

**Universidad de las Ciencias Informáticas**

**“Facultad 3”**



**“PREDICTOR: SISTEMA DE DESCARGA Y PROCESAMIENTO  
AUTOMATIZADO DE PATENTES. ROL ANALISTA DE SISTEMAS.”**

**Trabajo de diploma para optar por el título de  
Ingeniero en Ciencias Informáticas**

**Autores:** Mairelys Plaza Matos y Yaniet Piñeiro Pérez

**Tutor:** Lic. Humberto Varona González

**Consultante:** Yusnelkis Milanés Guisado

**Junio de 2007**

## DECLARACIÓN DE AUTORÍA

Nosotras: Mairelys Plaza Matos y Yaniet Piñeiro Pérez, nos declaramos como únicas autoras de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) para que hagan el uso del mismo de la manera que estimen conveniente.

Y para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del 2007.

Mairelys Plaza Matos

Yaniet Piñeiro Pérez

---

Firma del autor

---

Firma del autor

Lic. Humberto Varona González

---

Firma del tutor

## AGRADECIMIENTOS

*A mi Padre y Señor por darme la vida y una familia a la que quiero mucho.*

*A mis padres y hermanos por confiar siempre en mí y apoyarme en todo momento.*

*A Daiana y Yorgenis por aceptarme como una hermana.*

*A Humberto por su amor, apoyo y comprensión.*

*A mis amigos y los amigos de mis amigos, especialmente a Yani, Lissy, Danaysa, Yaifin.....*

*A los profesores: Pascual, Eugenia, Maikel y Karina por su incondicionalidad, por sus ideas y recomendaciones en esta investigación.*

*A la UCI por formarnos como Ingenieros en Ciencias Informáticas y a Fidel por confiar en que seremos los forjadores del futuro.*

*Mairelys*

*A mis padres y a mi familia por su amor y por apoyarme en cada momento de mi vida.*

*A mis amistades, en especial a Margarita por su amistad, cariño y por ser como una hermana para mí.*

*A Rolando por brindarme su ayuda en todo momento.*

*A todos los profesores que de una forma u otra me ayudaron en la realización de la tesis.*

*A la Revolución y a Fidel por darme la oportunidad de pertenecer a la Universidad de las Ciencias Informáticas y formar parte de la tropa de futuro.*

*Yaniet*

## DEDICATORIA

*A mis padres, porque gracias a su cariño y apoyo incondicional he logrado realizar uno de mis mayores sueños.*

*Mairelys*

*A mis padres Dania y Ramón por darme todo su amor y comprensión, por ayudarme a lograr mi sueño.*

*A mi hermanita querida Yohandra por apoyarme y preocuparse por mí.*

*A mi novio Luis Ernesto por su amor y apoyarme todos estos años.*

*A mi familia por acompañarme en todo momento.*

*Janiet*

## RESUMEN

Los documentos de patentes constituyen una fuente imprescindible para conocer el estado y la evolución del desarrollo tecnológico así como la actividad innovadora. En la medida en que ha aumentado la disponibilidad de la información sobre patentes, han aumentado también los sistemas de búsqueda, recuperación y análisis de dicha información que permiten el estudio y aprovechamiento del enorme potencial estratégico que brinda su análisis. El proyecto *Delfos* surge en la Universidad de las Ciencias Informáticas por la necesidad de los especialistas de la consultoría del Ministerio de la Informática y las Comunicaciones (MIC) de disponer de un sistema para la obtención de información específica sobre patentes que se encuentran publicadas en bases de datos de Internet. En este trabajo se describe el análisis del *Sistema de descarga y procesamiento automatizado de patentes* propuesto para realizar dicha actividad de una forma rápida y eficiente. Para ello se realizó un estudio sobre las metodologías de desarrollo de software más usadas actualmente, así como el uso de patrones del análisis y los aspectos concernientes al rol Analista de Sistemas. Finalmente se evaluó el modelo propuesto mediante el uso de métricas dirigidas a garantizar la calidad de la especificación y del modelo de casos de uso del Sistema, obteniéndose resultados positivos.

Palabras clave [patentes, metodologías de desarrollo, patrones de análisis, analista de sistemas]

# Tabla de Contenidos

<b>INTRODUCCIÓN.....</b>	<b>1</b>
<b>CAPÍTULO 1 FUNDAMENTACIÓN DEL TEMA .....</b>	<b>7</b>
1.1. Base datos de Patentes .....	7
1.2. Herramientas de Búsqueda, Recuperación y Análisis de Patentes .....	8
1.3. Metodologías de Desarrollo de Software más usadas actualmente.....	10
1.3.1. Concepto de metodología de desarrollo de software.....	10
1.3.2. Extreme Programming (XP) .....	11
1.3.3. Object Modeling Technique (OMT) .....	14
1.3.4. Rational Unified Process (RUP) .....	15
1.4. El Analista de Sistemas.....	18
1.5. Patrones de casos de uso.....	20
1.6. Ingeniería de requisitos .....	22
1.7. Técnicas de captura de requisitos .....	27
1.7.1. Las entrevistas.....	28
1.7.2. Tormenta de ideas (Brainstorming) .....	28
1.7.3. JAD (Join Application Development / Desarrollo conjunto de aplicaciones).....	29
1.7.4. Cuestionarios y Checklists .....	29
1.7.5. Comparación de terminología .....	29
<b>2 CAPÍTULO 2 SOLUCIÓN PROPUESTA.....</b>	<b>31</b>
2.1. Selección de la Metodología de Desarrollo de Software .....	31
2.2. Selección de herramientas a utilizar según la metodología.....	32
2.2.1. Rational Rose Enterprise Edition.....	33
2.3. Modelo del negocio .....	34
2.4. Descripción de los casos de uso del negocio .....	37
2.5. Captura de requisitos. Técnicas, métodos y plantillas a utilizar .....	41

2.6.	Especificación de requisitos del software .....	42
2.6.1.	Requerimientos Funcionales .....	44
2.6.2.	Requerimientos no Funcionales .....	45
2.7.	Aplicación de patrones del análisis en el modelo del sistema .....	47
2.8.	Descripción de los casos de uso del sistema .....	49
2.9.	Prototipo de sistemas.....	70
<b>3</b>	<b>CAPÍTULO 3 VALIDACIÓN DE LOS RESULTADOS .....</b>	<b>72</b>
3.1.	Modelo del negocio. ....	73
3.2.	Especificación de requisitos del sistema.....	73
3.2.1.	Aplicación de las Métricas de la Calidad de la Especificación de Requisitos .....	74
3.3.	Prototipo no funcional.....	76
3.4.	Comprobación De La Calidad Del Modelo De Casos De Uso Del Sistema .....	77
3.4.1.	Definición de factores para el Modelo de casos de uso del Sistema.....	79
	<b>CONCLUSIONES.....</b>	<b>84</b>
	<b>RECOMENDACIONES .....</b>	<b>85</b>
	<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>86</b>
	<b>BIBLIOGRAFÍA.....</b>	<b>88</b>

## Índice de Figuras

FIGURA 1: HITOS Y FASES DE RUP .....	17
FIGURA 2: LOS NUEVE FLUJOS DE TRABAJO DE RUP .....	17
FIGURA 3: CARACTERÍSTICAS DE UN ANALISTA DE SISTEMAS .....	19
FIGURA 4: DIAGRAMA DE CASOS DE USO DEL NEGOCIO .....	36
FIGURA 5: DIAGRAMA DE ACTIVIDADES DEL CASO DE USO DEL NEGOCIO BUSCAR PATENTES .....	39
FIGURA 6: DIAGRAMA DE ACTIVIDADES DEL CASO DE USO PROCESAR PATENTES .....	41
FIGURA 7: DIAGRAMA DE CASOS DE USO DEL SISTEMA .....	48
FIGURA 8: CONSTRUCCIÓN DE PROTOTIPOS .....	70
FIGURA 9: GRÁFICO DE CONTROL DE LA CALIDAD DE LA ESPECIFICACIÓN DE REQUISITOS .....	75
FIGURA 10: GRÁFICO DE CONTROL DE LA CALIDAD DEL MODELO DE CASOS DE USO PARA LA PRIMERA REVISIÓN .....	80
FIGURA 11: GRÁFICO DE CONTROL DE LA CALIDAD DEL MODELO DE CASOS DE USO PARA LA SEGUNDA REVISIÓN .....	82

## Índice de Tablas

TABLA 1: ACTOR DEL NEGOCIO .....	37
TABLA 2: TRABAJADOR DEL NEGOCIO .....	37
TABLA 3: DESCRIPCIÓN DEL CASO DE USO BUSCAR PATENTES .....	37
TABLA 4: DESCRIPCIÓN DEL CASO DE USO DEL NEGOCIO PROCESAR PATENTES .....	39
TABLA 5: DESCRIPCIÓN DEL CASO DE USO AUTENTICAR .....	49
TABLA 6: DESCRIPCIÓN DEL CASO DE USO GESTIONAR DATOS USUARIO .....	50
TABLA 7: DESCRIPCIÓN DEL CASO DE USO GESTIONAR DATOS USUARIO .....	54
TABLA 8: DESCRIPCIÓN DEL CASO DE USO CONFIGURAR CONEXIÓN A INTERNET .....	58
TABLA 9: DESCRIPCIÓN DEL CASO DE USO PROCESAR PATENTES .....	60
TABLA 10: DESCRIPCIÓN DEL CASO DE USO BUSCAR PATENTES .....	62
TABLA 11: DESCRIPCIÓN DEL CASO DE USO REALIZAR BÚSQUEDA BÁSICA .....	63
TABLA 12: DESCRIPCIÓN DEL CASO DE USO REALIZAR BÚSQUEDA AVANZADA .....	65
TABLA 13: DESCRIPCIÓN DEL CASO DE USO REPORTAR RESULTADOS .....	68
TABLA 14: ATRIBUTOS Y FACTORES PARA LA MEDICIÓN DE LA CALIDAD DEL DIAGRAMA DE CASOS DE USO DEL SISTEMA .....	78



# INTRODUCCIÓN

## *Actualidad del tema:*

La creciente necesidad de innovar en las empresas, tanto de base científica y tecnológica como convencionales, obliga a establecer procesos formalizados de análisis de los conocimientos, tecnologías, y “saber hacer” (know how) disponibles o en desarrollo en el entorno, entre lo que cabe destacar las patentes<sup>1</sup>(COMAI 2006).

La información sobre patentes incluye no sólo el contenido de los documentos de patente publicados sino también las informaciones bibliográficas y de otro tipo relativas a las patentes de invención, a los certificados de inventor, a los certificados de utilidad y a los modelos de utilidad. Se trata de la recopilación más completa, actualizada y mejor clasificada de documentos técnicos sobre tecnologías nuevas e innovadoras. La Organización Mundial de la Propiedad Intelectual (OMPI) establece que la utilización de la información sobre patentes como tal se ha extendido a muchas actividades comerciales tácticas y estratégicas diferentes, a la investigación y a las actividades de formulación de políticas a nivel nacional, institucional o de las empresas.

Ejemplo de ello es el aumento sostenido de la inversión extranjera directa en la India desde que se introdujo la reforma en materia de patentes y marcas a principios de los años 90. Esta tendencia fue incluso más perceptible en el Brasil, donde se registró un aumento extraordinario de esas inversiones tras la promulgación de una nueva ley de propiedad industrial en 1996 (de 4.400 millones de dólares en 1995 a 32.800 millones en 2000) (IDRIS 2003).

---

<sup>1</sup> Contrato entre el Estado y los inventores, el primero de ellos le otorga un título que le confiere al titular el derecho de monopolizar (temporalmente) la explotación industrial y comercial de la invención patentada. El inventor debe divulgar el contenido de la invención para favorecer el progreso técnico. El documento en el que aparece la divulgación de la invención es el documento de patente (SANCHEZ 2006).

Japón, Estados Unidos y Finlandia, tienen las más altas productividades en patentes del planeta. En particular EE.UU. tiene una poderosa infraestructura legal, y la producción de patentes más grande del mundo. La base de datos de la United States Patent & Trademark Office (USPTO) es la más grande del planeta, y es una fuente inconmensurable de información tecnológica y científica, quizás equivalente a Medline (Medline, 2002) en términos de importancia. En Alemania el estado a través del Bundesministerium für Bildung und Forschung (Ministerio Federal para la Educación y la Investigación) considera al patentamiento como el medio para asegurar el futuro de Alemania, y junto al apoyo a la investigación científica, les permitirá colocarse internacionalmente en primera fila.

Paap (COMAI 2006) considera que mediante el análisis realizado a patentes es posible obtener:

- Los principales actores (competidores y colaboradores actuales y potenciales) y sus áreas de concentración.
- Los desplazamientos de interés de los citados actores evaluando la mayor o menor importancia que conceden a una tecnología o a una línea de investigación y desarrollo.
- La organización del esfuerzo técnico y los movimientos de personal en el tiempo entre departamentos.
- Las estrategias de patentes utilizadas por los participantes y las oportunidades y amenazas de las estrategias “alrededor de las patentes”.

Según Breitzman (HERNÁNDEZ 2006) la evolución de la cantidad de patentes por año (edad de la tecnología), los países de origen y los de destino de la tecnología, los principales objetos de patentes, las relaciones entre estos parámetros, así como el ciclo de vida de la tecnología, son algunos de los parámetros que usualmente se pueden determinar para estudiar el estado de la tecnología basado en información de patentes.

Dada la situación antes descrita se plantea la siguiente situación problemática: Actualmente la consultoría Delfos es el centro coordinador del sistema de información del Ministerio de la Informática y las

Comunicaciones de Cuba, y como tal es la encargada de ejercer una vigilancia tecnológica<sup>2</sup> permanente en diferentes temáticas de interés para el desarrollo de ese sector, mediante un proceso de búsqueda, descarga y análisis de información relacionada con las patentes.

Esta actividad es realizada en la consultoría Delfos de forma manual, luego a la información descargada se le aplican diferentes tipos de filtros para homogeneizarla y extraer datos fundamentales de patentes. Todo esto trae consigo demoras y en ocasiones, se producen incumplimientos en la realización de dichas tareas. Por este motivo Delfos propuso a la Universidad de las Ciencias Informáticas, el desarrollo de un software que fuera capaz de descargar y procesar grandes volúmenes de información considerando las patentes como fuentes fundamentales del mismo.

El desarrollo de software es una actividad eminentemente colaborativa y necesaria en la cual diversos roles son desempeñados por los participantes: el cliente, especialistas en pruebas, analistas, diseñadores, jefes de proyecto, etc. Pero en el proyecto Delfos no existe una apropiada definición de roles, además la falta de organización y de procesos adecuados en su desarrollo, golpea de manera constante el avance del mismo. Específicamente la inexistencia de un analista de sistemas ha propiciado que existan problemáticas tales como:

- No existe una visión detallada de los procesos que ocurren en la organización.
- No se ha comprendido la estructura y dinámica de la organización en la que se implementará el sistema.
- No se ha establecido un convenio entre clientes y desarrolladores sobre lo que debe hacer el Sistema.

Lo cual permite plantear el siguiente problema: No están generados los artefactos del rol de analista de sistemas que garanticen el entendimiento común entre el cliente y los desarrolladores del proyecto Delfos.

---

<sup>2</sup> La vigilancia tecnológica está estrechamente unida a la gestión de la innovación y a la estrategia de la empresa. Esta se proyecta sobre la toma de decisiones empresariales alertando sobre posibles amenazas y oportunidades, aportando nuevos elementos y enfoques y reduciendo el riesgo.

EL problema planteado se enmarca en el siguiente objeto de estudio: Aplicación de la Ingeniería de software para el desarrollo de proyectos productivos.

Para resolver el problema planteado se define, como objetivo general de este trabajo: La realización del análisis y especificación detallada del negocio y del sistema que permita a los desarrolladores la implementación de un software confiable, mantenible y eficiente que satisfaga las necesidades de información de los especialistas de Delfos.

Cuyos objetivos específicos son:

- Elaborar los artefactos derivados del proceso de análisis al Sistema de descarga y procesamiento automatizado de patentes (Predictor).
- Evaluar los artefactos mediante el uso de métricas para la evaluación de la calidad.

El campo de acción determinado por el objeto de estudio planteado, para darle cumplimiento a este trabajo es el siguiente: Análisis del sistema para el software de gestión de patentes (Predictor).

Esta investigación está basada en la hipótesis de que si se desarrolla un análisis detallado y eficiente del sistema que permita un resultado satisfactorio en su implementación, se obtendrán mejores resultados en el ejercicio de vigilancia tecnológica realizado por los especialistas de Delfos.

Para el cumplimiento de los objetivos planteados se proponen las tareas de investigación siguientes:

- Desarrollar el estudio del estado del arte de las herramientas de descarga de información publicada en bases de datos de Internet para lograr un conocimiento más amplio del negocio en cuestión.
- Realizar un estudio de las metodologías para el desarrollo de sistemas informáticos más usadas actualmente que permita seleccionar la más adecuada para la obtención de los artefactos necesarios para el correcto diseño y posterior implementación del Sistema.
- Establecer las estrategias de captura de requisitos que permitan la obtención de los requerimientos del Sistema.
- Describir los artefactos derivados del análisis del Sistema para facilitar su comprensión.

La estrategia de investigación utilizada en el presente trabajo es la exploratoria, ya que se realizó la exploración de las diferentes metodologías y patrones que pueden ser utilizados por el analista, con vista a desarrollar una modelación adecuada para la implementación del sistema, que resuelva las necesidades de la consultoría Delfos.

Se planteó el problema, enfocando la problemática de cómo obtener el correcto análisis y modelación de un sistema, mediante un enfoque histórico lógico, ya que se realizó un estudio del estado del arte de las actividades que desempeña el analista de sistemas en el desarrollo de sistemas informáticos. También se sigue un método Hipotético Deductivo, pues a partir de un problema, se plantearon objetivos específicos a los cuales se les dará solución (partiendo de una hipótesis) en el transcurso de la investigación. Otro de los métodos que se siguió es el de Modelación pues mediante el mismo se pueden crear abstracciones con vista a explicar la realidad. Este método está presente en todo el proceso de desarrollo del Sistema, ya que para el análisis y especificación del mismo se necesitó de la elaboración de diagramas, figuras, modelos de negocio, de sistemas y otros artefactos importantes. Todo esto se fusiona logrando que la investigación tome un carácter sistémico-integral, pues se estudió al objeto mediante la determinación de sus componentes, así como la relación entre ellos conformando la realidad como totalidad.

También se realiza el seguimiento de métodos empíricos. En este caso la entrevista, como técnica fundamental para obtener información mediante una conversación profesional y de esta forma realizar la captura de requisitos del software en cuestión.

Con el desarrollo del análisis del Sistema se esperan obtener como resultado los modelos correspondientes a la metodología de desarrollo que se lleve a cabo y que permitan el entendimiento común de los procesos del negocio tanto para los desarrolladores del Sistema como sus usuarios finales. Resultados que serán la entrada al flujo de diseño que tendrá en cuenta el diseñador que forma parte del equipo de desarrollo.

Los resultados de los integrantes del equipo de desarrollo del proyecto Delfos- según el rol que desempeñe cada cual- conformarán un todo lográndose el resultado final: *Predictor*, herramienta que será de mucha utilidad para el servicio de vigilancia tecnológica, ya que resulta de gran importancia para la competitividad de una empresa el disponer del conocimiento de los cambios, la evolución y las expectativas de las variables que influyen en su sector de actividad.

El desarrollo de este trabajo está organizado en 3 capítulos. El capítulo 1 trata de la Fundamentación del tema: se hace un breve estudio de las herramientas de descarga, recuperación y análisis de patentes que se encuentran en bases de datos de Internet. Se hace un estudio de algunas de las metodologías de desarrollo de software más usadas actualmente con el objetivo de seleccionar luego la más idónea para el proyecto. Se plantean las estrategias y métodos de captura de requisitos, así como los patrones del análisis, más conocidos como patrones de casos de uso, que servirán para modelar correctamente los casos de uso del sistema. El capítulo 2 contiene la solución propuesta pues se realiza el desarrollo de la actividad del analista de sistemas en el proyecto Delfos usando la metodología RUP (propuesta de solución para la obtención de los artefactos del análisis). En él se presenta el Modelo del Negocio, donde se encuentra reflejada toda la lógica del negocio de la consultoría Delfos. Se presentan las técnicas seleccionadas para la captura de los requisitos funcionales y no funcionales derivados de las necesidades del cliente; y se obtiene el Modelo de casos de uso del sistema con sus respectivas realizaciones. El capítulo 3 contiene el análisis de los resultados que se derivaron del proceso de análisis a Predictor. Para ello se aplican métricas para la Especificación de Requisitos y un Modelo de Métricas para el Análisis y el Diseño Orientados a Objetos que ayudará a controlar o validar los casos de uso que se obtuvieron en el Modelo del Sistema una vez capturados los requisitos.

# CAPÍTULO 1 FUNDAMENTACIÓN DEL TEMA

En este capítulo se mostrará alguna información sobre las bases de datos de patentes y varios sistemas que ya han sido implementados para la descarga y análisis de información; además de que se hará un análisis crítico de investigaciones anteriores y de fuentes con enfoques, teorías y modelos relacionados con el estudio del estado del arte del Analista de Sistemas. Quedarán especificadas las técnicas utilizadas por el analista para la captura de requisitos, así como los patrones del análisis y metodologías de desarrollo de software más usadas en la actualidad.

## 1.1. Base datos de Patentes

En la actualidad el entorno en el que un usuario típico desarrolla su trabajo puede incluir tanto el acceso a bases de datos de patentes, como el acceso a otras bases de datos bibliográficas, normalmente de tipo científico–tecnológico que se suelen complementar entre ellas. Por otro lado, estas bases de datos pueden estar situadas tanto en un puesto de la red local (por ejemplo en una base de datos propia o en una base de datos comercial en cdrom) como en un sitio web en Internet (COMAI 2006).

Por lo general las bases de datos bibliográficas son las más comunes e incluyen campos como título de la patente, clasificación internacional, datos de prioridad, otras referencias, etc. Los distribuidores, llamados generalmente Host, ofrecen el acceso a varias bases de datos en línea por ejemplo DIALOG incluye el acceso a más de 400 bases de datos sobre diferentes áreas del conocimiento. Relacionadas con información de patentes y tecnológica; en general, se encuentran a Claims/U.S Patents, a Trade and Industry Index, a World Patents Index, etc. (SÁNCHEZ 1999).

También se pueden mencionar las bases de datos Esp@cenet, World Wide (International), United States Patent and Trademark Office (USPTO), State Intellectual Property Office (SIPO) de China, Patent Abstract of Japan (PAJ) a las cuales se conectan sistemas implementados para la descarga y análisis de las patentes, artículos científicos y otro tipo de información. Con estos documentos provenientes de la descarga, se realiza el procesamiento de dicha información y se obtienen luego de su análisis los

productos de Inteligencia Empresarial, entre los que se destacan: los estudios de tendencias, estudios de mercado y de estrategia.

## 1.2. Herramientas de Búsqueda, Recuperación y Análisis de Patentes

Diversos sistemas de búsqueda, recuperación y análisis de patentes y otro tipo de información han aparecido constantemente en Internet, entre los más destacados (que utilizan las grandes compañías dedicadas a la Vigilancia Tecnológica y a la Inteligencia Competitiva) se tiene:



*Matheo Patent:* Realiza la búsqueda, recuperación y análisis de patentes de las bases de datos de la Oficina de Patentes de los EE.UU. (USPTO) y de la Oficina Europea de Patentes (EPO). Tiene una interfaz amigable y posee diversas funcionalidades para la recuperación y análisis de patentes. ([www.matheo-software.com](http://www.matheo-software.com)).



*PatentHunter 3.0:* Busca, descarga y realiza otras funciones como por ejemplo, mostrar el documento en formato PDF, HTML y permite imprimirlo.

Su versión demo se encuentra disponible en: ([www.patenthunter.com](http://www.patenthunter.com)).



*BizInt Smart Charts:* Carga la información obtenida de bases de datos de patentes como la *World Patents Index (WPI)*, CA/CAplus, MicroPatent<sup>3</sup>, Delphion y otras. Permite cargar esa información en una hoja electrónica,

reformatearla y hacerle análisis estadísticos. ( [www.bizcharts.com/index.html](http://www.bizcharts.com/index.html)).

---

<sup>3</sup> MicroPatent: Es en este momento el líder Mundial en la producción y distribución de Información sobre patentes y Marcas Registradas. Esta importante empresa proporciona soluciones tajantes a la línea de investigación, en la entrega y manejo de documentos.





*Aureka*: Software desarrollado por Micropatent. En la versión ThemeScape los temas se representan visualmente en mapas con aspecto cartográfico, identificando los conceptos predominantes y sus relaciones; además se pueden comparar compañías, competidores o tecnologías. En la versión Citation Tree construye un árbol de citas a partir de una patente seleccionada. ([www.micropat.com/static/test.htm](http://www.micropat.com/static/test.htm), [www.infovis.net/printMag.php?num=167&lang=2](http://www.infovis.net/printMag.php?num=167&lang=2) )

*Delphion-Text Clustering*: Es una función en línea que extrae los términos más relevantes del texto de los resultados de una búsqueda, analiza sus relaciones y los presenta en un mapa. ([www.delphion.com/products/research/products-cluster](http://www.delphion.com/products/research/products-cluster)).

*Bibexcel*: Permite combinar la información extractada de diferentes campos de un registro, realiza conteos por frecuencia, co-ocurrencias de diversos elementos, y emparejamiento bibliográfico (bibliographic coupling). Además de estas tareas, cuenta con un procedimiento para encontrar enlaces (links) de citas entre diferentes documentos de un conjunto determinado. (<http://www.umu.se/inforsk> ).

**PatentLab-II** *PatentLab-II*: Se utiliza sólo para analizar datos descargados de la base de datos Thomson Delphion. Posibilita la agrupación por familias de patentes, para evitar las duplicidades en el análisis; los conteos de los diferentes campos tales como año de publicación, entidad solicitante o titular, inventor, temáticas, etc.

**VantagePoint** *VantagePoint 4.0*: Software para el análisis de información científico-técnica y de patentes desarrollado por Search Technologies. Permite la visualización de relaciones mediante matrices de co-ocurrencia, mapas tecnológicos y la creación de tesauros para la reducción de datos. Además, realiza análisis estadísticos multidimensionales para identificar grupos y relaciones entre conceptos, autores, países. (<http://www.thevantagepoint.com>).

Aunque ya existen sistemas (como los anteriores) que permiten la búsqueda, descarga y análisis de información para el ejercicio de vigilancia tecnológica, la consultoría Delfos determinó que es necesario el desarrollo de un sistema sencillo que se base solamente en la descarga y procesamiento de patentes, y no en el análisis de información, ya que para ello se utilizan otros programas muy efectivos a los que simplemente le especifican los parámetros de análisis deseados. A esto hay que agregar que los sistemas

antes mencionados no son gratuitos, por lo que hay que pagar sus licencias además de que tienen otras funcionalidades que no son de interés para los especialistas de Delfos.

### **1.3. Metodologías de Desarrollo de Software más usadas actualmente**

#### **1.3.1. Concepto de metodología de desarrollo de software**

El impacto del software en nuestra sociedad y en la cultura continúa siendo profundo. Al mismo tiempo que crece su importancia, la comunidad del software trata continuamente de desarrollar tecnologías que hagan más sencillo, rápido y menos costosa la construcción de programas de computadora de alta calidad. La tecnología que comprende un proceso, un juego de métodos y un conjunto de herramientas se llama *ingeniería de software*.(PRESSMAN 1998a)

Piattini (CATALDI 2000) definió como metodología al "... conjunto de procedimientos, técnicas, herramientas, y un soporte documental que ayuda a los desarrolladores a realizar nuevo software".

Y Maddison (CATALDI 2000) define metodología como un conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de sistemas de información.

Aunque no se haya llegado a un consenso en cuanto al concepto de metodología, se puede observar que las definiciones dicen mucho de lo que se conoce como tal, por tanto sí se tiene un acuerdo al respecto y es en cuanto a los procedimientos a seguir para obtener un producto informático.

Con el paso de los años estas metodologías se han venido desarrollando para el proceso de software, todas con sus ventajas, desventajas y utilidad en algunos tipos de proyectos y problemas. Es decir, no existe una metodología software universal, esta debe adecuarse a las características específicas de cada proyecto (recursos, equipo de desarrollo, etc.) que exigen que el proceso sea configurable.

A la hora de afrontar el desarrollo de este trabajo, se han evaluado las metodologías más usadas actualmente para elegir la que más se ajusta al mismo. Entre ellas se puede mencionar a *Extreme Programming (XP)*, *Object Modeling Technique (OMT)* y *Rational Unified Process (RUP)*.

### 1.3.2. Extreme Programming (XP)

Extreme Programming se encuentra entre las llamadas Metodologías Ágiles, que surgieron como una alternativa a los procesos de desarrollo de software tradicionales. Esta metodología se basa en la realimentación entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Es una metodología adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Es una de las metodologías de desarrollo de software con más éxito en la actualidad. Se utiliza en proyectos con equipo de desarrollo pequeños y con plazo de entrega corto. La metodología consiste en una programación rápida o extrema. Una particularidad es que tiene como miembro del equipo al usuario final (CARRILLO 2004).

Fases de la metodología XP, Tomado de <http://programacionextrema.tripod.com/fases.htm>

#### *1ª Fase: Planificación del proyecto*

Historias de usuario: Las historias de usuario tienen la misma finalidad que los casos de uso pero con algunas diferencias: Constan de 3 ó 4 líneas escritas por el cliente en un lenguaje no técnico sin hacer mucho hincapié en los detalles; en esta etapa no se habla ni de algoritmos para la implementación del sistema, ni de diseños de base de datos adecuados, etc.

Release planning: Luego es necesario crear un plan de publicaciones o "Release plan"<sup>4</sup>, donde se indiquen las historias de usuario que se implementarán para cada versión de la aplicación y las fechas en las que se publicarán dichas versiones del programa.

---

<sup>4</sup> Release plan: Es una planificación donde los desarrolladores y clientes establecen los tiempos de implementación ideales de las historias de usuario, la prioridad con la que serán implementadas y las historias de usuario que serán implementadas en cada versión del programa.

- Iteraciones. Todo proyecto que siga la metodología X.P se ha de dividir en iteraciones de aproximadamente 3 semanas de duración. Al comienzo de cada iteración los clientes deben seleccionar las historias de usuario definidas en el "Release planning" que serán implementadas.
- Velocidad del proyecto. La velocidad del proyecto es una medida que representa la rapidez con la que se desarrolla el mismo; basta con contar el número de historias de usuario que se pueden implementar en una iteración; de esta forma, se sabrá el cupo de historias que se pueden desarrollar en las distintas iteraciones.
- Programación en pareja. El trabajo en pareja involucra a dos programadores trabajando en el mismo equipo; mientras uno codifica haciendo hincapié en la calidad de la función o método que está implementando, el otro analiza si ese método o función es adecuado y está bien diseñado.
- Reuniones diarias. Es necesario que los desarrolladores se reúnan diariamente y expongan sus problemas, soluciones e ideas de forma conjunta.

### *2ª Fase: Diseño*

Diseños simples: La metodología X.P sugiere que hay que conseguir diseños simples y sencillos. Hay que procurar hacerlo todo lo menos complicado posible para conseguir un diseño fácilmente entendible e implementable que a la larga costará menos tiempo y esfuerzo desarrollar.

Glosarios de términos: Usar glosarios de términos y una correcta especificación de los nombres de métodos y clases ayudará a comprender el diseño y facilitará sus posteriores ampliaciones y la reusabilidad del código.

Riesgos: Si surgen problemas potenciales durante el diseño, X.P sugiere utilizar una pareja de desarrolladores para que investiguen y reduzcan al máximo el riesgo que supone ese problema.

Funcionalidad extra: Nunca se debe añadir funcionalidad extra al programa aunque se piense que en un futuro será utilizada. Sólo el 10% de la misma es utilizada lo que demuestra que el desarrollo de funcionalidad extra es un desperdicio de tiempo y recursos.

Refactorizar: Es mejorar y modificar la estructura y codificación de códigos ya creados sin alterar su funcionalidad. Supone revisar de nuevo la codificación para procurar optimizar su funcionamiento.

Tarjetas C.R.C (Clase - Responsabilidad - Colaborador): Su objetivo es facilitar la comunicación y documentar los resultados. Para cada clase identificada se rellenará una tarjeta de este tipo y se especificará su finalidad así como otras clases con las que interaccione.

### *3ª Fase: Codificación*

La codificación debe hacerse atendiendo a estándares y patrones de codificación ya creados. Programar bajo estándares mantiene el código consistente y facilita su comprensión y la escalabilidad.

X.P sugiere un modelo de trabajo usando repositorios de código donde las parejas de programadores publican cada pocas horas sus códigos implementados y corregidos junto a los test que deben pasar. X.P también propone un modelo de desarrollo colectivo en el que todos los programadores están implicados en todas las tareas; cualquiera puede modificar o ampliar una clase o método de otro programador si es necesario y subirla al repositorio de código. El permitir al resto de los programadores modificar códigos que no son suyos no supone ningún riesgo ya que para que un código pueda ser publicado en el repositorio tiene que pasar los test de funcionamiento definidos para el mismo.

### *4ª Fase: Pruebas*

Uno de los pilares de la metodología X.P es el uso de test para comprobar el funcionamiento del código que estamos desarrollando. Estos se deben crear con frameworks específicos para tests (JUnit, por ejemplo para Java). Hay que someter a tests a las distintas clases del sistema omitiendo los métodos más triviales. Además se deben crear los aquellos que se aplicarán a una clase/método antes de implementarla; en el apartado anterior se explicó la importancia de crear antes los test que el código. Un punto importante es crear tests que no tengan ninguna dependencia del código que en un futuro evaluará. Conjuntamente hay que crearlos abstrayéndose del futuro código, de esta forma se asegurará la independencia del test respecto al código que evalúa.

### 1.3.3. Object Modeling Technique (OMT)

La metodología OMT fue creada por James Rumbaugh y Michael Blaha en 1991, mientras James dirigía un equipo de investigación de los laboratorios General Electric. Es una de las metodologías de análisis y diseño orientada a objetos, más madura y eficiente que existe en la actualidad. La gran virtud que aporta esta metodología es su carácter de abierta (no propietaria), que le permite ser de dominio público y, en consecuencia, sobrevivir con enorme vitalidad. Esto facilita su evolución para acoplarse a todas las necesidades actuales y futuras de la ingeniería de software (ROJAS 1997).

OMT pone énfasis en la importancia del modelo y uso de modelo para lograr una abstracción, en el cual el análisis está enfocado en el mundo real para un nivel de diseño, también pone detalles particulares para modelado de recursos de la computadora. Esta metodología puede ser aplicada en varios aspectos de implementación incluyendo archivos, base de datos relacionales, base de datos orientados a objetos. OMT está construido alrededor de descripciones de estructura de datos, constantes, sistemas para procesos de transacciones (ESPINOZA 1997). Por tanto se puede decir que esta metodología enfatiza en el análisis, no siendo así en la implementación y más en los datos que en las funciones que permiten la estabilidad en el proceso de desarrollo del software.

#### Fases de la metodología OMT

**Análisis:** Es donde el analista define lo que debe hacer el sistema deseado, es decir sus características mediante la construcción del modelo de dominio, mostrando las propiedades importantes y además se obtiene el modelo de objeto y la relación que existe entre ellos.

**Diseño del sistema:** Durante esta fase se toman decisiones importantes sobre la arquitectura del sistema, para organizarlo en subsistemas teniendo en cuenta la estructura del análisis y de la arquitectura propuesta.

**Diseño de objetos:** En esta fase, es donde el diseñador de objetos apoyándose en el modelo de análisis realizado para construir el modelo del diseño, incluyendo algunos detalles de implementación, como son los algoritmos necesarios para implementar cada una de las clases existentes en el sistema.

Implementación: En esta fase hay que tener en cuenta los principios de la ingeniería de software para que exista una correspondencia adecuada de la implementación del sistema con el diseño del mismo.

*La metodología OMT emplea tres clases de modelos para describir el sistema:*

Modelo de objetos: Es donde se describe la estructura de los objetos del sistema.

Modelo dinámico: Es donde se muestra la secuencia de las operaciones a realizar, así como la secuencia de los sucesos.

Modelo funcional: Es donde se capturan todas las funcionalidades del sistema.

#### **1.3.4. Rational Unified Process (RUP)**

Al igual que cualquier notación, el Proceso Unificado actúa como un modelo que puede adaptarse a cualquier tipo de proyecto y empresa ya sean grandes o pequeñas. Además de esto la metodología RUP (Rational Unified Process)...” es un modelo que involucra un análisis de riesgo, cubre todo el ciclo de vida del producto, soporta un enfoque de desarrollo iterativo e incremental” esto permite que los riesgos críticos sean resueltos antes de hacer grandes inversiones, se obtenga retroalimentación temprana por parte de los usuarios y las pruebas y la integración sean continuas, “proporciona iteraciones tempranas que se enfocan en validar y producir una arquitectura de software, y un ciclo de desarrollo inicial que toma la forma de un prototipo ejecutable que gradualmente evoluciona convirtiéndose en el sistema final y además tiene implícito en su proceso de desarrollo la evaluación continua de la calidad con respecto a los requerimientos de calidad deseados”(KRUCHTEN 1996).

Está equilibrado por ser el producto final de tres décadas de desarrollo y uso práctico donde se han unificado varias técnicas -partiendo del Lenguaje Unificado de Modelado (UML). Su desarrollo como producto sigue un camino desde el *Proceso Objectory* (primera publicación en 1987) pasando por el *Proceso Objectory de Rational* (publicado en 1997) hasta el *Proceso Unificado de Rational* (publicado en 1998).

En su modelación define como sus principales elementos (JACOBSON 1999):

- Trabajadores (“quién”)

Precisa responsabilidades (rol) y habilidades de un individuo o grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo para realizar determinadas actividades o desarrollar determinados artefactos.

- Actividades (“cómo”)

Es una tarea que tiene un propósito claro, es realizada por un trabajador y manipula elementos.

- Artefactos (“qué”)

Pieza de información tangible que es creada, modificada y usada por los trabajadores al realizar actividades. Este puede ser además un modelo, un elemento de un modelo, o un documento.

- Flujo de actividades (“Cuándo”)

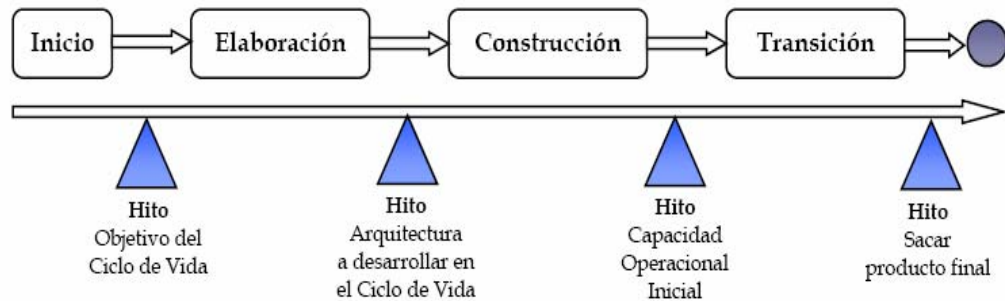
Secuencia de actividades realizadas por trabajadores y que produce un resultado de valor observable.

#### Fases de la metodología RUP

Cada ciclo en RUP está compuesto por cuatro fases, y dentro de cada una de ellas los líderes o los desarrolladores pueden descomponer (en iteraciones) el trabajo. Cada fase termina con un hito (ver Figura 1) y cada uno de estos se determina por la disponibilidad de un conjunto de artefactos.

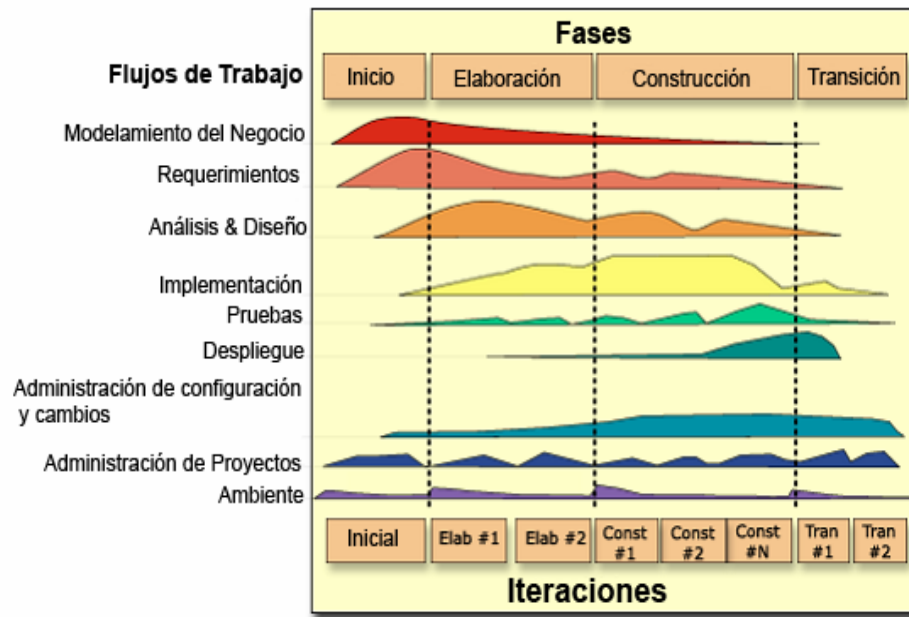
Los hitos tienen muchos objetivos. El más crítico es que los directores deben tomar ciertas decisiones cruciales antes de que el trabajo pueda continuar con la siguiente fase. Los hitos también permiten a la dirección, y a los mismos desarrolladores, controlar el progreso del trabajo según pasa por esos cuatro puntos clave (JACOBSON 1999).





**Figura 1: Hitos y Fases de RUP**

Para cada una de las cuatro fases se debe trabajar en seis disciplinas o flujos de trabajo, por lo que el proceso es iterativo, y a medida que se avanza en el tiempo de desarrollo se debe lograr pasar a una etapa superior. El esfuerzo que se dedica a cada una de las disciplinas de ingeniería depende del objetivo de la fase en que se encuentre el proyecto. Los tres últimos flujos, son llamados flujos de gestión o soporte y tienen actividades en todas las fases del proyecto. Esto se puede observar en la Figura 2.



**Figura 2: Los nueve flujos de trabajo de RUP**

*Fases que propone RUP (JACOBSON 1999):*

*Inicio:* En esta fase se establece un acuerdo entre todos los interesados acerca de los objetivos del proyecto. Esta fase es significativamente primaria para el desarrollo del software, pues se asegura de identificar los riesgos relacionados con el negocio y requerimientos. Para proyectos de mejora de software existente esta fase es más breve y se centra en asegurar que vale la pena y es posible desarrollar el proyecto.

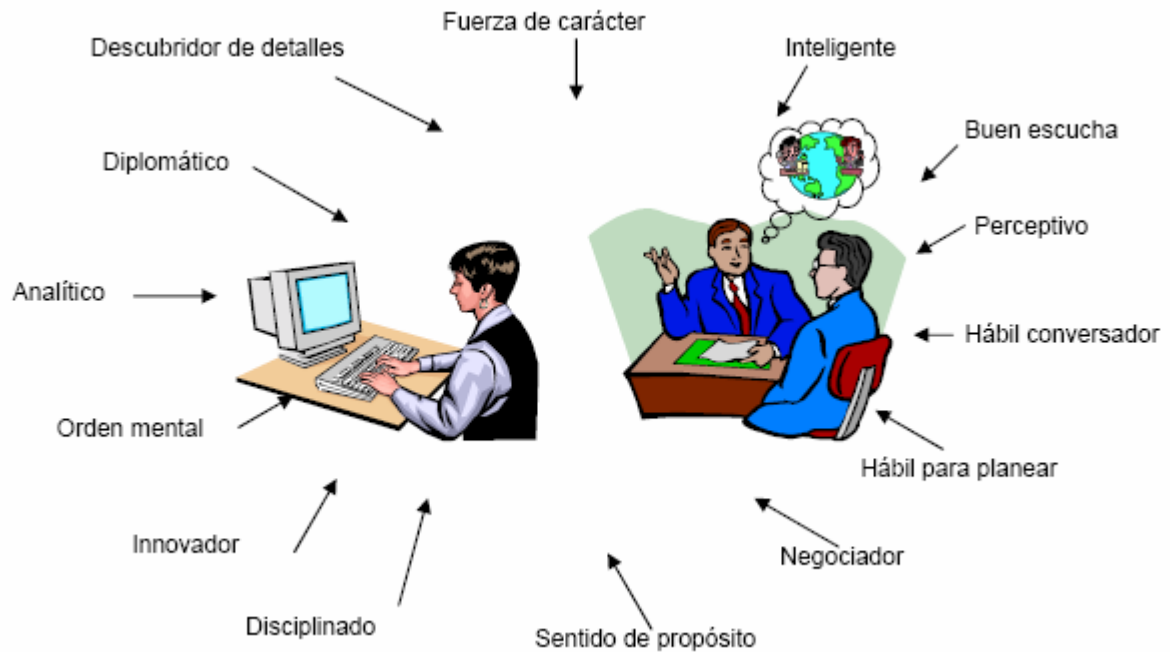
*Elaboración:* En la fase de Elaboración se establece la arquitectura base del sistema para proveer bases estables para el esfuerzo de diseño e implementación en la siguiente fase. La arquitectura debe abarcar todas las consideraciones de mayor importancia de los requerimientos y una evaluación del riesgo.

*Construcción:* El objetivo general de la fase de construcción es obtener un producto listo para ser distribuido como versión beta y ser sometido a pruebas. Esta fase es la más compleja de todas pues se necesita de más personal y de más tiempo, por esto además requiere de más iteraciones que las fases anteriores.

*Transición:* Esta fase se centra en implantar el producto en su entorno de operación. La forma en que el proyecto lleva a cabo este objetivo varía con la naturaleza de la relación del producto con su mercado.

#### **1.4. El Analista de Sistemas**

El Analista de Sistemas es la persona que debe establecer la relación con la organización o empresa solicitante de sistemas de información, y debe llevar sus requerimientos al procesamiento electrónico. Para realizar satisfactoriamente estas actividades requiere que además de tener una formación integral en informática, cuente con determinadas características personales las cuales se muestran en la Figura 3.



**Figura 3: Características de un Analista de Sistemas**

Como se expresa en (JACOBSON 1999) el Analista de Sistemas tiene como objetivo, identificar las necesidades del cliente y evaluar los conceptos que tiene el negocio mediante la entrevista para realizar la captura de los requisitos y así asignar las funcionalidades al software. Además “debe reconocer los elementos básicos del problema tal y como los percibe el cliente/usuario” (PRESSMAN 1998b).

Para ello debe recorrer desde los niveles más altos de la empresa (gerentes y directivos), hasta los niveles más bajos (obreros y empleados) para determinar quiénes realmente necesitan la información, con qué oportunidad y grado de detalle de cada peldaño de la escalera institucional. "Los gerentes y empleados tienen buenas ideas a qué es lo que sí trabaja y qué es lo que no, qué causa problemas y qué no, dónde son necesarios los cambios y dónde no"(SENN 1992).

Es responsable además de delimitar el sistema, encontrando los actores y los casos de uso y asegurando que el modelo de casos de uso es completo y consistente. Para la consistencia, el Analista de Sistemas puede utilizar un glosario para conseguir un acuerdo en los términos comunes, nociones y conceptos durante la captura de los requisitos.

El rol de analista se ha vinculado en este trabajo, con la construcción de un sistema de búsqueda, descarga y procesamiento de patentes publicadas en bases de datos de Internet que surge por la necesidad de contar con un software en una de las consultorías del MIC. Este se realiza para ejercer de una forma más rápida y eficiente la vigilancia tecnológica, ya que los sistemas de este tipo que existen en el mundo son software propietario.

### **1.5. Patrones de casos de uso**

Un patrón es una pieza de literatura que describe un problema de diseño y una solución general para el problema en un contexto particular. La importancia de utilizar patrones en la creación de sistemas complejos ha sido largamente reconocida en muchas disciplinas. Cada patrón describe un problema que ocurre una y otra vez, y describe la solución a ese problema, de tal manera que dicha solución pueda ser usada un millón de veces más, sin hacerlo necesariamente dos veces del mismo modo (RIDA0 2000).

Los patrones no se proponen descubrir ni expresar nuevos principios de la ingeniería de software. Todo lo contrario: intentan codificar el conocimiento, las expresiones y los principios ya existentes: mientras más trillados y generalizados, mejor.

Los patrones de escenarios (como también se le llaman) que han sido utilizados para crear los casos de uso que reflejan de forma más exacta los requisitos verdaderos, son más fáciles de traducir a sistemas de trabajo, y son más simples de mantener. Utilizando estos patrones, los arquitectos, los analistas, los ingenieros, etc., pueden alcanzar mejores resultados.

#### *Patrones de descripción*

Reglas de negocio: Definición estática

Descripción: Este patrón se aplica a todos los casos de uso que modelan los servicios que son afectados por las reglas de negocio definidas en la organización. Sin embargo, no tiene influencia en la estructura del modelo de caso de uso; ya que se ocupa de la descripción de los casos de uso. Las reglas se describen en un documento separado, referido por las descripciones relevantes del caso de uso.

Aplicabilidad: Este patrón es apropiado utilizarlo cuando no hay necesidad de cambiar dinámicamente las reglas del negocio mientras que el sistema está funcionando.

#### *Patrones de estructura*

Extensión o inclusión concreta: Inclusión

Descripción: En este patrón, existe una relación de inclusión del caso de uso base con el caso de uso incluido. Este último puede ser instanciado como el mismo. El caso de uso base puede ser concreto o abstracto.

Aplicabilidad: Se utiliza cuando un flujo puede ser incluido en el flujo de un caso de uso y también puede ejecutarse dentro de este.

Concordancia: Especialización

Descripción: Patrón de la concordancia que contiene casos de uso de la misma clase. En este caso, se modelan como especializaciones de un tipo común de caso de uso. Todas las acciones en el caso de uso de tipo común son heredadas por los casos de uso específicos, donde otras acciones pueden ser agregadas o las acciones heredadas pueden ser especializadas.

Aplicabilidad: Este patrón es aplicable cuando los modelados usados por los casos de uso son del mismo tipo, y este tipo se debe hacer visible en el modelo.

Múltiples actores: Rol común

Descripción: Es un patrón alternativo, donde hay dos actores que desempeñan el mismo papel hacia el caso de uso. Este papel es representado por otro actor, heredado por los actores que comparten este papel.

Aplicabilidad: Es aplicable cuando, desde el punto de vista del caso de uso, hay solamente una entidad externa que obra recíprocamente con cada caso de uso.

CRUD: Completo

Este patrón consiste en un caso de uso llamado *Gestión de Información* o CRUD Information, que fusiona las diferentes operaciones que pueden ser realizadas como simples casos de uso tales como: crear, leer, actualizar y eliminar segmentos de información dentro de un caso de uso formando una sola unidad conceptual. Este patrón debe ser utilizado cuando todos los flujos contribuyen al mismo valor de negocio y son todos cortos y simples. CRUD posee algunas ventajas obvias como es por ejemplo, la reducción del tamaño del modelo agrupando las 4 funciones básicas en una sola haciendo el modelo más claro para el analista.

### **1.6. Ingeniería de requisitos**

Desde el inicio del desarrollo de sistemas, los ingenieros se han encontrado con un gran problema: la identificación de los requisitos del sistema. Esto es debido a que no es un proceso que pueda ser determinado matemáticamente. Es más bien un proceso en el cual los datos son extraídos de las personas y estos datos pueden variar, dependiendo de la persona a la cual se esté consultando. Ahora, ¿cómo se puede asegurar que se ha especificado un sistema que recoge las necesidades del cliente y satisface sus expectativas? No hay una respuesta segura para a esta difícil pregunta, es por eso que la Ingeniería de Requisitos ha trabajado arduamente para tratar de desarrollar técnicas que permitan hacer este proceso de una forma más eficiente y segura.

La ingeniería de requisitos facilita el mecanismo apropiado para comprender lo que quiere el cliente, analizando, confirmando su viabilidad, negociando una solución razonable, especificando la solución sin ambigüedad, validando la especificación y gestionando los requisitos para que se transformen en un sistema operacional (THAYER 1997).

La Ingeniería de Requisitos es una disciplina de la Ingeniería de Software, donde se identifica el propósito del sistema, dirección y alcance del mismo. Es un conjunto de actividades y transformaciones que pretenden comprender las necesidades de un sistema software y convertir la declaración de estas necesidades en una descripción completa, precisa y documentada de los requerimientos del sistema

siguiendo un determinado estándar. Los requisitos constituyen el enlace entre las necesidades reales de los clientes, usuarios y stakeholders<sup>5</sup>.

El proceso de Ingeniería de Requisitos puede ser descrito en seis pasos (PRESSMAN 1998c):

#### Identificación De Requisitos

Preguntar al cliente, los usuarios y a los que están involucrados en los objetivos del sistema o producto y sean expertos, investigar cómo los sistemas o productos se ajustan a las necesidades del negocio, y finalmente cómo el sistema va a ser utilizado, es una tarea que parece fácil a simple vista, pero que en realidad no lo es. En (SAWYER 1997) los autores sugieren un conjunto de actuaciones para la obtención de requisitos, que están descritos en las tareas siguientes:

- Valorar el impacto en el negocio y la viabilidad técnica del sistema propuesto.
- Identificar las personas que ayudarán a especificar requisitos y contrastar su papel en la organización.
- Definir el entorno técnico (arquitectura de computación, sistema operativo, necesidades de telecomunicaciones) en el sistema o producto a desarrollar e integrar.
- Identificar restricciones de dominio (características específicas del entorno de negocio en el dominio de la aplicación) que limiten la funcionalidad y rendimientos del sistema o producto a construir.
- Definir uno o más métodos de obtención de requisitos (entrevistas, grupos de trabajo, equipos de discusión).
- Solicitar la participación de muchas personas para que los requisitos se definan desde diferentes puntos de vista, asegurarse de identificar lo fundamental de cada requisito registrado.
- Identificar requisitos ambiguos como candidatos para el prototipado, y crear escenarios de uso para ayudar a los clientes/usuarios a identificar mejor los requisitos fundamentales.

---

<sup>5</sup> Stakeholders: Personas u organizaciones que están activamente implicadas en el negocio ya sea porque participan en él o porque sus intereses se ven afectados.

### Análisis De Requisitos y Negociación

Es corriente en clientes y usuarios solicitar más de lo que puede realizarse, consumiendo recursos de negocio limitados. También es relativamente común en clientes y usuarios el proponer requisitos contradictorios, argumentando que su versión es esencial por necesidades especiales.

El ingeniero del sistema debe resolver estos conflictos a través de un proceso de negociación. Los clientes, usuarios y el resto del personal que interviene deberán clasificar sus requisitos y discutir los posibles conflictos según su prioridad. Los riesgos asociados con cada requisito son identificados y analizados. Se efectúan estimaciones del esfuerzo de desarrollo que se utilizan para valorar el impacto de cada requisito en el coste del proyecto y en el plazo de entrega. Utilizando un procedimiento iterativo, se irán eliminando requisitos, se irán combinando y/o modificando para conseguir satisfacer los objetivos planteados.

### Especificación De Requisitos

La especificación del sistema es el producto final sobre los requisitos del sistema obtenido por el ingeniero. Sirve como fundamento para la ingeniería del hardware, ingeniería del software, la ingeniería de bases de datos y la ingeniería humana. Describe la función y características de un sistema de computación y las restricciones que gobiernan su desarrollo. La especificación delimita cada elemento del sistema. La especificación del sistema describe la información (datos y control) que entra y sale del sistema.

### Modelado Del Sistema

Como casi todas las técnicas de modelado usadas en la ingeniería del software y de sistemas, el esquema del modelado del sistema permite al analista crear una jerarquía en detalle.

### Validación De Requisitos

La validación de requisitos examina las especificaciones para asegurar que todos los requisitos del sistema han sido establecidos sin ambigüedad, sin inconsistencias, sin omisiones, que los errores



detectados hayan sido corregidos, y que el resultado del trabajo se ajusta los estándares establecidos para el proceso, el proyecto y el producto.

### *Calidad De La Especificación De Los Requisitos*

En la idea de medir la calidad de una especificación de requisitos, una de las primeras aportaciones fue la propuesta por Davis en (DORIA 2001) donde, dada la lista de características de calidad de los requisitos que se resume más adelante, define como métricas la proporción de requisitos con defectos sobre el total de éstos, es decir, cocientes del tipo número de requisitos ambiguos/número total de requisitos.

Entre estas características de calidad se tienen las siguientes:

No ambigüedad: Tiene una única interpretación posible.

Completitud: La especificación de requisitos contempla todo lo que se supone que debe hacer el software, todas las respuestas a posibles entradas y situaciones, todas las páginas están numeradas y todos los elementos identificados, no hay secciones por determinar.

Corrección: Todo requisito contribuye a la satisfacción de una necesidad y todas las necesidades se hallan recogidas en la especificación.

Comprensibilidad: Todo lector puede entenderlo con un mínimo de explicación.

Verificabilidad: existen técnicas finitas y eficientes (de coste razonable) para verificar si el sistema satisface el requisito, un requisito no es verificable si es ambiguo, es un problema no resuelto o no se puede valorar el coste de probarlo.

Consistencia interna: No existe ningún subconjunto de requisitos que incluyan conflictos.

Consistencia externa: Ningún requisito entra en conflicto con otra documentación del proyecto.

Concisión: Es tan corto como sea posible sin disminuir la calidad de la especificación.

Independencia del diseño: Describe de forma pura comportamientos externos, por lo que existe más de un diseño e implementación posible.

Almacenado electrónicamente: Se ha realizado con un procesador de textos o a partir de una base de datos de requisitos.

Anotado con importancia relativa: Se puede determinar fácilmente cuales requisitos son los importantes y cuales son secundarios.

Anotado con estabilidad relativa: Se puede determinar fácilmente cuales requisitos son más susceptibles de cambio.

Anotado con versión: Se puede determinar fácilmente cuales requisitos se satisfarán en qué versiones del producto.

No redundancia: Cada requisito aparece una sola vez.

Al nivel correcto de detalle: Tan específico como para que cualquier sistema que satisfaga los requisitos, satisfaga las necesidades del cliente, pero lo suficientemente abstracto como para que cualquier sistema que satisfaga las necesidades del cliente satisfaga los requisitos.

Precisión: Se usan cantidades numéricas siempre que sea posible con el nivel adecuado de precisión.

Reusabilidad: Las frases, párrafos y secciones pueden ser fácilmente adoptadas o adaptadas para futuras especificaciones.

Trazabilidad: Está claro el origen del requisito, generalmente documentos anteriores del proyecto como requisitos de nivel superior, informes de investigación, etc.

Organización: Los contenidos están dispuestos de forma que se puedan localizar con facilidad y hay una relación lógica entre secciones adyacentes.

Referencias cruzadas: Figuran referencias: a un requisito idéntico (redundante), a un requisito que sea el mismo pero con más detalle o más abstracto, a un requisito del que depende o que depende de él.

Además, Davis considera que en una especificación de requisitos se pueden dar dos clases de errores:

Errores de conocimiento (*knowledge errors*): errores producidos por no conocer cuáles son los requisitos.

Errores de especificación (*specification errors*): errores producidos por no saber cómo especificar adecuadamente los requisitos.

Otro de los trabajos en esta línea es el propuesto por Hyatt y Rosenberg en 1996 (DORIA 2001), que expresa como objetivo la calidad en los requisitos argumentando que si al final del proceso de ingeniería de requisitos estos son ambiguos, incompletos o difíciles de entender aumenta el riesgo de que el producto final no sea satisfactorio. El conjunto de atributos que influyen en la calidad con sus métricas asociadas son:

Ambigüedad: Número de frases débiles y número de frases opcionales.

Completitud: Número de To Be Determined (TBD's).

Comprensibilidad: Estructura del documento e índice de legibilidad.

Volatibilidad: Número de cambios, número de requisitos y momento del ciclo de vida en el que se lleva a cabo el cambio.

Traceabilidad: Número de requisitos software no trazados a requisitos del sistema y número de requisitos software no trazados a pruebas y código fuente.

## Gestión De Requisitos

La *gestión de requisitos* es un conjunto de actividades que ayudan al equipo de trabajo a identificar, controlar y seguir los requisitos y los cambios en cualquier momento.

### 1.7. Técnicas de captura de requisitos

La captura de requisitos es la actividad mediante la cual, el equipo de desarrollo de un sistema de software extrae, de cualquier fuente de información disponible, las necesidades que debe cubrir dicho sistema. El proceso de captura de requisitos puede resultar complejo, principalmente si el entorno de trabajo es desconocido para el equipo de analistas, y depende mucho de las personas que participen en él. Algunas de las técnicas utilizadas para la captura de requisitos son las siguientes:

### 1.7.1. Las entrevistas

Deben ser planificadas, además se deben llevar a cabo las siguientes tareas previas (ESCALONA 2003):

- *Estudiar el dominio del problema:* Conocer las categorías y conceptos de la comunidad de clientes y usuarios es fundamental para poder entender las necesidades de dicha comunidad y su forma de expresarlas, y para generar en los clientes y usuarios la confianza de que el ingeniero de requisitos entiende sus problemas. Para tener claro el dominio del problema es necesario realizar un estudio profundo, consultar bibliografías sobre el tema así como conocer los antecedentes o el estado del arte de proyectos que ya se hayan implementado y que sean similares al que se este desarrollando.
- *Seleccionar a las personas a las que se va a entrevistar:* Usualmente se comienza por los directivos, que son quienes generalmente pueden dar una visión global del negocio; luego se continúa con los futuros usuarios y personal técnico que son quienes aportarán información más detallada sobre los procesos que ocurren en el negocio.
- *Determinar el objetivo y contenido de las entrevistas:* Para minimizar el tiempo de la entrevista es fundamental fijar el objetivo que se pretende alcanzar y determinar previamente su contenido.
- *Planificar las entrevistas:* Debe fijarse fecha, hora, lugar y duración. El lugar acordado debe ser un sitio agradable, y donde no ocurran interrupciones.

### 1.7.2. Tormenta de ideas (Brainstorming)

El objetivo de esta técnica es la generación de ideas en un ambiente libre de críticas o juicios. El brainstorming puede ayudar a generar una gran variedad de vistas del problema y a formularlo de diferentes formas, sobre todo al comienzo del proceso de elicitación, cuando los requisitos son todavía muy difusos. Esta técnica tiene la ventaja de que es muy fácil de aprender y requiere poca organización, de hecho, hay propuestas de realización de brainstorming por vídeo–conferencia a través de Internet. Por otro lado, al ser un proceso poco estructurado, puede no producir resultados con la misma calidad o nivel de detalle que otras técnicas.

### **1.7.3. JAD (Join Application Development / Desarrollo conjunto de aplicaciones)**

Esta técnica resulta una alternativa a las entrevistas. Es una práctica de grupo que se desarrolla durante varios días y en la que participan analistas, usuarios, administradores del sistema y clientes. Está basada en cuatro principios fundamentales: dinámica de grupo, el uso de ayudas visuales para mejorar la comunicación, mantener un proceso organizado y racional y una filosofía de documentación WYSIWYG (What You See Is What You Get, lo que ve es lo que obtiene), es decir, durante la entrevista se trabajará sobre lo que se generará. Tras una fase de preparación del JAD al caso concreto, el equipo de trabajo se reúne en varias sesiones. En cada una de ellas se establecen los requisitos de alto nivel a trabajar, el ámbito del problema y la documentación. Durante la sesión se discute en grupo sobre estos temas llegándose a una serie de conclusiones que se documentan. En cada sesión se van concretando más las necesidades del sistema. Esta técnica presenta una serie de ventajas frente a las entrevistas tradicionales ya que ahorra tiempo al evitar que las opiniones de los clientes se tengan que contrastar por separado. Pero requiere un grupo de participantes bien integrados y organizados.

### **1.7.4. Cuestionarios y Checklists**

Esta técnica requiere que el analista conozca el ámbito del problema en el que está trabajando. Consiste en redactar un documento con preguntas cuyas respuestas sean cortas y concretas, o incluso cerradas por unas cuantas opciones en el propio cuestionario (Checklist). Este cuestionario será cumplimentado por el grupo de personas entrevistadas o simplemente para recoger información en forma independiente de una entrevista.

### **1.7.5. Comparación de terminología**

Uno de los problemas que surge durante la elicitación de requisitos es que usuarios y expertos no llegan a entenderse debido a problemas de terminología. Esta técnica es utilizada en forma complementaria a otras para obtener consenso respecto de la terminología a ser usada en el proyecto de desarrollo. Para ello es necesario identificar el uso de términos diferentes para los mismos conceptos (correspondencia), misma terminología para diferentes conceptos (conflictos) o cuando no hay concordancia exacta ni en el vocabulario ni en los conceptos (contraste) (ESCALONA 2003).

Existen otras técnicas complementarias o de apoyo como la observación, el estudio de documentación, los cuestionarios, la inmersión en el negocio del cliente o haciendo que los analistas de sistemas o ingenieros de requisitos sean aprendices del cliente.

## CAPÍTULO 2 SOLUCIÓN PROPUESTA

En este capítulo se realizará la selección de la metodología que se empleará durante todo el proceso de desarrollo de Predictor. Además se definirá la herramienta CASE que servirá para modelar los artefactos generados del proceso de análisis del sistema así como los patrones que se tendrán en cuenta a la hora de confeccionar el modelo de casos de uso del sistema.

### 2.1. Selección de la Metodología de Desarrollo de Software

Después de haber realizado un estudio de las diferentes metodologías más usadas en la actualidad se puede decir que la más indicada para el desarrollo de un sistema de gestión como el que se implementará es la metodología RUP puesto que constituye uno de los estándares internacionales que más aceptación ha tenido en los últimos quince años en el desarrollo informático de nuestra humanidad. Además de ello cuenta con una herramienta CASE robusta, que utiliza un proceso de desarrollo iterativo controlado, permite el trabajo en equipo y es capaz de generar código en distintos lenguajes de programación a partir de un diseño UML.

Además según (KRUCHTEN 1996) es una metodología que se encarga de:

- Asegurar la producción de un software de alta calidad que reúna las necesidades de los usuarios finales dentro de un plan y un presupuesto predecible.
- Proveer un enfoque disciplinado para asignar tareas y responsabilidades dentro del desarrollo del sistema.
- Proveer un camino metódico, sistemático para desarrollar, diseñar y validar una arquitectura.
- Reducir en gran medida el riesgo que representa la construcción de sistemas complejos, porque evoluciona de forma incremental partiendo de sistemas más pequeños en los que ya se tiene confianza.

## 2.2. Selección de herramientas a utilizar según la metodología

A medida que los sistemas que hoy se construyen se tornan más y más complejos, las herramientas de modelado con UML ofrecen muchos beneficios para todos los involucrados en un proyecto, por ejemplo, administrador del proyecto, analistas, arquitectos, desarrolladores y otros. Las herramientas CASE<sup>6</sup> de modelado con UML nos permiten aplicar la metodología de análisis y diseño orientados a objetos y abstraernos del código fuente, en un nivel donde la arquitectura y el diseño se tornan más obvios y más fáciles de entender y modificar. Cuanto más grande es un proyecto, es más importante utilizar una herramienta CASE. Al usar las herramientas CASE:

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

El modelo actúa como el plano y guiará finalmente la construcción del sistema. De manera semejante, la administración es capaz de ver, puntualmente y desde un alto nivel, una representación del diseño y comprender lo que está sucediendo. Por estas razones, las herramientas CASE de UML acompañadas con metodologías, nos brindan una forma de representar sistemas demasiados complejos para comprenderlos a través de su código fuente subyacente y nos permiten desarrollar la solución de software correcta más rápido y más económicamente.

---

<sup>6</sup> Cuando se integran herramientas para que la información creada por una herramienta la pueda utilizar otra, se establece un sistema de soporte para el desarrollo del software llamado *ingeniería del software asistida por computadora* (CASE) (PRESSMAN 1998).



La herramienta CASE que se escogió para el modelamiento y refinamiento de los artefactos que se obtienen del análisis del Sistema es *Rational Rose Enterprise Edition*; una herramienta que es reconocida como el líder tecnológico por su rol en el desarrollo del UML, logrado en gran parte por los esfuerzos de Grady Booch, Ivar Jacobson, y Jim Rumbaugh, los tres más importantes autores del UML.

### **2.2.1. Rational Rose Enterprise Edition**

Rational Rose Enterprise Edition propone la utilización de cuatro tipos de modelo para realizar un diseño del sistema, utilizando vistas estática, dinámica, lógica y física de los modelos del sistema. Permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software.

Permite visualizar, entender, refinar los requerimientos y la arquitectura antes del enfrentamiento con el código. Esto permite evitar esfuerzos innecesarios en el ciclo de desarrollo. Si se utiliza una sola herramienta de modelamiento a través del ciclo de vida del desarrollo de software se podrá asegurar que se está construyendo el sistema correcto. El modelo arquitectónico puede ser rastreado hacia el modelo de procesos de negocios y los requerimientos de sistema.

Utiliza un proceso de desarrollo iterativo controlado, donde el desarrollo se lleva a cabo en una secuencia de iteraciones. Cada iteración comienza con una primera aproximación del análisis, diseño e implementación para identificar los riesgos del diseño, los cuales se utilizan para conducir la iteración, primero se identifican los riesgos y después se prueba la aplicación para que éstos se hagan mínimos.

Cuando la implementación pasa todas las pruebas que se determinan en el proceso, ésta se revisa y se añaden los elementos modificados al modelo de análisis y diseño. Una vez que la actualización del modelo se ha modificado, se realiza la siguiente iteración. Rational Rose Enterprise Edition permite que haya varias personas trabajando a la vez en el proceso iterativo controlado, para ello posibilita que cada desarrollador opere en un espacio de trabajo privado que contiene el modelo completo y tenga un control exclusivo sobre la propagación de los cambios en ese espacio de trabajo.

Se puede generar código en distintos lenguajes de programación a partir de un diseño en UML. Rational Rose Enterprise Edition proporciona mecanismos para realizar la denominada Ingeniería Inversa, es decir, a partir del código de un programa, se puede obtener información sobre su diseño.

### **2.3. Modelo del negocio**

El primer artefacto propuesto por el proceso unificado de desarrollo de software es el Modelo de Negocio. Este modelo permite comprender la estructura y dinámica de una organización, identificar los problemas actuales e identificar mejoras potenciales para eliminar esos problemas. Mediante él se persigue asegurar que los clientes y desarrolladores tengan una misma idea de las actividades que tienen lugar en la organización.

El Modelo de Negocio está comprendido dentro de la fase de inicio del proceso unificado y dentro del flujo de trabajo modelamiento del negocio, que es donde el analista se detiene a entender el contexto mediante el cual el sistema se va a ubicar además de identificar las posibilidades de mejora de los procesos.

#### *Especificación de las reglas del negocio*

En una organización, tanto los procesos como los datos que estos manejan, están restringidos por las reglas del negocio. Estas reglas aseguran que la actividad de la empresa se lleva a cabo de acuerdo a restricciones impuestas desde el entorno (leyes o normas) o desde dentro de la propia organización.

A continuación se muestran las reglas del negocio que han sido identificadas:

- La consultoría Delfos, entidad rectora de la vigilancia tecnológica en el MIC, identifica las Tecnologías de Información y Comunicaciones (TICs) con mayor potencial que se encuentran en fase de investigación y desarrollo.
- Los expertos en TICs valoran las tecnologías identificadas y señalan las que se adecuan mejor a nuestra realidad nacional, las que contribuyan al fortalecimiento de las mismas y a alcanzar la soberanía tecnológica.
- Los grupos consultores del sistema de vigilancia tecnológica del MIC realizan estudios para determinar el estado del arte de las tecnologías identificadas como de mayor potencial para su implementación.

- Los documentos de patentes contienen información técnica, comercial y legal de mucha importancia para el desarrollo de estudios de vigilancia tecnológica.
- Se realiza la búsqueda, descarga, homogeneización, almacenamiento y análisis de la información de bases de datos de patentes como Espacenet, World Wide (International), United States Patent and Trademark Office (USPTO), State Intellectual Property Office (SIPO) de China y Patent Abstract of Japan (PAJ), de forma automatizada para lograr la eliminación de errores en este proceso y así permitir tiempos más rápidos de respuesta a los clientes.
- Se realizan estudios que se generan de la información de patentes para contribuir estratégicamente al proceso de toma de decisiones de especialistas y directivos de las organizaciones del MIC.

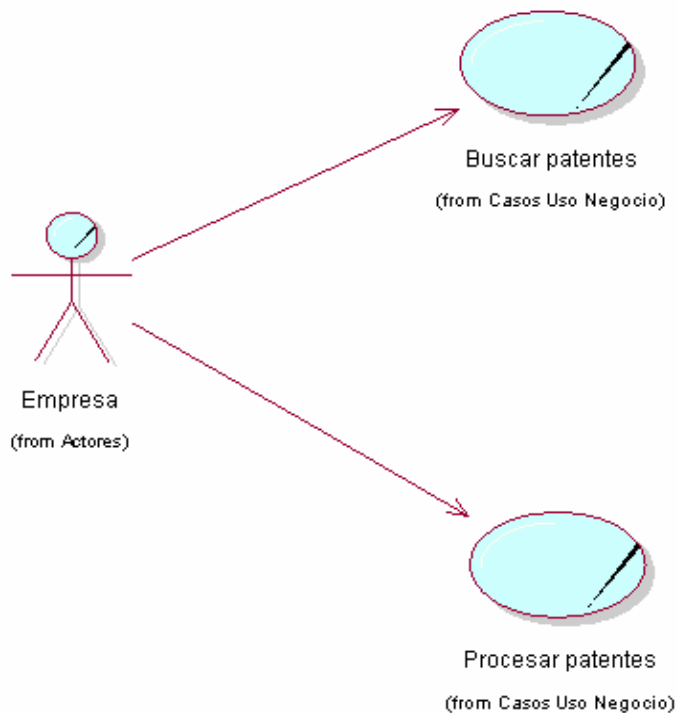
#### *Descripción general del Negocio*

Actualmente la consultoría Delfos orienta su actividad a la gestión de información para el MIC, como actividad fundamental de la organización y que da respuesta directa al objeto social aprobado. La misma, considerando que las ciencias de la información alcanza internacionalmente estadios superiores de desarrollo y que el papel del gestor de información se fortalece con el progreso de la sociedad de la Información y las TICs, abarca las perspectivas de la gestión de información desde los estudios en temas tecnológicos de interés para el MIC, como son las tendencias basadas en el análisis de grandes volúmenes de información de patentes que son descargadas de bases de datos disponibles en Internet.

Para realizar este análisis los especialistas de Delfos toman la información necesaria sobre las patentes (título, resumen, autor, clasificación internacional, etc.) y las organizan en ficheros que luego abren en Excel o Procite. Esto constituye un proceso lento y no permite en muchas ocasiones la entrega a tiempo de la solicitud realizada por la empresa que necesita la información.

Para tener una visión general de los procesos de negocio de la organización con fronteras bien establecidas, puede construirse un *diagrama de casos de uso del negocio*, en el cual aparece cada proceso del negocio como un caso de uso. Este diagrama permite mostrar los límites y el entorno de la organización bajo estudio. Por esta razón, sólo aparecerán en este diagrama los actores del negocio correspondientes a los roles externos al sistema, de forma que los procesos de negocio en los que sólo tomen parte roles internos a la organización no estarán conectados a ningún actor. En la Figura 4 se

muestra el diagrama de casos de uso del negocio para nuestro proyecto; es un diagrama de casos de uso UML formado por casos de uso del negocio y el actor *Empresa* (entidad que se beneficia con estos procesos) quien arranca la realización de los casos de uso Buscar patentes y Procesar patentes.



**Figura 4: Diagrama de casos de uso del Negocio**

Si no se hubieran definido claramente los procesos del negocio, con sus actores y trabajadores, entonces hubiera sido necesario identificar conceptos y tratar de relacionarlos en un modelo de objetos, el cual es llamado Modelo de Dominio.

Para la descripción de los casos de uso del negocio de manera que se pueda ofrecer más información sobre los procesos que ocurren en el mismo se utilizó la tabla de descripciones que ofrece el Rational Rose en una de sus plantillas. (Anexo 2)

**Tabla 1: Actor del Negocio**

Actor	Descripción
Empresa	Es una empresa o cualquier institución que necesite de la información relacionada con la temática de interés que pueda ofrecer la Consultoría Delfos. Dicha empresa es quien inicia todas las acciones que dan comienzo a los procesos de negocio analizados, y al mismo tiempo es el principal beneficiado con el resultado de dichos procesos.

**Tabla 2: Trabajador del Negocio**

Trabajador	Descripción
Especialista	Es el encargado de conectarse a las Bases de Datos de Internet para realizar la búsqueda, descarga y procesamiento de las patentes relacionadas con la información que necesita el actor del negocio.

#### 2.4. Descripción de los casos de uso del negocio

**Tabla 3: Descripción del caso de uso Buscar Patentes**

Caso de Uso:	Buscar patentes
Actores:	Empresa
Trabajadores:	Especialista
Resumen:	El caso de uso se inicia cuando una empresa realiza una solicitud a la consultoría Delfos sobre patentes determinadas. El Especialista se conecta a las BD disponibles y descarga la información relacionada con el criterio de búsqueda.

Precondiciones:	Que se realice la solicitud de la empresa.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
1. La Empresa interesada realiza la solicitud de la búsqueda de patentes.	<p>1.1 El Especialista se conecta a la base de datos donde se realizará la búsqueda.</p> <p>1.2 El Especialista introduce el criterio de búsqueda relacionado con la solicitud de la empresa.</p> <p>1.3 El Especialista descarga y almacena la información obtenida de la búsqueda.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
<i>Acción 1.2</i>	Si el término introducido no es correcto, no se obtienen resultados en la búsqueda.
Poscondiciones	Resultado de la búsqueda relacionado con un criterio determinado

Diagrama de Actividades

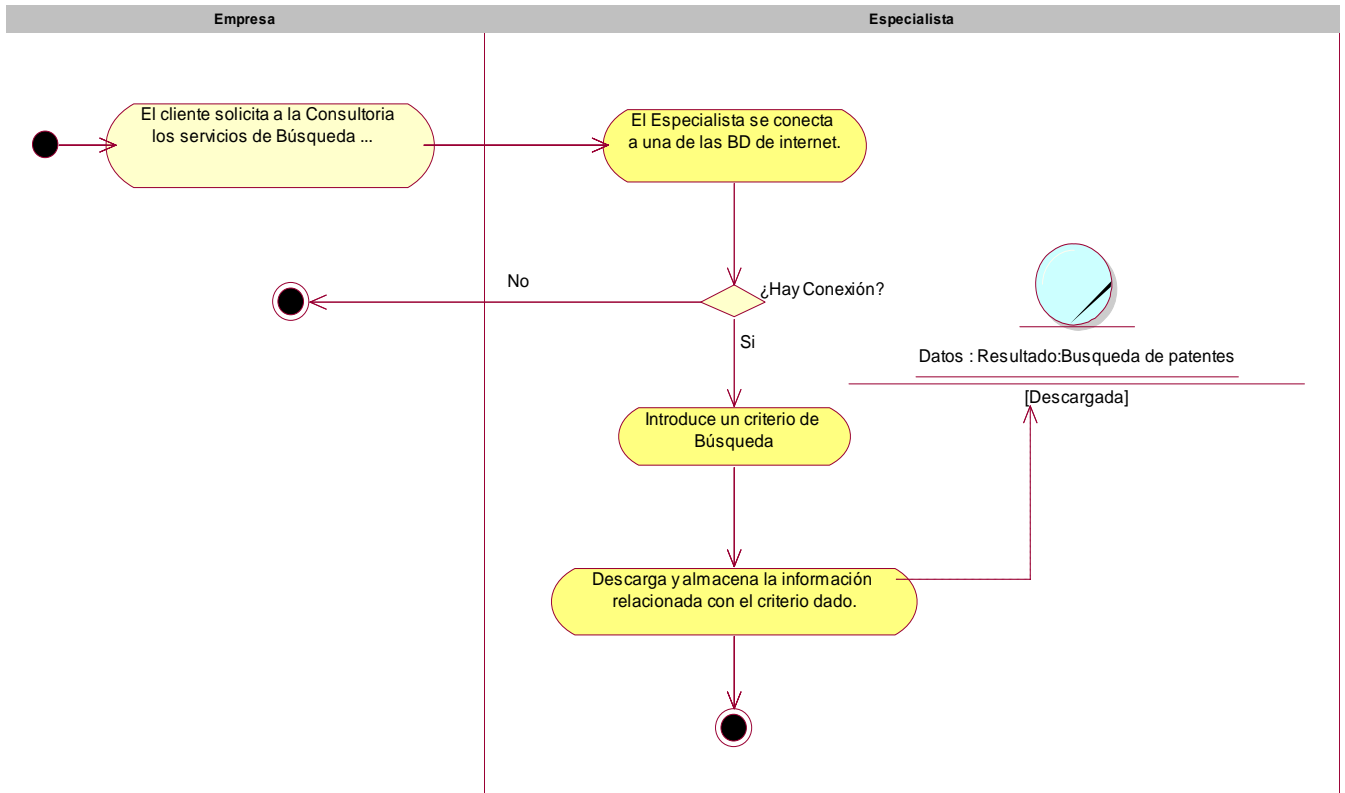


Figura 5: Diagrama de actividades del caso de uso del negocio Buscar patentes

Tabla 4: Descripción del caso de uso del negocio Procesar patentes

Caso de Uso:	Procesar patentes
Actores:	Empresa
Trabajadores:	Especialista
Resumen:	El caso de uso se inicia cuando el Especialista después de obtener la información toma los datos que le interesan de la patente (título, resumen, fecha de invención, etc.) mediante la homogeneización y filtrado de los documentos y esta forma llevar a cabo los

	estudios de mercado, análisis de tendencia y otros.	
Precondiciones:	Que se realice la solicitud de la empresa.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1. La Empresa realiza la solicitud de los resultados.	1.1. El Especialista realiza el procesamiento de la información almacenada.  1.2. El Especialista realiza los estudios y análisis de las patentes.	
Flujos Alternos		
Acción del Actor	Respuesta del Negocio	
Poscondiciones	Información procesada	



## Diagrama de Actividades

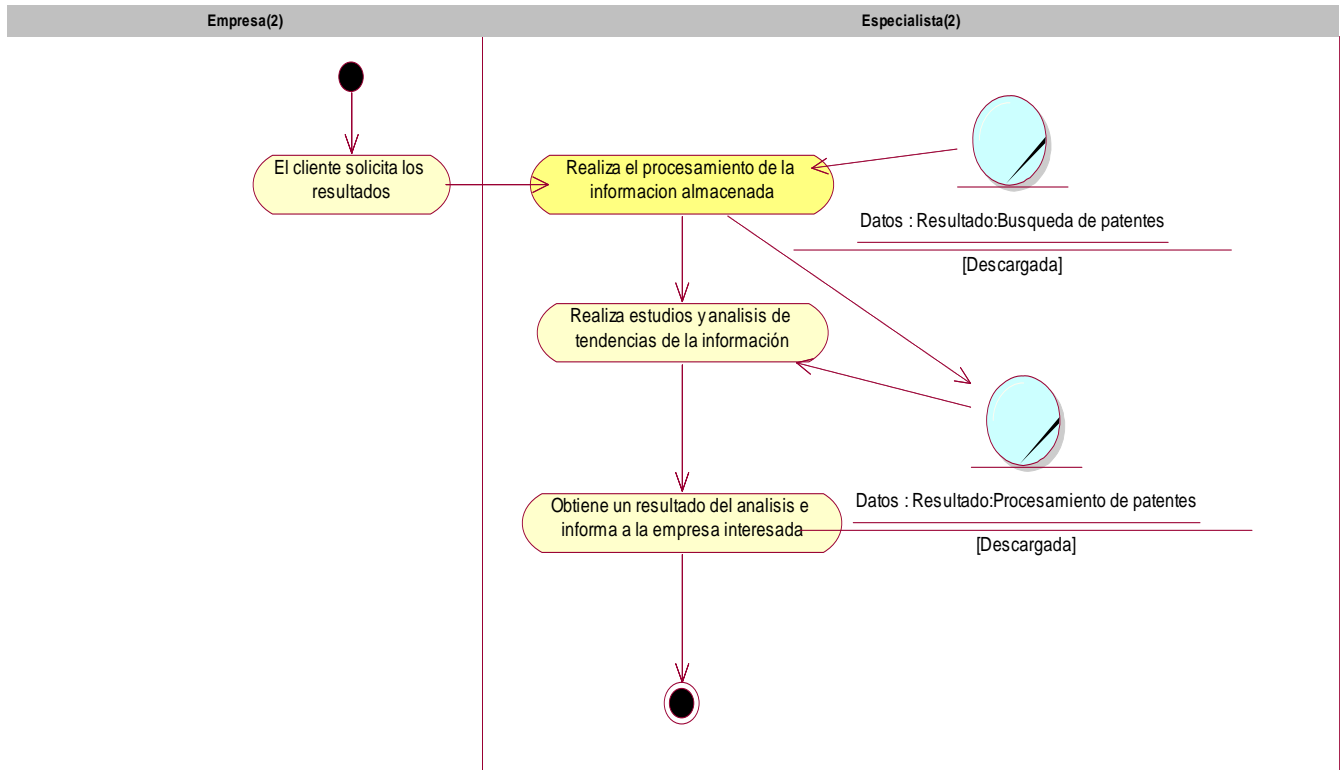


Figura 6: Diagrama de actividades del caso de uso Procesar patentes

## 2.5. Captura de requisitos. Técnicas, métodos y plantillas a utilizar

Una de las etapas cruciales cuando se concibe un sistema informático reside en captar y mostrar de forma apropiada los requisitos de usuario, y eso viene dado por el efecto que tiene en el resto del ciclo de desarrollo del software.

Primeramente se realizó la Tormenta de Ideas, técnica que puede ayudar a generar una gran variedad de vistas del problema y a formularlo de diferentes formas, sobre todo al comienzo del proceso de elicitación, cuando los requisitos son todavía muy difusos. Mediante la misma se pudo obtener una visión global de los requisitos que ayudaron a establecer un criterio común entre usuarios y desarrolladores sobre las funcionalidades del Sistema

Posteriormente se realizaron las entrevistas como una de las formas más efectivas de obtención de información de las especificaciones del negocio y por tanto, de una mejor comprensión del problema y de las necesidades específicas del cliente, por lo que constituyó una técnica de vital importancia en la elicitación de los requisitos del Sistema. Para ello se comenzó por los directivos de Delfos, ya que son los pueden ofrecer una visión detallada del negocio; se continuó con los futuros usuarios (especialistas), y con el personal técnico, quienes aportaron detalles sobre el entorno operacional de la organización.

Para la elaboración de los diferentes procedimientos y artefactos de este trabajo de tesis se utilizaron las plantillas definidas por el Proceso Unificado del Software (RUP), ya que mediante estas se puede lograr una mayor organización del trabajo, queda constancia de todo lo que se hizo y se le viabiliza el trabajo al personal del equipo de calidad, para que hagan las revisiones necesarias y de esta manera contribuir a obtener un producto de alta calidad

De las plantillas que ofrece el Rational Rose se utilizaron las siguientes:

- Modelo del Negocio (Anexo 1)
- Realización de los Casos de Uso del Negocio (Anexo 2)
- Especificaciones de requisitos de software (Anexo 3)
- Especificaciones de casos de uso del sistema (Anexo 4)
- Prototipo de interfaz de usuario (Anexo 5)
- Documento visión
- Glosario de términos

## **2.6. Especificación de requisitos del software**

Los requisitos generales del producto se obtienen del cliente. Estos requisitos comprenden necesidades de información y control, funcionalidad del producto y comportamiento, rendimiento general del producto, diseño, restricciones de interfaz y otras necesidades especiales (PRESSMAN 1998d).

Después que se conocen dichos requisitos la misión del analista es otorgarle la funcionalidad a los componentes que conforman el sistema. Para realizar este proceso y asegurar el buen funcionamiento del mismo se precisó de la participación activa de los especialistas y directivos de la consultoría Delfos.

### *Descripción general del Sistema*

Para que se pueda ejercer una mejor vigilancia tecnológica se requiere de la informatización de los procesos de búsqueda, descarga y procesamiento de patentes realizados por los especialistas de dicha institución. Estos procesos se realizan de forma manual, por lo que se desea diseñar una aplicación Windows que permita ejecutar de una forma rápida y eficiente dicha actividad y que de esta forma puedan analizar la información con la calidad y el tiempo requeridos.

Se desea que la aplicación permita al especialista la conexión a varias bases de datos como son: Word Wide (International), Espacenet, United States Patent and Trademark Office (USPTO), State Intellectual Property Office (SIPO) de China y Patent Abstract of Japan (PAJ) para realizar la búsqueda de patentes, la cual puede realizarse de forma básica o avanzada, en dependencia de la necesidad de información del usuario. Para ello el especialista debe autenticarse antes en el sistema y poder configurar la conexión a Internet, ya que esta puede efectuarse mediante un Proxy y si posee una cuota de Internet podrá hacer uso de ella mientras no exceda del límite de la misma.

Después de realizar la búsqueda, la aplicación deberá permitir el procesamiento de la información descargada, mediante la eliminación del código html seleccionando luego los parámetros de interés (titulo de la patente, resumen, autor, etc.). Todas las patentes descargadas de un término específico deben guardarse en un fichero especificado por el usuario para luego abrir con el Excel y Procite.

Además de ello se le propuso poder realizar el procesamiento de la información guardada momentos antes de tener la aplicación ejecutándose, por lo que si tienen archivos de patentes sin procesar podrán hacerlo con la ayuda de la aplicación disminuyendo considerablemente el tiempo en que demorarían si lo hicieran manualmente.

El Administrador así como el Especialista podrá autenticarse en el sistema. Este será el encargado de gestionar la información referente a las bases de datos disponibles en Internet y de los usuarios que usen

la aplicación, estableciendo un control sobre la misma, ya sea para adicionar, eliminar o modificar los datos.

La aplicación Windows estará basada en una arquitectura multicapas, y la misma será implementada en c#. El sistema gestor de bases de datos sería SQL Server. Además se utilizaría como componente de abstracción a datos la biblioteca ADOdb.

*A continuación se muestran los requisitos funcionales y no funcionales del Sistema.*

### **2.6.1. Requerimientos Funcionales**

El Sistema debe ser capaz de:

- R1 - Controlar que cada usuario acceda a la información que le corresponda.
- R1.1- Definir niveles de acceso.
- R2 - Configurar la conexión a Internet.
- R3 - Hacer búsquedas en cualquiera de las bases de datos disponibles en Internet que contengan información sobre patentes.
- R4 - Realizar búsqueda básica.
- R5 - Realizar búsqueda avanzada.
- R6 - Descargar la información acorde con el criterio de búsqueda que emplee el usuario.
- R7- Realizar el procesamiento de la información que se haya encontrado en las bases de datos de Internet.
- R8 - Guardar la información obtenida de la descarga, en el directorio seleccionado por el usuario.
- R9 - Permitir al administrador del sistema registrar los datos de los usuarios.
- R10 - Permitir al administrador del sistema actualizar los datos de los usuarios.
- R11 - Permitir al administrador del sistema eliminar los datos de los usuarios.

- R12 - Permitir al administrador del sistema adicionar a la aplicación una nueva base de datos de Internet.
- R13- Permitir al administrador del sistema actualizar la información de las bases de datos.
- R14 - Permitir al administrador del sistema eliminar la base de datos de Internet que elija.
- R15 - Mostrar un reporte de las búsquedas realizadas.

### **2.6.2. Requerimientos no Funcionales**

#### *Requerimientos no funcionales de Diseño e Implementación*

- En la implementación del sistema se utilizará un estándar de codificación.
- Se utilizarán diferentes herramientas para desarrollo del sistema como son: Rational Rose 2003, MS SQL Server 2000, Visual Studio.NET 2005
- Las bibliotecas de clases que se emplearán serán: The Base Class Libraries (BCL) y [msdn2.microsoft.com/en-us/netframework/aa569603.aspx](http://msdn2.microsoft.com/en-us/netframework/aa569603.aspx).

#### *Requerimientos no funcionales de Software*

- El sistema operativo sobre el que se trabajará será Windows XP o superior.
- En la PC donde se encuentre el Servidor de Bases de Datos debe tener instalado el Microsoft SqlServer 2000.
- Las PCs donde se ejecute el Sistema debe tener instalado el FrameWork. NET 2.0

#### *Requerimientos no funcionales de Hardware*

- Es necesario que se cuente con una red local donde existan al menos dos computadoras. En una se ejecutará el Sistema y en la otra se encontrará el Servidor de Base de Datos por razones de seguridad.
- En la computadora donde esté el Sistema debe haber conexión a Internet.

- Es recomendado para el Sistema Operativo una PC PENTIUM III o superior con 300 MHz como mínimo, con 128 MB de RAM, y mínimo 20 GB de disco duro.
- La comunicación entre las máquinas donde estará el sistema y el Servidor de Base de Datos será a través del protocolo de TCP/IP.

*Requerimientos no funcionales de Portabilidad*

- El software podrá ser usado sólo bajo los sistemas operativos Windows XP o superior.

*Requerimientos no funcionales de Seguridad.*

- Solo las personas autorizadas podrán tener acceso a la información. Las únicas personas autorizadas son el Especialista y el Administrador del Sistema.

*Requerimientos no funcionales de Rendimiento*

- Rápido acceso de búsqueda a la información en tiempos relativamente cortos (aproximadamente 150 resultados en 10 minutos), dependiendo de la conexión a Internet.

*Requerimientos no funcionales de Soporte*

- Para garantizar el soporte a los clientes de la Consultoría Delfos, existirá la posibilidad de que ellos emitan sus quejas y sugerencias a los desarrolladores del Sistema, ya sea por correo electrónico o por cualquier otra vía de comunicación por un periodo de 1 mes.

*Requerimientos no funcionales de Usabilidad*

- El Sistema contará con un Manual de Usuario que servirá para guiar a los especialistas en el manejo del Sistema.

*Requerimientos Adicionales*

- El usuario podrá hacer cambios en cuanto a los colores de la interfaz configurando esta a su gusto.

## 2.7. Aplicación de patrones del análisis en el modelo del sistema

Para desarrollar el modelo del sistema se aplicaron patrones de casos de uso o escenarios, por la importancia que tienen a la hora de conformar los casos de uso y con el objetivo de reflejar los requisitos funcionales definidos por el cliente y los desarrolladores de la forma más exacta posible y así lograr mejores resultados en el producto a desarrollar. Los patrones aplicados son los siguientes:

### *Concordancia: Especialización*

El patrón Especialización se aplicó en el Modelo del Sistema mediante los casos de uso *Realizar búsqueda avanzada* y *Realizar búsqueda básica*, especializaciones del caso de uso *Buscar patentes*, pues son casos de uso de una misma clase y heredan todas las acciones del caso de uso de tipo común (caso de uso *Buscar patentes*).

### *Múltiples actores: Rol común*

Se aplicó este patrón, ya que existen dos actores del sistema (Administrador y Especialista), que dan inicio a los mismos casos de uso, esto ocurre ya que el Administrador es a la vez un Especialista y por tanto hereda el comportamiento de este rol. Este patrón permite que varios actores inicien uno o varios casos de uso mediante un mismo rol, que en este caso es el *Especialista*.

### *Extensión o inclusión concreta: Inclusión.*

Como se explicó en el capítulo anterior este patrón se aplica cuando un flujo puede ser incluido dentro del flujo de un caso de uso, así como ejecutarse dentro de él mismo. En este caso es necesario porque en el Modelo del Sistema el comportamiento definido para el caso de uso *Procesar patentes* es un proceso que se inserta explícitamente dentro del comportamiento definido para el caso de uso *Buscar patentes* y puede ejecutarse dentro de su flujo.

CRUD

Este patrón es aplicado en los casos de uso *Gestionar Datos de Usuario* y *Gestionar Bases de Datos*, ya que en estos se agrupan las funciones básicas (Adicionar, modificar y eliminar) en una sola haciendo el modelo más claro para el analista.

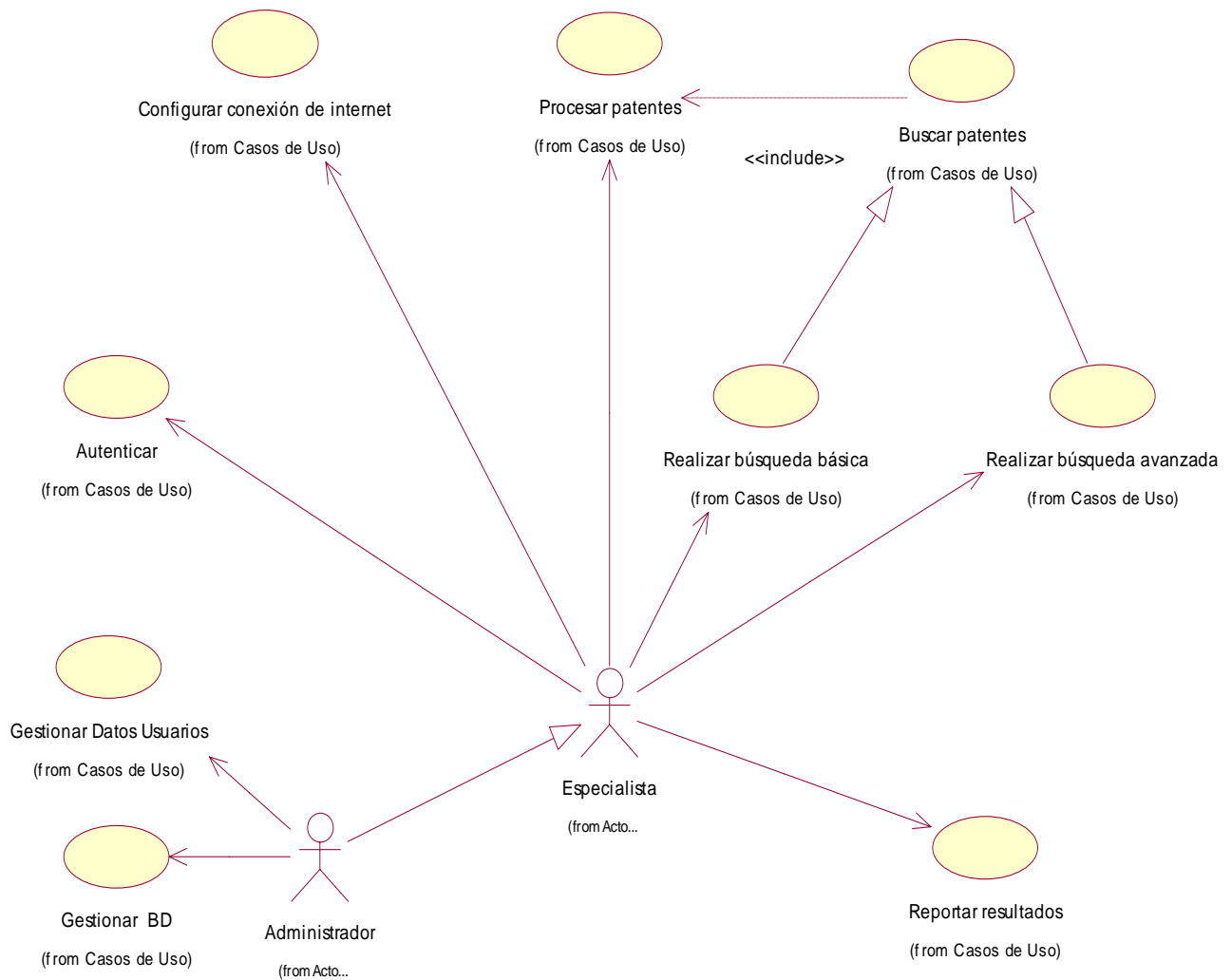


Figura 7: Diagrama de Casos de Uso del Sistema

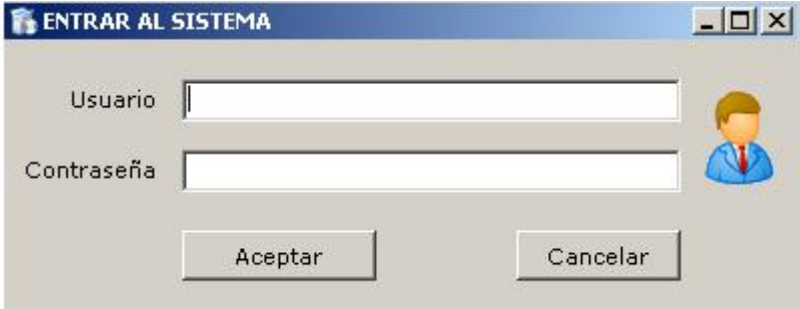


## 2.8. Descripción de los casos de uso del sistema

A continuación se muestra la descripción de los casos de uso expandidos ya que pueden ser útiles para alcanzar un conocimiento más profundo de los procesos y de los requerimientos. Cada caso de uso se describirá mediante una plantilla que puede rellenarse a partir de la especificación de la actividad asociada. La plantilla que se propone está basada en RUP, (Anexo 4).

**Tabla 5: Descripción del caso de uso Autenticar**

Caso de Uso:	Autenticar	
Actores:	Especialista (inicia)	
Resumen:	El Caso de Uso se inicia cuando el Especialista introduce los datos que se le piden para acceder a la aplicación, estos se verifican y finaliza el caso de uso dándole los permisos y habilitándole la entrada.	
Precondiciones:		
Referencias	R1	
Prioridad	Primario	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El Especialista introduce:  Usuario:  Contraseña:	1.1. El sistema encripta la contraseña.  1.2. Busca el usuario y compara la contraseña.  1.3. En caso de ser correcto se le asignan los permisos.	

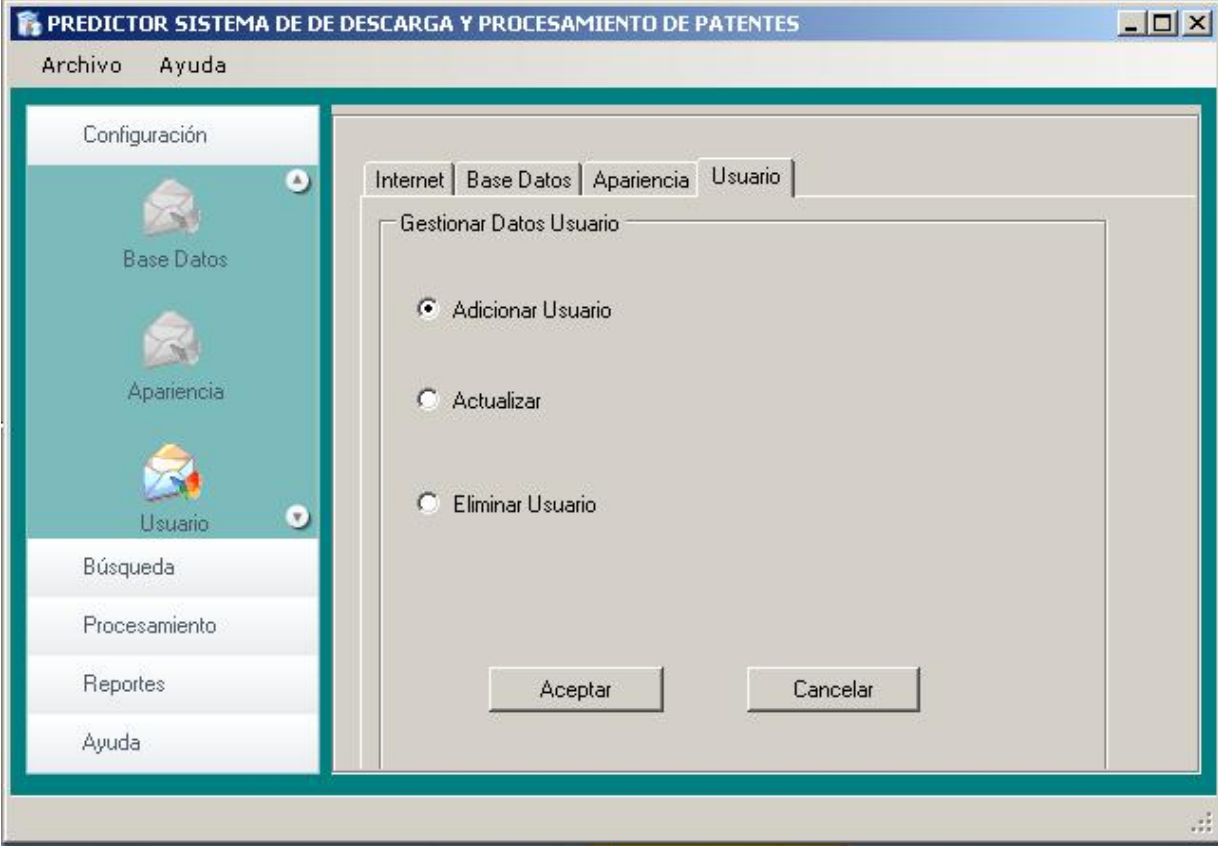
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Acción 1.3	En caso de ser incorrectos los datos, se envía un mensaje de aviso.
<b>Prototipo de Interfaz</b>	
	
Poscondiciones	Se habilitan las funcionalidades según los privilegios.

**Tabla 6: Descripción del caso de uso Gestionar Datos Usuario**

Caso de Uso:	Gestionar Datos Usuario
Actores:	Administrador (inicia)
Resumen:	El caso de uso se inicia cuando el Administrador de la aplicación selecciona la opción de Gestionar Datos Usuario, luego escoge la acción a realizar e introduce los datos necesarios. El sistema realiza la acción seleccionada por el Administrador y termina el caso de uso.
Referencias:	R9, R10, R11
Precondiciones:	El actor se ha autenticado previamente.
Prioridad	Secundario

Flujo Normal de Eventos	
1. El Administrador solicita la interfaz correspondiente a la gestión de los datos de los usuarios para adicionar, eliminar o modificar alguno de ellos.	1.1 El sistema muestra una interfaz visual mostrando las opciones: “Adicionar Usuario”, “Actualizar Datos Usuario” y “Eliminar Usuario”.
Sección “ Adicionar Usuario”	
Acción del Actor	Respuesta del Sistema
1. El Administrador selecciona la opción de Adicionar Usuario.	1.1 El sistema muestra el formulario a completar para la adición de un nuevo usuario.
2. El Administrador introduce los datos solicitados por el sistema.	2.1 El sistema verifica los datos introducidos por el Administrador.  2.2 Si los datos introducidos son correctos el sistema adiciona dicha información en la Base de Datos correspondiente y termina el caso de uso.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Acción 2.2	Si los datos introducidos por el Administrador son incorrectos el sistema muestra un mensaje de error indicando donde está el dato erróneo e indica al usuario retornar a la acción 2.
Flujo Normal de Eventos	
Sección “Actualizar Datos Usuario”	
Acción del Actor	Respuesta del Sistema
1. El Administrador selecciona la opción de Actualizar	1.1 El sistema muestra un listado con los usuarios

<p>Datos Usuario.</p> <p>2. El Administrador selecciona el usuario para la actualización.</p> <p>3. El Administrador realiza los cambios necesarios a los datos.</p>	<p>existentes en la Base de Datos.</p> <p>2.1 El sistema localiza los datos del usuario y los muestra, listos para actualizar.</p> <p>3.1 El sistema verifica los datos introducidos por el Administrador.</p> <p>3.2 Si los datos están correctos el sistema actualiza los datos del usuario en la Base de datos correspondiente.</p>
<p>Flujos Alternos</p>	
<p>Acción del Actor</p>	<p>Respuesta del Sistema</p>
<p>Acción 3.2</p>	<p>Si los datos introducidos por el Administrador son incorrectos el sistema muestra un mensaje de error indicando donde está el dato erróneo e indica al usuario retornar a la acción 3.</p>
<p>Flujo Normal de Eventos</p>	
<p>Sección “ Eliminar Usuario”</p>	
<p>Acción del Actor</p>	<p>Respuesta del Sistema</p>
<p>1. El Administrador selecciona la opción de Eliminar Usuario.</p> <p>2. El Administrador selecciona el usuario para la eliminación.</p> <p>3. El Administrador selecciona la opción Eliminar Usuario.</p> <p>4. El Administrador confirma si quiere o no Eliminar Usuario.</p>	<p>El sistema muestra un listado con los usuarios existentes en la Base de Datos.</p> <p>2.1 El sistema localiza los datos del usuario y los muestra, listos para eliminar.</p> <p>3.1 El sistema muestra un mensaje de advertencia para</p>

	<p>la acción a realizar.</p> <p>4.1 Si el Administrador acepta el sistema elimina los datos del usuario seleccionado y culmina el caso de uso.</p>
<p>Flujos Alternos</p>	
<p>Acción del Actor</p>	<p>Respuesta del Sistema</p>
<p>Acción 3.1:</p>	<p>Si el Administrador cancela la acción culmina el caso de uso sin ejecutar ninguna acción.</p>
<p style="text-align: center;">Prototipo de interfaz</p> 	
<p>Poscondiciones</p>	<p>Información del usuario adicionada a la Base de Datos.</p>

	<p>Información del usuario actualizada en la base de Datos.</p> <p>Información del usuario eliminada de la base de Datos.</p>
--	---

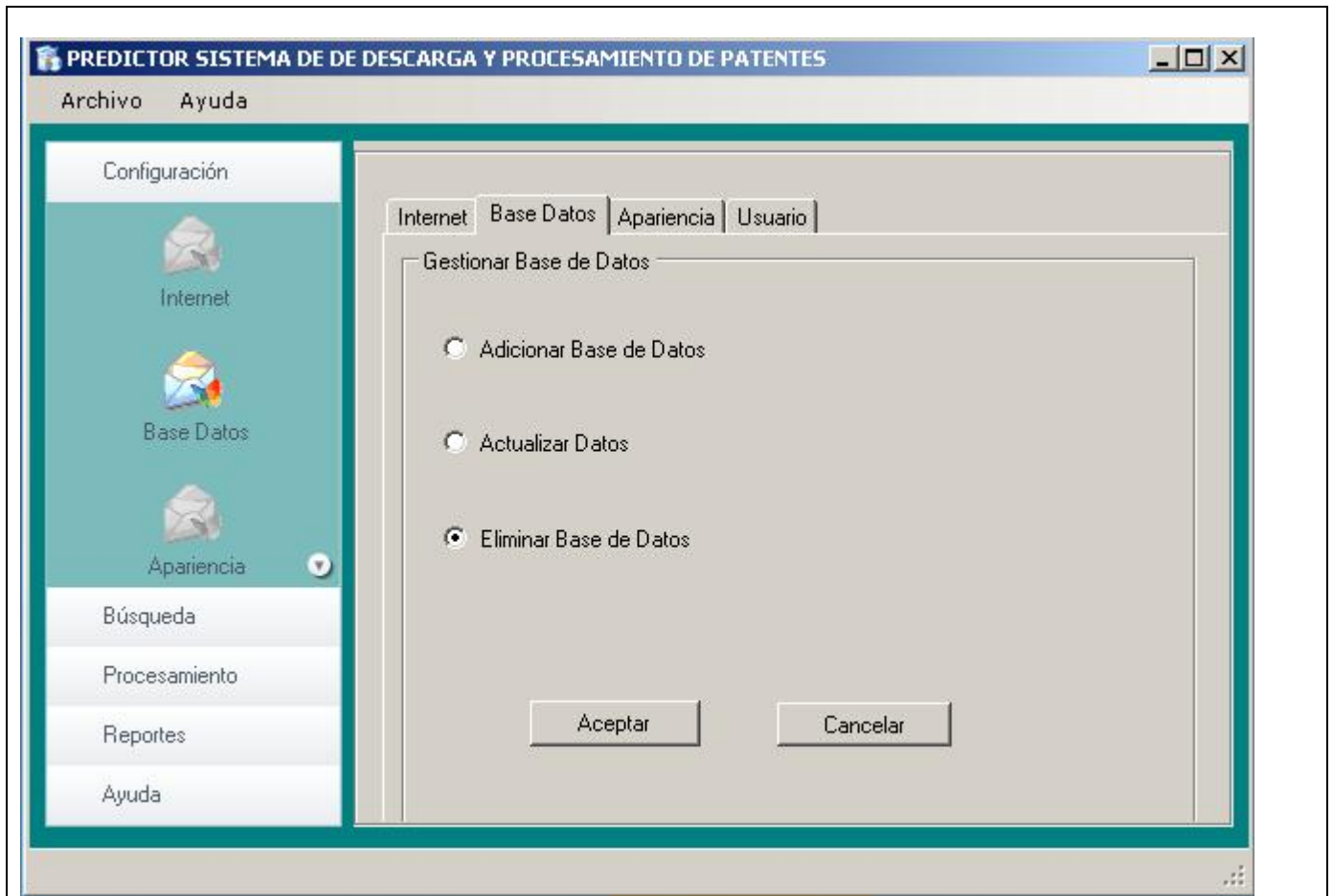
**Tabla 7: Descripción del caso de uso Gestionar Datos Usuario**

Caso de Uso:	Gestionar Bases de Datos	
Actores:	Administrador (inicia)	
Resumen:	El caso de uso se inicia cuando el Administrador de la aplicación selecciona la opción de Gestionar Bases de Datos, luego escoge la acción a realizar e introduce los datos necesarios. El sistema realiza la acción seleccionada por el Administrador y termina el caso de uso.	
Referencias:	R12, R13, R14	
Precondiciones:	El actor se ha autenticado previamente.	
Prioridad	Secundario	
Flujo Normal de Eventos		
1. El Administrador solicita la interfaz correspondiente a la gestión de las bases de datos para adicionar, eliminar o modificar alguno de ellos.	1. El sistema muestra una interfaz visual mostrando las opciones: "Adicionar Base de Datos", "Actualizar Base de Datos" y "Eliminar Base de Batos".	
Sección " Adicionar Base de Datos"		
Acción del Actor	Respuesta del Sistema	
1. El Administrador selecciona la opción de Adicionar Base de Datos.	1.1 El sistema muestra el formulario a completar para la adición de una nueva base de datos.	

2. El Administrador introduce los datos solicitados por el sistema.	<p>2.1 El sistema verifica los datos introducidos por el Administrador.</p> <p>2.2 Si los datos introducidos son correctos el sistema adiciona dicha Base de Datos y termina el caso de uso.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Acción 2.2	Si la url introducida por el Administrador es incorrecta el sistema envía un mensaje de error mostrando donde está el dato erróneo e indica al usuario retornar a la acción 2.
Flujo Normal de Eventos	
Sección "Actualizar Base de Datos"	
Acción del Actor	Respuesta del Sistema
<p>1. El Administrador selecciona la opción de Actualizar Base de Datos.</p> <p>2. El Administrador selecciona la base de Datos para la actualización.</p> <p>3. El Administrador realiza los cambios necesarios a los datos.</p>	<p>1.1 El sistema muestra un listado con las bases de datos existentes.</p> <p>2.1 El sistema localiza los datos de la misma y los muestra, listos para actualizar.</p> <p>3.1 El sistema verifica los datos introducidos por el Administrador.</p> <p>3.2 Si los datos están correctos el sistema actualiza los datos de la Base de Datos correspondiente.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Sistema

<i>Acción 3.2</i>	Si los datos introducidos por el Administrador son incorrectos el sistema muestra un mensaje de error indicando donde está el dato erróneo e indica al usuario retornar a la acción 3.
Flujo Normal de Eventos	
Sección “ Eliminar Base de Datos”	
Acción del Actor	Respuesta del Sistema
<p>1. El Administrador selecciona la opción de Eliminar Base de Datos.</p> <p>2. El Administrador selecciona la Base de Datos a eliminar.</p> <p>3. El Administrador selecciona la opción Eliminar Base de Datos.</p> <p>4. El Administrador confirma si quiere o no Eliminar la Base de Datos.</p>	<p>El sistema muestra un listado con las Bases de Datos existentes.</p> <p>2.1 El sistema localiza los datos de la misma y los muestra, listos para eliminar.</p> <p>3.1 El sistema muestra un mensaje de advertencia para la acción a realizar.</p> <p>4.1 Si el Administrador acepta el sistema elimina los datos de la base de datos seleccionada y culmina el caso de uso.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
<i>Acción 3.1:</i>	Si el Administrador cancela la acción se culmina el caso de uso sin ejecutar ninguna acción.
Prototipo de interfaz	





Poscondiciones

Información de la Base de Datos adicionada.

Información actualizada de la Base de Datos.

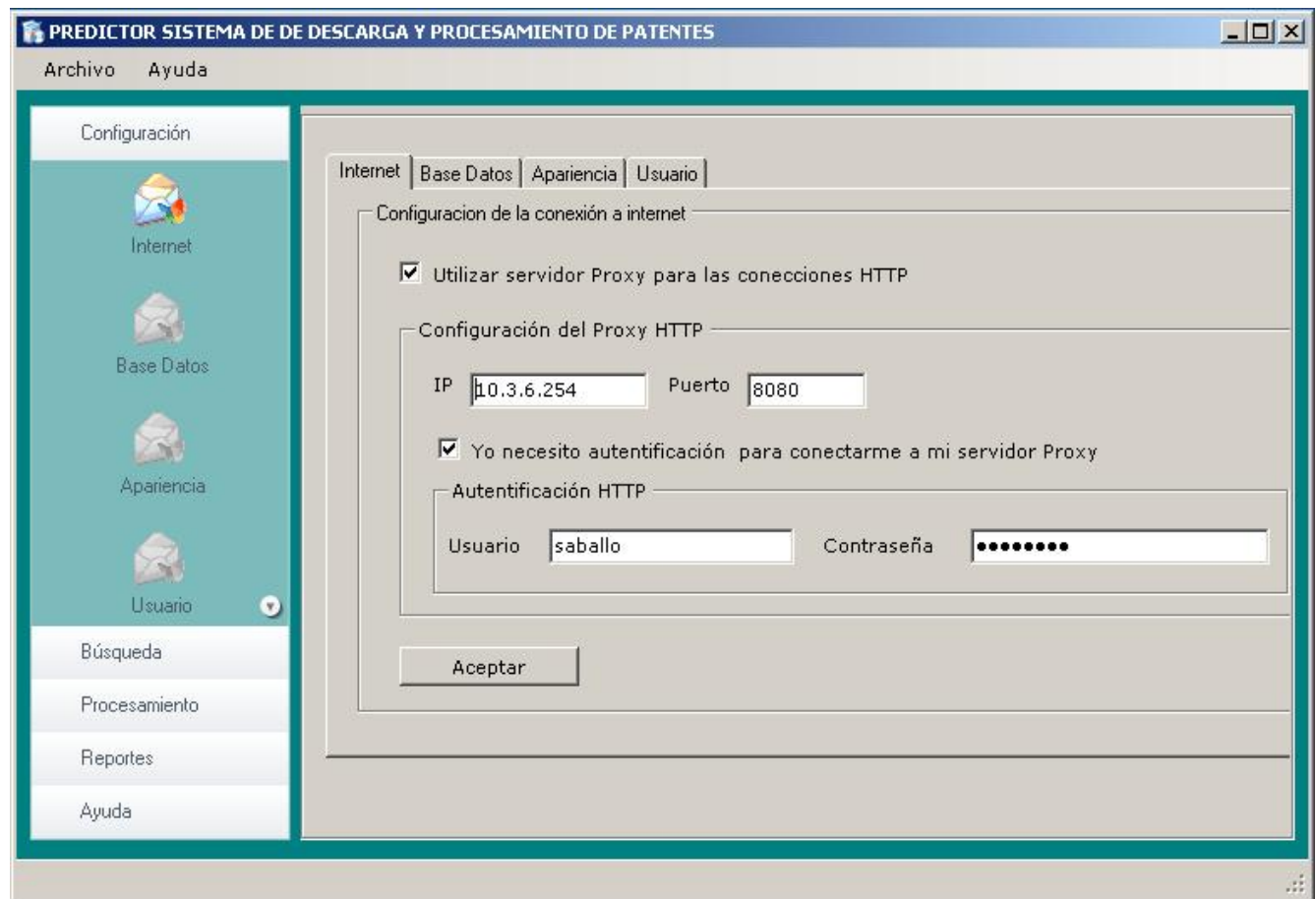
Información de la Base de Datos eliminada.

Tabla 8: Descripción del caso de uso Configurar conexión a Internet

Caso de Uso:	Configurar conexión a Internet	
Actores:	Especialista(inicia)	
Resumen:	El caso de uso se inicia cuando el Especialista introduce los datos que se le piden para configurar la conexión a Internet, estos se verifican y finaliza dándole los permisos y habilitándole la conexión.	
Precondiciones:	El Especialista debe conocer el tipo de conexión.	
Referencias	R2	
Prioridad	Primario	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. Si la conexión es directa el Especialista desmarca las opciones de configuración del servidor proxy y de usuario.	1.1. El sistema habilita la conexión.	
2. Si la conexión no es directa y el Especialista solo necesita configurar el servidor, introduce número de IP y número de puerto del proxy.	2.1 El sistema verifica los datos introducidos y habilita la conexión.	
3. Si su conexión debe realizarse mediante una cuenta de internet el Especialista introduce los datos referidos al servidor y los datos de usuario.	3.1. El sistema encripta la contraseña. 3.2. Busca el usuario y compara la contraseña. 3.3. En caso de ser correcto se le asignan los permisos y se habilita la conexión.	

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Acción 1.1 y Acción 2.2	En caso de no ser correctos se envía un mensaje de aviso.

*Prototipo de Interfaz*



Poscondiciones	Se habilita la conexión.
----------------	--------------------------

Tabla 9: Descripción del caso de uso Procesar patentes

Caso de Uso:	Procesar patentes	
Actores:	Especialista (inicia)	
Resumen:	El Caso de Uso se inicia cuando el Especialista decide realizar el procesamiento de patentes que han sido descargadas de Internet con el objetivo de realizar el análisis de las mismas y evaluar los resultados.	
Precondiciones:	El usuario se haya autenticado en el sistema.	
Referencias	R7	
Prioridad	Primario	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
<ol style="list-style-type: none"> <li>1. El Especialista selecciona la opción “Procesar Patentes”.</li> <li>2. El Especialista busca en el directorio la información que desea procesar.</li> <li>3. El Especialista selecciona la información.</li> <li>4. El Especialista selecciona la base de datos a la cual corresponde la información.</li> </ol>	<ol style="list-style-type: none"> <li>4.1 El sistema procesa la información (elimina los Tabs HTML de la patente).</li> <li>4.2 El sistema almacena el resultado del procesamiento sobrescribiendo el fichero donde se encontraba la información sin procesar.</li> </ol>	
Flujos Alternos		
Acción del Actor	Respuesta del Sistema	

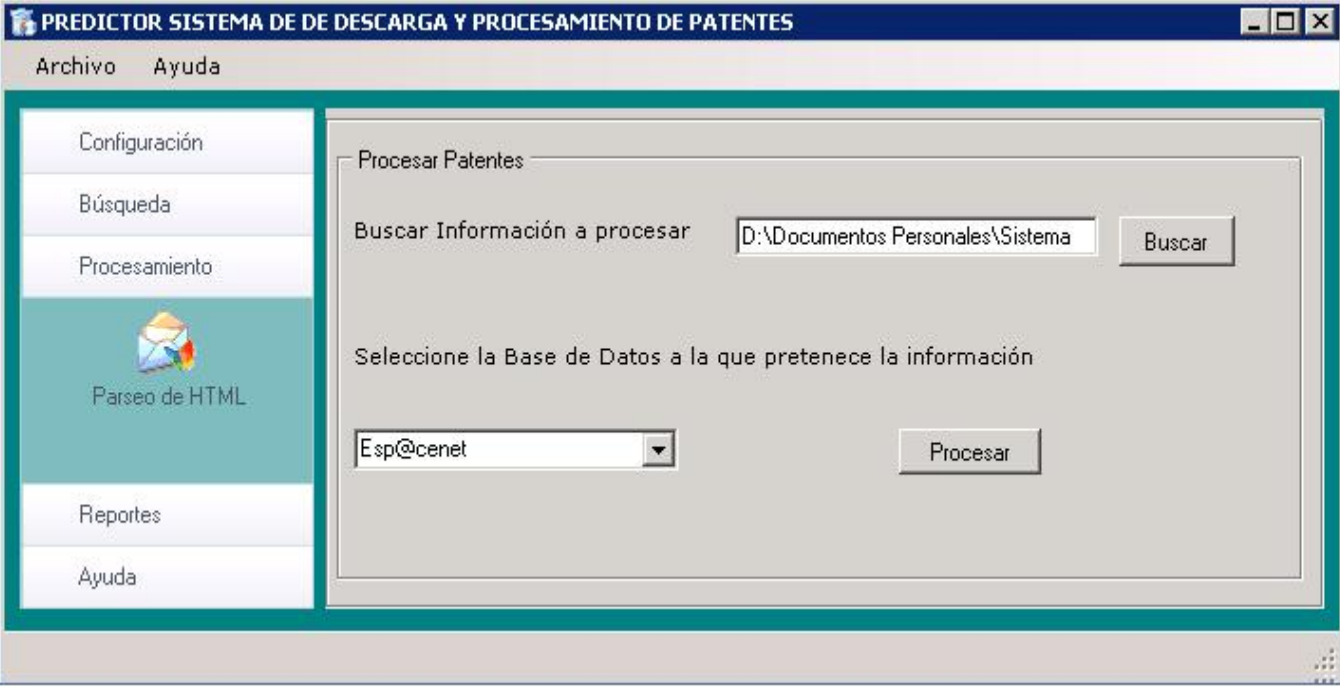
Acción 4.1	Si la información seleccionada no es una patente, el sistema envía un mensaje de error.
<p>Prototipo de Interfaz</p> 	
Poscondiciones	Se obtiene la información procesada.

Tabla 10: Descripción del caso de uso Buscar patentes

Caso de Uso:	Buscar patentes	
Actores:	Casos de uso “Realizar búsqueda avanzada” y “Realizar búsqueda básica”	
Resumen:	El Caso de Uso se inicia cuando el usuario introduce los datos que se le piden para poder realizar la búsqueda de información. Esta se puede realizar de forma básica (que sería una búsqueda simple) o avanzada.	
Precondiciones:	Que el usuario ya se haya autenticado en el sistema.	
Referencias	R3,R8	
Prioridad	Primario	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
<p>1. El usuario selecciona el tipo de búsqueda(Básica o Avanzada)</p> <p>2. El usuario introduce los datos: (Nombre del proyecto, Titulo, Comentarios), y selecciona el lugar donde desea guardar el resultado de la búsqueda.</p>	<p>1.1 El sistema muestra la interfaz para introducir los datos de la búsqueda.</p> <p>2.1. El sistema toma los datos.</p>	
Flujos Alternos		
Acción del Actor	Respuesta del Sistema	
Acción 2.1	En caso de introducir mal los datos para realizar la búsqueda el sistema envía un mensaje de error.	


## Prototipo de Interfaz

Entre los datos del nuevo proyecto

Nombre del Proyecto :

Título :

Comentarios :

Carpeta de Trabajo :  

Aceptar Cancelar

Poscondiciones

Tabla 11: Descripción del caso de uso Realizar búsqueda básica

Caso de Uso:	Realizar búsqueda básica (Buscar patentes).
Actores:	Especialista (inicia)
Resumen:	El Caso de Uso se inicia cuando el Especialista decide realizar la búsqueda simple de información, para esto selecciona el criterio por el cual desea

	realizar la búsqueda, finalizando con el resultado de la misma.
Precondiciones:	Estar autenticado en el Sistema
Referencias	R4,R6
Prioridad	Primario
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
<p>1. El usuario selecciona Base de Datos de patentes.</p> <p>2. El usuario introduce el término de búsqueda.</p> <p>3. El Especialista selecciona qué va a buscar (“Palabras en el titulo” o “Palabras en el titulo o resumen”).</p>	<p>3.1 El sistema toma los datos y realiza la descarga y procesamiento de información teniendo en cuenta los datos introducidos.</p> <p>3.2 Se muestran cantidad de resultados.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
<i>Acción 1.1</i>	En caso de introducir mal los datos para realizar la búsqueda o deja alguno de los campos vacíos el sistema envía un mensaje de error.
<i>Prototipo de Interfaz</i>	



Poscondiciones	Se obtiene información referente al tema de búsqueda.
----------------	---

**Tabla 12: Descripción del caso de uso Realizar búsqueda avanzada**

Caso de Uso:	Realizar búsqueda avanzada (Buscar patentes)
Actores:	Especialista (inicia)
Resumen:	El Caso de Uso se inicia cuando el Especialista decide realizar la búsqueda de información de forma más específica, para esto el usuario selecciona el criterio por el cual desea realizar la misma, pero además puede especificar: “Fecha de publicación de la patente”, “Clasificación Internacional de Patentes (IPC)”, “solicitante” y finaliza con la obtención del resultado de la búsqueda.

Precondiciones:	Estar autenticado en el Sistema	
Referencias	R5, R6	
Prioridad	Primario	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
<ol style="list-style-type: none"> <li>1. El usuario selecciona Base de Datos de patentes.</li> <li>2. El usuario introduce el término de búsqueda.</li> <li>3. El Especialista selecciona qué va a buscar (“Palabras en el título” o “Palabra en el título o resumen”).</li> <li>4. El Especialista introduce “Fecha de publicación de la patente”.</li> <li>5. El Especialista introduce la “Clasificación Internacional de Patentes (IPC)”.</li> <li>6. El Especialista introduce el “Solicitante”.</li> </ol>	<ol style="list-style-type: none"> <li>6.1. El sistema toma los datos y realiza la descarga de información ya procesada, teniendo en cuenta los datos introducidos.</li> <li>6.2 Se muestran cantidad de resultados.</li> </ol>	
Flujos Alternos		
Acción del Actor	Respuesta del Sistema	
<i>Acción 6.1</i>	En caso de introducir mal los datos para realizar la búsqueda el sistema envía un mensaje de error.	
<i>Prototipo de Interfaz</i>		

PREDICTOR SISTEMA DE DE DESCARGA Y PROCESAMIENTO DE PATENTES

Archivo Ayuda

Configuración

Búsqueda

Básica

Avanzada

Procesamiento

Reportes

Ayuda

Seleccione las bases de datos de patentes donde se desea hacer la búsqueda

- ES-ESP@CENET (España)
- WORLD WIDE (International)
- USTPO (United State Trade Patent Office)
- PAJ (Patent Abstract of Japan)
- SIPO (State Intellectual Property Office of PRC)
- Todas

Termino de búsqueda

Seleccionar que buscar

- Palabras en el título
- Palabras en el título o resumen

Fecha de Publicación  
Formato: YYYYMMDD

Clasificación Internacional de patentes (IPC)

Solicitante

Descargar Cancelar Todo

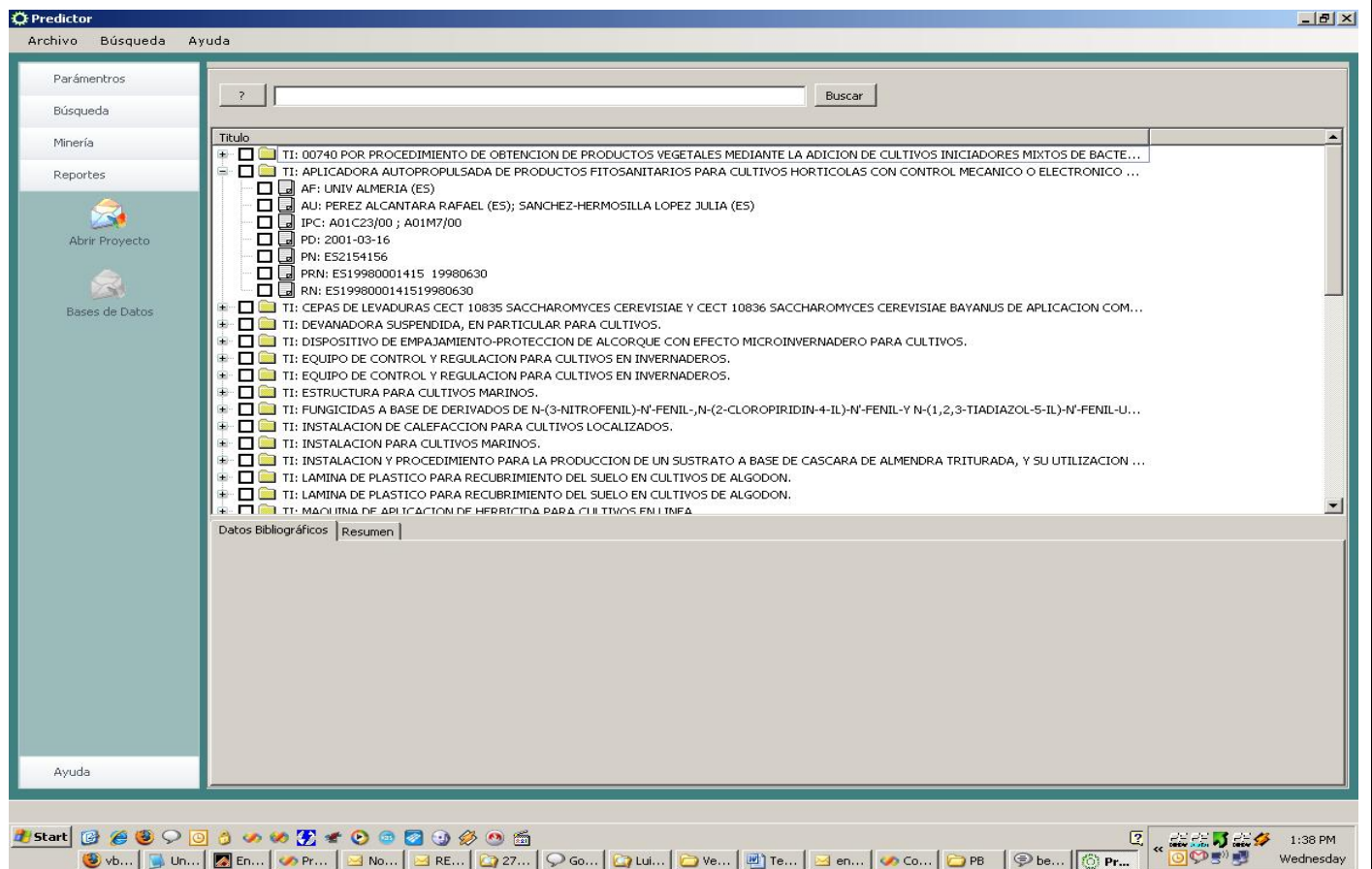
Poscondiciones

Se obtiene información referente al tema de búsqueda especificado.

**Tabla 13: Descripción del caso de uso Reportar resultados**

Caso de Uso:	Reportar resultados
Actores:	Especialista (inicia)
Resumen:	El Caso de Uso se inicia cuando el Especialista desea ver el resultado de las descargas realizadas, para esto el usuario selecciona la información de las patentes que desea ver el resultado y el mismo finaliza con la obtención de la lista con los resultados de la descarga seleccionada.
Precondiciones:	Que se haya realizado la búsqueda.
Referencias	R15
Prioridad	Secundario
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
<p>1. El Especialista selecciona la opción Abrir proyecto.</p> <p>2. El Especialista selecciona un proyecto.</p>	<p>1.1 El sistema muestra el historial de todos los proyectos mostrando el nombre, fecha de modificación y referencia.</p> <p>2.1 El sistema muestra un listado con los resultados referentes al término de búsqueda especificado.</p>
Flujos Alternos	

## Prototipo de Interfaz



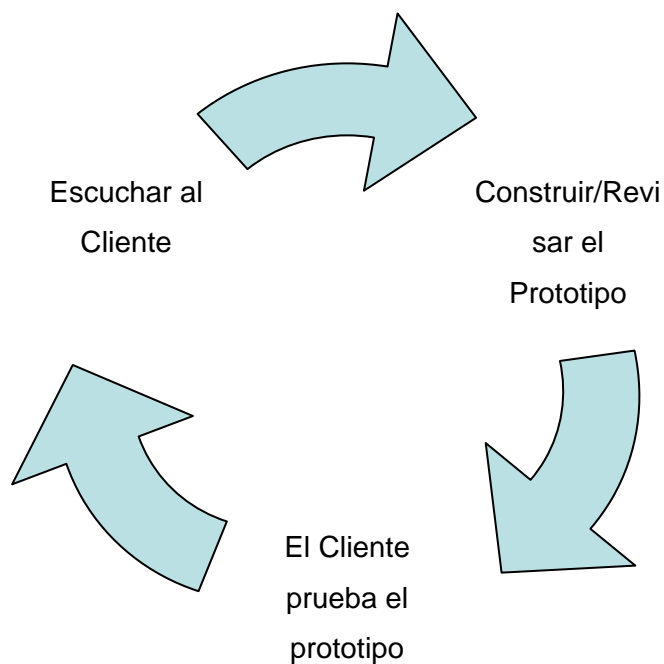
## Acción 2.1

Si el usuario no selecciona correctamente la información el sistema envía un mensaje de error. *“Debe seleccionar un proyecto”*

## 2.9. Prototipo de sistemas

En esta etapa se definieron los objetivos del software, en conjunto con el cliente, y una vez identificados los requisitos se procedió a la realización de un diseño rápido.

El diseño rápido se centra en una representación de los aspectos del software que serán visibles para el usuario/cliente (por ejemplo: enfoques de entrada y formatos de salida). El diseño rápido lleva a la construcción de un prototipo, Figura 8.



**Figura 8: Construcción de prototipos**

El prototipo lo evalúa el cliente/usuario y se utiliza para refinar los requisitos del software a desarrollar. La iteración ocurre cuando el prototipo se pone a punto para satisfacer las necesidades del cliente, permitiendo al mismo tiempo que el desarrollador comprenda mejor lo que necesita hacer (PRESSMAN 1998e).

Para la realización de este diseño rápido en el proyecto se hizo una partición a la complejidad del software estableciéndose prototipos de acuerdo a un área funcional determinada que es representada por los casos de uso definidos en el diagrama de casos de uso.

Sin importar la forma en que éste se aplique, el paradigma de construcción de prototipos ayuda al desarrollador de software y al cliente a entender mejor cuál será el resultado de la construcción cuando los requisitos estén satisfechos.

## CAPÍTULO 3 VALIDACIÓN DE LOS RESULTADOS

En las últimas décadas han surgido numerosas propuestas que definen conjuntos de métricas<sup>7</sup> de calidad para la evaluación de productos y procesos de software en distintos dominios de aplicación. Las métricas de software se refieren a un amplio rango de medidas para el software de computadoras, estas pueden ser aplicadas a organizaciones, procesos y productos los cuales directamente afectan a la estimación de costos. También existen en la actualidad algunos trabajos que proponen marcos conceptuales para la documentación de dichas métricas y catálogos (que brindan información de utilidad en los proyectos) de medición y/o evaluación de software.

De este trabajo se obtuvieron los artefactos que son la entrada al proceso de diseño que llevó a cabo uno de los desarrolladores del proyecto Delfos. Para asegurar la validez de dichos artefactos se emplearon las Métricas de la Calidad de la Especificación de requisitos, propuesta por Davis(PRESSMAN 1998f), así como el Modelo de Métricas para el Análisis y Diseño Orientado a Objetos Basado en UML (específicamente aplicado a casos de uso), propuesto por la Universidad EAFIT, Medellín, Colombia. Dichas métricas ayudarán a controlar o validar los requisitos que se obtuvieron del cliente y los casos de uso que se obtuvieron en el Modelo del Sistema.

---

<sup>7</sup> Métrica: Medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado. (EJIOGU 1993)



### **3.1. Modelo del negocio.**

La realización del Modelo de Negocio permitió que se comprendiera la estructura y dinámica de la organización; gracias a ello se pudieron identificar las Reglas del Negocio y aunque no se identificaron mejoras potenciales para el mismo sí se propusieron mejoras en los requerimientos que minimizan el esfuerzo de los especialistas a la hora de realizar sus tareas de Vigilancia Tecnológica.

Un resultado valioso de esta actividad fue que se logró que los clientes (especialistas de Delfos) y desarrolladores (equipo de desarrollo) llegaran a un entendimiento común de los procesos que tienen lugar en la Consultoría, muestra de ello es la propuesta del Diagrama de Casos de Uso del Negocio representado en la Figura 4.

En general, los artefactos que se obtuvieron en esta etapa, Diagrama de Casos de Uso del Negocio, sus realizaciones (como son las Descripciones de los Casos de Uso del Negocio y los Diagramas de Actividades) las Reglas del Negocio y otro. Además del intercambio de ideas entre los especialistas y desarrolladores, permitieron que se derivaran los requerimientos del Sistema, que han sido el hilo conductor del proceso de desarrollo de Predictor.

### **3.2. Especificación de requisitos del sistema.**

Existe una lista de características para poder valorar la calidad del modelo de análisis y la correspondiente especificación de requisitos: Especificidad (ausencia de ambigüedad), corrección, compleción, comprensión, capacidad de verificación, consistencia externa e interna, capacidad de logro, concisión, trazabilidad, capacidad de modificación, exactitud y capacidad de reutilización. Aunque muchas de las características parecen ser de naturaleza cualitativa, Davis (PRESSMAN 1998f) sugiere que todas pueden representarse usando una o más métricas.

### 3.2.1. Aplicación de las Métricas de la Calidad de la Especificación de Requisitos

En estas métricas se emplea una lista de características que pueden ser utilizadas para valorar la calidad del modelo de análisis y la correspondiente especificación de requisitos. Por las características de los requisitos capturados, sólo es aplicada la métrica que mide la ausencia de ambigüedad en los mismos y que servirá para ofrecer al cliente unos requisitos fáciles de entender y de probar.

Para medir la especificidad de los requisitos Davis (PRESSMAN 1998f) propone una métrica basada en la consistencia de la interpretación de los revisores para cada requisito:

Primeramente se tiene a  $n_r$  que representa el total de requisitos de la especificación:

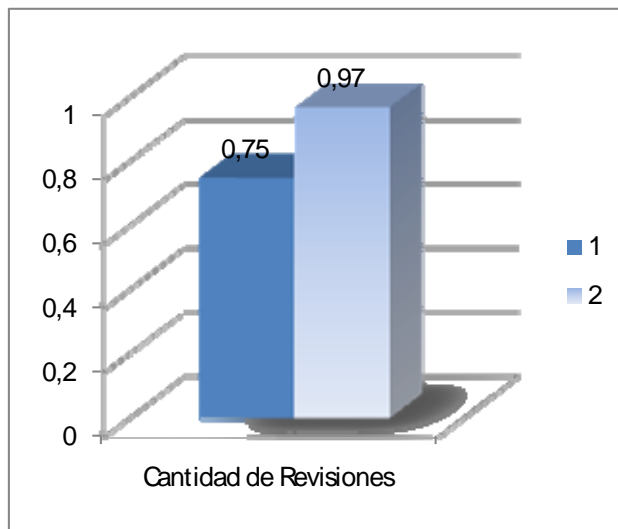
$$n_r = n_f + n_{nf}$$

Donde  $n_f$  es el número de requisitos funcionales y  $n_{nf}$  es el número de requisitos no funcionales. Luego se puede medir la especificidad de los requisitos con la fórmula:

$$Q = \frac{n_{ui}}{n_r}$$

Donde  $n_{ui}$  es el número de requisitos para los que todos los revisores tuvieron interpretaciones idénticas. Cuanto más cerca de 1 esté el valor de  $Q$ , menor será la ambigüedad de la especificación.

Para poder evaluar la métrica de la especificidad de los requisitos se realizaron dos revisiones, en las cuales se consultaron a cuatro revisores con el objetivo de obtener el menor nivel de ambigüedad y la mayor claridad posible en los mismos y así reflejar de forma más exacta las necesidades del cliente. Los resultados se muestran en la Figura 9



**Figura 9: Gráfico de Control de la Calidad de la Especificación de Requisitos**

Revisión 1: En esta revisión algunos de los requisitos funcionales y no funcionales presentaron problemas de redacción y de ambigüedad, estos son: los RNF del diseño, de software y de hardware y dos de los funcionales. Por lo que de una totalidad de 32 requisitos entre funcionales y no funciones los revisores tuvieron la misma interpretación para 24 de ellos.

$$Q = \frac{24}{31} = 0.75$$

Revisión 2: En esta última revisión apenas se detectaron problemas pues en la primera se realizó un análisis profundo de los requisitos, solo se señaló la existencia de ambigüedad en el RNF de rendimiento. Por lo que de una totalidad de 32 requisitos entre funcionales y no funcionales los revisores tuvieron la misma interpretación para 31 de ellos.

$$Q = \frac{30}{31} = 0.97$$

Después de analizar los resultados alcanzados en las revisiones realizadas, se puede concluir que los requisitos obtenidos tienen ausencia de ambigüedad prácticamente en su totalidad, ya que el valor de Q es muy cercano a 1. Esto demuestra el grado de especificidad o nivel de calidad que presentan los

mismos, elemento que hay que tener muy presente ya que si estos son incompletos o difíciles de entender entonces aumentaría el riesgo de que el producto final no sea satisfactorio.

### **3.3. Prototipo no funcional.**

En este trabajo se aplicó uno de los principios operativos del análisis donde se obtuvo un modelo de software que sería la guía en el desarrollo posterior del diseño de la aplicación. Este modelo de software es el llamado prototipo de sistemas y actúa como una forma de mostrarle al cliente cómo quedarán reflejados los requisitos en la aplicación.

#### *Ventajas de la realización del prototipo en el proyecto*

En el proyecto Delfos se realizó una retroalimentación temprana y frecuente con los clientes de la Consultoría, quienes ayudaron a modificar el prototipo de interfaz logrando que tuviera una respuesta más ágil a las necesidades actuales. Los cambios se realizaron al inicio del desarrollo del proyecto puesto que estos son menos caros que si se realizara en una etapa más avanzada del mismo. Esto permitió que se desechara un modelo de sistema que no satisfacía todas las necesidades de los usuarios y se trabajara en la construcción de uno que se ajustara mejor a sus expectativas. El resultado de este nuevo prototipo es el presentado en el Capítulo 2 en cada una de las descripciones de los casos de uso del sistema, el cual se considera que es un prototipo evolutivo<sup>8</sup>, ya que muestra de forma clara y específica los requisitos funcionales del sistema.

---

<sup>8</sup> Este es un prototipo que seguirá el diseño y la implementación del software. PRESSMAN, R. S. Ingeniería de Software: un enfoque práctico. en. Quinta Edición. Addison-Wesley, 1998g. 192-193.p.

### 3.4. Comprobación De La Calidad Del Modelo De Casos De Uso Del Sistema

Las métricas de calidad establecen una medida cuantitativa del grado en que los factores indiquen una mala calidad. Para cada métrica se determina un nombre, un significado, un umbral<sup>9</sup> y una acción sugerida<sup>10</sup>, Anexo 6.

El Modelo de Métricas para Casos de Uso presentado tiene por objetivo medir la calidad de la funcionalidad del Sistema a partir del diagrama de casos de uso generado en este proyecto de software. Dicho modelo define cuatro atributos genéricos de propiedades de calidad: *consistencia*, *correctitud*, *completitud* y *complejidad*, que tienen un significado concreto de acuerdo al tipo de artefacto software y al nivel de abstracción que éste describe. Un atributo se analiza en términos de un conjunto de factores cada uno de los cuales tendrá asociada una métrica.

Del Modelo de Métricas propuesto se tomaron solamente aquellos factores que se consideraron fundamentales a la hora de evaluar el Diagrama de Casos de Uso del Sistema, de acuerdo a las particularidades del mismo.

Como se dijo anteriormente este modelo define cuatro atributos principales. En la Tabla 14 se muestran dichos atributos con sus factores asociados:

---

<sup>9</sup> El umbral es un valor heurístico para establecer un conjunto de rangos deseables e indeseables de los valores de las métricas para la fase medida. Estos permiten identificar las anomalías en el proceso de desarrollo de una aplicación.

<sup>10</sup> Las acciones sugeridas son observaciones sobre aspectos a considerar para mejorar el comportamiento del factor medido. Básicamente los resultados de la medición son avisos para establecer un plan de acción cuando la métrica tiene un valor fuera del umbral establecido.

**Tabla 14: Atributos y factores para la medición de la calidad del Diagrama de Casos de Uso del Sistema**

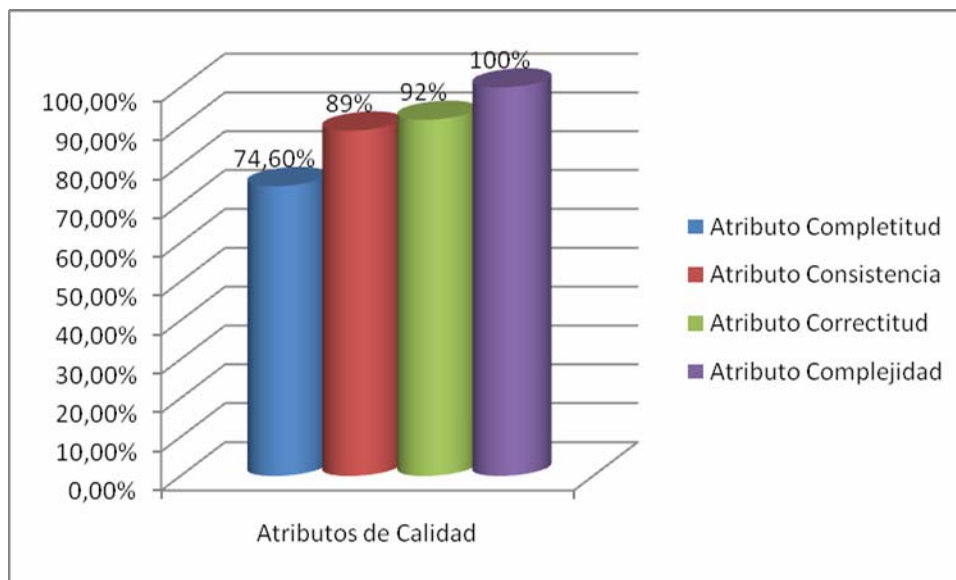
Atributo	Factores
<p>Complejidad (Grado en que se ha logrado detallar todos los casos de uso relevantes.)</p>	<p>¿Se presenta una descripción detallada (descripción extendida esencial) de todos los casos de uso del sistema?</p> <p>¿Se presenta una descripción resumida (descripción de alto nivel) de todos los casos de uso?</p> <p>¿Están definidos todos los requisitos que justifican la funcionalidad de los casos de uso?</p> <p>¿Todos los requisitos han sido considerados en algún caso de uso?</p> <p>¿Están todas las acciones del flujo de eventos redactadas en función del responsable?</p> <p>¿Se describen las condiciones de excepción que debe contemplar cada flujo de eventos?</p> <p>¿Todos los casos de uso han sido clasificados de acuerdo a su relevancia (primario / secundario / opcional)?</p>
<p>Consistencia (Grado en que los casos de uso del sistema describen las interacciones adecuadas entre el usuario y el sistema.)</p>	<p>¿El nombre dado a los casos de uso es una expresión verbal que describe alguna funcionalidad relevante en el contexto del usuario?</p> <p>¿Representan los casos de uso una interacción observable por un actor?</p> <p>¿Está adecuadamente redactado (en el lenguaje del usuario) el flujo de eventos?</p> <p>¿La descripción del flujo de eventos se inicia con la descripción de una acción externa originada por un actor o por una condición interna del sistema claramente identificable?</p> <p>¿Existe una adecuada separación entre el flujo básico de eventos y los flujos alternos?</p>
<p>Correctitud (Grado en que las interacciones actor /</p>	<p>¿Existe para cada caso de uso por lo menos un usuario responsable?</p> <p>¿Representa el caso de uso requisitos comprensibles por el usuario?</p>

sistema soportan adecuadamente el proceso del sistema.)	<p>¿Las interacciones definidas describen la funcionalidad requerida del sistema?</p> <p>¿Las interacciones definidas introducen mejoras al proceso actual?</p>
Complejidad (Grado de claridad en la presentación de los elementos que describen el contexto y la claridad del sistema.)	<p>¿Los elementos dentro del diagrama están adecuadamente ubicados de manera que facilitan su interpretación?</p>

### 3.4.1. Definición de factores para el Modelo de casos de uso del Sistema

Un modelo de casos de uso describe los requisitos funcionales de un actor (usuario, sistema, dispositivo, etc.) en términos de las interacciones que éste realiza con el sistema. Dichas interacciones se describen por medio de uno o más flujos de eventos (básicos, alternativos, excepcionales) que se suceden para llevar a cabo una tarea y que reciben el nombre de escenarios. Aunque los casos de uso no determinan la arquitectura básica del sistema, éstos se convierten en el hilo conductor que orienta el proceso de desarrollo. La consideración de calidad de un modelo orientado a objetos desde la óptica de los casos de uso y sus escenarios asociados, permitirá establecer criterios de calidad relevantes con respecto al grado de adecuación del modelo a las necesidades o deseos del usuario, aspecto difícil de considerar desde la óptica estructural.

A continuación se muestran las gráficas de los factores que se consideraron de mayor peso a la hora de definir la calidad del proceso de análisis al *Sistema de descarga y procesamiento automatizado de patentes*. Se realizaron dos revisiones al Modelo de Casos de Uso del Sistema. De la primera de ellas se obtuvieron los resultados mostrados en la Figura 10.



**Figura 10: Gráfico de Control de la Calidad del Modelo de Casos de Uso para la primera revisión**

#### *Revisión 1: Observaciones*

##### Para el atributo Completitud

No se presentaba una descripción detallada (descripción extendida esencial) de todos los casos de uso del sistema sino sólo de cinco de estos representando el 55.60% del total de casos de uso. El valor resultante (representado por los casos de uso a los que no se le realizó una descripción detallada (44.40%)) se encuentra por encima del umbral definido por la métrica, por tal motivo se sugiere: Interactuar con el cliente para realizar la definición extendida del caso de uso que sea consistente con la definición a alto nivel.

No se presentó una descripción resumida (descripción de alto nivel) de los casos de uso del sistema, pero no se determinó una acción a sugerir pues no se considera importante para este trabajo contar con dicha descripción ya que se tienen las descripciones detalladas de los casos de uso, las cuales brindan una información más profunda de los procesos y por tanto más conveniente para el mejor entendimiento.



No se encontraban representados todos los requisitos que justifican la funcionalidad de los casos de uso. Seis de ellos se encontraban bien definidos, por lo que dicho factor representa en esta revisión un valor igual al 33.33 % del total de requisitos que deben justificar la funcionalidad de los casos de uso, por tal motivo se sugiere: Revisar la lista de requisitos para determinar cuáles serán apoyados por cada caso de uso.

Todos los requisitos han sido considerados en algún caso de uso. Las acciones del flujo de eventos de todos los casos de uso se encuentran redactadas en función del responsable y estos últimos han sido clasificados de acuerdo a su relevancia (Primario, Secundario, Opcional). Todos estos resultados positivos sumados y divididos por la cantidad total de factores que incluye el atributo Completitud dan un valor de 74.60% para la calidad del mismo.

Para el atributo Consistencia

Solo cuatro casos de uso estaban representados por una expresión verbal, representando un valor para este factor igual al 55.50% del total de casos de uso. El flujo de eventos está adecuadamente redactado en el lenguaje del usuario y se inicia con la descripción de una acción originada por un actor. Además de ello existe una adecuada separación entre el flujo básico de eventos y los flujos alternos, siendo representados en una tabla de descripción detallada definida en una de las plantillas que ofrece RUP para la documentación adecuada del proyecto (Anexo 4). Todos estos resultados positivos sumados y divididos por la cantidad total de factores que incluye este atributo dan un valor de 88.88% para la calidad del mismo.

Para el atributo Correctitud

Todos los casos de uso poseen al menos un usuario responsable, no existe ninguno que no sea iniciado por un actor y representan requisitos comprensibles por el usuario. Las interacciones que se describen en el Modelo de Casos de Uso detallan la funcionalidad requerida del Sistema para seis de los casos de uso, representando un valor para este factor del 44.4% del total de casos de uso.

Todos los procesos del Sistema introducen mejoras al proceso actual, específicamente el caso de uso *Procesar patentes* permitirá al Especialista realizar el procesamiento a patentes que no hayan sido

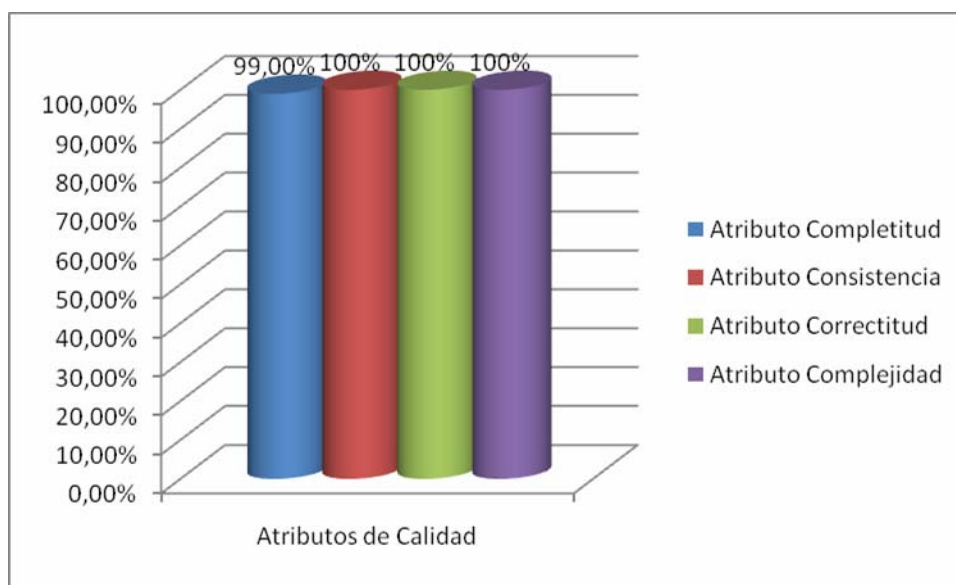
descargadas de Internet mediante Predictor. Todos los resultados positivos promediados dan un valor de 91.67% para la calidad del mismo.

Para el atributo Complejidad

Los elementos dentro del diagrama están adecuadamente ubicados de manera que facilitan su interpretación, representando un 100% de la calidad del atributo.

### *Revisión 2: Observaciones*

En la Figura 11 se muestran los resultados de la segunda revisión realizada al Modelo de Casos de Uso del Sistema.



**Figura 11: Gráfico de Control de la Calidad del Modelo de Casos de Uso para la segunda revisión**

No se realizó la descripción detallada (descripción extendida esencial) de todos los casos de uso del sistema. A todos los demás factores se le dio cumplimiento.

### *Resumen*

Los resultados de la evaluación del Diagrama de Casos de Uso demuestran que se ha obtenido un modelo con un alto grado de funcionalidad. Se ha probado que el modelo cumple con las características o atributos principales que definen la calidad, la cual esta dada por el grado con el cliente percibe que el software satisface sus expectativas. Se demostró que el Diagrama de Casos de Uso es un diagrama completo pues contiene el área funcional relevante que apoya el sistema. Se presentó una descripción detallada de todos los casos de uso donde se definió el flujo de eventos en función del responsable y donde se describieron las condiciones de excepción que contemplan estos flujos.

Los casos de uso se nombraron con una expresión verbal que describe una funcionalidad relevante, concediéndole una mayor comprensión al diagrama de casos de uso. Asimismo las interacciones definidas describen la funcionalidad requerida del Sistema, donde se introdujeron mejoras al proceso que minimizan el esfuerzo de los especialistas a la hora de realizar sus tareas de Vigilancia Tecnológica.

Todos los casos de uso representan acciones en el flujo de eventos que son asignadas a un responsable, en este caso ese responsable sería el Especialista o el Administrador del sistema. Por otro lado, se pudo comprobar que todos los requisitos funcionales se encuentran reflejados en el diagrama, (justificando la funcionalidad de estos últimos) y que son comprensibles y aceptables por el usuario.

Además se puede mencionar, que el Diagrama de Casos de Uso está acorde con lo especificado por la metodología RUP. Se encuentran bien modeladas las relaciones de generalización/especialización entre actores, así como entre casos de uso. Se usaron las relaciones de inclusión correctamente, teniendo en cuenta que el comportamiento definido para un caso de uso incluido es un proceso que se inserta explícitamente dentro del comportamiento definido para el caso de uso base. Y por último se definieron los elementos dentro del diagrama ubicándolos de manera que facilitan su interpretación tanto para los desarrolladores como para el cliente/usuario.

## CONCLUSIONES

En el presente trabajo se lograron cumplir los objetivos propuestos ya que:

1. Se desarrolló el proceso de análisis al Sistema de Descarga y Procesamiento Automatizado de Patentes del que se obtuvieron los artefactos que resolvieron la problemática existente.
  - Gracias al estudio de los procesos del negocio de la Consultoría Delfos se logró la comprensión de la estructura y dinámica de la misma, además de que se obtuvo una visión detallada de dichos procesos (mediante su modelación y descripción) siguiendo los pasos de la Metodología RUP.
  - Se aseguró que usuarios finales y desarrolladores tuvieran un entendimiento común de la organización permitiendo que se obtuvieran los requisitos del Sistema que verdaderamente responden a las necesidades y expectativas de los clientes. Para ello se aplicaron varias técnicas de captura de requisitos, destacándose como las más efectivas: la Entrevista realizada a los clientes y la técnica de Tormenta de ideas.
2. Se hizo una evaluación de la calidad del proceso de análisis realizado al Sistema mediante un conjunto de Métricas para la Calidad de la Especificación de Requisitos y un Modelo de Métricas para el Análisis y Diseño Orientados a Objetos de la que se obtuvo un resultado satisfactorio.

## RECOMENDACIONES

Para lograr una mejor calidad en el desarrollo del *Sistema de descarga y procesamiento automatizado de patentes* se recomienda para próximas iteraciones:

- Construir una base de datos local donde estarán guardadas las patentes descargadas de búsquedas anteriores, de esta forma se evitará el almacenamiento de información repetida.
- Agregar un plugin a Predictor que adicione a la herramienta la funcionalidad o la capacidad de analizar la información de patentes mediante matrices de co-ocurrencia, mapas tecnológicos y la creación de tesauros para la reducción de datos. Además de realizar análisis estadísticos multidimensionales para identificar grupos y relaciones entre conceptos, autores y países.
- Realizar la búsqueda y descarga de información relacionada no solo con patentes sino con las publicaciones científico - técnicas, entre otras, para determinar el estado actual y las principales tendencias en las investigaciones científicas, tecnológicas y/o comerciales en el sector de las TICs.

## REFERENCIAS BIBLIOGRÁFICAS

CARRILLO, M. G. *JavaMod: una API para la gestión de programas Java*. INGENIERÍA INFORMÁTICA, UNIVERSIDAD REY JUAN CARLOS, 2004. [2007] Disponible en: <http://vido.escet.urjc.es/javacet/JavaCET%200.9-beta1%20Presentation.pdf>

CATALDI, Z. *Metodología de diseño, desarrollo y evaluación de software educativo*. Facultad de Informática. UNLP, 2000. [2007] Disponible en: <http://www.fi.uba.ar/laboratorios/lsi/cataldi-tesisdemagistereninformatica.pdf>

DORIA, H. G. *Las Métricas de Software y su Uso en la Región*. Departamento de Ingeniería en Sistemas Computacionales. Cholula, Puebla, México, Universidad de las Américas, Puebla, 2001. [2007] Disponible en: [http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lis/gonzalez\\_d\\_h/](http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/gonzalez_d_h/)

ESPINOZA, J. A. M. *Metodología OMT*, 1997. [2007]. Disponible en: <http://www.monografias.com/trabajos6/meto/meto.shtml>

HERNÁNDEZ, R. G. *La información de marcas como indicador de innovación tecnológica.*, Grupo de Vigilancia Tecnológica de la Consultoría del Ministerio de la Informática y las Comunicaciones. 2006.

IDRIS, K. *La propiedad intelectual al servicio del crecimiento económico*. Organización Mundial de la Propiedad Intelectual (OMPI), 2003. [2007]: Disponible en: [http://www.wipo.int/freepublications/es/intproperty/888/wipo\\_pub\\_888\\_1.pdf](http://www.wipo.int/freepublications/es/intproperty/888/wipo_pub_888_1.pdf)

KRUCHTEN, P. W. R. *"A Rational Development Process"*. 1996.

PRESSMAN, R. S. *Ingeniería de Software, un enfoque práctico*. Quinta Edición. Addison Wesley Professional 1998a.

---. *Ingeniería de Software: un enfoque práctico*. Quinta Edición. Addison Wesley Professional 1998b. pp. 182

---. *Ingeniería de Software: un enfoque práctico*. Quinta Edición. Addison Wesley Professional, 1998c. pp. 171-175

---. *Ingeniería de Software: un enfoque práctico*. Quinta Edición. Addison Wesley Professional 1998d. pp. 171

---. *Ingeniería de Software: un enfoque práctico*. Quinta Edición. Addison Wesley Professional 1998e. pp. 21-22

---. *Ingeniería de Software: un enfoque práctico*. en. Quinta Edición. Addison Wesley Professional, 1998f. pp. 331-332

---. Ingeniería de Software: un enfoque práctico. en. Quinta Edición. Addison Wesley Professional, 1998g. pp. 192-193.

ROJAS, V. M. C. G. Y. J. C. O. *Metodología OMT*, 1997. [2007] [Disponible en: <http://www.monografias.com/trabajos13/metomt/metomt.shtml>]

SÁNCHEZ, M. V. G. *Patentometría. Herramienta para el análisis de oportunidades tecnológicas.*, Universidad de La Habana, 1999. [2007] Disponible en: <http://www.um.es/cugio/tesis/patentometria.pdf>

SAWYER, S. I. Y. P. *Requirements Engineering*. Editorial John Wiley & Sons, 1997.p. ISBN-13: 978-0-471-97444-4 -

SENN, J. A. *Análisis y Diseño de Sistemas de Información*, Segunda Edición. Editorial McGraw Hill, 1992. ISBN-10: 9684229917. ISBN-13: 9789684229914

THAYER, R. H., M. DORFMAN. *System and Software Requirements Engineering*. IEEE Computer Society Press, 1997. ISBN-13: 9780818689215.

## BIBLIOGRAFÍA

**A. Davis, S. Overmyer, K. Jordan, J. Caruso, F. Dandashi, A.Dinh, G. Kincaid, G. Ledeboer, P. Reynolds, P. Sitaran, A. Ta, y M. Theofanos.** (1)

A. DAVIS, S. O., K. JORDAN, J. CARUSO, F. DANDASHI, A.DINH, G. KINCAID, G. LEDEBOER, P. REYNOLS, P. SITARAN, A. TA, Y M. THEOFANOS. *Identifying and measuring quality in software requirements specification: En Proceedings 1st Int'l Software Metrics Symposium*, 1993.

**Alessandro Comai, Joaquín Tena, Juan Carlos Vergara** (1)

ALESSANDRO COMAI, J. T., JUAN CARLOS VERGARA. *Software para la vigilancia tecnológica de patentes: evaluación desde la perspectiva de los usuarios. El profesional de la información*, 2006. Vol.15, No. 6, pp. 452–458.

**Anegón-Solana, Félix de Moya, Víctor Herrero** (1)

ANEGÓN-SOLANA, F. D. M., VÍCTOR HERRERO. *Visibilidad internacional de la producción científica iberoamericana en biblioteconomía y documentación*, 1991-2000. [2007]. Disponible en: [http://www.madrimasd.org/Queesmadrimasd/indicadores/documentos/Visibilidad\\_Internacional.pdf](http://www.madrimasd.org/Queesmadrimasd/indicadores/documentos/Visibilidad_Internacional.pdf) -

**Carrillo, Micael Gallego** (1)

CARRILLO, M. G. *javaMod: una API para la gestión de programas Java*. INGENIERÍA INFORMÁTICA, UNIVERSIDAD REY JUAN CARLOS, 2004. [2007] Disponible en: <http://vido.escet.urjc.es/javacet/JavaCET%200.9-beta1%20Presentation.pdf>

**Casallas, Rubby** (1)

CASALLAS, R. *Una visión basada en los permanentes cambios tecnológicos y otros aspectos relacionados*, Nº 93 Julio - Septiembre de 2005. , 2005. [2006]. Disponible en: <http://www.acis.org.co/index.php?id=547>

**Cataldi, Zulma** (1)

CATALDI, Z. *Metodología de diseño, desarrollo y evaluación de software educativo*. Facultad de Informática. UNLP, 2000. Disponible en: <http://www.fi.uba.ar/laboratorios/lsi/cataldi-tesisdemagistereninformatica.pdf>



**Cárdenas, Gema C.** (1)

CÁRDENAS, G. C. *Herramientas cuantitativas. Experiencia de su aplicación en la Consultoría Biomundi.*, Consultoría Biomundi/IDICT. Dirección de Inteligencia Corporativa del Instituto de Documentación e Información Científica y Tecnológica, del Ministerio de Ciencia, Tecnología y Medio Ambiente., 2006. [2007]. Disponible en: [http://www.intempres.pco.cu/Intempres2006/Intempres2006/Propuestos%20a%20posters/GEMA\\_INTEMPRES.pdf](http://www.intempres.pco.cu/Intempres2006/Intempres2006/Propuestos%20a%20posters/GEMA_INTEMPRES.pdf)

**Dickinson, Q. Todd** (1)

DICKINSON, Q. T. *Reconciling Research and the Patent System*, Issues in Science and Technology. Summer 2000 Vol. 16. No. 4. pp 64-70. ISSN: 07485492

**Doria, Heidi G.** (1)

DORIA, H. G. *Las Métricas de Software y su Uso en la Región*. Departamento de Ingeniería en Sistemas Computacionales. Cholula, Puebla, México, Universidad de las Américas, Puebla, 2001. [2007] Disponible en: [http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lis/gonzalez\\_d\\_h/](http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/gonzalez_d_h/)

**Dürsteler, Juan C.** (1)

DÜRSTELER, J. C. *Análisis de Patentes* La revista digital de InfoVis.net, 2005-06-05. [2007]. Disponible en: <http://www.infovis.net/printMag.php?num=167&lang=1>

**Ejiogu, Len** (1)

EJIOGU, L. *Software Engineering with Formal Metrics*. John Wiley & Sons. 1993. ISBN-10: 047156155X ISBN-13: 978-0471561552

**Espinoza, Javier A. Moya** (1)

ESPIÑOZA, J. A. M. *Metodología OMT*, 1997. [2007]. Disponible en: <http://www.monografias.com/trabajos6/meto/meto.shtml>

**Guardado, Ma. de Jesús Zamarripa** (1)

GUARDADO, M. D. J. Z. *Metodología de la Investigación Social*, Dpto. de Investigación CNEP 2003. [2007]. Disponible en: <http://www.cnep.org.mx/Informacion/teorica/metodologia.htm>

**Hernández, Rolando González** (1)

HERNÁNDEZ, R. G. *La información de marcas como indicador de innovación tecnológica*. Consultoría del Ministerio de la Informática y las Comunicaciones, 2006.

**Hong, Soonwoo** (1)

HONG, S. *La magia de la información sobre patentes*, 2005 [2007]. Disponible en: [http://www.wipo.int/sme/es/documents/patent\\_information.htm#info](http://www.wipo.int/sme/es/documents/patent_information.htm#info)

**Idris, Kamil** (1)

IDRIS, K. *La propiedad intelectual servicio del crecimiento económico*. Organización Mundial de la Propiedad Intelectual (OMPI), 2003. [2007]: Disponible en: [http://www.wipo.int/freepublications/es/intproperty/888/wipo\\_pub\\_888\\_1.pdf](http://www.wipo.int/freepublications/es/intproperty/888/wipo_pub_888_1.pdf)

**Ivar Jacobson, Grady Booch & James Rumbaugh** (1)

IVAR JACOBSON, G. B. J. R. *El Proceso Unificado de Desarrollo de Software*. 1999. 464 p. ISBN: 84-7829-036-2

**J. A. Carsí, I. Ramos, J. Silva, J. Pérez, V. Anaya** (1)

J. A. CARSI, I. R., J. SILVA, J. PÉREZ, V. ANAYA. *Un Generador Automático de Planes de Migración de Datos*, Departamento de Sistemas Informáticos y Computación Universidad Politécnica de Valencia, 2002. [ 2007]. Disponible en: <http://issi.dsic.upv.es/archives/f-1050406290873/No1Art02.pdf>

**Jiménez, Beatriz Bernárdez** (1)

JIMÉNEZ, B. B. *Una Aproximación Empírica a la Verificación de Especificaciones de Requisitos para Sistemas de Información*. Departamento de Lenguajes y Sistemas Informáticos. Sevilla, Universidad de Sevilla, 2003. Disponible en: [http://www.lsi.us.es/docs/doctorado/memorias/Memoria\\_ProyectoInvestigador.pdf](http://www.lsi.us.es/docs/doctorado/memorias/Memoria_ProyectoInvestigador.pdf)

**Juan Sánchez, Jorge Belenguer, Pablo Belenguer, David Pascual** (1)

JUAN SÁNCHEZ, J. B., PABLO BELENGUER, DAVID PASCUAL. *VRU: Un método para validar requisitos y generar interfaces de usuario multiplataforma.*, Departamento de Sistemas Informáticos y Computación Universidad Politécnica de Valencia, 2002. [ 2007]. Disponible en: [www.lbd.dcc.ufmg.br/bdbcomp/servlet/Trabalho?id=4478-4k](http://www.lbd.dcc.ufmg.br/bdbcomp/servlet/Trabalho?id=4478-4k)

**Kruchten, Philippe, W. Royce** (1)

KRUCHTEN, P. "A Rational Development Process". 1996. p. ISBN:

**Larman, Craig** (1)

LARMAN, C. *UML y Patrones, Introducción al análisis y diseño orientado a objetos.*, 1999. p. ISBN: 970-17-0261-1

**Leyva, Juan** (1)

LEYVA, J. *Fundamentos de Extreme Programming* 2005. [ 2007]. Disponible en: [http://www.programacion.com/blogs/84\\_metricas\\_web/archive/526\\_fundamentos\\_de\\_extreme\\_programming\\_parte\\_ii.html](http://www.programacion.com/blogs/84_metricas_web/archive/526_fundamentos_de_extreme_programming_parte_ii.html)

**Luis E. Mendoza, María A. Pérez, Gabriela Díaz-Antón, Anna Grimán, Patricia Ottaviano, Nilzaris Cova** (1)

LUIS E. MENDOZA, M. A. P., GABRIELA DÍAZ-ANTÓN, ANNA GRIMÁN, PATRICIA OTTAVIANO, NILZARIS COVA. *Mejoras en RUP para la implementación de aulas virtuales: caso de estudio IESA*, [2007]. Disponible en: [http://www.lisi.usb.ve/publicaciones/07%20integracion%20de%20sistemas/integracion\\_18.pdf](http://www.lisi.usb.ve/publicaciones/07%20integracion%20de%20sistemas/integracion_18.pdf)

**Marcela Daniele, Paola Martellotto, Gabriel Baum** (1)

MARCELA DANIELE, P. M., GABRIEL BAUM. *Traducción del Modelo Genérico del Modelo de Negocio a Object-Z*, 2004-2006. [2007]. Disponible en: <https://dc.exa.unrc.edu.ar/wicc/papers/IngenieriaSoftware/99.pdf>

**Marcela Ridao, Jorge Doorn, Julio César Sampaio do Prado Leite** (1)

MARCELA RIDAO, J. D., JULIO CÉSAR SAMPAIO DO PRADO LEITE. *Uso de Patrones en la Construcción de Escenarios*, 2000. [2007]. Disponible en: [http://wer.inf.puc-rio.br/WERpapers/artigos/artigos\\_WER00/ridao.pdf](http://wer.inf.puc-rio.br/WERpapers/artigos/artigos_WER00/ridao.pdf)

**M.J. Escalona, N. Koch** (1)

M.J. ESCALONA, N. K. *Ingeniería de Requisitos en Aplicaciones para la Web: Un estudio comparativo*, 2003. [2007]. Disponible en: URL: <http://www.pst.informatik.uni-muenchen.de/personen/kochn/ideas03-escalona-koch.pdf>

**Molpeceres, Alberto** (1)

MOLPECERES, A. *Procesos de desarrollo: RUP, XP y FDD*, 2002. [2007]. Disponible en: <http://www.javaHispano.org>

**O'Reilly, Tim** (1)

O'REILLY, T. The Internet Patent Land Grab. (Company Business and Marketing) (*Company Business and Marketing*), 2000, 43(6): 29. Disponible en: <http://portal.acm.org/citation.cfm?id=336460.336471&coll=GUIDE&dl=GUIDE&CFID=530635&CFTOKEN=44177177>

**Övergaard Gunnar, Palmkvist Karin** (1)

ÖVERGAARD G., P.K. *Use Cases Patterns and Blueprints*, Addison Wesley Professional. 2004. p. ISBN: 0-13-145134-0

**Pressman, Roger S.** (1)

PRESSMAN, R. S. *Ingeniería de Software, un enfoque práctico*, Addison Wesley Professional. 1998. p. ISBN: 0-201-17888-5

**Rojas, Víctor M. Chávez Gaona y Juan C. Olivares** (1)

ROJAS, V. M. C. G. Y. J. C. O. *Metodología OMT*, 1997. [2007] Disponible en: <http://www.monografias.com/trabajos13/metomt/metomt.shtml>

**Sánchez, María Victoria Guzmán** (1)

SÁNCHEZ, M. V. G. *Patentometría. Herramienta para el análisis de oportunidades tecnológicas.*, Universidad de La Habana, 1999. [2007] Disponible en: <http://www.um.es/cugio/tesis/patentometria.pdf>

**Sawyer Pete , Sommerville Ian** (1)

SAWYER, S. I. Y. P. *Requirements Engineering*. JOHN WILEY & SONS, 1997.p. ISBN-13: 978-0-471-97444-4 -

**SENN, James A.** (1)

SENN, J. A. *Análisis y Diseño de Sistemas de Información*, Segunda Edición. Editorial McGrawHill, 1992.

**Schmuller, Joseph** (1)

SCHMULLER, J. *Aprendiendo UML en 24 horas*. México, Pearson Education, 2000. 425 p. ISBN: 968-444-463-X

**Sosa Jost, Eliana B. - Valesani, María E. - Mariño, Sonia I.** (1)

SOSA JOST, E. B.-V., MARIA E. - MARIÑO, SONIA I. *Software educativo para apoyo a la enseñanza de la Técnica de Modelado y Diseño Orientado a Objetos*, Departamento de Informática. Facultad de Ciencias Exactas y Naturales y Agrimensura., 2004. [2007]. Disponible en: <http://www.unne.edu.ar/Web/cyt/com2004/8-Exactas/E-031.pdf>

**Straub., Pablo** (1)

STRAUB., P. *Desarrollo de Software Orientado a los Negocios con Métodos Ágiles*, AgileShift Systems Engineering, 2004-2005. p.

**Swartz, Nikki** (1)

SWARTZ, N. U.S. patent office moves data online. (Up front: news, trends & analysis). (Brief Article) (*Brief Article*), 2003, 37(3): 8(1).

**Thayer, Richard H. & Dorfman, Merlin** (1)

THAYER, R. H., M. DORFMAN. *System and Software Requirements Engineering*. IEEE Computer Society Press, 1997. ISBN-13: 9780818689215

**Torregrosa, Marta** (1)

TORREGROSA, M. *Metodología de la investigación en filosofía*, 2004. [2006]. Disponible en: <http://www.unav.es/gep/Metodologia/EstrategiasInvestigacion.html>

**Whitten, J. y Bentley, L.** (1)

WHITTEN, J. Y. B., L. *Systems Analysis and Design Methods*. Fourth Edition. Irwin McGraw Hill, 1998.

**Wieggers, Karl E.** (1)

WIEGERS, K. E. *Automating Requirements Management. Process Impact*, 1999. [2007] Disponible en: [http://www.processimpact.com/articles/rm\\_tools.pdf](http://www.processimpact.com/articles/rm_tools.pdf)