

IDENTIFICACION DE REQUISITOS: UN ENFOQUE BASADO EN TAXONOMIA VERBAL

IDENTIFICATION OF REQUIREMENTS: A FOCUS BASED ON A VERB TAXONOMY

RICARDO A. GACITÚA B.¹

RESUMEN

Uno de los principales problemas a los cuales se enfrenta un diseñador de software es la distorsión que se puede producir entre las etapas del desarrollo que componen un ciclo de vida. Un ejemplo de ello lo constituye la identificación de requisitos y la posterior especificación de éstos. En particular, en este tema, se han realizado grandes avances. Sin embargo, es en la etapa de identificación de requisitos en la que el cliente intenta expresar su concepto, a veces confuso de funcionalidad y rendimiento, y transformarlos a elementos concretos. El desarrollador, por su parte, actúa como un interrogador y consultor, siendo visto como la persona que resuelve problemas. Es en esta etapa en la cual abundan las malas interpretaciones o falta de información, y en la que los supuestos y paradigmas intentan suplir las carencias que no han sido expresadas explícitamente, lo cual provoca grandes distorsiones en la representación de la realidad. El presente trabajo muestra una técnica de identificación sistemática de requisitos funcionales, basada en una taxonomía de verbos, en la que se pretende, a partir de identificar el nivel conceptual de un verbo (por ejemplo: *administrar* es de nivel superior que *controlar* y, controlar es de nivel superior que *listar*), identificar objetos elementales –definidos en esta propuesta como componentes individuales de software previamente implementados asociados al nivel más bajo de la taxonomía de verbos– tales como: tipos de reportes pre hechos, pantallas de consulta, mantenedores de tablas, entre otros. En este sentido, y a partir de la identificación de los niveles de verbos, éstos se traducen finalmente a un conjunto de objetos elementales previamente implementados, lo cual trae como beneficios entre otros: 1) una mejor estimación de esfuerzo, en la etapa de planificación, de un proyecto de desarrollo; 2) un aumento sustancial del reuso de componentes de software u objetos elementales, y 3) un aumento en el nivel de trazabilidad o seguimiento sobre los requisitos y su posterior implementación.

PALABRAS CLAVES: Ingeniería de software, Métodos y técnicas, Requisitos.

1. INTRODUCCION

En cada uno de los proyectos de desarrollo de software, la experiencia nos muestra que es esencial para asegurar el éxito una com-

presión total y acabada de los requisitos de software. Sin embargo, es frecuente encontrar clientes y usuarios disgustados y frustrados a consecuencia de que el software no hace ni responde a los requisitos básicos que se esperaba que tuviera. En este sentido, y tal como lo afirma Pressman (Pressman, 1998):

¹Depto. Sistemas de Información, Facultad de Ciencias Empresariales, Universidad del Bío-Bío, Avenida Collao 1202, Concepción-Chile, E-mail: rgacitua@ubiobio.cl

El análisis y la especificación de los requisitos puede parecer una tarea relativamente sencilla, pero las apariencias engañan. El contenido de comunicación es muy denso. Abundan las ocasiones para las malas interpretaciones o falta de información. Es muy probable que haya ambigüedad. El dilema al que se enfrenta el ingeniero de software puede entenderse muy bien repitiendo la famosa frase de un cliente anónimo: "Sé que cree que entendió lo que piensa que dije, pero no estoy seguro de que se dé cuenta de que lo que escuchó no es lo que yo quise decir".

La anterior frase pone de manifiesto lo importante de reducir la ambigüedad presente en la comunicación a objeto de que la interpretación, y posterior especificación, sea lo más ajustada a la realidad.

Muchas son las estrategias propuestas para reducir esta brecha. Grupos de expertos, tal como Gause y Weinberg (Gause *et al.*, 1989) proponen un conjunto de preguntas pre-establecidas tendientes a sistematizar el acercamiento a los verdaderos requisitos. Por otra parte, técnicas como TFEA (Técnicas para facilitar las especificaciones de la aplicación, ejemplo de estos son JAD, y The Method) (Zahniser M., 1990) pretenden, a partir de la formación de un equipo conjunto de clientes y desarrolladores, sistematizar un trabajo con la finalidad de establecer un conjunto preliminar de requisitos.

2. ESTABLECIENDO UNA PROPUESTA

2.1. Antecedentes preliminares

Varios son los esfuerzos que se han realizado en el ámbito de la determinación de requisitos utilizando lenguaje natural, ejemplo de ello es posible encontrar en Hars *et al.* (1999 a y b) y Wilson *et al.*, 1999. En este sentido, es posible afirmar que una de las principales líneas de investigación está centrada en la equivalencia que existe entre: verbos y procedimientos, sustantivos y entidades, etc. Otras líneas, en cambio, se centran en el traspaso desde una especificación o descripción de un problema en lenguaje natural a una representación gráfica, tal cual se muestra por ejemplo en la Figura 1, que ha sido extraída de una presentación de un proyecto denominado Sistema ACAPULCO de los autores Hars y Marchewka. En dicho proyecto se logra, a partir de una frase escrita en lenguaje natural, una representación gráfica utilizando condiciones y secuencias. Se plantea que con esta interpretación gráfica se permite una mejor especificación del problema y facilita la validación por parte del usuario.

Por otro lado, la mayor parte de las propuestas publicadas en la literatura basan su desarrollo en la estructura de verbos y sintaxis del idioma inglés, lo que no permite ser utilizado directamente con otros idiomas, tal como el español, por ejemplo, por tener estructuras distintas, variaciones en los sinónimos de verbos, entre otros aspectos. Por estas razones, y atendiendo al principal problema de la identificación y especificación de requisitos, no sólo de la interpretación de éste sino del interés por traducirlo en productos de software concretos, es que se propone un enfoque alternativo y diferente.

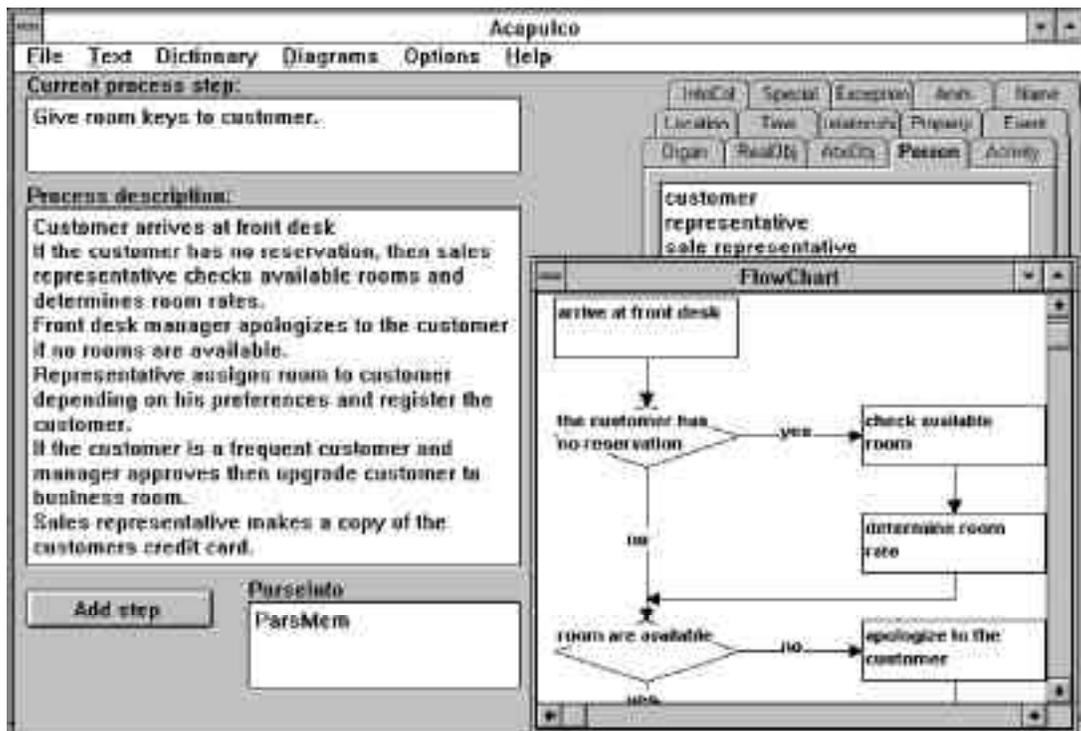


Figura 1. Pantalla de una herramienta generada en el proyecto Sistema ACAPULCO.

2.2. Proponiendo un nuevo enfoque del problema

A diferencia del caso anterior, la propuesta descrita a continuación ha surgido y ha sido validada empíricamente en proyectos de desarrollo de software administrativos que se basan precisamente en el elemento que aporta la ambigüedad: el lenguaje natural. Son muchas las experiencias en las cuales los clientes (no precisamente expertos en temas de informática, sino expertos en temas de procesos de negocios o en otros ámbitos) intentan imponer exigencias y puntos de vista subjetivos utilizando su lenguaje natural, ambiguo, genérico, no preciso, entre otras características. Lo anterior permite destacar al *lenguaje de comunicación* como un primer elemento importante y que genera serios inconvenientes. Mientras el desarrollador, habituado a su lenguaje técnico de: objetos, entidades, relaciones, procesos, RAM, pro-

tolos, cliente/servidor, base de datos, etc., debe intentar acercarse e interpretar en su propio lenguaje lo planteado por su cliente, éste intenta explicar con sus propios conceptos y términos de su área de especialización sus necesidades de información. Lamentablemente, es frecuente encontrar desarrolladores que intentan explicar elementos de un software en un lenguaje técnico, muy ininteligible para el cliente. En este sentido, el autor plantea una primera premisa que no es el cliente quien tiene que dominar el lenguaje técnico, sino que es el diseñador que tiene que interpretar el lenguaje del cliente. Parece raro hablar que tanto el cliente como el diseñador utilizan lenguajes distintos¹. Sin embargo, el esfuerzo de ambos debe centrar-

¹Es importante destacar en este caso, que lo único distinto en el lenguaje es el conjunto de términos y siglas utilizados asociados a un área específica, los cuales sí provocan confusiones, pues se pretende que el "otro" entienda intuitivamente el conjunto de términos y siglas propios.

se sólo en utilizar un subconjunto común del lenguaje natural. Esto es, identificar:

- Las acciones que pretenden ser realizadas y sus restricciones.
- Los resultados que se desean obtener de un software, en forma de fuentes de información (reportes, gráficos, listados, indicador, acción sobre dispositivos, consultas, etc.).

Pues bien, si este subconjunto del lenguaje natural constituye una porción común entre clientes y diseñadores, es este subconjunto el que debería permitir –al menos en un gran porcentaje– establecer las *acciones y resultados esperados*, con mayor certeza. Considerando lo anterior, a objeto de formalizar dicho subconjunto, se propone una primera clasificación de componentes, tal cual se muestra en la Tabla 1:

Tabla 1. Analogía entre lenguaje y el modelo a utilizar.

Lenguaje	Técnica	Explicación
Acción	Verbos	En la forma de los verbos, y su definición, que componen el lenguaje natural.
Resultados esperados	Objetos elementales	En la forma de Listados, Pantallas de Consultas, Gráficos, etc. Pre-hechos

2.3. Identificando jerarquías conceptuales

Si bien la relación anterior es fundamental para la definición de la técnica, no es menos cierto que al hablar de verbos estamos ha-

blando de un concepto que indica acción. Por su parte, las acciones pueden implicar una jerarquía de acciones. Es decir, una acción implica otras acciones. Por ejemplo:

Nivel Superior		Nivel Inferior
ADMINISTRAR	implica...	PLANIFICAR DOTAR ORGANIZAR DIRIGIR CONTROLAR

Por lo tanto, cualquier petición podría definirse o expresarse como una jerarquía de acciones o verbos que lo conforman. En este sentido, cuando un cliente dice: “Yo quiero *administrar* un recurso”, se deduce que ese requerimiento tiene un alcance muy superior, si sólo dijese “Yo, quiero *controlar* un recurso”.

La jerarquía propuesta tiene sentido sólo si se vislumbra la posibilidad de *sistematizar* las peticiones y/o requisitos soportados so-

bre un conjunto de acciones pre-definidas y asociadas a las capacidades que se obtienen de la tecnología. Por ejemplo, no deberían considerarse las acciones de presentir, intuir, entre otras, pues no son factibles de implementar utilizando tecnología computacional, aún. Como fundamento de lo anterior, se utilizará como base una taxonomía de verbos propuesta por Benjamin Bloom, muy utilizada en el área de educación (Bloom, 1964). En dicha taxonomía se establece que

para lograr que una persona logre un correcto aprendizaje deben definirse tres diferentes áreas de conocimiento del ser humano:

- El área *cognoscitiva*: que incluye aquellos objetivos que se refieren al recuerdo o reconocimiento de informaciones y al desarrollo de habilidades y capacidades intelectuales más complejas.
- El área *afectiva*: que incluye aquellos objetivos que incluyen proposiciones de resultados de aprendizajes, en que las manifestaciones conductuales predominantes, en los alumnos, son: de intereses, emociones, actitudes, valores, juicios y formas de adaptación personal o social.
- El área *psicomotora*: que incluye todos aquellos objetivos educacionales que se proponen, como resultados concretos de una acción educativa, la manifestación –por parte del alumno– de conductas en las cuales el aspecto predominante de ellas

son habilidades y destrezas neuro-musculares o físicas.

Por lo tanto, de acuerdo a la clasificación anterior, el área relacionada con el tratamiento de la información y la identificación de requisitos corresponde al área cognoscitiva, pues es lo que se asocia al reconocimiento de información. Es obvio, entonces, que se descarte el área afectiva y el área psicomotora, pues no se asocian a la identificación de requisitos o de resultados de información, que constituyen la base del desarrollo de software.

2.4. Detallando el área cognoscitiva

Según la taxonomía de Bloom, el área cognoscitiva contiene 6 niveles de abstracción en los cuales es posible clasificar cualquier objetivo a lograr, tal cual se muestra en la Tabla 2.

Tabla 2. Niveles del Area Cognoscitiva.

Nivel	Area Cognoscitiva	Ejemplo
6	Evaluación	Criticar
5	Síntesis	Proponer
4	Análisis	Analizar
3	Aplicación	Realizar
2	Comprensión	Renombrar
1	Conocimiento	Identificar

La Tabla 2 expone claramente el hecho de que para alcanzar el nivel de Evaluación es necesario haber alcanzado antes los 5 niveles anteriores. Por lo tanto, utilizando el lenguaje natural es posible jerarquizar, en función del nivel de conocimiento que se requiera, los diferentes verbos y clasificarlos lógicamente para un objetivo educativo o de detección de necesidades (identificación de requisitos). Según lo anterior, las necesida-

des se deben captar según el punto de vista del cliente y/o usuario(s) y luego clasificarlos para facilitar su posterior tratamiento como requisitos. En este sentido, la propuesta se basa en un enfoque algo similar en estructura, pero de contenido muy distinto al propuesto por Fernando Flores (Flores, 1991) la que se basa principalmente en una taxonomía basada en peticiones, compromisos y ofertas.

Se debe destacar que al hablar de taxo-

nomía de verbos nos referimos a una lista exhaustiva de verbos clasificados por cada

nivel, propuesta por Benjamin Bloom, ejemplo de ello se muestra Tabla 3:

Tabla 3. Primera categoría: Nivel de Conocimiento.

Aparear	Declarar	Formular	Relacionar
Aseverar	Definir	Identificar	Relatar
Asociar	Denominar	Indicar	Renombrar
Cambiar	Describir	Inventariar	Reordenar
Casar	Designar	Juntar	Repetir
Catalogar	Diferenciar	Listar	Reproducir
Categorizar	Discriminar entre	Localizar	Rotular
Clasificar	Distinguir entre	Llamar	Seleccionar
Citar	Encerrar con; en	Marcar	Señalar
Combinar	Enumerar	Nombrar	Separar
Computar	Enunciar	Numerar	Subrayar
Consignar	Esbozar	Ordenar	Tasar
Contar	Escribir	Reagrupar	Unir

Las acciones anteriores implican la memorización, el recuerdo o reproducción, en formas similares a aquellas en que fueron recibidas y aprendidas. Estas informaciones pueden estar conformadas por y desde simples datos aislados, hasta cuerpos de antecedentes estructurados en principios o teorías.

2.5. Los requisitos tratados como objetivos de un cliente

Una de las actividades más relevantes en la construcción de un software es el hecho de obtener las necesidades del cliente de forma clara y en un lenguaje (representación) comprensible y comunicable. Por lo tanto, nuestro objetivo es especificar requisitos que además permitan ser monitoreados (trazables). Esto es, realizar el seguimiento desde que las necesidades se plantean hasta que se especifican como requisitos funcionales. Si se asume que los requisitos son los objetivos que el cliente quiere que el software logre,

entonces el problema de la identificación se transforma en la asociación de necesidades a objetivos (requisitos - acciones) de software que sean alcanzables y monitoreables. Se debe destacar la enorme similitud entre los objetivos de sistemas y los objetivos educacionales que plantea Bloom, los cuales, al igual que los requisitos, son difíciles de identificar y cualquier ambigüedad impedirá que el aprendizaje de cierta información esté de acuerdo a la forma en que planeó inicialmente.

2.6. Analogía de taxonomías

Si se toman los conjuntos de verbos clasificados por Bloom para cada nivel, es posible establecer una analogía directa con los niveles organizacionales y con los objetivos de desarrollo (taxonomía de requisitos) de sistemas. Para ello, se presenta en la Tabla 4 la analogía definida:

Tabla 4. Analogía entre taxonomías.

Nivel de abstracción organizacional	Taxonomía de requisitos	Taxonomía de verbos
Visión - Misión	Nivel de Negocios	Nivel de Evaluación
Estrategias	Nivel de Función	Nivel de Síntesis
Metas	Nivel de Sistemas	Nivel de Análisis
Actividades	Nivel de Proceso	Nivel de Aplicación
Tareas	Nivel de Módulo	Nivel de Comprensión
Item de Tarea	Nivel de Instrucción Elemental	Nivel de Conocimiento

Considerando lo anterior, si se desea clasificar un requerimiento en lenguaje natural se analiza en primer lugar la taxonomía de verbos, la que presenta un conjunto de verbos clasificados por nivel. Si existe duda en el nivel del verbo se utiliza la taxonomía de requisitos, que es equivalente en lo que

se refiere a conjunto de verbos pero es más específica respecto a niveles de sistema. En último término, se asocian a niveles de abstracción de la organización. Para una mayor claridad del requisito, éste se expresa finalmente utilizando un formato de definición que posee la siguiente estructura:

$$\text{Requisito Funcional} = \text{Verbo} + \{ \text{objeto} \} + \{ \text{Restricción o Parámetro} \}$$

Ejemplo:

Requisito1 = LISTAR + (Cliente) + (Ingreso > 10/11/2001)
 Requisito2 = IMPRIMIR +(proveedor, cliente)+ ()

Es necesario recordar que una taxonomía no es sólo una estructura jerárquica, es también una estructura que relaciona a los dis-

tintos elementos (verbos) con otros de diferente nivel de abstracción en forma lógica. Por ejemplo: OPERAR con un CLIENTE.

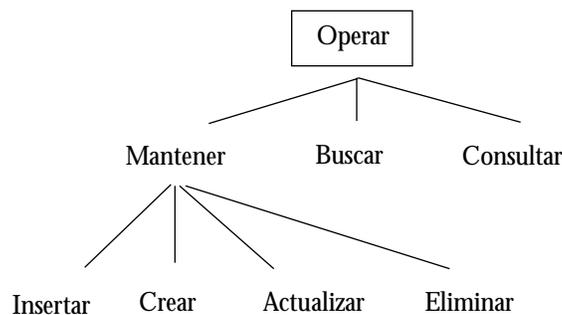


Figura 2. Jerarquía de requerimiento: operar con un cliente.

En este caso (Figura 2), se tiene que el hecho de operar con un cliente implica otras acciones que pudieran estar implícitas y que corresponde a acciones de diferentes niveles, por ejemplo: Operar con un Cliente pertenece al tercer nivel de Aplicación (ver Tabla 2), a diferencia de las acciones de Insertar, Crear, Actualizar y Eliminar, que corresponden al nivel 2 de Comprensión. Lo anterior implica que el hecho que se tenga que Operar con un cliente, implica inmediatamente que deben existir los niveles anteriores.

Tal como se dijo anteriormente, los Objetos Elementales o Atómicos son acciones del nivel más bajo que se implementan como productos de software elementales. Lo anterior, es de enorme importancia para el

modelo taxonómico de requisitos, ya que permite monitorear los requisitos hasta componentes de software pre-hechos. Por lo tanto, es posible apreciar que, al fusionar los enfoques de requisitos, verbos y niveles de abstracción en una sola taxonomía, los requisitos de un cliente se pueden estructurar jerárquicamente. Esta estructuración es lógica, completa, y real, respecto a las expectativas del negocio, y demuestra que los niveles de abstracción utilizados concuerdan con la descomposición natural de los requisitos.

En este sentido, la finalidad del modelo taxonómico de requisitos es establecer la jerarquía de Objetivos, tal como se muestra en la siguiente figura:

Nivel conceptual 1	Nivel conceptual 2	Nivel conceptual (n-1)	Nivel conceptual (n)	Nivel elemental
ADMINISTRAR	Planificar				Tipo Reporte 1 Tipo Reporte 2 Tipo Reporte n
	Organizar				Tipo Consulta 1 Tipo Consulta 2 Tipo Consulta i
	Dirigir				Tipo Listado 1 Tipo Listado 2 Tipo Listado m
	Controlar	Listar			Tipo Gráfico 1 Tipo Gráfico 2 Tipo Gráfico k
		Consultar	Calcular		Tipo Mantenimiento 1 Tipo Mantenimiento 2 Tipo Mantenimiento z
		Imprimir			Tipo Indicador 1 Tipo Indicador 2 Tipo Indicador p
		Dotar			

Figura 3. Ejemplo de Jerarquía de Verbos.

En la Figura 3 es posible apreciar que, a partir de la sola interpretación de lo que realmente se tiene como objetivo, se intenta establecer un “acuerdo” y una visión rápida respecto de la verdadera magnitud de un software. Lo que se espera, entonces, es que a partir de una rápida jerarquización verbal se pueda determinar, entre otros, la cantidad de componentes elementales involucrados. Es decir, que los requisitos de un software se satisfacen finalmente por una cantidad “x” de pantallas de consultas, una cantidad “y” de listados, una cantidad “t” de gráficos, etc, que pudieran ser pre-definidos.

Como ejemplo, cuando un cliente plantea textualmente (caso real):

... me interesa administrar la maquinaria de la sección de producción de modo tal que podamos obtener un listado de los repuestos utilizados y, posteriormente, realizar un control respecto de los gastos de mantención en que se incurre.

Es posible deducir que la palabra administración no se utiliza en su real dimensión, puesto que no se planifica, organiza, dirige ni dota, sólo se realiza un control sobre el recurso. Por otra parte, los objetivos se plantean en niveles conceptuales distintos y se actúa sobre recursos distintos. El ejemplo anterior se puede representar como sigue:

Nivel conceptual 1	Nivel conceptual 2	Nivel conceptual 3
ADMINISTRAR	Controlar (Maquinaria)	Listar (Repuestos)

Figura 4. Niveles conceptuales del ejercicio.

3. FORMA DE USO Y PROYECCIONES

La técnica basada en taxonomía de requisitos ha sido aplicada inicialmente en la etapa de identificación de requisitos. Se puede afirmar que esta técnica permite acelerar la identificación de requisitos previa identificación de los objetivos del cliente y posterior jerarquización de ellos. Además, mejora la capacidad para estimar la magnitud o tamaño del software, así como el nivel de reusabilidad que se puede aplicar a la posterior construcción. Del mismo modo, se

ha comprobado que la validación, por parte del usuario, ha sido mucho más simple y ha permitido que él dimensione en forma más precisa tamaño del software a construir (medido en hora- hombre). Para usar esta técnica, los principales clientes de las áreas afectadas (ejecutivo, operador, analista, programador, etc.) completarán la documentación de las necesidades de la organización haciendo uso de *cuestionarios pre-definidos* y, sobre esa base, se acordará una jerarquización de objetivos traducidos a jerarquía de verbos. La idea es entender al cliente usando su propio lenguaje mediante el apoyo de

esta técnica. Por otro lado, la técnica facilita el reconocimiento de los requisitos redundantes mediante su formato de entrada (verbo, objeto, restricción) a través del desglose de los requisitos en objetos atómicos ya que, si existe algún requerimiento redundante,

obligatoriamente existirán objetos atómicos repetidos. Un ejemplo de definición de requisitos, realizado a un aserradero en la ciudad de Concepción, Chile, se presenta a continuación:

Tabla 5. Ejemplo de Categoría de Requisitos.

Identificador	Verbo	Resultado	Objetos	Condición 1
REQ-OP-PROV-1	Ingresar	Nuevos datos	Proveedor	-
REQ-OP-INS-5	Listar	Datos	Insumos	<= Stock Mínimo
REQ-TA-CLI-10	Consultar	Datos	Clientes	= Key

En este ejemplo existen criterios que corresponde a la clasificación por nivel jerárquico (operativo, táctico, estratégico), así

como la utilización del formato de definición, anteriormente presentado:

Identificador - verbo - resultado - objetos - condiciones

En lo que se refiere a la determinación de importancia de los requisitos, el modelo es jerárquico y permite estipular diversos niveles de abstracción de verbos. Para este efecto, existen componentes de software ele-

mentales pre-hechos, tales como: listados con formatos definidos, pantallas de consulta, etc., que corresponde al nivel más bajo o elemental. En la Figura 5 se presenta una pantalla de un sistema desarrollado en el

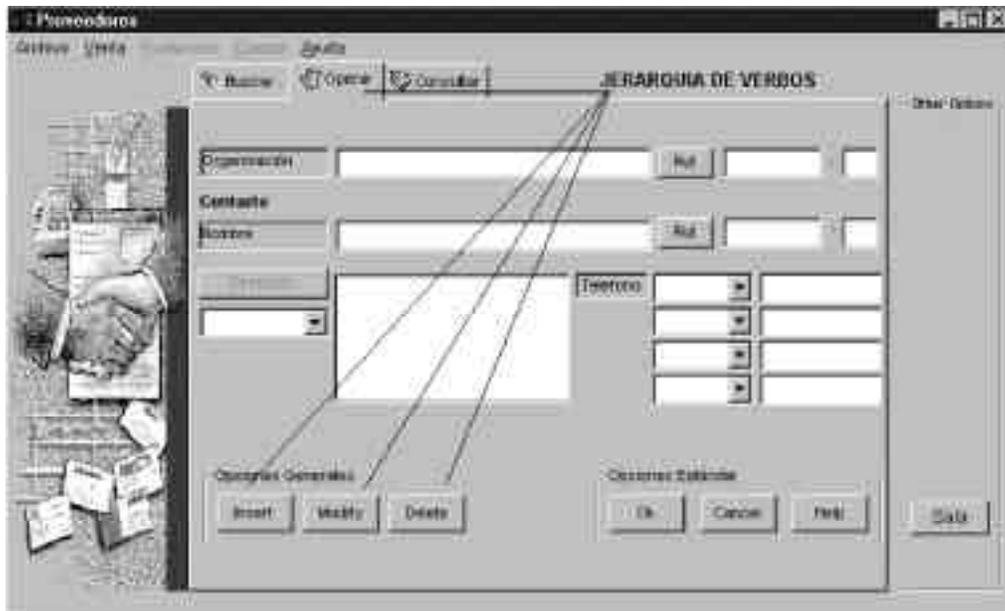


Figura 5. Pantalla de sistema construido.

cual se ha utilizado en el esquema de jerarquía de verbos. En ella es posible identificar tres grandes acciones: operar, buscar y consultar.

El siguiente paso de este trabajo es construir un ambiente de apoyo a la identificación de requisitos en el cual se jerarquicen los objetivos de un software, se especifiquen vía la taxonomía de verbos, se asocien a los niveles atómicos y se realice posteriormente un trazo de su implementación.

4. CONCLUSIONES

Muchas de las dificultades que se presentan en la fase de identificación de requisitos se pueden evitar mejorando el lenguaje de comunicación. En este sentido, el hecho de proponer una técnica de identificación de requisitos basada en una taxonomía y jerarquía de verbos, posibilita un mejor análisis y, posteriormente, un mejor seguimiento de ellos. El presente trabajo sienta las bases conceptuales para la construcción de una herramienta de soporte para la identificación de requisitos que permitirá satisfacer los siguientes objetivos:

1. Sistematizar y mejorar la comunicación entre los clientes y desarrolladores, mediante la utilización de un enfoque gráfico que determine acciones y resultados, en forma precisa. En este sentido, es preciso establecer que la propuesta desarrollada por Bloom es ampliamente aceptada y precisa para el idioma español para el tratamiento de información. En aquellos casos en que se utilice alguna jerga especial, se considera necesario realizar un trabajo de interpreta-

ción que permita definir los requisitos sólo en el contexto definido por la taxonomía de verbos de Bloom.

2. Permitir la traza sobre de los requisitos y mejorar el control que se realiza sobre éstos.

3. Facilitar el re-uso de componentes, dado que los objetos atómicos o elementales son componentes de software pre-hechos, probados y validados.

4. Mejorar la precisión en la estimación del esfuerzo de desarrollo. Lo anterior, dado que al obtener la jerarquía de objetivos y verbos representada en un diagrama, se plantea como hipótesis que será posible determinar por la extensión de dicho diagrama una aproximación del esfuerzo de desarrollo de todos los componentes que componen finalmente el sistema.

Por lo pronto, el uso de esta técnica ha permitido reducir la ambigüedad producida por el lenguaje. En este mismo sentido, los clientes han visto sistemáticamente cómo se han traducido sus necesidades a verbos que indican acciones concretas, lo que permite que ellos mismos sean capaces de “tasar” la magnitud y la complejidad de un software. Esto es, hoy es distinto requerir “administrar” un recurso que “listar” sus atributos, la magnitud y el nivel de conceptualización le indican que los niveles de descomposición o jerarquía de verbos tienen envergaduras distintas y, por ende, esfuerzos asociados diferentes. En resumen, la idea es entender al cliente usando su propio lenguaje mediante el apoyo de esta técnica y mejorar la capacidad de estimar de mejor forma el esfuerzo (horas-hombre) de desarrollo de software.

5. BIBLIOGRAFIA

- PRESSMAN, R. (1998). *Ingeniería de software: Un enfoque práctico*, 4ª. Edición, Mc. Graw-Hill.
- GAUSE, D. C. and G.M. WEINBER (1989). *Exploring Requirements: Quality Before Design*, Dorset House.
- ZAHNISERM, R.A. (1990). "Building Software in Groups", *American Programmer*, vol. 3, núms. 7/8, july/agosto.
- HARS, A. y MARCHEWKA, J. (a), (1999). "La aplicación del procesamiento del lenguaje natural para el análisis de requisitos", www.cwa.mdx.ac.uk
- HARS, ALEXANDER y MARCHEWKA, J. (b), (1999). "Advancing CASE productivity by Using Natural Language Processing and Computerized Ontologies: The ACAPULCO Sytem", 1996, <http://www-rcf.usc.edu>
- WILSON, W., ROSEMBERG, L. y HYATT, L. (1999). "Análisis de calidad de especificación automatizada de requisitos de lenguaje natural", <http://satc.gsfc.nasa.gov>.
- BLOOM, BENJAMIN S., BERTRAM B. M. and KRATHWOHL, D. R. (1964). *Taxonomy of Educational Objectives* (two vols: The Affective Domain & The Cognitive Domain). New York. David McKay.
- FLORES, F. (1991). *Inventando la empresa del siglo XXI*, Ed. Hachette.