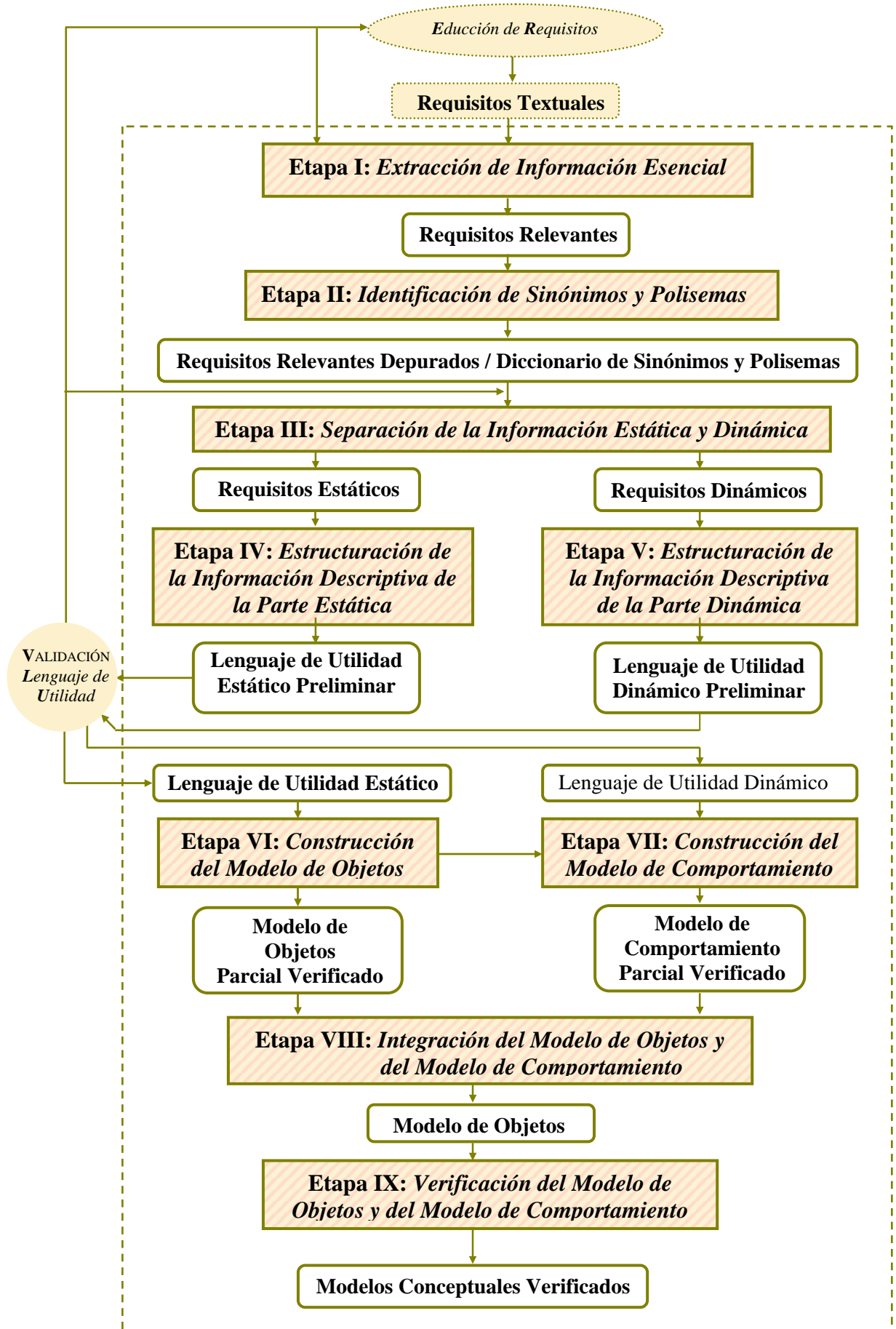


Figura 2.11. Etapas y Productos del método MATE

Durante la Etapa 3: *Separación de la Información Estática y Dinámica*, se dividen los Requisitos Relevantes Depurados en dos subconjuntos, Descripción de la Parte Estática y Descripción de la Parte Dinámica. En la Etapa 4: *Estructuración de la Información Descriptiva de la Parte Estática*, se estructura la Descripción de la Parte Estática en un Lenguaje de Utilidad Estático Preliminar. Este lenguaje es el que se usará posteriormente para la construcción del MO. En la Etapa 5: *Estructuración de la Información Descriptiva de la Parte Dinámica*, se estructura la Descripción de la Parte Dinámica, dando lugar al Lenguaje de Utilidad Dinámico Preliminar, que es lo que se usará para la construcción del MC en una etapa posterior. Los Lenguajes de Utilidad se validan, ya que a partir de ellos se obtendrán, los modelos conceptuales, el MO y el MC. Esta es la tarea que se realizaría en el proceso Validación del Lenguaje de Utilidad, dando lugar al Lenguaje de Utilidad Estático, y al Lenguaje de Utilidad Dinámico. En caso de que el usuario detecte inconsistencias en los Lenguajes de Utilidad, éstas pueden ser causadas por errores o “incompletudes” en la descripción textual o en cualquiera de las etapas previas del método. El primer caso implicaría la repetición del proceso de Educción de Requisitos, y una nueva iteración del método; mientras que el segundo conllevaría una nueva iteración desde la etapa que provocó la inconsistencia. Esto último se representa en la figura 2.11 con una flecha a la línea del contorno de las etapas del método. En las Etapas 6 y 7, *Construcción del MO* y *Construcción del MC*, respectivamente, se desarrollan el MO Parcial y MC. Ambos modelos se integran en la Etapa 8: *Integración del MO y del MC*, para añadir los métodos proporcionados por el MC en el MO. Finalmente, los dos modelos son verificados conjuntamente en la Etapa 9: *Verificación del MO y del MC*, dando lugar al MO Verificado y al MC Verificado.

Se describe, a continuación, cada una de las etapas del método presentado.

II.2.5.1. ETAPA 1: EXTRACCIÓN DE LA INFORMACIÓN ESENCIAL

Esta etapa parte de la descripción textual obtenida durante el proceso de Educción de Requisitos. El objetivo de la misma es extraer toda la información útil para el proceso de Análisis de un problema. Durante el Análisis se construyen los modelos conceptuales a partir de la información contenida en la descripción textual. De la información que puede contener este documento, no toda ella es útil para el

proceso de análisis. En concreto, los modelos conceptuales se construyen a partir de la información que constituye los requisitos funcionales [Wieringa, 96][Blum, 96]. Otro tipo de información necesaria para la construcción de los modelos conceptuales es la información implícita, muy importante para el entendimiento y tratamiento de la información explícita que constituye las especificaciones de un sistema [Gerstl, 92].

Por lo tanto, el objetivo de esta etapa es, por una parte, descartar de la descripción textual la información que no sea útil para el Análisis, es decir, toda la información que no constituya un requisito funcional; y por otra, hacer explícita la información implícita necesaria para entender los requisitos funcionales incluidos en la descripción textual.

Por tanto, se realizan dos tareas fundamentales dentro de esta primera etapa:

Tarea 1.1: Eliminación de información no necesaria para el Análisis

La información no necesaria para el Análisis es, como se ha indicado anteriormente, la que no forma parte de los requisitos funcionales. Por tanto, según [IEEE, 93], la información a eliminar es la que hace referencia a los siguientes aspectos:

- ✗ Relación entre el producto a analizar y otros productos.
- ✗ Características de los usuarios de la aplicación.
- ✗ Suposiciones o dependencias que pueden afectar a los requisitos del sistema, por ejemplo, una suposición sobre la existencia de un determinado sistema operativo, en caso de que no existiera habría que realizar cambios en el sistema.
- ✗ Requisitos de rendimiento tanto desde el punto de vista estático como dinámico.
 - Requisitos de rendimiento estático pueden ser los que hagan referencia al número de terminales o usuarios de la aplicación, al número de ficheros o registros a manejar, así como el tamaño de los mismos.
 - Requisitos de rendimiento dinámico son los que hacen referencia al tiempo de proceso de las transacciones y volumen de datos.
- ✗ Información relativa al diseño e implementación como:
 - Formatos de informes de entrada o salida.
 - Descripción de los algoritmos para la implementación de las operaciones.
 - Uso de determinado lenguaje de programación, compilador o base de datos.
 - Descripción de los algoritmos para la implementación de las operaciones.

- Uso de determinado lenguaje de programación, compilador o base de datos.
- Limitaciones hardware como ocupación de disco, memoria, arquitectura o plataforma.
- ✗ Limitaciones del sistema, es decir, qué no hará la aplicación.
- ✗ Información relacionada con la seguridad del sistema, como características para protegerlo de un uso malintencionado, destrucción, o intentos de modificación no autorizada del mismo.
- ✗ Características del sistema que faciliten su mantenibilidad.
- ✗ Requisitos de conversión o portabilidad.
- ✗ Propiedades de las interfaces de usuario, interfaces con otros sistemas hardware o software.
- ✗ Extensiones futuras de la aplicación.
- ✗ Información descriptiva o explicativa de algún hecho, lugar geográfico, etc.
- ✗ Información redundante.
- ✗ Información no esencial, es decir, que no necesite tenerse en cuenta en la aplicación, por estar fuera del contexto de la misma.

En definitiva, se eliminan de la descripción textual toda aquella información que no constituya un requisito funcional del sistema a resolver.

Tarea 1.2: Inclusión de información implícita

Como se ha indicado anteriormente, también se hará explícita, durante esta etapa, la información relacionada con los requisitos funcionales que permanezca implícita. Esta información puede provenir de varias fuentes:

- ✗ Información implícita en las estructuras lingüísticas de las especificaciones. Por ejemplo, si en el documento de requisitos de un sistema de compras aparece la estructura “clientes potenciales”, habrá que indicar explícitamente que “los clientes pueden no haber realizado ninguna compra”.
- ✗ Información implícita en el dominio general del conocimiento. Esta información es la que se conoce independientemente del tipo de aplicación a desarrollar. Por ejemplo, se sabe que “un mismo nombre de persona puede hacer referencia a varias personas”.
- ✗ Información implícita en el dominio de la aplicación. Esta información es propia de cada sistema a desarrollar. El conocimiento de la información de este tipo depende

del grado de familiarización del analista con el dominio de la aplicación a desarrollar.

En caso de que la información implícita relativa al problema, no se haga explícita en su totalidad en esta fase, durante los distintos procesos de verificación de los modelos del análisis, se detectarán inconsistencias o lagunas provenientes de falta de información, lo que conducirán a hacer explícita dicha información en iteraciones sucesivas, bien en esta etapa, o recurriendo al usuario durante el proceso de Educación de Requisitos.

La información de la descripción textual que es un requisito funcional, junto con la información implícita añadida, constituye los Requisitos Relevantes, que son la salida de esta etapa. El resumen de toda esta etapa se muestra en la figura 2.12.

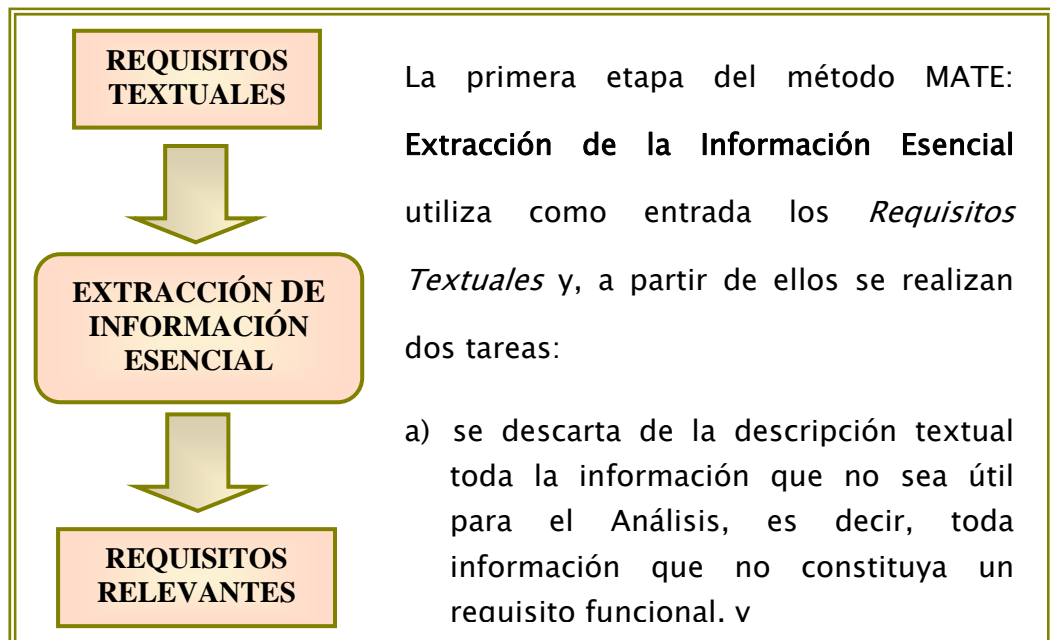


Figura 2.12. Resumen de la Etapa 1 del Método MATE

II.2.5.2. ETAPA 2: IDENTIFICACIÓN DE SINÓNIMOS Y POLISEMAS

El objetivo de esta etapa es eliminar ambigüedades de las especificaciones. En unas especificaciones textuales pueden haber sinónimos, y también se pueden encontrar palabras polisémicas. La existencia de este tipo de información puede dificultar la modelización, e incluso puede provocar una modelización incorrecta. Por lo que es importante detectarla y reemplazarla por las palabras adecuadas. En el caso de los *sinónimos* se ha de buscar un nombre o un grupo nominal único lo más

representativo posible, que va a ser el que proporciona mayor semántica a la estructura nominal en función de cada dominio concreto. En el caso de las *palabras polisémicas*, hay que proporcionar distintos nombres, para diferenciar los distintos significados que adoptan estas palabras.

Para realizar estas actividades se crea un Diccionario de Sinónimos y Polisemas donde se identifica de los **Requisitos Relevantes**, para cada nombre o frase sustantiva con alguna de estas características, una descripción, los sinónimos encontrados en los Requisitos Relevantes que serán sustituidos por el nombre o frase sustantiva descrita, y las palabras polisémicas de ese nombre o frase sustantiva. Este diccionario tiene el formato descrito en la tabla 2.3.

Tabla 2.3. Formato del Diccionario de Sinónimos y Polisemas

<i>Nombre (frase sustantiva):</i>	<i>Descripción:</i> <i>Sinónimos:</i> <i>Polisemas:</i>
<i>Nombre (frase sustantiva):</i>	<i>Descripción:</i> <i>Sinónimos:</i> <i>Polisemas:</i>
...	...

La salida de esta etapa está constituida por el **Diccionario de Sinónimos y Polisemas**, y por los **Requisitos Relevantes Depurados**. Estos últimos son equivalentes a los Requisitos Relevantes en los que se han sustituido los sinónimos y palabras polisémicas que figuran en el Diccionario de Sinónimos y Polisemas. La figura 2.13 muestra un resumen de esta segunda etapa del método.

Se muestra en la tabla 2.4, un ejemplo del mencionado Diccionario de Sinónimos y Polisemas.

II.2.5.3. ETAPA 3: SEPARACIÓN DE LA INFORMACIÓN ESTÁTICA Y DINÁMICA

En las especificaciones útiles de un problema hay información relativa al componente estático y al dinámico del mismo.

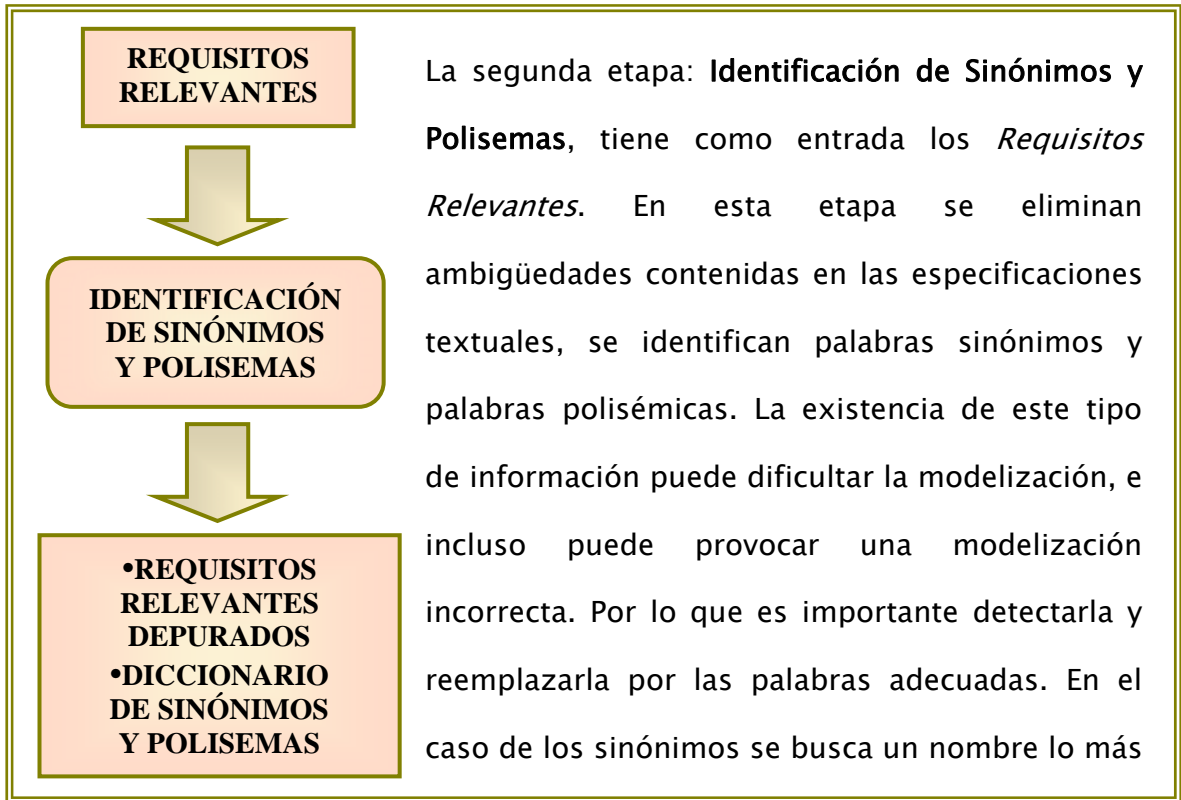


Figura 2.13. Resumen de la Etapa 2 del método MATE

Tabla 2.4. Ejemplo del Diccionario de Sinónimos y Polisemas

<p><i>Nombre (frase sustantiva):</i> Nombre de empresa</p>	<p><i>Descripción:</i> nombre de la empresa de autos <i>Sinónimos:</i> nombre (de empresa) <i>Polisemas:</i> nombre (de modelo) nombre (de vendedor) nombre (de cliente) nombre (de proveedor)</p>
--	--

El primero está compuesto por información con la que va a trabajar el sistema; mientras que el segundo está formado por la descripción de los procesos que operan sobre la información del modelo estático.

El objetivo del método MATE es definir un modelo que represente la parte estática y otro la parte dinámica de la aplicación a analizar. La información contenida en ambos modelos es diferente por lo que el método para desarrollarlos difiere, a su vez, en cada uno de los dos casos. Por ello, se separará la información

relativa a la parte estática de la aplicación y la relativa a la parte dinámica. Del primer tipo de información se crea un MO con las clases, los atributos y las relaciones. De la información relativa al funcionamiento del sistema, se obtiene el MC que contiene información representativa de los eventos que provocan la realización de operaciones sobre los objetos. Estas operaciones dan lugar a métodos que se añaden en las clases correspondientes del MO, análisis que se efectuará en profundidad en la etapa 9 de MATE.

La información descriptiva de la parte estática de la aplicación contiene propiedades estructurales de la información que ha de manejar la aplicación, es decir, contiene la información que ha de guardar el sistema para realizar su cometido. Constituyen los **Requisitos Estáticos** de la aplicación.

La información descriptiva de la parte dinámica de la aplicación especifica interacciones y eventos que afectan a la información descrita en la parte anterior. Indicará cuál es el comportamiento esperado del sistema a desarrollar. Constituyen los **Requisitos Dinámicos** de la aplicación.

No existe en el lenguaje natural ningún tipo de expresiones predefinidas para representar ambos tipos de información, por lo que se carece de un algoritmo definido para llevar a cabo el proceso de separación de esta información. Se puede dar alguna guía, como por ejemplo que la información que describe el comportamiento de un sistema puede ir acompañada de expresiones temporales y condicionales (cuando....., si.....). Sin embargo, también se podría expresar el comportamiento de un sistema sin recurrir a este tipo de expresiones. El estudio de las distintas relaciones entre las oraciones de un documento, es lo que se denomina Análisis del Discurso [Hobbs, 90], y es una de las ramas de la lingüística más compleja. No se proporcionan reglas objetivas para identificar los tipos de relaciones, depende en cada caso del documento tratado, de la información implícita, y del conocimiento del dominio del lector.

Por todo ello, la realización de esta tarea por el analista dependerá del contenido de los **Requisitos Relevantes Depurados** y del **Diccionario de Sinónimos y Polisemas** obtenidos en la etapa anterior. Esta desventaja podría ser tomada como un reto a resolver por los ingenieros del software.

Una breve descripción de esta etapa se encuentra en la figura 2.14.

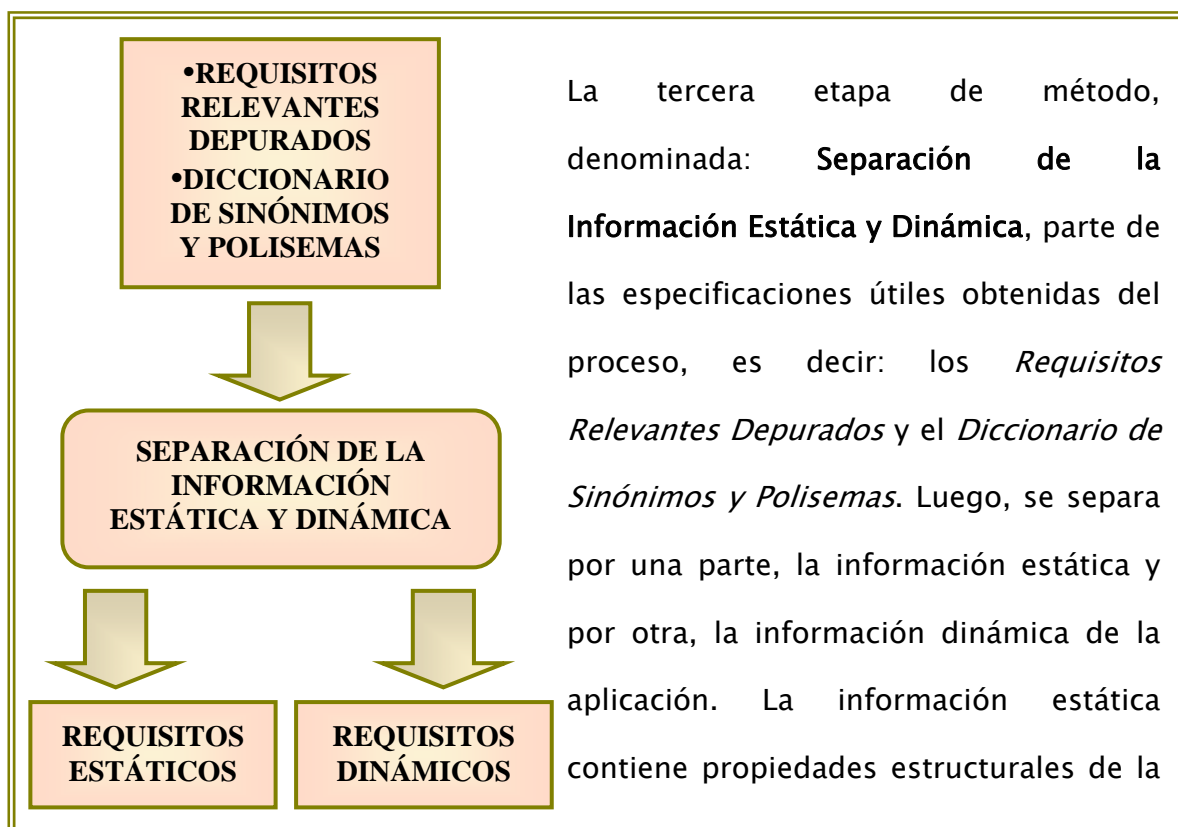


Figura 2.14. Resumen de la Etapa 3 del método MATE

II.2.5.4. ETAPA 4: ESTRUCTURACIÓN DE LOS REQUISITOS ESTÁTICOS

Los **Requisitos Estáticos** se utilizan para la construcción del MO. Para determinar qué información lingüística está asociada con los componentes del MO, se estructuran los Requisitos Estáticos mediante la gramática del *Lenguaje de Utilidad Estático* (sección II.2.4.1, figura 2.9). El lenguaje de utilidad que se obtiene tras esta etapa recibe el nombre de **Lenguaje de Utilidad Estático Preliminar**. Este lenguaje será validado posteriormente dando lugar al *Lenguaje de Utilidad Estático*. En la figura 2.15 se encuentra un resumen de esta etapa.

Para transformar los Requisitos Estáticos en base a la gramática descrita, se han de seguir las guías que se exponen a continuación:

- ✘ Emplear oraciones simples enunciativas afirmativas numeradas consecutivamente y etiquetadas con el identificador del patrón lingüístico al que pertenecen.
- ✘ El verbo vendrá expresado en tercera persona del singular o plural, ya sea en voz activa o en pasiva.

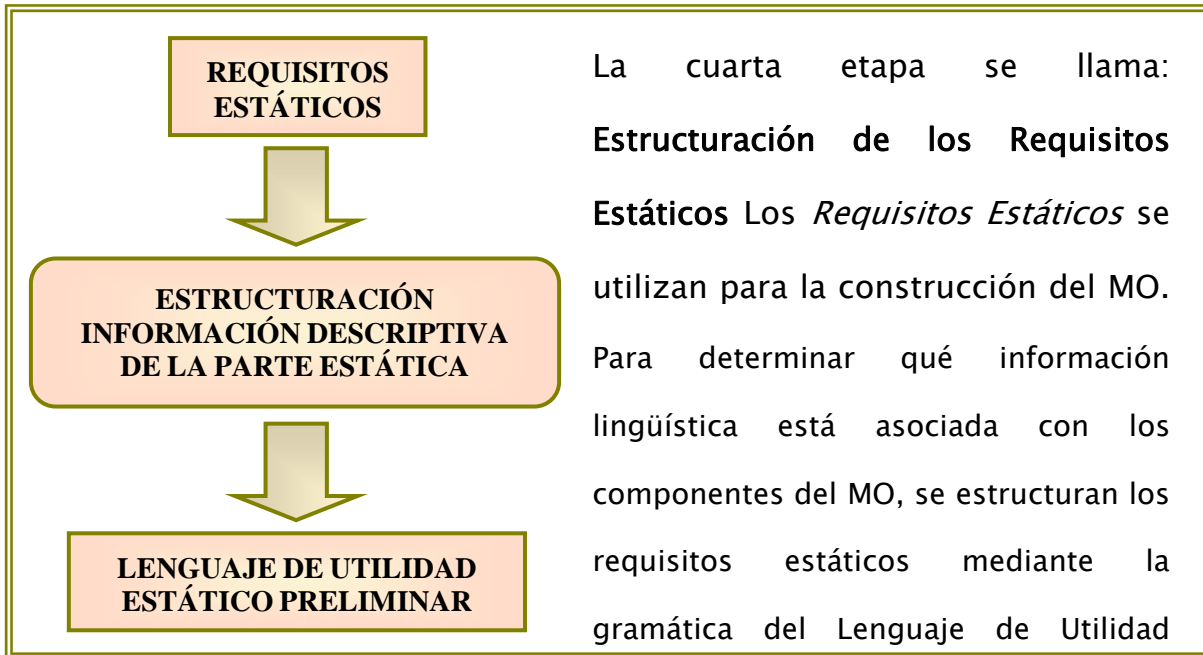


Figura 2.15. Resumen de la Etapa 4 del método MATE

- ✘ Las oraciones de tipo “hay X en Y”, o “existe X en Y” se pondrán de la forma “Y tiene X”.
- ✘ Las oraciones en las que no sea posible indicar el sujeto o el complemento, se eliminarán de esta parte, puesto que si realmente proporcionan información significativa, dicho sujeto o complemento aparecerá en otra oración con la estructura mencionada ya sea en la función de sujeto o de cualquiera de los complementos.
- ✘ Sustituir los pronombres que no formen parte de la estructura verbal, por el nombre o grupo nominal al que reemplazan.
- ✘ Cuando exista un modificador que referencia a varios sintagmas nominales, se hará explícita dicha referencia en cada uno de ellos. Por ejemplo, la oración “una persona tiene al menos una dirección y un teléfono de contacto”, se sustituirá por “una persona tiene al menos una dirección y al menos un teléfono de contacto”.
- ✘ Eliminar todas las referencias al sistema, por ejemplo sustituir la frase “el sistema genera una factura por cada contrato” por “cada contrato da lugar a una factura”.
- ✘ Eliminar los adverbios, y los adjetivos modales (por ejemplo, probable o necesario).

- ✘ Si aparecen explícitamente adjetivos cuantificativos definidos numerales (dos, cien, una decena, ...) como modificadores del sujeto o de los complementos, se sustituirán por los adjetivos cuantificativos indefinidos más representativos (varios, algunos, ...). Por ejemplo, la frase “un pedido puede tener hasta 20 productos” se reemplazaría por “un pedido puede tener varios productos”. Durante el proceso de diseño sí tendría importancia el adjetivo numeral, para determinar el número de registros o de accesos a determinada información. Ello podría dar lugar a distintas optimizaciones. Sin embargo, en el proceso de análisis este tipo de información no se considera, ya que durante este proceso se trabaja únicamente con la información relevante del problema para construir un modelo conceptual.

II.2.5.5. ETAPA 5: ESTRUCTURACIÓN DE LOS REQUISITOS DINÁMICOS

De los **Requisitos Dinámicos** se construye el MC. Para definir el MC asociado con una descripción en lenguaje natural de un sistema, es necesario restringir dicha descripción en lenguaje natural, mediante el *Lenguaje de Utilidad Dinámico*.

Lo que conlleva a estructurar los Requisitos Dinámicos según la gramática del Lenguaje de Utilidad Dinámico (sección II.2.4.2, figura 2.10). El lenguaje de utilidad que se obtiene tras esta etapa recibe el nombre de **Lenguaje de Utilidad Dinámico Preliminar**, que será validado posteriormente por el usuario dando lugar al *Lenguaje de Utilidad Dinámico*.

La descripción del comportamiento de un sistema se estudiará independientemente por cada “caso de uso”, de dicho sistema, según el método MATE. Se entiende por caso de uso, la expresión de las diferentes formas en que los usuarios u otros sistemas externos pueden usar el sistema a construir, o bien, las diferentes formas en que este sistema puede funcionar de forma autónoma. En un caso de uso, todas las acciones que realiza un sistema están relacionadas unas con otras. Por tanto, al describir el comportamiento del sistema, se han de hacer explícitas dichas relaciones que, como ya se ha indicado, pueden existir expresadas de múltiples maneras.

El Lenguaje de Utilidad Dinámico Preliminar estará constituido por las descripciones de cada uno de los casos de uso del sistema. Es conveniente realizar

estas descripciones en orden cronológico respecto al proceso de resolución del problema, ya que esto facilitará la construcción posterior del MC.

Las estructuras lingüísticas que constituyen el Lenguaje de Utilidad Dinámico Preliminar se definen teniendo en cuenta las siguientes guías por cada caso de uso:

- ✦ Numerar las oraciones de cada patrón lingüístico de la forma: número de caso de uso. número de oración que lo describe. Por ejemplo, las n oraciones del primer caso de uso que se estudia están numeradas como: 1.1, 1.2, ..., 1. n , y así sucesivamente para los restantes casos de uso.
- ✦ Tanto en la oración subordinada como en la principal, se sustituyen los pronombres por los grupos nominales a los que reemplazan.
- ✦ Los adverbios o expresiones adverbiales de tiempo, como, por ejemplo, “mensualmente”, “diariamente”, ..., se transforman usando la conjunción *sí*, por ejemplo: “si llega el final del mes”.
- ✦ Tanto la oración-subordinada como la oración-principal tendrán los verbos en activa.

La salida de esta tarea está formada por un conjunto de oraciones condicionales que constituyen el Lenguaje de Utilidad Dinámico Preliminar. El resumen correspondiente a esta etapa se muestra en la figura 2.16.

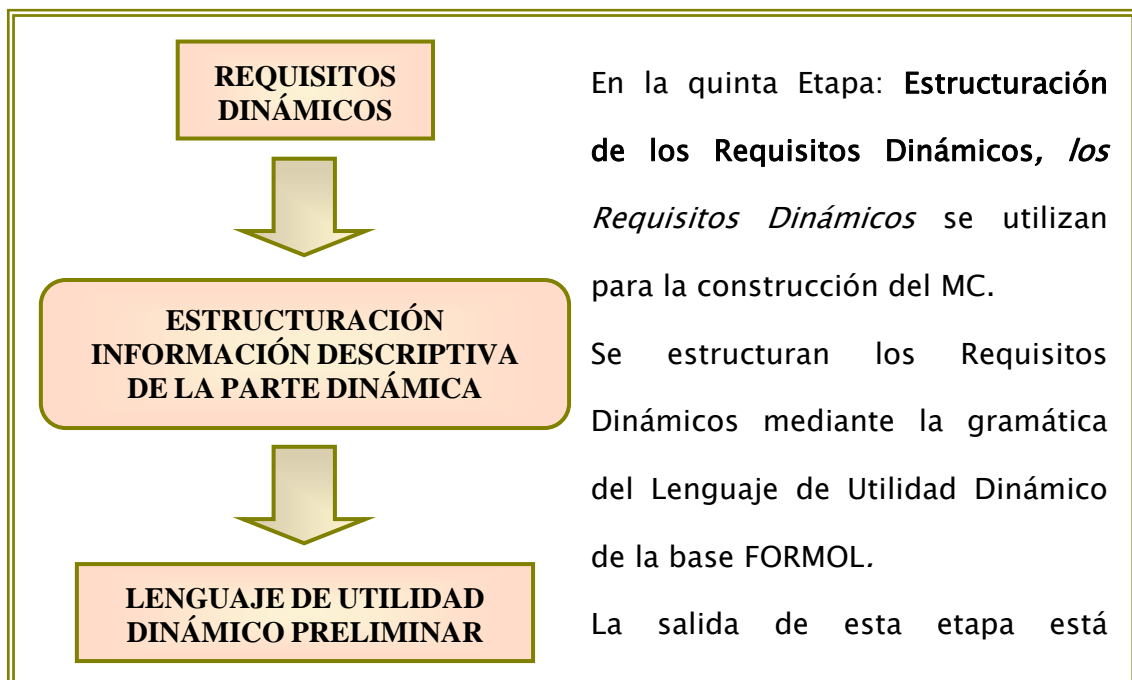


Figura 2.16. Resumen de la Etapa 5 del método MATE

El conjunto de oraciones obtenidas tras aplicar las etapas 4 y 5, es decir, el Lenguaje de Utilidad Estático Preliminar y el Lenguaje de Utilidad Dinámico Preliminar, es la información que constituye el punto de partida para el proceso de construcción del MO y del MC, por ello el analista se ha de asegurar que esta información representa adecuadamente lo que el usuario ha expresado en la descripción textual. Las restricciones propuestas para la construcción de los Lenguajes de Utilidad son simples y no tienen por qué cambiar la semántica de la información que transforman. Sin embargo, sería una buena práctica, presentar estos Lenguajes de Utilidad Preliminares al usuario, con el fin de asegurar la equivalencia semántica entre la información contenida en éstos y la información que el usuario tiene en mente. Por otra parte, este proceso sirve para validar la información implícita que se ha añadido en la Etapa 2: Extracción de la Información Esencial. Esta tarea es la que se realiza en el proceso de Validación del Lenguaje de Utilidad. El resultado de este proceso de validación da lugar al Lenguaje de Utilidad Estático y al Lenguaje de Utilidad Dinámico.

II.2.5.6. ETAPA 6: CONSTRUCCIÓN DEL MODELO DE OBJETOS

Durante esta etapa se realizan un conjunto de procedimientos, que, a partir del Lenguaje de Utilidad Estático, proporciona la modelización conceptual de la estructura estática del sistema. Para ello se han de realizar las tareas que muestran en la figura 2.17.

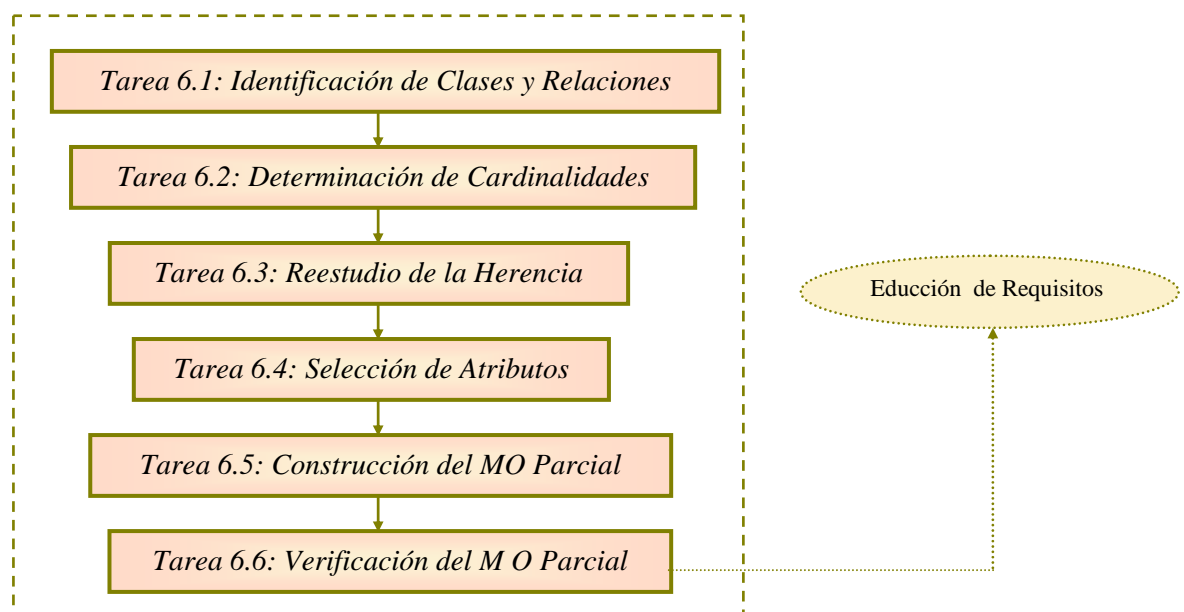


Figura 2.17. Tareas de la Etapa 6: Construcción del Modelo de Objetos

Tarea 6.1. Identificación de las clases y relaciones, las clases y los distintos tipos de relaciones formarán las piezas clave del MO. En esta tarea se aplican los resultados proporcionados por FORMOL.

Tarea 6.2. Definición de las cardinalidades, relativas a las relaciones de agregación y asociaciones.

Tarea 6.3. Reestudio de la herencia, para aumentar el grado de compartición de características entre las superclases y subclases.

Tarea 6.4. Selección de atributos, entre las clases identificadas hasta el momento.

Tarea 6.5. Construcción del MO Parcial, donde se construye el MO con las clases, atributos, relaciones y cardinalidades. Se denomina parcial, porque no es el MO completo, ya que faltan por determinar los métodos.

Tarea 6.6. Verificación del MO Parcial, donde se estudia la completud y consistencia del modelo, dando lugar a nuevas iteraciones si fuera necesario.

Los productos con los que se trabaja durante la Etapa 6: Construcción del MO se muestran en la figura 2.18.

Las estructuras lingüísticas que componen el Lenguaje de Utilidad Estático son la entrada de la Tarea 6.1: Identificación de Clases y Relaciones, donde se asigna a cada una de ellas alguno de los patrones conceptuales estáticos, con los que se obtienen las Clases y Relaciones del MO. De las Clases y Relaciones y del Lenguaje de Utilidad Estático, se obtienen Cardinalidades de las Relaciones aplicando una serie de guías durante la Tarea 6.2: Determinación de Cardinalidades. En la Tarea 6.3: Reestudio de la Herencia, se estudian las Clases y Relaciones con el fin de aumentar el grado de compartición de características similares entre clases. Durante esta tarea se pueden obtener Jerarquías de Herencia Optimizadas. A partir de estas Jerarquías de Herencia Optimizadas y de las demás Clases y Relaciones, se obtienen los Atributos de las Clases, en la Tarea 6.4: Identificación de Atributos, donde se identifican aquellas clases que son, en realidad, atributos de otras clases. Con la información de las tareas anteriores se construye *el MO Parcial*, en la Tarea 6.5: Construcción del MO Parcial, que contiene las clases, los atributos, las relaciones y las cardinalidades. A este modelo le faltan los métodos para ser completo. Dichos métodos se añadirán en una etapa posterior. Finalmente, se realizará la Tarea 6.6: Verificación del MO Parcial donde se obtendrá el **MO Parcial Verificado**, que se emplea posteriormente en la Etapa 7: Construcción del MC, y en la Etapa 8: Integración del MO y del MC.

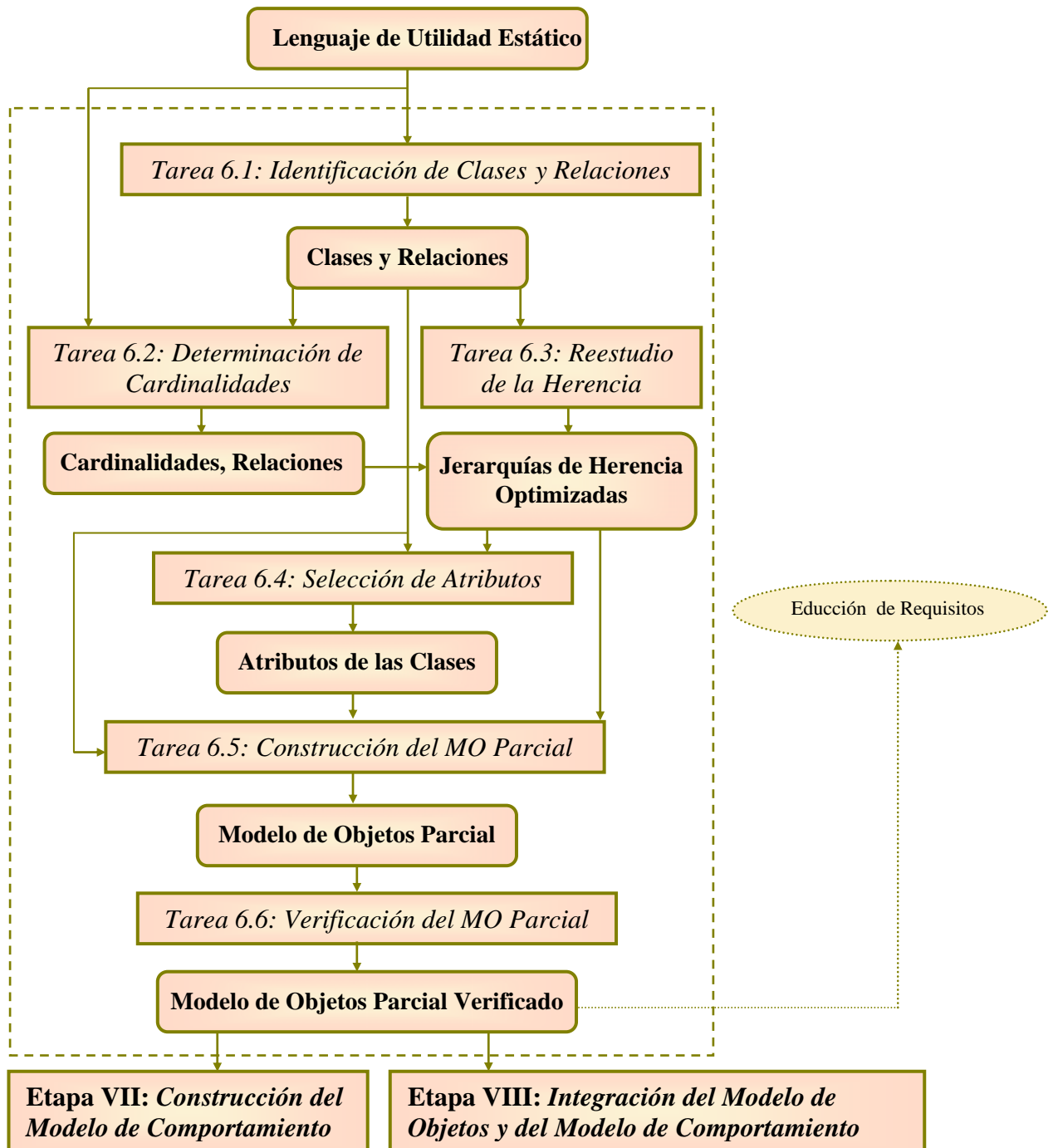


Figura 2.18. Tareas y Productos de la Etapa 6: Construcción del Modelo de Objetos

La figura 2.19 muestra un resumen de la descripción de esta etapa.

II.2.5.7. ETAPA 7: CONSTRUCCIÓN DEL MODELO DE COMPORTAMIENTO

A partir del Lenguaje de Utilidad Dinámico, se construye. Para realizar la construcción del MC se han de llevar a cabo dos tareas que se muestran en la figura 2.20. Estas tareas son las que se detallan a continuación.

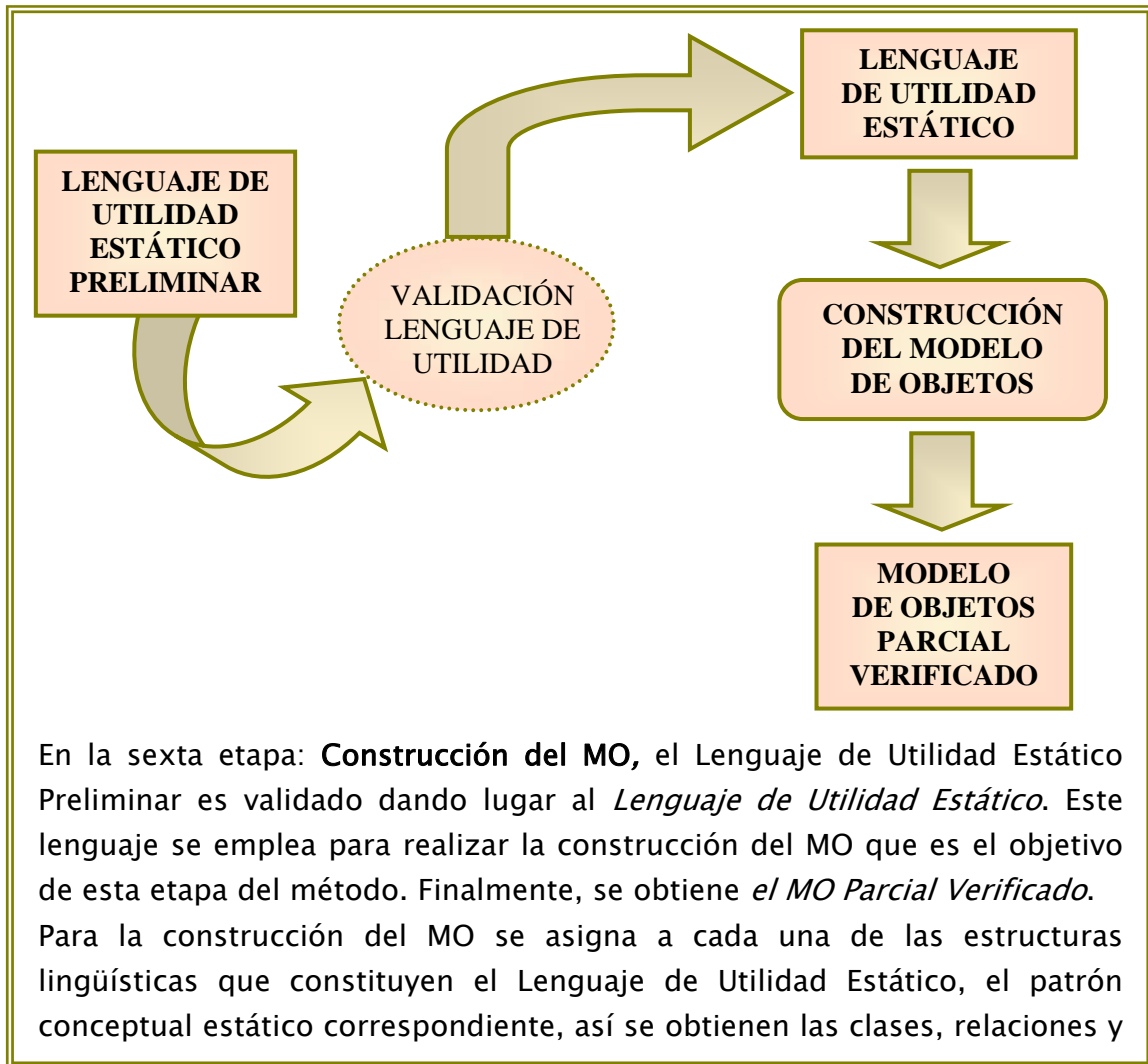


Figura 2.19. Resumen de la Etapa 6 del método MATE

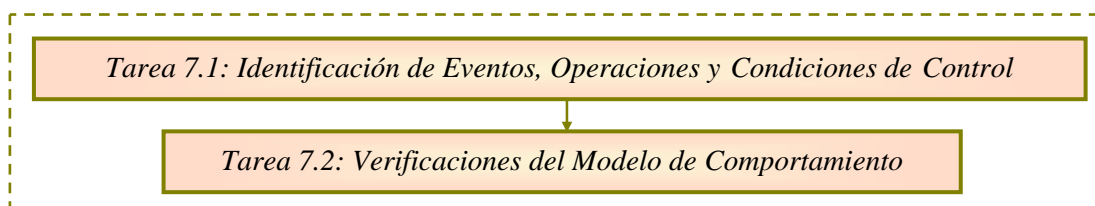


Figura 2.20. Tareas de la Etapa 7: Construcción del Modelo de Comportamiento

Tarea 7.1. Identificación de Eventos, Operaciones y Condiciones de Control, donde se identifican los componentes del MC, y se construye el modelo correspondiente.

Tarea 7.2. Verificación del MC, donde se comprueba la consistencia de dicho modelo.

En la Tarea 7.1: Identificación de Eventos, Operaciones y Condiciones de Control, se extraen estos componentes aplicando un determinado patrón de modelización sobre las oraciones que constituyen la entrada de esta tarea, y

tomando como referencia las clases del **MO Parcial**. El resultado de la Tarea 7.1 es *el MC*, que será verificado en la Tarea 7.2: Verificación del MC, dando lugar al **MC Verificado**. Este modelo se empleará en las Etapa 8: Integración del MO y del MC, y en la Etapa 9: Verificación del MO y del MC. Los productos generados por las tareas de la figura 2.20, aparecen en la figura 2.21.

En la figura 2.22, se detalla la descripción de la séptima etapa del método MATE.

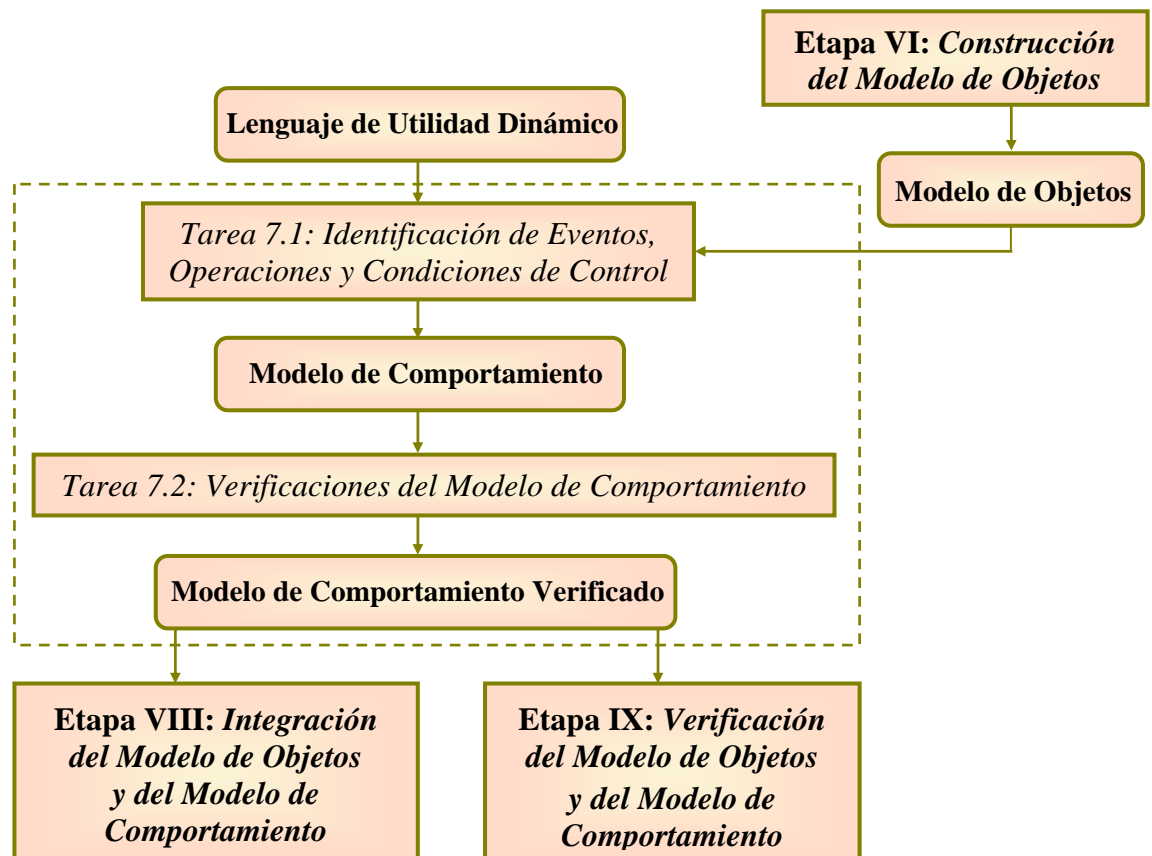


Figura 2.21. Tareas y Productos de la Etapa 7: Construcción del Modelo de Objetos

II.2.5.8. ETAPA 8: INTEGRACIÓN DEL MODELO DE OBJETOS Y DEL MODELO DE COMPORTAMIENTO

Las operaciones de los objetos de las clases del **MO** son las operaciones internas del **MC**. Estas operaciones provocan cambios en el estado de los objetos del sistema. Las clases a las que pertenecen estos objetos, serán las contenedoras de estas operaciones. Para realizar esta asignación se realizan los siguientes pasos:

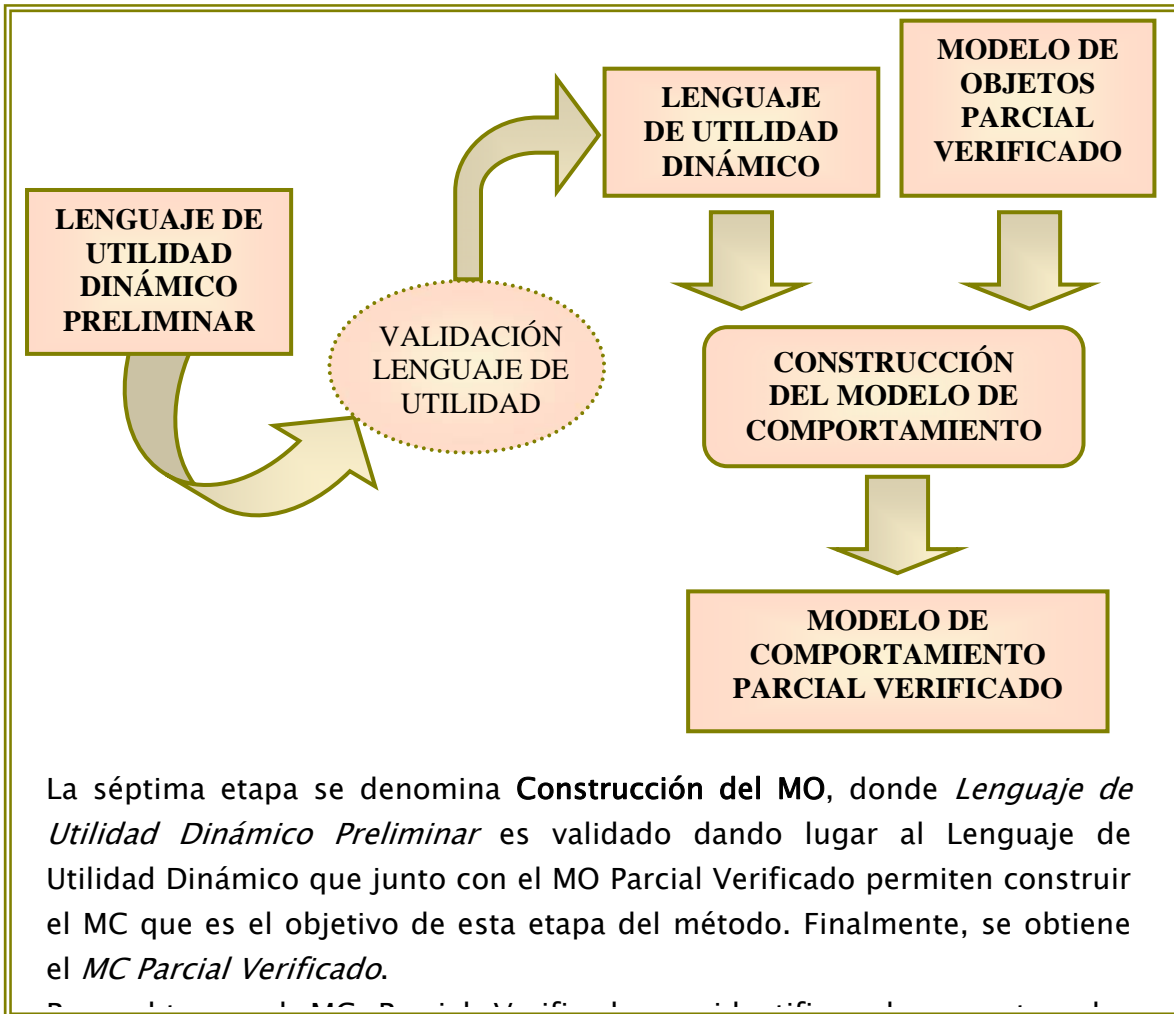


Figura 2.22. Resumen de la Etapa 7 del método MATE

1. Definir una matriz de clases y operaciones, cuyas filas corresponden a las clases identificadas en el **MO Parcial**, y las columnas a las operaciones internas básicas del **MC Verificado**.
2. Los nombres de las operaciones vendrán indicados por el verbo en infinitivo seguido por los núcleos de los complementos que los acompañan en singular. Se cumplimenta la matriz marcando la casilla m_{ij} , siempre que la operación j suponga una operación básica sobre algún objeto de la clase i .

Si en la columna correspondiente a una operación hay más de una casilla marcada, todas las marcas deben corresponder a filas de clases pertenecientes a algún subconjunto de una rama de una jerarquía de herencia. En caso contrario, se debe volver a la etapa anterior, ya que este caso indicaría que no todas las operaciones obtenidas son básicas.

En la tabla 2.5, se muestra un ejemplo suponiendo identificadas las clases y jerarquía de herencia de la figura 2.23, y constituyendo O1, O2, O3, O4 y O5 las operaciones.

2. Asignar cada operación a una clase. Para ello estudiar cada operación:

2.1 Si en su columna correspondiente hay solo una marca, se asignará a la clase correspondiente a la fila marcada.

2.2 Si en su columna correspondiente hay varias marcas, se asignará a aquella clase marcada de nivel superior en la jerarquía de herencia cuyos hijos estén todos marcados.

Tabla 2.5. Matriz Clases / Operaciones

	O1	O2	O3	O4	O5
C1		x			x
C2				x	x
C3		x			x
C4	x				
C5			x		

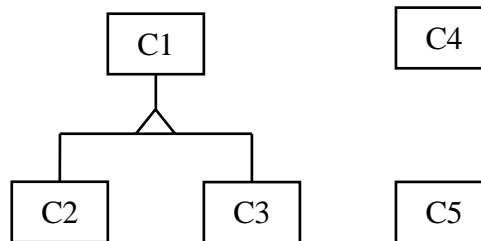


Figura 2.23. Clases y Jerarquías de Herencia

En el ejemplo anterior y siguiendo estas reglas la asignación de métodos sería:

C1 : O5 C3 : O2 C5 : O3
 C2 : O4 C4 : O1

La salida de esta etapa será el **MO**, con todos sus componentes, clases, atributos, relaciones, cardinalidades y operaciones.

La figura 2.24 muestra un resumen de la penúltima etapa de MATE.

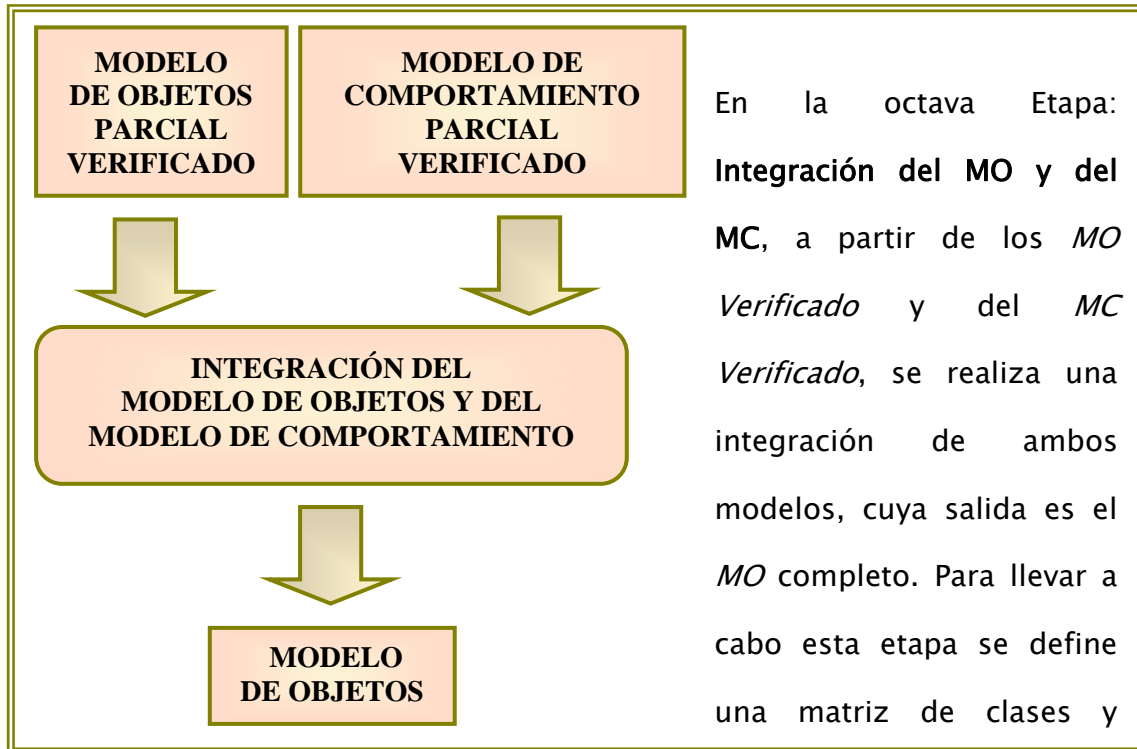


Figura 2.24. Resumen de la Etapa 8 del método MATE

II.2.5.9. ETAPA 9: VERIFICACIÓN DE LOS MODELOS DE OBJETOS Y DE COMPORTAMIENTO

En esta etapa se realiza una verificación conjunta de los modelos que constituyen la modelización conceptual del sistema software. Para ello se comprueba, que entre las clases a las que pertenecen los métodos, que se comunican entre sí en el **MC** a través de eventos, existe una relación en el **MO**. A través de estas relaciones se comunican dichas clases para el intercambio de mensajes, es decir, para implementar los eventos del MC.

La descripción de esta última etapa se encuentra en la figura 2.25.

II.2.5.10. RESUMEN DE TODAS LAS ETAPAS DEL MÉTODO MATE

En la tabla 2.6 se muestran las entradas, descripción y salida que cada etapa involucrada en el método MATE.

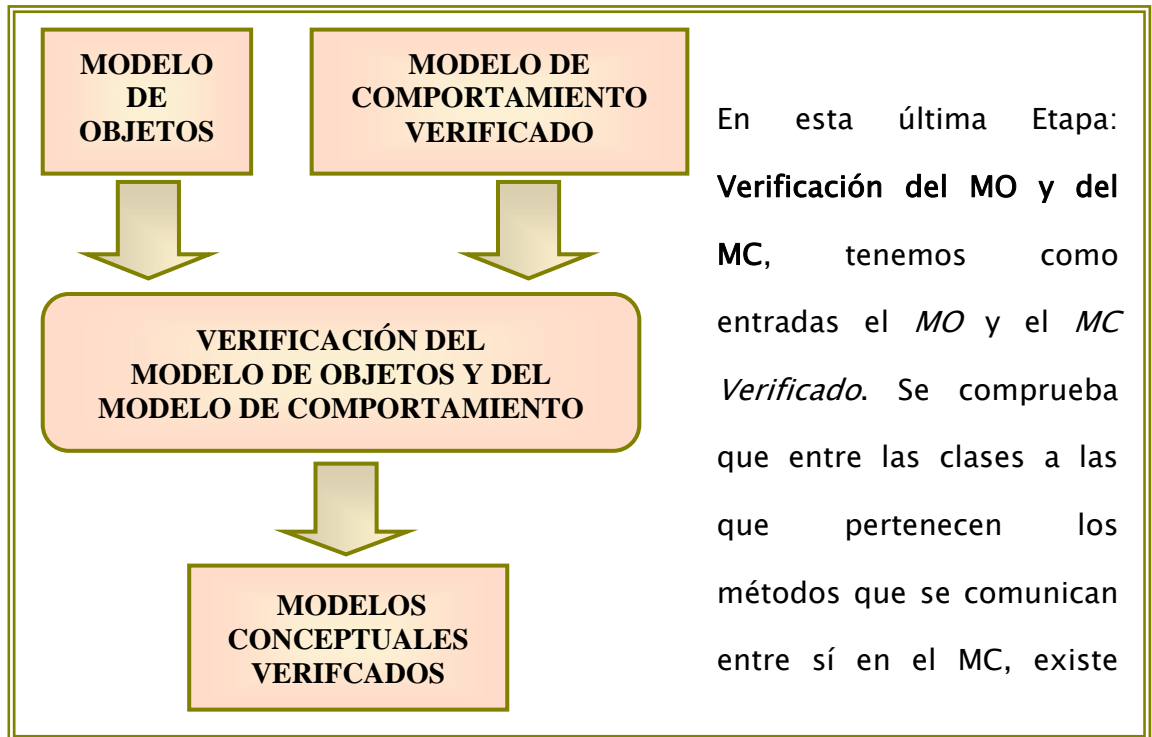


Figura 2.25. Resumen de la Etapa 9 del método MATE

Tabla 2.6. Documentos de entrada y salida y descripción de las etapas del método MATE

Etapa	Entrada	Descripción	Salida
1. Extracción de la Información Esencial	- <i>Requisitos Textuales</i>	Se elimina la información que no constituye un requisito funcional y se incluye información implícita.	- <i>Requisitos Relevantes</i>
2. Identificación de Sinónimos y Polisemas	- <i>Requisitos Relevantes</i>	Se eliminan ambigüedades de las especificaciones.	<i>Diccionario de Sinónimos y Polisemas</i>
3. Separación de la Información Estática y Dinámica	- <i>Requisitos Relevantes Depurados</i> - <i>Diccionario de Sinónimos y Polisemas</i>	Se separa la información relativa a la parte estática y a la parte dinámica de la aplicación.	- <i>Requisitos Estáticos</i> - <i>Requisitos Dinámicos</i>
4. Estructuración de los Requisitos Estáticos	- <i>Requisitos Estáticos</i>	Se estructuran los requisitos estáticos mediante la gramática del Lenguaje de Utilidad Estático de FORMOL.	- <i>Lenguaje de Utilidad Estático</i>

Tabla 2.6. Documentos de entrada y salida y descripción de las etapas del método MATE (Continuación)

5. Estructuración de los Requisitos Dinámicos	- <i>Requisitos Dinámicos</i>	Se estructuran los requisitos dinámicos mediante la gramática del Lenguaje de Utilidad Dinámico de FORMOL.	- <i>Lenguaje de Utilidad Dinámico</i>
6. Construcción del MO	- <i>Lenguaje de Utilidad Estático</i>	Se identifican clases y relaciones, cardinalidades, herencia, atributos, y se construye y verifica el MO Parcial.	- <i>MO Parcial Verificado</i>
7. Construcción del MC	- <i>MO Parcial Verificado</i> - <i>Lenguaje de Utilidad Dinámico</i>	Se identifican eventos, operaciones y condiciones de control. Se construye el MC y se verifica el MC.	- <i>MC Verificado</i>
8. Integración del MO y MC	- <i>MO Parcial Verificado</i> - <i>MC Verificado</i>	Se define una matriz de clases y operaciones. Las filas son las clases del MO Parcial. Las columnas son las operaciones internas básicas del MC Verificado.	- <i>MO</i>
9. Verificación del MO y MC	- <i>MO, MC Verificado</i>	Se comprueba que entre las clases a las que pertenecen los métodos que se comunican entre sí en el MC a través de eventos exista una relación en el MO.	- <i>MO Verificado</i> - <i>MC Verificado</i>

II.2.6. PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE

× Descripción

El PU es un proceso de desarrollo de software. Es un proceso *bien definido* y *bien gestionado*, es una solución al proceso del software.

PU es más que un simple proceso, es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto.

Está basado en componentes, lo cual quiere decir que el sistema software en construcción esta formado por componentes software interconectados a través de interfaces bien definidas.

Es un proceso de la IS que ofrece una aproximación disciplinada a la asignación de tareas y responsabilidades de una organización de desarrollo.

Su objetivo es asegurar la producción de software de alta calidad que satisfaga las necesidades de los usuarios dentro de un calendario y un presupuesto predecibles. Es decir, consta de un conjunto de actividades que se necesitan para transformar los requisitos del usuario en un sistema software.

Para hacer que esto funcione se necesita que el PU sea polifacético, es decir, que tenga en cuenta ciclos, fases, flujos de trabajo, gestión de riesgo, control de calidad, gestión del proyecto y control de la configuración. El PU encuadra e integra esas diferentes facetas. De esta forma, los desarrolladores pueden construir herramientas que soporten la automatización del proceso entero, de cada flujo de trabajo individualmente, de la construcción de los diferentes modelos y de la integración del trabajo a lo largo del ciclo de vida.

El PU utiliza el Lenguaje Unificado de Modelado reconocido por las siglas UML, del inglés, Unified Modeling Language, para preparar todos los esquemas de un sistema software. De hecho, UML es una parte esencial del PU, sus desarrollos fueron paralelos.

✘ Evolución

El PU es el producto final de tres décadas de desarrollo y uso práctico. Su camino como producto sigue una trayectoria desde el Proceso Objectory pasando por el Proceso Objectory de Rational hasta el Proceso Unificado de Rational. Su desarrollo ha recibido peso de muchas fuentes y la influencia sobre el producto de los métodos de Ericsson y Rational; las principales raíces del PU se reflejan en la figura 2.26.

El PU tiene la alianza de técnicas de desarrollo, a través de UML y la combinación del trabajo de muchos metodologistas. Surge de la unificación de tres metodologías de desarrollo basadas en el paradigma de la OO: OOSE (Object Oriented Software Engineering), Booch y OMT (Object Modeling Technique).

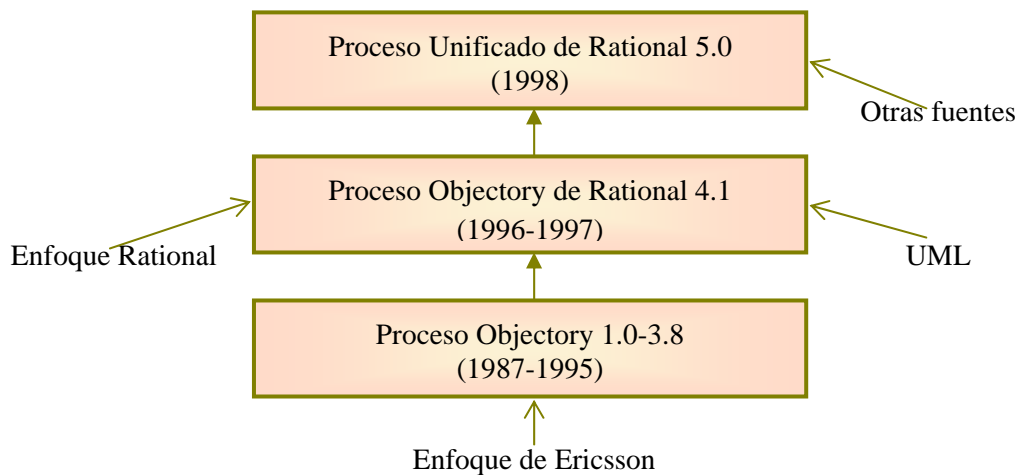


Figura 2.26. Evolución del PU

✘ Frases Claves

Los verdaderos aspectos definitorios del PU se resumen en tres frases claves: dirigido por casos de uso, centrado en la arquitectura e iterativo e incremental. Esto es lo que hace único al PU.

- Está *dirigido por casos de usos*: una iteración es un caso de uso. Un caso de uso es un fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante. Los casos de uso representan los requisitos funcionales. Todos los casos de uso juntos constituyen el modelo de casos de uso, el cual describe la funcionalidad total del sistema. Los casos de uso no son sólo una herramienta para especificar los requisitos del sistema, sino que guían su diseño, implementación y prueba; esto es, guían el proceso de desarrollo. Basándose en el modelo de casos de uso, los desarrolladores crean una serie de modelos de diseño e implementación que llevan a cabo los casos de uso. Los desarrolladores revisan cada uno de los modelos para que se correspondan conforme al modelo de casos de uso. Dirigido por casos de usos quiere decir que el proceso de desarrollo sigue un hilo conductor, es decir avanza a través de una serie de flujos de trabajo que parten de los casos de uso. Los casos de uso se especifican, se diseñan y los casos de uso finales son la fuente a partir de la cual ingenieros de prueba construyen sus casos de prueba. Aunque los casos de uso guían el proceso, no se desarrollan aisladamente: guían la arquitectura del sistema y la arquitectura del sistema

influye en la selección de los casos de uso. Por ende, tanto la arquitectura como los casos de uso maduran según avanza el ciclo de desarrollo.

- Está *centrado en la arquitectura*: la arquitectura en un sistema software se describe mediante diferentes vistas del sistema en construcción. El concepto de arquitectura software incluye los aspectos estáticos y dinámicos más significativos del sistema. La arquitectura surge de las necesidades de la empresa, como las perciben los usuarios, los inversores y se refleja en los casos de uso. La arquitectura es una vista del diseño completo con las características más importantes resaltadas, dejando de lado los detalles. Cada producto tiene tanto una función como una forma: la función corresponde a los casos de uso y la forma a la arquitectura. Por un lado, los casos de uso deben encajar en la arquitectura cuando se llevan a cabo. Por otro lado, la arquitectura debe permitir el desarrollo de todos los casos de uso requeridos, ahora y en el futuro. Tanto la arquitectura como los casos de uso deben evolucionar en paralelo. El arquitecto debe poseer una comprensión general de los casos de uso antes de comenzar la creación del esquema arquitectónico. Los casos de uso se especifican y maduran, entonces se descubre más de la arquitectura; esto, a su vez, lleva a la maduración. Este proceso continúa hasta que se considere que la arquitectura es estable.
- Es *iterativo e incremental*: es práctico dividir el trabajo en partes pequeñas o mini proyectos. Cada mini proyecto es una iteración que resulta de un incremento. Las iteraciones hacen referencia a pasos en el flujo de trabajo (comúnmente denominadas etapas del desarrollo de software) y los incrementos, al crecimiento del producto. Para una efectividad máxima, las iteraciones deben estar controladas, esto es, deben seleccionarse y ejecutarse de una forma planificada. Es por esto por lo que son mini proyectos. En cada iteración, los desarrolladores identifican y especifican los casos de uso relevantes, crean un diseño utilizando la arquitectura seleccionada como guía, implementan el diseño mediante componentes y verifican que los componentes satisfacen los casos de uso. Si una iteración cumple con sus objetivos, el desarrollo continúa con la siguiente iteración. Cuando una iteración no cumple sus objetivos, los desarrolladores deben revisar sus decisiones previas y probar con un nuevo enfoque. La

arquitectura proporciona la estructura sobre la cual guiar las iteraciones, mientras que los casos de uso definen los objetivos y dirigen el trabajo de cada iteración.

✘ Ciclo de Vida

El PU se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema. Cada ciclo concluye con una versión del producto para los clientes. Cada ciclo consta de cuatro fases: *Iniciación* (se define el ámbito y alcance del proyecto y se desarrollan los casos de negocio), *Elaboración* (se planifica el proyecto, se especifican en detalle la mayoría de los casos de usos (requisitos funcionales) y se diseña la arquitectura base del sistema), *Construcción* (se produce el producto software) y *Transición* (el producto se convierte en una versión beta operando en la comunidad de usuarios finales, se corrigen los problemas y se incorporan mejoras sugeridas en la revisión). Cada fase se subdivide a la vez en iteraciones, como ya se ha dicho y se puede observar en la figura 2.27.

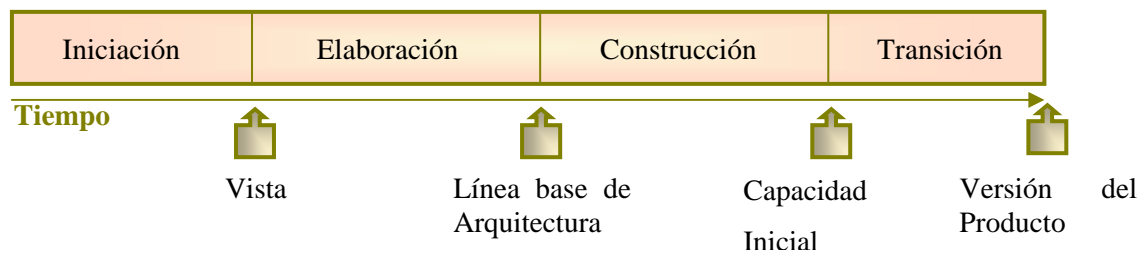


Figura 2.27. Flujos de Trabajo del PU basado en [Jacobson, 00]

✘ Productos Finales

Cada versión produce una nueva versión del sistema y cada versión es un producto preparado para su entrega. Consta de un cuerpo de código fuente incluido en componentes que puede compilarse y ejecutarse, además de manuales y otros productos asociados. El producto terminado debe ajustarse a las necesidades del usuario. Incluye requisitos, casos de uso, especificaciones no funcionales y casos de prueba. Incluye el modelo de la arquitectura y el modelo visual, es decir artefactos modelados por UML.

A medida que los objetivos del sistema se comprenden mejor, los propios requisitos pueden cambiar. Los desarrolladores deben afrontar cada nuevo ciclo de desarrollo del software, por lo que necesitan todas las representaciones del producto

software para el desafío. La figura 2.28 muestra la dependencia que existe entre los diferentes modelos o representaciones del producto software.

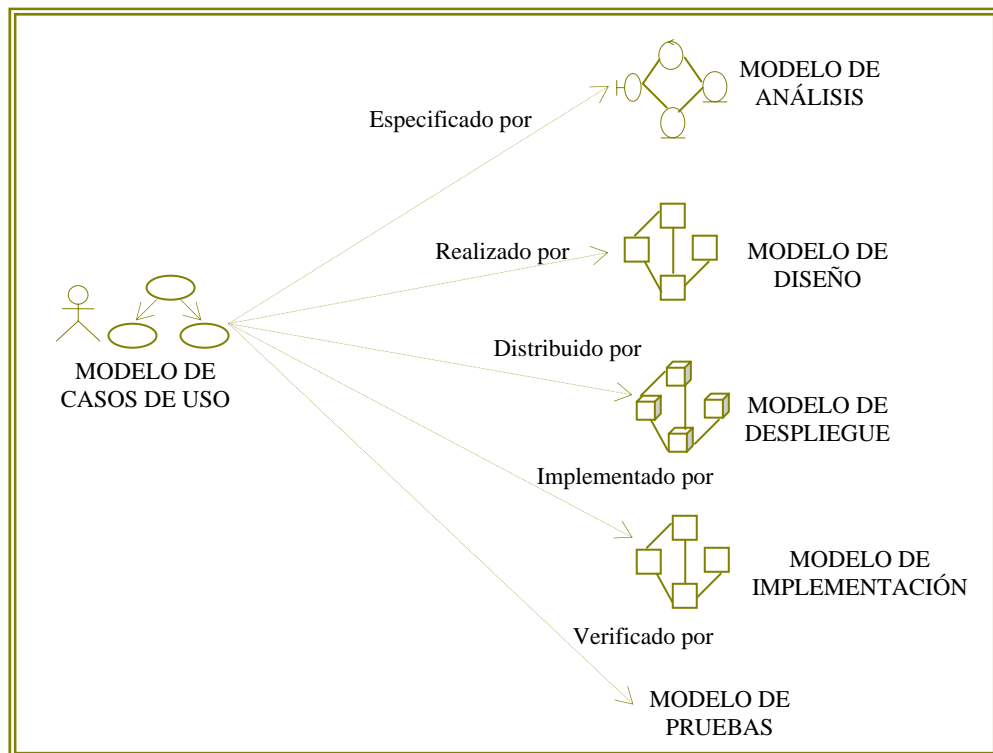


Figura 2.28. Los Modelos del PU basado en [Jacobson, 00]

✘ Diferentes Modelos

La construcción de un sistema es un proceso de construcción de modelos. La misma, utiliza diferentes modelos para describir todas las perspectivas distintas del sistema, sus relaciones y restricciones.

Para llevar a cabo el ciclo de manera eficiente se necesitan todas las representaciones o modelos de elaboración del producto software para el PU, ellos son: *Modelo de Caso de Usos* (con todos los casos de uso y su relación con los usuarios), *Modelo de Análisis*, *Modelo de Diseño*, *Modelo de Implementación*, *Modelo de Despliegue* y *Modelo de Prueba*. En la tabla 2.7, se especifican brevemente características de los mismos.

El sistema debe tener también un *Modelo del Dominio* o *Modelo del Negocio* el cuál, determina los procesos más importantes que describe el contexto del negocio en el que se halla el sistema; y, por supuesto, una representación

arquitectónica. Todos estos modelos están relacionados y juntos representan el sistema como un todo.

Tabla 2.7. Modelos del PU

Modelos	Descripción
<i>Modelo de Análisis</i>	<ul style="list-style-type: none"> ▪ Refina los casos de uso otorgándoles más detalle. ▪ Asigna la funcionalidad del sistema a un grupo de objetos que proporcionan el comportamiento.
<i>Modelo de Diseño</i>	<ul style="list-style-type: none"> ▪ Define la estructura estática del sistema en la forma de subsistemas, clases e intervalos. ▪ Refleja los casos de uso como colaboraciones entre los subsistemas e interfases.
<i>Modelo de Implementación</i>	<ul style="list-style-type: none"> ▪ Incluye los componentes, que representan el código fuente y la correspondencia de las clases con los componentes.
<i>Modelo de Despliegue</i>	<ul style="list-style-type: none"> ▪ Define la configuración de los nodos de procesamiento y las correspondencias de los componentes con esos nodos físicos.
<i>Modelo de Prueba</i>	<ul style="list-style-type: none"> ▪ Precisa los casos de prueba para validar los casos de uso.

Los desarrolladores comienzan capturando los requisitos del cliente en la forma de casos de uso en el modelo de casos de uso. Después, analizan y diseñan el sistema para cumplir con los casos de uso, creando en primer lugar un modelo de análisis, luego uno de diseño, otro de implementación, este último incluye todo el código, es decir, los componentes. Finalmente, los desarrolladores preparan un modelo de prueba que les permite verificar que el sistema proporciona la funcionalidad descrita en los casos de uso. El modelo de implementación es el más formal, mientras que el modelo de casos de uso es el menos formal, en el sentido de ser factible su compilación y ejecución. El modelo de casos de uso se describe mediante lenguaje natural. La figura 2.29, contiene la relación entre la serie de flujos de trabajo (etapas del desarrollo de software) y los modelos de PU.

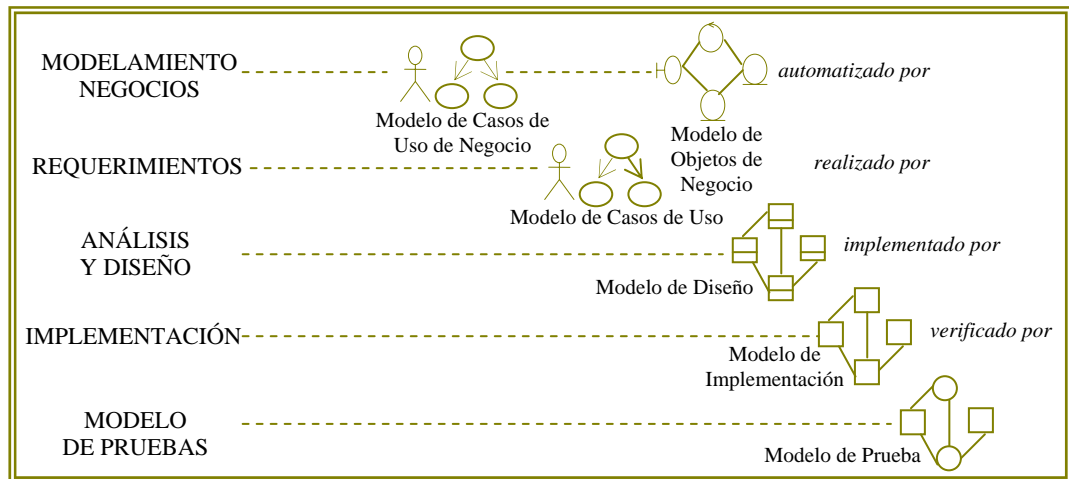


Figura 2.29. Los Flujos de Trabajo y los Modelos del PU basado en [Jacobson, 00]

✘ Fases de cada Ciclo

Cada ciclo se desarrolla a lo largo del tiempo. Este tiempo se divide en cuatro fases. A través de una secuencia de modelos, los implicados visualizan lo que está sucediendo en esas fases. Dentro de cada fase se descompone adicionalmente el trabajo en iteraciones con sus incrementos resultantes. Cada fase termina en un hito. Cada hito se determina por la disponibilidad de un conjunto de artefactos, es decir, ciertos modelos o documentos que han sido desarrollados hasta alcanzar un estado predefinido. Los hitos permiten a los desarrolladores controlar el progreso del trabajo según pasa por esos cuatro puntos clave.

Brevemente se detallan cada una de las fases, en relación a las etapas del ciclo de vida del sistema (flujos de trabajo):

- *Fase de Iniciación:* desarrolla una descripción del producto final a partir de una buena idea y se presenta el análisis del negocio para el producto. Se identifican y priorizan los riesgos más importantes, se planifica en detalle la fase de elaboración y se estima el proyecto de manera aproximada.
- *Fase de Elaboración:* se especifican en detalle la mayoría de los casos de uso del producto y se diseña la arquitectura del sistema. La arquitectura se expresa en forma de vistas de todos los modelos del sistema, los cuales juntos representan al sistema entero. Esto implica que hay vistas arquitectónicas del modelo de casos de uso, del modelo de análisis, del modelo de diseño, del modelo de implementación y del modelo de despliegue. El resultado de esta fase

es una línea base de la arquitectura. Al final de la fase, el director del proyecto está en disposición de planificar las actividades y estimar los recursos necesarios para terminar el proyecto.

- *Fase de Construcción:* se crea el producto, la línea base arquitectónica crece hasta convertirse en el sistema completo. La descripción evoluciona hasta convertirse en un producto preparado para ser entregado a la comunidad de usuarios. Al final de esta fase, el producto contiene todos los casos de uso que la dirección y el cliente han acordado para el desarrollo de esta versión. Esta versión puede que no esté completamente libre de defectos. Muchos de esos defectos se descubrirán y solucionarán durante la fase de transición.
- *Fase de Transición:* cubre el período durante el cual el producto se convierte en versión beta. En la versión beta un número reducido de usuarios con experiencia prueba el producto e informa de defectos y deficiencias. Los desarrolladores corrigen los problemas e incorporan algunas mejoras sugeridas en una versión general dirigida a la totalidad de la comunidad de usuarios. La fase de transición conlleva actividades como la fabricación, formación del cliente, el proporcionar una línea de ayuda y asistencia y la corrección de los defectos que se encuentren tras la entrega.

× UML

UML es un lenguaje estándar de modelado para el software, un lenguaje para la visualización, especificación, construcción y documentación de los artefactos de sistemas en los que el software juega un papel importante.

UML es sólo un lenguaje y por tanto es sólo una parte de un método de desarrollo de software. Es independiente del proceso, aunque para utilizarlo óptimamente se necesita un proceso dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental, como es el PU.

Permite a los desarrolladores visualizar los resultados de su trabajo en esquemas o diagramas estandarizados. UML les proporciona un vocabulario que incluye tres categorías: elementos, relaciones y diagramas.

Posee cuatro tipos de elementos: estructurales, de comportamiento, de agrupación y de anotación. Siete tipos principales de elementos estructurales: casos

de uso, clases, clases activas, interfaces, componentes, colaboraciones y nodos. Dos tipos de elementos de comportamiento: interacciones y máquinas de estado. Cuatro tipos de agrupaciones: paquetes, modelos, subsistemas y marcos de trabajo. Y un tipo principal de elementos de anotación: notas. Como se observa en la figura 2.30.

Dentro de la segunda categoría, encontramos tres tipos de relaciones: dependencia, asociación y generalización.

Y en la tercera categoría, la de diagramas, UML proporciona nueve tipos, diagramas de clases, de objetos, de componentes y de despliegue (corresponden a diagramas estáticos) y diagramas de casos de uso, de secuencia, de colaboración, de estados y de actividad (corresponden a diagramas dinámicos).

En cada modelo del PU, se utilizan diferentes diagramas, como se expone a continuación:

- Modelo de Casos de Uso: diagramas de casos de usos, diagrama de secuencia, diagrama de colaboración y diagrama de actividad.
- Modelo de Análisis y Diseño: diagrama de clases, diagrama de objetos, diagrama de secuencia, diagrama de colaboración y diagrama de actividad.
- Modelo de Despliegue: diagrama de despliegue, diagrama de secuencia y diagrama de colaboración.
- Modelo de Implementación: diagrama de componentes, diagrama de secuencia y diagrama de colaboración.
- Modelo de Pruebas: todos los diagramas.

Para cerrar el apartado, basta remitirse literalmente a la definición encontrada en el glosario de Jacobson [Jacobson, 00] sobre UML.

Lenguaje Unificado de Modelado: lenguaje estándar para el modelado de software, lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema con gran cantidad de software. Lenguaje usado por PU. Lenguaje que permite a los desarrolladores, visualizar el producto de su trabajo (artefactos) en esquemas o diagramas estandarizados.

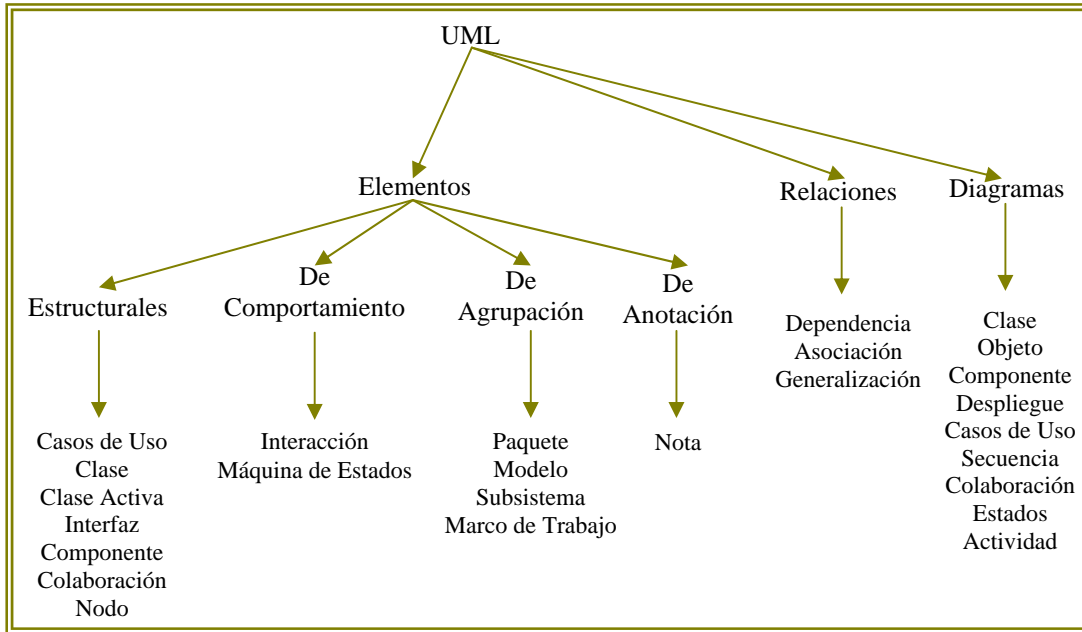


Figura 2.30. Elementos Conceptuales de UML

II.2.7. FASE DE ANÁLISIS

* Descripción

En la captura de requisitos, se utiliza el lenguaje del cliente para llegar a un acuerdo de lo que se debe hacer, por ende, *es probable que aún queden aspectos sin resolver relativos a los requisitos del proceso, debido a que el cliente emplea un lenguaje intuitivo y a la vez impreciso*. Los mismos requisitos capturados constituyen la entrada a la etapa de análisis², en la cual la vista se centra en el sistema desde la perspectiva de los desarrolladores y no en el problema que afecta al cliente.

Durante la fase de análisis del PU, se estudian los requisitos que se describieron en la etapa de requisitos, refinándolos y estructurándolos. El objetivo de hacerlo es conseguir una comprensión más precisa de ellos y una descripción de los mismos que sea fácil de mantener y que ayude a estructurar el sistema entero.

² Moreno en [Moreno, 97] hace hincapié en que no debe quedar ninguna información, sin ser considerada para conformar un producto software conciso y coherente conforme al contexto modelizado. Esto lo logra en su trabajo de tesis doctoral.

✘ Modelo de Análisis

El lenguaje que se utiliza en el análisis se basa en un modelo de objetos conceptual, que se denomina *modelo de análisis*. En el modelo de análisis existen dos conjuntos de diagramas de modelado: los diagramas de clases y los diagramas de interacción (colaboración), donde se plasman los aspectos estáticos y dinámicos, respectivamente. El modelo de análisis ofrece un mayor poder expresivo y una mayor formalización.

Características principales del Modelo de Análisis:

- Descrito por el lenguaje del desarrollador.
- Integra la vista interna del sistema.
- Estructurado por clases y paquetes estereotipados, que proporcionan la estructura a la vista interna.
- Utilizado fundamentalmente por los desarrolladores para comprender cómo debería darse forma al sistema, es decir, cómo debería ser diseñado e implementado.
- No debería contener redundancias, inconsistencias, etc..., entre requisitos.
- Esboza cómo llevar a cabo la funcionalidad dentro del sistema, incluida la funcionalidad significativa para la arquitectura; sirve como una primera aproximación al diseño.
- Define realizaciones de casos de uso y cada una de ellas representa el análisis de un caso de uso del modelo de casos de uso.

Con el fin de sintetizar sus bondades, analizar los requisitos en la forma de un modelo de análisis es importante por varios motivos:

- Ofrece una especificación más precisa de los requisitos que la que tenemos como resultado de la captura de requisitos, incluyendo el modelo de casos de uso.
- Describe utilizando el lenguaje de los desarrolladores y puede por tanto introducir un mayor formalismo y ser utilizado para razonar sobre los funcionamientos internos del sistema.
- Estructura los requisitos de un modo que facilita su comprensión, su preparación, su modificación, y en general, su mantenimiento.

- Puede considerarse como una primera aproximación al modelo del diseño y es por tanto una entrada fundamental cuando se da forma al sistema en el diseño y la implementación.

✘ Clase de Análisis

Dentro del modelo de análisis, los casos de usos se describen mediante clases de análisis y sus objetos. Las clases de análisis representan abstracciones de clases y posiblemente de subsistemas de diseño del sistema. Esto se representa mediante colaboraciones dentro del modelo de análisis que se llaman realizaciones de casos de uso-análisis.

Esta abstracción posee las siguientes características:

- Se centra en el tratamiento de los requisitos funcionales y pospone los no funcionales, una desventaja importante en esta etapa del PU, denominándolos requisitos especiales, hasta llegar a las actividades de diseño e implementación.
- Esto hace que una clase de análisis sea más evidente en el contexto del dominio del problema, más conceptual, a menudo de mayor granularidad que sus contrapartidas de diseño e implementación.
- Define u ofrece una interfaz en términos de operaciones y de sus firmas. En cambio, su comportamiento se define mediante responsabilidades en un más alto nivel y menos formal. Una responsabilidad es una descripción textual de un conjunto cohesivo del comportamiento de una clase.
- Precisa atributos, aunque esos atributos también son de un nivel bastante alto. Los tipos de atributos son conceptuales y reconocibles en el dominio del problema, mientras que los tipos de los atributos en las clases de diseño y la implementación suelen ser tipos de lenguajes de programación. Además, los atributos identificados durante el análisis con frecuencia pasan a ser clases en el diseño y la implementación.
- Las clases de análisis siempre encajan en uno de tres *estereotipos* básicos: de interfaz, de entidad o de control.

✘ Estereotipo

Las clases de análisis se encuentran clasificadas por su modo de aportar recursos al sistema. Cada uno de estos tipos de clases representa un estereotipo de una clase, esto significa que, según su estereotipo designado, una clase actuará de tal o cual manera y de manera similar que las clases de su mismo estereotipo. Cada estereotipo implica una semántica específica, lo cuál constituye un método potente y consistente de identificar y describir las clases de análisis y contribuye a la creación de un modelo de objetos y una arquitectura robustos. Estos tres estereotipos están estandarizados en UML y se utilizan para ayudar a los desarrolladores a distinguir el ámbito de las diferentes clases. Cada estereotipo tiene su propio símbolo como lo vemos en la tabla 2.8.

✘ Realización de caso de uso-análisis

Una realización caso de uso-análisis es una colaboración dentro del modelo de análisis que describe cómo se lleva a cabo y se ejecuta un caso de uso determinado en término de las clases del análisis y de sus objetos. Una realización de caso de uso proporciona por tanto una relación directa hacia un caso de uso concreto del modelo de casos de uso.

Una realización de casos de uso posee una descripción textual del flujo de casos, diagramas de clases que muestran sus clases del análisis participantes y diagramas de interacción que muestran la realización de un flujo o escenario particular del caso de uso en términos de interacciones de objetos del análisis. Además, debido a que describimos una realización de casos de uso en términos de clases de análisis y de sus objetos, se centra de manera natural en los requisitos funcionales. Por tanto, al igual que las propias clases del análisis, puede posponer el tratamiento de los requisitos no funcionales hasta las actividades subsiguientes de diseño e implementación, llamándoles requisitos especiales en la realización.

En las tablas 2.9 y 2.10, se describe cada uno de los diagramas del modelo de análisis y sus destacadas diferencias, respectivamente.

Tabla 2.9. Diagramas del Modelo de Análisis

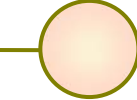


Nombre	Símbolo	Descripción del Estereotipo
Clases de Interfaz		<p>Se utilizan para modelar la interfaz entre el sistema y sus actores (es decir, entre usuarios y sistemas externos). Esta interacción implica recibir (y presentar) información y peticiones de (y hacia) los usuarios y los sistemas externos. Modelan las partes del sistema que dependen de sus actores, lo cual implica que clarifican y reúnen los requisitos en los límites del sistema. Representan, a menudo, abstracciones de ventanas, formularios, paneles, interfaces de comunicación, interfaces de impresoras, sensores y terminales. Describen lo que se obtiene con la interacción (es decir, la información y las peticiones que se intercambian entre el sistema y sus actores). No es necesario que describan cómo se ejecuta físicamente la interacción, ya que esto se considera en las actividades de diseño e implementación.</p>
Clases de Entidad		<p>Se utilizan para modelar información que posee una vida larga y que es a menudo persistente. La clase de entidad modela la información y el comportamiento asociado de algún fenómeno o concepto, como una persona, un objeto o un suceso del mundo real. Estas clases reflejan la información de un modo que beneficia a los desarrolladores al diseñar e implementar el sistema, incluyendo su soporte de persistencia. Por otro lado, las clases de entidad del negocio (o clases del dominio) describen el contexto del sistema y pueden incluir información que el sistema no maneja en absoluto. Un objeto de entidad no ha de ser necesariamente pasivo y puede tener en ocasiones un comportamiento complejo relativo a la información que representa. Los objetos de entidad aíslan los cambios en la información que representan. Suelen mostrar una estructura de datos lógica y contribuyen a comprender qué información depende del sistema.</p>
Clases de Control		<p>Representan coordinación, secuencia, transacciones y control de otros objetos y se usan con frecuencia para encapsular el control de un caso de uso en concreto. Las clases de control se utilizan para representar derivaciones y cálculos complejos, como la lógica del negocio, que no puede asociarse con ninguna información concreta, de larga duración, almacenada por el sistema. Los aspectos dinámicos del sistema se modelan con estas clases, debido a que ellas manejan y coordinan las acciones y flujos de control principales y delegan trabajo a otros objetos (es decir, objetos de interfaz y de entidad). En resumen, las clases de control encapsulan, y por lo tanto aíslan, los cambios del control, la coordinación, la secuencia, las transacciones y a veces la lógica del negocio que implica a varios otros objetos.</p>

Tabla 2.9. Diagramas del Modelo de Análisis (Continuación)

Tabla 2.9. Diagramas del Modelo de Análisis (Continuación)

Diagrama	Definición
<p>Diagrama de Clases</p>	<p>Una clase de análisis y sus objetos normalmente participan en varias realizaciones de casos de uso y algunas de las responsabilidades, atributos y asociaciones de una case concreta suelen ser sólo relevantes para una única realización de casos de uso. Por tanto, es importante durante el análisis coordinar todos los requisitos sobre una clase de objetos que pueden tener diferentes casos de uso. Para entender mejor como se relacionan las clases intervinientes en la realización de un caso de uso es conveniente agruparlas en un diagrama mostrando las clases participantes, los actores que interactúan generando actividades. Para hacerlo, adjuntamos <i>diagramas de clases</i> a las realizaciones de casos de uso, mostrando sus clases participantes y sus relaciones. Las relaciones que existen entre ellos indican la existencia de una posible interacción tanto entre un actor y una clase como también entre clases. La figura 2.31 muestra un ejemplo del diagrama de clases.</p>
<p>Diagrama de Interacción</p>	<p>La secuencia de acciones en un caso de usos comienza cuando un actor invoca el caso de uso mediante el envió de algún tipo de mensaje al sistema. Si se considera el interior del sistema como un objeto de interfaz recibirá este mensaje del actor. El objeto de interfaz enviará a su vez un mensaje a algún otro objeto y de esta forma los objetos implicados interactuarán para llevar a cabo los casos de uso. En el análisis se prefiere mostrar la interacción con <i>diagramas de colaboración</i> ya que el objetivo fundamental es identificar requisitos y responsabilidades sobre los objeto y no identificar secuencias de interacción detalladas y ordenadas cronológicamente. En los diagramas de colaboración, se muestran las interacciones entre objetos creando enlaces entre ellos y añadiendo mensajes a esos enlaces. El nombre de un mensaje debería denotar el propósito del objeto invocante en la interacción con el objeto invocado. La figura 2.32 muestra un ejemplo del diagrama de colaboración.</p>

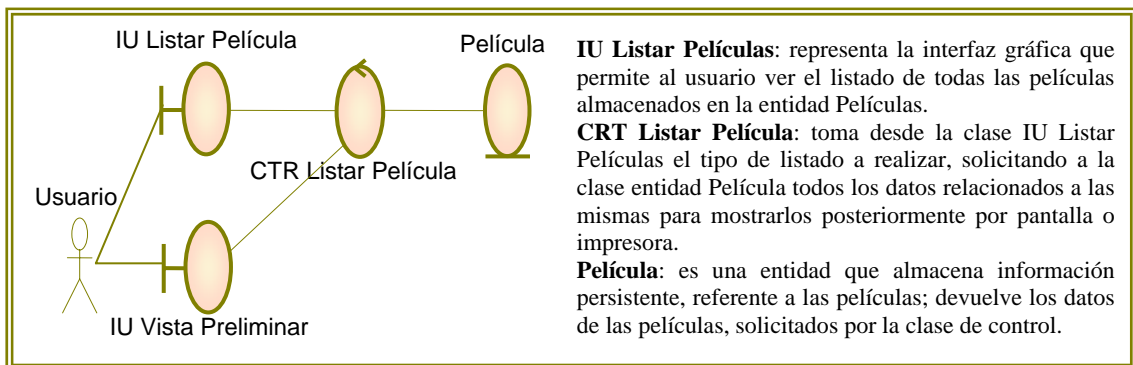


Figura 2.31. Ejemplo del Diagrama de Clases del Modelo de Análisis del PU

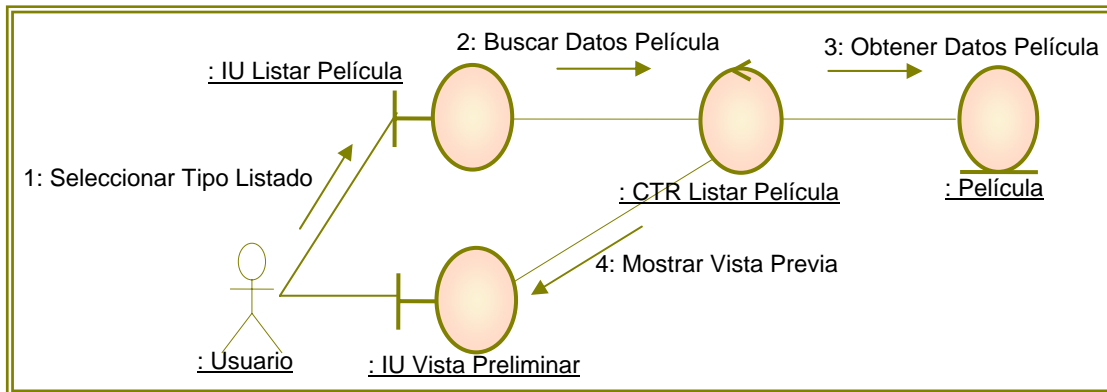


Figura 2.32. Ejemplo del Diagrama de Colaboración del Modelo de Análisis del PU

Tabla 2.10. Diferencia entre los Diagramas del Modelo de Análisis

DIFERENCIAS	
Diagrama de Clases	Diagrama de Colaboración
<ol style="list-style-type: none"> 1) Modela el esqueleto que seguirán los objetos al realizar cada caso de uso. 2) En un diagrama de clases se habla únicamente de clases. 3) Representa información estática del sistema. 	<ol style="list-style-type: none"> 1) Modela sólo las clases necesarias para la realización de los distintos escenarios identificados dentro del caso de uso. 2) En un diagrama de colaboración se habla en término de los objetos, éstos son instancias de las clases anteriores. 3) Representa información dinámica del sistema.

✘ Flujo de Suceso-Análisis

Los diagramas (especialmente los diagramas de colaboración) de una realización de caso de uso pueden ser difíciles de leer por sí mismos, de modo que puede ser útil un texto adicional que los explique. Este texto debería escribirse en término de objetos, particularmente objetos de control que interactúan para llevar a cabo el caso de uso. El texto no debería mencionar ninguno de los atributos, responsabilidades y asociaciones del objeto, debido a que cambian con bastante frecuencia y sería difícil mantenerlos.

Observamos el siguiente texto, que explica los ejemplos de las figuras 2.31 y 2.32: “El usuario solicita al sistema listar las películas alquiladas, seleccionando el tipo de listado que quiere realizar (1). Una vez que la opción es registrada de la

clase interfaz IU Listar Películas se realiza la búsqueda de las películas (2) que están alquiladas a la clase CTR Listar Película, se obtienen los datos de las películas (3) de la clase entidad Película y se realiza una vista previa (4) por medio de la clase interfaz IU Vista Preliminar.”

✘ Paquetes del Análisis

Los paquetes del análisis proporcionan un medio para organizar los artefactos del modelo de análisis en piezas manejables. Un paquete de análisis puede constar de clases de análisis, de realizaciones de casos de uso y de otros paquetes del análisis.

Tienen las siguientes características:

- Los paquetes del análisis pueden representar una separación de intereses de análisis.
- Deberían crearse en base a los requisitos funcionales y en el dominio del problema (la aplicación o el negocio) y deberían ser reconocibles por las personas con conocimiento del dominio. Los paquetes del análisis no deberían basarse en requisitos no funcionales o en el dominio de la solución.
- Probablemente se convertirán en subsistemas en las dos capas de aplicación superiores del modelo de diseño o se distribuirán entre ellos. En algunos casos, podrían incluso reflejar una capa completa de primer nivel en el modelo de diseño.

✘ Vista del Modelo de Análisis

La descripción de la arquitectura contiene una vista de la arquitectura del modelo de análisis, que muestra sus artefactos significativos para la arquitectura.

Los siguientes artefactos del modelo de análisis normalmente se consideran significativos para la arquitectura:

- La descomposición del modelo de análisis en paquetes de análisis y sus dependencias suele tener su efecto en los subsistemas de las capas superiores durante el diseño y la implementación y es por tanto relevante para la arquitectura en general.
- Las clases fundamentales del análisis como las clases de entidad que encapsulan un fenómeno importante del dominio del problema; las clases de interfaz que

encapsulan interfaces de comunicación importantes y mecanismos de interfaz de usuarios; las clases de control que encapsulan importantes secuencias con una amplia cobertura (es decir, aquellas que coordinan realizaciones de casos de uso significativos); y clases del análisis que son generales, centrales y que tienen muchas relaciones con otras clases del análisis. Suele ser suficiente con considerar significativa para la arquitectura una clase abstracta pero no sus subclasses.

- Realizaciones de casos de uso que describen cierta funcionalidad importante y crítica; que implica muchas clases del análisis y por tanto tienen una cobertura amplia, posiblemente a lo largo de varios paquetes de análisis; o que se centran en un caso de uso importante que debe ser desarrollado al principio en el ciclo de vida del software y por tanto es probable que se encuentre en la vista de la arquitectura del modelo de casos de uso.

II.2.8. CONCLUSIONES SOBRE EL PROCESO UNIFICADO

En base a todo lo aprendido sobre este proceso de desarrollo de software, se concluye lo siguiente.

PU dista mucho de ser solamente un modelo de análisis con sus correspondientes diagramas de clases y colaboración pertenecientes a la fase de Análisis. Este proceso constituye el avance más reciente dentro de los paradigmas de la OO y surge en los últimos años como la aproximación, dentro de la perspectiva orientada a objetos, más apropiada para el logro satisfactorio del desarrollo de software.

Proporciona un procedimiento de modelización **definido**, lo que significa que especifica una secuencia de pasos específicos para llevar a cabo el proceso de Análisis. También facilita herramientas usadas para automatizar las tareas puntualizadas en el proceso mediante una colección de actividades y roles para realizar el proceso de modelización del proceso software. Esta modelización se realiza a través de *heurísticas intuitivas, correctas y completas*. Hablamos de heurísticas, al referirnos a que el método proporciona un conjunto de guías que surgen de la mera intuición del analista; correctas, al hacer mención a que dichas guías son aplicables sin excepciones; y completas, haciendo alusión a que si bien existen las mencionadas guías, éstas no justifican formalmente las asignaciones realizadas para identificar los componentes de los modelos orientados a objetos.

CAPÍTULO III. FORMALIZACIÓN DEL MODELO DE PROCESO MULTINIVEL

*“No descubriremos nuevas tierras hasta que no nos atrevamos
a perder de vista la orilla durante algún tiempo”*
André Gide

III.1 INTRODUCCIÓN

Según lo referido en la *hipótesis 1*, con el fin de satisfacer los objetivos formulados y a partir de la especificación textual en lenguaje natural, se aplican las nueve etapas del método MATE a la descripción informal del Nivel de Ingeniería del modelo multinivel; permitiendo, de esta forma, obtener su modelización conceptual. Dicha modelización servirá para validar la eficacia del método, mediante el análisis formal del modelo de proceso software considerado.

La especificación textual citada describe los procesos y procedimientos involucrados en el Nivel de Ingeniería del MPM. En este capítulo, se demuestra cómo se obtienen los modelos conceptuales que representan el mencionado nivel. Para ello, se aplican las etapas del método MATE [Acuña, 00b], explicadas desde el apartado II.2.5 en adelante.

Por motivos de espacio en la construcción del Modelo de Objetos, se utilizan abreviaciones, las que figuran en el Anexo B.

III.1.1. EXTRACCIÓN DE LA INFORMACIÓN ESENCIAL

En esta etapa se descarta, de la descripción textual del Nivel de Ingeniería del MPM (sección II.1.2), toda la información que no sea útil para el Análisis, es decir, toda información que no constituya un requisito funcional; y se hace explícita la información implícita necesaria para entender los requisitos funcionales de la descripción textual.

En este caso, se obtienen como resultados los siguientes Requisitos relevantes:

- **Proceso Organización y Proceso Ingeniería**

El Proceso Organización (PO) y el Proceso Ingeniería (PI) forman parte del Nivel de Ingeniería del MPM.

La modelización de la organización en el Proceso Organización se identifica con el Procedimiento Cultural. La modelización del proceso en el Proceso Ingeniería se identifica con el Procedimiento de Modelo de Proceso.

El Proceso Organización tiene los siguientes documentos de entrada:

Especificación del Universo Ecológico (EUE), Registros Históricos de Modelos de Organizaciones y Modelo de Procesos Organizacionales (RHPO) y Tecnologías Transferidas (TT). El Proceso Organización genera el Modelo de Organización Actual (MOA), el Modelo de Organización Futura (MOF) y el Modelo de Proceso Organizacional (MPO).

Las fases Investigación Exploratoria (IE), Investigación Cualitativa (IC), Investigación Cuantitativa (ICU) y Análisis y Evaluación (AE) forman parte del Procedimiento Cultural. La fase Investigación Exploratoria contiene las etapas Extracción de Información Esencial (EIE) y Estudio de Dominio Organizacional (EDO). Las Características Relevantes (CR) y el Modelo de Problemas (MP) son resultados de las tareas anteriores, respectivamente. La fase Investigación Cualitativa contiene las etapas Formulación de los Procesos Relevantes (FPR) y Determinación de las Capacidades de las Personas (DCP). Las etapas antes mencionadas suministran como resultado los Modelos de Procesos (MPROC) y Modelo de Personas Validado (MPERS), respectivamente. La fase Investigación Cuantitativa tiene las tareas Construcción de los Modelos Conceptuales (CMC) y Determinación de la Cultura y la Estrategia de Mejora de la Organización (DCEMO). Los Modelos de Actividades y Roles (MAR), y el Modelo Cultural Validado (MC) son salidas respectivas de las tareas mencionadas. La fase Análisis y Evaluación posee las etapas Definición de Procesos Factibles Culturalmente (DPFC) y Validación del Modelo de Proceso Organizacional (VPMO). Las etapas Definición de Procesos Factibles Culturalmente y Validación del Modelo de Proceso Organizacional suministran respectivamente el Modelo de Proceso Organizacional (MPORG) y el Modelo de Proceso Organizacional Validado (MPOV).

El Proceso Ingeniería se inicia en base a los Modelos de Organización Actual y Organización Futura y al de Proceso Organizacional. El Proceso Ingeniería tiene los siguientes documentos de entrada: Registros Históricos (RH) de los Modelos de Proceso Software Generales (MPSG) y los Modelos del Ciclo de Vida del Software (MCVS), Modelo del Dominio (MD) e Innovaciones Metodológicas y Técnicas de la Ingeniería del Software y la Ingeniería del Conocimiento (IMTI). Como documentos de salida contiene: los Modelos de Proceso Software Generales, los Modelos del Ciclo de Vida del Software,

Fines, Políticas y Decisiones Estratégicas (FPDE) y Tecnologías (T).

Las fases Exploración (E), Mejora (M), Análisis (A), Construcción (C) y Validación Verificación (VV) forman parte del Procedimiento de Modelo de Proceso.

El Proceso Organización (PO) y el Proceso Ingeniería (PI) son realizados por profesionales de un equipo principal y de un equipo de soporte, para cada uno de los procesos. En ambos equipos trabajan profesionales de distintas disciplinas. Cada profesional de cada equipo se identifica por un rol, un nombre y un código. Un nombre de profesional puede corresponder a varios profesionales. Un código corresponde a un profesional.

- **Procedimiento Cultural**

En la etapa I.1, Extracción de Información Esencial, la información de la Especificación del Universo Ecológico junto con la información implícita constituyen las Características Relevantes.

En la etapa I.2, Estudio de Dominio Organizacional, en base a las Características Relevantes, se estructuran los elementos que integran situaciones problemáticas en el dominio de la organización conformando el Modelo de Problemas.

En la etapa II.1, Formulación de Procesos Relevantes, a partir del Modelo de Problemas se identifican los procesos relevantes y se expresa una solución a través de una definición radical, para luego elaborar el Modelo de Procesos.

En la etapa II.2, Determinación de las Capacidades de las Personas, el Modelo de Procesos se utiliza para la construcción del Modelo de Personas Preliminar. El Modelo de Personas Preliminar válido da lugar al Modelo de Personas; en caso contrario el proceso se repetiría desde la Educción del Universo Ecológico, o bien, desde cualquier etapa previa.

En la etapa III.1, Construcción de los Modelos Conceptuales, a partir del Modelo de Procesos se construye el Modelo de Actividades y Roles.

En la etapa III.2, Determinación de la Cultura y la Estrategia de la Mejora de la Organización, en base a los Modelos de Personas y de Actividades y Roles se proporciona el Modelo Cultural Preliminar. El Modelo Cultural Preliminar válido da lugar al Modelo Cultural, caso contrario el proceso se repetiría

desde la Educción del Universo Ecológico, o bien, desde cualquier etapa previa.

En la etapa IV.1, Definición de Procesos Factibles Culturalmente, desde el Modelo Cultural se determinan los cambios y procesos que originan el Modelo de Proceso Organizacional.

En la etapa IV.2, Validación del Modelo de Proceso Organizacional, se valida el Modelo de Proceso Organizacional, generándose el Modelo de Proceso Organizacional Validado.

- **Procedimiento de Modelo de Proceso**

En la etapa I, Exploración, a partir del análisis de los modelos de Organización Futura, de Proceso Organizacional y de los Registros Históricos de Procesos Software generales y específicos se obtiene el Modelo de Requisitos.

En la etapa II, Mejora, sobre la base del análisis de los modelos de Organización Actual y Futura y el Modelo de Proceso Organizacional se diagnostica el estado de los Modelos de Proceso Software Generales de la organización. Una vez realizado el diagnóstico de los Modelos de Proceso Software Generales, se determina el Modelo de Mejora Organizacional.

En la etapa III, Análisis, se estudian el Modelo de Proceso Organizacional y los conocimientos de los expertos del proceso software, se comprenden las relaciones relevantes de los elementos del Modelo de Proceso y se construyen los Modelos Conceptuales.

En la etapa IV, Construcción, desde los Modelos Conceptuales generados, se construyen Modelos Formales.

En la etapa V, Validación–Verificación, se comprueba que los Modelos Formales cumplan con los objetivos definidos; en caso contrario se repite el proceso desde las etapas previas.

III.1.2. IDENTIFICACIÓN DE SINÓNIMOS Y POLISEMAS

En esta etapa se eliminan ambigüedades contenidas en las especificaciones textuales, se identifican palabras sinónimos y palabras polisémicas. Como resultado

se obtiene un Diccionario de Sinónimos y Polisemas, con el formato descrito en la tabla 2.3 de la sección II.2.5.2. Este diccionario se muestra en la tabla 3.1.

III.1.3. SEPARACIÓN DE LA INFORMACIÓN ESTÁTICA Y DINÁMICA

De las especificaciones útiles obtenidas del proceso se separa por una parte la información estática y por otra parte la información dinámica de la aplicación. La primera información contiene propiedades estructurales de la información que ha de manejar la aplicación y constituyen los Requisitos Estáticos y la segunda especifica interacciones y eventos que afectan a la información descrita en la primera y constituyen los Requisitos Dinámicos de la aplicación.

Los *Requisitos Estáticos* obtenidos en esta etapa coinciden *casualmente* con la especificación del Proceso Organización y el Proceso Ingeniería detallada en la Etapa 1, Extracción de Información Esencial.

Los *Requisitos Dinámicos* del proceso son los siguientes:

- **Procedimiento Cultural**

La información de la parte dinámica del Procedimiento Cultural coincide *casualmente* con los resultados obtenidos en la Etapa 1, Extracción de Información Esencial.

- **Procedimiento de Modelo de Proceso**

La información de la parte dinámica del Procedimiento Modelo de Proceso coincide *casualmente* con los resultados obtenidos en la Etapa 1, Extracción de la Información Esencial.

III.1.4. ESTRUCTURACIÓN DE LOS REQUISITOS ESTÁTICOS

La información de los requisitos estáticos del proceso se estructura mediante la gramática del Lenguaje de Utilidad Estático de la base FORMOL, visto en la sección II.2.4.1 y da lugar a la obtención del Lenguaje de Utilidad Estático Preliminar.

Tabla 3.1. Diccionario de Sinónimos y Polisemas del MPM

Nombre: Análisis	<i>Descripción:</i> Examen minucioso. <i>Palabras Polisémicas:</i> Análisis; Análisis (y Evaluación).
Nombre: Construcción	<i>Descripción:</i> Elaboración. <i>Palabras Polisémicas:</i> Construcción; Construcción (de los Modelos Conceptuales).
Nombre: Determinación	<i>Descripción:</i> Establecer un criterio o concepto. <i>Palabras Polisémicas:</i> Determinación (de la Cultura y Estrategia de Mejora de la Organización); Determinación (de las Capacidades de las Personas).
Nombre: Equipo	<i>Descripción:</i> Conjunto de profesionales de diferentes áreas. <i>Palabras Polisémicas:</i> Equipo (Principal PO); Equipo (Soporte PO); Equipo (Principal PI); Equipo (Soporte PI).
Nombre: Investigación	<i>Descripción:</i> Indagar desde diferentes puntos de vista los procesos. <i>Palabras Polisémicas:</i> Investigación (Cualitativa); Investigación (Cuantitativa); Investigación (Exploratoria).
Nombre: Modelo	<i>Descripción:</i> Representación de la realidad de la organización y del proceso. <i>Sinónimos:</i> Modelos Productos <i>Palabras Polisémicas:</i> Modelo (de la Organización Futura); Modelo (de la Organización Actual) Modelo (de Proceso Organizacional); Modelo (de Problemas); Modelo (de Procesos); Modelo (de Personas Preliminar); Modelo (de Personas) Modelo (de Actividades y Roles); Modelo (Cultural Preliminar); Modelo (Cultural); Modelo (de Proceso Organizacional Validado); Modelo (de Mejora Organizacional); Modelos (Conceptuales); Modelo (del Dominio); Modelo (de Proceso Software Generales); Modelo (del Ciclo de Vida del Software); Modelo (de Requisitos); Modelo (de Mejora); Modelo (de Proceso); Modelo (Formales).
Nombre: Procedimiento	<i>Descripción:</i> Método práctico para llevar a cabo un objetivo. <i>Palabras Polisémicas:</i> Procedimiento (Cultural); Procedimiento (de Modelo de Proceso).
Nombre: Proceso	<i>Descripción:</i> Conjunto de actividades, productos, personas y organización. <i>Palabras Polisémicas:</i> Proceso (Organización); Proceso (Ingeniería); Proceso (Software).
Nombre: Registros Históricos	<i>Descripción:</i> Escritos que sirven para documentar los modelos realizados en el pasado. <i>Palabras Polisémicas:</i> Registros Históricos (de los Modelos de Proceso Software Generales y los Modelos del Ciclo de Vida del Software); Registros Históricos (de Modelo de Organizaciones y Modelo de Proceso Organizacionales).
Nombre: Tecnologías	<i>Descripción:</i> Documentos de entrada-salida a los procesos principales. <i>Palabras Polisémicas:</i> Tecnologías; Tecnologías (Transferidas).
Nombre: Validación	<i>Descripción:</i> Pruebas de aceptación, validación y verificación de los modelos. <i>Palabras Polisémicas:</i> Validación (del Modelo de Proceso Organizacional); Validación (Verificación).

Las estructuras gramaticales obtenidas se muestran a continuación. Para cada oración, se realiza el análisis correspondiente. En estas estructuras, las palabras en *cursiva* representan el *sintagma nominal*, las palabras en **negrita** el **verbo** empleado y las palabras subrayadas representan el complemento de la oración.

Se comienza con el análisis de las estructuras correspondientes a los requisitos estáticos del Nivel de Ingeniería del MPM detectados:

- 1) El Proceso Organización y el Proceso Ingeniería forman parte del Nivel de Ingeniería del MPM.

El Proceso Organización y el Proceso Ingeniería **forman**
L_{3,2} *sintagma nominal* **verbo composición-contenido**
parte del Nivel de Ingeniería del MPM.
complemento

- 2) La modelización de la organización en Proceso Organización se identifica con el Procedimiento Cultural.

La modelización de la organización en Proceso Organización **se identifica**
L₄ *sintagma nominal* **verbo**
con el Procedimiento Cultural.
complemento

- 3) La modelización del proceso en el Proceso Ingeniería se identifica con el Procedimiento de Modelo de Proceso.

La modelización del proceso en el Proceso Ingeniería **se identifica**
L₄ *sintagma nominal* **verbo**
con el Procedimiento de Modelo de Proceso.
complemento

- 4) El Proceso Organización tiene los siguientes documentos de entrada: EUE, RHPO y TT.

El Proceso Organización **tiene**
L_{3,1} *sintagma nominal* **verbo composición-componente**
los siguientes documentos de entrada: EUE, RHPO y TT.
complemento

- 5) El Proceso Organización genera los siguientes documentos de salida: MOA y MOF y el MPO.

El Proceso Organización

genera

L₂ *sintagma nominal*

verbo_general

los siguientes documentos de salida: MOA y MOF y el MPO.

complemento

6) El proceso Organización es realizado por profesionales de un equipo principal y de un equipo de soporte.

El proceso Organización

es realizado

L₂ *sintagma nominal*

verbo_general

por profesionales de un equipo principal y de un equipo de soporte.

complemento

7) En ambos equipos trabajan profesionales de distintas disciplinas.

En ambos equipos

trabajan

profesionales de distintas disciplinas.

L₂ *sintagma nominal*

verbo_general

complemento

8) Cada profesional de cada equipo se identifica por un rol, nombre y un código.

Cada profesional de cada equipo

se identifica

por un rol, nombre y un código.

L₄ *sintagma nominal*

verbo

complemento

9) Un nombre de profesional puede corresponder a varios profesionales.

Un nombre de profesional

puede corresponder

a varios profesionales.

L₂ *sintagma nominal*

verbo_general

complemento

10) Un código corresponde a un profesional.

Un código

corresponde

a un profesional.

L₂ *sintagma nominal*

verbo_general

complemento

11) Las Fases IE, IC, ICU y AE forman parte del Procedimiento Cultural.

Las Fases IE, IC, ICU y AE

forman parte

del Procedimiento Cultural.

L_{3,2} *sintagma nominal*

verbo composición-contenido

complemento

12) La fase Investigación Exploratoria contiene las etapas EIE y EDO.

La fase Investigación Exploratoria

contiene

las etapas EIE y EDO.

L_{3,1} *sintagma nominal*

verbo composición-componente

complemento

13) La etapa EIE suministra como resultado las CR.

La etapa EIE **suministra** como resultado las CR.
L₂ sintagma nominal **verbo_general** complemento

14) El MP es resultado de la etapa EDO.

El MP **es resultado** de la etapa EDO.
L₂ sintagma nominal **verbo_general** complemento

15) La fase Investigación Cualitativa contiene las etapas FPR y DCP.

La fase Investigación Cualitativa **contiene** las etapas FPR y DCP.
L_{3,1} sintagma nominal **verbo composición-componente** complemento

16) La etapa FPR suministra como resultado el MPROC.

La etapa FPR **suministra** como resultado el MPROC.
L₂ sintagma nominal **verbo_general** complemento

17) El MPERS validado es resultado de la etapa DCP.

El MPERS validado **es resultado** de la etapa DCP.
L₂ sintagma nominal **verbo_general** complemento

18) La fase Investigación Cuantitativa tiene las tareas CMC y DCEMO.

La fase Investigación Cuantitativa **tiene** las tareas CMC y DCEMO.
L_{3,1} sintagma nominal **verbo composición-componente** complemento

19) La etapa CMC suministra como resultado el MAR.

La etapa CMC **suministra** como resultado el MAR.
L₂ sintagma nominal **verbo_general** complemento

20) El MC validado es resultado de la etapa DCEMO.

El MC validado **es resultado** de la etapa DCEMO.
L₂ sintagma nominal **verbo_general** complemento

21) La fase Análisis y Evaluación posee las etapas DPFC y VPMO.

La fase Análisis y Evaluación **posee** las etapas DPFC y VPMO.
L_{3,1} sintagma nominal **verbo composición-componente** complemento

22) La etapa DPFC suministra como resultado el MPORG.

La etapa DPFC **suministra** como resultado el MPORG.

L₂ sintagma nominal **verbo_general** complemento

23) El MPOV es resultado de la etapa VMPO.

El MPOV **es resultado** de la etapa VMPO.

L₂ sintagma nominal **verbo_general** complemento

24) El Proceso Ingeniería se inicia en base a los MOA, MOF y al MPO.

El Proceso Ingeniería **se inicia** en base a los MOA, MOF y al MPO.

L₂ sintagma nominal **verbo_general** complemento

25) El Proceso Ingeniería tiene los siguientes documentos de entrada: RH, MD e IMTI.

El Proceso Ingeniería **tiene**

L_{3,1} sintagma nominal **verbo composición-componente**

los siguientes documentos de entrada: RH, MD e IMTI.

complemento

26) Como documentos de salida contiene MSPG, MCVS, FPDE y T.

Como documentos de salida **contiene** MSPG, MCVS, FPDE y T.

L_{3,1} sintagma nominal **verbo composición-componente** complemento

27) Las Fases E, M, A, C y VV forman parte del Procedimiento de Modelo de Proceso.

Las Fases E, M, A, C y VV **forman parte**

L_{3,2} sintagma nominal **verbo composición-contenido**

del Procedimiento de Modelo de Proceso.

complemento

28) El proceso Ingeniería es realizado por profesionales de un equipo principal y de un equipo de soporte.

El proceso Ingeniería **es realizado**

L₂ sintagma nominal **verbo_general**

por profesionales de un equipo principal y de un equipo de soporte.

complemento

29) En ambos equipos trabajan profesionales de distintas disciplinas.

En ambos equipos **trabajan** profesionales de distintas disciplinas.
 L₂ *sintagma nominal* **verbo_general** complemento

30) Cada profesional de cada equipo se identifica por un rol, un nombre y un código.

Cada profesional de cada equipo **se identifica** por un rol, un nombre y un código.
 L₄ *sintagma nominal* **verbo** complemento

31) Un nombre de profesional puede corresponder a varios profesionales.

Un nombre de profesional **puede corresponder** a varios profesionales.
 L₂ *sintagma nominal* **verbo_general** complemento

32) Un código corresponde a un profesional.

Un código **corresponde** a un profesional.
 L₂ *sintagma nominal* **verbo_general** complemento

III.1.5. ESTRUCTURACIÓN DE LOS REQUISITOS DINÁMICOS

Se estructuran los Requisitos Dinámicos mediante la gramática del Lenguaje de Utilidad Dinámico de la base FORMOL. Esta gramática se detalló en la sección II.2.4.2. La salida de esta etapa está formada por un conjunto de oraciones condicionales que constituyen el Lenguaje de Utilidad Dinámico Preliminar.

A continuación, se estructuran los Requisitos Dinámicos del Procedimiento Cultural. Los Requisitos Dinámicos estructurados del Procedimiento de Modelo de Proceso se muestran al finalizar el estudio de las estructuras del Procedimiento Cultural.

En las oraciones condicionales que se especifican se remarcan con letra **negrita** la **oración subordinada** y se subraya la oración principal. Luego el análisis será similar al ya procesado, tanto para las oraciones subordinadas como para las principales. Es decir, las palabras en *cursiva* representan el *sintagma nominal*, las palabras en **negrita** el **verbo** empleado y las palabras subrayadas representan el complemento de la oración.

Se comienza con el análisis de las estructuras correspondientes a los requisitos dinámicos del Nivel de Ingeniería del MPM detectados:

1.4. Si y sólo sí **se obtiene el Modelo de Procesos** entonces se construye el Modelo de Personas Preliminar.

▪ **ANÁLISIS DE LA ORACIÓN SUBORDINADA**

Se obtiene *el Modelo de Procesos*

verbo_general *sintagma nominal*

▪ **ANÁLISIS DE LA ORACIÓN PRINCIPAL**

Se construye *el Modelo de Personas Preliminar*

verbo_general *sintagma nominal*

1.5. Si y sólo sí **el Modelo de Personas Preliminar es válido** entonces se crea el Modelo de Personas.

▪ **ANÁLISIS DE LA ORACIÓN SUBORDINADA**

El Modelo de Personas Preliminar **es** válido

sintagma nominal **verbo_general** complemento

▪ **ANÁLISIS DE LA ORACIÓN PRINCIPAL**

Se crea *el Modelo de Personas*

verbo_general *sintagma nominal*

1.6. Si y sólo sí **el Modelo de Personas Preliminar es inválido** entonces se repite el proceso desde la Educción del Universo Ecológico, o bien, desde cualquier etapa previa.

▪ **ANÁLISIS DE LA ORACIÓN SUBORDINADA**

El Modelo de Personas Preliminar **es** inválido

sintagma nominal **verbo_general** complemento

▪ **ANÁLISIS DE LA ORACIÓN PRINCIPAL**

Se repite

verbo_general

el proceso desde la Educción del Universo Ecológico, o bien, desde cualquier etapa previa

sintagma nominal

1.7. Si y sólo sí **se obtiene** el Modelo de Procesos entonces se construye el Modelo de Actividades y Roles.

▪ **ANÁLISIS DE LA ORACIÓN SUBORDINADA**

Se obtiene *el Modelo de Procesos*

verbo_general *sintagma nominal*

▪ **ANÁLISIS DE LA ORACIÓN PRINCIPAL**

Se construye *el Modelo de Actividades y Roles*

verbo_general *sintagma nominal*

1.8. Si y sólo sí **se obtienen los Modelos de Personas y de Actividades y Roles** entonces se genera el Modelo Cultural Preliminar.

▪ ANÁLISIS DE LA ORACIÓN SUBORDINADA

Se obtienen los Modelos de Personas y de Actividades y Roles

verbo_general *sintagma nominal*

▪ ANÁLISIS DE LA ORACIÓN PRINCIPAL

Se genera el Modelo Cultural Preliminar

verbo_general *sintagma nominal*

1.9. Si y sólo sí **el Modelo Cultural Preliminar es válido** entonces se crea el Modelo Cultural.

▪ ANÁLISIS DE LA ORACIÓN SUBORDINADA

El Modelo Cultural Preliminar es válido

sintagma nominal **verbo_general** complemento

▪ ANÁLISIS DE LA ORACIÓN PRINCIPAL

Se crea el Modelo Cultural

verbo_general *sintagma nominal*

1.10. Si y sólo sí **el Modelo Cultural Preliminar es inválido** entonces se repite el proceso desde la Educción del Universo Ecológico, o bien, desde cualquier etapa previa.

▪ ANÁLISIS DE LA ORACIÓN SUBORDINADA

El Modelo Cultural Preliminar es inválido

sintagma nominal **verbo_general** complemento

▪ ANÁLISIS DE LA ORACIÓN PRINCIPAL

Se repite

verbo_general

el proceso desde la Educción del Universo Ecológico, o bien, desde cualquier etapa previa

sintagma nominal

Nota: la estructura de la oración 1.10 coincide con la estructura de la oración 1.6, ya que hacen referencia a la etapa de validación de los modelos, repitiéndose el proceso en caso de no-validez.

1.11. Si y sólo sí **se obtiene el Modelo Cultural** entonces se determina el Modelo de Proceso Organizacional.

▪ ANÁLISIS DE LA ORACIÓN SUBORDINADA

Se obtiene el Modelo Cultural

verbo_general *sintagma nominal*

▪ **ANÁLISIS DE LA ORACIÓN PRINCIPAL**

Se determina *el Modelo de Proceso Organizacional*

verbo_general *sintagma nominal*

1.12. Si y sólo sí **se valida el Modelo de Proceso Organizacional** entonces se genera el Modelo de Proceso Organizacional Validado.

▪ **ANÁLISIS DE LA ORACIÓN SUBORDINADA**

Se valida *el Modelo de Proceso Organizacional*

verbo_general *sintagma nominal*

▪ **ANÁLISIS DE LA ORACIÓN PRINCIPAL:**

Se genera *el Modelo de Proceso Organizacional*

verbo_general *sintagma nominal*

2. Procedimiento de Modelo de Proceso

2.1. Si y sólo sí **se analizan los modelos de Organización Futura, de Proceso Organizacional y de los Registros Históricos de Procesos Software Generales** entonces se obtiene el Modelo de Requisitos.

▪ **ANÁLISIS DE LA ORACIÓN SUBORDINADA**

Se analizan

verbo_general

los modelos de Organización Futura, de Proceso Organizacional y de los Registros Históricos de Procesos Software Generales

sintagma nominal

▪ **ANÁLISIS DE LA ORACIÓN PRINCIPAL**

Se obtiene *el Modelo de Requisitos*

verbo_general *sintagma nominal*

2.2. Si y sólo sí **se analizan los modelos de Organización Actual y Futura y el Modelo de Proceso Organizacional** entonces se diagnostica el estado del MPSG de la organización.

▪ **ANÁLISIS DE LA ORACIÓN SUBORDINADA**

Se analizan

verbo_general

los modelos de Organización Actual y Futura y el Modelo de Proceso Organizacional

sintagma nominal

▪ **ANÁLISIS DE LA ORACIÓN PRINCIPAL**

Se diagnostica *el estado del MPSG de la organización*

verbo_general *sintagma nominal*

2.3. Si y sólo sí **se realiza el diagnóstico del MPSG** entonces se determina el Modelo de Mejora Organizacional.

▪ ANÁLISIS DE LA ORACIÓN SUBORDINADA

Se realiza *el diagnóstico del MPSG*

verbo_general *sintagma nominal*

▪ ANÁLISIS DE LA ORACIÓN PRINCIPAL

Se determina *el Modelo de Mejora Organizacional*

verbo_general *sintagma nominal*

2.4. Si y sólo sí **se estudia el Modelo de Proceso Organizacional y los conocimientos de los expertos del proceso software** entonces se comprende las relaciones relevantes de los elementos del Modelo de Proceso.

▪ ANÁLISIS DE LA ORACIÓN SUBORDINADA

Se estudia

verbo_general

el Modelo de Proceso Organizacional y los conocimientos de los expertos del proceso software
sintagma nominal

▪ ANÁLISIS DE LA ORACIÓN PRINCIPAL

Se comprende *las relaciones relevantes de los elementos del Modelo de Proceso*

verbo_general *sintagma nominal*

2.5. Si y sólo sí **se estudia el Modelo de Proceso Organizacional y los conocimientos de los expertos del proceso software** entonces se construyen los Modelos Conceptuales.

▪ ANÁLISIS DE LA ORACIÓN SUBORDINADA

Se estudia

verbo_general

el Modelo de Proceso Organizacional y los conocimientos de los expertos del proceso software
sintagma nominal

▪ ANÁLISIS DE LA ORACIÓN PRINCIPAL

Se construyen *los Modelos Conceptuales*

verbo_general *sintagma nominal*

2.6. Si y sólo sí **se generaron los Modelos Conceptuales** entonces se construyen los Modelos Formales.

▪ ANÁLISIS DE LA ORACIÓN SUBORDINADA

Se generaron *los Modelos Conceptuales*

verbo_general *sintagma nominal*

▪ ANÁLISIS DE LA ORACIÓN PRINCIPAL

Se construyen *los Modelos Formales*

verbo_general *sintagma nominal*

2.7. Si y sólo sí **los modelos cumplen con los objetivos definidos** entonces los Modelos Formales son válidos.

▪ **ANÁLISIS DE LA ORACIÓN SUBORDINADA**

Los modelos **cumplen** con los objetivos definidos

sintagma nominal **verbo_general** complemento

▪ **ANÁLISIS DE LA ORACIÓN PRINCIPAL**

Los Modelos Formales **son** válidos

sintagma nominal **verbo_general** complemento

2.8. Si y sólo sí **los modelos no cumplen con los objetivos definidos** entonces se repite el proceso desde las etapas previas.

▪ **ANÁLISIS DE LA ORACIÓN SUBORDINADA**

Los modelos no **cumplen** con los objetivos definidos

sintagma nominal **verbo_general** complemento

▪ **ANÁLISIS DE LA ORACIÓN PRINCIPAL**

Se repite *el proceso desde las etapas previas*

verbo_general *sintagma nominal*

III.1.6. CONSTRUCCIÓN DEL MODELO DE OBJETOS

En esta etapa se asigna a cada una de las estructuras lingüísticas que constituyen el Lenguaje de Utilidad Estático, el patrón conceptual estático correspondiente, descrito en la sección II.2.2.1 y se obtienen las clases, relaciones y cardinalidades. En la figura 3.1 se presenta el MO Parcial en donde están representadas las clases, relaciones y cardinalidades del Nivel de Ingeniería del MPM. Luego se reestudian las jerarquías de herencia y se determinan los atributos de las clase.

Con la información obtenida, se construye y verifica el MO representado en la figura 3.2.

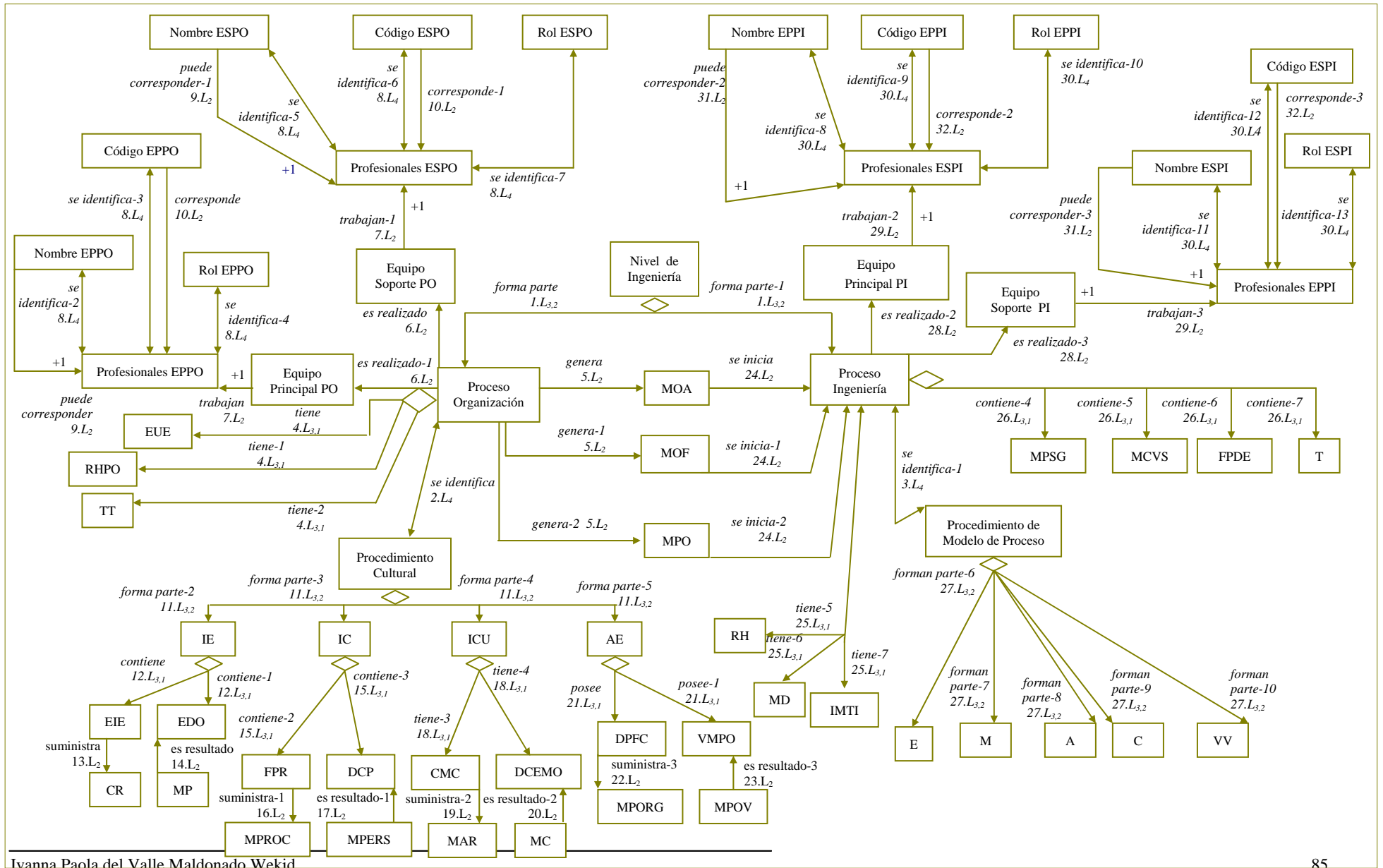


Figura 3.1. Modelo de Objetos Parcial del Nivel de Ingeniería

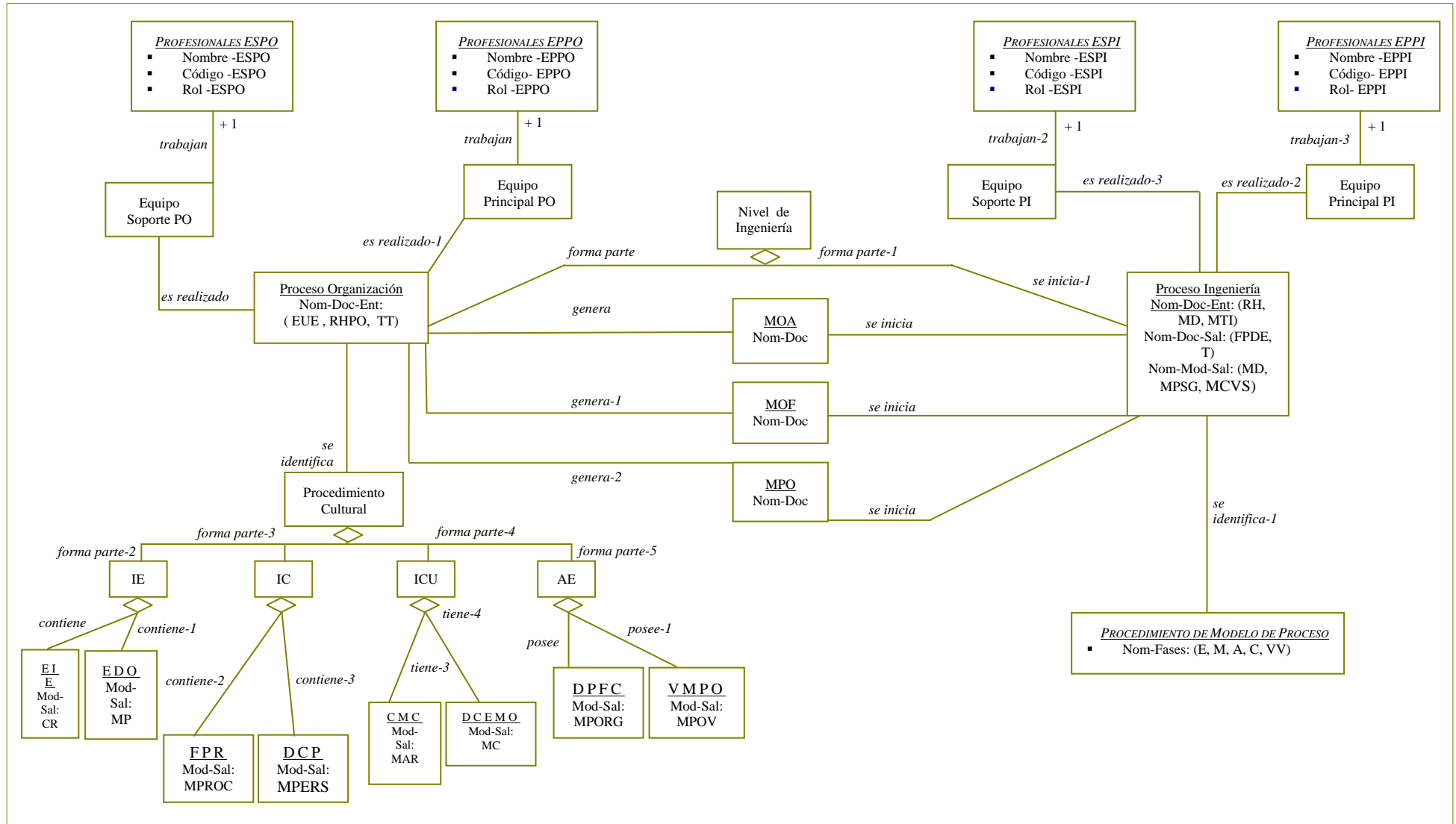


Figura 3.2. Modelo de Objetos del Nivel de Ingeniería del MPM

III.1.7. CONSTRUCCIÓN DEL MODELO DE COMPORTAMIENTO

En esta etapa se desarrolla el MC del Lenguaje de Utilidad Dinámico, descrito en la sección II.2.2.2. Para ello, se identifican los eventos, operaciones y condiciones de control y por último se comprueba la consistencia de dicho modelo. En esta última tarea se comprueba que todas las operaciones que pertenecen al mismo caso de uso están interrelacionadas. En esta aplicación se tienen dos casos de uso, el expresado por la operación cuya numeración está precedida por un **1**, y el expresado por las operaciones cuya numeración está precedida por un **2**. Se puede observar que en ambos casos todas las operaciones están interrelacionadas unas con otras, mediante al menos un evento interno.

El MC global del Nivel de Ingeniería se presenta en las figuras 3.3 y 3.4.

III.1.8. INTEGRACIÓN DEL MODELO DE OBJETOS Y DEL MODELO DE COMPORTAMIENTO

Para llevar a cabo esta etapa se define una matriz de clases y operaciones, cuyas filas corresponden a las clases identificadas en el MO, y las columnas a las operaciones internas básicas del MC.

Se marcan las casillas m_{ij} siempre que la operación j suponga una operación básica sobre algún objeto de la clase i .

Los resultados para el Procedimiento Cultural y el Procedimiento de Modelo de Proceso se representan en las tablas 3.4 y 3.5.

Por ejemplo, observando las tablas nombradas, para la clase del MO del Procedimiento Cultural: “**Modelo de Organización Actual (MOA)**” se realiza la operación del MC del Procedimiento Cultural: “**1.2 Se conforma el Modelo de Problemas**”. Para la clase del MO del Procedimiento de Modelo de Proceso: “**Modelo de Organización Futura (MOF)**” se realiza la operación del MC del Procedimiento de Modelo de Proceso: “**2.5 Se construyen los Modelos Conceptuales**”. En conclusión, existe una *correspondencia* entre clases y operaciones, que verifica la integración de ambos modelos conceptuales.

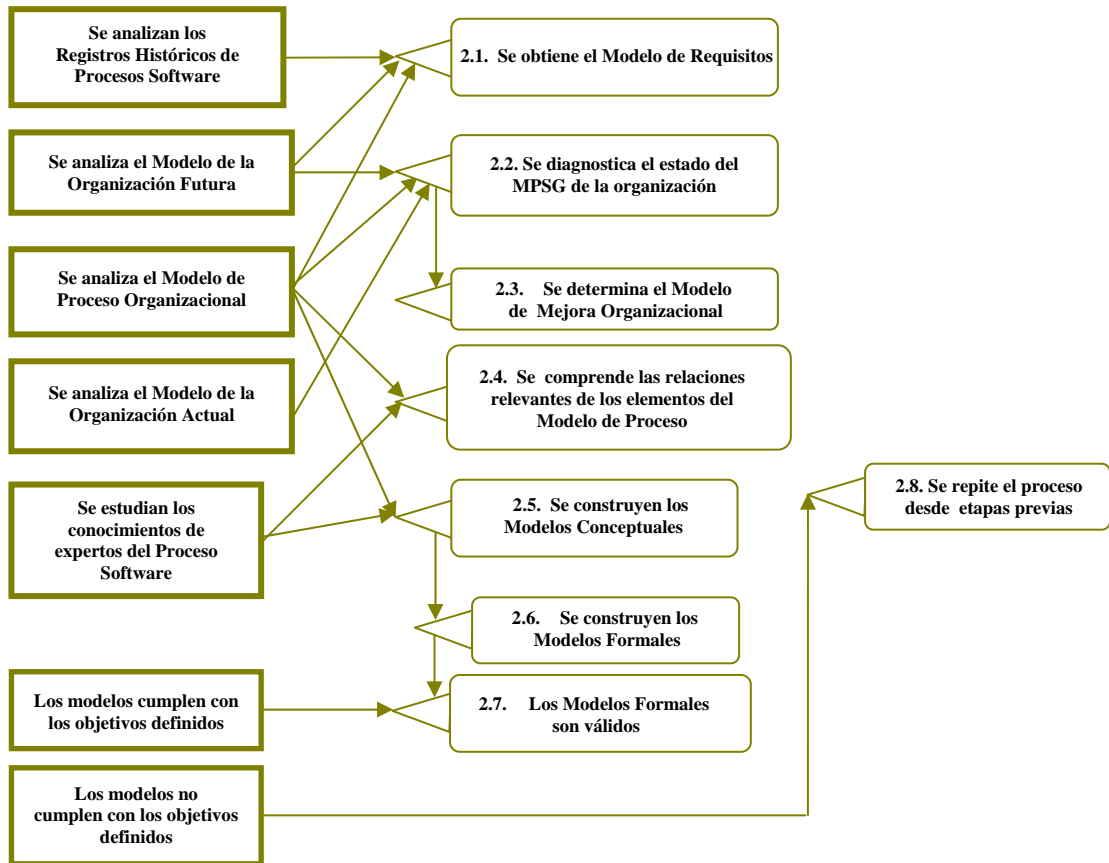


Figura 3.3. Modelo de Comportamiento del Procedimiento de Modelo de Proceso

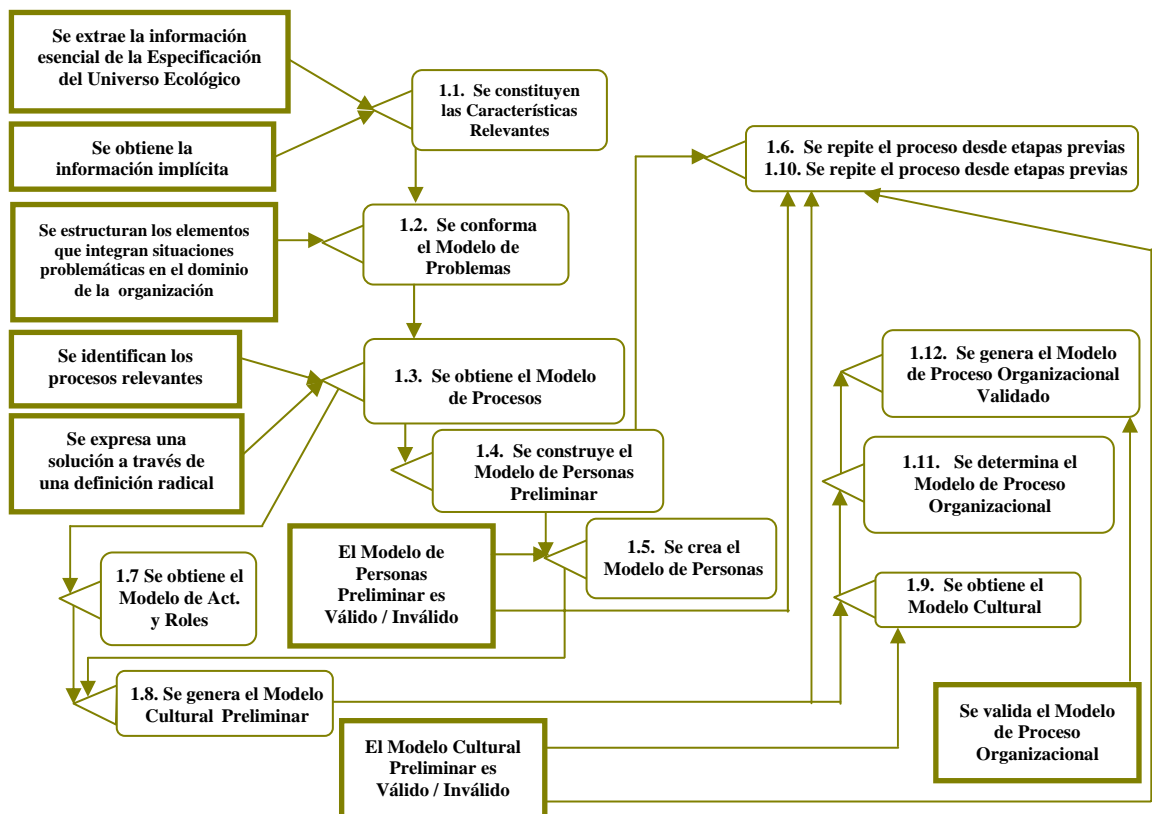


Figura 3.4. Modelo de Comportamiento del Procedimiento Cultural

Tabla 3.4. Matriz de asignación de operaciones a clases del Procedimiento Cultural

CLASES	ESTRUCTURAS										
	1.1 Se constuye las Características Relevantes	1.2 Se conforma el Modelo de Problemas	1.3 Se elabora el Modelo de Procesos	1.4 Se construye el Modelo de Personas Preliminar	1.5 Se crea el Modelo de Personas	1.6, 1.10 Se repite el proceso	1.7 Se obtiene el Modelo de Procesos	1.8 Se crea el Modelo Cultural Preliminar	1.9 Se crea el Modelo Cultural	1.11 Se determina el Modelo de Proceso Organizacional	1.12 Se genera el Modelo de Proceso Organizacional Validado
Nivel de Ingeniería											
Proceso Organización											
Procedimiento Cultural											
Investigación Exploratoria (IE)											
Investigación Cualitativa (IC)											
Investigación Cuantitativa (ICU)											
Análisis y Evaluación (AE)											
Extracción de la Información Esencial (EIE)											
Estudio de Dominio Organizacional (EDO)											
Formulación de los Procesos Relevantes (FPR)											
Determinación de las Capacidades de las Personas (DCP)											
Construcción de los Modelos Conceptuales (CMC)											
Determinación de la Cultura y Estrategia de Mejora de la Organización (DCEMO)											
Definición de Procesos Factibles Culturalmente (DPFC)											
Validación Del Modelo de Proceso Organizacional (VMPO)											
Modelo de Organización Actual (MOA)		X	X	X	X						
Modelo de Organización Futura (MOF)							X	X	X		
Modelo de Proceso Organizacional (MPO)										X	X
Equipo Principal PO											
Profesionales EPPO											
Equipo Soporte PO											
Profesionales ESPO											
Proceso Ingeniería											
Procedimiento Modelo de Proceso											
Equipo Principal PI											
Profesionales EPPI											
Equipo Soporte PI											
Profesionales ESPI											

Tabla 3.5. Matriz de asignación de operaciones a clases del Procedimiento de Modelo de Proceso

CLASES	ESTRUCTURAS							
	2.1. Se Obtiene el Modelo de Requisitos	2.2 Se diagnostica el estado del MPSG de la Organización	2.3 Se determina el Modelo de Mejora	2.4 Se comprende las Relaciones Relevantes de los elementos del Modelo de Proceso	2.5 Se construye los Modelos Conceptuales	2.6 Se construye los Modelos Formales	2.7 Los Modelos Formales son Válidos	2.8 Se repite el proceso desde las etapas previas
Nivel de Ingeniería								
Proceso Organización								
Procedimiento Cultural								
Investigación Exploratoria (IE)								
Investigación Cualitativa (IC)								
Investigación Cuantitativa (ICU)								
Análisis y Evaluación (AE)								
Extracción de la Información Esencial (EIE)								
Estudio de Dominio Organizacional (EDO)								
Formulación de los Procesos Relevantes (FPR)								
Determinación de las Capacidades de las Personas (DCP)								
Construcción de los Modelos Conceptuales (CMC)								
Determinación de la Cultura y Estrategia de Mejora de la Organización (DCEMO)								
Definición de Procesos Factibles Culturalmente (DPFC)								
Validación Del Modelo de Proceso Organizacional (VMPO)								
Modelo de Organización Actual (MOA)			X					
Modelo de Organización Futura (MOF)	X	X						
Modelo de Proceso Organizacional (MPO)	X	X		X	X			
Equipo Principal PO								
Profesionales EPPO								
Equipo Soporte PO								
Profesionales ESPO								
Proceso Ingeniería								
Procedimiento Modelo de Proceso								
Equipo Principal PI								
Profesionales EPPI								
Equipo Soporte PI								
Profesionales ESPI								

III.1.9. VERIFICACIÓN DEL MODELO DE OBJETOS Y DEL MODELO DE COMPORTAMIENTO

Se comprueba que entre las clases a las que pertenecen los métodos que se comunican entre sí en el MC, existe una relación en el MO. En este caso de aplicación, los resultados de la verificación son correctos.

En esta sección culminamos con uno de los aportes principales de este trabajo, lograr una formalización de un modelo de proceso software con la aplicación de un método de análisis formal.

CAPÍTULO IV. ANÁLISIS E INTERPRETACIÓN DE LOS RESULTADOS

"Invertir en conocimientos produce siempre los mejores intereses"
Benjamín Franklin

IV.1. CONSIDERACIONES PREVIAS

Este trabajo presenta un estudio empírico para comparar la eficiencia y la eficacia del método MATE respecto al PU. Para ello, fueron valoradas características sobresalientes y críticas de cada metodología, en el momento de conformar el experimento por muestras. Este estudio comparativo tiene por finalidad contrastar, precisar las ventajas y desventajas existentes y examinar ambas técnicas, para determinar el método de análisis más adecuado, que represente perfectamente el problema analizado y cumpla con las expectativas deseadas.

Para abordar de manera planificada el experimento: se identifican elementos que pueden poner en peligro la validez del estudio; se determinan amenazas, riesgos y retos que se deben enfrentar para que los resultados sean representativos y significativos; se consideran márgenes de error aceptables y previsibles; por último, se detectan factores de influencia indeseable para que no afecten la calidad y veracidad de los resultados.

Se trabaja con alumnos del último año de la carrera Licenciatura en Sistemas de Información de la Universidad Nacional de Santiago del Estero, con consideraciones previas de su estado curricular. También se considera la participación de profesionales del área informática y carreras afines de diferentes ámbitos universitarios del país. Los sujetos representativos de la realidad sumaron un total de 60 sujetos.

Se formaron dos muestras, que paralelamente tuvieron dos subdivisiones cada una, para responder encuestas sobre el Modelo de Objetos de MATE, el Modelos de Comportamiento de MATE, el Diagrama de Clases del PU y el Diagrama de Colaboración del PU. Cada encuesta se planifica para que los datos a recopilar sean razonables y de suma validez. Las respectivas encuestas se pueden consultar en el Anexo C.

El grupo total fue dividido equitativamente y al azar. Las encuestas se asignaron aleatoriamente, siguiendo las leyes de la estadística y de manera totalmente transparente para los integrantes de cada subdivisión. Cada encuesta tendrá el mismo nivel de complejidad, construcciones concretas y respuestas directas para lograr un examen valioso y positivo en esta experiencia. Se abordan sólo aspectos evaluables cualitativamente.

Mediante estímulos pedagógicos, se orienta a los sujetos a comprometerse voluntariamente a responder con seriedad, en tiempo y forma las encuestas. Se logra una convocatoria muy satisfactoria.

Los sujetos de una y otra encuesta, reciben una formación previa, documentada en cada muestra, para introducir el concepto del método, el modelo con el que se trabaja y el desarrollo paso a paso de su aplicación mediante un ejemplo típico no excesivamente complejo. En general, la idea que se procura es la de hacer más didáctica esta experiencia, de cara a los sujetos. En segunda instancia, la actividad solicitada es la construcción de un modelo, para apreciar la comprensión producida en los integrantes de cada muestra sobre los métodos propuestos. Finalmente, se busca, como ya se citó, resaltar defectos y virtudes para poder obtener una comparación de ambos métodos. Por ende, se hace hincapié en este hecho con la utilización de un cuestionario, con preguntas directas para evaluar dicha virtud o defecto y además, opciones indirectas para dar soporte a la respuesta primera, con la ambición de conseguir el éxito del experimento, que no afloraran dudas para evitar segundas interpretaciones y lograr un argumento justo en las conclusiones.

De esta forma, existe un doble reto: enfrentar una experiencia totalmente abierta a colaboración por parte de los alumnos y profesionales y por otra parte, la comprensión de los métodos de análisis explicados, su aplicación bajo una consigna clara de construcción y finalmente, las respuestas a preguntas sobre el entendimiento y construcción realizadas.

Como desenlace, el proyecto tuvo una duración de un mes y medio, donde se entregaron las encuestas y asimismo se recibieron los resultados de cada sujeto, con un total de 15 personas examinadas por cada encuesta.

Los resultados arrojados por el estudio apuntan al análisis y discusión de los datos para la conformación de las conclusiones finales y sobre todo para proporcionar una herramienta utilizable en el desarrollo del proceso software, como es MATE. Además, se desea que el producto final de esta tesis de graduación, consolide una base sólida puesta a disposición de otras investigaciones.

Antes de entrar en detalles sobre la interpretación de los resultados, es foco de importancia que ningún sujeto de la muestra tiene antecedentes del método MATE, pero no así sobre el PU, dado que todos los encuestados poseen algún conocimiento previo. Esto no es sorprendente debido a que el PU es completamente conocido, en

consecuencia la elección de este último, se ha debido al impacto y auge del mismo bajo el enfoque de desarrollo orientado a objetos.

✘ Obstáculos a superar

Previo al profundo estudio de los resultados alcanzados, se presentan los retos y desafíos involucrados en la práctica que se desarrolla, ellos son:

1. Conseguir la colaboración voluntaria de las personas seleccionadas por su estado curricular y profesionales que acepten la invitación a participar en esta experiencia.
2. Que las encuestas satisfagan efectivamente cada objetivo deseado.
3. Por parte de los sujetos encuestados, enfrentar y comprender los ejemplos escogidos.
4. Aplicar los conceptos de la OO para realizar la fase de Análisis con ambos métodos.
5. Entender y aplicar con destreza los métodos, en especial las consignas de construcción.
6. Confiar en que las respuestas sean realizadas concienzudamente, sean tomadas con seriedad y el realismo que el caso amerita.

IV.2. ANÁLISIS Y EVALUACIÓN DE LAS VARIABLES RESPUESTAS

El estudio experimental, llevado a cabo mediante las cuatro encuestas sobre uno y otro modelo, se calificó en base a 10 variables respuestas o marcos de referencia. Estas dan las pautas para que la hipótesis del trabajo de tesis sea refutada, aceptada, o bien, se tome en consideración resultados excepcionales no esperados, en función de la comparación, análisis y evaluación de los datos obtenidos.

Las variables respuestas serán tratadas en los apartados siguientes junto a la descripción y análisis de los resultados alcanzados con los gráficos correspondientes.

IV.2.1. VARIABLE RESPUESTA 1: FACILIDAD DE APRENDIZAJE

Este criterio de valoración se refiere al indicador de tiempo reloj (en minutos) insumido en cada muestra hasta comprender el ejemplo típico presentado en cada encuesta.

Para esto, cada sujeto indica el tiempo insumido en la comprensión de los ejemplos: del Modelo de Objetos y del Modelo de Comportamiento de MATE y del Diagrama de Clases y del Diagrama de Colaboración del PU.

Por ejemplo, si sumamos los tiempos indicados por los sujetos de cada muestra, se observa que: para los modelos estructurales, el Modelo de Objetos de MATE insume 30 minutos más que el Diagrama de Clases del PU; y para los modelos dinámicos, se observa que el Modelo de Comportamiento de MATE insume 27 minutos más que en el Diagrama de Colaboración del PU.

Los mayores tiempos son insumidos por los modelos de MATE, se infiere que este resultado es producto del desconocimiento del método, en cambio sobre el PU los sujetos tienen nociones, lo que ayuda en la comprensión de los diagramas de su modelo de análisis.

En la figura 4.1, se representa en la variable “y”, la suma del tiempo reloj insumido en el aprendizaje de cada método. Analizando el gráfico, el tiempo insumido para el Modelo de Objetos es de 109 minutos, mientras que para el Diagrama de Clases es de 79 minutos. El tiempo insumido para el Modelo de Comportamiento es de 88 minutos, mientras que para el Diagrama de Colaboración es de 61 minutos. De esta manera, se observa que las diferencias no son muy significativas para esta variable de referencia.

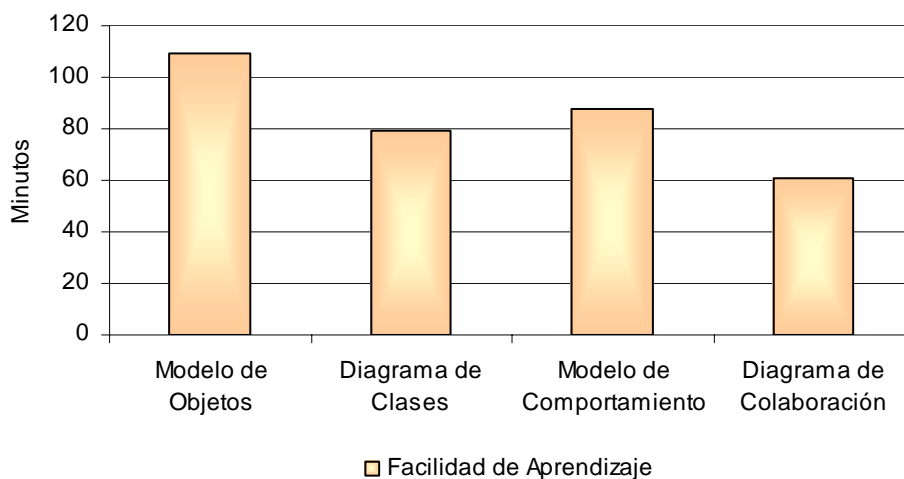


Figura 4.1. Representación Gráfica de la Variable Respuesta 1

IV.2.2. VARIABLE RESPUESTA 2: FACILIDAD DE REPRESENTACIÓN

Este criterio de valoración se refiere al indicador de tiempo reloj (en minutos) insumido por cada individuo para realizar la representación gráfica o textual que se solicita en cada encuesta. Es decir, cada sujeto realiza las siguientes tareas: *completa* un modelo en base a su descripción textual, *construye* un modelo en base al análisis de oraciones, *optimiza* un ejemplo trivial, *explica* textualmente un diagrama dado.

Para esto, cada sujeto indica el tiempo insumido en la realización de las tareas antes mencionadas. Este parámetro de evaluación considera la suma de los tiempo reloj (en minutos) que toman los sujetos de cada encuesta para llevar a cabo las consignas antes mencionadas.

Si sumamos los tiempos indicados por los sujetos de cada muestra, se observa que: para los modelos estructurales, el Modelo de Objetos de MATE insume 155 minutos más que el Diagrama de Clases del PU; y para los modelos dinámicos, se observa que el Modelo de Comportamiento de MATE insume 110 minutos más que en el Diagrama de Colaboración del PU.

Los mayores tiempos son insumidos en el desarrollo de los modelos del método MATE, el nuevo paradigma presentado. Se infiere que los modelos de MATE toman bastante tiempo para el análisis de las oraciones de la descripción textual de partida. Para el PU no se realiza ningún análisis previo, por lo que directamente se lleva a cabo la consigna solicitada.

En la figura 4.2., en la variable “y”, se representa la suma del tiempo reloj insumido en el aprendizaje de cada método. Analizando el gráfico, el tiempo insumido para el Modelo de Objetos es de 234 minutos, mientras que para el Diagrama de Clases es de 79 minutos. El tiempo insumido para el Modelo de Comportamiento es de 172 minutos, mientras que para el Diagrama de Colaboración es de 62 minutos. Estas diferencias se consideran relevantes para esta variable respuesta.

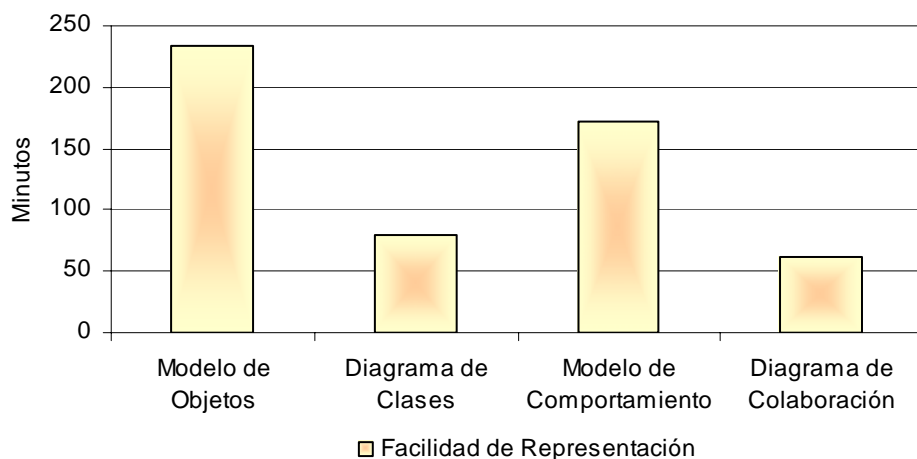


Figura 4.2. Representación Gráfica de la Variable Respuesta 2

IV.2.3. VARIABLE RESPUESTA 3: IDENTIFICACIÓN DE LOS ELEMENTOS

Este criterio de valoración se refiere a la facilidad de análisis de las oraciones de la descripción textual de la situación estudiada. Es decir, a la facilidad para identificar los componentes claves (clases, atributos, operaciones, relaciones) para la construcción de los modelos conceptuales. La apreciación de esta variable respuesta se atribuye solamente a encuestas de MATE, respecto a si las oración son de ayuda o no para la construcción.

Para la estimación de esta variable respuesta cada sujeto ha identificado las clases, los atributos, las operaciones y relaciones en las oraciones que describen un ejemplo típico presentado para el aprendizaje del método.

Los sujetos indican si es fácil la identificación de las clases, atributos, operaciones y relaciones y ayuda en el desarrollo de los modelos finales de MATE. Se les presentó, además oraciones para examinar y opciones múltiples de modelos resultantes construidos a partir del análisis de dichas oraciones. Estos ejercicios sirvieron para respaldar la respuesta anterior.

A partir de los resultados, se infiere que la facilidad de análisis se debe a que MATE guía en forma detallada e indica los pasos a seguir en el análisis de las necesidades representadas textualmente en las oraciones. Esta particularidad propia de MATE ayuda en la construcción y lo hace preciable y virtuoso en comparación con otros métodos de la OO.

En la figura 4.3, se observa que 23 sujetos sobre un total de 30, contestaron que las oraciones ayudan en la identificación de los elementos que constituyen los modelos de MATE. El resto se abstuvo de responder.

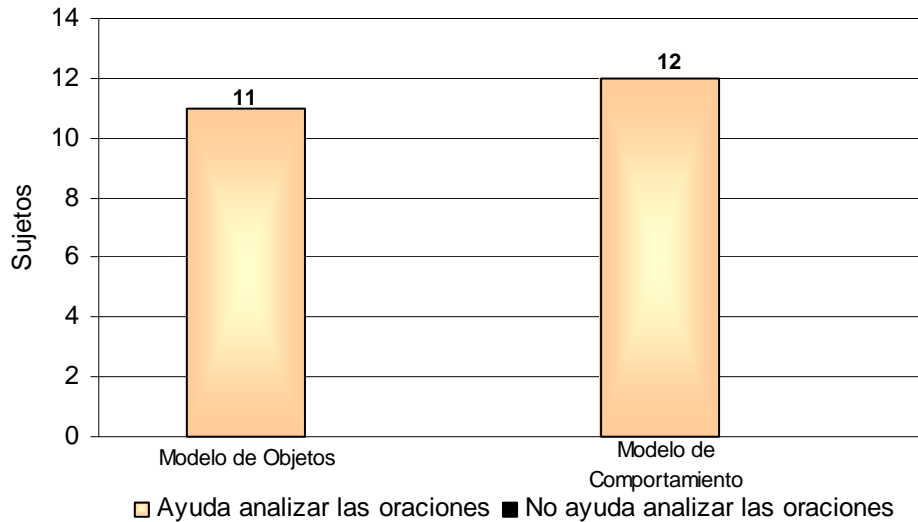


Figura 4.3. Representación Gráfica de la Variable Respuesta 3

IV.2.4. VARIABLE RESPUESTA 4: TRATAMIENTO DE LA INFORMACIÓN

Este criterio de valoración se refiere a la cantidad de sujetos que consideran que la información implícita y / o explícita existente en la descripción textual ayuda (o no) en la comprensión y construcción de los modelos. La apreciación de esta variable respuesta se atribuye solamente a encuestas de MATE, con respuesta binaria (sí, no).

Para analizar esta variable cada sujeto analiza la descripción textual sobre un problema dado y debe identificar la información implícita y / o explícita presentada.

Los sujetos de la muestra respondieron que la identificación de la información implícita y / o explícita es fundamental y ayuda en la construcción de modelos que se correspondan con la realidad modelizada. Como pregunta de control se presentó, tanto para el Modelo de Objetos como para el Modelo de Comportamiento, opciones múltiples de información implícita como explícita. En el primer modelos el 87% de los sujetos señaló la respuesta correcta y en el segundo modelos, el 67% hizo lo propio.

Estos resultados son razonables ya que se trata de lenguaje informal, y si no es contemplada la información significativa o irrelevante, incurrirá en un modelo con información incorrecta, inconsistente, incompleta y ambigua.

El tratamiento de la información es una de las características que privilegian a MATE sobre cualquier otro enfoque conocido. La primera etapa del método: Extracción de la Información Esencial, elimina la información no necesaria para el análisis, comprensión y construcción de los modelos de salida e incluye información que aparece implícitamente en la descripción textual de partida.

En la encuesta de MATE, 24 sujetos, sobre un total de 30, dijeron que tratar la información ayuda en la construcción de los modelos conceptuales, como se puede apreciar en la figura 4.4. Los sujetos restantes contestaron que la identificación de la información, tanto relevante como no significativa, no ayuda en la modelización.

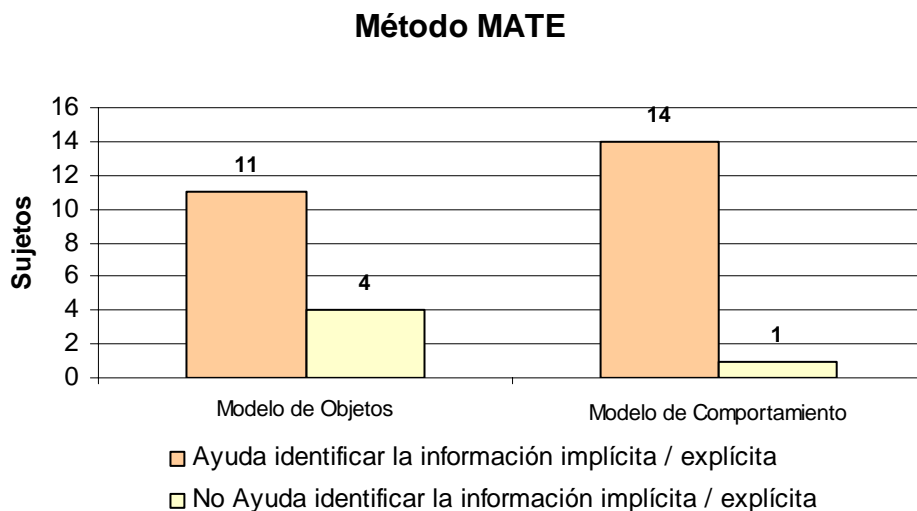


Figura 4.4. Representación Gráfica de la Variable Respuesta 4

IV.2.5. VARIABLE RESPUESTA 5: UNIFORMIDAD DE LOS MODELOS

Este criterio de valoración se refiere a la cantidad de sujetos que ha completado/construido/optimizado/explicado el ejemplo solicitado y han logrado modelos similares. Esto se tiene en cuenta porque favorece a la uniformidad de los resultados obtenidos y también para evitar la diversidad de posibles soluciones a un Análisis, sin saber formalmente cuál es la más adecuada. Se obtiene esta respuesta por observación en la coincidencia de la cantidad de clases identificadas, atributos

para cada clase incorporada, relaciones nuevas, uso correcto de la notación, etc. en los modelos construidos por los sujetos.

Para analizar esta variable respuesta, se tuvo en cuenta los Modelo de Objetos de MATE completados, los Modelo de Comportamiento de MATE construidos, los Diagrama de Clases del PU optimizados y los Diagrama de Colaboración del PU explicados por los sujetos.

Los resultados que se obtuvieron fueron los siguientes:

Todos los Modelos de Objetos construidos por diferentes sujetos fueron coincidentes. En relación al Modelo de Comportamiento los modelos en general fueron incompletos y poco claros. Sobre el Diagrama de Clases la mayoría de los sujetos no realizaron la optimización. Finalmente, en el Diagrama de Colaboración todas las descripciones textuales que explicaban el diagrama presentado eran coincidentes con el mismo, sin embargo, la notación no fue la correspondiente.

Se infiere entonces que: la uniformidad en la construcción de los Modelo de Objetos se debe al análisis previo de la descripción textual dada, que ayudó en la identificación de los elementos y tratamiento de la información implícita y / o explícita. Respecto al Modelo de Comportamiento, se puede hacer la salvedad de que la notación utilizada por MATE podría ser reformulada o actualizada para un mayor entendimiento en la modelización. En relación a los diagramas del PU, se observa que los Diagramas de Clases no fueron coincidentes, mientras que la explicación textual del Diagrama de Colaboración fue uniformemente redactada, sin hacer uso de la notación solicitada.

IV.2.6. VARIABLE RESPUESTA 6: CONSTRUCCIÓN DE LOS MODELOS

Este criterio de valoración se refiere a la cantidad de sujetos que han completado/construido/optimizado/explicado un determinado modelo y resulta legible. Se obtiene los resultados mediante la evaluación de las actividades solicitadas en la consigna de construcción.

Para valorizar esta variable se evalúa si el análisis de las oraciones realizado por los alumnos es correcta, si la notación ha sido bien utilizada y la representación de los modelos es correcta.

Los sujetos completaron sin dificultades el Modelo de Objetos, detectando las clases, los atributos, operaciones y relaciones. El Modelo de Comportamiento fue el que más complicaciones tuvo en su construcción, en general esta tarea no fue exitosa. El Diagrama de Clases fue realizado sin complicaciones y en el Diagrama de Colaboración se observó problemas en relación a la notación, como se mencionó en la variable respuesta anterior.

Por lo tanto se infiere que los Modelos de Objetos fueron completados correctamente porque MATE presenta tratamiento de la especificación textual que permite una construcción acorde a las necesidades del problema presentado. Los Modelos de Comportamiento construidos presentan confusión con las clases y sus relaciones. Los Diagramas de Clases no fueron difíciles de optimizar, posiblemente esto se deba a su notación simple y clara, por último, las explicaciones realizadas por los sujetos sobre el Diagrama de Colaboración dado fue correcta, a excepción de las notaciones del diagrama.

Se observa en la figura 4.5 que los modelos de Mate, sobre todo el Modelo de Comportamiento, fueron los de mayor complicación. Los diagramas de PU son los que no presentan grandes dificultades, sin embargo los sujetos también señalaron problemas con la notación del Diagrama de Colaboración. En general las dificultades de ambos métodos se presentaron en la interpretación de las notaciones y en el establecimiento de las relaciones.

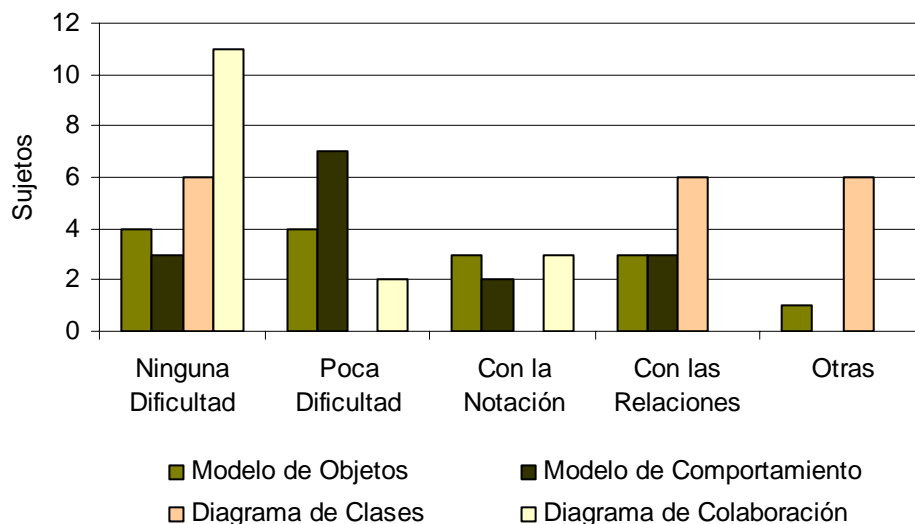


Figura 4.5. Representación Gráfica de la Variable Respuesta 6

IV.2.7. VARIABLE RESPUESTA 7: DEFECTOS DETECTADOS

Este criterio de valoración se refiere a la cantidad de errores encontrados en los modelos construidos/completados/optimizados/explicados por los sujetos. Se obtiene esta respuesta por análisis y evaluación de los modelos construidos por los sujetos. Los errores considerados son:

- Construcción del modelo incorrecta.
- Interpretación errónea de la consigna.
- Uso incorrecto de la notación y las relaciones.

Los resultados obtenidos del análisis de los Modelos de Objetos fue que el 90% lo hizo adecuadamente, de la construcción de los Modelos de Comportamiento 55% de los sujetos cometió errores, la consigna sobre la optimización de los Diagramas de Clases no fue cumplida por algunos sujetos, pero en general el margen de error fue mínimo y la explicación realizada de los Diagramas de Colaboración contó con un 75% de respuestas óptimas aunque presente el mal uso de la notación, como se mencionó anteriormente.

Se infiere, al ser MATE desconocido para los sujetos, que los errores detectados son razonables y satisfactorias las construcciones realizadas. No así respecto al PU, de quien los sujetos, tenía conocimiento previo.

IV.2.8. VARIABLE RESPUESTA 8: INTUICIÓN VERSUS GUÍAS

Este criterio de valoración se refiere a la cantidad de sujetos que respondieron sobre la preferencia de un método para el proceso de modelización.

Para obtener los resultados sobre esta variable respuesta se realiza el planteo a los sujetos sobre la preferencia de trabajar con guías y metodologías que ayuden en el desarrollo de sistemas, o bien, si prefieren basarse en su experiencia y criterio personal.

La mayoría optó por un método que presente pasos detallados para obtener los modelos resultantes del análisis, como lo hace MATE. En el PU, juega un papel importante los antecedentes, la habilidad y destreza del desarrollador, su conocimiento previo y la capacidad de intuición para descubrir los objetos claves que componen la modelización orientada a objetos.

Este marco de referencia es de vital importancia por ser una de las principales brechas que existe entre un método y el otro: MATE presenta pasos que guían al desarrollador del software durante todo el proyecto, PU se apoya en la experiencia del analista para construir los proyectos. PU paga un alto precio, debido a que en la fase de análisis pueden quedar requerimientos de usuario sin tratar, vagamente descritos, o bien, muchos detalles que se podrían haber precisado con mayor formalidad. Estos aspectos pendientes son resueltos en etapas posteriores. Es importante hacer notar que el modelo de análisis del PU hace abstracciones y evita resolver algunos problemas y tratar algunos requisitos que se supone es mejor posponer al diseño y a la implementación [Jacobson, 00].

Como vemos en los porcentajes de la figura 4.6, el 23% de los sujetos se inclinaron por una combinación entre guías y experiencia al encarar un proyecto de software, el 55% de ellos eligieron guiarse de un estudio y entendimiento del problema antes de realizar su modelización conceptual y el porcentaje restante se inclinó por la intuición, experiencia previa o criterio personal al modelizar determinado contexto de análisis.

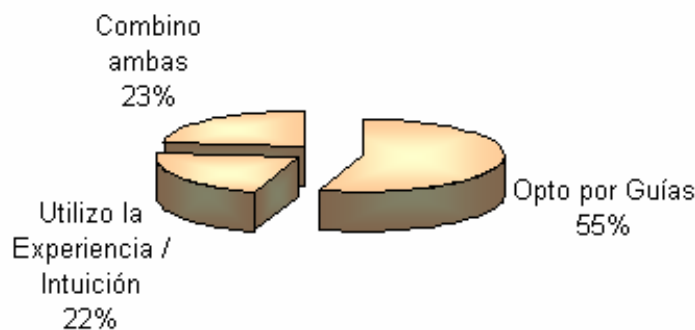


Figura 4.6. Representación Gráfica de la Variable Respuesta 8

Como una forma de avalar esta variable respuesta, se solicitó a los sujetos de cada muestra que emitan un comentario sobre el método que se les asignó para su aprendizaje y desarrollo. Las opiniones fueron positivas para ambos métodos. Se valora las apreciaciones sobre el método MATE debido a que los sujetos no tenían experiencia previa sobre el mismo.

En el Anexo D, se citan literalmente, las opiniones de los sujetos sobre MATE y PU.

IV.2.9. VARIABLE RESPUESTA 9: REPRESENTACIÓN DE LA REALIDAD

Este criterio de valoración se refiere a la cantidad de sujetos que consideran que un modelo representa de manera más correcta y adecuada la situación real bajo estudio.

Para obtener los resultados sobre esta variable se considera si el Modelo de Objetos, el Modelo de Comportamiento, el Diagrama de Clases y el Diagrama de Colaboración modelizan correctamente todos los requisitos analizados.

Como respuesta, en relación a los ejemplos presentados, los sujetos contestaron que los modelos del método MATE representan totalmente el problema modelado, pero a su vez los modelos resultantes son más complejos que los diagramas del PU que representan el ejemplo de manera parcial.

Se infiere entonces que gracias al estudio de la especificación en lenguaje textual del problema los modelos de MATE se construyen de manera completa. Las respuestas sobre los diagramas del PU son aceptables, dado que este método va postergando para etapas posteriores como el diseño y la implementación algunos requisitos que directamente no considera en el análisis.

En las figuras subsiguientes se representan las respuestas de los sujetos. En el gráfico 4.7 (a) sobre los diagramas de PU, el 83% contesta que reflejan parcialmente la realizada modelizada, el 17% contesta que representan totalmente la realidad y ninguno responde que la realidad no es reflejada. En el gráfico 4.7 (b), sobre modelos de MATE, el 84% responde que los modelos presentados representan totalmente el problema dado, el 13% dicen que lo hace de manera parcial y los restantes responden que no reflejan la realidad modelada.

IV.2.10. VARIABLE RESPUESTA 10: ELECCIÓN DEL MÉTODO

Este criterio de valoración se refiere a la cantidad de sujetos que eligieron y fundamentaron su selección sobre si aceptarían trabajar con un método u otro.

Las respuestas de los sujetos correspondieron a favor de la utilización del métodos MATE para el proceso de análisis; coincidentemente sobre el PU los sujetos dijeron que no era suficiente la fase de análisis para llevar a cabo un desarrollo de sistemas.

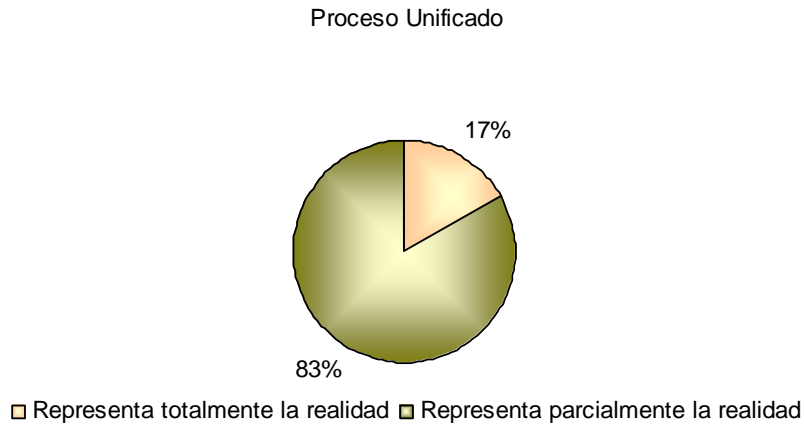


Figura 4.7. Representación Gráfica de la Variable Respuesta 9 (a)

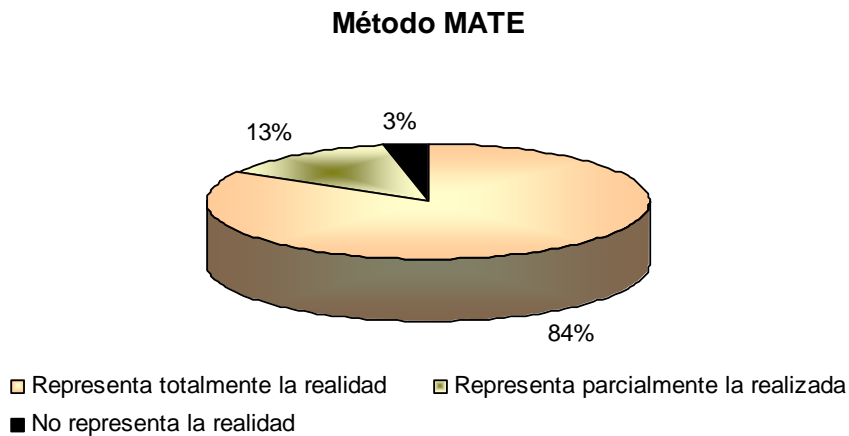


Figura 4.7. Representación Gráfica de la Variable Respuesta 9 (b)

Se infiere que aceptar emplear MATE es razonable porque empieza y termina con los modelos de salida correctamente modelizados, gracias al estudio previo que realiza; sin embargo, PU diagrama los requisitos por etapa integrando nuevas consideraciones en cada fase hasta arribar a los diagramas finales.

En la figura 4.8, el 40% de los sujetos elige encarar un proyecto de desarrollo utilizando MATE, el 10% no aceptaría trabajar con este método. El 20% elige emplear el modelo de análisis del PU, mientras que el 30% de los sujetos no abordarían un desarrollo solamente realizando la fase de análisis del PU.

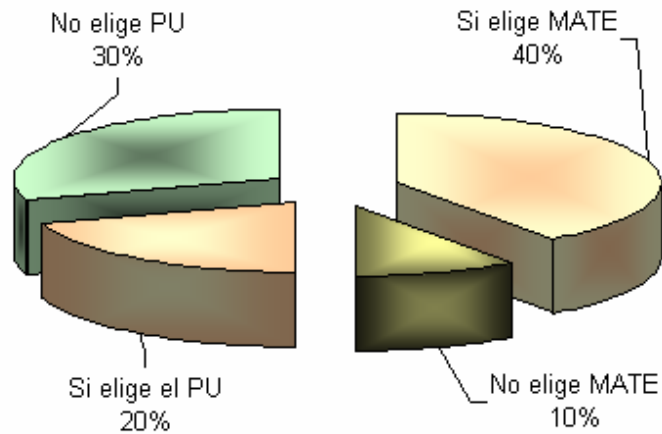


Figura 4.8. Representación Gráfica de la Variable Respuesta 10

IV.2.11. SÍNTESIS DE RESULTADOS

Con los datos generados a partir de los criterios anteriores, se completan las tablas 4.1, 4.2 y 4.3, para facilitar la evaluación comparativa de ambos métodos y lograr un resumen de los datos obtenidos.

Tabla 4.1. Comparación de los Tiempos

Menor tiempo insumido				
	Modelo de Objetos de MATE	Modelo de Comportamiento de MATE	Diagrama de Clases del PU	Diagrama de Colaboración del PU
Variable Respuesta 1			X	X
Variable Respuesta 2			X	X

En la tabla precedente, el menor tiempo insumido en el aprendizaje de los modelos presentados y realización de las actividades solicitadas fue atribuido al PU.

Tabla 4.2. Comparativa de las Características Destacadas

Características Destacadas				
	Modelo de Objetos de MATE	Modelo de Comportamiento de MATE	Diagrama de Clases del PU	Diagrama de Colaboración del PU
Variable Respuesta 3	X	X		
Variable Respuesta 4	X	X		
Variable Respuesta 5	X			X
Variable Respuesta 6			X	X
Variable Respuesta 7	X			X

En la tabla 4.2, el análisis de oraciones y tratamiento de la información implícita y / o explícita son excluyentes del método MATE. La uniformidad se presentó en los Modelos de Objeto de MATE y en el Diagrama de Colaboración del PU. Los diagramas del PU fueron los más comprensibles y de fácil notación y finalmente, en consideración en las actividades de construcción, el Modelo de Objetos de MATE y el Diagrama de Colaboración del PU fueron los que menos errores presentaron.

Tabla 4.3. Preferencia de Métodos

Selección del Método		
	MATE	PU
Variable Respuesta 8	X	X
Variable Respuesta 9	X	X
Variable Respuesta 10	X	

En la tabla 4.3, se resume lo siguiente: 1) ambos métodos fueron positivos para los sujetos en todas las tareas emprendidas en las encuestas, 2) que MATE refleja totalmente la realidad y que el PU lo hace parcialmente y 3) que el novedoso método MATE fue aceptable para los sujetos que lo elegirían antes que a PU para realizar una modelización.

CONCLUSIONES

“Crear posible algo es hacerlo cierto”
Christian Friedrich Hebbel

Las conclusiones de este trabajo están relacionadas, en primer lugar, con la modelización del Modelo de Proceso Multinivel (MPM) mediante el método MATE y en segundo lugar a la experiencia que permite contrastar dos métodos, MATE y el PU.

Con respecto a la modelización del Modelo de Proceso Multinivel se expresa que:

- ✦ El método MATE proporciona un conjunto de reglas justificadas, completas y correctas que han permitido identificar los elementos relevantes del MPM para construir sus modelos conceptuales.
- ✦ MATE proporciona un conjunto de guías que determinan de forma exhaustiva los pasos a seguir para llevar a cabo la tarea de modelización y construir los modelos adecuados al MPM.
- ✦ La obtención del modelo orientado a objetos del proceso multinivel se ha realizado mediante un método definido y formal.
- ✦ Al trabajar con los Lenguajes de Utilidad (patrones lingüísticos) de MATE, lleva al analista a pensar más sobre el problema, en lugar de modelizar directamente determinada información.
- ✦ La aplicación de la correspondencia entre los patrones lingüísticos y los patrones conceptuales justificada formalmente a través de FORMOL ha asegurado que los elementos de la modelización conceptual orientada a objetos de los procesos del MPM obtenidos sean correctos y el modelo conceptual multinivel esté justificado.
- ✦ El Modelo de Objetos y el Modelo de Comportamiento del MPM resultantes permiten soportar la modelización directa de las entidades del mundo real.
- ✦ La modelización de los conceptos reales del universo del discurso, permiten que la representación del proceso se obtenga de una forma directa y cercana a la realidad, reduciendo el vacío semántico y facilitando la tarea de análisis.
- ✦ El modelo conceptual del proceso multinivel obtenido es versátil, ya que permite, fácilmente, el diseño e implementación en cualquier tipo de paradigma, pudiendo dar lugar por ejemplo a un sistema experto, a una base de datos, a un sistema orientado a objetos, o incluso a un software procedimental.

- ✘ El modelo conceptual del proceso multinivel asegura el recubrimiento y modelización de todos los elementos (procesos, productos, personas, roles, organización) influyentes del proceso software.
- ✘ El MPM formalizado tiene un rango de aplicabilidad mayor que los modelos de proceso software tradicionales.

Por otro lado, en relación a la efectividad y eficiencia de MATE con respecto al PU, se puede inferir lo siguiente:

- ✘ Entender y aplicar los Lenguajes de Utilidad implica un mejor estudio y lucidez del problema a solucionar y que el desarrollador consiga una comprensión profunda acerca del sistema que está construyendo, tarea fundamental para realizar un buen análisis.
- ✘ MATE, además, especifica explícitamente criterios para completar, incluyendo información implícita, y verificar los modelos conceptuales resultantes, lo que permite asegurar que dichos modelos sean exactos.
- ✘ El Lenguaje de Utilidad conduce al analista a considerar los conceptos de la OO evitando modelizaciones desacertadas.
- ✘ Trabajar con los Lenguajes de Utilidad conlleva a insumir un tiempo adicional previo a la realización de la modelización conceptual. No obstante, el tiempo insumido en esta tarea se recompensa con creces al aplicar los resultados de FORMOL.
- ✘ FORMOL brinda criterios rigurosos para identificar directamente los elementos de los modelos conceptuales finales, logrando una correcta modelización, detección de redundancias, solución de inconsistencias que se van heredando y representación de la realidad de manera adecuada.
- ✘ El método que deriva del tratamiento de la descripción textual de entrada proporciona una serie de reglas formales que favorece la uniformidad de los resultados. Los modelos correspondientes a MATE son similares entre ellos, lo que evita la diversidad de posibles soluciones a un análisis.

- ✘ MATE explica, guía, apoya y asiste al analista etapa tras etapa en la actividad de modelado, de una manera que agiliza la comprensión y trae menos complicaciones a la hora del desarrollo.

El método MATE es un método que asegura la construcción correcta de los modelos correspondientes al proceso de Análisis. Es definido ya que especifica de forma rigurosa una serie de etapas y un conjunto de productos intermedios obtenidos como resultado de las acciones a realizar en dichas etapas y es formal ya que determina de forma exhaustiva los pasos a seguir para lograr la modelización conceptual. Por todo lo dicho, su proceso de Análisis está formalizado.

MATE es eficiente porque al guiar a los analistas en la actividad de modelización, realiza un aprovechamiento máximo de todos sus atributos y capacidades (lenguajes de utilidad, patrones conceptuales, aplicación de las etapas, etc.). La eficiencia se enfoca en los recursos, en manejarlos de la mejor manera posible. Pero aunque la eficiencia es importante la eficacia es aún más decisiva. La eficacia, tiene que ver con los resultados, está relacionada con la consecución de los objetivos. MATE fue creado de acuerdo a un objetivo que cumple satisfactoriamente: conduce correctamente en la ejecución de sus etapas, obteniendo modelos conceptuales idóneos, de calidad y significativa representación de la realidad. En consecuencia, MATE también es productivo.

BIBLIOGRAFÍA

*“Algunos libros son probados, otros devorados,
poquísimos masticados y digeridos”
Francis Bacon*

- [Abbot, 83] R. Abbot. Program Design by Informal English Description. Communications of the ACM. Nov. 1983.
- [Abdel, 91] T. Abdel-Hamid y S. Madnick, Software Project Dynamics: An Integrated Approach. (Prentice-Hall, 1991).
- [Acuña, 00a] S. T. Acuña y G. Barchini, “Knowledge and Software Engineering: a Methodological Framework to Symbiotic Software Process Modeling”. The Journal for the Integrated Study of Artificial, Intelligence. Cognitive Science and Applied Epistemology (CCAI'00), Vol. 17, Nº 1-2, (2000) 3-55.
- [Acuña, 00b] S. T. Acuña, M. Sosa, I. P. Maldonado y M. Y. Lescano “Modelización del Nivel de Ingeniería de un Modelo de Proceso Multinivel. Aplicación del Método de Análisis Formal MATE”. Anales de las Octavas Jornadas Universitarias sobre Computación de Santiago del Estero “Nuevas Tendencias en Ingeniería de Software” (JUCSE'00) (Octubre 2000) 19-74.
- [Acuña, 01] S. T. Acuña, M. Sosa, I. P. Maldonado y M. Y. Lescano “Modeling the Organization Process of a Multilevel Process Model Applying a Formal Analysis Method”. Anales del Simposio Argentino en Ingeniería de Software, 30º Jornadas Argentinas de Informática e Investigación Operativa (ASSE-JAIIO) (Septiembre 2001) 29-43.
- [Acuña, 97] S. T. Acuña, G. Barchini y C. Lasserre, “Symbiosis of software and knowledge engineering: A multilevel software process cycle model”. Proceedings of The 9th International Conference on Software Engineering and Knowledge Engineering (SEKE'97) (Junio 1997) 87-96.
- [Acuña, 99a] S. T. Acuña y G. Barchini, “The socio-cultural environment in the software process modeling”. Proceedings of The XIX International Conference of the Chilean Computer Science Society (SCCC'99), IEEE Press (Noviembre 1999) 216-227.
- [Acuña, 99b] S. T. Acuña y C. Lasserre. “Proceso Software. Un Modelo para la Ingeniería del Software y del Conocimiento”. EDUNJU. ISBN 950-721-117-9. (Noviembre, 1999).
- [Adelson, 85] B. Adelson y E. Soloway, “The role of domain experience in software design”. IEEE Trans. Software Engineering 11, 11 (Noviembre 1985) 1351-1360.
- [Adriaens, 95] G. Adriaens. “Technological evaluation of a controlled language application: Precision, recall and convergence test for SECC”. Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation, Leuven, Bélgica (1995) 123-141.
- [Argila, 94] C. Argila. Finding and Keeping Good Objects. Ammerican Programmer, October, 1994.
- [Bandinelli, 93] S. C. Bandinelli, A. Fuggetta y C. Ghezzi, “Software process model evolution in the SPADE environment”. IEEE Trans. Software Engineering 19, 12 (Diciembre 1993) 1128-1144.
- [Barbier, 92] F. Barbier. “Object oriented analysis of systems through their dynamical aspects”. Journal of Object Oriented Programming (Mayo 1992) 45-51.
- [Basili, 88] V. R. Basili, H. D. Rombach, “The TAME project: Towards improvement oriented software environments”. IEEE Trans. Software Engineering 14, 6 (June 1988) 758-773.
- [Block, 93] C.H. Block, M.R. MacMillan. A Prototype for Extracting Objects and Relationships from Software Specifications. The Journal of Knowledge Engineering. Spring 1993, vol. 16, nº 1.
- [Blum, 96] B. I. Blum, Beyond Programming: To a New Era of Design. (Oxford University Press, 1996).
- [Boehm, 81] B. W. Boehm, Software Engineering Economics. (Prentice-Hall, Englewood Cliffs, 1981).
- [Booch, 96] G. Booch. Análisis y Diseño Orientado a Objetos con Aplicaciones. (Addison Wesley, 1996).

- [Booch, 99] G. Booch, J. Rumbaugh y I Jacobson, *El Lenguaje Unificado de Modelado*. (Addison Wesley, 1999)
- [Buchholz, 96] E. Buchholz, H. Cyruaks, A. Düsterhöft, H. Mehlaai, B. Thalheim. *Applying Natural Language Dialogue Tool for Designing Data Bases*. Proceedings of the Second Workshop on Application of Natural Language to Data Bases, Amsterdam, The Netherlands, 1996.
- [Burg, 95] J. F. M. Burg y R. van de Riet. "Color-X linguistically-based event modelling: A general approach to dynamic modeling. Proceeding of the Seventh International Conference on Advanced Information Systems Engineering, Jyväskylä, Finlandia (1995) 26-39.
- [Burg, 97] J. F. M. Burg. *Linguistic Instruments in Requirements Engineering*. Tesis Doctoral. Vrije Universiteit, Amsterdam, The Netherlands (1997).
- [Calles, 93] M. Calles. *Método de Especificación Orientada a Objetos Derivada de una Especificación Estructurada*. Tesis Doctoral, Facultad de Informática, Universidad Politécnica de Madrid, España (1993).
- [Canals, 94] G. Canals, N. Boudjlida, J. C. Derniame, C. Godart y J. Lonchamp, "ALF: A framework for building process-centred software engineering environments". In *Software Process Modelling and Technology* cap. 7 Research Studies Press (1994) 153-185.
- [Carasik, 88] R. Carasik y C. Grantham, "A case study of CSCW in a dispersed organization". Proceedings of the CHI'88, Human Factors in Computing Systems, ACM (1988) 61-65.
- [Chen, 94] J-Y. Chen, "An integrated OO analysis and design method emphasizing E/C relationship and operation finding". *Journal of Systems and Software*. 24 (1994).
- [Chomsky, 89] N. Chomsky. *El Lenguaje y los Problemas del Conocimiento*. (Visor, 1989).
- [Coad, 95]. P.Coad. *Objects Models: Strategies, Patterns and Applications*. Englewood Cliffs, NJ: Yourdon Press/Prentice Hall, 1995.
- [Curtis, 85] B. Curtis, *Human Factors in Software Development*. (IEEE Computer Society, 1985).
- [Curtis, 88] B. Curtis, H. Krasner y N. Iscoe, "A field study of the software design process for large systems". *Communications of the ACM* 31, 11 (Noviembre 1988) 1268-1287.
- [Curtis, 92] B. Curtis, M. Kellner y J. Over, "Process modeling". *Communications of the ACM* 35, 9 (Septiembre 1992) 75-90.
- [Cyre, 95] W. Cyre. "A requirements sublanguage for automated analysis". *International Journal of Intelligent Systems*, 10 (1995) 665-689.
- [Dauphin, 96] E. Dauphin. "Corpora studies: A contribution to the definition of a controlled language". *First International Workshop on Controlled Languages Applications*, Leuven, Bélgica (1996) 193-204.
- [Derksen, 96] C.F. Derksen, P.J.M. Frederiks. *Paraphrasing as a Technique to Support OO Analysis*. TR-CSI-R9-603, January 1996. Computer Science Institute, Faculty of Mathematics and Informatics. Catholic University of Nijmegen, The Netherlands.
- [Dowson, 90] M. Dowson, B. Nejmech y W. Riddle, "Concepts for process definition and support". Proceedings of the Sixth International Software Process Workshop. (Octubre 1990).
- [Engels, 94] G. Engels y L. Groenewegen, "SOCCA: Specifications of coordinated and cooperative activities". In *Software Process Modelling and Technology* cap. 4. (Research Studies Press, 1994) 71-102.
- [Faukl, 97] S. R. Faulk. *Software Requirements: A Tutorial*. (Software Engineering. IEEE Computer Society Press, 1997).

- [Feiler, 93] P. H. Feiler y W. S. Humphrey, "Software process development and enactment: Concepts and definitions". Proceedings of the Second International Conference on Software Process (Febrero 1993) 28-40.
- [Finkelstein, 94] Finkelstein, J. Kramer y B. Nuseibeh, Software Process Modelling and Technology. (Research Studies Press, 1994).
- [Flood, 91] R. Flood and M. Jackson, Creative Problem Solving: Total Systems Intervention (John Wiley & Sons, 1991).
- [Franch, 99] X. Franch y J. M. Ribó. "Using UML for software process modelling". Proceedings of the Second International Conference of the Unified Modeling Language, UML'99, IEEE (Octubre 1999) 292-307.
- [Frederiks, 95] P. J. H. Frederiks, C. H. A. Kister y Th. P van de Weide. Object Oriented Analysis using Informal Language. Technical Report Computer Science Institute - R9516, Faculty of Mathematics and Informatics, Catholic University of Nijmegen, Nijmegen, Países Bajos (1995).
- [Fuchs, 96] N. Fuchs y R. Schwitter. "Attempto controlled english (ACE)". First International Workshop on Controlled Language Applications. Katholieke Universiteit Leuven, Leuven, Bélgica (1996) 124-136.
- [Gerstl, 92] P. Gerstl. "Linking linguistic and non-linguistic information". Data & Knowledge Engineering 8, 3 (Julio 1992) 305-222.
- [Guindon, 88] R. Guindon y B. Curtis, "Control of cognitive processes during design: What tools would support software designers?". Proceedings of the CHI'88, Human Factors in Computing Systems, ACM (1988) 263-268.
- [Hastie, 87] R. Hastie, "Experimental evidence on group accuracy". In Information Processing and Group Decision-Making, Ed. G. Owen y B. Grofman, (JAI Press, 1987) 129-157.
- [Honiden, 93] S. Honiden, N. Kotaka y Y. Kishimoto. Formalizing Specification Modelling in OOA. (IEEE Software, 1993).
- [Hoobs, 90] J. R. Hoobs. Literature and Cognition. Lecture Notes, Centre for the Study of Language and Information, Lelan Stanford Junior University, Gran Bretaña (1990).
- [Huijsen, 95] W-O. Huijsen. Introduction to Controlled Languages. <http://www.wots.let.ruu.nl/www/Controlled-language> (Junio 1995).
- [IEEE, 91] IEEE Standard for Developing Software Life Cycle Processes, IEEE Standard 1074-1991.
- [IEEE, 93] IEEE. Guide to Software Requirements Specification. IEEE Std. 830-1993.
- [Iivari, 91] J. Iivari "Object Oriented information system analysis: A framework for object identification". Proceedings of the Twenty-Fourth Annual Hawaii International Conference on System Sciences, Hawaii, U.S.A (1991).
- [Iivari, 95] J. Iivari. Object-Oriented as Structural, Functional and Behavioural Modelling: a Comparison of Six Methods for Object-Oriented Analysis. Information and Software Technology, Vol. 38, 1996, pp: 155-163.
- [Jacobson, 00] I. Jacobson, G. Booch y J. Rumbaugh, El Proceso Unificado de Desarrollo de Software. (Addison Wesley, 2000)
- [Jacobson, 92] I. Jacobson. Object-Oriented Software Engineering: a Use Case Approach. Addison-Wesley, Wokingham, Gran Bretaña, 1992.
- [Junkermann, 94] G. Junkermann, B. Peuschel, W. Schäfer y S. Wolf, "MERLIN: Supporting cooperation in software development through a knowledge-based environment". In Software Process Modelling and Technology cap. 5. (Research Studies Press, 1994) 103-129.

- [Kawalek, 96] P. Kawalek y D. G. Wastell, "Organisational design for software development: a cybernetic perspective". In *Lecture Notes in Computer Science, Software Process Technology: Proceedings of the 5th European Workshop 1149* (Springer-Verlag, 1996) 258-270.
- [Kellner, 91] M. I. Kellner, "Software process modelling support for management planning and process modelling". *Proceedings of the First International Conference on Software Process* (Octubre 1991) 8-28.
- [Kristen, 94] G. Kristen. *Object Orientation: The KISS-method. From Information Architecture to Information System*. (Addison-Wesley, 1994).
- [Laudon, 96] Laudon, K, Laudon, J. "Administración de los Sistemas de Información. Organización y Tecnología", Edit Prentice Hall, 3era, 1996, México.
- [Lehman, 91] M. Lehman, "Software engineering, the software process and their support". *Software Engineering Journal* 6, 5 (Setiembre 1991) 243-258.
- [Lonchamp, 90] J. Lonchamp, K. Benali, C. Godart y J. C. Derniame, "Modeling and enacting software processes: an analysis". *Proceedings of the 14th Annual International Computer Software and Applications Conference* (Octubre-Noviembre 1990) 727-736.
- [Martin, 92] J. Martin y J. Odell. *Object Oriented Analysis and Design*. (Prentice Hall, 1992).
- [McChesney, 95] R. McChesney, "Toward a classification scheme for software process modelling approaches". *Information and Software Technology* 37, 7 (1995) 363-374.
- [Meyer, 88] B. Meyer. *Object Oriented Software Constructions*. (Prentice Hall, 1988).
- [Min, 97] Sang-Yoon Min y Doo-Hwan Bae, "MAM nets: A Petri-net based approach to software process modeling, analysis and management". *Proceedings of the 9th International Conference on Software Engineering and Knowledge Engineering* (Junio 1997) 78-86.
- [Moreira, 94] A.M.D. Moreira, R.G. Clark. *Rigorous Object-Oriented Analysis*. *Lecture Notes in Computer Science, OO Methodologies and Systems*. *Proceedings of the International Symposium ISOOM'94, Palermo, Italy, September, 1994*.
- [Moreno, 97] M. Moreno, "Método Formal de Modelización Conceptual para Sistemas Software". Tesis Doctoral. Facultad de Informática, Universidad Politécnica de Madrid (1997).
- [Naduri, 95] S. Naduri, S. Rugeber. *Requirements Validation via Automated Natural Language Parsing*. *Proceedings of the Twenty-Eighth Hawaii International Conference of Systems Sciences, Hawaii, U.S.A. 1994*.
- [Nguyen, 96] M. N. Nguyen y R. Conradi, "Towards a rigorous approach for managing process evolution". In *Lecture Notes in Computer Science, Software Process Technology: Proceedings of the 5th European Workshop 1149*. Springer-Verlag (1996) 18-35.
- [Northrop, 97] L.M. Northrop. *Object-Oriented Development in Software Engineering*. IEEE Computer Society Press, 1997.
- [Paulk, 95] M. C. Paulk, C. V. Weber y M. B. Chrissis, *The Capability Maturity Model: Guidelines for Improving the Software Process*. (Addison-Wesley, 1995).
- [Penedo, 91] M. H. Penedo y C. Shu. "Acquiring experiences with the modelling and implementation of the project life-cycle process: the PMDB work". *Software Engineering Journal* 6, 5 (Septiembre 1991) 259-274.
- [Pulman, 96] S. G. Pulman. "Controlled language for knowledge representation". *First International Workshop on Controlled Language Applications, Leuven, Bélgica* (1996) 233-242.
- [Rolland, 92] C. Rolland, C. Proix. *A Natural Language Approach for Requirements Engineering*. *Proceedings of the Conference of Advanced Information Systems Engineering'92*. Manchester, United Kingdom, 1992.

- [Rumbaugh, 91] J. Rumbaugh, M. Blaha, W Premerlani, F Hedí y W Sorensen. Object Modeling Technique. (Prentice Hall, 1991).
- [Rumbaugh, 96] J. Rumbaugh. "Notation notes: Principles for choosing notation". Journal of Object-Oriented Programming (Mayo 1996) 11-14.
- [Scacchi, 84] W. Scacchi, "Managing software engineering projects: A social analysis". IEEE Trans. Software Engineering 10, 1 (Enero 1984) 49-59.
- [Seaman, 94] B. Seaman y V. R. Basili, "OPT: An approach to organizational and process improvement". Proceedings of AAAI Symposium on Computational Organizational Design (Marzo 1994).
- [Sherdil, 96] K. Sherdil y N. H. Madhavji, "Human-oriented improvement in the software process". In Lecture Notes in Computer Science, Software Process Technology: Proceedings of the 5th European Workshop 1149 (Springer-Verlag, 1996) 145-166.
- [Shlaer, 92] S. Shaler y S. Mellor. Modeling the World in States. Prentice Hall. 1992.
- [Sigfried, 96] S. Sigfried. Understanding Object Oriented Software Engineering. (IEEE Computer Society Press, 1996).
- [Sommerville, 95] I. Sommerville and T: Rodden, "Human, social and organisational influences on the software process". Lancaster University, Computing Department, Cooperative Systems Engineering Group, technical Report: CSEG/2/1995 (1995), pp. 1-21.
- [Sutcliffe, 97] A. G. Sutcliffe. Object-Oriented Systems Development: Survey of structured methods. In Software Engineering. (IEEE Computer Society Press, 1997) 160-169.
- [Thamhain, 87] H. J. Thamhain y D. L. Wilemon, "Building high performance engineering project teams". IEEE Trans. Engineering Management 34, 3 (Marzo 1987) 130-137.
- [Tjoa, 93] A.M. Tjoa, L. Berger. Transformation of Requirements Specification Expressed in Natural Language into an EER Model. Proceedings of the 12th International Conference on ER Approach (ER'93) Arlington, Texas, USA, 1993.
- [Tseng, 92] F.S.C. Tseng. A.C.P. Chen, W-P. Yang. On Mapping Natural Language Constructions into Relational Algebra Through E-R Representation. Data & Knowledge Engineering, vol. 9, 1992/93.
- [Tully, 89] Tully, "Representing and enacting the software process". Proceedings of the 4th International Software Process Workshop, ACM SIGSOFT Software Engineering Notes 14, 4 (Junio 1989).
- [ULPGC, 99] "Introducción a la Teoría de Sistemas. Señales y Sistemas: Conceptos". Grupo de Informática y Sistemas. Universidad de las Palmas de Gran Canaria (ULPGC). Disponible vía <http://serdis.dis.ulpgc.es/> (Junio 1999).
- [Van de Riet, 95] R. Van de Riet y J. F. M. Burg. "The impact of linguistics on conceptual models: Consistency and understability". Proceedings of the First International Workshop on Applications of Natural Language to Data Bases (1995).
- [Wang, 97] S. Wang. "A synthesis of natural language, semantic network, and objects for bussines process modelling". Canadian Journal of Administrative Sciences (1997).
- [Wieringa, 96] R. J. Wieringa. Requirements Engineering: Frameworks for Understanding (Wiley, 1996).
- [Yourdon, 95] E. Yourdon, K. Whitehead, J. Thomann, K.Oppel, P. Nevermann. Mainstream Objects. An Analysis and Design Approach for Business. Prentice Hall 1995.

ANEXO A

CONCEPTOS LINGÜÍSTICOS

Se emplea un subconjunto del lenguaje natural como información de partida para la construcción de los modelos conceptuales. A continuación se dan unas breves definiciones de los conceptos lingüísticos que se usarán para su construcción.

En el estudio de la lingüística destacan tres ramas: la *morfología*, la *sintaxis* y la *semántica*.

La **morfología** estudia los distintos tipos de palabras. Las clases o categorías morfológicas son: el *artículo*, el *sustantivo*, el *adjetivo*, el *pronombre*, el *verbo*, el *adverbio*, la *preposición*, la *conjunción* y la *interjección*. Tienen un papel relevante en los sublenguajes que se propone el *sustantivo*, el *verbo* y la *conjunción*.

Un *sustantivo* o *frase sustantiva* se define como la palabra, o conjunto de palabras, que expresa la esencia de lo que nombra, lo que son las cosas, por ejemplo, mesa, árbol, idea, etc. Los sustantivos se pueden clasificar en *sustantivos comunes* o *sustantivos propios*. Un *sustantivo común*, que es el empleado en los sublenguajes propuestos, se utiliza para designar el conjunto al que pertenece la realidad que nombra. Un *sustantivo propio* nombra la realidad particular, diferenciándola del resto de realidades.

Un *verbo* es un tipo de palabra que expresa la acción que realiza o sufre cualquier realidad. La *conjunción* es un tipo de palabra que actúa de nexo entre elementos del mismo nivel gramatical (conjunción coordinante) o entre elementos de distinto nivel gramatical (conjunción subordinante). Para los sublenguajes que se proponen se usará la *conjunción coordinante copulativa* “y”, la *conjunción coordinante disyuntiva* “o”, la *conjunción subordinante condicional* “si” y la *conjunción subordinante consecutiva* “entonces”. También existe otro nexo entre elementos gramaticales que es la *aposición* o coma “,”.

La **sintaxis** es la ciencia que estudia las relaciones entre los distintos tipos de palabras estudiadas por la morfología, para construir unidades superiores de significado más completo que el que esas palabras representan independientemente. La unidad mínima que se ha de tener en cuenta en sintaxis es el *sintagma* y la máxima la oración.

Un sintagma es un palabra o conjunto de palabras con significado propio, que cumple una función sintáctica dentro del discurso. Las funciones sintácticas

que pueden tener un sintagma son las de *sujeto*, *verbo*, o *complemento* dentro de una *oración*.

Una *oración* es una palabra o conjunto de palabras que expresan un sentido gramatical completo. Es la unidad más pequeña de sentido completo en sí misma, en la que ese puede fragmentar el discurso, una oración expresa una relación entre sus partes, el *sujeto* y el *predicado*.

El *sujeto* de una oración es un sintagma de tipo *sintagma nominal* que puede parecer o no y nombra la realidad de la que se va a decir algo. Un *sintagma nominal* es el que tiene como parte central, como núcleo, un sustantivo, pronombre o frase sustantiva. Un *sintagma nominal* precedido por una proposición se denomina *sintagma proposicional*. Desde el punto de vista de los sublenguajes propuestos se considerará como *sintagma nominal*.

El *predicado* es un *sintagma verbal*, que es el que tiene como núcleo un verbo, más los *sintagmas* que complementan dicho verbo. El núcleo del *sintagma verbal* puede ser cualquiera de las formas verbales de la conjugación del verbo, o la perífrasis verbal (verbo más infinitivo, gerundio o participio, por ejemplo, voy a comer). Para los sublenguajes propuestos, el núcleo del *sintagma verbal* está formado por el conjunto de palabras que proporcionan la semántica asociada con la acción indicada por el verbo. Por ejemplo, en la oración “los alumnos están bajo la responsabilidad de su tutor”, la estructura verbal sería la estructura “estar bajo la responsabilidad” y no estrictamente “estar”.

Los sintagmas que complementan al verbo en el predicado reciben el nombre de *complementos*. Los *complementos* pueden hacer la función de *complemento directo*, *complemento indirecto*, *complemento circunstancial de tiempo*, *modo*, *aspecto*, *causa*, *instrumento*, *compañía*, *materia* o *complemento agente*. Desde el punto de vista de los sublenguajes que se proponen no es necesario distinguir entre los distintos tipos de complementos.

Las oraciones se pueden dividir en oraciones simples u oraciones compuestas, según si la expresión del significado de dicha oración se realiza mediante un único verbo, o mediante varios verbos, respectivamente. Las oraciones simples se pueden clasificar teniendo en cuenta la actitud psicológica del emisor. Desde este punto de vista se pueden dividir en enunciativas, interrogativas, exclamativas, exhortativas, dubitativas y desiderativas. Para los sublenguajes, interesan únicamente las oraciones enunciativas, que son las que afirman (*afirmativas*) o niegan (*negativas*)

de forma objetiva la acción expresada. Las oraciones compuestas se pueden clasificar en yuxtapuestas (si la oración compuesta está formada por varias oraciones simples sin ningún nexo que las una), coordinadas (si se establece una relación de unión, oraciones coordinadas copulativas, o exclusión, oraciones coordinadas disyuntivas, entre oraciones simples, independientes gramaticalmente) o subordinadas (si entre las oraciones simples que forman la oración compuesta existe una relación de dependencia sintáctica de una de las oraciones -oración subordinada- respecto de la otra oración -oración principal-). Dentro de las oraciones subordinadas, las oraciones subordinadas condicionales son de especial interés para la aplicación del método MATE. En este tipo de aplicaciones se hace depender el cumplimiento de lo expresado en la oración principal del cumplimiento de lo expresado en la oración subordinada.

Por último, la semántica es la ciencia que estudia el significado de los distintos tipos de palabras, las distintas realidades que puede representar una palabra. Desde este punto de vista, son de interés para la aplicación del método, los sinónimos que son palabras distintas con el mismo significado y las palabras polisémicas, es decir, palabras iguales con distinto significado.

ANEXO B

ACRÓNIMOS

Los siguientes son los más utilizados y significativos en este libro:

- (A) Análisis
- (AE) Análisis y Evaluación
- (C) Construcción
- (CMC) Construcción de los Modelos Conceptuales
- (CR) Características Relevantes
- (DC) Diagrama de Colaboración del Proceso Unificado
- (DCEMO) Determinación de la Cultura y la Estrategia de Mejora de la Organización
- (DCL) Diagrama de Clases del Proceso Unificado
- (DCP) Determinación de las Capacidades de las Personas
- (DPFC) Definición de Procesos Factibles Culturalmente
- (IC) Investigación Cualitativa
- (ICU) Investigación Cuantitativa
- (IE) Investigación Exploratoria
- (IMTI) Innovaciones Metodológicas y Técnicas de la Ingeniería del Software y la Ingeniería del Conocimiento
- (M) Mejora
- (MATE) Método de Análisis derivado de una Explicación Textual
- (MC) Modelo Cultural Validado
- (MC) Modelo de Comportamiento de MATE
- (MCVS) Modelos del Ciclo de Vida del Software
- (MD) Modelo del Dominio
- (MO) Modelo de Objetos de MATE
- (MOA) Modelo de Organización Actual
- (MOF) Modelo de Organización Futura
- (MP) Modelo de Problemas
- (MPERS) Modelo de Personas
- (MPO) Modelo de Proceso Organizacional
- (MPORG) Modelo de Proceso Organizacional
- (MPOV) Modelo de Proceso Organizacional Validado

(MPROC) Modelos de Procesos
(MPSG) Modelos de Actividades y Roles
(MAR) Modelos de Proceso Software Generales
(PI) Proceso Ingeniería
(PO) Proceso Organización
(PU) Proceso Unificado
(RH) Registros Históricos
(RHPO) Modelo de Procesos Organizacionales
(T) Tecnologías
(TT) Tecnologías Transferidas
(VPMO) Validación del Modelo de Proceso Organizacional
(VV) Validación Verificación

ANEXO C

ENCUESTA :

METODO DE ANÁLISIS DERIVADO DE UNA EXPECIFICACIÓN TEXTUAL (MODELO DE OBJETOS)

FINALIDAD la presente encuesta forma parte de la validación empírica del Método de Análisis derivado de una Explicación Textual (**MATE**), correspondiente al Trabajo de Graduación denominado “**Método de Análisis derivado de una Especificación Textual. Su Validación Empírica**” para optar por el Título de Licenciado en Sistemas de Información de la UNSE. Los resultados de las encuestas servirán para la aceptación o refutación de la hipótesis planteada.

La encuesta se estructura en tres partes:

- **PARTE A:** Presentación conceptual de **MATE** y ejemplificación de un caso típico *paso a paso*.
- **PARTE B:** Resolver la consigna.
- **PARTE C:** Responder el cuestionario.

DESTINATARIOS: Alumnos de la carrera Licenciatura en Sistemas de Información y carreras relacionadas.

Nota: Por favor, registrar el tiempo reloj que insume la realización de la encuesta (PARTE A y PARTE B) de la presente.

PARTE A: CONCEPTO DE MATE Y UN EJEMPLO “PASO A PASO”

TIEMPO INSUMIDO EN LEER ESTA PARTE:

El método MATE *parte de la descripción textual* del problema y *va guiando paso a paso* al analista en la construcción de los modelos que representan una solución para el usuario. Los modelos de salida de MATE son el **MODELO DE OBJETOS (MO)**, que modeliza los aspectos estáticos y el **MODELO DE COMPORTAMIENTO (MC)**, que modeliza los aspectos dinámicos [Moreno, 97].

EJEMPLO

PASO 1: Descripción textual de los aspectos estáticos de una empresa destinada a la venta de vehículos.

“La empresa de venta de vehículos trabaja con un conjunto de clientes. La empresa se encuentra ubicada en una de las mejores zonas comerciales. Cada cliente está identificado por un código de cliente, un nombre y una dirección. El proveedor se corresponde con un código, nombre y dirección. Cada modelo de vehículo corresponde exclusivamente a un proveedor. Cada modelo se identifica por un código y tiene un nombre. El pedido guarda la fecha de solicitud y el número de pedido. Se considera que, los clientes pueden haber realizado varios o ningún pedido. Pueden existir modelos que no son solicitados en ningún pedido. Un pedido corresponde a un único cliente y a un modelo.”

PASO 2: Se analiza y modeliza *cada oración* con el método MATE.

La descripción textual de partida puede o no contener información *implícita*, que incorporaremos con una nueva oración; o bien, *explícita*, que descartaremos.

Un ejemplo de oración con información *explícita* es: “La empresa se encuentra ubicada en una de las mejores zonas comerciales”. Como no interfiere en el análisis del sistema, se elimina.

PASO 3: Una vez finalizado el paso anterior, analizamos cada oración de la descripción textual resultante para detectar: las *clases* participantes (en rojo), sus *atributos* (en azul) y *relaciones* entre ellas. El verbo (en verde) establece la relación que existe entre las clases.

PASO 4: Modelizamos una oración ejemplo de la siguiente manera, para obtener el MO de MATE.

Observemos la oración ejemplo, donde se aplicó el PASO 3 y el PASO 4:

La **empresa trabaja** con un conjunto de **clientes**.

Cada **cliente** está identificado por un **código** de cliente y un **nombre**.



PASO 5: El **MO** se construye en base al análisis realizado de cada oración de la descripción textual resultante incorporando información implícita y excluyendo información explícita. Esta es una de las principales particularidades del método MATE.

Finalmente, el **MO** de MATE, para este caso práctico, queda diseñado de la siguiente manera.



PARTE B: COMPLETA EL DIAGRAMA

TIEMPO INSUMIDO EN LEER ESTA PARTE:

Con la descripción textual precedente, *completa* el **MO** de la PARTE A.

Nota: Puedes copiar el MO anterior y completarlo. Utiliza una página en blanco.

Ayuda: hay 3 clases y 5 relaciones más.

“La empresa trabaja con un grupo de vendedores. Los vendedores se identifican por un código de vendedor y cada vendedor tiene un nombre. Todo vendedor ha realizado como mínimo un pedido. Un pedido corresponde a un único vendedor, pudiendo solicitar cualquier número de opciones. Pueden existir opciones que no son solicitadas en ningún pedido. Cada modelo tiene asignadas, muchas opciones. Cada opción es propia de un solo modelo y se vende a un determinado precio. Cada opción se identifica por un código y por un nombre. Una entrega notifica un pedido. Cada pedido aparecerá únicamente en una entrega. Cada entrega está identificada por un número de entrega y contiene el importe del vehículo vendido.”

PARTE C: CUESTIONARIO

Cuando corresponda, marca con **X** la respuesta elegida.

- 1) ¿Qué tiempo te insumió entender el ejemplo de la PARTE A para el MO?
 ¿Qué tiempo te insumió *completar*, en la PARTE B, el MO?

- 3) ¿Has comprendido el concepto de MATE, presentado en la PARTE A, mediante el análisis de las oraciones?
 - i. El análisis de las oraciones ayudó a la comprensión
 - ii. El análisis de las oraciones no se entendió
 - iii. Tomaron tiempo destinado a la comprensión del modelo en sí
 - iv. Otra opinión (escribe abajo)

4) **Lee el siguiente ejemplo:**

“Un videoclub tiene a sus clientes identificados por un número de socio, nombre y dirección. Cada película se identifica por un código. También se guarda el título y los actores. Se clasifican en películas para mayores de 18 años o aptas para todo público. Cuando un socio alquila una película, el sistema comprueba la identificación del socio y asigna la película a dicho socio. El socio realiza posteriormente la devolución, liberando el sistema la película. El sistema se ocupa de la actualización constante de los clientes que adeudan películas. Cuando el videoclub compra una película, el sistema la da de alta, y cuando ésta se deteriora entonces la descarta. El club no piensa, por el momento, abrir sucursales, ya que está ubicado en una localización donde no tiene competencia.”

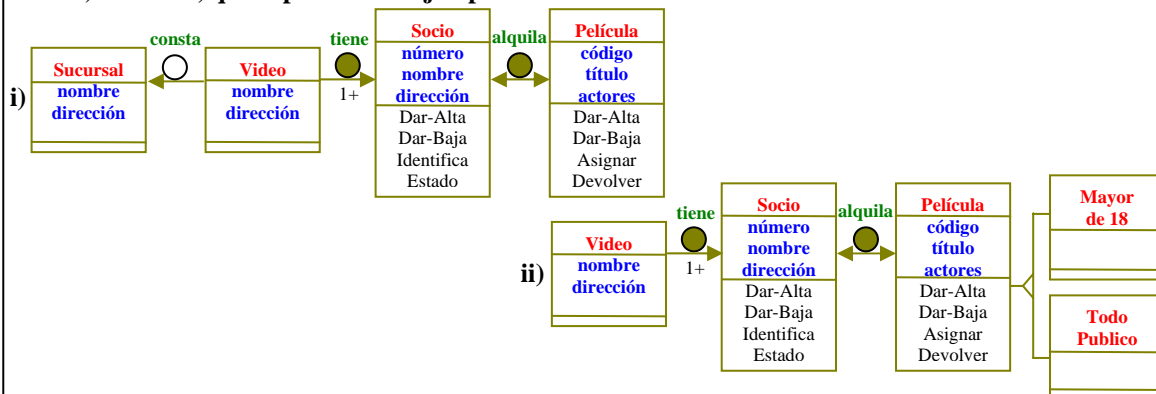
Marca una oración que contenga información explícita:

- i. Cada película se identifica por un código.....
- ii. Un videoclub alquila películas a sus clientes identificados por un número de socio, nombre, dirección y fecha de nacimiento.....
- iii. El club no piensa, por el momento, abrir sucursales, ya que está ubicado en una localización donde no tiene competencia.....

5) **¿Incluir la información implícita y / o explícita ayudó en la diagramación del MO?**

- i. Si
- ii. No

6) **El MO, que representa el ejemplo anterior es el:**



- 6) **Incluir la información implícita y / o explícita hace del MO, un modelo que:**
(puedes marcar más de una opción)
- i. Representa totalmente la realidad modelada
 - ii. No representación la realidad modelada
 - iii. Representa parcialmente la realidad modelada
 - iv. El modelo es más redundante
 - v. El modelo es más complicado
 - vi. Otra opinión (escribe abajo)
- 7) **¿Cuáles han sido las dificultades encontradas al *completar* el MO de la PARTE B?**
- i. Ninguna
 - ii. Poca, el modelo es poco complicado
 - iii. Dificultad con la notación
 - iv. En el entendimiento de las relaciones
 - v. Otras (escribe abajo)
- 8) **¿Crees que el método MATE resultaría provechoso en el desarrollo de sistemas, según las características presentadas? Justifica tu respuesta.**
- i. Si
 - ii. No
- Justificación (escribe abajo):
- 9) **En el momento de realizar el análisis de un proyecto de software:**
- i. Seleccionas una método que guíe el proceso
 - ii. Utilizas solamente tu intuición y/o experiencia previa
- 10) **Comentarios**

ENCUESTA :

METODO DE ANÁLISIS DERIVADO DE UNA EXPECIFICACIÓN TEXTUAL (MODELO DE COMPORTAMIENTO)

FINALIDAD la presente encuesta forma parte de la validación empírica del Método de Análisis derivado de una Explicación Textual (MATE), correspondiente al Trabajo de Graduación denominado “Método de Análisis derivado de una Especificación Textual. Su Validación Empírica” para optar por el Título de Licenciado en Sistemas de Información de la UNSE. Los resultados de las encuestas servirán para la aceptación o refutación de la hipótesis planteada.

La encuesta se estructura en tres partes:

- **PARTE A:** Presentación conceptual de MATE y ejemplificación de un caso típico *paso a paso*.
- **PARTE B:** Resolver la consigna.
- **PARTE C:** Responder el cuestionario.

DESTINATARIOS: Alumnos de la carrera Licenciatura en Sistemas de Información y carreras relacionadas.

Nota: Por favor, registrar el tiempo reloj que insume la realización de la encuesta (PARTE A y PARTE B) de la presente.

PARTE A: CONCEPTO DE MATE Y UN EJEMPLO “PASO A PASO”

TIEMPO INSUMIDO EN LEER ESTA PARTE:

El método MATE *parte de la descripción textual* del problema y *va guiando paso a paso* al analista en la construcción de los modelos que representan una solución para el usuario. Los modelos de salida de MATE son el **MODELO DE OBJETOS (MO)**, que modeliza los aspectos estáticos y el **MODELO DE COMPORTAMIENTO (MC)**, que modeliza los aspectos dinámicos [Moreno, 97].

EJEMPLO

PASO 1: Descripción textual de los aspectos dinámicos de una empresa destinada a la venta de vehículos. “El vendedor comunica el pedido al sistema. Una vez que el sistema tiene constancia de los pedidos, se los confirma al cliente y, semanalmente, envía a los proveedores dichos pedidos. Los proveedores al realizar las entregas de los vehículos a los clientes, las notifican a la empresa quien les paga.”

PASO 2: Se analiza y modeliza cada oración con el método MATE.

La descripción textual presentada puede o no contener información implícita, que incorporaremos con una nueva oración; o bien, explícita, que descartaremos.

Un ejemplo de oración con información implícita es: “...el vendedor comunica el pedido al sistema...”. Como interfiere en el análisis del sistema, incorporamos esta información para la modelización como se muestra a continuación.

“**Cuando se efectúa una venta, el vendedor comunica el pedido al sistema. Una vez que el sistema tiene constancia de los pedidos, se los confirma al cliente y, semanalmente, envía a los proveedores dichos pedidos.** Los proveedores al realizar las entregas de los vehículos a los clientes, las notifican a la empresa quien les paga.”

PASO 3: Una vez finalizado el paso anterior, analizamos cada oración de la descripción textual resultante utilizando la estructura “Si y sólo si entonces”. De esta forma, se determinan las oraciones principales y las subordinadas dentro de cada estructura.

Observemos la siguiente oración:

Oración 1: *Si y sólo si un vendedor efectúa una venta entonces el vendedor comunica el pedido al sistema*

Oración Subordinada

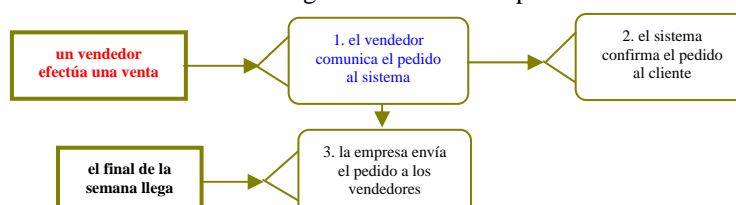
Oración Principal

PASO 4: Modelizamos la oración ejemplo de la siguiente manera, para obtener el MC de MATE empleando la estructura de abajo.



PASO 5: El MC se construye en base al análisis realizado de cada oración de la descripción textual de partida incorporando información implícita y excluyendo información explícita. Esta es una de las principales particularidades del método MATE.

Finalmente, el MC de MATE, para este caso práctico, queda diseñado de la siguiente manera y corresponde a las oraciones resaltadas con negrita en el texto del paso 2.



P A R T E B : CONSTRUYE EL DIAGRAMA

TIEMPO INSUMIDO EN LEER ESTA PARTE:

Con la descripción textual precedente, *construye* el MC que se corresponda a las dos oraciones en negrita. “Cuando se efectúa una venta, el vendedor comunica el pedido al sistema. Una vez que el sistema tiene constancia de los pedidos, se los confirma al cliente. **El sistema envía a los proveedores dichos pedidos, quienes informan a la empresa. Los proveedores al realizar las entregas de los vehículos a los clientes, las notifican a la empresa quien les paga.**”

Puedes copiar y pegar la simbología a emplear (**desagrupa los elementos, trabaja en una página en blanco**):



P A R T E C : CUESTIONARIO

Cuando corresponda, marca con **X** la respuesta elegida.

- 1) ¿Qué tiempo te insumió entender el ejemplo de la PARTE A para el MC?
 ¿Qué tiempo te insumió *construir*, en la PARTE B, el MC?
- 2) ¿Has comprendido el concepto de MATE presentado en la PARTE A, mediante el análisis de las oraciones?
 - i. El análisis de las oraciones ayudó a la comprensión
 - ii. El análisis de las oraciones no se entendió
 - iii. Tomaron tiempo destinado a la comprensión del modelo en sí
 - iv. Otra opinión (escribe abajo)

3) **Lee el siguiente ejemplo:**

“Cuando un socio alquila una película, el sistema comprueba la identificación del socio y asigna la película a dicho socio. El socio realiza posteriormente la devolución, liberando el sistema la película. El sistema se ocupa de la actualización constante de los clientes que adeudan películas. Cuando el videoclub compra una película, el sistema la da de alta, y cuando ésta se deteriora entonces la descarta.”

Marca una oración con información implícita:

- i. El sistema se ocupa de la actualización constante de los clientes que adeudan películas.....
- ii. Cada película se identifica por un código
- iii. Cuando el videoclub compra una película, el sistema la da de alta, y cuando ésta se deteriora entonces la descarta.

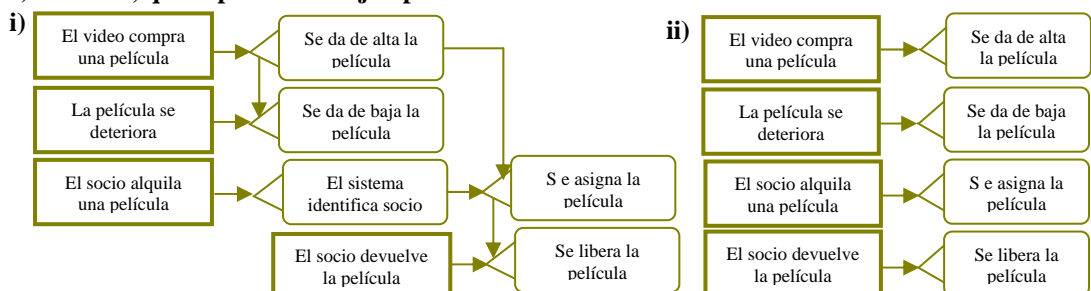
¿Qué implica la oración identificada?

- i. Implica que el sistema da de alta y baja las películas
- ii. Implica que se puede encontrar las películas identificadas con un código.....
- iii. Implica que si un cliente es moroso debe darse de baja

4) **¿Incluir la información implícita y / o explícita ayudó en la diagramación del MC?**

- i. Si
- ii. No

5) **El MC, que representa el ejemplo es el:**



- 6) **Incluir la información implícita y / o explícita hace del MC, un modelo que:**
(puedes marcar más de una opción)
- i. Representa totalmente la realidad modelada
 - ii. No representación la realidad modelada
 - iii. Representa parcialmente la realidad modelada
 - iv. El modelo es más redundante
 - v. El modelo es más complicado
 - vi. Otra opinión (escribe abajo)
- 7) **¿Cuáles han sido las dificultades encontradas al *completar* el MC de la PARTE B?**
- i. Ninguna
 - ii. Poca, el modelo es poco complicado
 - iii. Dificultad con la notación
 - iv. En el entendimiento de las relaciones
 - v. Otras (escribe abajo)
- 8) **¿Crees que el método MATE resultaría provechoso en el desarrollo de sistemas, según las características presentadas? Justifica tu respuesta.**
- i. Si
 - ii. No
- Justificación (escribe abajo):
- 9) **Comentarios:**

ENCUESTA :

PROCESO UNIFICADO (DIAGRAMA DE CLASES)

FINALIDAD: la presente encuesta forma parte de la validación empírica del Método de Análisis derivado de una Explicación Textual, correspondiente al Trabajo de Graduación denominado “Método de Análisis derivado de una Especificación Textual. Su Validación Empírica” para optar por el Título de Licenciado en Sistemas de Información de la UNSE. Los resultados de las encuestas servirán para la aceptación o refutación de la hipótesis planteada.

La encuesta se estructura en tres partes:

- **PARTE A:** Presentación conceptual del Proceso Unificado (PU) y ejemplificación de un caso típico.
- **PARTE B:** Resolver la consigna.
- **PARTE C:** Responder el cuestionario.

DESTINATARIOS: Alumnos de la carrera Licenciatura en Sistemas de Información y carreras relacionadas.




Nota: Por favor, registrar el tiempo reloj que insume la realización de la encuesta (PARTE A y PARTE B) de la presente.

PARTE A: CONCEPTO DEL PU Y EJEMPLO

TIEMPO INSUMIDO EN LEER ESTA PARTE:

El PU es un Proceso de Desarrollo de Software basado en UML. Es iterativo, centrado en la arquitectura y dirigido por los casos de uso. El análisis, flujo de trabajo del PU, refina y estructura los requisitos del usuario, representándolos a través del **DIAGRAMA DE CLASES** (*modeliza aspectos estáticos*) y el **DIAGRAMA DE COLABORACIÓN** (*modeliza aspectos dinámicos*)

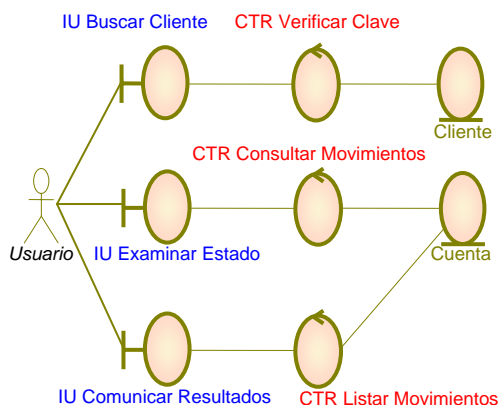
Para representar los diagramas del PU se emplea:

1.  para modelizar la **CLASE DE INTERFAZ (IU)**, que conecta al usuario con el sistema.
2.  para modelizar la **CLASE DE CONTROL (CTR)**, que coordina las acciones y deriva el trabajo.
3.  para modelizar la **CLASE DE ENTIDAD**, que almacena información que persistente en el sistema.

EJEMPLO

El diagrama estudiado representa el **proceso de consulta del estado de una cuenta** de usuario en un cajero automático.

El *Diagrama de Clases* resultante es:



CLASES

- “**IU Buscar Cliente**”: representa la interfaz gráfica que permite al usuario ingresar su clave para verificar su existencia.
- “**IU Examinar Estado**”: representa la interfaz gráfica que permite al usuario consultar los movimientos correspondientes a su cuenta.
- “**IU Comunicar Resultados**”: representa la interfaz gráfica que permite al usuario ver los movimientos correspondientes a su cuenta corriente.
- “**CTR Verificar Clave**”: toma desde la clase IU Buscar Cliente la clave del usuario y posteriormente consulta en la entidad Cliente la existencia de la clave ingresada.
- “**CTR Consultar Movimientos**”: toma desde la clase IU Examinar Estado la consulta ingresada por el usuario y posteriormente busca en la entidad Cuenta los movimientos realizados en la misma.
- “**CTR Listar Movimientos**”: toma desde la clase entidad Cuenta los movimientos realizados en la cuenta y realiza un listado para ser emitido por el cajero.
- “**Cliente**”: es una entidad que almacena información persistente, referente a las claves junto a los datos del usuario de cada cuenta.
- “**Cuenta**”: es una entidad que almacena información persistente, referente a todos los movimientos de cada cuenta.

PARTE B: OPTIMIZACIÓN DEL DIAGRAMA

TIEMPO INSUMIDO EN LEER ESTA PARTE:

Se solicita *optimizar*, en base a tu experiencia y criterio personal, el Diagrama de Clases presentado en Parte A. Puedes copiar y pegar la simbología a emplear (*desagrupa los elementos, trabaja en una página en blanco*):



PARTE C: CUESTIONARIO

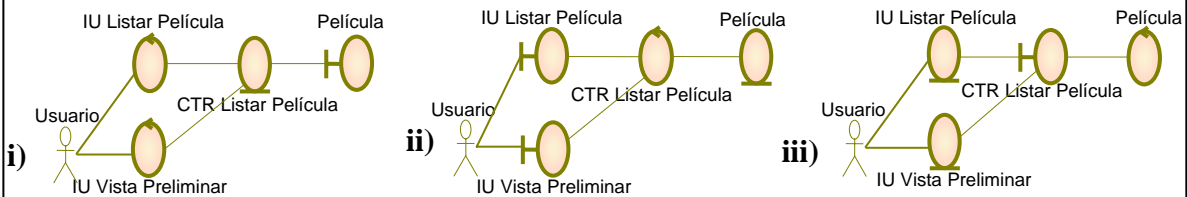
Cuando corresponda, marca con **X** la respuesta elegida.

- 1) **¿Qué tiempo te insumió entender el ej. de la PARTE A para el Diagrama de Clases?**
- 2) **¿Tienes conocimientos previos o experiencia con el PU?**
 - i. Poco
 - ii. Mucho
- 3) **¿Has comprendido el concepto del PU presentado en la PARTE A?**
 - i. Se comprendió en su totalidad el concepto del PU.....
 - ii. No se entendieron sus símbolos
 - iii. Es complejo y poco claro en su totalidad
 - iv. No lo entendí en absoluto
 - v. Otra opinión (escribe abajo)
- 4) **Elige el Diagrama de Clases que representan las clases siguientes:**

IU Listar Películas: representa la interfaz gráfica que permite al usuario ver el listado de todas las películas almacenadas en la entidad Películas.

CRT Listar Película: toma desde la clase IU Listar Películas el tipo de listado a realizar, solicitando a la clase entidad Película todos los datos relacionados a las mismas para mostrarlos posteriormente por pantalla o impresora.

Película: es una entidad que almacena información persistente, referente a las películas; devuelve los datos relacionados a las películas, solicitados por la clase de control.



- 5) **Según tu criterio, el diagrama presentado anteriormente es:** (puedes marcar más de una)
 - i. Correcto
 - ii. Incorrecto
 - iii. Completo
 - iv. Incompleto
 - v. Podría mejorarse
- 6) **En la PARTE B, qué tiempo te insumió la construcción del Diagrama de Clases?**
- 7) **¿Cuáles han sido las dificultades encontradas al optimizar el Diagrama de Clases de la PARTE B?**
 - i. Ninguna
 - ii. Dificultad con la notación
 - iii. En el entendimiento de las relaciones
 - iv. Otra opinión (escribe abajo)
- 8) **¿Crees que la propuesta del PU es suficiente para la construcción de modelos durante la etapa del análisis?**
 - i. Si
 - ii. No

¿Porqué? (escribe abajo)

9) **El Diagrama de Clases obtenido en la PARTE A, es un modelo que:**

- i. Representa totalmente la realidad modelada
- ii. No representación la realidad modelada
- iii. Representa parcialmente la realidad modelada

10) **En el momento de realizar el análisis de un proyecto de software:**

- iii. Seleccionas una método que guíe el proceso
- iv. Utilizas solamente tu intuición y/o experiencia previa

11) **Comentarios**

ENCUESTA :

PROCESO UNIFICADO (DIAGRAMA DE COLABORACIÓN)

FINALIDAD: la presente encuesta forma parte de la validación empírica del Método de Análisis derivado de una Explicación Textual, correspondiente al Trabajo de Graduación denominado “Método de Análisis derivado de una Especificación Textual. Su Validación Empírica” para optar por el Título de Licenciado en Sistemas de Información de la UNSE. Los resultados de las encuestas servirán para la aceptación o refutación de la hipótesis planteada.

La encuesta se estructura en tres partes:

- **PARTE A:** Presentación conceptual del Proceso Unificado (PU) y ejemplificación de un caso típico.
- **PARTE B:** Resolver la consigna.
- **PARTE C:** Responder el cuestionario.

DESTINATARIOS: Alumnos de la carrera Licenciatura en Sistemas de Información y carreras relacionadas.




Nota: Por favor, registrar el tiempo reloj que insume la realización de la encuesta (PARTE A y PARTE B) de la presente.

PARTE A: CONCEPTO DEL PU Y EJEMPLO

TIEMPO INSUMIDO EN LEER ESTA PARTE:

El PU es un Proceso de Desarrollo de Software basado en UML. Es iterativo, centrado en la arquitectura y dirigido por los casos de uso. La etapa de análisis, flujo de trabajo del PU, refina y estructura los requisitos del usuario, representándolos a través del **DIAGRAMA DE CLASES** (*modeliza aspectos estáticos*) y el **DIAGRAMA DE COLABORACIÓN** (*modeliza aspectos dinámicos*)

Para representar los diagramas del PU se emplea:

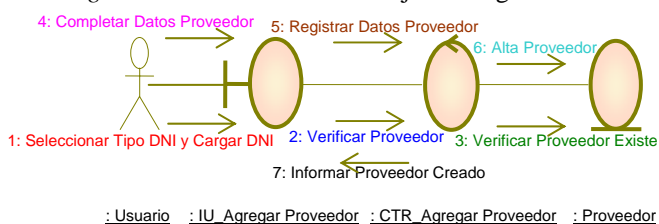
4.  para modelizar la **CLASE DE INTERFAZ (IU)**, que conecta al usuario con el sistema.
5.  para modelizar la **CLASE DE CONTROL (CTR)**, que coordina las acciones y deriva el trabajo.
6.  para modelizar la **CLASE DE ENTIDAD**, que almacena información que persistente en el sistema.

EJEMPLO

El diagrama estudiado representa a una empresa encargada de la venta de vehículos, se considera particularmente los proveedores de la misma. En el escenario 1, el proveedor no pertenece a la empresa de vehículos (Alta de Proveedor).

Por favor, primero entender el diagrama ejemplo antes de leer su explicación textual.

El Diagrama de Colaboración del ej. es el siguiente:



CLASES:

- “IU Agregar Proveedor”: solicita al usuario los datos correspondientes al proveedor que se quiere ingresar.
- “CTR Agregar Proveedor”: toma desde la clase IU Agregar Proveedor los datos del proveedor y los transfiere a la clase entidad Proveedor .
- “Proveedor”: toma los datos del nuevo proveedor transferidos de la clase control y se encarga de almacenarlos por largo tiempo en el sistema.

EXPLICACIÓN DEL ESCENARIO 1

El usuario solicita al sistema agregar un nuevo proveedor ingresando el Tipo y Número de documento (1).

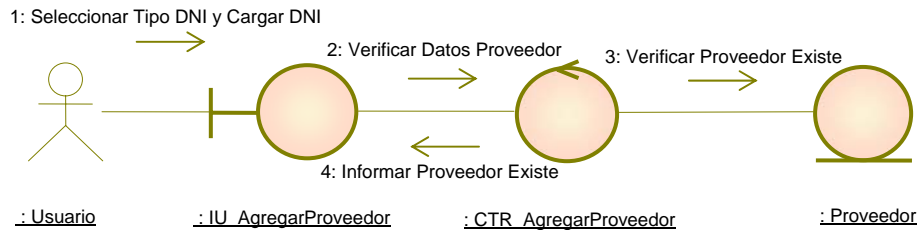
La interfaz IU Agregar Proveedor envía los datos a la clase de control CTR Agregar Proveedor, y esta verifica la existencia de dichos datos (2,3) en la clase entidad Proveedor. El usuario ingresa el resto de los datos del proveedor a registrar (4) y la clase de control CTR Agregar Proveedor se encarga de controlar

PARTE B: EXPLICACIÓN DEL DIAGRAMA

TIEMPO INSUMIDO EN LEER ESTA PARTE:

El siguiente es un Diagrama de Colaboración sobre un proveedor perteneciente a la empresa. En el escenario 2, se modeliza un proveedor que pertenece a la empresa de vehículos (Proveedor Existente).

Explica lo que representa, de acuerdo a tu comprensión del diagrama.



EXPLICACIÓN DEL ESCENARIO 2

PARTE C: CUESTIONARIO

Cuando corresponda, marca con **X** la respuesta elegida.

1) **¿Qué tiempo te insumió entender el ej. de la PARTE A para el Diagrama de Colaboración?**
¿Qué tiempo te insumió la explicación, en la PARTE B, del Diagrama de Colaboración? ...

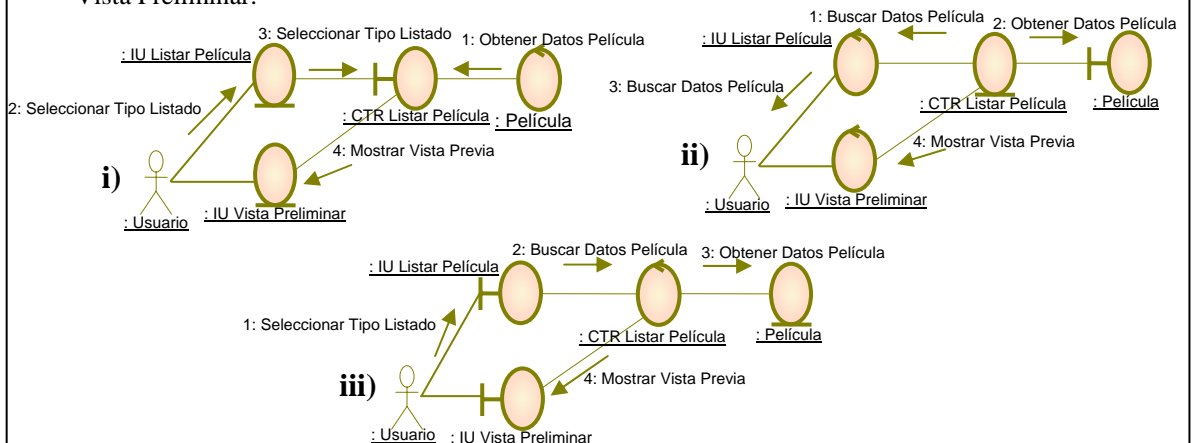
- 2) **¿Has comprendido el concepto del PU presentado en la PARTE A?**
- v. Se comprendió en su totalidad el concepto del PU.....
 - vi. No se entendieron sus símbolos
 - vii. Es complejo y poco claro en su totalidad
 - viii. No lo entendí en absoluto
 - ix. Otra opinión (escribe abajo)

3) **¿Fue más comprensible el Diagrama de Colaboración de la PARTE A una vez leída la explicación textual del mismo?**

- iii. Si
- iv. No

4) **Elige el Diagrama de Colaboración que representa la siguiente explicación:**

“El usuario solicita al sistema listar las películas alquiladas, seleccionando el tipo de listado que quiere realizar. Una vez que la opción es registrada de la clase interfaz IU Listar Películas se realiza la búsqueda de las películas que están alquiladas a la clase CTR Listar Película, se obtienen los datos de las películas de la clase entidad Película y se realiza una vista previa por medio de la clase interfaz IU Vista Preliminar.”



- 5) **El Diagrama de Colaboración obtenido en la PARTE A, es un modelo que:**
- i. Representa totalmente la realidad modelada
 - ii. No representación la realidad modelada
 - iii. Representa parcialmente la realidad modelada
- 6) **¿Cuáles han sido las dificultades encontradas al explicar el Diagrama de Colaboración de la PARTE B?**
- vi. Ninguna
 - vii. Poca, el diagrama es poco complicado
 - viii. Dificultad con la notación
 - ix. En el entendimiento de las relaciones
 - x. Otras (escribe abajo)

7) **Comentarios**

ANEXO D

OPINIONES DE LOS SUJETOS ENCUESTADOS

SOBRE MATE:

“Parece un método sencillo y claro para modelar sistemas, porque tiene en cuenta los aspectos relevantes, dejando de lado aquellos innecesarios, que tal vez complicarían la representación de sistemas.”

“Es interesante el método para el desarrollo de sistemas, ya que ofrece una alternativa para representar su comportamiento y aspectos estáticos, haciendo un análisis de la descripción textual. Esto me parece un paso que muchas veces se deja de lado, debido a que se trata de añadir en la modelización de sistemas, detalles redundantes que lo único que hacen es complicar la construcción del modelo y por consiguiente que este resulte poco claro.”

“Pienso que es una manera muy sencilla de representar el comportamiento de un sistema y que resultaría muy útil a la hora de representar sistemas ya que hay personas sin estudios en el tema que se dedican a hacerlos y no realizan ninguna documentación ni representación de los mismos, por lo cual es difícil entenderlos cuando otra persona, con estudios o sin ellos, intenta leer el código de dicho sistema. Este sería un método útil para enseñar por su sencillez y claridad de representación.”

“Muy interesante el método y tanto el modelo de objetos como el modelo de comportamiento son fáciles de entender, le podría servir a muchas personas que nunca han estudiado sistemas.”

“Creo que el método puede ser provechoso (previo entrenamiento en su interpretación) por que tiende a mejorar la transmisión y comprensión de la información. Por ende es un camino mas corto a la eficiencia.”

“Desde mi punto de vista, el método es sencillo de aplicar y fácil de entender.”

“Con el método es mas fácil para entender las relaciones en un sistema, y mejor ya al entender las relaciones se puede modelar mas fácilmente la realidad.”

“Me gusta mucho el modelo presentado, gracias por mostrármelo.”

“El modelo de objetos ayuda a la mejor comprensión del problema y a su rápida solución para la creación del sistema.”

“Respecto al análisis de las oraciones del modelo de comportamiento resulto difícil la comprensión por que es un análisis fuera de lo común para mi.”

“El modelo de objetos del método es muy completo y muy fácil de entender, el modelo de comportamiento me resulta un poco incompleto.”

“El modelo de comportamiento me resultó un poquito mas complicado para entender, no estoy segura de cómo utilizar la notación, creo que una consulta me aclararía las dudas. Pero aun así no me parece que el método sea difícil.”

SOBRE PU:

“Me parece que son muy claro los diagramas.”

“Pienso que no es complejo, solo que tendría que profundizar mas en su teoría, y así poder entender todo el PU.”

“Leí varias veces. Leer varias veces el primer escenario, me hizo comprender rápido el segundo escenario para realizar la parte B. Me parece que este método es una alternativa interesante de hacer modelos. No me pareció difícil.”

“Es fácil entenderlo y el problema se hace mas sencillo explicado y representado con este método.”

“Por lo que entiendo es lo más adecuado para dar las soluciones en esta etapa, no obstante al modelar simplemente estamos reduciendo la complejidad que nos presenta la realidad, el resultado siempre dependerá del grado de abstracción que cada uno internalice al momento de afrontar la realización de la misma.”

“Ninguna complicación, a pesar de que no tengo conocimientos sólidos, fue muy clara esta parte.”

“El Diagrama de Clases fue leída después del Diagrama de colaboración, la idea es mas clara y por lo tanto insume menos tiempo en la comprensión del mismo.”

“Este modelo no resulta suficiente para modelizar una situación.”