



Universidad de las Ciencias  
Informáticas

**PROPUESTA DE PROCEDIMIENTO  
PARA LA CAPTURA DE REQUISITOS  
EN LOS PROYECTOS DE PORTALES DE LA UCI**

Trabajo de diploma para optar por el título de:

**Master en Gestión de  
Proyectos Informáticos**

**Autor: Ing. Maidely Calderón Montero**

**Tutor: DrC. Rolando Alfredo Hernández León**

---

**Ciudad de la Habana, Julio 2007**

---

**Dedicatoria**

*A mis padres, siento la distancia de estos últimos 4 años,  
pero siempre los tendré presente.*

*A mi hermano, con los años adquirimos experiencias  
que no podemos desaprovechar.*

### Agradecimientos

Primero, quisiera que no existieran celos por el orden ni por aquellos que pudieran faltar. Siempre los tendré presente a todos.

A mi tutor, gracias por la amistad que existe entre nosotros, ha sido muy importante.

A mis padres y toda mi familia, gracias por el apoyo durante estos años y por confiar en mi capacidad, mucha o poca pero es mía.

A la memoria de mi tía Rubia, siempre te tengo presente.

A mi hermana Nela, muchos años juntas, algunas incomprensiones pero te quiero mucho.

A Frank, Fb, por tu amor y dedicación, fuiste gran parte de mi inspiración y mi confianza.

A mi Pina, por tenerme siempre presente y soportar mis noches de lágrimas interminables.

A Piolin, por los años que compartimos juntos, fueron muy importantes en mi vida.

A mis niños lindos, Keiver y Yerandy, se que siempre podré contar con Uds.

A Gladys, Enier y mis Alumnos Ayudantes por el apoyo incondicional.

A mi chinata (el chino), el muñequito del canal educativo (Arian) y mi flaco lindo (Amaury), siempre están en mi corazón.

A mis compañeras de la maestría, Yadira, Yeleny, Dainys y Yenín por las noches de preocupación.

A el Guille, Rolan, Orlenis, Arnaldo, Pernía1 y Pernía2, Eder, Roberto, William, Abel, Graciela, Made, el Chony, el Gabo y mis niñas de 4to año, su amistad y su compañía ha sido importante para mi.

A todos aquellos que pusieron su granito para que mis sueños se hicieran realidad.

### Resumen

La captura de los requisitos es uno de los flujos más importantes en el desarrollo de un software, ya que es donde se definen las funcionalidades del mismo. Este trabajo ofrece un procedimiento detallado, que sirve de guía para la ejecución del proceso de captura de requisitos en los sistemas de portales Web. Para lograr la definición del procedimiento se analizaron las vías que existen actualmente para llevar a cabo el proceso a diferentes escalas y posteriormente se determinó que no existía una que se ajustara al tipo de sistema que se analiza, los portales Web. Se procedió entonces a realizar una caracterización del levantamiento de requisitos de los proyectos de portales de la *UCI*, determinando los principales problemas que causaron la necesidad de definir un nuevo procedimiento. Se llevó a cabo la reconceptualización del mismo, contribuyendo a una mejor definición de los requisitos del sistema, garantizando la calidad del proceso y una mejor satisfacción del cliente.

### **Abstract**

The requirement management is one of the most important workflow in the software development process, because in this workflow is where we define the software features. This work offers a detailed procedure, very useful for the software requirement management on Web portal system. In order to get the definition of the procedure, a research was made, to determinate the existing ways to do this process in different scales, and we get as a conclusion, the non existence of any to apply on Web portal systems. A characterization of the requirement management in Web portal systems on projects here on *UCI*, was made to determinate the main problems that cause the necessity of define a new procedure to do this task. The procedure was detailed helping the requirement management process, guaranteeing the quality of the process and the client satisfaction.

---

Dedicatoria .....	I
Agradecimientos.....	II
Resumen.....	III
Abstract .....	IV
Introducción.....	1
Capítulo 1: Fundamentos Teóricos .....	6
Introducción .....	6
1.1 La Ingeniería de Requerimiento ( <i>IR</i> ).....	6
1.2 La <i>IR</i> y los Modelos Clásicos de Desarrollo de Software.....	7
1.3 La <i>IR</i> , Actividades, Técnicas y Metodologías .....	9
1.4 La <i>IR</i> y las Metodologías de Desarrollo .....	19
1.4.1 WSDM.....	20
1.4.2 RNA .....	21
1.4.3 HFPM .....	22
1.4.4 UWE .....	23
1.4.5 W2000.....	23
1.4.6 UWA .....	25
1.4.7 NDT.....	26
1.4.8 Design-driven Requirements Elicitation .....	29
1.4.9 RUP.....	29
1.5 Los Sistemas Tradicionales y los Sistemas Web.....	32
Conclusiones.....	35
Capítulo 2: Caracterización del Entorno y Métodos Utilizados. ....	36
Introducción .....	36
2.1 Conceptos de Interés.....	36
2.2 Elementos a Considerar en el Levantamiento de Requisitos .....	38
2.3 Caracterización de la Captura de Requisitos en los Portales Web de la <i>UCI</i> .....	40
2.4 Población, Muestra, Tipo de Investigación y Métodos Utilizados .....	43
2.4.1 Población .....	43
2.4.2 Muestra .....	44
2.4.3 Tipo de Investigación.....	44

---

2.4.4 Métodos Utilizados .....	44
2.5 Análisis de Resultados .....	48
Conclusiones.....	52
Capítulo 3: Descripción del Procedimiento .....	53
Introducción .....	53
3.1 Premisas a Cumplir .....	53
3.2 Descripción de la Propuesta .....	54
3.2.1 Actividad 1: Captura de Requisitos .....	55
3.2.2 Actividad 2: Especificación de Requisitos .....	59
3.2.3 Actividad 3: Validación de Requisitos .....	62
3.3 Propuesta de Herramientas y Metodologías.....	65
3.3.1 Herramientas .....	65
3.3.2 Metodologías.....	68
3.4 Validación de la Propuesta .....	70
Conclusiones.....	72
Conclusiones.....	73
Recomendaciones.....	74
Bibliografía .....	75
Glosario de Términos .....	79
Anexos .....	83
Anexo 1 – Cronograma de Captura de Requisitos .....	83
Anexo 2 – Proyecto Técnico .....	84
Anexo 3 – Plan de Requisitos.....	85
Anexo 4 – Glosario de Términos.....	86
Anexo 5 – Especificación de Requisitos.....	87
Anexo 6 – Seguimiento de Requisitos.....	91
Anexo 7 – Solicitud de Cambios.....	92

Figura 1.1: Capas de la Ingeniería de Software .....	8
Figura 1.2: Modelo en espiral de la Ingeniería de Requisitos .....	9
Figura 1.3: Proceso de desarrollo en <i>W2000</i> .....	25
Figura 1.4: Ingeniería de Requerimiento en NDT .....	28
Figura 2.1: Encuesta Aplicada.....	47
Figura 2.2: Guión de la Entrevista .....	47
Figura 2.3: Actividades Realizadas .....	48
Figura 2.4: Técnicas Utilizadas.....	49
Figura 2.5: Aspectos a Considerar .....	50
Figura 2.6: Roles Implicados .....	51
Figura 3.1: Algoritmo Prototipo Evolutivo .....	64



Tabla 3.1: Análisis de las tareas.....	71
--	----

## Introducción

A nivel mundial y nacional se lleva a cabo la realización de sistemas de software, con el objetivo de automatizar los procesos de una determinada empresa o entidad garantizando una mayor eficiencia en la misma, prestando a su vez, servicios con la mejor calidad posible. Existen pasos fundamentales que se deben seguir para garantizar que el producto realizado cuente con dicha calidad y pueda ser regulado por las normas que rigen este proceso.

Los especialistas dedicados a la realización de sistemas de software guían el proceso a través de metodologías de desarrollo garantizando la realización de sistemas mediante una serie de pasos fundamentales que facilitan la evolución del proceso de desarrollo. Estas metodologías están compuestas por etapas o fases y dentro de las mismas, se describen una serie de flujos que se deben seguir para garantizar la obtención de un producto con un nivel de aceptación adecuado y la documentación necesaria para su futuro seguimiento.

En dichas metodologías, el proceso inicial y fundamental que se debe ejecutar para llevar a cabo la realización de un sistema de software es el conocido como Ingeniería de Requerimiento (*IR*) o Levantamiento de Requisitos. Este proceso es de vital importancia ya que es donde el cliente esclarece sus necesidades y enmarca los aspectos fundamentales que desea sean automatizados. Dentro de la *IR* se hace necesario hacer énfasis en los requerimientos en sí, ya que estos son las funcionalidades que finalmente deberá tener el sistema.

Conjuntamente con la metodología que se utilice para guiar el proceso de desarrollo de un sistema, existen diferentes modelos de procesos genéricos que se adecuan a las características específicas de cada proyecto, en este caso también existe una estrecha relación entre la *IR* y dichos modelos de procesos, entre los que podemos citar: Modelo en cascada, Modelo iterativo, Modelo en espiral, entre otros.

## Introducción

---

El proceso de *IR* es aplicable a cualquier tipo de sistema de software. Dentro de las categorías genéricas existentes que se le pueden dar a los sistemas de software que se desarrollan, encontramos la referente a los Software basado en Web. Este tipo de aplicaciones se caracteriza, fundamentalmente, por facilitar a los usuarios el trabajo con información almacenada a través de un servidor Web publicado en Internet o en una intranet.

Las aplicaciones Web son muy conocidas debido a la practicidad del navegador Web como cliente ligero; además, no existe la necesidad de distribuir e instalar un software en miles de clientes potenciales, permitiendo actualizar de manera más sencilla la aplicación. Podemos mencionar ejemplos de aplicaciones tales como: webmails, wikis, weblogs, tiendas en línea y la wikipedia entre otras.

Internamente dentro de las clasificaciones de sistemas Web que se pueden implementar, podemos encontrar los llamados Portales Web, los cuales se caracterizan por ofrecer al usuario, de forma fácil e integrada, el acceso a una serie de recursos y de servicios, entre los que suelen encontrarse buscadores, foros, documentos, aplicaciones, compra electrónica, etc.

Para realizar el levantamiento de requisitos de un proyecto, resulta de interés primeramente, conocer las características del sistema que se implementará, el ambiente de desarrollo y las personas que mueven o manejan la información en la entidad. Los portales Web tienen características específicas que no se pueden pasar por alto a la hora de realizar un buen levantamiento de requisitos, ya que aspectos tales como: la clasificación de los requerimientos de ese tipo de sistemas, la arquitectura de información, entre otros, no se encuentran en otro tipo de sistemas bajo el mismo comportamiento o con la misma influencia en el sistema final que se quiere obtener.

En la Universidad de las Ciencias Informáticas (*UCI*), se desarrollan diversos proyectos productivos que tributan a perfiles definidos para cada una de las áreas que los ejecutan y estos responden a las diferentes categorías existentes, entre ellos los sistemas de portales Web. En cada uno de los sistemas desarrollados se realizó inicialmente el proceso de *IR*, reflejándose positivamente más en unos proyectos que en otros.

## Introducción

---

Existen consecuencias que pueden tornarse graves al llevar a cabo un mal proceso de definición de los requerimientos de un software. Dentro de esas consecuencias podemos mencionar algunas como por ejemplo: insatisfacción por parte del cliente con la calidad del producto y/o la apariencia final del mismo; pueden quedarse funcionalidades que no se implementen en el sistema; el cliente puede no estar de acuerdo con determinados resultados obtenidos a través de una de las funcionalidades del sistema, entre otras.

El poco dominio o el uso incorrecto de las técnicas existentes para ejecutar el levantamiento de requisitos en los sistemas de portales que se han implementado en la *UCI*, en alguna medida ha traído consigo, que algunos proyectos de los que se han desarrollado no cuenten con la calidad requerida, otros no lleguen a implementarse completamente y alguno de ellos sobrepasen el tiempo previsto para su desarrollo. Por lo que ha existido la necesidad de realizar un estudio sobre la vía óptima que debe seguirse para la determinación de los requisitos de los portales Web que se implementan en la *UCI* y así garantizar la satisfacción del cliente con el producto obtenido.

Teniendo en cuenta la problemática presentada con anterioridad, enunciamos el siguiente **problema de investigación**: los procedimientos que existen para el levantamiento de requisitos durante el desarrollo de un software son muy generales y no abordan las características específicas de los proyectos de portales Web, afectando la ejecución de los mismos. Siendo el **objeto de estudio** el procedimiento de desarrollo de portales Web de la *UCI* y el **campo de acción** el levantamiento de requisitos de los proyectos de portales Web de la propia entidad.

La investigación estará encaminada a cumplir el **objetivo general** de identificar las características específicas del levantamiento de requisitos en los proyectos de portales Web de la *UCI* y proponer un procedimiento que mejore su ejecución. Como vía para lograr el objetivo propuesto en función del problema planteado, formulamos la **hipótesis** si se identifican las características específicas de los proyectos de portales Web a tener en cuenta para realizar el levantamiento de requisitos se podrá proponer un procedimiento para mejorar su ejecución.

## Introducción

---

Para darle seguimiento a la investigación y lograr una mejor organización de la misma, nos planteamos las siguientes **tareas de investigación**:

1. Elaborar el Marco Teórico de la investigación.
2. Realizar un diagnóstico del levantamiento de requisitos en los proyectos de portales Web de la UCI.
3. Desarrollar y proponer un procedimiento para la ejecución del levantamiento de requisitos en los proyectos de portales Web de la UCI.

La **población** que se estudiará está definida por los proyectos de portales desarrollados en la universidad, constituyendo cada portal, la **unidad de estudio** y tomando como muestra el 100 % de la población. La estrategia a seguir será la **investigación descriptiva**; empleando **métodos teóricos** tales como: el **histórico**, el **hipotético – deductivo** y el **sistémico**. Además, podemos mencionar dentro de los **métodos empíricos**: la **encuesta** y la **entrevista**.

Los resultados obtenidos presentan un estudio de las actividades, técnicas y metodológicas que se utilizan para la ejecución del levantamiento de requisitos en los proyectos Web; las características de los proyectos de portales Web de la UCI en cuanto al levantamiento de requisitos y un procedimiento para la ejecución de levantamiento de requisitos en ese tipo de proyectos.

El documento estará estructurado en tres capítulos, los cuáles están conformados con la información que a continuación se presenta.

En el capítulo 1 se realiza la fundamentación teórica del tema, donde se exponen los criterios y opiniones respecto al tema de investigación, expuesto en la bibliografía consultada por los autores a nivel nacional e internacional, abordando las características fundamentales sobre la Ingeniería de Requerimiento, las actividades que se desarrollan dentro de la misma, las tareas para cumplimentar cada una de las actividades, técnicas y metodologías empleadas.

En el capítulo 2 se abordan una serie de conceptos de necesaria comprensión, sobre los elementos que identifican a los sistemas de portales. Se realiza un estudio sobre las

## Introducción

---

características que se deben tener en cuenta durante el levantamiento de requisitos para los portales Web de la *UCI*, así como el análisis de los métodos utilizados y los resultados obtenidos.

En el capítulo 3 se detalla la propuesta del procedimiento a seguir para ejecutar un levantamiento de requisitos adecuado en los proyectos de portales Web de la *UCI*. Para ello se detallan las actividades, las tareas, una propuesta de técnicas y herramientas que pueden emplearse para la gestión de los requisitos. Finalmente se valida la propuesta con especialistas en el tema.

En la investigación, como aporte fundamental, se detallará el procedimiento a seguir para el levantamiento de requisitos en los sistemas de portales, se definirá una plantilla que agrupe los elementos de la arquitectura de información y se tendrá en cuenta la clasificación de los requisitos que facilite un mejor cumplimiento de los objetivos del sistema.

## Capítulo 1: Fundamentos Teóricos

### Introducción

La Ingeniería de Requerimiento (*IR*) se convierte cada vez más en una actividad primordial en el desarrollo de sistemas. En los últimos años han surgido grandes cambios en esta área, reconociendo la importancia que tiene el proceso y las consecuencias que trae consigo una definición no adecuada de los requisitos de un software. En el desarrollo de este capítulo se abordan los aspectos fundamentales tales como conceptos, actividades y características de la *IR*, así como la posición a diversos niveles en investigaciones realizadas sobre el tema. Se plantea además, la relación que existe entre la *IR* y los modelos clásicos de desarrollo de software; y con las metodologías de desarrollo. Finalmente se manifiestan las diferencias existentes entre los sistemas Web y los sistemas tradicionales.

### 1.1 La Ingeniería de Requerimiento (*IR*)

Diversos son los autores que hacen referencia a la *IR*, definiendo primeramente el concepto de Requerimiento o Requisito de software como elemento fundamental dentro de la *IR*.

En (WORTMANN, 2001) se plantea que “el término requerimiento se refiere a una necesidad del usuario o una característica necesaria, función o atributo de un sistema ideado, que puede ser percibido desde una posición externa”.

El concepto planteado por *RUP* “Rational Unified Process”, metodología que se utiliza en la *UCI* para el desarrollo de sistemas, consisten en que “un requisito es una condición o capacidad con la cual debe estar conforme el sistema”. (RATIONAL, 2003)

Y según la IEEE 1233-1998 requisito es:

“A) Una condición o capacidad que necesita el usuario para resolver algún problema o alcanzar un objetivo.

B) Condición o capacidad que debe cumplir o poseer un sistema o componente del sistema para satisfacer un contrato, estándar, especificación u otro documento formalmente impuesto.

C) Una representación documental de una condición o capacidad como en A o en B.”(IEEE, 1998). Concepto que tomaremos como referencia durante toda la investigación.

Quedando definido el elemento principal dentro de la *IR*, podemos definir entonces la *IR* como: (SUMANO, 1999)

1. “La Ingeniería de Requerimiento es la disciplina encargada de establecer los servicios que un sistema debe suministrar y las restricciones bajo las cuales debe operar”. (SOMMERVILLE, 1996)
2. “La *IR* es el proceso sistemático para desarrollar requerimientos a través de un proceso cooperativo e iterativo de análisis del problema, documentación de las observaciones resultantes en varios formatos de representación y verificación de la precisión del entendimiento ganado”. (BURG, 1997)

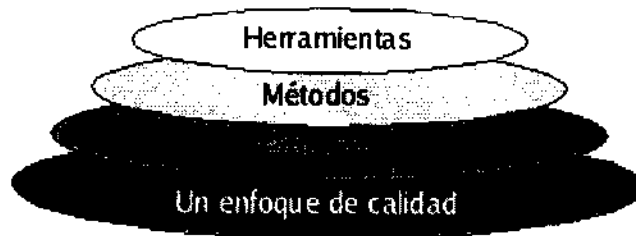
La definición a tener en cuenta en esta investigación será la que a continuación se plantea:

"Ingeniería de requerimiento es un enfoque sistémico para recolectar, organizar y documentar los requerimientos del sistema; es también el proceso que establece y mantiene acuerdos sobre los cambios de requerimientos, entre los clientes y el equipo del proyecto".(RATIONAL, 2003)

### **1.2 La *IR* y los Modelos Clásicos de Desarrollo de Software**

La Ingeniería de Software es una tecnología multicapa según se muestra en la siguiente figura.





**Figura 1.1: Capas de la Ingeniería de Software**

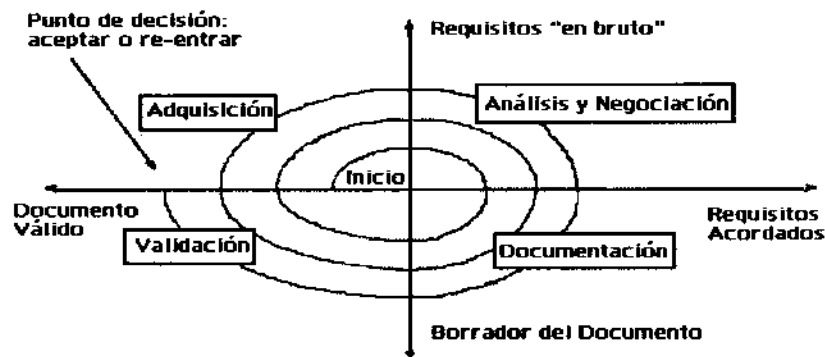
En la primera capa, de manera ascendente, podemos apreciar que cualquier enfoque de ingeniería debe ser un enfoque de calidad. La capa de proceso, es el fundamento de la Ingeniería de Software. Los métodos, indican como construir técnicamente el software y las herramientas, proporcionan un enfoque para el proceso y para los métodos.

Para la solución de problemas reales a través de la construcción de un sistema de software existen diferentes modelos de procesos, que se ajustan según la naturaleza del proyecto y de la aplicación, de los métodos y herramientas a utilizar, los cuáles son de vital importancia para la coordinación, desarrollo y control de proyectos de software real.(PRESSMAN, 2005)

Entre estos modelos podemos citar:

1. Modelo Lineal Secuencial
2. Modelo de Construcción de prototipos
3. Modelo de Desarrollo Rápido de Aplicaciones (DRA)
4. Modelos evolutivos de proceso del software:
  - a. Modelo Incremental
  - b. Modelo Espiral
  - c. Modelo WinWin
  - d. Modelo de desarrollo concurrente
5. Desarrollo basado en componente
6. Modelo de métodos formales
7. Técnicas de Cuarta Generación

De los modelos mencionados anteriormente, los más utilizados son: cascada, espiral y prototipos, por ser los más completos. La *IR* se adecua a los modelos mencionados. A continuación se observa un ejemplo de ese ajuste, permitiendo de esta manera que el proceso sea configurable y se adapte a los diferentes sistemas teniendo en cuenta sus características.



**Figura 1.2: Modelo en espiral de la Ingeniería de Requisitos**

### 1.3 La *IR*, Actividades, Técnicas y Metodologías

En un estudio realizado por (SUMANO, 1999) quedan plasmadas las actividades que se deben realizar en la *IR*, las que a continuación se mencionan:

1. Establecimiento de la necesidad del sistema
2. Factibilidad del sistema
3. Análisis de requerimientos
4. Definición de requerimientos
5. Especificación de requerimientos

Dejando claro que el documento más importante que se obtendrá será el de Especificación de Requerimientos de Software (ERS). Se hace énfasis además, en que una de las actividades más importantes es el Análisis de Requerimientos (*AR*), compuesta a su vez por otras actividades, para la cuál propone un grupo de métodos con el objetivo de lograr la captura de los requisitos; mencionando las más utilizadas:

1. Introspección

2. Entrevistas
3. Lluvia de ideas
4. Cuestionarios
5. Análisis de protocolos
6. Observación
7. Análisis del discurso
8. Elaboración de prototipos

Se relacionan diferentes metodologías que se emplean en el *AR*, pero cada una de ellas responde a un enfoque específico<sup>1</sup> que se ajusta a alguna de las actividades que se realizan dentro del *AR*:

1. “Join Application Design” (*JAD*)
2. “COHERENCE”
3. “Soft System Methodology” (*SSM*)
4. Análisis estructurado moderno
5. “Objectory”
6. Metodología de Desarrollo de Sistemas Estructurados de Datos (*DSED*)
7. COLOR-X
8. “RARE-IDIOM”

En el libro de (YOUNG, 2004) se abordan, entre otros aspectos, las actividades que se deben realizar para llevar a cabo el levantamiento de requisitos de un proyecto.

A menudo se piensa que las actividades relacionadas con los requerimientos son principalmente la captura de los requisitos y el seguimiento de estos en el ciclo de vida. En realidad existen otras actividades relacionadas con el tema, que deben ser incluidas en el ciclo de vida y se mencionan a continuación:(YOUNG, 2004)

1. Identificación de los interesados: incluye al interesado en el sistema y sus funcionalidades.
2. Comprender las necesidades del cliente y sus expectativas con el sistema: está relacionado con el proceso de elicitación de los requisitos.

---

<sup>1</sup> Existen 7 clasificaciones de enfoques: Usuarios, Análisis de Metas y Objetivos, Análisis de tareas, Análisis basado en escenarios, Análisis de formas, Enfoque de Lenguaje natural y Reuso. (Sumano 1999)

3. Identificación de los requisitos: implica la declaración de requerimientos en sentencias (oraciones) simples y suministro de ellos como un plan.
4. Clarificar y repetir los requerimientos: se realiza para asegurar que se describan las verdaderas necesidades del cliente y están en una forma que puede ser entendida y usada por los desarrolladores del sistema.
5. Análisis de requerimientos: se realiza para asegurar que estos son bien definidos y que se adaptan a los criterios de un buen requerimiento.
6. Definición de requerimiento: note que cada grupo interesado puede tener una perspectiva considerablemente diferente del sistema y de las exigencias del mismo.
7. Especificación de requerimiento: requiere que se incluyan todos los detalles exactos de cada requerimiento de modo que pueda ser incluido en un documento de especificación u otra documentación, dependiendo del tamaño del proyecto.
8. Priorizar los requerimientos: todos los requerimientos no son de igual importancia a los clientes y los usuarios del sistema planificado. Priorizar los requerimientos proporciona la oportunidad de dirigir la prioridad más alta primero y posiblemente liberar una versión de un producto con esos requerimientos más importantes y luego otra con los demás.
9. Derivar los Requerimientos: hay algunos requerimientos que ocurren debido al diseño de un sistema, pero no proporcionan una ventaja directa al usuario final.
10. División de requerimientos: clasificación de los requerimientos como aquellos que pueden ser encontrados por el hardware, el software, el entrenamiento, y la documentación, por ejemplo.
11. Asignación de requerimientos: asignación de requerimientos a subsistemas diferentes y componentes del sistema. Las asignaciones no siempre pueden estar satisfechas solamente por un subsistema o componente.
12. Rastreo de requerimientos: se necesita rastrear cada requerimiento, de modo que se pueda verificar que cada requerimiento está siendo clasificado.
13. Requerimientos administrativos: debe ser posible suprimir y modificar requerimientos durante todas las fases de diseño del sistema, desarrollo, integración, pruebas,

despliegue, y la operación. El depósito de requisitos consiste en un juego de artefactos y bases de datos.

14. Pruebas y verificación de requisitos: proceso de prueba de requerimientos, diseños, código, proyectos de prueba, y productos del sistema para asegurar que los requerimientos fueron definidos.
15. Validar los requisitos: Esto es el proceso para confirmar que los verdaderos requisitos son puestos en práctica en el sistema entregado. La validación de requisitos debería ser priorizada.

El autor plantea que existen alrededor de 40 técnicas para la elicitación de requisitos y de las mismas solo se hace énfasis en las mencionadas por (SUMANO, 1999).

Otro de los autores consultados (WIEGERS, 2006), plantea en su libro *More About Software Requirements*, que las actividades que se deben realizar en el levantamiento de requisitos son las siguientes:

Se puede subdividir el desarrollo de los requisitos en elicitación, análisis, especificación y validación. (WIEGERS, 2006). La elicitación involucra las acciones que se realizan para comprender a nuestros usuarios y descubrir sus necesidades. La actividad principal de análisis de requerimientos es para obtener los requerimientos más detallados de los requerimientos de alto nivel. La actividad de análisis también implica la creación de múltiples vistas de los requerimientos, tales como prototipos, modelos de análisis gráficos y pruebas. Además, incluye establecer las prioridades de los requerimientos y determinar los requisitos que faltan. La especificación involucra relacionar los diferentes tipos de requerimientos de forma tal que facilite la comunicación entre los trabajadores del proyecto. Finalmente, la validación asegura que dichos requerimientos estén correctos, que satisfagan las necesidades del cliente y tengan todas las características de un requerimiento de alta calidad.

El proceso de Gestión de los Requisitos (*GR*) es definido por (MCDONALD, 2005) como el proceso encargado de la identificación, asignación y seguimiento de los requisitos. Siendo las principales tareas la Recolección, la Documentación, la Verificación y la Gestión de cambios durante el proceso de determinación de los requisitos.

Se plantea además, que el proceso del ciclo de vida de la *GR* debe ser flexible y adaptable, variando el alcance y la implementación de ese ciclo de vida en dependencia de un conjunto de factores, citando uno fundamentalmente, por la importancia que tiene para esta investigación: el propósito y uso de la aplicación final.

En “Software Requirements. Encapsulation, Quality and Reuse” (LUTOWSKI, 2005) se trata el proceso de requerimientos libres, planteando que no se enfoca a los casos de uso, ni a las descripciones de escenarios, ni a los diagramas de procesos, haciendo énfasis en la especificación y prototipos de la interfaz externa del software. Se plantea además, que el proceso apropiado es definir el modelo de la entidad, delimitar la caja negra del sistema, el diagrama de contexto y finalmente el proceso de requerimientos. El proceso de requerimiento se basa en la realización de las siguientes actividades:

1. Compilación contra compra y evolución.
2. Priorizar la calidad de los requerimientos.
3. Identificación de acciones.
4. Planificación de la interfaz de usuario.
5. Definir comunicación de protocolos externos.
6. Identificar similitudes entre los requisitos.
7. Rehusar librerías.
8. Especificar respuestas.
9. Priorizar requerimientos.
10. Desarrollar prototipo de interfaz externa.
11. Obtener opiniones del usuario.
12. Diseño de objetos.

En un estudio un poco más detallado sobre la *IR* el autor (MEDINA, 2004) realiza un análisis profundo sobre las diferentes actividades, técnicas y herramientas que se deben emplear durante el proceso.

Se manifiesta que las actividades y tareas dentro de dichas actividades a ejecutar son las siguientes:

1. Obtención de requerimiento

- a. Comprender el problema que se está resolviendo
  - b. Buscar y recolectar información
  - c. Definir los límites y restricciones del sistema
  - d. Identificar a las personas o usuarios interesados por el sistema
  - e. Recolectar y clasificar requerimientos
2. Análisis de requerimiento
    - a. Priorizar los requerimientos
    - b. Encontrar dependencias entre los requerimiento y etiquetar los requerimientos
    - c. Resolver conflictos entre requerimientos
    - d. Negociar con flexibilidad
3. Especificación de requerimiento
    - a. Especificar los requerimientos funcionales y no funcionales
    - b. Determinar el tipo de estándar
    - c. Elegir la herramienta de especificación
    - d. Utilizar modelos y diagramas
    - e. Utilizar cualquier información de soporte o guía
4. Validación de requerimiento
    - a. Verificar validez
    - b. Verificar consistencia
    - c. Verificar integridad
    - d. Verificar realismo
    - e. Generar casos de prueba
5. Administración de requerimiento
    - a. Administrar los cambios de los requerimientos
    - b. Establecer políticas de rastreo
    - c. Control de versiones del documento de requerimientos (MEDINA, 2004)

En el estudio aparece además, una explicación detallada de cada una de las técnicas que se pueden emplear en las diferentes actividades de la *IR*, coincidiendo con las que se han planteado anteriormente. Sucediendo así mismo para las metodologías enunciadas en la tesis

doctoral, las cuáles coinciden con las metodologías ya planteadas y no haciendo referencia a dichas metodologías para sistemas Web.

Otro de los actores consultados fue (AURUM y WOHLIN, 2005), los que plantean en su libro que, en la gestión de requisitos debe lograrse un buen entendimiento por parte de los clientes, de los requisitos que se especifican en el producto de software, manifestando como prácticas esenciales para lograr ese objetivo las siguientes: Elicitación, especificación y modelación de los requerimientos; Priorizar requerimientos; Dependencias entre requerimientos y Análisis de Impacto; Negociación de los requerimientos y Control de Calidad de los requerimientos.

En un estudio comparativo sobre la *IR* en aplicaciones para la Web, se especifican las actividades que se realizan durante el mismo y una propuesta de las técnicas que se pueden utilizar durante estas actividades. A continuación se explica, en un resumen, el estudio realizado por los autores.

Independientemente del tipo de aplicaciones que se vaya a implementar, el equipo de desarrollo o los analistas pertenecientes al mismo, tienen que enfrentarse al proceso de definición de los requisitos, a través de los cuales quedarán identificados los requisitos que el sistema debe cumplir para satisfacer las necesidades de los usuarios. (ESCALONA y KOCH, 2002)

El proceso de especificación de los requisitos debe dividirse en 3 actividades principales, las cuales se mencionan a continuación:

1. Captura de requisitos
2. Definición de requisitos
3. Validación de requisitos

Se detalla además, para cada una de las actividades mencionadas, las técnicas que deben emplearse.



**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.

Dentro de las técnicas podemos mencionar:

1. Cuestionario
2. Desarrollo Conjunto de Aplicaciones (*JAD*)
3. Tormenta de ideas “Brainstorming”
4. Mapeo de conceptos “Concept mapping”
5. Diseños “Sketches” y Tablas de Historial “Storyboards”
6. Casos de uso
7. Cuestionarios y Listas de verificación “checklists”
8. Comparación de terminología

Las técnicas mencionadas anteriormente representan un conjunto de las más utilizadas, aunque existen otras como: análisis de otros sistemas, estudio de la documentación, etc., así como la combinación de varias de estas técnicas.

♣ **Definición de requisitos:** actividad mediante la cuál se definen los requisitos del software.

Entre las técnicas podemos mencionar:

1. Lenguaje natural
2. Glosario y ontologías
3. Plantillas o patrones
4. Escenarios
5. Casos de uso
6. Lenguajes Formales

**Validación de requisitos:** tiene como misión demostrar que la definición de los requisitos precisa realmente el sistema que el usuario necesita o el cliente desea. Es necesario asegurar que el análisis realizado y los resultados obtenidos de la etapa de definición de requisitos son correctos. Pocas son las propuestas existentes que ofrecen técnicas para la realización de la validación y muchas de ellas consisten en revisar los modelos obtenidos en la definición de requisitos con el usuario para detectar errores o inconsistencias. Aún así, existen algunas técnicas que pueden aplicarse para ello:

1. “Reviews” o “Walk-throughs”
2. Auditorias
3. Matrices de trazabilidad
4. Prototipos

La diferencia radica en las propuestas metodológicas que se pueden aplicar durante la *IR* para aplicaciones tradicionales y para las aplicaciones Web. Por lo que se plantean un conjunto de metodologías que se aplican para este tipo de sistemas, las cuáles se describirán en el próximo epígrafe, analizando el comportamiento de la *IR* en cada una de ellas.

En el libro “Managing Software Requirements: A Use Case Approach”, (LEFFINGWELL y WIDRIG, 2003), *CMM* provee una vista comprensiva de las actividades que deben ser aplicadas para mejorar la calidad del software e incrementar la productividad. La gestión de requisitos es una parte integrada de este proceso donde el manejo de los requerimientos como entidades vivas es el centro del desarrollo de las actividades. Tanto la elicitación como los requerimientos, son documentados con la misma importancia que el código del producto. Este proceso pone al equipo en función del control del proyecto y ayuda a los miembros del equipo a gestionar el proyecto y el alcance del mismo. Por último, el manejo activo de los cambios en los requerimientos mantiene el proyecto bajo control y ayuda a asegurar la calidad del software. Además, se provee una importante validación del concepto de gestión de requerimientos, tomando en cuenta algunos consejos de alto nivel para insertar requerimientos orientados al proceso de desarrollo cíclico. Esto no nos dice como hacer la gestión de requerimientos.

La *ISO 9000* hace énfasis en los requerimientos al principio de los esfuerzos de desarrollo para mantener o ayudar esta tarea. Si los esfuerzos de diseño técnico y desarrollo están orientados a la presentación, la organización es más probable que produzca un sistema donde las especificaciones o requerimientos son algo más confiable después de pruebas exhaustivas y de las actividades de validación al final del ciclo de vida para asegurar la calidad. Como *SEI-CMM*, *ISO 9000* no nos dice específicamente como hacer realmente la gestión de requerimientos, no obstante los miembros del equipo van a estar obligados a definirlos,

implementarlos y adherirlos a un proceso gestión de requerimientos efectivo, en los cuales ellos mismos son declarados para cumplir el objetivo del mismo. (LEFFINGWELL y WIDRIG, 2003)

De lo enunciado anteriormente se puede plantear que no se tendrá en cuenta el enfoque planteado por Young ya que se define un ámbito muy abarcador de las actividades que se realizan en la *IR*. Así como tampoco se tendrá en cuenta la propuesta de Wiegers ni de Lutowski, donde se hace un buen análisis sobre las actividades que se deben realizar en la *IR* pero no se hace referencia a las técnicas o metodologías que se debe emplear para ejecutar el proceso. En el caso de McDonald solo se plantea superficialmente las actividades pero no se hace referencia tampoco a las técnicas ni propuestas metodológicas. En lo planteado por Aurum y Wohlin realizan una descripción muy parecida a la realizada por Medina, acerca de las actividades y tareas a ejecutar dentro de ellas, por lo que se tomará en cuenta en los aspectos que planteen y que difieran de la propuesta realizada por Medina.

Sin embargo, en el estudio realizado por (Sumano, 1999) al igual que el de (Medina, 2004), ambos describen detalladamente las actividades que se deben realizar, las técnicas que se pueden emplear y las metodologías durante el proceso de *IR*. Pero no se puede dejar de mencionar que en ambos casos no se diferencia la realización del proceso para sistemas tradicionales y para sistemas Web. En el caso de (Escalona y Koch, 2002), se plantea un enfoque para los sistemas Web que debe tenerse en cuenta, ya que son los únicos autores que referencia el proceso para este tipo de sistemas. Por lo que para la propuesta que se detallará en el próximo capítulo de esta investigación, se tomará como base las actividades planteadas por Escalona y Koch, pero siguiendo el enfoque de tareas a realizar dentro de las mismas planteado por Medina, adecuándolas a las características específicas de los sistemas de portales Web y apoyándonos en la clasificación que hacen también Escalona y Koch de los requisitos de un sistema para la Web.

### 1.4 La IR y las Metodologías de Desarrollo

Existen metodologías basadas en una combinación de los modelos de proceso genéricos (cascada, evolutivo, incremental, etc.); dichas metodologías se clasifican en Estructuradas y Orientadas a Objetos, si tomamos como criterio las notaciones usadas para especificar artefactos producidos en las actividades de Análisis y Diseño; sin embargo, considerando su filosofía de desarrollo, aquellas metodologías con mayor énfasis en la planificación y control del proyecto, en especificación precisa de requisitos y modelado, reciben el sobrenombre de Metodologías Tradicionales (o peyorativamente denominada Metodologías Pesadas, o Peso Pesado). Otras metodologías, denominadas Metodologías Ágiles, están más orientadas a la generación de código con ciclos muy cortos de desarrollo, haciendo especial hincapié en el trabajo en equipo e involucrando activamente al cliente en el proceso.

De una forma u otra, cada una de estas metodologías hace referencia en su ciclo de desarrollo como actividad o flujo de trabajo, al levantamiento de requisitos. Por lo que se manifiesta que, independientemente de la metodología que se seleccione para darle seguimiento al ciclo de desarrollo del software, se realizará en algún momento del proceso, el levantamiento de requisitos.

Existen diferentes metodologías y cada una de ellas se ajusta según la clasificación del tipo de proyecto que se vaya a desarrollar. En el caso particular de las aplicaciones Web, existen un número diverso de metodologías que describen como llevar a cabo el proceso de desarrollo de ese tipo de sistemas, ya que presentan un conjunto de características que no son fáciles de modelar a través de una metodología para software tradicional. A continuación se relacionan algunas de las metodologías existentes para la modelación de sistemas Web, explicando fundamentalmente las más utilizadas y/o reconocidas en la construcción de sistemas de ese tipo. Estas metodologías son las siguientes:

1. *WSDM* “Web Site Design Method”
2. *SOHDM* “Scenario-based Object-Oriented Hypermedia Design Methodology”

3. *RNA* “Relationship-Navigational Analysis”
4. *HFPM* “Hypermedia Flexible Process Modeling”
5. *OOHDM* “Object Oriented Hypermedia Design Model”
6. *UWE* “UML-Based Web Engineering”
7. *W2000*
8. *UWA* “Ubiquitous Web Applications”
9. *NDT* “Navigational Development Techniques”
10. “Design-driven Requirements Elicitation”

Se abordarán las características fundamentales de algunas de ellas y se tratarán además, las características de la metodología *RUP*, ya que es la que se emplea actualmente en la *UCI* para modelar los sistemas de software.

### **1.4.1 WSDM “Web Site Design Method”**

El método para diseño de sitios Web (*WSDM*) se basa fundamentalmente en la definición del sistema en base a los grupos de usuarios. Los autores dividen la creación de los sitios en dos grupos: Kiosco Web y Aplicación Web. Tras esta división los autores especifican que su propuesta está orientada a la primera de las clasificaciones, no resultando adecuada para la segunda.

El modelo de diseño de sitios Web se divide en cuatro fases: modelo de usuario, diseño conceptual, diseño de la implementación e implementación. A su vez, el modelo de usuario se divide en dos subfases: clasificación y descripción. Por su parte el diseño conceptual se divide en otras dos subfases: modelado de objetos y diseño navegacional.(ESCALONA, 2001)

En la primera fase se intenta detectar los perfiles de usuarios que se van a presentar en la aplicación. Para ello, se deben realizar dos tareas:

1. Clasificación de usuarios: identificar y clasificar los usuarios que accederán al sistema.
2. Descripción de los grupos de usuarios: describir con más detalles a los usuarios detectados en la etapa anterior.

En la fase de diseño conceptual se propone crear primero un modelo conceptual y luego un modelo de navegación, apoyándose para ello en las etapas siguientes:

1. Modelo de objetos: modelar formalmente los requisitos de información.
2. Diseño Navegacional: obtener un modelo para representar las posibilidades de navegación del sistema.

En el diseño de la implementación se debe diseñar la interfaz y el entorno de usuario, pero los autores de *WSDM* no proponen ninguna técnica en específico para la ejecución de este modelo. Finalmente se debe desarrollar la fase de implementación para obtener el ejecutable del sistema, no proponiendo además, ninguna técnica para ello.

#### **1.4.2 RNA "Relationship-Navigational Analysis"**

*RNA* plantea una secuencia de pasos centrándose fundamentalmente en el flujo de trabajo de análisis. El proceso de trabajo que presenta se basa en la realización de las siguientes fases: (ESCALONA y KOCH, 2002)

1. Fase 1 - Análisis del entorno: El propósito de esta fase es el de estudiar las características de la audiencia. Consiste en determinar y clasificar a los usuarios finales de la aplicación en grupos según sus perfiles.
2. Fase 2 - Elementos de interés: En esta fase se listan todos los elementos de interés de la aplicación. Por elementos de interés se entienden los documentos, las pantallas que se van a requerir, la información, etc.
3. Fase 3 - Análisis del conocimiento: Esta fase consiste en desarrollar un esquema que represente a la aplicación. Para ello *RNA* propone identificar los objetos, los procesos y las operaciones que se van a poder realizar en la aplicación, así como las relaciones que se producen entre estos elementos.
4. Fase 4 - Análisis de la navegación: En esta fase el esquema obtenido en la fase anterior es enriquecido con las posibilidades de navegación dentro de la aplicación.

5. Fase 5 - Implementación del análisis: Una vez obtenido el esquema final en el que ya se encuentran incluidos los aspectos de navegación, se pasa el esquema a un lenguaje entendible por la máquina.

La propuesta de *RNA* es quizás una de las que más ha resaltado la necesidad de trabajar con la especificación de requisitos, incluyendo tareas como el análisis del entorno y de los elementos de interés. Además, resulta interesante pues plantea la necesidad de analizar los requisitos conceptuales de manera independiente a los navegacionales.

### 1.4.3 HFPM “Hypermedia Flexible Process Modeling”

La propuesta de *HFPM* describe un proceso detallado que cubre todo el ciclo de vida de un proyecto software. *HFPM* propone un total de trece fases para las cuales se especifican a su vez una serie de tareas. Para este estudio es principalmente relevante la primera fase, denominada de modelado de requisitos, cuyas tareas son las siguientes:(ESCALONA y KOCH, 2002)

1. Descripción breve del problema. No indica ninguna técnica concreta pudiendo realizarse esta descripción mediante el lenguaje natural.
2. Descripción de los requisitos funcionales mediante casos de uso.
3. Realizar un modelo de datos para esos casos de uso, proponiendo el uso de un modelo de clases.
4. Modelar la interfaz de usuario. Para ello, se propone el uso de sketches y prototipos que permitan presentar los datos al usuario.
5. Modelar los requisitos no funcionales. En éstos se incluyen la navegación, la seguridad, etc.

Se puede observar que la propuesta de *HFPM* ofrece mayor detalle a la hora de realizar el tratamiento de los requisitos. Sin embargo, no ofrece técnicas concretas, especialmente a la hora de trabajar con los requisitos no funcionales.

### 1.4.4 UWE “UML-Based Web Engineering”

Es una propuesta metodológica basada en el Proceso Unificado y *UML* para el desarrollo de aplicaciones Web, que cubre todo el ciclo de vida de este tipo de aplicaciones centrandose además, su atención en aplicaciones personalizadas.(ESCALONA y KOCH, 2002)

El proceso de captura de requisitos de *UWE* distingue entre la tarea de licitar requisitos, definir y validar los requisitos. El resultado final de la captura de requisitos en *UWE* es un modelo de casos de uso acompañado de documentación que describe los usuarios del sistema, las reglas de adaptación, los casos de uso y la interfaz. *UWE* clasifica los requisitos en dos grandes grupos: funcionales y no funcionales. Los requisitos funcionales tratados por *UWE* son:

1. Requisitos relacionados con el contenido
2. Requisitos relacionados con la estructura
3. Requisitos relacionados con la presentación
4. Requisitos relacionados con la adaptación
5. Requisitos relacionados con los usuarios

Además, propone como técnicas apropiadas para la captura de los requisitos de un sistema Web las entrevistas, los cuestionarios, los “checklists” y los casos de uso, los escenarios y el glosario para la definición de requisitos. Para la validación propone “walk-throughs”, auditorias y prototipos.

### 1.4.5 W2000

*W2000* ha sido la evolución a la orientación a objetos de su antecesora *HDM*. Con respecto a *HDM* existen dos diferencias básicas. La primera de ellas es que *HDM* no era realmente una propuesta metodológica sino un modelo enriquecido del diagrama Entidad-Relación, *W2000* sí que propone un ciclo de vida para el desarrollo de sistemas Web. La otra gran diferencia es que *W2000* se centra en el paradigma de la orientación a objetos. Pero a pesar de estas



diferencias, todas las aserciones que propuso *HDM* han sido aceptadas como válidas en *W2000* y han sido adecuadas a la orientación a objetos.

El ciclo de vida de *W2000* es mostrado en la figura 1.3. El proceso comienza con una fase de análisis de requisitos basado principalmente en los casos de uso. Con el conocimiento adquirido durante la fase de requisitos, se pasa a la fase de diseño de hipermedia. En ella, se realizan dos modelos: el modelo conceptual y el modelo navegacional. Para ello, los autores han modificado y extendido algunos modelos de *UML* como el diagrama de clases o el diagrama de estados. Por último, se pasa a una fase de diseño funcional, en el que se ha adaptado el diagrama de secuencias para expresar la funcionalidad del sistema.

Una característica importante de *W2000*, que se resaltará más adelante, es que separa el aspecto de la navegación y la estructura de la información desde las primeras fases del ciclo de vida. Así como puede verse en la figura 1.3 existe un análisis de requisitos funcionales y un análisis de requisitos navegacionales. Sin embargo, la técnica usada para ambos tipos de requisitos son los casos de uso, sin especificar realmente cómo se pueden llegar a separar, tratar, identificar y licitar ambos tipos de requisitos de manera específica.(ESCALONA, 2004)

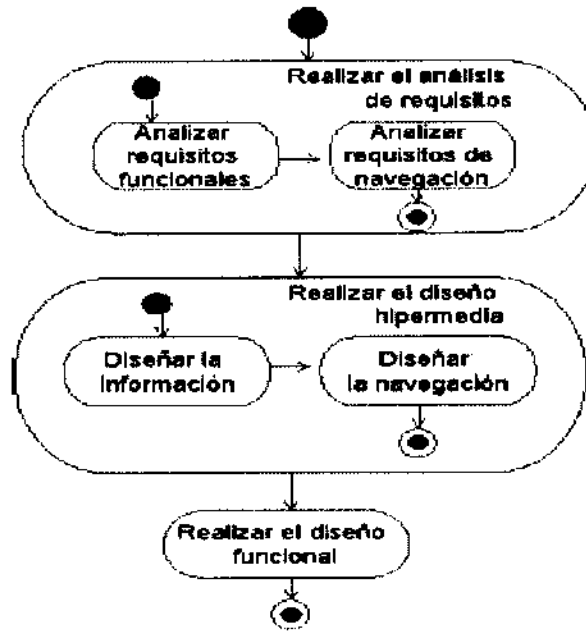


Figura 1.3: Proceso de desarrollo en W2000

#### 1.4.6 UWA “Ubiquitous Web Applications”

UWA ha nacido de la colaboración entre diferentes grupos de trabajo, por lo que resulta realmente una agrupación de propuestas y técnicas. En concreto, la propuesta de W2000 se encuentra incluida en UWA. Sin embargo, W2000 ha sido incluida en UWA sólo en la fase de diseño hipertexto, siendo ambas propuestas diferentes en la fase de definición de requisitos. Por esta razón han sido incluidos en este trabajo en forma separada. El proceso de captura de requisitos en UWA comienza definiendo los diferentes roles de usuario que pueden interactuar con el sistema, los objetivos globales del sistemas y la relación entre éstos.

El proceso continúa haciendo un refinamiento de esos objetivos globales, concretándolos en subobjetivos. Estos subobjetivos son estudiados y refinados para detectar conflictos entre ellos. De esta forma, se concretizan aún más dividiéndolos en requisitos. Los requisitos son clasificados en varios tipos: de contenido, de estructura de contenido, de acceso, de

navegación, de presentación, de operaciones de usuario y de operaciones del sistema. De esta forma, los requisitos se van refinando hasta que solo pertenezcan a uno de estos grupos. Y finalmente los requerimientos son asignados a artefactos de diseño o a reglas de customización.

Para definir los objetivos, *UWA* propone una notación propia, basada en una plantilla. La definición de los actores y la relación con los objetivos se hace usando un diagrama basado en casos de uso. Por último, para definir y refinar los subobjetivos y los requisitos, utilizan una notación gráfica propia que denominan grafo de refinamiento de objetivos, el refinamiento de este grafo permite ir representando la relación entre los requisitos y hacer un seguimiento para validar la consecución de los objetivos del sistema. Una vez que los requisitos son detectados, hacen uso de *XML* para definirlos de una manera formal. (ESCALONA y KOCH, 2002)

### 1.4.7 NDT “Navigational Development Techniques”

*NDT* es una propuesta que da mucha importancia a la fase de la Ingeniería de Requisitos. El proceso de desarrollo que propone *NDT* se podría catalogar como un proceso “bottom-up”, en el sentido de que defiende el realizar un gran esfuerzo en la tarea de captura, definición y tratamiento de requisitos, con idea de sentar una base adecuada y correcta, con el menor número de errores posibles. En la siguiente, figura 1.4 (ESCALONA, 2004), se muestra mediante un diagrama de actividades, el proceso de *IR* propuesto por *NDT*. Como puede verse, muchas de las actividades principales del proceso, se dividen a su vez en tareas. El proceso comienza con una primera actividad en la que se hace un estudio del entorno del futuro sistema, se preparan y realizan las entrevistas, determinando cuáles son los objetivos del mismo. Tras esta primera actividad, se suceden un conjunto de actividades en las que se captura y definen los diferentes requisitos del sistema. *NDT* contempla 5 tipos de requisitos:

1. Requisitos de almacenamiento de información, que a su vez se dividen en dos: requisitos de almacenamiento y naturalezas. Con ellos, se define qué información va almacenar el sistema y qué relaciones se establecen entre ellas.

2. Requisitos de actores. En este grupo de requisitos se definen qué roles de usuarios pueden interactuar con el sistema, así como las relaciones que se establecen entre ellos.
3. Requisitos funcionales, que van a representar las necesidades de funcionalidad que ofrece el sistema.
4. Requisitos de interacción, que definen cómo interactúa cada rol de usuario con el sistema y cómo se puede navegar a través del mismo. Los requisitos de interacción se definirán mediante las frases y los prototipos de visualización. Las frases permitirán establecer los criterios de recuperación y los prototipos de visualización indican cómo se mostrará la información, cómo se podrá navegar en el sistema y cómo se ofrecerán las posibilidades funcionales del sistema.
5. Requisitos no funcionales, que van a englobar otros requisitos no tratados anteriormente.

Tras el estudio de los requisitos, *NDT* propone realizar una validación de los mismos. Si al validar los requisitos no se detectan errores o incongruencias, se generará el Documento de Requisitos del Sistema (DRS), que es el resultado de la fase y que, además de ser el producto de la misma, es la base para la realización del análisis. Si durante la validación se detectaran errores o incongruencias, el flujo de trabajo se dirigiría hacia la actividad en la que se hayan detectado dichos errores o incongruencias.

Por ello, a pesar de que el proceso de *IR* aparece en la figura anterior como secuencial, *NDT* permite la vuelta atrás en el proceso. En muchas ocasiones, esta vuelta atrás no hay que realizarla en la validación. Puede darse el caso de que durante la definición de los prototipos de visualización, por ejemplo, se detecten errores en la definición de los requisitos funcionales, en cuyo caso se puede volver para rectificarlo. Estas posibilidades de vuelta atrás no han sido, sin embargo, detalladas en el gráfico para su mejor entendimiento.

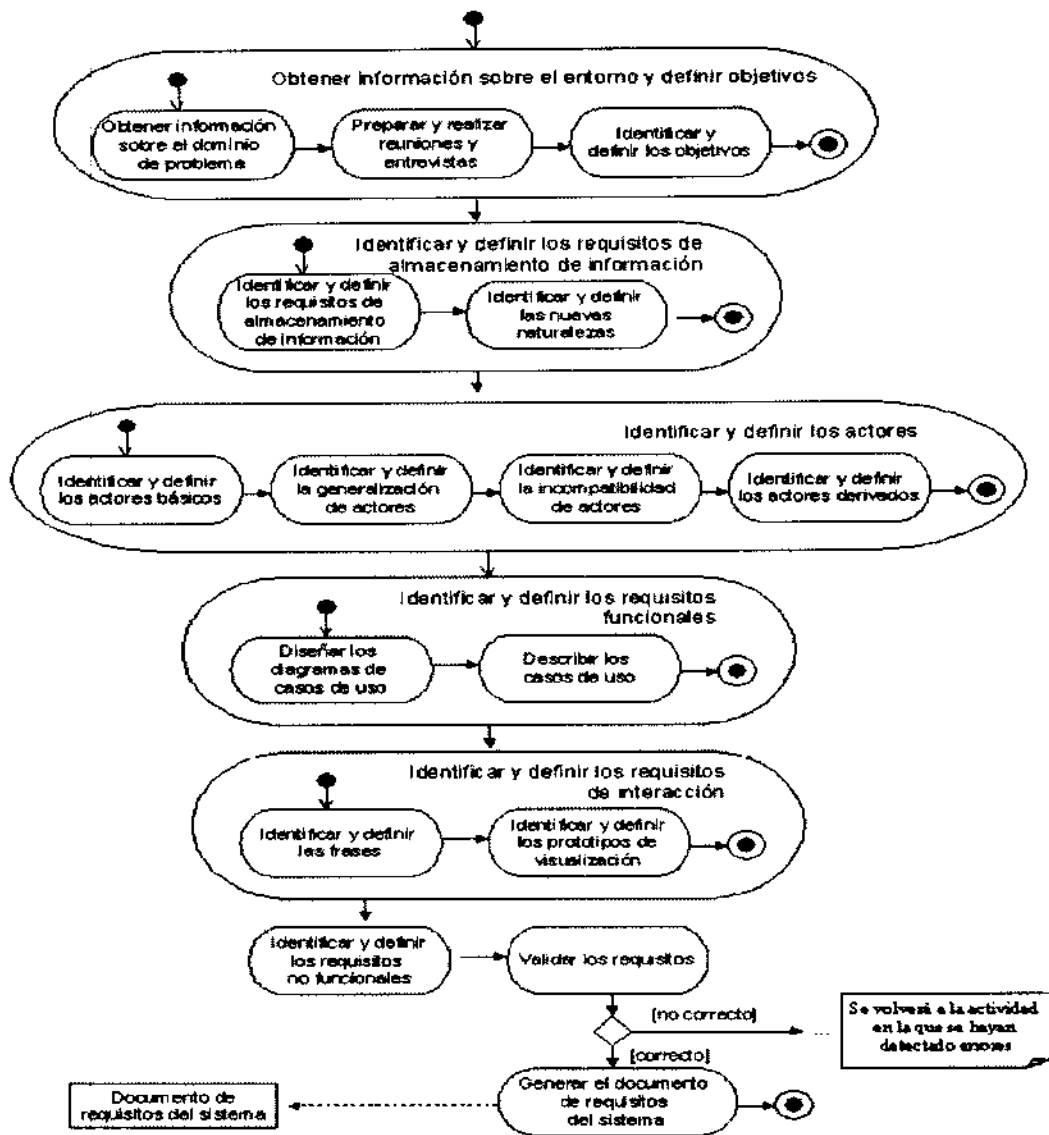


Figura 1.4: Ingeniería de Requerimiento en NDT

### **1.4.8 Design-driven Requirements Elicitation**

La propuesta consiste en realizar la captura, definición y validación de requisitos durante el proceso de diseño. Ello hace necesario que las actividades de diseño sean realizadas de modo que los requerimientos pueden ser tratados y administrados. El proceso se basa en el uso de prototipos para ayudar al cliente en la exploración de las posibles soluciones y de los problemas que tienen que ser resueltos. Los usuarios o clientes definen sus requerimientos basándose en la observación o trabajo con estos prototipos. Es un proceso iterativo que consiste en reducir la incertidumbre del cliente. El ciclo tiene tres fases: evaluación, especificación y construcción.

Este proceso fue definido sobre la base de un exhaustivo análisis de “best practices” en el desarrollo de aplicaciones comerciales para el entorno Web. Esta metodología trata a todos los requisitos de la misma manera; estos requisitos son: de contenido, de protocolo de interfaces, de estructura navegacional, de “look and feel”, de representación interna de datos, de versionamiento, de control de cambios, de seguridad, de gestión de contenido, de acceso de control, de eficiencia, de monitoreo del usuario, de soporte de funcionalidad, de adaptación del sistema, de identificación del usuario y sus derechos de acceso, etc. (ESCALONA y KOCH, 2002)

### **1.4.9 RUP “Rational Unified Process”**

El Proceso Unificado es un proceso de desarrollo de software que consiste en realizar un conjunto de actividades con el objetivo de transformar los requisitos del usuario en un sistema de software.(JACOBSON, I., BOOCH, G. y RUMBAUGH, J., 2000)

El Proceso Unificado utiliza el lenguaje de modelado *UML*, para la representación de los esquemas de un sistema de software, formando parte esencial del mismo y sus desarrollos fueron paralelos. Existen tres características que sustentan el *RUP*, caracterizando su ciclo de

vida: centrado en la arquitectura, iterativo e incremental y dirigido por casos de uso. *RUP* como proceso define como sus elementos fundamentales:

1. Trabajadores (quién): define el comportamiento y responsabilidades de un individuo, grupo de individuos, sistema automatizado o máquina que trabajan en conjunto como un equipo realizando las actividades.
2. Actividades (cómo): propósito claro, es realizada por un trabajador y manipula elementos.
3. Artefactos (qué): productos tangibles que son producidos, modificados y usados por las actividades.
4. Flujo de actividades (cuándo): secuencia de actividades realizadas por los trabajadores y que producen un resultado de valor observable.

*RUP* está compuesto por cuatro fases, y un grupo de flujos de trabajo los que a continuación se mencionan.

Fases:

1. Conceptualización (Concepción o Inicio)
2. Elaboración
3. Construcción
4. Transición

Flujos de trabajo

1. Modelamiento del negocio
2. Requerimientos
3. Análisis y diseño
4. Implementación
5. Prueba (Testeo): Busca los defectos a lo largo del ciclo de vida.
6. Instalación
7. Administración del proyecto
8. Administración de configuración y cambios
9. Ambiente

La vida de un sistema transcurre a través de ciclos de desarrollo, desde su nacimiento hasta su muerte, en cada ciclo se repite el Proceso Unificado de Desarrollo, cada ciclo consta de cuatro fases, mencionadas anteriormente y cada ciclo concluye con una versión del producto.

De las metodologías analizadas se puede decir que, en el caso de *RUP*, a pesar de ser una metodología robusta, empleada mayormente a todos los niveles en nuestro país para darle seguimiento al ciclo de vida de los sistemas que se implementan, parte del supuesto de que el ciclo de vida es el mismo para todos los tipos de sistemas (los tradicionales y los sistemas Web), definiendo un conjunto de artefactos que se deben generar y perfeccionar por cada iteración que se ejecute de cada fase y flujo de trabajo que propone la misma, pero solo diferenciando para los sistemas Web, los estereotipos de los artefactos construidos a través del lenguaje de modelación que utiliza la misma, *UML*. No tiene en cuenta además, las características de este tipo de sistemas, así como tampoco los tipos de requisitos que son necesarios detallar para un mejor entendimiento y modelación de un sistema Web, planteando fundamentalmente, la clasificación de estos en funcionales y no funcionales.

En el caso de las restantes metodologías tratadas, las cuáles si se adecuan más al tipo de sistemas Web, solo algunas de ellas hacen referencia detallada a cómo debe realizarse el proceso de *IR*. La metodología "Design-Driven Requirement Elicitation" propone una nueva vía para llevar a cabo el proceso, la cuál no se tendrá en cuenta a pesar que maneja los requisitos, aunque con otra nomenclatura, que gestiona *NDT*.

En el caso de *HFPM*, *UWE*, *W2000*, *UWA* y *NDT* son las que mejor describen el proceso de *IR* como parte del ciclo de vida, proponiendo actividades a realizar durante el mismo y en algunas de ellas un conjunto de técnicas que se emplean para lograr la calidad del proceso. Pero hay que resaltar el caso de *NDT* que es la única que trata la clasificación de los requisitos propuestos por Escalona y Koch; además, es la propuesta metodológica que aborda elementos relacionados con las actividades que se deben realizar durante la *IR* y propone técnicas para realizar algunas de las actividades. Por lo que se analizará para la propuesta, la inclusión de esta metodología.



### 1.5 Los Sistemas Tradicionales y los Sistemas Web

Existen un grupo de características propias en el desarrollo de sistemas Web que son de vital importancia ya que es precisamente en ellas donde radica la diferencia de ese tipo de sistemas con los sistemas de software tradicionales, siendo necesario diferenciar el proceso de *IR* para este tipo de aplicaciones. Estas características son:

1. La mayoría de los sitios Web seguirán teniendo la característica de orientación a la documentación: en lo previsible no hay razón para pensar que se abandone esta característica básica surgida desde los inicios de la Web. Las páginas se generan estática o dinámicamente.
2. Las aplicaciones Web continuarán focalizadas en la interfaz de usuario y en su apariencia y estética “look & feel”, favoreciendo la creatividad visual y la incorporación de multimedia. Muchos sitios están dirigidos por objetivos de presentación, promoción de productos y servicios, y metas de negocios; esto parece ir en incremento.
3. La mayoría de los sitios Web seguirán siendo orientados al contenido. Esta característica se observa en muchas aplicaciones de software que ofrecen documentación y ayuda en línea; sin embargo el alcance de la incorporación de contenido en un sitio Web es mucho más abarcador. Además, el proceso de autoría es generalmente propio de desarrollos Web o CD-ROMs multimediales.
4. Un sitio Web puede atender una multiplicidad de perfiles de usuario y sus distintos requerimientos. El construir un sitio orientado a audiencias no siempre es una tarea sencilla.
5. El medio donde corren las aplicaciones basadas en la Web es generalmente más impredecible que el medio donde se ejecutan las aplicaciones de software tradicionales. Esto puede afectar la percepción y aceptabilidad que tiene el usuario del artefacto.
6. Un sitio Web implica un mayor acercamiento entre arte y ciencia que lo que puede implicar un artefacto de software.

7. En general un proyecto Web de mediana y gran escala requiere mayor multiplicidad de tipos y niveles de habilidades (y roles) de los participantes que un proyecto de software.
8. La Web representa un nuevo paradigma en la distribución de documentación y software.
9. La mayoría de los desarrollos en la Web requieren estar operativos en un corto lapso de tiempo. Esto puede dificultar la aplicación del mismo nivel de rigor formal o semiformal de varias actividades del proceso que en un proyecto tradicional. (OLSINA, 1999)

Otra de las razones por las que es necesario diferenciar el proceso de *IR* para sistemas Web, es por las clasificaciones que se tiene de los tipos de requerimientos existentes. En (MEDINA, 2004) se plantea que las clasificaciones de los tipos de requerimientos que existen, está enfocada a la audiencia a quienes van dirigidos y a las características del sistema. En este caso se puntualizará las diferencias en cuanto a la clasificación de requerimientos en base a las características del sistema para sistemas tradicionales y para los sistemas Web.

Generalmente para los sistemas tradicionales durante el proceso de *IR*, los requerimientos se clasifican en:

1. Funcionales y
2. No funcionales ( JACOBSON, I., BOOCH, G. y RUMBAUGH, J., 2000)

Existiendo una subdivisión para los requerimientos no funcionales entre los que podemos citar:

1. Requerimientos del producto
2. Requerimientos organizacionales
3. Requerimientos externos
4. Requerimientos de datos
5. Requerimientos de interfaz, entre otros.

Pero existe dentro de los requerimientos funcionales una clasificación que se debe tener en cuenta para los sistemas basados en Web. Estos tipos de requerimientos son:

1. Requisitos de datos: también denominados requisitos de contenido, requisitos conceptuales o requisitos de almacenamiento de información. Éstos requisitos responden a la pregunta de qué información debe almacenar y administrar el sistema.
2. Requisitos de interfaz (al usuario): también llamados en algunas propuestas requisitos de interacción o de usuario. Responden a la pregunta de cómo va a interactuar el usuario con el sistema.
3. Requisitos navegacionales: recogen las necesidades de navegación del usuario.
4. Requisitos de personalización: describen cómo debe adaptarse el sistema en función de qué usuario interactúe con él y de la descripción actual de dicho usuario.
5. Requisitos transaccionales o funcionales internos: recogen qué debe hacer el sistema de forma interna, sin incluir aspectos de interfaz o interacción. También son conocidos en el ambiente Web como requisitos de servicios.
6. Requisitos no funcionales: son, por ejemplo, los requisitos de portabilidad, de reutilización, de entorno de desarrollo, de usabilidad, de disponibilidad, etc. (ESCALONA y KOCH, 2002)

Una última, pero no menos importante razón, que se debe tener en cuenta en la diferenciación del levantamiento de requisitos para sistemas tradicionales y sistemas Web, es el caso de la Arquitectura de Información (*AI*). El proceso de *AI* no es más que el arte y la ciencia de organizar información para ayudar a las personas a desempeñar eficientemente sus necesidades de información. Ello conlleva investigación, análisis, diseño e implementación.

Debe tenerse en cuenta en cada uno de los sistemas de software que se vaya a implementar, pero en el caso de los sistemas Web, debe prestársele especial atención, ya que es uno de los elementos más importantes por las características que posee un sistema Web, en el que no se conoce, muchas veces, los usuarios que accederán al sitio, cuáles son sus costumbres de información y mucho menos el impacto que tendrá en ellos el sitio cuando accedan al mismo.

En el segundo capítulo se detallará un poco más los aspectos necesarios a considerar en la *AI* de un sistema Web, para una mejor comprensión y poder analizar cuáles de ellos se deben tener en cuenta durante el levantamiento de requisitos del proyecto.

### **Conclusiones**

Hasta esta etapa de la investigación se llegó a la conclusión de que no existen todos los componentes o artefactos necesarios para modelar la arquitectura de información y otros elementos característicos de los sistemas portales Web; así como tampoco existe un procedimiento detallado que fundamente cómo realizar el levantamiento de requisitos en ese tipo de sistemas.

# Capítulo 2: Caracterización del Entorno y Métodos Utilizados.

## Introducción

En el capítulo anterior quedaron detalladas las diferentes formas de ejecutar el proceso de *IR* y se arribó al enfoque que se tomará como base para la propuesta. Estamos en condiciones en este capítulo de abordar algunos conceptos relacionados con los sistemas Web específicamente, necesarios para la comprensión de los aspectos que se tratarán como particularidades del levantamiento de requisitos para un sistema de portales. Se caracterizará la manera de realizar el proceso de levantamiento de requisitos en los proyectos de portales de la *UCI* y finalmente, se brindará una descripción acerca de la constitución de la población, el proceso de selección de la muestra, el tipo de investigación y los métodos utilizados durante la investigación, con una pequeña explicación en qué consiste cada uno de ellos. Sentando las bases para la definición de la propuesta que será detallada en el próximo capítulo.

### 2.1 Conceptos de Interés

Comenzaremos planteando el concepto de Sitio Web (Web Site), que no es más que un grupo de documentos html relacionados y archivos asociados, guiones y bases de datos que son servidos por un servidor de http en el World Wide Web. Los documentos html en un sitio Web generalmente cubren uno o más temas relacionados y se interconectan a través de hipervínculos. La mayoría de los sitios Web tienen una página de inicio como su punto de arranque, que frecuentemente funciona como un índice para el sitio. Muchas organizaciones grandes, como corporaciones, tendrán uno o más servidores de http dedicados a un solo sitio Web. Sin embargo, un servidor http también puede servir varios sitios Web pequeños, como los poseídos por individuos. Los usuarios necesitan un explorador de Web y una conexión de Internet para acceder a un sitio Web. (SÁNCHEZ, 2001)

Por lo que podemos decir entonces que uno de los tipos de sistemas que encontramos dentro de las clasificaciones para los sistemas Web, son los llamados Portales Web, definiéndose como un Sitio Web que sirve como pasarela a Internet. Un portal es una colección de enlaces, contenido y servicios diseñados para guiar a los usuarios a la información que les interesa para encontrar noticias interesantes, información meteorológica, sitios comerciales, habitaciones de chat y similares. Yahoo!, Excite, MSN.com y Netscape NetCenter son ejemplos de portales. (SÁNCHEZ, 2001)

Los desarrolladores Web necesitan de metodologías específicas mediante las cuales diseñar productos Web usables y accesibles. (HASSAN y MARTÍN, 2004)

Ahora bien, ¿cuándo consideramos que un sistema Web es centrado en el usuario?, ¿qué entendemos por Usabilidad y Accesibilidad de un sitio Web?

Un sistema Web es Centrado en el Usuario si se caracteriza por asumir que todo el proceso de diseño y desarrollo del sitio Web debe estar conducido por el usuario, sus necesidades, características y objetivos. Centrar el diseño en sus usuarios (en oposición a centrarlo en las posibilidades tecnológicas o en nosotros mismos como diseñadores) implica involucrar desde el comienzo a los usuarios en el proceso de desarrollo del sitio; conocer cómo son, qué necesitan, para qué usan el sitio; testar el sitio con los propios usuarios; investigar cómo reaccionan ante el diseño, cómo es su experiencia de uso; e innovar siempre con el objetivo claro de mejorar la experiencia del usuario. (HASSAN, M. Y.; MARTÍN, F. F. J. y IAZZA, G, 2004)

La usabilidad, definición ofrecida por la (ISO-9241-11, 1998), no es más que "el grado de eficacia, eficiencia y satisfacción con la que usuarios específicos pueden lograr objetivos específicos, en contextos de uso específicos".

Un concepto que viene aparejado al de usabilidad es el de accesibilidad. Éste ya no se refiere a la facilidad de uso, sino a la posibilidad de acceso. En concreto a que el diseño, como prerequisite imprescindible para ser usable, posibilite el acceso a todos sus usuarios potenciales, sin excluir a aquellos con limitaciones individuales - discapacidades, dominio del

---

idioma, ... – o limitaciones derivadas del contexto de acceso - software y hardware empleado para acceder, ancho de banda de la conexión empleada, etc.- (HASSAN y MARTÍN, 2003)

Se da la paradoja de que mientras que un diseño usable requiere delimitar a su audiencia potencial con el fin de diseñar para lo concreto, un diseño accesible implica la necesidad de diseñar para la diversidad y heterogeneidad de necesidades de acceso presentadas por esta audiencia específica. Cuando la audiencia para la que se diseña es muy amplia y presenta necesidades de acceso muy diferentes, normalmente se hace necesaria la puesta a disposición de varias versiones del diseño o un diseño adaptable, como son las conocidas "versiones solo texto" o versiones en varios idiomas. (HASSAN, M. Y.; MARTÍN, F. F. J. y IAZZA, G, 2004)

### **2.2 Elementos a Considerar en el Levantamiento de Requisitos**

Existen elementos fundamentales que se deben tener en cuenta a la hora de modelar un portal Web, y uno de ellos es la Arquitectura de Información (AI). Existen diferentes definiciones del término, a continuación plantearemos algunas de ellas.

“Una combinación de la organización de la información del contenido del sitio en categorías y la creación de una interfaz para sostener esas categorías”(RONDA, 2005)

Otra de las definiciones que se plantean es:

“El arte y la ciencia de organizar información para ayudar a las personas a desempeñar eficientemente sus necesidades de información. Ello conlleva investigación, análisis, diseño e implementación”. (GRANOLLERS, 2004)

Definiéndose también como:

1. La combinación de organización, etiquetado y esquemas de navegación dentro de un sistema de navegación.
2. El diseño estructural de un espacio de información que facilite la finalización de tareas y el acceso intuitivo a los contenidos.
3. El arte y la ciencia de estructurar y clasificar los sitios Web y las Intranets para ayudar a las personas a encontrar y gestionar la información.(GRANOLLERS, 2004)

Guiando la investigación basada en la definición planteada anteriormente por Granollers.

La Arquitectura de Información estará cumpliendo sus objetivos cuando un usuario entre por primera vez al sitio y pueda reconocer a quién pertenece el Sitio Web; lo pueda entender de forma rápida y sin esfuerzo y logre encontrar la información ofrecida fácilmente. Para ello existen elementos dentro de la *AI* que no pueden dejar de mencionarse, tales como:

1. Definición de Objetivos del Sitio
2. Definición de Audiencia
3. Definición de Contenidos del Sitio
4. Definición de la Estructura del Sitio
5. Definición de los Sistemas de Navegación
6. Definición del Diseño Visual ( COLECTIVO DE AUTORES, 2004)

Dentro de los términos mencionados que identifican la *AI*, los más importantes a detallar y que constituyen elementos básicos dentro de la misma son los que hacen referencia a: la **información** en sí, el **contexto** en el que ésta es presentada y los **usuarios** que finalmente observarán la misma. Por lo que para la comprensión de estos términos debemos ser capaces de responder a las siguientes preguntas que se plantean para cada uno de ellos.

Para los Usuarios:

1. ¿Cuáles son las motivaciones que llevan a los usuarios a visitar nuestro sitio?
2. ¿Qué quieren los usuarios de este sitio?
3. ¿Qué necesitan de él?
4. ¿Quiénes son?
5. ¿Qué audiencias son más importantes?
6. ¿Cómo navegan?
7. ¿Qué términos utilizan para navegar, buscar y clasificar la información?
8. ¿Cuáles son sus necesidades de información?

Para el contenido:

1. ¿Cómo puedo organizar el contenido que tengo?
2. ¿Qué contenido tiene valor?
3. ¿De qué contenido puedo deshacerme?
4. ¿Cómo pueden emerger las respuestas del contenido que tengo?



5. ¿Cómo debe el contenido ser organizado y etiquetado?

Para el contexto:

1. ¿Quién, dentro de la organización, tiene poder de tomar decisiones?
2. ¿Qué quieren del sitio?
3. ¿Qué factores culturales y/o políticos pueden afectar a la arquitectura?
4. ¿Hay similares experiencias previas?, ¿Acabaron bien? ¿Por qué?
5. ¿Con qué recursos contamos (humanos, económicos, tiempo, tecnológicos)?
6. ¿Cómo la arquitectura será sostenida y mantenida?
7. ¿Están preparados para la AI?(GRANOLLERS, 2004)

### **2.3 Caracterización de la Captura de Requisitos en los Portales Web de la UCI**

Inicialmente la producción de software en la *UCI* se realizaba de manera genérica, no existía una estructura adecuada para el desarrollo según el tipo de sistemas que se necesitaba implementar. Por lo que con el pasar de los años, una vez reunida las condiciones de personal calificado, recursos tecnológicos, entre otros, para el desarrollo de sistemas, cuyo producto final es un portal Web, se crea la Dirección de Portales en la *UCI*, a cargo del desarrollo de este tipo de sistemas.

Inicialmente el personal que conformaba la dirección no era suficiente para emprender los procesos que se debían desarrollar, como la investigación en función de mejorar el proceso de desarrollo y a la vez, ejecutar los proyectos solicitados por los clientes. Por lo que esto trajo consigo que el proceso de levantamiento de requisitos de los proyectos era insuficiente, debido a que no se realizó la adecuación del proceso para este tipo de sistema, constituyendo una limitante que influyó en la calidad de los sistemas resultantes.

Actualmente se realizó una reestructuración en la dirección mencionada y los proyectos que allí se desarrollaban, pasaron a formar parte de los proyectos productivos que se ejecutan en las áreas o facultades que forman parte de la universidad; de este modo se amplía el personal

que desarrolla este tipo de sistemas, permitiendo realizar otras actividades para mejorar el proceso de desarrollo de portales.

En el año 2006, en la *UCI* se desarrollaron un total de 24 sistemas de portales, los que se encontraban distribuidos en cada una de las facultades de la universidad. Todos los proyectos mencionados anteriormente culminaron su proceso de ejecución en el período analizado.

En cada uno de los proyectos se llevó a cabo el levantamiento de requisitos, realizándose como punto de partida posterior a la solicitud realizada por parte del cliente a la *UCI*. Dicho proceso se realizó a través del procedimiento que se explicará a continuación.

La aproximación inicial de un cliente a la universidad partía de solicitar servicios para la realización de un portal Web, a continuación la Dirección de portales asignaba un líder para la ejecución del proyecto en conjunto con un equipo de desarrollo, compuesto fundamentalmente por analistas, arquitectos, diseñadores, programadores y arquitectos de información a fin de explorar sus necesidades desde las perspectivas de diferentes especialistas y poder orientarles en correspondencia.

Seguidamente, los miembros que tenían las funciones orientadas al paso inicial – levantamiento de requisitos – como parte del proceso de desarrollo, se dirigían a la entidad solicitante para entrevistar al cliente y determinar las necesidades precisas para la creación del portal, la clasificación del mismo en portal informativo o portal de gestión, las características y el contenido de información a publicar, el logotipo que representaba a la entidad en caso de tenerlo, y otros elementos de carácter más técnicos como la arquitectura de información necesaria para poder desarrollar el sistema.

Posteriormente a la obtención de las funcionalidades que iba a tener el portal, se pasaba a la modelación del mismo a través de la utilización de algunos artefactos de la metodología *RUP*, la cual se utiliza como norma estándar en la universidad para modelar los sistemas de software.

Luego, siguiendo un cronograma de reuniones establecido con el cliente, se procedía al análisis de la correspondencia entre las características del portal en construcción y los requisitos establecidos por el cliente, y se realizaban las modificaciones pertinentes según las solicitudes planteadas en dichas reuniones.

Los principales problemas que surgieron en ese momento, fueron los siguientes:

- ✚ Manifestaciones de insatisfacciones por parte del cliente al equipo de desarrollo acerca del sistema que se estaba implementando, porque las funcionalidades requeridas no eran las que inicialmente ellos habían planteado.

El análisis de esta situación indicó que la causa radicaba en el hecho de no firmarse previamente un contrato con los acuerdos tomados en las entrevistas realizadas y no haber constancia oficial de la información recuperada ni de las peticiones realizadas por el cliente en diferentes etapas.

- ✚ Utilización de la metodología *RUP*, bajo el fundamento de ser la metodología establecida como estándar en la universidad sin realizar las adecuaciones necesarias para ese tipo de sistemas.

Esta metodología define un proceso genérico para la modelación de sistemas de software, pero en determinadas situaciones, debido a que existen tipos de sistemas con características más específicas, es necesario realizarle adecuaciones, integrar nuevos elementos o no utilizar algunos de los que la metodología propone. Uno de los elementos que ejemplifica esto es la arquitectura de información, donde *RUP* no propone ningún artefacto que facilite su modelación, siendo este un aspecto importante dentro de los sistemas de portales.

Otra de las modificaciones a realizar planteadas por la autora, sería en el Documento de Requerimientos que define *RUP*, puesto que este no aborda la clasificación de requisitos expuesta en el capítulo anterior.

Tomando como base las experiencias obtenidas durante el desarrollo de los primeros sistemas, la Dirección de Portales enfrenta las dos problemáticas anteriores, priorizando la solución de

---

una de ellas – específicamente, la insatisfacción del cliente con el equipo de desarrollo sobre el sistema en implementación – porque era el núcleo inicial para garantizar la ejecución de los proyectos.

La solución asumida por la Dirección de Portales consistió en crear un documento como guía para la entrevista inicial con el cliente, que funciona como contrato, en el que se agrupan una serie de elementos – descritos anteriormente – que permitían que el cliente no tuviera la posibilidad de cambiar los acuerdos tomados en dichas entrevistas o reuniones de trabajo con facilidad y sin que quedara constancia para el equipo de desarrollo, quedaban definidos los requisitos que debía cumplir el portal y se definía la arquitectura de información del mismo así como los elementos de diseño y el flujo de trabajo necesarios para la implementación del portal.

Desde la fecha hasta el momento, dentro de los problemas detectados, aún queda pendiente de solución, el uso de una metodología adecuada para la modelación del proceso. Para ello se propone la utilización de técnicas que favorezcan el proceso, las cuales no se emplearon anteriormente ya fuera por desconocimiento y/o porque estas técnicas no se correspondían con las características de este tipo de sistemas y/o porque el procedimiento planteado en estas técnicas no era el más adecuado.

### **2.4 Población, Muestra, Tipo de Investigación y Métodos Utilizados**

#### **2.4.1 Población**

Para llevar a cabo la investigación primeramente debemos definir cuál será la población. La cuál se definió como los proyectos de portales que se desarrollaron en el período comprendido entre el mes de enero y diciembre del propio año 2006 en la UCI. Estos proyectos ascienden a una cantidad de 24, constituyendo la población de esta investigación.

### 2.4.2 Muestra

Teniendo en cuenta que todos los portales desarrollados en el período antes mencionado, a pesar de haber sido ejecutados en diferentes momentos del período, fueron desarrollados por el mismo equipo reducido de personas. De ese equipo, las personas implicadas en el levantamiento de requisitos del 100 % de los proyectos, argumentaron sus experiencias en la ejecución del proceso, por lo que se tomará como muestra el 100 % de la población de los proyectos.

### 2.4.3 Tipo de Investigación

En el caso de esta investigación, la estrategia a seguir será la *investigación descriptiva*. Este tipo de investigación se lleva a cabo cuando existen los conocimientos necesarios que permitieron el planteamiento de la hipótesis de manera descriptiva, conociendo los elementos necesarios para caracterizar el fenómeno que queremos estudiar.

### 2.4.4 Métodos Utilizados

Posteriormente a la selección de la estrategia de investigación, debemos seleccionar los métodos que se emplearán en la misma, teniendo en cuenta que dichos métodos se clasifican de la siguiente forma: *métodos teóricos* y *métodos empíricos*. Los *métodos teóricos* no son más que aquellos que nos permiten el estudio de las características del objeto de investigación que no se pueden observar directamente. Los otros métodos empleados fueron los *métodos empíricos*, ya que son los que facilitarán el proceso de análisis de la situación en el entorno.

Método *Histórico*, el cuál permitió analizar la trayectoria completa del fenómeno, su condicionamiento a los diferentes períodos analizados, revelando las etapas principales de su desenvolvimiento y las conexiones históricas fundamentales.

---

Método *hipotético – deductivo*, donde partimos de la hipótesis y siguiendo reglas lógicas de deducción se llega a nuevos conocimientos y predicciones, deduciendo y explicando leyes e hipótesis.

Método *sistémico*, que no es más que el estudio del objeto mediante la determinación de sus componentes, así como la relación entre ellos que conforma una realidad como totalidad. Esa relación determina por un lado la estructura y la jerarquía de cada componente en el objeto y por otra parte su dinámica.

Dentro de los métodos empíricos empleamos la *entrevista* y la *encuesta*, los cuáles facilitarán el proceso de caracterización de los proyectos de portales de la universidad. En el caso de la *encuesta*, se realiza a través de la utilización de un cuestionario previamente elaborado, donde se recopila la información necesaria para la caracterización de los proyectos de portales en cuanto al levantamiento de requisitos. Para la confección de la *encuesta* se analizaron los indicadores definidos para cada una de las variables de investigación, los que se utilizaron para confeccionar el cuestionario, analizando como se manifiestan estos indicadores en la producción de portales de la universidad y teniendo en cuenta el criterio de las personas que realizan el proceso, permitiendo medir los conceptos que se evalúan a través de preguntas elaboradas lo más concreta posible. Luego se realizó un pilotaje para garantizar que los encuestados comprenden el cuestionario confeccionado y finalmente se aplica el mismo para su posterior análisis.

Para la *entrevista* se planifica una conversación entre el investigador y el entrevistado, obteniendo parte de la información necesaria, constituyendo su uso un medio para el conocimiento cualitativo de los fenómenos. Para la confección de la *entrevista*, primeramente se analizaron las características del entrevistado y su posición o conocimientos sobre el tema a tratar. Se confeccionaron una serie de preguntas que sirven de guía y que permitieran complementar el proceso de obtención de información necesaria para poder realizar la caracterización de los proyectos de portales de la universidad, en cuanto a la forma en que se realizó el levantamiento de requisitos en cada uno de ellos.

Se muestra a continuación la encuesta aplicada a miembros de los equipos de desarrollo y el guión utilizado para la entrevista con los líderes de proyectos y los especialistas en información.

1- ¿Cuál de las siguientes actividades Ud realiza durante el levantamiento de requisitos de su proyecto? Marque cuál(es) de ellas.	
<input type="checkbox"/> Obtención de requerimiento	<input type="checkbox"/> Análisis de requerimiento
<input type="checkbox"/> Especificación de requerimiento	<input type="checkbox"/> Validación de requerimiento
<input type="checkbox"/> Administración de requerimiento	<input type="checkbox"/> Ninguna
2- ¿Conoce algunas de las siguientes técnicas para el Análisis de requisitos? Marque cuál(es) de ellas.	
<input type="checkbox"/> Introspección	<input type="checkbox"/> Entrevistas
<input type="checkbox"/> Lluvia de ideas	<input type="checkbox"/> Cuestionarios
<input type="checkbox"/> Análisis de protocolos	<input type="checkbox"/> Observación
<input type="checkbox"/> Análisis del discurso	<input type="checkbox"/> Elaboración de prototipos
3- ¿Aplica Ud las técnicas mencionadas anteriormente?	
<input type="checkbox"/> Sí	<input type="checkbox"/> No
4- ¿Si conoce alguna otra técnica a aplicar en alguna de las actividades mencionadas en la pregunta 1, plantee cuáles son?	
5- ¿Qué aspectos de los que a continuación se mencionan, debe tener en cuenta para el levantamiento de requisitos?	
<input type="checkbox"/> Arquitectura de Información	<input type="checkbox"/> Diseño Gráfico
<input type="checkbox"/> Flujo de trabajo	<input type="checkbox"/> Mapa de navegación
<input type="checkbox"/> Otros. Cuáles:	
6- ¿Qué personas (Roles) Ud considera que debe participar en el levantamiento de requisitos? Mencione cuáles.	
7- ¿Cómo Ud realiza la trazabilidad de los requisitos de su proyecto?	

<input type="checkbox"/> Matriz de trazabilidad	<input type="checkbox"/> Auditorias
<input type="checkbox"/> Prototipos	<input type="checkbox"/> Otras. Cuáles:

**Figura 2.1: Encuesta Aplicada**

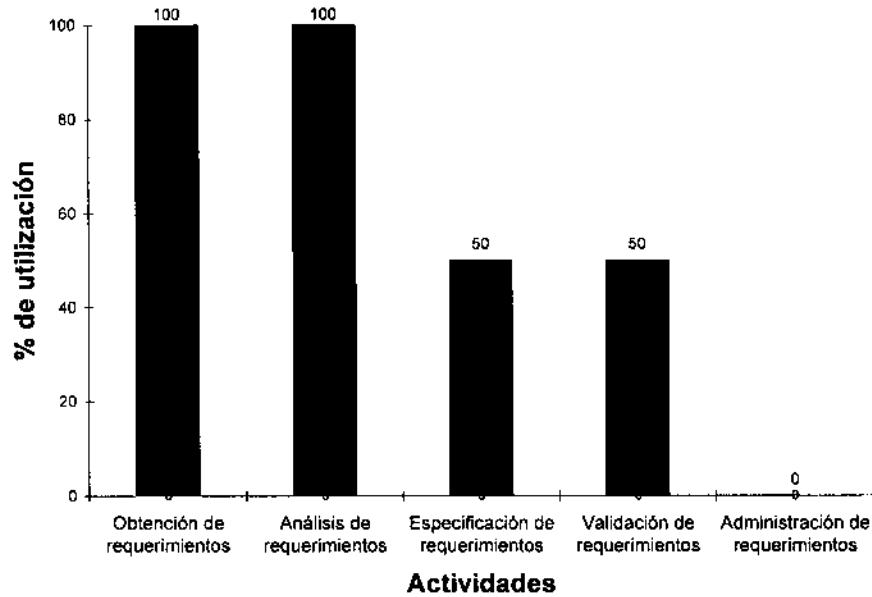
1. ¿Cuál es su nombre?, ¿de qué es graduado?
2. ¿Ha formado Ud. parte de un equipo de desarrollo para la ejecución de un sistema de portales?
3. ¿Qué rol ha desempeñado dentro del proyecto?
4. ¿Ha formado parte del equipo que realizó el levantamiento de requisitos del mismo?
5. ¿Qué actividades se realizaron durante el proceso?
6. Describa como se realizó el mismo.
7. ¿Tuvo éxito el proyecto o fracasó?
8. ¿Qué problemas Ud. considera que influyeron en el fracaso del proyecto?, ¿uno de los problemas pudo haber sido una mala definición de los requerimientos del sistema?
9. ¿Qué aspectos negativo Ud. cree que tiene la forma de ejecutar el proceso de levantamiento de requisitos para los proyectos de portales?
10. ¿Qué aspectos característicos de ese tipo de sistema Ud. cree que se deben tener en cuenta a la hora de realizar le proceso de levantamiento de requisitos?
11. ¿Qué metodología utilizan para modelar el sistema?, ¿por qué?, ¿reúne todos los elementos necesarios la misma?, ¿cuáles faltan y que no pueden ser modelados con la metodología empleada?

**Figura 2.2: Guión de la Entrevista**



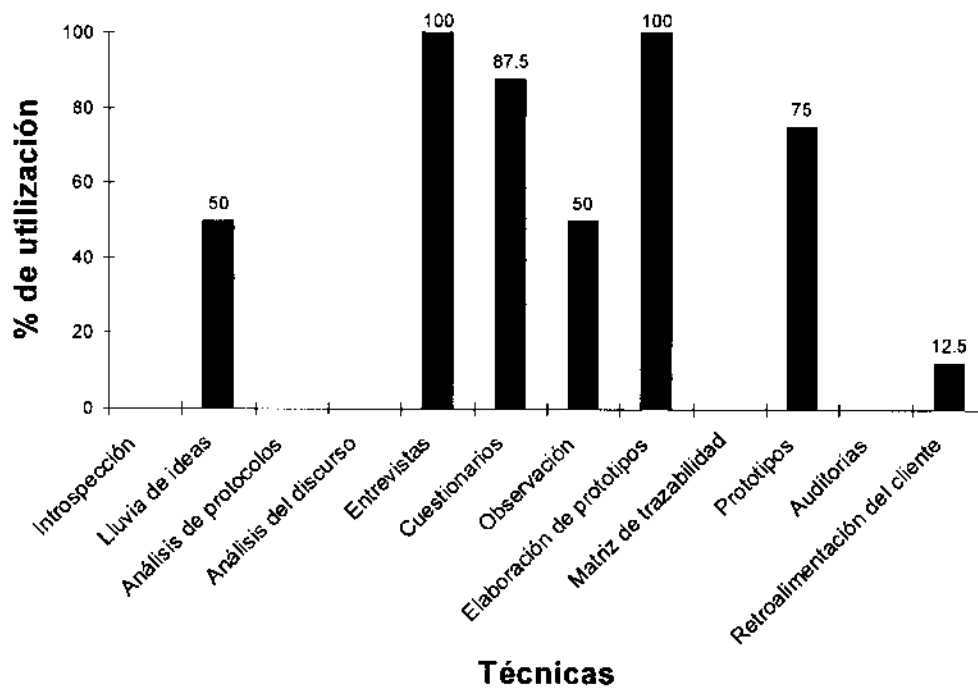
## 2.5 Análisis de Resultados

Se realizó la entrevista a un total de 8 personas y la encuesta se aplicó a un total de 12 personas, constituyendo un 100 %.



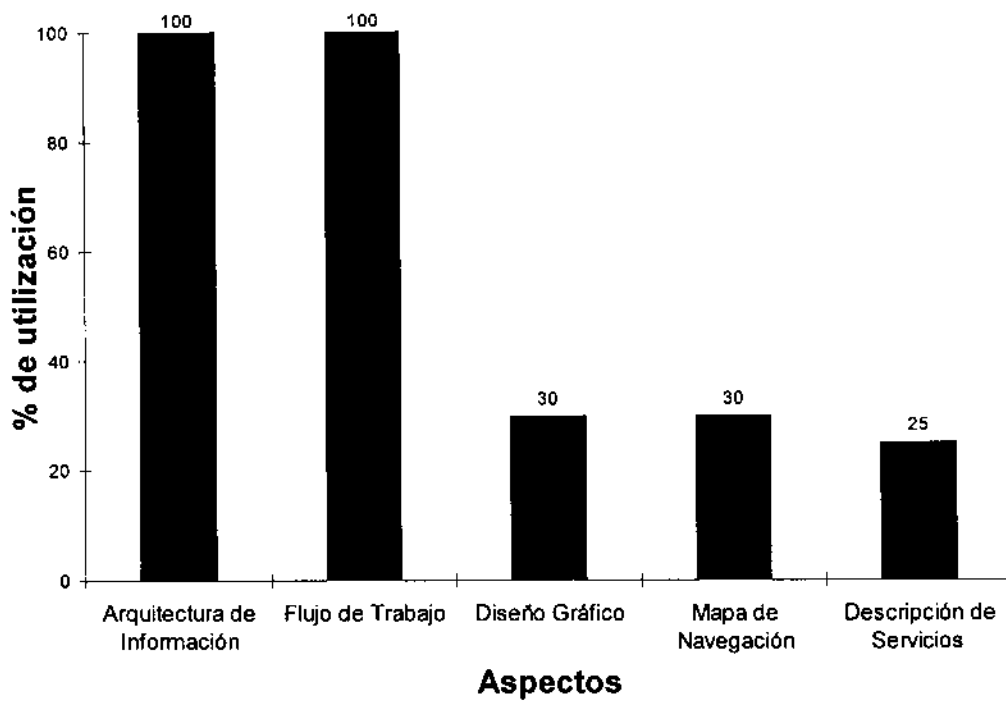
**Figura 2.3: Actividades Realizadas**

Como se observa en la figura anterior, el 100 % de los encuestados realizan solo 2 actividades de las que se plantearon en la encuesta (Obtención de Requerimiento y Análisis de Requerimiento) para el levantamiento de requisitos de los proyectos en los que ellos participaron. Las otras 2 actividades planteadas, Especificación y Validación de Requerimientos, solo el 50 % las realiza.



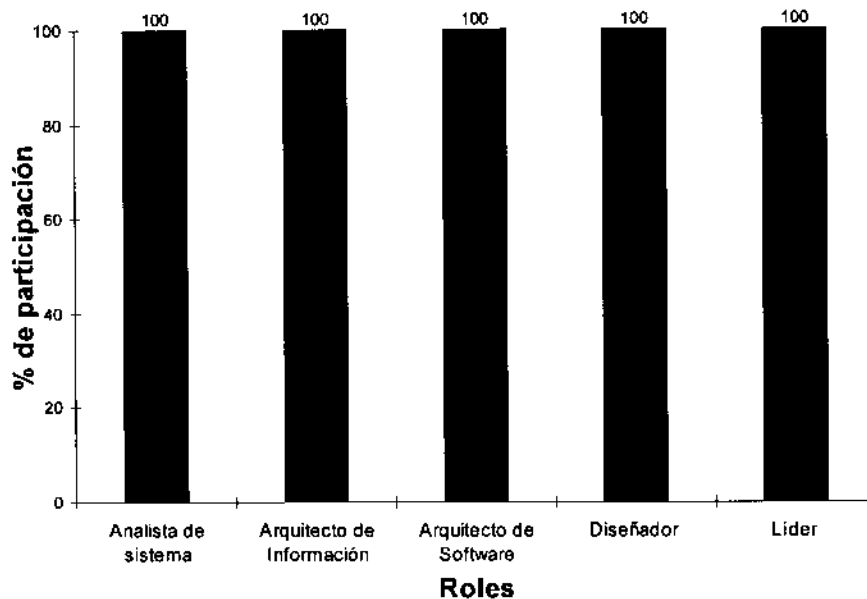
**Figura 2.4: Técnicas Utilizadas**

En el caso de las técnicas que se relacionaron en la pregunta 2, el 100 % utilizó las técnicas de Entrevista y Elaboración de prototipos, un 87.5 % utilizó la técnica de Cuestionario y un 50 % de los encuestados empleó las técnicas de Lluvia de Ideas o Brainstorming y la Observación. En el caso de las técnicas relacionadas para la validación de los requerimientos del sistema, solo se plantea la utilización de Prototipos por un 75 % de las personas encuestadas. Un 12.5 % plantea que se debe realizar una Retroalimentación con los clientes del sistema.



**Figura 2.5: Aspectos a Considerar**

En el caso de los elementos que se deben tener en cuenta durante el levantamiento de requisitos para ese tipo de proyecto, el 100 % de los encuestados manifiesta que se debe incluir la Arquitectura de Información y los Flujos de trabajo. Para el caso del Diseño Gráfico y el Mapa de Navegación, solo consideran si inclusión un 30 % de los encuestados. Agregando algunos (25 %) que se debe incluir además una descripción de servicios que prestará el portal.



**Figura 2.6: Roles Implicados**

En el caso de los roles que deben participar durante el proceso de levantamiento de requisitos del sistema, el 100 % considera que deben ser los que a continuación se relacionan: Analista de sistema, Arquitecto de Información, Arquitecto de Software, Diseñador y el Líder del proyecto.

Con los elementos presentados anteriormente, se puede arribar al planteamiento de los siguientes problemas que influyen en la calidad final de los requerimientos que se definen del proceso de levantamiento de requisitos para los sistemas de portales:

- ✚ No se realizan las actividades necesarias para el tratamiento de los requisitos, quedando por efectuar una de las actividades importante para el proceso, la validación de los requerimientos.
- ✚ La utilización de una sola técnica de las mencionadas para la validación de los requisitos – la técnica de prototipos – pudiendo emplearse otras técnicas que permitan realizar una mejor validación de las funcionalidades que finalmente tendrá el portal.

- ↓ No existe la vía adecuada para modelar los aspectos de la arquitectura de información relacionados en el capítulo.
- ↓ La ausencia de un estudio para la adecuación de la metodología *RUP*, que agrupe o modele mejor los sistemas de portales o la utilización de otra metodología para modelar esos sistemas.
- ↓ La inclusión de un rol para la revisión de la calidad de la especificación de los requerimientos, que trabaje en conjunto con los analistas y los demás roles que intervienen en el proceso.

Por lo que se propone en el próximo capítulo, la explicación detallada de un procedimiento que mejore los problemas anteriores y que garantice la determinación lo más correcta posible de los requisitos de un sistema de portales.

### **Conclusiones**

Con este capítulo concluimos que con la utilización de los métodos se obtuvo una caracterización de los proyectos de portales en cuanto al levantamiento de requisitos; se identificaron los elementos que afectan ese proceso en dichos proyectos, lo cual facilitará la mejora del proceso de captura de los requisitos.

### **Capítulo 3: Descripción del Procedimiento**

#### **Introducción**

Partiendo de los resultados obtenidos en la encuesta aplicada, analizados en el capítulo anterior, la propuesta que se describirá en este capítulo se basará en la mejora del proceso de levantamiento de requisitos para los proyectos de portales de la universidad. Se centrará en la definición de las tareas que se deben ejecutar en cada una de las actividades definidas y además, se propondrán algunas técnicas para llevar a cabo dichas tareas y se indicarán además, las personas involucradas en el proceso en cuestión. Finalmente se realizará un análisis de las posibles herramientas que se podrán emplear para el proceso de gestión de los requerimientos y de las posibles metodologías que se pueden emplear durante el proceso.

#### **3.1 Premisas a Cumplir**

Primeramente se planteará que la mejora en el proceso estará enfocada al planteamiento de una serie de actividades, detalladas a través de la ejecución de tareas, que mejoren el proceso de levantamiento de requisitos. La propuesta de utilización de técnicas que también permiten garantizar la calidad del proceso así como la confección de una plantilla con los elementos necesarios para la modelación de la arquitectura de información del portal que se desee implementar.

A continuación se plantean algunas premisas que no podemos pasar por alto a la hora de definir el procedimiento para la ejecución del levantamiento de requisitos y que permitirán guiar el proceso de manera eficiente. Estos aspectos a los que se hace referencia son:

- Garantizar la definición de actividades concretas y objetivamente realizables durante el levantamiento de requisitos del proyecto.
- Definir claramente las tareas a llevar a cabo en cada una de las actividades definidas en el punto anterior.

- 
- Brindar una descripción detallada y concisa de cada una de las tareas mencionadas anteriormente y especificar con claridad las personas (roles) involucrados en cada una de las tareas así como los entregables que se deben obtener como conclusión de las mismas.
  - Garantizar el seguimiento de los requisitos a través de alguna de las técnicas existentes, garantizando que se de cumplimiento a cada uno de ellos.
  - Obtener la documentación necesaria que detalle el proceso de levantamiento de requisitos de un portal, quedando definidos de forma clara y precisa las funcionalidades que tendrá el portal.

Basado en los aspectos anteriores se detallará como se debe llevar a cabo el proceso de levantamiento de requisitos en los proyectos de portales.

### **3.2 Descripción de la Propuesta**

Para la definición del procedimiento brindaremos primeramente de manera general. las actividades a ejecutar y las tareas para cada una de ellas, las que se describirán posteriormente con un mayor nivel de especificidad. En esa descripción se detallará poco más en que se debe realizar en cada una de ellas, las personas encargadas de realizarlas y los entregable que se deben, así como las técnicas que se proponen para le ejecución de las mismas.

#### Actividad 1: Captura de Requisitos

Tareas:

1. Análisis del entorno.
2. Definir usuarios finales.
3. Definir contenidos del portal.
4. Definir funcionalidades del portal.
5. Recolectar y clasificar los requerimientos.

#### Actividad 2: Especificación de Requisitos

Tareas:

1. Encontrar dependencias, etiquetar y priorizar los requerimientos.

2. Especificar los requerimientos según su clasificación<sup>2</sup>.
3. Utilización de una herramienta para la gestión de requerimientos.

### Actividad 3: Validación de Requisitos

#### Tareas:

1. Verificar la calidad<sup>3</sup> en la especificación de los requerimientos.
2. Realizar la trazabilidad de los requerimientos.
3. Comprobar ajuste a peticiones del cliente.
4. Resolver conflictos de cambios en los requerimientos.

A continuación se especificará cada una de las actividades mencionadas:

#### **3.2.1 Actividad 1: Captura de Requisitos**

##### Descripción:

Es la actividad mediante la cual el analista extrae, de cualquier fuente de información proporcionada por el cliente, las necesidades de éste. Todos los elementos de información deben ser almacenados e identificados. Pueden provenir de conversaciones con el cliente o estar recogidos en ficheros de datos, gráficos, normativas legales, etc.(Díez, 2001)

##### Objetivos:

1. Identificar las necesidades del cliente que conllevan a la creación de un portal como vía de solución a los problemas.
2. Identificar los objetivos del sistema, el alcance del mismo y los usuarios finales.
3. Generar los documentos necesarios para el almacenamiento y control de los elementos identificados durante el proceso.

##### Precondiciones:

Ante una solicitud de realizar un nuevo proyecto que requiera la creación de un portal, la UCI asigna un Líder de proyecto que realice las investigaciones iniciales, para determinar si se llevará a cabo el proyecto, haciendo para ello un estudio de viabilidad. En caso de ser factible, se procede a constituir el equipo de desarrollo y posteriormente se procede a realizar el

---

<sup>2</sup> La clasificación de los requisitos se detalló en el Capítulo 1. Dicha clasificación se abordará además, en la especificación de las actividades y tareas que se desarrollará posteriormente.

<sup>3</sup> La calidad está referida a los aspectos de validez, consistencia e integridad de los requerimientos.



---

levantamiento de los requisitos del sistema. Para llevar a cabo ese flujo se crea un Cronograma de Captura de Requisitos. (Ver Anexo 1)

Tareas:

1. **Análisis del entorno:** Se debe realizar un estudio detallado sobre el entorno en el que funcionará el sistema, las condiciones bajo las cuáles se ejecutará el mismo, el alcance, los objetivos y lograr un entendimiento y/o familiarización por parte del equipo de desarrollo en cuanto a estos aspectos.
2. **Definir usuarios finales:** Identificar los usuarios potenciales del portal, determinando sus características principales con el objetivo de presentarle la información lo más acorde posible a sus necesidades.
3. **Definir contenidos del portal:** Determinar los contenidos de información que desea mostrar el cliente y agruparlos según criterios específicos, que se deben tener en cuenta para la visualización de la información. Refiérase a posicionamiento de la información, accesibilidad y usabilidad de la misma, etc.
4. **Definir funcionalidades del portal:** Identificar las prestaciones que tendrá el portal en cuestión. La definición de los objetivos puede contribuir con la determinación de las funcionalidades del mismo.
5. **Recolectar y clasificar los requerimientos:** Lograr una especificación detallada de los requerimientos del portal, teniendo en cuenta la clasificación de los requisitos que se plantea a continuación:
  - a) Requisitos de datos.
  - b) Requisitos de interfaz.
  - c) Requisitos navegacionales
  - d) Requisitos de personalización.
  - e) Requisitos transaccionales o funcionales internos.
  - f) Requisitos no funcionales.

Participantes:

Se debe conformar un equipo multidisciplinario para la ejecución del proceso compuesto básicamente por los siguientes roles:

1. Líder del proyecto.
2. Analista principal.

3. Diseñador.
4. Comunicador social.
5. Especialista en información.
6. Psicólogo organizacional.

Entregables:

1. Proyecto técnico. (Ver Anexo 2)
2. Plan de Requisitos. (Ver Anexo 3)
3. Glosario de Términos. (Ver Anexo 4)

Técnicas:

Existen un conjunto de técnicas que se emplean para facilitar la obtención de la información deseada. A continuación describiremos las más utilizadas en la captura de los requisitos de un sistema.

Entrevista

Las entrevistas son la técnica de elicitación más utilizada, y de hecho son prácticamente inevitables en cualquier desarrollo ya que son una de las formas de comunicación más naturales entre personas. En las entrevistas se pueden identificar tres fases: Preparación, Realización y Análisis. (Durán y Bernárdez, 2000). Dentro de la actividad de Preparación de la entrevista se realizan las siguientes fases: Estudiar el dominio del problema, Seleccionar a las personas a las que se va a entrevistar, Determinar el objetivo y el contenido de las entrevistas y Planificar la entrevista. Como parte de la Realización de la entrevista encontramos las siguientes fases: Apertura, Desarrollo y Terminación. Para el Análisis de la entrevista, una vez realizada, es necesario leer las notas tomadas, pasarlas a limpio, reorganizar la información, contrastarla con otras entrevistas o fuentes de información, etc. Una vez elaborada la información, se puede enviar al entrevistado para confirmar los contenidos. También es importante evaluar la propia entrevista para determinar los aspectos mejorables.

Desarrollo Conjunto de Aplicaciones (*JAD*)

Es una alternativa a las entrevistas individuales que se desarrolla a lo largo de un conjunto de reuniones en grupo durante un período de 2 a 4 días. En estas reuniones se ayuda a los clientes

---

y usuarios a formular problemas y explorar posibles soluciones, involucrándolos y haciéndolos sentirse partícipes del desarrollo. Esta técnica se base en cuatro principios: dinámica de grupo, el uso de ayudas visuales para mejorar la comunicación (diagramas, transparencias, multimedia, herramientas CASE, etc.), mantener un proceso organizado y racional y una filosofía de documentación WYSIWYG “What You See Is What You Get”, lo que se ve es lo que se obtiene), por la que durante las reuniones se trabaja directamente sobre los documentos a generar. (Durán y Bernárdez, 2000). Existen seis roles o clases de participantes que forman parte del *JAD*, los cuales son: Jefe del *JAD*, Analista, Patrocinador ejecutivo, Representantes de los usuarios, Representantes de sistemas de información y Especialistas. Dentro de las fases que se ejecutan podemos mencionar: Adaptación, Celebración de las sesiones *JAD* y la Conclusión.

### Tormenta de ideas

El Brainstorming o tormenta de ideas es una técnica de reuniones en grupo cuyo objetivo es la generación de ideas en un ambiente libre de críticas o juicios. Las sesiones de Brainstorming suelen estar formadas por un número de cuatro a diez participantes, uno de los cuales es el jefe de la sesión, encargado más de comenzar la sesión que de controlarla. Como técnica de elicitación de requisitos, 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. Está compuesta por tres fases: Preparación, Generación y Consolidación.(Durán y Bernárdez, 2000)

### Cuestionarios

Es una técnica la cual constituye una manera de obtener información que permite a los desarrolladores del sistema recopilar opiniones, posturas, conductas y características de los diversos usuarios que son encuestados y que se encuentran involucrados en la operación del sistema actual o en la implantación de un sistema nuevo. Los cuestionarios son eficaces si la gente de la organización se encuentra dispersa o si esta involucrada mucha gente en el proyecto de sistema. Al igual que la entrevista, un cuestionario puede redactarse usando preguntas abiertas o preguntas cerradas, además es importante usar un vocabulario que refleje el lenguaje de los involucrados en la organización. El uso de preguntas cerradas en los

---

cuestionarios permite a los analistas medir los resultados mediante gráficas. Esta técnica suele combinarse con las entrevistas para mejorar la obtención de información del sistema.(Medina, 2004)

### Observación

Es una técnica en donde el desarrollador se sumerge en el ambiente de trabajo de los usuarios, para observar el trabajo diario que éstos realizan anotando los aspectos importantes, observando factores sociales y organizacionales que afecten el sistema. Existen dos enfoques de esta técnica y son la etnografía y grabar lo observado.(Medina, 2004)

### Elaboración de prototipos

Se procede a la generación de un “casarón” que presenta la forma de interfaz y simula algunas funcionalidades. Este método es útil porque permite observar las reacciones del usuario y sus habilidades, se pueden incorporar sugerencias e innovaciones antes de entregar una primera versión completa. El sistema resultante será más cercano al usuario. En la actualidad hay una gran gama de herramientas de software que permiten desarrollar un prototipo rápido; puede hacerse inclusive desde un procesador de palabras que convierta su aplicación a una página html o utilizando herramientas visuales que ya contienen todo tipo de ventanas, botones y elementos variados. El método de prototipo rápido presenta algunas desventajas como son: el usuario piensa que ya es la versión definitiva y el computólogo lo toma para “parcharla”. Suele suceder que el usuario crea que resultó fácil el desarrollo y pida que los tiempos de entrega se acorten.(Sumano, 1999)

### **3.2.2 Actividad 2: Especificación de Requisitos**

#### Descripción:

En esta actividad se deben definir los requisitos y negociarlos con el cliente, para lograr obtener el documento de Especificación de Requerimientos, aplicándole técnicas de revisión al mismo, garantizando que se obtenga una especificación de requisitos lo más certera posible, evitando al máximo los problemas que puede traer consigo una mala definición de requisitos del portal.

#### Objetivos:

1. Obtener el documento de Especificación de Requisitos.

### Precondiciones:

Previamente se tienen que haber identificado los requisitos del sistema, para poder proceder a la documentación de los mismos a través de esta actividad.

### Tareas:

1. Encontrar dependencias, etiquetar y priorizar los requerimientos: Se debe analizar la definición inicial o informal de los requisitos para poder establecer las dependencias que existen entre ellos; asignar un identificador único que permita rastrearlo en todo momento y establecer niveles de prioridad en función de las peticiones del cliente.
2. Especificar los requerimientos según su clasificación: Elaborar una descripción formal de los requisitos, clasificándolos según los tipos de requisitos vistos con anterioridad; realizarle revisiones al documento en conjunto con el equipo de desarrollo y el cliente, en caso de ser necesario, para obtener una especificación formal.
3. Utilización de una herramienta para la especificación: Se podrá utilizar una herramienta para llevar la gestión de los requisitos. En un próximo epígrafe se propondrá la utilización de alguna de ellas.

### Participantes:

1. Especificador de requisitos.
2. Revisor técnico.

### Entregables:

1. Documento de Especificación de Requisitos. (Ver Anexo 5)

### Técnicas:

#### Lenguaje Natural

El lenguaje natural es utilizado en la mayoría de las veces para escribir los requerimientos del sistema con el fin de que los usuarios puedan comprenderlos sin necesidad de poseer conocimientos técnicos de desarrollo y comprobar que sus necesidades están expresadas en el documento de requerimientos. Por otro lado, aunque el lenguaje natural sea flexible y sencillo puede contener problemas de interpretación. Algunos problemas que se pueden encontrar en la descripción de los requerimientos utilizando el lenguaje natural son:

- ✚ Falta de claridad. Es difícil utilizar el lenguaje de forma precisa y no ambigua.

- 
- ✚ Confusión de Requerimientos. Se dificulta la distinción de los requerimientos funcionales de los no funcionales, las metas del sistema y la información para el diseño.
  - ✚ Conjunción de Requerimientos. Varios requerimientos diferentes se expresan de forma conjunta como un único requerimiento.
  - ✚ La comprensión del lenguaje natural radica en que los lectores y redactores de la especificación utilicen la misma palabra para el concepto.
  - ✚ La especificación del lenguaje natural es demasiado flexible. Se puede decir lo mismo de formas completamente diferentes.
  - ✚ No existe una forma fácil de modularizar los requerimientos. El lenguaje natural es difícil exhibir todos los requerimientos relacionados.

Existen dos modificaciones del lenguaje natural que restringen la manera de redactar los requerimientos y son lenguaje natural estructurado y lenguaje de descripción de diseño.(Medina, 2004)

### Notaciones Gráficas

Es una manera ilustrativa de generar la especificación de los requerimientos mediante un conjunto de modelos. Los modelos utilizan un lenguaje gráfico, complementado con anotaciones textuales que describen el problema a resolver y el sistema a desarrollar. En muchos casos, los modelos de requerimientos del sistema son más entendibles que la descripción en lenguaje natural, debido a las representaciones gráficas. Son considerados vías directas entre los procesos de requerimientos y diseño.(Medina, 2004). Los modelos se utilizan en la Ingeniería de Requerimientos para desarrollar una comprensión del sistema existente a reemplazar, mejorar o para especificar el sistema requerido. Un modelo es una abstracción del sistema que se va a desarrollar, más que una representación alternativa de ese sistema, por lo que una abstracción omite detalles simplificando y seleccionando las características más sobresalientes de la entidad representada. En contraste, una representación de un sistema debe mantener toda la información. Por lo que existen diferentes tipos de modelos que se basan en varios enfoques de abstracción y representan al sistema desde diferentes perspectivas. Estas perspectivas son:

- ✚ Perspectiva externa:

- Modelos de contexto.
- ✚ Perspectiva de comportamiento:
  - Modelos de flujo de datos.
  - Modelos de máquinas de estado.
- ✚ Perspectiva estructural:
  - Modelo Entidad – Relación.
  - Modelo de datos semánticos
- ✚ Perspectiva Orientada a Objeto:
  - Modelos de Casos de Uso.
  - Modelos de clases jerárquicas (herencia).
  - Modelado de comportamiento de objetos (*UML*).
- ✚ *CASE*
- ✚ Prototipo

### Casos de Uso

Es como técnica de definición de requisitos como más ampliamente han sido aceptados los casos de uso. Actualmente se ha propuesto como técnica básica del proceso *RUP*. Sin embargo, son varios los autores que defienden que pueden resultar ambiguos a la hora de definir los requisitos por lo que hay propuestas que los acompañan de descripciones basadas en plantillas o de diccionarios de datos que eliminen su ambigüedad.

### 3.2.3 Actividad 3: Validación de Requisitos

Descripción:

Una vez que han sido definidos los requisitos del portal, se debe realizar un proceso de validación de los mismos, para garantizar que la definición planteada de cada uno de ellos es la más adecuada. A través de la utilización de algunas técnicas se propone hacer revisiones a los modelos obtenidos que verifiquen el cumplimiento de cada uno de los requisitos especificados en etapas anteriores.

Objetivos:

1. Validar la especificación de los requisitos.

Precondiciones:

Se debió realizar la especificación de los requisitos del sistema.

Tareas:

1. Verificar la calidad en la especificación de los requerimientos: A través de la aplicación de técnicas realizar revisiones constantes a la definición de los requisitos, para garantizar que estos estén escritos de la forma correcta, sin ambigüedades, verificables, consistentes, entendible por el cliente, trazables en todos los sentidos, organizados y de manera concisa.
2. Realizar la trazabilidad de los requerimientos: Llevar la matriz de trazabilidad para comprobar que los requerimientos obtenidos inicialmente se transformaron en alguna de las funcionalidades del portal. Se puede evaluar a través de la revisión del cumplimiento de los objetivos del portal.
3. Comprobar ajuste a peticiones del cliente: Se tiene que realizar una revisión de los requerimientos definidos con el cliente para comprobar que esos son los que él realmente desea que sean implementados en el sistema.
4. Resolver conflictos de cambios en los requerimientos: Llevar un documento de Control de cambios con las modificaciones realizadas a los requerimientos en caso de ser necesarios. Se deberá registrar la petición de cambio en la definición de los requerimientos y analizar en el equipo de desarrollo si es factible o no la realización del cambio. En caso de ser aceptado, efectuar el cambio en el documento de Especificación de los requisitos.

Participantes:

1. Líder del proyecto.
2. Analista principal.
3. Especificador de requisitos.
4. Revisor técnico.
5. Cliente.

Entregables:

1. Documento de Seguimiento (Control de cambios). (Ver Anexo 6)
2. Solicitud de Cambio. (Ver Anexo 7)

Técnicas:

Construcción de Prototipo



En este enfoque de validación se presenta un modelo ejecutable del sistema a los usuarios y clientes, con el fin de que puedan realizar pruebas y comprobar si cumplen con sus necesidades reales. En la figura 3.1 se muestra un algoritmo del desarrollo de un prototipo evolutivo. Se basa en la idea de construir una implementación inicial, exponerla a los comentarios de los clientes y refinarla mediante la repetición de las etapas hasta que se haya desarrollado un sistema adecuado.(Medina, 2004)

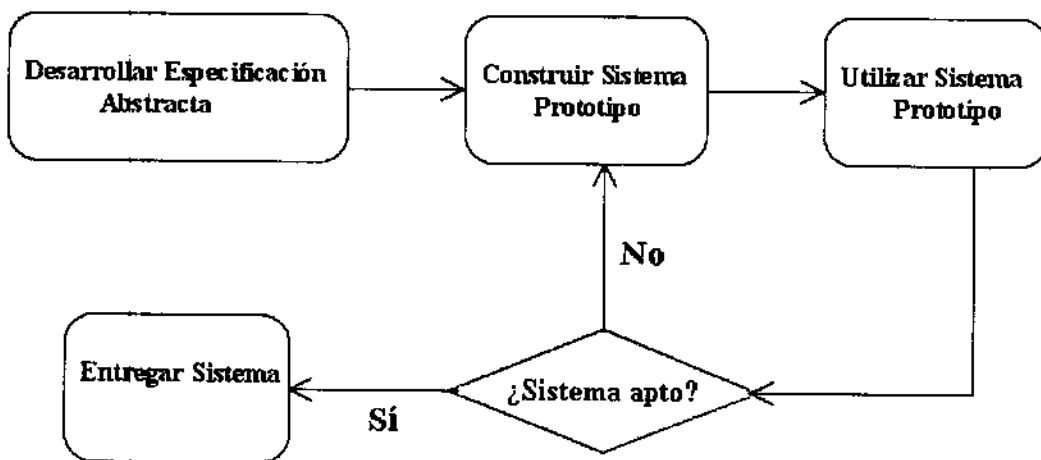


Figura 3.1: Algoritmo Prototipo Evolutivo

Generación de Casos de Prueba

Los requerimientos deberán ser probados. Si las pruebas pueden ser consideradas como parte del proceso de validación, esto frecuentemente permite describir los problemas en los requerimientos.(Medina 2004)

Matriz de trazabilidad

Esta técnica consiste en marcar los objetivos del sistema y chequearlos contra los requisitos del mismo. Es necesario ir viendo qué objetivos cubre cada requisito, de esta forma se podrán detectar inconsistencias u objetivos no cubiertos.(Escalona y Koch, 2002)

Se realiza el análisis de cumplimiento de requisitos del sistema contra los objetivos definidos inicialmente, para garantizar que todas las funcionalidades cumplan con las peticiones del cliente.

### “Reviews o Walk-throughs”

Esta técnica consiste en la lectura y corrección de la completa documentación o modelado de la definición de requisitos. Con ello solamente se puede validar la correcta interpretación de la información transmitida. Más difícil es verificar consistencia de la documentación o información faltante.(Escalona y Koch, 2002)

### Prototipos

Algunas propuestas se basan en obtener de la definición de requisitos prototipos que, sin tener la totalidad de la funcionalidad del sistema, permitan al usuario hacerse una idea de la estructura de la interfaz del sistema con el usuario. Esta técnica tiene el problema de que el usuario debe entender que lo que está viendo es un prototipo y no el sistema final.(Escalona y Koch, 2002)

## **3.3 Propuesta de Herramientas y Metodologías**

### **3.3.1 Herramientas**

La mayoría de las herramientas de gestión de requisitos realizan principalmente las mismas funciones. Estas herramientas permiten a los desarrolladores del sistema, importar grandes documentos de una variedad de formatos estándar de procesadores de palabras. Pueden ser generados una variedad de vistas de documentos utilizando tanto los atributos como las relaciones, generalmente vistas específicas de trazabilidad tales como matrices de trazabilidad. De la misma manera, plantillas de documentos pueden ser configuradas para crear nuevos documentos compuestos.(Mcdonald, 2005)

Las herramientas de gestión de requisitos deben cumplir ciertas funciones entre las cuales se puede mencionar: identificación de requisitos individuales; asignación a un destino y clasificación de requisitos; grupo de requerimientos (recopilación), revisión, identificación/punto de arranque; proveer un interfaz de datos básicos.

---

Del conjunto amplio de herramientas existentes que se utilizan para la gestión de requisitos, solo algunas de ellas – las que a continuación se relacionan – cubren la mayor parte de las funcionalidades que deben cumplir, por lo que solo se mostrarán algunas características de ellas de manera general y se detallará más la IRqA, que aunque no posee versión para software libre, es una de las que mejor cubre el ciclo de vida de los requerimientos.

1. “IBM Rational RequisitePro”
2. IRqA
3. “CaliberRM”
4. DOORS ERS

### “IBM Rational RequisitePro”

Es una herramienta centrada en documentos, que almacena los requisitos asociándolos a documentos (aunque también permite guardarlos directamente en la base de datos), mientras que las otras herramientas están orientadas a requisitos. Auxilia especialmente en el control de cambio de requisitos, con trazabilidad para especificaciones de software y pruebas. Está muy unido a MS Word ya que es asociado a Microsoft Development. La herramienta permite el uso de “Oracle” sobre “Unix” o Windows como “back-end database” y también soporta “SQL Server” sobre Windows.

### IRqA

Según sus siglas, se denomina “Integral Requisite Analyzer”. Es una herramienta de ingeniería de requisitos especialmente diseñada para soportar el proceso completo de ingeniería de requisitos. En IRqA el ciclo de especificación completo incluye la captura de requisitos, análisis, especificación de sistema, validación y la organización de requisitos es soportada por modelos estándares.

### “CaliberRM”

Es para sistemas grandes y complejos y proporciona una base de datos de requisitos con trazabilidad. La compañía ve a los requisitos como parte del proceso de gestión de la calidad del software, el cual es considerado también, las pruebas “testing” y el trazado de defectos

---

“defect tracking”. “Caliber” está basado en Internet y maneja referencia de documentos, responsabilidad de usuario, trazabilidad, prioridad y estado entre otras características.

### DOORS

Su nombre según las siglas es “Dynamic Object Oriented Requirements System”. A diferencia del resto de las herramientas, considera los requisitos como objetos y los documentos como módulos. Tiene una orientación basada en objetos, frente a “RequisitePro” y “Caliber-RM”, que manejan solamente requisitos y sus atributos. Es una herramienta para organizaciones grandes que necesitan controlar complejos conjuntos de usuarios y requisitos de sistemas con una completa trazabilidad. Proporciona buena visualización de tales documentos como jerárquicas, y su lenguaje de extensión permite una gran variedad de soporte de herramientas a ser construidas.

De las herramientas mencionadas anteriormente, según un estudio realizado por (McDonald 2005), se exponen determinadas características que permiten corroborar que dichas herramientas reúnen las condiciones para ser utilizadas en la gestión de los requisitos y en el caso de IRqA, es una de las que sugerimos para el proceso de gestión de requisitos por el manejo que permite de los mismos; entre otras características que se detallarán a continuación.

1. Permite añadir requisitos de otras herramientas a través de APIs de la herramienta que posibilita extender las funciones y acceder a información de otras herramientas (COM, Java).
2. Posibilita la asociación de requisitos con archivos externos a través de archivos guardados en disco, archivos localizados y controlados por software “Configuration Management Tool/System” o “WebPages” (URLs).
3. Posee clasificación jerárquica; clasificación de requisitos basada en la relación de trazabilidad existente entre ellos; clasificación basada en el dominio o gestión de atributos; clasificación basada en atributos definidos por el usuario; clasificación basada en la partición de acceso, en el subsistema y relación con otros elementos de la especificación.
4. Dispone de la construcción y modelo del dominio del problema utilizando:

- a. Conceptos: elementos del dominio del problema representados a través de clases conceptuales.
  - b. Diagramas de concepto: representación gráfica del dominio del problema.
5. Permite la creación de relación de trazabilidad entre requisitos y:
- a. Otros requisitos.
  - b. Elementos del dominio del problema (conceptos, entidades).
  - c. Elementos de la especificación de la solución.
  - d. Escenarios.
  - e. IRqA “test cases” y “TestDirectos tests”.
  - f. Clases de implementación.
  - g. Código fuente a través de la asociación con archivos externos.
  - h. Vistas de elementos relacionados.
  - i. Matriz de trazabilidad.
6. Los requisitos del usuario y los de la especificación de la solución son gestionados, esto proporciona algunas herramientas para verificar que todos los requisitos son satisfechos y donde (en que parte de la solución). Por lo tanto no es necesario utilizar herramienta externa para la validación de la especificación. También es fácil descubrir los elementos de la especificación que no están relacionados con ningún requisito.

Es importante resaltar que aunque la herramienta no posee versión para Software Libre, al menos posee un componente (IRqA@NET) que permite conexión de una máquina cliente en Linux, hacia un servidor de IRqA en Windows, que posibilita gestionar al equipo de desarrollo los requisitos del sistema.

### **3.3.2 Metodologías**

Como se mencionó en el capítulo 1, existen un grupo de metodologías que se adecuan más a los sistemas Web y entre ellos, los sistemas de portales. Entre las metodologías analizadas, mencionamos una de ellas, que es la que mejor cubre el ciclo de vida de los requerimientos y que se adecua al procedimiento de se definió. Es el caso de la metodología “Navigational Development Techniques” (*NDT*).

---

*NDT* es una propuesta metodológica compuesta por un proceso en el que se plantean técnicas para capturar, describir y validar los requisitos de un sistema Web y, partiendo de esos requisitos, generar de manera sistemática los modelos de análisis del sistema. El ciclo comienza definiendo los objetivos y en base a éstos se describe un proceso por el que se pueden capturar y definir los diferentes requisitos del sistema. Éstos son clasificados y tratados según la misma clasificación que se empleó en la propuesta realizada. (ESCALONA, M. J. y OTROS, 2003)

Una vez validados estos requisitos, el proceso de *NDT* propone generar tres modelos: el modelo conceptual, que representa mediante un diagrama de clases la estructura estática del sistema; el modelo de navegación, que representa mediante un conjunto de diagramas con una notación muy similar a la del diagrama de clases la forma en que se podrá navegar en el sistema; y el modelo de interfaz abstracta, que mediante un conjunto de prototipos evaluables, permite mostrar cómo se va a interactuar con el sistema.

La característica más destacable del proceso propuesto por *NDT* es que el paso de especificación de requisitos a estos modelos se hace de una manera sistemática e independiente. Es una manera sistemática porque *NDT* define algoritmos que indican cómo conseguir cada modelo a partir de la definición de requisitos. Y es independiente porque, a pesar de que existen relaciones entre los modelos, hecho que es imposible de evitar puesto que todos se refieren a un mismo sistema, no es necesario conseguir el modelo conceptual para conseguir el modelo de navegación o el de interfaz abstracta. Se enumeran a continuación un resumen de las características que posee *NDT*:

1. El proceso de *NDT* está totalmente basado en la definición de requisitos. Si se profundiza más en él, se verá que a su vez, el proceso de definición de requisitos se basa mucho en los diferentes roles de usuario que puedan aparecer en el sistema.
2. *NDT* no contempla las fases avanzadas del ciclo de vida del proyecto, como podrían ser diseño, implementación o mantenimiento. Es un proceso que permite conseguir el modelo conceptual, el modelo de navegación y el modelo de interfaz abstracta del sistema de una manera sistemática.

3. Es una propuesta orientada al proceso, puesto que para cada actividad se define de manera concreta cuál es el proceso a seguir, detallándolo de manera exhaustiva.
4. Es también una propuesta orientada al producto. En la especificación de requisitos se genera un documento de especificación de requisitos y en el análisis, el documento de análisis del sistema y los prototipos que permiten evaluar los resultados a los usuarios y clientes.
5. Además, también se puede añadir que *NDT* es una técnica orientada a la técnica, puesto que define de manera exhaustiva todas las técnicas y los modelos que propone.

### 3.4 Validación de la Propuesta

Para realizar la validación de la propuesta se consultaron personas especialistas en el tema. Inicialmente se aplicó una encuesta para determinar el grado de conocimientos que tenían en el desarrollo de proyectos de portales y en la captura de requisitos para ese tipo de sistemas. Determinando que el 100% de las personas seleccionadas como especialistas en el tema, un total de 8 personas, tienen conocimientos entre medios y altos en el tema. Otra de las consideraciones realizadas por los especialistas, es que sí es necesaria la definición de un procedimiento que sirva de guía para la ejecución del levantamiento de los requerimientos de un sistema de portales. A continuación se relacionan los aspectos más importantes que se analizaron con la aplicación de la encuesta de validación.

En el caso del análisis para determinar si las tareas que se ejecutan dentro de las actividades definidas en el procedimiento, son las correctas, se muestra en la siguiente tabla:

**Tabla 3.1: Análisis de las tareas**

	Tareas a ejecutar	Cantidad por Nivel			
		MA	A	PA	NA
	Análisis del entorno	6	2	0	0
	Definir usuarios finales	5	3	0	0
	Definir contenidos del portal	6	2	0	0
	Definir funcionalidades del portal	3	5	0	0
	Recolectar y clasificar los requerimientos	6	2	0	0
	Encontrar dependencias, etiquetar y priorizar los requerimientos	3	5	0	0
	Especificar los requerimientos según su clasificación	3	5	0	0
	Utilización de una herramienta para la gestión de requerimientos	2	6	0	0
	Verificar la calidad en la especificación de los requerimientos	3	5	0	0
	Realizar la trazabilidad de los requerimientos	5	3	0	0
	Comprobar ajuste a peticiones del cliente	6	2	0	0
	Resolver conflictos de cambios en los requerimientos	4	4	0	0

Se puede comprobar, según los resultados planteados en la tabla anterior, que todos los especialistas plantean que las tareas definidas en el procedimiento se corresponden con las características de los sistemas de portales y consideran correcta su definición, comprobando esto ya que están en un rango entre “muy adecuadas” (MA) y “adecuadas” (A).

El resto de los aspectos analizados radican en:

- ✦ La importancia que tiene la definición del procedimiento para los proyectos de portales.



- ✚ Las mejoras que puede traer para la ejecución de los proyectos, la puesta en marcha del procedimiento definido.
- ✚ La influencia en el establecimiento y mantenimiento de los acuerdos tomados entre el equipo de desarrollo y los clientes del sistema.
- ✚ La influencia que tiene en la definición de los objetivos del sistema y en la inclusión de los elementos de arquitectura de información.

Arrojando que un 90% de los especialistas concuerdan con los puntos anteriores manifestando su posición a favor de los mismos. En el caso del resto, el 10%, están de acuerdo pero consideran que con la puesta en marcha del procedimiento pueden surgir nuevos elementos que deben tenerse en cuenta en la propuesta, señalando que sería una mejora que contribuiría a realizar un procedimiento óptimo y que cumpla con las expectativas del equipo de desarrollo que implemente sistemas de portales en la *UCI*.

### **Conclusiones**

Para concluir, se obtuvo una descripción detallada del procedimiento, en cuanto a actividades, tareas, técnicas, herramientas y metodologías que ayudan a la captura de los requisitos para los sistemas de portales Web de la *UCI*.

### **Conclusiones**

Teniendo en cuenta el estudio realizado y las tareas ejecutadas para cumplimentar el mismo, se plantea como conclusiones que se obtuvo un estudio sobre las vías existentes para el levantamiento de requisitos de los proyectos, modelos clásicos de procesos y metodologías, determinando que no existe una que se adecue a los proyectos de portales. Se logró una caracterización de los proyectos de portales de la *UCI* en cuanto a la forma de realizar el levantamiento de requisitos, identificando los principales problemas que afectan la ejecución del proceso, lo que facilitó detallar un nuevo procedimiento que mejora la captura de requisitos para ese tipo de proyecto.

### Recomendaciones

- ⬇ Realizar un estudio con un mayor nivel de detalle y determinar si existe una herramienta para la gestión de requisitos que sea compatible con el sistema operativo Linux, cumpliendo con las políticas de migración de la universidad.
- ⬇ Poner en práctica el procedimiento definido, con el objetivo de acumular experiencias que posteriormente sean utilizadas en aras de mejorar el procedimiento.
- ⬇ Lograr la adecuación de una metodología, o conjunto de ellas, que sirvan para modelar el ciclo de vida completo de un portal Web y no solo la etapa de captura de los requisitos.

## Bibliografía

1. AURUM, A. y WOHLIN, C. (2005). *Engineering and Managing Software Requirements*. 477p. Germany: Springer.
2. Burg, J. F. M. (1997). *Linguistics Instruments in Requirements Engineering*. Vrije Universiteit, IOS Press.
3. BOGGS, W y BOGGS, M. (2002). *Mastering UML with Rational Rose 2002*. 701p. SYBEX.
4. COLECTIVO DE AUTORES. (2004). *Guía para el desarrollo de sitios Web*. [Consultado el: Abril de 2007]. Disponible en: <http://www.guiaweb.gob.cl>.
5. Díez, G. A. (2001). *IRQA y el desarrollo de proyectos: Experiencias Prácticas*. [Consultado el: Mayo de 2007]. Disponible en: [http://www.lsi.us.es/~amador/JIRA/Ponencias/JIRA\\_Diez.pdf](http://www.lsi.us.es/~amador/JIRA/Ponencias/JIRA_Diez.pdf).
6. Durán, T. A. y Bernárdez, J. B. (2000). *Metodología para la Elicitación de Requisitos de Sistemas Software*. [Consultado el: Diciembre de 2006]. Disponible en: <http://www.lsi.us.es/~informes/lsi-2000-10.pdf>.
7. Durán, T. A. y Bernárdez, J. B. (2001). *Metodología para el Análisis de Requisitos de Sistemas Software*. [Consultado el: Diciembre de 2006]. Disponible en: [http://www.willydev.net/descargas/WillyDEV\\_Metodologia\\_Analisis.pdf](http://www.willydev.net/descargas/WillyDEV_Metodologia_Analisis.pdf).
8. ESCALONA, M. J. (2001). *Metodologías para el desarrollo de sistemas de información global: análisis comparativo y propuesta*. [Consultado el: Abril de 2007]. Disponible en: <http://www.lsi.us.es/docs/informes/EstadoActual.pdf>.
9. ESCALONA, M. J. (2004). *Modelos y técnicas para la especificación y el análisis de la navegación en sistemas de software*. [Consultado el: Abril de 2007]. Disponible en: <http://www.lsi.us.es/docs/doctorado/tesis/tesis.pdf>.
10. ESCALONA, M. J. y KOCH, N. (2002). *Ingeniería de Requisitos en Aplicaciones para la Web: Un estudio comparativo*. [Consultado el: Diciembre de 2006]. Disponible en: <http://ourproject.org/docman/view.php/144/215/requisitos.pdf>.
11. ESCALONA, M. J y OTROS. (2002). *NDT: Navigational Development Techniques*. [Consultado el: Abril de 2007]. Disponible en: <http://lsi.ugr.es/~gedes/actividades/Dolmen4/a11.pdf>.

12. ESCALONA, M. J. y OTROS. (2001). *Desarrollo de la navegación en entornos Web*. [Consultado el: Abril de 2007]. Disponible en: <http://www.dlsi.ua.es/webe02/articulos/5.pdf>.
13. ESCALONA, M. J. y OTROS. (2003). *Utilización de NDT y de las Técnicas de Satisfacción de Restricciones para la Generación de Itinerarios Culturales*. Revista Computación y Sistemas. Vol. 7-Nº3-pp.76-91. [Consultado el: Abril de 2007]. Disponible en: <http://www.ejournal.unam.mx/compuysistemas/vol07-02/CYS07202.pdf>.
14. GRANOLLERS, T. (2004). *MPlu+a. Una Metodología que integra la Ingeniería del Software, la Interacción Persona-Ordenador y la accesibilidad en el contexto de equipos de desarrollo multidisciplinares*. [Consultado el: Diciembre de 2006]. Disponible en: <http://www.tesisenxarxa.net/TESIS UdL/AVAILABLE/TDX-0218107-133615//Tgsa4de5.pdf>.
15. HASSAN, M. Y. y MARTÍN, F. F. J. (2004). *Propuesta de adaptación de la metodología de diseño centrada en el usuario para el desarrollo de sitios Web accesibles*. Revista Española de Documentación Científica. Vol. 27-Nº3-2004. [Consultado el: Mayo de 2007]. Disponible en: [http://www.nosolousabilidad.com/hassan/DCU\\_accesible.pdf](http://www.nosolousabilidad.com/hassan/DCU_accesible.pdf).
16. HASSAN, M. Y y MARTÍN, F. F. J. (2003). *Que es la Accesibilidad Web*. Revista No Solo Usabilidad. Vol. Nº2-14/07/2003. [Consultado el: Mayo de 2007]. Disponible en: <http://www.nosolousabilidad.com/articulos/accesibilidad.htm>.
17. HASSAN, M. Y.; MARTÍN, F. F. J. y IAZZA, G. (2004). *Diseño Web Centrado en el Usuario: Usabilidad y Arquitectura de la Información*. [Consultado el: Mayo de 2007]. Disponible en: <http://www.hipertext.net/web/pag206.htm>.
18. HASSAN, M. Y. y NÚÑEZ, P. A. (2005). *Diseño de arquitecturas de información: Descripción y Clasificación*. Revista No Solo Usabilidad. Vol. Nº4. [Consultado el: Mayo de 2007]. Disponible en: [http://www.nosolousabilidad.com/articulos/descripcion\\_y\\_clasificacion.htm](http://www.nosolousabilidad.com/articulos/descripcion_y_clasificacion.htm).
19. HERNÁNDEZ, L. R. A.; RÍOS, D. J. L. y COELLO, G. S. (1998). *El paradigma cuantitativo de la investigación científica*. Maestría en Ciencias de la Educación. Centro Universitario Sancti Spiritus. Cuba.
20. ISO-9241-11. (1998). *Ergonomic requirements for office work with visual display terminals (VDT)s - Part 11 Guidance on usability*.
21. IEEE. (1998). *IEEE Guide for developing System Requirements Specifications*.
22. JACOBSON, I., BOOCH, G. y RUMBAUGH, J. (2000). *El proceso Unificado de Desarrollo de Software*. 435p. Madrid, España: Addison Wesley.

23. KOCH, N.; KRAUS, A. y HENNICKER, R. *The authoring process of the UML-based Web engineering approach*. [Consultado el: Mayo de 2006]. Disponible en: <http://www.dsic.upv.es/~west/iwwost01/files/contributions/NoraKoch/Uwe.pdf>.
24. LARMAN, C. (1999). *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. 536p. México: Prentice Hall.
25. LEFFINGWELL, D. y WIDRIG, D. (2003). *Managing Software Requirements: A Use Case Approach, 2nd Edition*. 544p. USA: Addison Wesley.
26. LUTOWSKI, R. (2005). *Software Requirements. Encapsulation, quality and reuse*. 243p. USA: Auerbach Publications.
27. MCDONALD, L. B. A. (2005). *Definición de perfiles en herramientas de gestión de requisitos*. [Consultado el: Diciembre de 2006]. Disponible en: [http://www.willydev.net/descargas/WillyDEV\\_Metodologia\\_Analisis.pdf](http://www.willydev.net/descargas/WillyDEV_Metodologia_Analisis.pdf).
28. MEDINA, M. J. C. (2004). *Análisis comparativo de técnicas, metodologías y herramientas de Ingeniería de Requerimientos*. [Consultado el: Abril de 2007]. Disponible en: <http://www.cs.cinvestav.mx/Estudiantes/TesisGraduados/2004/tesisJuanCarlosM.pdf>.
29. OLSINA, L. A. (1999). *Metodología Cuantitativa para la Evaluación y Comparación de la Calidad de Sitios Web*. [Consultado el: Abril de 2007]. Disponible en: <http://www.di.uniovi.es/~cueva/investigacion/tesis/WebsiteQEM.pdf>.
30. PRESSMAN, R. S. (2005). *Ingeniería de Software: Un enfoque práctico*. 5ta Edición. 435p. La Habana: Editorial Félix Varela.
31. RATIONAL. (2003). *Rational Unified Process Extended Help*. [Disponible en: Software, R. Rational Unified Process Extended Help, 2003].
32. ROBERTSON, S y ROBERTSON, J. (2006). *Mastering the Requirements Process. 2nd Edition*. 245p. USA: Addison Wesley.
33. RONDA, L. R. (2005). *La Arquitectura de la Información y las Ciencias de la Información*. Revista No Solo Usabilidad. Vol. N°4. [Consultado el: Mayo de 2007]. Disponible en: [http://www.nosolousabilidad.com/articulos/ai\\_cc\\_informacion.htm](http://www.nosolousabilidad.com/articulos/ai_cc_informacion.htm).
34. SÁNCHEZ, C. *Diccionario de Informática e Internet*. (2001). 165p. [Consultado el: Abril 2007].
35. SÁNCHEZ, J. y OTROS. *VRU: Un método para validar requisitos y generar interfaces de usuario multiplataforma*. Consultado el: Abril de 2006]. Disponible en: [http://oomethod.dsic.upv.es/anonimo/..%5Cfiles%5CInConferenceArticle%5C2002\\_WER.pdf](http://oomethod.dsic.upv.es/anonimo/..%5Cfiles%5CInConferenceArticle%5C2002_WER.pdf).

36. SCHMULLER, J. (1999). *Aprendiendo UML en 24 horas*. 448p. México: Pearson Educación.
37. SOMMERVILLE, I. (1996). *Software Engineering. 5th Edition*. 653p. USA: Addison Wesley.
38. SUMANO, L. M. A. (1999). *Análisis de requerimientos de software. Estado del arte*. [Consultado el: Diciembre de 2006]. Disponible en: <http://www.geocities.com/diegolp/ingsof/requerimientos.pdf>.
39. TRAMULLAS, J. (2001). *Diseño de información para la Web, 1996-2000. Un análisis bibliográfico selectivo*. [Consultado el: Diciembre de 2006]. Disponible en: <http://www.ingentaconnect.com/content/routledg/epri/2001/00000010/00000012/art00008?crawler=true>.
40. VALDERAS, A. P. J. (2004). *Especificación de requisitos en el desarrollo de aplicaciones Web*. [Consultado el: Abril de 2007]. Disponible en: <http://oomethod.dsic.upv.es/anonimo/..%5Cfiles%5CTechnicalReport%5Creq.pdf>.
41. WIEGERS, K. E. (2006). *More About Software Requirements: Thorny Issues and Practical Advice*. 216p. USA: Microsoft Press.
42. WINDLE, D. R y ABREO, L. R. (2002). *Software Requirements Using the Unified Process: A Practical Approach*. 288p. USA: Prentice Hall.
43. WORTMANN, J. (2001). *Concurrent Requirements Engineering with UML Subset based on Component Schema Relationships*. [Consultado el: Mayo de 2007]. Disponible en: [http://edocs.tu-berlin.de/diss/2001/wortmann\\_jan.pdf](http://edocs.tu-berlin.de/diss/2001/wortmann_jan.pdf).
44. YOUNG, R. R. (2004). *The Requirements Engineering Handbook*. 245p. Boston, London: Artech House.

### Glosario de Términos

#### A

**Artefacto:** Productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.

#### C

**Captura de Requisitos (CR):** Proceso durante el cual se identifica un problema y se especifica los requisitos que debe cumplir un producto de software.

**Caso de Uso:** Notación utilizada para representar los requerimientos funcionales de un sistema basada en el esquema propuesto por Ivar Jacobson a principios de la década del '90.

**Cliente:** Aquella persona o empresa que contrata al desarrollador de software.

**CMM:** Capacity Maturity Model fue desarrollado por el Software Engineering Institute (SEI), su principal premisa es la calidad de un producto determinada por la calidad del proceso utilizado para desarrollarlo y mantenerlo. Es un modelo de referencias sobre buenas prácticas, consolidadas y probadas para el desarrollo y mantenimiento de productos y servicios, cubriendo todo el ciclo de vida, desde la concepción a la entrega y mantenimiento. Integra la Ingeniería de Software, la Ingeniería de Sistemas y la Adquisición de Productos y Servicios.

#### E

**Entrevista:** Acción y efecto de entrevistar o entrevistarse. Vista, concurrencia y conferencia de dos o más personas en lugar determinado, para tratar o resolver un negocio.

**Especificaciones de Requisitos:** Documento que describe lo que hace un sistema de software: sus funciones y sus atributos. Generalmente escritas desde el punto de vista del usuario.

#### H

**Herramienta CASE:** Computer Aided Software Engineering, son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas ayudan en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar



## Glosario de Términos

---

un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores.

### I

**IEEE:** Institute of Electrical and Electronics Engineers: Importante asociación de técnicos y profesionales, con sede en los Estados Unidos. Fue fundada en 1884 y favorece la investigación en campos diversos, como la tecnología aeroespacial, la computación, las comunicaciones y la tecnología biomédica. Promueve la estandarización de normas.

**Ingeniería de Software:** La Ingeniería de Software es una tecnología multicapa en la que, según Pressman, se pueden identificar: los métodos, el proceso y las herramientas.

**Ingeniería de Requerimiento (IR):** Se define, como un conjunto de actividades en las cuales, utilizando técnicas y herramientas, se analiza un problema y se concluye con la especificación de una solución (a veces más de una).

### J

**JAD:** Joint Application Development desarrollada por IBM en 1977, es una alternativa a las entrevistas individuales que se desarrolla a lo largo de un conjunto de reuniones en grupo durante un período de 2 a 4 días. En estas reuniones se ayuda a los clientes y usuarios a formular problemas y explorar posibles soluciones, involucrándolos y haciéndolos sentirse partícipes del desarrollo.

### L

**Lista de Chequeo:** Se entiende por lista de chequeo “cheks-list” a un listado de preguntas, en forma de cuestionario que sirve para verificar el grado de cumplimiento de determinadas reglas establecidas a priori con un fin determinado. El uso de estas listas está generalizado en rubros muy diversos que van desde verificar y determinar el potencial de mercados extranjeros hasta medir la confiabilidad y seguridad de sistemas informáticos, incluyendo ítems tales como la evaluación de criterios de usabilidad de un sitio de Internet, como así también la verificación de un plan.

### O

**Observación:** Es la acción de mirar detenidamente una cosa para asimilar en detalle la naturaleza investigada, su conjunto de datos, hechos y fenómenos.

### P

**Proceso de Desarrollo de Software:** Un Proceso de Desarrollo de Software es la definición del conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto, a modo de plantilla que explica los pasos necesarios para terminar el proyecto. Un PROCESO define: “QUIEN”, “QUE”, “CUANDO”, y “COMO” hay que realizar las cosas para alcanzar un determinado producto de software.

**Prototipo:** Es un modelo a escala o imitación de lo real, pero no tan funcional para que equivalga a un producto final, ya que no lleva a cabo la totalidad de las funciones necesarias del sistema final. Proporcionando una retroalimentación temprana por parte de los usuarios acerca del Sistema.

### R

**Rational RequisitePro:** Mantiene a todo el Equipo de desarrollo actualizado a través del proceso de desarrollo de aplicaciones haciendo que los requerimientos se puedan escribir, comunicar y cambiar fácilmente.

**Rational Rose:** La herramienta líder en el mundo de modelación visual para el proceso de modelación del negocio, análisis de requerimientos y diseño de arquitectura de componentes.

**Rational Unified Process (RUP):** En un proceso que de manera ordenada define tareas y quién cómo y cuándo el Equipo de desarrollo las hará, son un conjunto de metodologías adaptables al contexto y necesidades de cada organización. Define un ciclo de vida iterativo, priorizando el uso de lenguajes de modelado, casos de uso y centrado en la arquitectura.

**Requerimiento:** Una condición o capacidad necesaria para un usuario para resolver un problema o alcanzar un objetivo.

**Requerimiento funcional:** Alguna actividad o servicio que debe realizar el software y que puede ser comprobada.

**Requerimiento no funcional:** Limitación de hardware o software bajo la cual el sistema debe operar.

## Glosario de Términos

---

**Rol:** Un conjunto de expectativas de conducta asociadas a una persona, un patrón de comportamiento que se espera de quién desempeñe cada puesto, con cierta independencia de la persona que sea.

### S

**SEI:** Software Engineering Institute es un instituto federal estadounidense de investigación y desarrollo, fundado por el Congreso Estadounidense en 1984 para desarrollar modelos de evaluación y mejora en el desarrollo de software, que dieran respuesta a los problemas que generaba al ejército estadounidense la programación e integración de los sub-sistemas de software en la construcción de complejos sistemas militares. Financiado por el Departamento de Defensa estadounidense y administrado por la Universidad Carnegie Mellon.

**Storyboards:** Es una organización de gráficos, como una serie de ilustraciones o imágenes mostradas en secuencia para el propósito de pre visualizar un movimiento gráfico o una secuencia media interactiva, incluyendo un sitio Web interactivo.

### T

**Tormenta de ideas:** Es una herramienta de planeamiento que se puede utilizar para obtener ideas a partir de la creatividad de un grupo y con ello resolver un problema. El fundamento del brainstorming es la generación de ideas, en modo individual o en grupo, evitando evaluaciones inmediatas: la investigación científica ha demostrado que este principio es altamente productivo tanto en el esfuerzo individual como en el trabajo de grupo.

### U

**UCI:** Universidad de las Ciencias Informáticas

**UML:** “Unified Modeling Language” Lenguaje gráfico que brinda un vocabulario y reglas para especificar, construir, visualizar y documentar los artefactos de un sistema utilizando el enfoque orientado a objetos.

**Usuario:** Aquella persona o grupo de personas dentro de la empresa que utilizarán el software desarrollado, incluido el tomador de decisiones.

**Usuario Final:** Toda aquella persona u organización siendo influenciada o ejerciendo influencia sobre el software que está siendo construido.

## **Anexos**

### **Anexo 1 – Cronograma de Captura de Requisitos**

**Portada**

**Revisiones Históricas**

**Reglas de Confidencialidad**

**Índice**

1- Introducción

1.1- Propósito

1.2- Alcance

1.3- Descripción del cronograma

*Nota aclaratoria:*

*Solo se detallará la plantilla definida en la investigación (Especificación de Requisitos. El resto a las que se hace alusión son las que actualmente se utilizan en la documentación de los proyectos.*

**Anexo 2 – Proyecto Técnico**

**Portada**

**Control del Documento**

**Reglas de Confidencialidad**

**Historial de Versiones**

**Índice**

1- Antecedentes

2- Problemática

3- Objetivo General

3.1- Objetivos Especificos

4- Alcance

5- Solución Propuesta

5.1- Características de la Solución

5.2- Etapas y Entregables

5.3- Metodología

5.4- Estructura del Equipo Ejecutor

6- Impactos y Beneficios Esperados

**Anexo 3 – Plan de Requisitos**

**Portada**

**Revisiones Históricas**

**Reglas de Confidencialidad**

**Índice**

**1- Introducción**

1.1- Propósito

1.2- Alcance

1.3- Definiciones, Acrónimos y Abreviaturas

1.4- Referencias

**2- Roles involucrados**

2.1- Por la Parte Cliente

2.2- Por la Parte del Equipo de Desarrollo

**3- Definición de la Documentación**

**4- Estrategia de Captura de Requerimientos**

4.1- Flujo Propuesto

4.2- Etapas

**Anexo 4 – Glosario de Términos**

**Portada**

**Revisiones Históricas**

**Reglas de Confidencialidad**

**Índice**

1- Introducción

1.1- Propósito

1.2- Alcance

1.3- Referencias

1.4- Resumen

2- Definiciones

2.1- Término 1

2.2- Término 2

2.3- Grupo de términos

---

**Anexo 5 – Especificación de Requisitos**

**Portada**

**Revisiones Históricas**

**Reglas de Confidencialidad**

**Historial de Versiones**

**Índice**

- 1- Introducción
  - 1.1- Propósito
  - 1.2- Referencias
- 2.- Participantes en el proyecto
  - 2.1- Por la parte Cliente
  - 2.2- Por la parte del Equipo de desarrollo
- 3.- Objetivos del sistema
- 4.- Especificación de requisitos del sistema
  - 4.1- Requisitos de datos
  - 4.2- Requisitos de interfaz
  - 4.3- Requisitos navegacionales
  - 4.4- Requisitos de personalización
  - 4.5- Requisitos funcionales
    - 4.5.1- Diagramas de casos de uso del sistema
    - 4.5.2- Definición de actores
    - 4.5.3- Casos de uso del sistema



4.6- Requisitos no funcionales  
5.- Matriz de trazabilidad de requisitos

Especificación de cada uno de los elementos de la plantilla mostrada anteriormente:

**Portada**

*[En la portada del documento se agrupan elementos tales como: el nombre del proyecto, nombre del módulo que se esté realizando, la versión del mismo así como un logotipo que identifique el proyecto en caso de tenerlo]*

**Revisiones Históricas**

*[Es una tabla compuesta por los siguientes campos: fecha, versión, descripción y nombre de la persona encargada de realizar las actualizaciones de las revisiones (autor)]*

**Reglas de Confidencialidad**

*[Texto donde se describen las reglas de confidencialidad del documento en cuestión]*

**Historial de Versiones**

*[Tabla que relaciona las versiones del documento]*

**Índice**

*[Tabla de contenido que permite el acceso directo a cada uno de los elementos relacionados en la plantilla]*

1- Introducción

*[Facilita una descripción breve de las principales características del sistema software que se va a desarrollar, la situación actual que genera la necesidad del nuevo desarrollo, la problemática que se acomete, y cualquier otra consideración que sitúe al posible lector en el contexto oportuno para comprender el resto del documento. Incluye propósito y referencias]*

1.1- Propósito

*[Describe el propósito del documento]*

1.2- Referencias

*[Facilita una lista completa de todos los documentos referenciados en algún otro lugar de la Especificación. Se debe identificar cada documento por título, número de reporte si es aplicable, fecha, y publicador. Especificar las fuentes de las cuales fueron obtenidas las referencias. Esta información puede ser facilitada por referencia a un apéndice o a otro documento]*

2.- Participantes en el proyecto

*[Describe las personas involucradas con el proyecto, teniendo en cuenta dos clasificaciones, las que se detallan a continuación. En cada uno de los casos se debe mostrar el nombre completo, papel o rol que desempeña en el proyecto, organización a la que pertenece y cualquier otra información que se necesaria reflejar]*

### 2.1- Por la parte Cliente

*[Datos relacionados con el cliente y especificados anteriormente]*

### 2.2- Por la parte del Equipo de desarrollo

*[Datos relacionados con el equipo de desarrollo y especificados anteriormente]*

### 3.- Objetivos del sistema

*[Especificación de la lista de objetivos que se esperan alcanzar cuando el portal esté en explotación]*

### 4.- Especificación de requisitos del sistema

*[Se detallarán los requisitos del portal que serán chequeados posteriormente, para analizar su cumplimiento a través de algún objetivo. Se especifican los tipos de requisitos que deben especificarse según la clasificación siguiente]*

#### 4.1- Requisitos de datos

*[Estos requisitos son denominados también requisitos de contenido, requisitos conceptuales o requisitos de almacenamiento de información. Éstos se refieren a la pregunta de qué información debe almacenar y administrar el sistema]*

#### 4.2- Requisitos de interfaz

*[También llamados en algunas propuestas requisitos de interacción o de usuario. Responden a la pregunta de cómo va a interactuar el usuario con el sistema]*

#### 4.3- Requisitos navegacionales

*[Estos requisitos recogen las necesidades de navegación del usuario]*

#### 4.4- Requisitos de personalización

*[Están referidos a cómo debe adaptarse el sistema en función de qué usuario interactúe con él y de la descripción actual de dicho usuario]*

#### 4.5- Requisitos funcionales

*[Los tradicionales requisitos de un sistema, que expresan condiciones o capacidades que el sistema debe cumplir, sin incluir aspectos de interfaz o interacción. También son conocidos en el ambiente Web como requisitos de servicios. Se detallan a través de los elementos que aparecen a continuación]*

##### 4.5.1- Diagramas de casos de uso del sistema

*[Se refleja la transformación de los requisitos del sistema en funcionalidades propiamente dichas. Es una representación gráfica que muestra la interacción de los actores del sistema con esas funcionalidades]*

##### 4.5.2- Definición de actores

*[Son los terceros que juegan un rol en el sistema, interactuando y beneficiándose con el mismo]*

### 4.5.3- Casos de uso del sistema

*[Fragmentos de funcionalidades que el sistema ofrece para aportar un resultado o valor para sus actores]*

### 4.6- Requisitos no funcionales

*[Son las propiedades o cualidades que debe tener un sistema, se clasifican de diversas formas, en requisitos de soporte, usabilidad, legales, entre otros]*

### 5.- Matriz de trazabilidad de requisitos

*[Matriz que debe irse construyendo a medida que avanza la definición de los requisitos, definida en las dimensiones de requisitos contra objetivos del sistema, garantizando que cada uno de los requisitos le de cumplimiento al menos a uno de los objetivos, en caso contrario se deben realizar revisiones a la definición de los requisitos]*

## **Anexo 6 – Seguimiento de Requisitos**

**Portada**

**Revisiones Históricas**

**Reglas de Confidencialidad**

**Índice**

1- Introducción

1.1- Propósito

1.2- Referencias

2- Seguimiento de Requisitos

2.1- Documento de Seguimiento

2.2- Herramientas

2.3- Seguimiento

2.4- Atributos

2.4.1- Estado

2.4.2- Beneficio

2.4.3- Esfuerzo

2.4.4- Estabilidad

2.5- Entregables del Proyecto

---

**Anexo 7 – Solicitud de Cambios**

**Portada**

**Revisiones Históricas**

**Reglas de Confidencialidad**

**Índice**

1- Introducción

1.1- Propósito

1.2- Referencias

2- Formulario de pedido de cambio

2.1- Identificación

2.1.1- Proyecto

2.1.2- Número

2.1.3- Tipo

2.1.4- Título

2.1.5- Fecha creación

2.1.6- Creada por

2.1.7- Prioridad

2.2- Problema actual

2.2.1- Descripción

2.2.2- Como repetir

2.2.3- Nuevos requerimientos

2.2.4- Condiciones bajo las que fue observado el problema

2.2.5- Ambiente actual. Hardware

2.2.6- Sistema operativo

2.3- Cambio propuesto (Creador)

2.3.1- Descripción

2.3.2- Costo estimado

2.4- Cambio propuesto (Equipo de desarrollo)

2.4.1- Acción

2.4.2- Decisión tomada

2.4.3- Elementos de configuración aceptados

2.4.4- Errores corregidos

2.4.5- Nuevas funcionalidades

2.5- Resolución

2.5.1- Tiempo y costo estimado del cambio propuesto

2.5.2- Desarrollador