

**Universidad de las Ciencias Informáticas  
Facultad #4**



**Título: “Estrategia para uso del prototipo como técnica para obtener, representar y validar requisitos así como la transición a la interfaz gráfica”**

Trabajo de Diploma para optar por el título de  
Ingeniero en Ciencias Informáticas

**Autor:** Arianna Camacho Hernández

**Tutor:** Ing. Arturo Cesar Arias Orizondo

**Consultante:** Lc. Luis J. Guzmán Hernández

Ciudad de La Habana, Junio del 2007

## DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al Proyecto SIGEP de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año 2007.

Arianna Camacho Hernández

---

Ing. Arturo César Arias Orizondo

---

## DATOS DE CONTACTO

**Tutor:** Ing. Arturo Cesar Arias Orizondo

Profesor Instructor. Graduado en el 2003 de Ingeniero Informático del Instituto Superior Politécnico José Antonio Echeverría (ISPJAE), Título de Oro, promedio 5 puntos. Ha impartido las asignaturas de Sistemas de Bases de Datos, Ingeniería de Software I y II, Gestión de Software en la Universidad de Ciencias Informáticas, participando en la elaboración de los programas de estas asignaturas. Miembro del tribunal de acreditación de competencias del Departamento de Ingeniería de Software. Vicedecano de producción e investigación de la UCI por dos cursos: 2004-2005, 2005-2006. Miembro del consejo científico de la UCI. Miembro de la comisión de carrera durante los cursos 2004-2005, 2005-2006. Ha impartido cursos de postgrado sobre RUP y UML. Participó en el Congreso y Feria TecnoInternet 2004 celebrado en Santiago de Chile, Chile. Ha participado en evaluaciones de calidad de productos de software para Venezuela. Diseñador de base de datos del proyecto SAFRE, primero de exportación de la UCI. Líder del proyecto de desarrollo de software para el sistema penitenciario venezolano desde el 2005.

**Consultante:** Lic. Luis Joaquín Guzmán Hernández

Graduado de Licenciatura en Ciencias de la Computación en la Universidad Central "Martha Abreu" de Las Villas en el año 2002. Trabajó durante dos años en el Centro de Información Científica de la Universidad como especialista en informática. Actuó como líder en varios proyectos de la rama de la gestión de información. En el año 2005 comienza a trabajar en la UCI, desempeñándose como especialista en una dirección de la infraestructura productiva. Ha participado en muchos proyectos de desarrollo de software, fundamentalmente en el desarrollo de aplicaciones para WEB. Actualmente se desempeña como Director de la dirección de producción # 2 de la Infraestructura Productiva Central.

# OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

Título: Estrategia para uso del prototipo como técnica para obtener, representar y validar requisitos así como la transición a la interfaz gráfica

Autor: Arianna Camacho Hernández

El tutor del presente Trabajo de Diploma considera que durante su ejecución la estudiante mostró las cualidades que a continuación se detallan:

---

---

---

---

---

---

---

---

---

---

---

---

---

Por todo lo anteriormente expresado considero que el estudiante está apto para ejercer como Licenciado en Ciencia de la Computación e Ingeniero Informático respectivamente; y propongo que se le otorgue al Trabajo de Diploma la calificación de \_\_\_\_.

\_\_\_\_\_  
Firma

\_\_\_\_\_  
Fecha

## AGRADECIMIENTOS

*A la Revolución, por ser artífice de este gran proyecto.*

*A toda mi familia, de manera especial a todos aquellos que siempre me han apoyado y han depositado su confianza en la materialización de este logro.*

*A los que les faltó la convicción en cierto momento, por ellos también trabajé mucho para devolvérselas, entre ellos mi madre, porque el esfuerzo sirvió para demostrarle que se puede ser bueno en lo que uno se plantea ser bueno.*

*A todos los que desinteresadamente consagraron parte de su tiempo para ofrecer muchas de las ideas que también son parte de este trabajo.*

*A todos,*

*Agradecida infinitamente.*

## DEDICATORIA

*A mi papá, porque este sueño es tan suyo como mío y ha añorado tanto alcanzarlo como yo. No es a su comprensión y apoyo incondicional en estos últimos 5 años a quienes honro con esta dedicatoria, sino a su ejemplo de toda una vida, meritorio del mayor de los galardones.*

*A alguien a quien infelizmente le faltó un poco de vida para verme como siempre soñó, uno de los seres más importantes en mi vida: mi abuela querida, a su presencia eterna para mí. Su paciencia desmedida, apoyo, comprensión, sus regaños y su amor infinito, tuvieron su recompensa, nada fue en vano!*

*A mi hermano también dedico este trabajo, a retribuirle una pequeña fracción de lo mucho que ha hecho por mí.*

*A todos los que han confiado en mí en momentos que me ha faltado convicción.*

*A toda mi familia, todos los que saben que son mis amigos, siéntanse parte de esta dedicatoria.*

## RESUMEN

En la República Bolivariana de Venezuela se está llevando a cabo un plan de automatización del sistema penitenciario con vistas a solucionar los problemas de escaso control de la información existentes en la actualidad. Como parte importante del programa antes mencionado, se ha llevado a cabo un estudio de los procesos de gestión penitenciaria de la Dirección General de Servicios Penitenciarios y las dificultades existentes conducen a la necesidad de automatizar los principales procesos en estas instituciones, mediante el desarrollo de un Sistema Integral de Gestión Penitenciaria (en lo adelante SIGEP).

La Ingeniería de Requisitos se considera la etapa más crítica en el proceso de desarrollo del software por el impacto que tiene en el desempeño de las demás. Lo ideal es que se concluya con una aceptación por parte del cliente de que los requerimientos cubren las necesidades de su sistema porque si se pasa al desarrollo sin una validación de que los requisitos son correctos, puede que el cliente no vea en el producto final la salida a la totalidad de sus problemas. El trabajo que se presenta a continuación propone una estrategia para apoyar la captura, representación y validación de requisitos del proyecto SIGEP con el empleo de un prototipo, que aporte una visión más clara acerca de la funcionalidad del sistema para los clientes y conlleve al desarrollo de una Ingeniería de Requisitos satisfactoria.

### PALABRAS CLAVE

Sistema Integral de Gestión Penitenciaria (SIGEP), estrategia, captura de requisitos, representación de requisitos, validación de requisitos, prototipo, funcionalidad.

# ÍNDICE

INTRODUCCIÓN .....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA .....	5
1.1 Introducción .....	5
1.2 Ingeniería de Requisitos .....	5
Fig 1.1 Actividades que contempla la Ingeniería de Requisitos.....	6
1.2.1 ¿Qué son los requisitos? .....	6
1.2.2 ¿Qué es la captura de requisitos? .....	7
1.2.3 Técnicas usadas para capturar requisitos .....	7
1.2.4 Definición de requisitos .....	9
1.2.5 Técnicas para Definición de requisitos .....	9
1.2.6 Validación de requisitos .....	12
1.2.7 Técnicas se usan para validar requisitos .....	12
1.3 Ingeniería de Requisitos desde el punto de vista metodológico .....	13
1.3.1 RUP.....	13
1.3.2 Design-driven Requirements Elicitation .....	13
1.4 El prototipo .....	14
1.4.1 ¿Qué es el prototipo?.....	14
1.4.2 Tipos de prototipo .....	15
1.4.3 Objetivos del prototipo .....	16
1.4.4 Finalidad del prototipo.....	17
1.4.5 Condiciones para emplear prototipo. ....	17
1.4.6 Etapas del método con prototipos.....	17
1.4.7 Estrategias para uso de prototipos .....	17
1.4.8 Papel del prototipo .....	18
1.4.9 ¿Cuándo se prototipa?.....	18
1.4.10 Normas que respaldan el uso del prototipo .....	19
1.5 Las interfaces de usuario .....	19
1.5.1 Herramientas útiles en la elaboración de Interfaces de Usuario.....	19
1.5.2 Tecnologías.....	21



1.6 Conclusiones .....	22
CAPÍTULO 2: EL USO DEL PROTOTIPO EN EL PROYECTO SIGEP .....	23
2.1 Introducción .....	23
2.2 Especificación de requisitos.....	23
2.2.1 Modelación del negocio .....	24
2.2.2 Definición de estructuración del sistema.....	24
2.2.2 Consideraciones previas al prototipado.....	25
2.2.3 Definición y descripción de funcionalidades .....	26
2.2.5 Proposición de pantallas preliminares .....	27
2.2.6 Montaje de pantallas en un “sistema navegable no funcional” .....	27
2.2.7 Tipo de prototipo .....	29
2.2.8 Documento descriptivo del prototipo.....	29
2.2.9 Las validaciones con el cliente y modificaciones.....	30
2.2.10 ¿Hasta dónde validar?.....	30
2.3 Del prototipo a la interfaz gráfica .....	31
2.3.1 Especificaciones Generales para el Diseño de la Aplicación IASP .....	31
2.3.2 Confección de las pantallas .....	32
2.4 Conclusiones .....	35
CAPÍTULO 3: ANÁLISIS DE LOS RESULTADOS .....	36
3.1 Introducción .....	36
3.2 Método empírico: Encuesta .....	36
3.2.1 Valoración de la técnica de uso del prototipo para especificar requisitos.....	37
Fig. 3.1 Prototipo como técnica para Especificación de Requisitos.....	38
3.2.2 Valoración del grado de dificultad del trabajo a partir del prototipo .....	38
Fig 3.2 Valoración del trabajo de diseñadores a partir del prototipo.....	39
3.2.3 Comparación de la técnica del prototipo con otras técnicas de especificación de requerimientos .....	40
Fig 3.3 Comparación con otras técnicas.....	41
3.3 Primera prueba de validación con el cliente .....	41
3.4 Ventajas obtenidas del uso del prototipo .....	42

3.4 Limitaciones del prototipo .....	44
3.5 Elementos adicionales necesarios.....	44
3.6 Conclusiones .....	44
CONCLUSIONES GENERALES .....	45
RECOMENDACIONES.....	46
BIBLIOGRAFÍA.....	46
REFERENCIA BIBLIOGRÁFICA .....	47
GLOSARIO .....	48

## **INTRODUCCIÓN**

El mundo está experimentando grandes cambios, el uso de las tecnologías de la informática y las comunicaciones (TIC) se hace cada vez más imprescindible y la mayoría de los países están enfocando sus esfuerzos en informatizar los procesos, con el fin de hacerlos de una manera más cómoda, rápida y eficiente. Es por ello que una de las principales tareas del Gobierno Cubano es desarrollar la Industria del Software, no solamente con el fin del desarrollo de sistemas para la informatización de la sociedad, sino también por los beneficios de insertarnos en el mercado de software a nivel mundial, dada su perspectiva económica. Con el objetivo de lograr un vínculo estrecho entre la Universidad y las Empresas se creó la Universidad de las Ciencias Informáticas (UCI), una Universidad Productiva que desarrolla en su seno un número importante de proyectos que persiguen las metas antes mencionadas, algunos comprometidos con entidades nacionales y otros, con un nivel de responsabilidad mayor ya que involucran el prestigio del país, con entidades de otros países. La hermana República Bolivariana de Venezuela por ejemplo, sostiene varios contratos con la universidad en este sentido, para poco a poco ir automatizando los procesos en algunas de sus organizaciones y de esta forma dar solución a una parte considerable de los problemas que en estas se originan.

En la Dirección General de Servicios Penitenciarios (DGSP), la mayor parte de la información se manipula de forma manual y no existe un mecanismo de control para esta. Esta situación dificulta la toma de decisiones con una visión realmente objetiva de la realidad en la mayoría de dichos centros.

La República Bolivariana de Venezuela presentó un proyecto de Modernización del Sistema Penitenciario para desarrollar un sistema informático que diera solución a estos inconvenientes y de esta forma atenuar las dificultades existentes en estas entidades. Con la meta principal de desarrollar dicho sistema software para dar salida a los problemas mencionados, fue creado en la UCI el proyecto SIGEP (Sistema Integral de Gestión Penitenciaria).

En el proceso de desarrollo del software cada etapa tiene una importancia trascendental y la Ingeniería de Requisitos es una de las fases más críticas dentro del ciclo de vida de un proyecto, pues en ella se capturan y definen las necesidades que debe cubrir el sistema. En el ciclo de desarrollo del software se presentan varios problemas y entre ellos las dificultades que presenta el cliente para establecer explícitamente desde el inicio todos los requerimientos. En los últimos años se ha estado reconociendo en mayor medida la importancia que tiene el buen desenvolvimiento de esta primera fase y los riesgos en que se incurren si esta es realizada en forma incompleta o incorrecta. Actividades propias de esta área, como

la captura de requisitos del usuario, la definición de requisitos y la validación de los mismos, son algunas de las consideradas más significativas en el desarrollo y la producción del software por el impacto que tiene su resultado en el resto de las fases, ya que de esta etapa se obtienen las necesidades del sistema y el equipo de desarrollo adquiere la información necesaria para desarrollar un producto de calidad.

Una Ingeniería de Requisitos no desarrollada de manera adecuada, tiene varias consecuencias, la mayoría dadas por la no exploración a fondo de la funcionalidad del sistema. Por tal motivo es que en ocasiones emergen requisitos funcionales en etapas avanzadas de desarrollo, pudiendo haber sido detectados al inicio. De esta forma sería menos costoso y no provocaría transformaciones en el tiempo previsto para cada actividad; además evitaría retrasos en la entrega del artefacto final. Suele suceder que se comienza a desarrollar un producto tras el levantamiento de los requisitos sin que estos hayan sido aprobados y pudiera ser que cuando esté consumado el producto el usuario no vea en el mismo la salida a la totalidad de sus problemas.

La importancia de esta fase es esencial puesto que los errores más comunes y más costosos de reparar, así como los que más tiempo consumen, se deben a una inadecuada ingeniería de requisitos. La mayor parte de los costos de mantenimiento se deben a cambios en las necesidades por lo que para reducir estos costos y por consiguiente los del ciclo de vida en su totalidad, debe establecerse una expresión más exacta de las necesidades del usuario. Existen propuestas que sugieren la introducción de un modelo evolutivo en el ciclo de desarrollo del software. Se basa en la idea de que, al usuario se le debe presentar, lo antes posible, un prototipo del sistema para la experimentación, retroalimentando con información a los constructores del sistema.

De la situación planteada anteriormente surge la “Necesidad de obtener una especificación de requisitos que pueda ser validada y aceptada o rechazada por los especialistas funcionales del cliente”. Con la elaboración de una “Estrategia para apoyar el proceso de captura, definición y validación de requisitos, utilizando un prototipo de interfaz gráfica”, se pretende dar respuesta a la necesidad anterior.

Para lograr esto se estudiarán los “Procesos de captura, definición y validación de requisitos” así como el “Proceso de obtención del prototipo” y como posible guía para encaminar la solución, los “Procesos de especificación de requisitos que han utilizado como técnica un prototipo de interfaz gráfica”.

Una vez planteada y definida la problemática principal a solucionar resulta conveniente trazar un objetivo general que sería: “Conformar una estrategia para representar los requisitos del software a través de un prototipo navegable y no funcional” y ya de manera más específica se pretende:

- Definir y describir los pasos para obtener el prototipo de interfaz.

- Evaluar (ventajas y limitaciones) la factibilidad del uso del prototipo durante la captura de requisitos del software.
- Enunciar las limitaciones del prototipo y los elementos adicionales necesarios para especificar los requisitos del software.
- Definir el prototipo como punto de partida a la implementación.

Para lograr los objetivos anteriores será necesario cumplimentar las siguientes tareas:

- Estudiar los procesos que se siguen en la fase de análisis para el prototipado.
- Estudiar las técnicas empleadas en la especificación de requisitos.
- Analizar resultados prácticos de uso del prototipo en el desarrollo del SIGEP.

Los métodos teóricos que serán utilizados para darle cumplimiento a estas tareas son los siguientes: Análisis Histórico-Lógico para estudiar la evolución y desarrollo del objeto de estudio de la investigación y Analítico-Sintético, haciendo énfasis en el análisis de las teorías, documentos, entre otros, que permite la extracción de los elementos más importantes para procesar la información y elaborar conclusiones. Se utilizarán además los métodos empíricos de la encuesta como técnica que permitirá a través de un conjunto de preguntas obtener la percepción del encuestado acerca del fenómeno que se investiga.

El logro de los objetivos anteriormente planteados requiere una investigación exhaustiva, y el resultado de la misma será distribuido en tres capítulos:

Capítulo 1: Previsto para elaborar una “Fundamentación teórica” que introduzca un grupo de conceptos que ayudarán a la comprensión de la materia en cuestión. Temas como la captura de requisitos, la definición y validación de los mismos, las técnicas empleadas en estos procesos así como el prototipo y la interfaz gráfica, serán tratados en este capítulo.

Capítulo 2: Ya en este capítulo entrando más en materia se verá el “Uso del prototipo en el proyecto SIGEP”, los pasos que se siguieron hasta obtenerlo, herramientas utilizadas en su construcción, así como la interfaz gráfica a implementar, siendo esta el punto concluyente del ciclo de vida del prototipo.

Capítulo 3: En este capítulo ya se analizarán los resultados, las ventajas y limitaciones que aportó el uso del prototipo, así como valoración de encuestas realizadas al grupo de diseñadores de software del proyecto.

Anexos: Aquí se incluirá el prototipo como sistema navegable así como una descripción textual breve de sus funcionalidades, las pautas para el diseño de la interfaz, el código CSS empleado para darle estilo a la interfaz y como resultado concluyente la interfaz obtenida.

# **CAPÍTULO 1: Fundamentación teórica**

## **1.1 Introducción**

El proceso de desarrollo de software es amplísimo por la variedad de aspectos que en el mismo influyen, desde la motivación hasta la tecnología y aunque hoy en día se cuenta con una larga lista de metodologías que han sido creadas para orientarlo, no resulta precisamente fácil entenderlo y mucho menos gestionarlo. Este capítulo enfocará la atención principalmente en los procesos de extracción, representación y validación de requisitos así como el proceso de obtención del prototipo de interfaz de usuario, aportando una base teórica que posibilitará la conformación de una estrategia para vincularlos en capítulos posteriores cuando se hable del desenvolvimiento de los mismos en el proyecto SIGEP. De igual modo se tratará el tema la interfaz gráfica de usuario, punto culminante del ciclo de vida del prototipo.

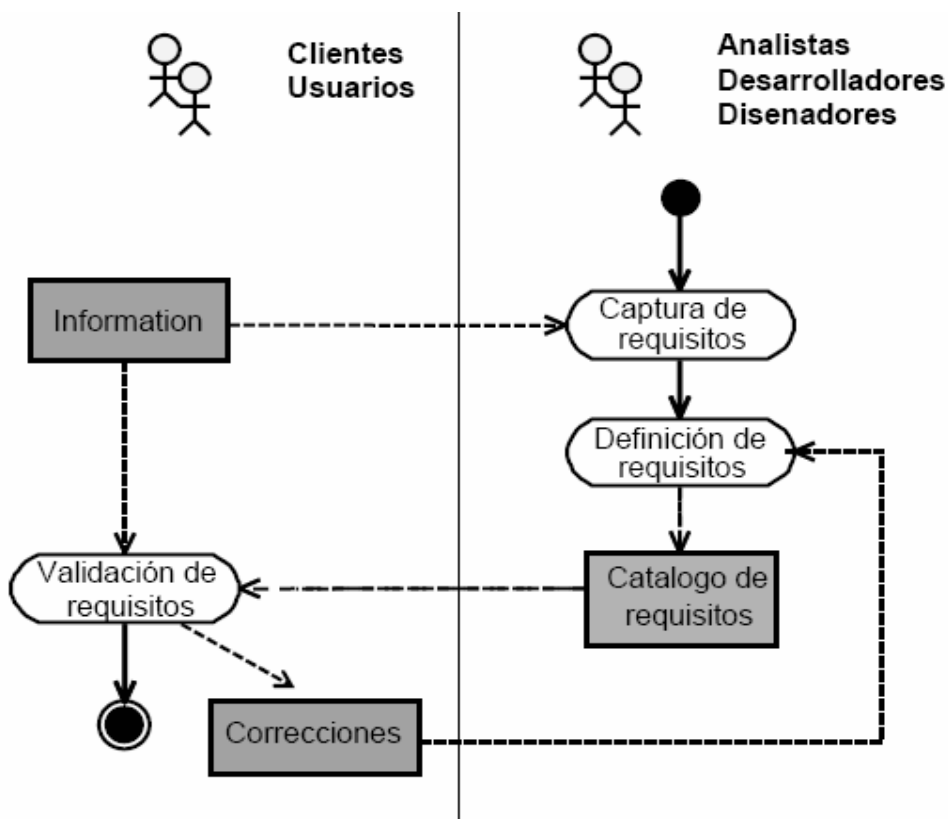
## **1.2 Ingeniería de Requisitos**

Una de las tareas más difíciles al crear sistemas de software consiste en implementar lo que realmente resuelve las necesidades del usuario. Aquellos proyectos construidos con requerimientos incompletos o difíciles de entender suelen terminar con un alcance limitado o distinto a lo que se deseaba o bien caen en un círculo vicioso de retrabajo, sumando los retrasos que elevan considerablemente el costo de desarrollo. Es por ello que La Ingeniería de Requisitos es una de las fases más significativas dentro del ciclo de vida de un proyecto software, y del buen desenvolvimiento de cada una de sus actividades (identificación, representación y validación de requisitos) depende gran parte del éxito del proyecto. (Ver Fig 1)

A pesar de la importancia que tiene la Ingeniería de Requerimientos, ha costado mucho que se le preste la atención adecuada a esta actividad. Aún quedan muchos desafíos que deben ser mejorados, tales como la integración de requerimientos funcionales y no funcionales, la evaluación de especificaciones alternativas, la formalización de la Especificación de Requerimientos de Software (SRS), entre otras.

Cada actividad y técnica de la IR utilizada individualmente, dará diversas soluciones para diferentes proyectos, pues cada método y técnica ofrece disímiles soluciones ante un problema, la técnica más apropiada para cada actividad dependerá del proyecto que esté desarrollándose.

**Fig 1.1 Actividades que contempla la Ingeniería de Requisitos**



### 1.2.1 ¿Qué son los requisitos?

Un requisito o requerimiento es:

- Una condición o capacidad requerida por un usuario para resolver un problema o alcanzar un objetivo.
- Una condición o capacidad que debe ser poseída por un sistema o componente del sistema para satisfacer un contrato, estándar, especificación, u otro documento formalmente impuesto.



Una representación documentada de una condición o capacidad como las descritas en los dos casos anteriores. (IEEE-Std-1233, 1998)

La IEEE del 1983 define los requisitos como: “Una condición o capacidad necesaria para resolver un problema o alcanzar un objetivo”.

## **1.2.2 ¿Qué es la captura de requisitos?**

En el proceso de desarrollo de un sistema, el equipo de desarrollo se enfrenta al problema de la identificación de requisitos. La definición de las necesidades del sistema es un proceso complejo pues en él hay que identificar los requisitos que el sistema debe cumplir para satisfacer las necesidades de los usuarios finales y de los clientes.

La captura de requisitos es la actividad mediante la que el equipo de desarrollo de un sistema de software extrae, de cualquier fuente de información disponible, las necesidades que debe cubrir dicho sistema. (IEEE-Std-1233, 1998)

## **1.2.3 Técnicas usadas para capturar requisitos**

El proceso de captura de requisitos puede resultar complejo, principalmente si el entorno de trabajo es desconocido para el equipo de analistas, y depende mucho de las personas que participen en él. Por la complejidad que todo esto puede implicar, la Ingeniería de Requisitos ha trabajado desde hace años en desarrollar prácticas que permitan hacer este proceso de una forma más eficiente y precisa. La aplicación de las técnicas adecuadas de captura de requisitos en los primeros momentos del ciclo de desarrollo supone una mejora considerable en términos de calidad y reducción de costes en la construcción del producto final.

Hay muchas técnicas para identificar los requerimientos, incluyendo las siguientes: (IEEE-Std-1233, 1998)

### **1.2.3.1 Lluvia de Ideas (Brainstorm)**

Es una técnica de reuniones en grupo cuyo objetivo es que los participantes muestren sus ideas de forma libre (María José Escalona, 2002). Consiste en la mera acumulación de ideas y/o información

sin evaluar las mismas. El grupo de personas que participa en estas reuniones no debe ser muy numeroso (máximo diez personas), una de ellas debe asumir el rol de moderador de la sesión, pero sin carácter de controlador. Como técnica de captura de requisitos es sencilla de usar y de aplicar. Promueve la introducción de principios creativos Además suele ofrecer una visión general de las necesidades del sistema, pero normalmente no sirve para obtener detalles concretos del sistema, por lo que suele aplicarse en los primeros encuentros.

### **1.2.3.2 Entrevistas y cuestionarios**

Las entrevistas y cuestionarios se emplean para reunir información proveniente de personas o de grupos. Durante la entrevista, el analista conversa con el encuestado; el cuestionario consiste en una serie de preguntas relacionadas con varios aspectos de un sistema. Por lo común, los encuestados son usuarios de los sistemas existentes o usuarios en potencia del sistema propuesto. En algunos casos, son gerentes o empleados que proporcionan datos para el sistema propuesto o que serán afectados por él. Con frecuencia, se utilizan preguntas abiertas para descubrir sentimientos, opiniones y experiencias generales, o para explorar un proceso o problema. Este tipo de preguntas son siempre apropiadas, además que ayudan a entender la perspectiva del afectado y no están influenciadas por el conocimiento de la solución. El éxito de esta técnica combinada, depende de la habilidad del entrevistador y de su preparación para la misma. Los analistas necesitan ser sensibles a las dificultades que algunos entrevistados crean durante la entrevista y saber cómo tratar con problemas potenciales. Asimismo, necesitan considerar no sólo la información que adquieren a través del cuestionario y la entrevista, sino también, su significancia (María José Escalona, 2002)

### **1.2.3.3 Comparación de terminología**

Uno de los problemas que surge durante la elicitación de requisitos es que usuarios y expertos no llegan a entenderse debido a problemas de terminología. Esta técnica es utilizada en forma complementaria a otras técnicas para obtener consenso respecto de la terminología a ser usada en el proyecto de desarrollo. Para ello es necesario identificar el uso de términos diferentes para los mismos conceptos (correspondencia), misma terminología para diferentes conceptos (conflictos) o cuando no hay concordancia exacta ni en el vocabulario ni en los conceptos (contraste). (María José Escalona, 2002)

#### **1.2.3.4 Prototipos**

Consiste en iterar en la fase de análisis tantas veces como sea necesario, mostrando prototipos al usuario para que pueda indicar de forma más eficiente los requisitos del sistema. La iteración finalizará cuando el usuario de el visto bueno al prototipo. Esto permite que al mismo tiempo el desarrollador entienda mejor lo que se debe hacer y el cliente vea resultados a corto plazo. Se utiliza para identificar los requisitos detallados de entrada, procesamiento o salida.

#### **1.2.3.5 Casos de Uso**

Aunque inicialmente se desarrollaron como técnica para la definición de requisitos, algunos autores proponen casos de uso como técnica para la captura de requisitos. Los casos de uso permiten mostrar el contorno (actores) y el alcance (requisitos funcionales expresados como casos de uso) de un sistema. Un caso de uso describe la secuencia de interacciones que se producen entre el sistema y los actores del mismo para realizar una determinada función. La ventaja esencial de los casos de uso es que resultan muy fáciles de entender para el usuario o cliente, sin embargo carecen de la precisión necesaria si no se acompañan con una información textual o detallada con otra técnica. (María José Escalona, 2002)

### **1.2.4 Definición de requisitos**

La definición de requisitos se refiere a la identificación de las funciones básicas del componente de programación en un sistema de equipo/personal/programación. Se pone atención en las funciones y restricciones bajo las cuales se deben desarrollar. De la definición de requisitos se obtiene una especificación que describe el ambiente de procesamiento, las funciones requeridas de los programas, restricciones de configuración sobre los programas, manejo de excepciones, cambios y modificaciones factibles, así como los criterios de aceptación de producto de programación.

### **1.2.5 Técnicas para Definición de requisitos**

También para la actividad de definición de requisitos en el proceso de Ingeniería de Requerimientos hay un gran número de técnicas propuestas. A continuación se describen brevemente algunas de ellas.

### **1.2.5.1 Lenguaje natural**

Resulta una técnica muy ambigua para la definición de los requisitos. Consiste en definir los requisitos en lenguaje natural sin usar reglas para ello. A pesar de que son muchos los trabajos que critican su uso, es cierto que a nivel práctico se sigue utilizando.

### **1.2.5.2 Glosario y ontologías**

La diversidad de personas que forman parte de un proyecto software hace que sea necesario establecer un marco de terminología común. Esta necesidad se vuelve más patente en los sistemas de información Web, puesto que el equipo de desarrollo en ellas suele ser más interdisciplinario. Por esta razón son muchas las propuestas que abogan por desarrollar un glosario de términos en el que se recogen y definen los conceptos más relevantes y críticos para el sistema. En esta línea se encuentra también el uso de ontologías, en las que no sólo aparecen los términos, sino también las relaciones entre ellos.

### **1.2.5.3 Plantillas o patrones**

Esta técnica, recomendada por varios autores (Durán A., 1999) (Escalona, 2002), tiene por objetivo el describir los requisitos mediante el lenguaje natural pero de una forma estructurada. Una plantilla es una tabla con una serie de campos y una estructura predefinida que el equipo de desarrollo va cumplimentando, usando para ello el lenguaje del usuario. Las plantillas eliminan parte de la ambigüedad del lenguaje natural al estructurar la información; cuanto más estructurada sea ésta, menos ambigüedad ofrece. Sin embargo, si el nivel de detalle elegido es demasiado estructurado, el trabajo de rellenar las plantillas y mantenerlas, puede ser demasiado tedioso.

### **1.2.5.4 Escenarios**

La técnica de los escenarios consiste en describir las características del sistema a desarrollar mediante una secuencia de pasos (Lu&Yu). La representación del escenario puede variar dependiendo del autor. Esta representación puede ser casi textual o ir encaminada hacia una representación gráfica en forma de diagramas de flujo (Weidenhaupt, 1999). El análisis de los escenarios, hechos de una forma u otra, pueden ofrecer información importante sobre las necesidades funcionales de sistema (Lowe, 1999)

### **1.2.5.5 Casos de uso**

Como técnica de definición de requisitos es como más ampliamente han sido aceptados los casos de uso. Actualmente se ha propuesto como técnica básica del proceso RUP (Kruchten, 1998). Sin embargo, son varios los autores que defienden que pueden resultar ambiguos a la hora de definir los requisitos (A, 2001), por lo que hay propuestas que los acompañan de descripciones basadas en plantillas o de diccionarios de datos que eliminen su ambigüedad.

### **1.2.5.6. Lenguajes Formales**

Otro grupo de técnicas que merece la pena resaltar como extremo opuesto al lenguaje natural, es la utilización de lenguajes formales para describir los requisitos de un sistema. Las especificaciones algebraicas como ejemplo, de técnicas de descripción formal, han sido aplicadas en el mundo de la Ingeniería de Requisitos desde hace años. Sin embargo, resultan muy complejas en su utilización y para ser entendidas por el cliente. El mayor inconveniente es que no favorecen la comunicación entre cliente y analista.

### **1.2.5.7 Aproximación prototipo**

Es habitual que en un proyecto software no se identifiquen los requisitos detallados de entrada, procesamiento o salida. Esta aproximación consiste en realizar la fase de definición de requisitos del sistema en base a estos tres factores:

- Un alto grado de iteración.
- Un muy alto grado de participación del usuario.
- Un uso extensivo de prototipos.

Las premisas clave de esta aproximación son:

- Que los prototipos constituyen un mejor medio de comunicación que los modelos en papel.
- Que la iteración es necesaria para canalizar, en la dirección correcta, el proceso de aprendizaje. Esta aproximación se enfoca a mejorar la efectividad del proceso de desarrollo y no a mejorar la eficacia de ese proceso.

## **1.2.6 Validación de requisitos**

Los requisitos una vez definidos necesitan ser validados. La validación de requisitos tiene como misión demostrar que la definición de los requisitos define 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. (María José Escalona, 2002)

## **1.2.7 Técnicas se usan para validar requisitos**

Existen un grupo de técnicas que se aplican para validar los requisitos entre ellas:

### **1.2.7.1 Revisiones**

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.

### **1.2.7.2 Auditorías**

La revisión de la documentación con esta técnica consiste en un chequeo de los resultados contra una checklist (lista de chequeo) predefinida o definida a comienzos del proceso, es decir, sólo una muestra es revisada.

### **1.2.7.3 Matrices de trazabilidad**

Esta técnica consiste en marcar los objetivos del sistema y chequearlos contra los requisitos del mismo (Durán A., 1999). Es necesario ir viendo qué objetivos cubre cada requisito, de esta forma se podrán detectar inconsistencias u objetivos no cubiertos.

### **1.2.7.4 Prototipos**

Esta técnica se utiliza para que el cliente valore si el prototipo tiene las capacidades de desempeño que la aplicación requiere. Explora las características que puedan poner en peligro su utilización futura, evalúa su funcionalidad y estabilidad. En función de las opiniones obtenidas de los clientes se pueden cambiar e incluso desechar algunas ideas para desarrollos futuros pero el objetivo principal de esta

técnica es lograr una aceptación de los requisitos funcionales. Esta técnica tiene el problema de que el usuario debe entender que lo que está viendo es un prototipo y no el sistema final.

## **1.3 Ingeniería de Requisitos desde el punto de vista metodológico**

### **1.3.1 RUP**

La metodología RUP, Rational Unified Process, es una de las más importantes para alcanzar un grado de certificación en el desarrollo del software. Existe un extenso grupo de motivos por los cuales resulta tentativa la idea de adoptar RUP pero se centrará la atención particularmente en sus fundamentos relacionados con las etapas que contempla la Ingeniería de Requisitos y el uso del prototipo.

Cuando se hace un sistema informático una de las etapas cruciales por el efecto que tiene en el resto del ciclo de desarrollo, es captar y mostrar de modo apropiado los requisitos.

RUP establece que en la captura de requisitos es donde se encuentran, se les da una prioridad y se estructuran los casos de uso (fragmentos de funcionalidad del sistema que proporcionan al usuario un resultado importante).

El RUP establece que luego de haber desarrollado un modelo de casos de uso que especifica qué usuarios hay y para qué necesitan utilizar el sistema, se proceda a prototipar la interfaz de usuario, de forma tal que se le permita al usuario llevar a cabo los casos de uso de manera eficiente. RUP propone comenzar con los casos de uso e intentar discernir qué se necesita de las interfaces para habilitar los casos de uso para cada actor.

Se ve el prototipo como una forma de direccionarse con el objetivo de reducir riesgo dado que se puede utilizar para medir la estabilidad de la tecnología principal. Permite además entender los requerimientos, ver una aproximación de apariencia que tendrá el producto así como medir su usabilidad.

### **1.3.2 Design-driven Requirements Elicitation**

Design-driven Requirements Elicitation es parte del proceso design-driven que proponen Lowe y Eklund (2002) para el desarrollo de aplicaciones en el entorno Web. 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” (mejores prácticas) 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. (Díez A, 2001)

## **1.4 El prototipo**

Luego de haber tratado estos procesos que están comprendidos en la Ingeniería de Requisitos así como el tratamiento metodológico a estos temas y habiendo observado un denominador común en los tres: el tema del prototipo, se procederá a abundar el tema de este como elemento común en la identificación, definición y validación de requisitos.

### **1.4.1 ¿Qué es el prototipo?**

Un prototipo es una representación limitada del diseño de un producto que permite a las partes responsables de su creación experimentar, probarlo en situaciones reales y explorar su uso (Alberto la Calle, 2006). Un prototipo es un modelo (representación, demostración o simulación) fácilmente ampliable y modificable de un sistema planificado, probablemente incluyendo su interfaz y su funcionalidad de entradas y salidas (Walter Maner , 1997). Un prototipo puede ser cualquier cosa, desde un trozo de papel con dibujos sencillos hasta un complejo software. Es un modelo a escala o facsímil 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 .



## 1.4.2 Tipos de prototipo

Baja Fidelidad vs. Alta Fidelidad

- Baja Fidelidad: Conjunto de dibujos (por ejemplo, una presentación de escenarios) que constituye una maqueta estática, no computarizada y no operativa de una interfaz de usuario para un sistema en planificación.
- Alta Fidelidad: Conjunto de pantallas que proporcionan un modelo dinámico, computarizado y operativo de un sistema en planificación.

Exploratorio vs. Experimental vs. Operacional

- Exploratorio: Prototipo no reutilizable utilizado para clarificar las metas del proyecto, identificar requerimientos, examinar alternativas de diseño o investigar un sistema extenso y complejo.
- Experimental: Prototipo utilizado para la validación de especificaciones de sistema
- Operacional: Prototipo iterativo que es progresivamente refinado hasta que se convierte en el sistema final.

Horizontal vs. Vertical

- Horizontal: Prototipo que modela muchas características de un sistema pero con poco detalle. Dicho detalle alcanzará una profundidad determinada, va a resultar especialmente útil en las etapas tempranas de diseño y tiene como objetivo el test del modo de interacción global, al contemplar funciones comunes que el usuario va a utilizar frecuentemente.
- Vertical: Prototipo que modela pocas características de un sistema pero con mucho detalle. Va a resultar especialmente útil en etapas más avanzadas del diseño y tiene como objetivo el test de detalles del diseño
- Diagonal: Prototipo horizontal hasta un cierto nivel, a partir del cual se puede considerar vertical

Global vs. Local

- Global: Prototipo del sistema completo. Prototipo horizontal expandido que modela una gran cantidad de características y cubre un amplio rango de funcionalidades. Va a resultar muy útil a lo largo de todo el proceso de diseño.
- Local: Prototipo de un único componente o característica del sistema de usabilidad crítica. Va a resultar de utilidad en algunas etapas específicas del proceso de diseño.

Existen principalmente dos tipos de prototipos:

- Prototipo rápido (o concept prototipe): El prototipado rápido es un mecanismo para lograr la validación pre-compromiso. Se utiliza para validar requerimientos en una etapa previa al diseño

específico. En este sentido, el prototipo puede ser visto como una aceptación tácita de que los requerimientos no son totalmente conocidos o entendidos antes del diseño y la implementación. El prototipo rápido puede ser usado como un medio para explorar nuevos requerimientos y así ayudar a "controlar" su constante evolución.

- Prototipo evolutivo: Desde una perspectiva diferente, todo el ciclo de vida de un producto puede ser visto como una serie incremental de detallados prototipos acumulativos. Tradicionalmente, el ciclo de vida está dividido en dos fases distintas: desarrollo y mantenimiento. La experiencia ha demostrado que esta distinción es arbitraria y va en contra de la realidad ya que la mayor parte del costo del software ocurre después de que el producto se ha entregado. El punto de vista evolutivo del ciclo de vida del software considera a la primera entrega como un prototipo inicial en el campo. Modificaciones y mejoras subsecuentes resultan en nuevas entregas de prototipos más maduros. Este proceso continúa hasta que se haya desarrollado el producto final. La adopción de esta óptica elimina la distinción arbitraria entre desarrollo y mantenimiento, resultando en un importante cambio de mentalidad que afecta las estrategias para la estimación de costos, enfoques de desarrollo y adquisición de productos.

### **1.4.3 Objetivos del prototipo**

Siempre se debe establecer cuál es su objetivo pues el prototipo puede ser útil en diferentes fases del proyecto.

En la fase de análisis se utiliza para obtener los requerimientos del usuario y en la fase de diseño se usa para ayudar a evaluar muchos aspectos de la implementación seleccionada, o sea verificar la factibilidad del diseño del sistema.

En la fase de Análisis de un proyecto, su principal propósito es obtener y validar los requerimientos esenciales, manteniendo abiertas, las opciones de implementación. Esto implica que se debe tomar los comentarios de los usuarios, pero se debe regresar a sus objetivos para no perder la atención. En la fase de Diseño, su propósito, basándose en los requerimientos previamente obtenidos, es mostrar las ventanas, su navegación, interacción, controles y botones al usuario y obtener una retroalimentación que nos permite mejorar el Diseño de Interfaz.

#### **1.4.4 Finalidad del prototipo**

El prototipo es una aplicación que funciona. La finalidad del prototipo es probar varias suposiciones formuladas por analistas y usuarios con respecto a las características requeridas del sistema. Los prototipos se crean con rapidez, evolucionan a través de un proceso interactivo y tienen un bajo costo de desarrollo.

#### **1.4.5 Condiciones para emplear prototipo.**

Las condiciones para aplicar prototipos son:

- No conocer los requerimientos.
- Evaluar los requerimientos.
- Costos altos de inversión.
- Alto riesgo.
- Nueva tecnología.

#### **1.4.6 Etapas del método con prototipos.**

Las etapas del método con prototipos son:

- Identificación de requerimientos conocidos .
- Desarrollo de un modelo de trabajo.
- Participación del usuario.
- Revisión del prototipo.
- Iteración del proceso de refinamiento.

#### **1.4.7 Estrategias para uso de prototipos**

Las estrategias para el desarrollo de prototipos son:

- Prototipos para pantallas: El elemento clave es el intercambio de información con el usuario.
- Prototipos para procedimientos de procesamiento: El prototipo incluye solo procesos sin considerar errores.

- Prototipos para funciones básicas: Sólo se desarrolla el núcleo de la aplicación, es decir solo los procesos básicos. (Lic. Domingo F. Donadello)

#### **1.4.8 Papel del prototipo**

Son útiles para comunicar, discutir y definir las ideas entre los diseñadores y las partes responsables. Los prototipos responden a preguntas y apoyan el trabajo de los diseñadores probando ideas, clarificando requisitos o definiendo alternativas (Alberto la Calle, 2006). Es muy útil hacer uso del prototipo en la fase de análisis para prevenir algunos problemas que pudieran presentarse después en la fase de desarrollo, con el prototipo pueden aparecer ciertas interrogantes que respondidas a tiempo evitarían un atraso en el proceso de desarrollo del software. El prototipo ayuda a comprender el funcionamiento del sistema, a explorar sus funcionalidades y a que el usuario tenga una idea general de lo que va a hacer el producto para que de esta forma se pregunte si es eso realmente lo que quiere y no ocurra que al final el cliente quede insatisfecho y la aplicación no sea lo que realmente necesita.

Brinda aportes de comunicación entre desarrolladores y clientes, logra un incremento de la participación constructiva del usuario, contribuye con el aseguramiento de la calidad, se reúne la mayor cantidad de requerimientos válidos, se hace una mejor gestión de las peticiones de cambios. (Walter Maner , 1997)

#### **1.4.9 ¿Cuándo se prototipa?**

Antes de comenzar, para mostrar el concepto a la directiva. Al comienzo para reunir los requerimientos iniciales de usuario. Después de empezar, para validar la evolución de los requerimientos de usuario. En las etapas intermedias, para validar las especificaciones del sistema. Entre las etapas intermedias y finales, para preparar a los usuarios y en las etapas finales para explorar soluciones a problemas de diseño o usabilidad específicos. (Walter Maner , 1997)

El prototipo actúa en varias fases del proceso de desarrollo del software y en todas es importante por las trabas que soluciona en cada una de ellas. Es importante resaltar la necesidad de empezar a implementarlo así sea con un nivel menos complejo, desde etapas tempranas, es decir, desde la captura de requisitos para comenzar a explorar las posibles dificultades que pudieran presentarse y que se llegue a la fase de desarrollo con ideas lo más claras posibles respecto al sistema a implementar.

### **1.4.10 Normas que respaldan el uso del prototipo**

La norma IEEE 830 en su edición de 1998 de Guía para el Desarrollo de Especificaciones de Requerimientos de Sistemas define al prototipo un modelo experimental del sistema o parte del sistema, ya sea funcional o no funcional, usado para obtener retroalimentación de los usuarios, para ser capaces de mejorar y especificar una interfaz de usuario compleja, para realizar estudios de factibilidad, o para identificar requerimientos.

El estándar ISO 13407 "Human centred design for interactive systems" (diseño centrado en el humano para sistemas interactivos), provee de un marco teórico para el desarrollo de actividades centradas en el usuario. Desarrolla prototipos y éstos son probados, cambiados o sustituidos con el feedback de los usuarios reales.

## **1.5 Las interfaces de usuario**

En el contexto del proceso de interacción persona-ordenador, la interfaz gráfica de usuario (IGU), es el artefacto tecnológico de un sistema interactivo que posibilita, a través del uso y la representación del lenguaje visual, una interacción amigable con un sistema informático.

La interfaz gráfica de usuario (en inglés Graphical User Interface, GUI) es un tipo de interfaz de usuario que utiliza un conjunto de imágenes y objetos gráficos (iconos, ventanas, tipografía) para representar la información y acciones disponibles en la interfaz. Habitualmente las acciones se realizan mediante manipulación directa para facilitar la interacción del usuario con la computadora.

### **1.5.1 Herramientas útiles en la elaboración de Interfaces de Usuario**

Para arreglar imágenes existen varios programas. En estos últimos años se han impuesto algunas soluciones específicamente enfocadas a la preparación de imágenes para la Web. Entre los programas usados con este fin se destacan Fireworks, LiveMotion (antes Imagestyler), Imageready (incluido en Photoshop) Y aparte de estos, Xara: una herramienta potente de dibujo vectorial y preparación de imágenes para la Web. Otra solución son los programas que crean contenido Flash (Flash y Director.) El programa de retoque gráfico mas empleado en el mundo es Photoshop, que es el Editor Estándar de imágenes raster a nivel internacional, brinda fidelidad tanto en el color, como topográficamente etcétera.

### 1.5.1.1 Photoshop

Photoshop es utilizado en todo el mundo por diseñadores, ilustradores, fotógrafos, videógrafos y autores multimedia para escanear imágenes, trabajar con fotografías digitales o imágenes generadas por ordenador, así como crear gráficos para web.

Photoshop nos ofrece cientos de herramientas de una impresionante calidad, con funciones y capacidades que van desde los retoques más básicos a fin de mejorar el color o la luminosidad de una imagen, hasta complicados montajes y transformaciones con lo que modificar completamente el aspecto de una foto. Todo ello sin olvidar el impresionante abanico de filtros y efectos especiales que también incluye Photoshop.

Junto a Adobe Photoshop podemos encontrar Adobe ImageReady, un paquete de herramientas para trabajar con gráficos web. Ofrece nuevas capacidades de optimización de gráficos para web, herramientas para máscaras, galerías web, historial de pinceles y mucho más. (2005)

Con Photoshop encontramos la solución óptima para la creación de trabajos profesionales tanto en impresión como en Web. Se han introducido mejoras que aumentan el control y productividad del programa, sin olvidar las funciones que han hecho de éste un programa imprescindible para todo aquel que trabaje con imágenes.

Photoshop CS2, la nueva versión de Adobe Photoshop, el programa de edición digital por excelencia, es toda una herramienta de producción para realce digital, retoque, composición fotográfica y animación.

### 1.5.1.2 Dreamweaver

Macromedia Dreamweaver es un editor de HTML visual. Dreamweaver hace muy fácil el crear complejas páginas Web dinámicas, con la conocida técnica de "arrastrar y soltar", permitiendo que los diseñadores puedan crear entornos Web y animaciones sofisticadas sin tener que escribir una sola línea de código.

Una de las características del programa es que se pueden optimizar las páginas para las diferentes versiones de los navegadores. Dreamweaver no modifica el código fuente, haciendo fácil el poder cambiar entre Dreamweaver y un editor de código no visual. Algunas otras características incluyen: un editor de imagen integrado, diferentes colores para la sintaxis HTML, soporte para posicionamiento absoluto, poder hacer cambios por todas las páginas usando elementos comunes, plantillas, e interfaz personalizado. (Terra, 2006)

Dreamweaver MX 2004 añade al éxito de Dreamweaver MX la integración de herramientas de presentación visuales, características de desarrollo de aplicaciones, y soporte de la edición de código

dentro de un entorno de plataforma independiente. Al soportar la mayoría de las tecnologías de servidor y HTML, Dreamweaver MX 2004 abre la potencia del desarrollo abierto, basado en estándares para la comunidad web mientras mejora las herramientas básicas que los profesionales necesitan. Dreamweaver MX 2004 también incluye SecureFTP, la función dinámica de validación de navegadores, la incorporación de la edición de gráficos, integración racionalizada con Microsoft Word y Excel, y soporte actualizado para ASP.NET y PHP.

Macromedia Dreamweaver MX 2004 forma parte de la familia MX 2004, que comparte nuevas características como la definición de sitios, la gestión de archivos, inspector de propiedades etc.

Dreamweaver MX 2004 incluye facilidad, controles de diseño basados en estándares para asegurar el diseño de alta calidad.

## **1.5.2 Tecnologías**

### **1.5.2.1 HTML**

HTML, un estándar propuesto por el Consorcio W3C, que pretende conseguir que todos los documentos Web sean compatibles en cualquier navegador (no solamente en ordenadores, sino también en cualquier dispositivo). El lenguaje HTML esta limitado a la hora de aplicarle forma a un documento.

Importante resaltar esta primera característica, que con HTML no existirán problemas por el tipo de navegador que tenga implementado el usuario, tampoco se limita por el tipo de tecnología, hardware etcétera. Usar HTML para la Web constituye una garantía importante teniendo en cuenta transformaciones que sufre la informática con el desarrollo acelerado de las tecnologías ( Murugarren, Joaquín Gracia, . 2004)

### **1.5.2.2 CSS**

CSS son las siglas de "Cascade StyleSheet", y especifica la forma del diseño de los documentos, tanto XHTML como HTML. El entorno de diseño completo construido sobre CSS, permite un desarrollo rápido y más eficiente de código puro, sitios profesionales con diseños sofisticados. La validación dinámica de los múltiples navegadores comprueba automáticamente las etiquetas y reglas CSS para la compatibilidad con los navegadores más importantes, resolviendo el más común quebradero de cabeza de los desarrolladores: la compatibilidad de diseños en los navegadores.

### **1.5.2.3 Java Script**

Javascript es utilizado para crear pequeños programitas encargados de realizar acciones dentro de una página Web. Con Javascript podemos crear efectos especiales en las páginas y definir interactividades con el usuario. Entre las acciones típicas que se pueden realizar en Javascript tenemos dos vertientes. Por un lado los efectos especiales sobre páginas Web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo. Por el otro, javascript nos permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que podemos crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo (Álvarez, Miguel, 2002).

## **1.6 Conclusiones**

Hasta aquí se ha logrado establecer la base teórica necesaria para facilitar la comprensión de la propuesta que se mostrará en el capítulo posterior. Se ha logrado cierta familiarización con los procesos de extracción, representación y validación de requisitos viendo conceptualmente en qué consisten y las técnicas que se utilizan para cada uno de ellos. Se muestra el prototipo como técnica para especificar los requerimientos y la interfaz gráfica de usuario como artefacto resultante del prototipo.



## **CAPÍTULO 2: El uso del prototipo en el proyecto SIGEP**

### **2.1 Introducción**

“Antes es mucho mejor que después”, la veracidad de esta frase en el marco del contexto que se está abordando, cobra veracidad tras 20 años de experiencia respecto a la ingeniería de la usabilidad en software, que demuestran que en proyectos la mayor mejora en la experiencia del usuario viene de una recogida de datos realizada lo antes posible. Los prototipos son un método de usabilidad, no se conocen cuáles serán las tecnologías populares de creación de interfaz dentro de 20 años, pero las técnicas de evaluación temprana con prototipos seguirán teniendo un valor. Se ha decidido dedicar el presente capítulo a abordar el tema de la conformación del prototipo en fases tempranas de desarrollo en el proyecto SIGEP tomando como referencia los motivos anteriormente expuestos para mitigar los daños mencionados. Se pretende describir el proceso de captura de requisitos y conformación del prototipo desde esta etapa con una secuencia de pasos para conformar una estrategia que sirva para guiar el desarrollo con prototipado desde etapas tempranas.

### **2.2 Especificación de requisitos**

En la especificación de requerimientos se llevan a cabo las actividades de captura, definición y validación de requisitos. El objetivo es captar y mostrar de modo apropiado los requisitos de usuario. La etapa de captura de requisitos requiere de la ejecución de varios procesos: Modelación del negocio, identificación de actores y casos de uso que interactuarían con el sistema, definición y descripción de la funcionalidad, prototipado para mostrar los requisitos y validación de los mismos.

### **2.2.1 Modelación del negocio**

Se realiza para alcanzar un entendimiento común acerca del funcionamiento del sistema penitenciario por parte del equipo de desarrollo y los interesados en el proyecto.

Para alcanzar esta meta es necesario, primero, determinar qué procesos de la organización están listos para avanzar en su comprensión, desarrollando la Evaluación de la Organización y; segundo, modelar los procesos de la organización, para lo cual se realiza la Comprensión del Negocio. En la Evaluación de la Organización se identifican los procesos de la organización que están listos para avanzar en su comprensión y en la Comprensión del Negocio se modelan los procesos de negocio para alcanzar un entendimiento común respecto a ellos entre el equipo de desarrollo y los interesados.

### **2.2.2 Definición de estructuración del sistema**

De estudios preliminares realizados se tenían propuestas de subsistemas para el SIGEP, proposiciones que a lo largo de este proceso se fueron reafirmando o refinando. De la modelación del negocio surgieron nuevos módulos, algunos que se habían propuesto desaparecieron, otros cambiaron de nombre. La comprensión del negocio le facilita al analista decidir con mayor base los módulos que incluye cada subsistema, este es un proceso de aproximaciones que va desde lo general a lo particular. Una vez definidos los módulos se esclarecen los objetivos que se persiguen con este, es decir, el motivo por el cual fue creado.

Las metas por las cuales se concibieron los módulos conducen a la necesidad de especificar el conjunto de acciones que se van a llevar a cabo dentro de ese módulo y las personas que interactúan en cada una de esas acciones, es por ello que luego de modelado el negocio y llegado a este punto en que se reafirman los módulos de cada subsistema, se procede a definir el sistema para identificar los trabajadores del negocio que pueden beneficiarse del sistema para desarrollar sus actividades, se enmarcan estas actividades por sus objetivos dentro del módulo que se asocie según sus propósitos junto a los trabajadores que se benefician del desenvolvimiento de cada actividad. Estos trabajadores se capturan como actores y las actividades o procesos como Casos de Uso y Salidas del Sistema.

Los CU, una vez identificados necesitan ser descritos.

Inicialmente se intentó describir textualmente algunos de los CU pero el cliente presentaba problemas para identificar funcionalidades en estas descripciones y describir funcionalmente qué quería que hiciera el sistema, que necesitaba que este lograra etc. Para ello se procedió a confeccionar un prototipo que le ayudara al cliente a reconocer visualmente sus necesidades, las características para su sistema etc.

### **2.2.2 Consideraciones previas al prototipado**

En la Dirección General de Servicios Penitenciarios (DGSP), los procesos no están bien definidos y existen continuos cambios, lo cual dificulta la identificación y estabilización de los requisitos del sistema, y esta situación pone en riesgo el éxito del proyecto. Con el propósito de atenuar este conflicto se concibió el desarrollo del software por etapas.

La capacidad para definir paso a paso, iteración tras iteración, con modelado visual y con retroalimentación del RUP (Rational Unified Process) fue lo que hizo que se adoptara como metodología de referencia. El RUP propone la realización de un gran número de actividades que usualmente los proyectos se comprometen a desarrollar pero que no realizan en su totalidad debido a falta de tiempo o a que descubren que no eran necesarios.

En SIGEP, surge la imperiosa necesidad de acelerar de cierta forma el proceso de captura y representación de requisitos, para llegar a validar los mismos cuanto antes y proceder a conformar la interfaz gráfica y abrir paso de esta forma a la implementación, con el objetivo de asegurar en etapas inmediatamente posteriores, un entregable en la fecha pactada. Por ello se recurre a un principio del Modelado Ágil: producir sólo los artefactos y realizar las actividades que tienen un propósito dentro del proyecto, decidiendo así emprender la confección del prototipo de interfaz en que se basaría la validación de requisitos, desde la identificación de los mismos, y comenzar a explotar esta técnica desde esta etapa.

Es necesario resaltar que se llevaron a cabo otras técnicas para extraer los requisitos. Se efectuaron entrevistas para obtener información proveniente de algunos dirigentes en la organización así como de los especialistas funcionales de la parte cliente venezolana, se obtuvieron datos para el sistema propuesto.

Mediante la observación también se percibieron algunas características importantes, tanto relacionadas con el usuario que utilizaría el sistema como con el funcionamiento de los procesos en la entidad, pero de

una simple observación, no pueden extraerse características más profundas y funcionalidades que se perciben únicamente mediante la interacción con algo lo más similar posible al sistema final.

Se hicieron revisiones a la documentación que se maneja en los centros penitenciarios para extraer elementos de dato importantes a manejar con la aplicación.

Todas estas actividades fueron de importancia y se complementaron unas con las otras para apoyar un poco el proceso de extracción de las características a ofrecer en el sistema y las necesidades a solucionar, mas no facilitaban por sí solas la conclusión de dicho proceso porque es una organización en la cual se maneja una gran densidad de información y donde se realiza un número importante de acciones sobre relacionadas con documentación y con otros procesos específicos, el usuario es generalmente una persona que no tiene por qué ser informático por lo que no se le deben mostrar propuestas como casos de uso escritos a un nivel técnico que usualmente no les resulta comprensible.

Todo lo anteriormente expuesto, ligado a la escasa información existente con respecto a las características que debía tener el nuevo sistema para satisfacer las necesidades del usuario, se adjuntó a la situación de la institución (DGSP) para considerar viable el uso de la técnica del prototipo de interfaz para especificar los requerimientos y que el cliente pudiera validar dichos requisitos sin necesidad de leerse una cantidad interminable de casos de uso.

### **2.2.3 Definición y descripción de funcionalidades**

Se pretende ofrecer una definición completa y global de la funcionalidad de operación que se va a tener disponible en el prototipo. Una de las cosas principales al realizar un prototipo es mostrarle al usuario la funcionalidad que tendrá. Es importante que el sistema propuesto tenga la capacidad funcional necesaria para las tareas que los usuarios deben querer llevar a cabo. Esto se presenta con la finalidad de ver si se tiene claro lo que el cliente desea. La funcionalidad responde a los objetivos del sistema y constituye la primera aproximación a los casos de uso.

La descripción permite lograr un acercamiento a la conceptualización del sistema. Se realiza una descripción textual breve de cada una de las funcionalidades y son descritos los elementos de datos así como las acciones que pueden realizarse sobre dichos datos. Se describe brevemente el comportamiento y los mecanismos causales que lo producen. (Ver Anexo 1)

### **2.2.5 Proposición de pantallas preliminares**

Este es un proceso creativo, de análisis, y constituye una propuesta concreta a la solución del problema, es decir, a la funcionalidad como tal. Estas pantallas preliminares incluyen elementos de datos y botones de acción. Los botones de acción (por ejemplo, aceptar, enviar, cancelar, etc.) son uno de los elementos más importantes en las páginas web: muchas de las interacciones entre los usuarios y el sitio se realizan a través de ellos.

Esta proposición de pantallas hechas de manera preliminar fueron llevadas a sesiones de validación con el cliente, es muy importante durante la construcción del prototipo la constante retroalimentación con el cliente. Otro aspecto muy positivo y de gran ayuda en esta etapa fue la impresión en papel de estas pantallas para que fueran entregadas al cliente de modo que el mismo pudiese hacer apuntes sobre ellas. Dichas observaciones son recogidas, analizadas y tomadas en cuenta por el equipo de desarrollo para actividades posteriores.

### **2.2.6 Montaje de pantallas en un “sistema navegable no funcional”**

Una vez elaboradas las propuestas de pantallas preliminares, y tomadas en consideración las anotaciones hechas por el cliente así como sus criterios en las sesiones de validación en que el mismo participa como uno más del equipo, se comienza a trabajar en aras de convertir este grupo de pantallas separadas en un conjunto navegable en que se vincula unas con otras. El término que adoptado para nombrar este grupo de pantallas relacionadas entre sí es “sistema navegable no funcional”, y es definido como: una aplicación web estática en que aparecen las interfaces y los botones no ejecutan ninguna acción que no sea permitir la navegabilidad entre una pantalla y otra.

Para representar los datos y las relaciones entre ellos se elabora un modelo conceptual, o sea que no necesariamente tiene que ser hecha esta representación a través de un diagrama de clases o de entidad relación. Este modelo conceptual es confeccionado en el Mind Manager, un software de productividad diseñado para permitir a equipos visualizar pensamientos, ideas y planificar un ambiente escalable y todavía adaptable y que permite organizarse de una mejor manera, hacer listas y otro tipo de cosas. Fue hecho en un formato tal que pudiera ser comprendido fácilmente para que el usuario pudiera validarlo. Puede servir como punto de partida para pantallas así como del modelo de datos. (Ver Anexo 2)

### **2.2.6.1 Herramientas para confeccionar el prototipo**

Para montar estas pantallas en el sistema navegable no funcional que se definió anteriormente se hizo uso de dos herramientas básicamente, el Dreamweaver y el Visio.

La versión de Dreamweaver utilizada fue el Dreamweaver MX 2004. Macromedia Dreamweaver es un editor de HTML visual que hace muy fácil el crear complejas páginas Web dinámicas. Dreamweaver MX 2004 incluye potentes controles basados en normas para asegurar un diseño de alta calidad con una interfaz de usuario mejorada que aumenta al máximo el espacio de trabajo utilizable, muestra el contexto y el enfoque de manera más clara y es sumamente lógica y fácil de usar. La nueva pantalla de inicio proporciona un acceso rápido a los archivos utilizados recientemente y a recursos para tutoriales. Dreamweaver MX 2004 es la opción profesional para la creación de sitios y aplicaciones web. Resulta conveniente su empleo para confeccionar un prototipado pues para las funciones que se necesita, no es imprescindible leerse ningún manual ni ayuda ya que los componentes necesarios están totalmente visibles, fáciles de encontrar y emplear.

Un diseño de prototipo con Visio (o con otro programa de diseño similar) es directamente no programar nada, no hacer versión "prototipo" sino dibujar las ventanas como si de dicho prototipo se tratara. Se suelen luego incluir en un documento explicando lo que hará cada ventana y cada elemento de la misma, y de esta forma no hay necesidad de escribir código hasta que el cliente valida el prototipo de las pantallas. La versión Visio 2003 aporta un diseño de Interfaces Gráficas de Usuario (GUI) completo y fácil de emplear. Con esta herramienta se pueden diseñar ventanas que cualquiera podría pensar que son de un prototipo ejecutable, cuidadas al detalle, desde los botones de minimizar, cerrar, etc., iconos, toolbars, statusbars, controles...Cabe mencionar la facilidad de uso que no se le puede negar a Microsoft. No hubo problemas para encontrar opciones o parámetros determinados de forma simple e intuitiva aún sin haber leído ni manuales ni ayuda, un aspecto relevante teniendo siempre en cuenta el factor tiempo.

Combinadas ambas herramientas para conformar el sistema del que se hablaba anteriormente resulta una consideración valorable, Visio facilita la creación de formularios, la exportación de páginas Web... y Dreamweaver permite integrar todas estas páginas y lograr la navegabilidad entre ellas.

Obviamente la exportación de páginas web tiene sus limitaciones comparadas con las páginas web reales, pues no permite la entrada de datos a través de los campos de texto etc. No obstante este efecto se contrarresta con un poco de creatividad, agregando elementos adicionales que simulen o expliquen la funcionalidad.

### **2.2.7 Tipo de prototipo**

Hay muchos factores importantes que considerar cuando se planea hacer un prototipo para probarlo con usuarios. Se deben hacer cuidadosas elecciones acerca de la fidelidad, nivel de interactividad y el medio del prototipo. Se desarrolla un prototipo rápido para validar requerimientos en la etapa previa al diseño específico, también usado como un medio para explorar nuevos requerimientos y así ayudar a "controlar" su constante evolución. Un prototipo de fidelidad alta, pues todos los blancos de la página quedaron llenos permitiendo que se vea bien, aunque no con todos los problemas técnicos resueltos, por eso es que se considera un prototipo, con suficientes vínculos funcionando sólo para expresar el sentido del flujo del sitio. Este tipo de prototipo resulta más fácil de entender para los usuarios, y esa es una de las ideas fundamentales. Un prototipo además horizontal, pues a través de él se modelan muchas características del sistema aunque con poco detalle. Global porque el prototipo contempla el sistema completo, de tipo horizontal expandido que modela una gran cantidad de características. Experimental porque es utilizado para la validación de especificaciones de sistema.

### **2.2.8 Documento descriptivo del prototipo**

Como se mencionaba anteriormente no basta con una aplicación web estática digital, sino que resulta conveniente algo más que exprese la funcionalidad que no se percibe con este "sistema navegable no funcional". Es necesario y muy útil un documento que describa este prototipo, o sea, un documento descriptivo por módulo en el cual se describa el mismo. Se incluye también un mapa de navegación y luego se pasa a describir las pantallas referentes a cada una de las funcionalidades. Los elementos de dato son descritos así como los botones de acción, especificando para cada uno el estado en que queda el sistema una vez ejecutadas las acciones asociadas a los mismos. Se considera importante especificar para cada uno de ellos las precondiciones que deben darse para que se ejecuten según su curso normal. Se hace una tabla donde se indica para cada elemento de dato el tipo de dato que soporta y las reglas de validación asociadas. Los tipos de dato no se especifican de acuerdo a un lenguaje de Base de Datos general sino que se asocia a un tipo de dato genérico (cadena de caracteres, numérico, etc.). Se debe aclarar que las pantallas no deben estar vacías, se deben incluir ejemplos.

### **2.2.9 Las validaciones con el cliente y modificaciones**

Una vez arribado a este punto, las pantallas ya han sido validadas así como el mapa conceptual, por supuesto con las observaciones realizadas por la parte cliente, esta información tiene influencia en la siguiente versión del prototipo, la cual se presenta modificada, refinada.

Aquí se procede a una tercera ronda de validación. Primero se entrega el documento descriptivo con antelación a los especialistas funcionales (clientes) para que se lo estudien. Posteriormente se transita a una sesión de taller en que los analistas pasan a explicar el Sistema Navegable no Funcional. El papel del usuario con el prototipo puede resumirse en compromiso y honestidad, cuando este interactúa con el prototipo tiene las responsabilidades de utilizar y evaluar el prototipo las veces que sea necesario, identificar mejoras, sugerir las características no deseadas, describir los requerimientos de datos y describir la salida deseada.

Los analistas familiarizan al usuario con el prototipo, observan y evalúan la reacción del usuario al trabajar con el prototipo y pueden ver qué tan conveniente ha resultado el acoplamiento entre las necesidades y las características modeladas en el sistema; de esta forma se descubren nuevas perspectivas del prototipo e incluso si se encuentran satisfechos los usuarios o si existen dificultades para implantar el sistema. Suelen adjuntarse algunas sugerencias que resultan muy relevantes para esclarecer los caminos por los cuales dirigirse para refinar el prototipo y modificarlo de forma tal que satisfaga en mayor medida las necesidades de los usuarios.

Las modificaciones al prototipo y por ende al documento descriptivo son llevadas a cabo a partir de sugerencias recogidas en estos talleres.

### **2.2.10 ¿Hasta dónde validar?**

La etapa de validación anterior puede servir para organizar aspectos de la interfaz pero no es el centro, el cliente puede comenzar a aportar ideas acerca de la estética de la interfaz y se debe tratar de enfocar la atención en la obtención de información que afecte al prototipo para que aparezcan nuevos elementos de datos o acciones sobre estos y parar cuando ya no surjan, lo cual no quiere decir que no se tomen en cuenta las sugerencias. Se valida hasta un punto en que se ve que el prototipo es estable. Las etapas de captura de la información sobre lo que le gusta y lo que le desagrade a los usuarios y la conclusión en la siguiente versión del prototipo se repite varias veces hasta que estén usuarios y profesionales de sistema



de acuerdo en que el prototipo ha evolucionado lo suficiente o que una iteración más no traerá beneficios adicionales.

Este proceso concluye con una aceptación formal de parte del cliente con sus observaciones tomadas en cuenta posteriormente a la hora de realizar el diseño de la interfaz.

## **2.3 Del prototipo a la interfaz gráfica**

A través diferentes niveles evolutivos de prototipos se pueden lograr simulaciones del sistema, mas la interfaz gráfica del mismo, el entorno con que va a interactuar el usuario, implica niveles de detalle superiores. Es importante que la interfaz de usuario, (en lo adelante IU), esté bien diseñada, para que de esta forma el usuario encuentre la respuesta que espera a su acción.

### **2.3.1 Especificaciones Generales para el Diseño de la Aplicación IASP**

La “cara” que iba a tener la aplicación del SIGEP estaba siendo analizada inclusive antes de capturar los requisitos. Fueron varias las propuestas analizadas, versiones diferentes propuestas por el diseñador profesional que confeccionó la estética de la aplicación en lo que a combinación de colores, maquetado y formas se refiere. El líder del proyecto, el equipo de diseño gráfico en conjunto con y el diseñador, programaron encuentros en varias ocasiones para confrontar criterios acerca de las diferentes versiones y el nivel de aceptación que se le otorgaba a cada una de estas como fachada representativa para sistema penitenciario venezolano. Luego de varios intercambios de opiniones se decide adoptar la versión que para la mayoría de los observadores resultaba ser la más adecuada por el grado de claridad, nivel de seriedad aportado, combinaciones de colores etcétera. Se confecciona un documento que compendia los detalles que debe tener en cuenta el equipo de diseño gráfico para llevar estas características a las “pantallas”, las pautas del diseño de la interfaz. (ver Anexo 3)

A grandes rasgos la aplicación fue diseñada a una resolución de 1024 x 768 px, se optimizó particularmente la composición horizontal del contenido en aras de aprovechar al máximo las características técnicas de monitores y navegadores en general, mostrando así la mayor cantidad de información sin necesidad de escrollar la página.

Se utilizaron colores RGB sobrios y de comportamiento estable en los principales visualizadores (Internet Explorer, Mozilla, Opera, Safari etc.,) y monitores. El comportamiento de la gama cromática escogida (colores poco saturados y con altas claridades) estaba en armonía con los colores pautados en el Manual de Identidad Visual Institucional, manteniendo la imagen de seriedad que se deseaba connotar. Se utilizaron como tipografía principal para el contenido la Tahoma, por ser una tipografía de alto rendimiento en pantalla y común por defecto a la mayoría de los navegadores y la Formata (tipografía institucional) para el realce del nombre oficial de la aplicación.

Se personalizaron los íconos para que fueran reconocidos como un sistema, escogiéndose en cada caso los temas a iconizar en función del contenido y el uso.

## **2.3.2 Confección de las pantallas**

Con el documento de las pautas por las que se debe regir la confección de las pantallas independientes que conforman luego el sistema navegable que se desea, se cuenta con la base para emprender la elaboración de la fachada de la aplicación. Las pautas detallan el estilo que se muestra en un archivo PSD (archivo de photoshop) que reciben los diseñadores gráficos. Posteriormente hay que transformar este estilo en código, y resulta conveniente, por el número considerable de ventajas que implicaba su empleo, confeccionar una Hoja de Estilo en Cascada, o sea, CSS (Cascade Style Sheet) que aporte el estilo a la página, hablando en términos del documento HTML que recibe este estilo.

### **2.3.3.1 Trabajo a partir de los archivos PSD**

El equipo de diseño gráfico recibe un grupo de archivos PSD que es la extensión con que se salvan los archivos realizados en Photoshop, el estándar de imágenes raster a nivel internacional. Cada uno de estos representa las propiedades gráficas de una pantalla en específico y algunas de estas varían de una a otra pues no cumple objetivo varias pantallas con características muy similares en que solo cambien algunos aspectos tales como la ubicación de algún que otro contenido, la maquetación es algo que entra en acción posteriormente, aquí sólo se necesita tener plantillas de las cuales extraer elementos para incluir en las pantallas posteriormente mediante el empleo del CSS.

El PSD se abre con el Adobe Photoshop, su última versión CS2 resulta comodísima para trabajar pues incluye facilidades de trabajo con las capas que no contienen las versiones anteriores, una vez abierto el

archivo se procede a descomponer en fragmentos la plantilla, segmentos cuya disposición en la plantilla web haya sido calculado por el diseñador gráfico y pensado mediante el uso del CSS. Se selecciona la opción “salvar para web” y los fragmentos se salvarán como imágenes separadas para con ellas confeccionar las pantallas aprovechando las propiedades del CSS.

### **2.3.3.2 Obtención del CSS**

Existe un grupo considerable de ventajas que aporta el uso del CSS, para el equipo de diseño gráfico resulta conveniente usar las hojas de estilo por la facilidad con que se pueden realizar cambios posteriores a la aplicación sin tener que tocar el código general del documento y hacer cambios elemento por elemento, con solo variar los aspectos necesarios en la hoja de estilo o incluir nuevos elementos (clases, identificadores, etc.), es suficiente para que cada pantalla adopte la forma descrita por el código. Con el CSS se separa el contenido de formato en las páginas, o sea que permite independizar el contenido de una página Web de su estructura y da la posibilidad de alterar su aspecto sin necesidad de modificar la página, sino tan sólo su estilo. Otro factor muy importante a tener en cuenta es si es una aplicación grande, con un cúmulo considerable de páginas, aquí CSS logra acrecentar la densidad de las palabras clave dentro de los contenidos, y muchas de las etiquetas ocuparán muchísimo menos espacio. Esto también supone un menor peso para las páginas Web, lo cual agradecen los usuarios finales. El sistema para gestión de información del recluso cuenta con un número importante de páginas que comparten los mismos formatos para el encabezado y el cuerpo, y al usar CSS para describir este formato si en un momento dado se necesitara cambiar el aspecto del encabezado tan sólo se debe modificar la definición de su estilo sin tener que editar una por una esos cientos de páginas. Permite además después de incluidos bloques de texto en cualquier orden en el documento, darle un orden conveniente utilizando el llamado posicionamiento absoluto.

Hay que tener en cuenta los exploradores más usados en la red y el ideal de construir programas compatibles, que funcionen en ambos exploradores.

El equipo de diseño gráfico del proyecto SIGEP tuvo que tener en consideración la elaboración de dos hojas de estilo, para ver correctamente la aplicación tanto en el navegador Mozilla Firefox como en el Internet Explorer, de modo tal que cuando la aplicación levantase si el navegador utilizado era Mozilla cargaba el estilo para Firefox style\_1 (Ver Anexo 2) y si era Internet Explorer cargaba el style\_1\_IE (Ver Anexo 4)

### **2.3.3.3 Confección del documento HTML**

Cumplir con el estándar HTML abre puertas a diseñar páginas web para dispositivos móviles o nuevas tecnologías que vayan surgiendo. Los documentos HTML contienen párrafos, tablas, imágenes, hipervínculos y son interpretados por un explorador o navegador que se encarga de presentar el documento al usuario final. El uso de CSS permite cambiar el aspecto de estas páginas en cuestión de minutos. Esta resulta una combinación muy factible por la facilidad con que permite realizar cambios que se necesitan muchas veces en períodos de tiempo muy cortos.

Una vez confeccionada la hoja de estilo, se puede proceder a confeccionar el documento HTML con la ayuda siempre del dreamweaver para la inclusión de tablas, formularios... sin necesidad de ir al código no más que para hacer modificaciones en algunas propiedades. Este documento HTML adoptaba la forma prevista en las pautas del diseño mediante la invocación al CSS hecha a través de la etiqueta `<link rel="stylesheet" type="text/css" href="/direcciondelCSS">` incluida antes de comenzar con el cuerpo (`<body>`) del documento.

En la elaboración de este documento HTML es donde se dispone el contenido según la noción que da el prototipo de interfaz, y por supuesto siguiendo las pautas establecidas anteriormente para que la aplicación cumpla con las expectativas creadas. El CSS no es algo que se crea al inicio y ya no se modifica más, todo lo contrario, a medida que se van incluyendo elementos nuevos en la página van surgiendo nuevas necesidades de estilizar y de efectuar modificaciones en el código CSS y agregar nuevas clases, identificadores, etc.

Existen pantallas cuya lógica en la distribución del contenido no se percibe de una forma clara con el prototipo, es muy útil contar con la ayuda de los analistas porque inclusive existe una alta probabilidad de cambios con respecto a lo observado en el prototipo ya que como se mencionaba anteriormente, el prototipo no es el sistema final y es muy volátil, además de que a partir de aquí se comenzará la implementación del sistema real que promete satisfacer las necesidades del cliente en serio pues ya en este punto se suponen emergidos los requerimientos del sistema en casi su totalidad y las versiones se van tornando más estables. (Ver Anexo 5)

## **2.4 Conclusiones**

Este capítulo está fragmentado según el número de pasos seguidos para conformar el prototipo y a partir de este la interfaz gráfica de usuario. Esta secuencia de pasos está definida según el orden en que se llevan a cabo los procesos de manera que pueda servir como estrategia para orientar la especificación de requerimientos con uso de la técnica del prototipado en proyectos con características similares a las del SIGEP.

## **CAPÍTULO 3: Análisis de los resultados**

### **3.1 Introducción**

El análisis de los resultados de una propuesta se hace para que se puedan efectuar valoraciones acerca de la misma sobre la base de sus ventajas y limitaciones. Los resultados se consideran en virtud de su impacto y se suele tomar como referencia opiniones tanto de personas conocedoras del fenómeno como de personas que han interactuado con este. A continuación se muestran los resultados arrojados tras la aplicación de una encuesta al grupo de diseñadores de software del proyecto SIGEP para obtener su valoración acerca del empleo del prototipo para especificar requisitos de software y como punto de partida al desarrollo de su trabajo. Una de las maneras más contundentes de catalogar los resultados obtenidos tras la captura de requisitos, según la eficacia que tuvo, es con los resultados obtenidos tras una ronda de pruebas realizada a uno de los módulos del sistema y que se muestran tras el análisis de las encuestas. Una vez analizadas las encuestas así como los resultados de las pruebas se detallan beneficios aportados por el prototipo y su uso desde etapas tempranas del desarrollo así como sus limitaciones.

### **3.2 Método empírico: Encuesta**

La encuesta se realiza para, a través de un conjunto de preguntas, obtener una información acerca del mundo interior del encuestado o su percepción del fenómeno que se investiga, en este caso: el empleo de la técnica de prototipado.

La encuesta fue aplicada al grupo de diseñadores de software específicamente porque estos son los primeros que van a trabajar a partir del prototipo y en función de los objetivos del sistema. El objetivo de la encuesta es valorar ciertos aspectos relacionados con la especificación de los requisitos mediante el uso del prototipo y la percepción por parte de este equipo pues ellos en su actividad, deben descubrir los usuarios potenciales (la determinación de cuánto se puedan involucrar los usuarios depende del diseño), sus requisitos, además de decidir qué información incluir en el software. El proceso de diseño debe cumplir con los requisitos, que son esencialmente, un problema a resolver, y el prototipo fue el puente que se utilizó para expresar dichas necesidades.

Se elaboró una encuesta con el objetivo de conocer cómo valoran los diseñadores de software la aplicación de la técnica de uso del prototipo de interfaz desde etapas tempranas así como su factibilidad desde su perspectiva. (Ver Anexo 6)

Dicha encuesta fue distribuida, entregando un ejemplar impreso a cada uno de los integrantes del equipo de diseñadores. Se basó en una serie de tres preguntas que medían la factibilidad del uso de la técnica del prototipo para especificar requisitos, cuán fácil o difícil le resultó desempeñar de su trabajo (diseño de software) a partir del prototipo y la consideración como técnica de especificación de requisitos respecto a otras. El cuestionario se aplicó de forma anónima.

Los indicadores específicos utilizados para llegar al resultado final se basaron, en gran parte, en la retroalimentación recibida de las encuestas.

### **3.2.1 Valoración de la técnica de uso del prototipo para especificar requisitos**

Los indicadores para que el diseñador catalogara el uso del prototipo como técnica para especificar requisitos son los siguientes:

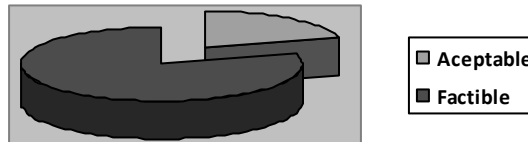
- Muy factible
- Factible
- Aceptable
- Poco factible
- No factible

Una de las aseveraciones que mostró mayor acuerdo fue que el prototipo resulta una técnica factible de especificación de requisitos.

Tras la elección realizada por cada uno de los miembros del equipo los resultados arrojados fueron los siguientes:

Cuatro de los cinco integrantes catalogó la técnica de uso del prototipo para especificar requisitos como “Factible”, para un 80% de la totalidad de los miembros y uno de los cinco diseñadores valoró como “Aceptable” la aplicación de la técnica, para un 20% de la totalidad de los miembros. Ver Fig 2

**Fig. 3.1 Prototipo como técnica para Especificación de Requisitos**



Las consideraciones tomadas de sus respuestas que argumentan las selecciones realizadas son las siguientes:

- ✓ El prototipo, primeramente, aterriza, o sea, da una idea más objetiva acerca del flujo en el que deben interactuar la aplicación y el cliente.
- ✓ Permite tener una idea de la funcionalidad global del sistema.
- ✓ El prototipo es factible para especificar los requisitos debido a que se tiene una visión de lo que va a ser y a hacer en la vida real la aplicación. Además se tiene un mayor dominio de la funcionalidad que se espera para el sistema.
- ✓ Para capturar requisitos el prototipo es una buena herramienta, esencialmente para validaciones con el cliente al cual le puede resultar más familiar un prototipo de la aplicación final que un documento descriptivo de los requisitos en un lenguaje técnico.

### **3.2.2 Valoración del grado de dificultad del trabajo a partir del prototipo**

Los indicadores para que los diseñadores valoraran el grado de dificultad del trabajo que realizan a través del prototipo son:

- Muy fácil
- Fácil
- Aceptable
- Dificultoso



- Muy dificultoso

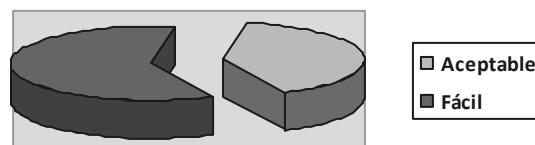
El mayor peso recae sobre la categorización de “Fácil” para el trabajo a partir del prototipo con un 60% y el 40% restante para una consideración de “Aceptable”.

Tras la elección realizada por cada uno de los miembros del equipo los resultados específicos arrojados fueron los siguientes:

Tres de los cinco integrantes catalogaron el trabajo a partir del prototipo como “Fácil”, para un 60% de la totalidad de los diseñadores que se sienten cómodos realizando el diseño a partir de un prototipo bien descrito y documentado.

Dos de los diseñadores consideró “Aceptable” el trabajo a partir del prototipo de interfaz, para un 40% de la totalidad que compone el equipo de diseño de software. Ver Fig 3.

**Fig 3.2 Valoración del trabajo de diseñadores a partir del prototipo**



Las fundamentaciones alegadas a sus respuestas a la pregunta anterior fueron:

- ✓ Ayuda a la toma de decisiones y a detectar funcionalidades.
- ✓ El prototipo introduce cierto grado de familiaridad con el sistema, a partir de él se obtiene además una idea de lo que se quiere implementar, aunque se necesita consultar las descripciones de la funcionalidad para definir bien los datos manipulados.

- ✓ Facilita comprender cómo se va a producir la interacción y están definidas las responsabilidades del sistema.

De la evaluación de las encuestas en este aspecto se extrajo una observación importante que alegar al prototipo como punto de partida hacia el diseño, y fue por este motivo que uno de los encuestados catalogó su trabajo a partir del prototipo como “Aceptable”.

**Observación:** Existe necesidad de algo que relacione los modelos de datos representados en el prototipo con los del modelo lógico de datos.

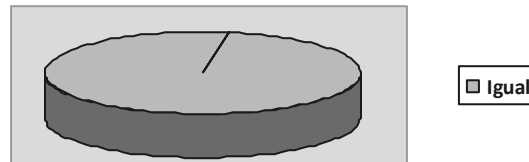
### **3.2.3 Comparación de la técnica del prototipo con otras técnicas de especificación de requerimientos**

Los indicadores para que los diseñadores compararan la técnica del prototipado con las demás técnicas usadas para especificar requisitos fueron:

- Mucho Mejor
- Mejor
- Igual
- Peor
- Mucho Peor

La mayor cantidad de aseveraciones en esta comparación recae sobre el calificativo “Igual”, para un 100% de integrantes del equipo de diseñadores que considera que la técnica de especificación de requisitos de prototipado, no es mejor ni peor que las demás técnicas, simplemente es una buena técnica para facilitar aparición de nuevos requisitos y validar los existentes. Ver Fig 4.

**Fig 3.3 Comparación con otras técnicas**



Las justificaciones alegadas por los diseñadores para sustentar su elección fueron:

- ✓ El prototipado es una buena técnica porque ayuda al cliente a tener una idea de lo que obtendrá con el producto. Facilita al equipo de desarrollo calcular el alcance del sistema.
- ✓ El prototipo facilita la comprensión de las necesidades del sistema, propicia una interacción más real que otras técnicas y es por ello que el cliente puede comprender en mayor medida el funcionamiento del futuro sistema.
- ✓ Con el prototipo pueden comenzar a explotarse funcionalidades desde la captura de requisitos y de este modo pueden surgir nuevas necesidades para el sistema a conformar.

### **3.3 Primera prueba de validación con el cliente**

Las pruebas de software son experimentos realizados para buscar errores en el mismo. Para esta primera prueba se tomó uno de los módulos del sistema y fue sometido a rondas de verificación por la parte cliente mediante interacciones con el mismo para explotar sus vulnerabilidades. Se realizaron durante dos días y con el objetivo esencial de probar la funcionalidad del módulo y detectar errores escondidos que pudieran ir degradando el funcionamiento del sistema.

El resultado fue un documento con las no conformidades (NC), es decir, los defectos, errores y sugerencias realizadas al equipo de desarrollo.

Fueron señaladas cinco no conformidades (NC) relacionadas con los requisitos funcionales.

- Emergieron dos nuevos requisitos.  
La solicitud de nuevo requisito implica plantearse un nuevo requisito en la solución pactada.
- Se sugirieron dos cambios de requisitos.  
Solicitud que implica modificar un requisito en la solución pactada.

Estos resultados evidencian el buen desenvolvimiento de de las actividades en la especificación de requisitos. El éxito de este proceso indica que la estrategia que se siguió fue adecuada, lo cual demuestra que el prototipo fue una técnica factible y logró especificar los requisitos de manera correcta.

### **3.4 Ventajas obtenidas del uso del prototipo**

Cuando se pensó en la utilización de un prototipo de interfaz se hizo teniendo en cuenta sus capacidades como método de validación con el usuario, ya que usualmente cuando se piensa en técnicas para realizar la especificación de los requisitos siempre se piensa en el primer instante en los Casos de Uso y estos no son el único modo para expresar los requisitos funcionales ya que son detallados y a menudo el cliente desea un resumen breve que identifique las funcionalidades más destacadas. En el Proceso Unificado, una característica del sistema es "un servicio observable externamente proporcionado por el sistema que cumple directamente una necesidad del personal involucrado" (24), observable más fácilmente a través de un prototipo de interfaz. La veracidad del planteamiento anterior se prueba mediante resultados corroborados con el desenvolvimiento de las actividades en el proyecto y la obtención de ciertos artefactos.

Debido a razones expuestas en capítulos anteriores se decide agilizar el proceso y comenzar a elaborar el prototipo desde etapas muy tempranas y a explotar sus capacidades. Este artefacto propició un ambiente favorable para el surgimiento de nuevos requisitos y necesidades del sistema. La experiencia con el

sistema bajo condiciones reales permitió familiaridad, que fue indispensable para determinar los cambios o mejoras necesarios así como la eliminación de características innecesarias. El éxito del uso del prototipo depende de qué tan pronto y con qué frecuencia se reciba la retroalimentación del usuario para hacer cambios y adecuarlos a las necesidades existentes, por ello su desarrollo para el proyecto se considera exitoso pues se efectuaron cambios iniciales durante el desarrollo del proyecto, y resultó menos costoso que si se realizara en etapas tardías, ya que el prototipo puede cambiar varias veces, es por ello que la flexibilidad y adaptabilidad son su esencia, la pauta del cambio la da la retroalimentación, la cual nos permite conocer la opinión del usuario sobre cambios a la entrada o salida de un proceso, que al evaluarla nos permite obtener los requerimientos y mejorar el sistema. El usuario final pudo emitir criterios sobre la forma del sistema pues participó activamente en la definición del mismo, hubo interacción con el equipo constantemente como si fuesen parte de este y de esta forma se sentía parte responsable del desarrollo, o sea que proporcionó una retroalimentación temprana por parte de los usuarios acerca del sistema y su participación activa desde el principio hizo que fueran los principales promotores del éxito. Los usuarios crearon expectativas en el futuro software.

El desarrollo del prototipo obligó a los analistas a pensar en la solución del problema y por lo tanto a profundizar en los requisitos del software. La información que se obtuvo con el uso de prototipos permitió al grupo de analistas establecer prioridades y reorientar sus planes de una manera menos costosa y con un mínimo de contratiempo.

El prototipo constituyó una forma clara de representar los requisitos del software.

Hizo posible que los requisitos pudieran ser validados y aprobados.

En las validaciones del prototipo se detectaron errores que de lo contrario hubieran emergido en las pruebas de aceptación del software, esta validación constituye una revisión anticipada del sistema.

Como punto de partida a la implementación y diseño de la aplicación puede catalogarse el uso del prototipo como una buena práctica pues con el prototipo el equipo de programadores y diseñadores obtuvo una idea clara de lo que se pretendía implementar así como de su alcance. Además permitió tener las pantallas listas para el desarrollo proporcionando una mejor percepción para el trabajo de los diseñadores gráficos.

Por último y no menos importante, se logró el propósito principal: quedó claro el concepto y objetivo del sistema.

### **3.4 Limitaciones del prototipo**

No todo son glorias para el prototipo pues además del número importante de ventajas que aporta mencionadas anteriormente, posee también algunas limitaciones, que no son tantas ni tan fuertes como para empañar todo lo bueno que aporta analizado ya, ni para no considerar su uso una decisión asertada. La primera limitante que tiene el prototipo es que él por sí solo no expresa funcionalidad del sistema: en qué proporción la interacción del usuario con el sistema permite modificar el estado de los datos del mismo.

El prototipo tampoco tiene implícitas las reglas del negocio, o sea, las restricciones para los procesos y los datos que se manejan en la organización, en este caso en la DGSP. Además no representa las relaciones entre los datos.

### **3.5 Elementos adicionales necesarios**

Por las limitaciones expuestas anteriormente que tiene el prototipo se adjuntan al mismo otros elementos adicionales para aportar esto de lo que carece él por sí solo y mejorar el grado de factibilidad que aporte su uso. Uno de estos elementos es la descripción textual de la funcionalidad asociada a cada pantalla, ofrecida mediante un documento de descripción del prototipo. Otro elemento es el modelo lógico de datos, que incluye tablas, entidades, y relaciones entre entidades, todas representadas mediante el Erwin Estudio. También se agregan las reglas del negocio para mostrar restricciones además de los requisitos no funcionales.

### **3.6 Conclusiones**

Mediante el estudio de las encuestas realizadas al grupo de diseñadores de software y análisis de los resultados de la primera prueba de aceptación se logró aseverar que el prototipo, en efecto, es una buena técnica de especificación de requerimientos.

## CONCLUSIONES GENERALES

La conclusión de este trabajo ha posibilitado cumplimentar los objetivos para los cuales se desarrolló la investigación, logrando los siguientes resultados:

Quedó definida una estrategia para apoyar la captura de requisitos en el proyecto SIGEP, facilitando la orientación de sus actividades.

De manera más específica

- Quedaron definidos y descritos los pasos para obtener el prototipo de interfaz durante y para la captura de requisitos dejando abierto ciertos temas que deben ser refinados en el desarrollo práctico.
- Se definió el prototipo como punto de partida a la implementación, es decir, la interfaz gráfica de usuario.
- Se extrajeron ventajas y limitaciones así como valoraciones de la factibilidad del uso del prototipo durante la captura de requisitos del software.

Las ideas relacionadas con la especificación de requisitos, muestra la importancia y repercusión en la calidad y efectividad del proceso de desarrollo del software.

## **RECOMENDACIONES**

Por los aportes que tiene la aplicación de una estrategia para guiar un proceso tan importante como el de la especificación de requisitos se recomienda el seguimiento de la estrategia propuesta para orientar el mismo.

Los objetivos específicos del trabajo no abarcan todos los temas relacionados con la especificación de requisitos de software; los cuales son amplios y diversos. Por lo que se considera necesario garantizar la identificación y seguimiento de elementos que pueden ser útiles en la optimización de la estrategia propuesta. Por lo que se sugiere:

- Agregar a la propuesta las reglas del negocio.
- Buscar un vínculo que relacione los modelos de datos representados en el prototipo con los del modelo lógico de datos.



## BIBLIOGRAFÍA

- **1233, IEEE Std. 1998.** Guía para el desarrollo de Especificaciones de Requerimientos de Sistemas. 1998.
- **A, Díez. 2001.** IRqA y el desarrollo de proyectos: Experiencias Prácticas. I Jornadas de Ingeniería de Requisitos Aplicadas. Sevilla, España : s.n., 2001.
- **Álvarez, Miguel. 2002.** Álvarez, Miguel. [En línea] 2002. <http://www.desarrolloweb.com/articulos/490.php>
- **IEEE-Std-1233. 1998.** Guía para el desarrollo de Especificaciones de Requerimientos de Sistemas. 1998.
- **Lic. Domingo F. Donadello.** METODOLOGIA INTEGRAL DE DESARROLLO DE SISTEMA . [En línea] [http://www.ub.edu.ar/catedras/ingenieria/ing\\_software/ubftecwwwdfd/mids\\_web/prototyp/estrdes.htm](http://www.ub.edu.ar/catedras/ingenieria/ing_software/ubftecwwwdfd/mids_web/prototyp/estrdes.htm) .
- **Lowe, D., Hall, W. 1999.** Hypermedia and the Web. An Engineering approach. 1999.
- **Lu, Yu.** From Requirements to Architectural Design using Goals and Scenarios Proceedings of the 6th Micon Workshop. . Canadá : s.n.
- **María José Escalona, Nora Koch. 2002.** Ingeniería de Requisitos en Aplicaciones para la Web, un estudio comparativo. Sevilla, España : s.n., 2002. Rincón del Vago. [En línea] <http://html.rincondelvago.com/problemas-que-plantea-la-planeacion-de-proyectos.html>.
- **2005.** Softonic. [En línea] 2005. <http://adobe-photoshop.softonic.com/>.
- **2006.** Terra. [En línea] 2006. <http://descargas.terra.es/ie/7564-Macromedia-DreamWeaver>.
- **Walter Maner . 1997.** [En línea] 1997. <http://www.sidar.org/recur/desdi/traduc/es/visitable/maner/Prototipado.htm>.
- **Weidenhaupt, K., Pohl, K., Jarke, M., Haumer, P. 1999.** Scenarios in Systems Development: Current Practice. 1999.

## REFERENCIA BIBLIOGRÁFICA

- **Durán A., Bernárdez, B., Ruiz, A. 1999.** A Requirements Elicitation Approach Based in Templates. 1999.
- **Escalona, M.J., Mejías, M., Torres, J. 2002.** Methodologies to develop web information systems and comparative. 2002.
- **Kruchten, P. 1998.** The Rational Unified Process . s.l. : Addison Wesley, 1998.
- **Alberto la Calle. 2006.** [En línea] 2006. [http://www.albertolacalle.com/hci\\_prototipos.htm](http://www.albertolacalle.com/hci_prototipos.htm) .
- **Murugarren, Joaquín Gracia. . 2004.** [En línea] . 2004. <http://www.webestilo.com/posicionamiento-web/html-css.phtml>.

## GLOSARIO

- ✓ **Artefacto:** En tecnología, artefacto es un dispositivo concebido y fabricado, sea de modo artesanal o industrial, por una o más personas. La característica principal de los artefactos es que cumplen una función técnica, es decir, sirven para hacer algo.
- ✓ **Elicitación:** La Elicitación consiste en las primeras actividades a realizarse en la Ingeniería de requisitos, aunque esta etapa no se puede “divorciar” de las demás, ya que seguramente se iterará a través de las mismas durante el desarrollo de los requerimientos.
- ✓ **Estrategia:** La estrategia es cualquier conjunto de acciones o comportamiento, sea deliberado o no. Definir la estrategia como un plan no es suficiente, se necesita un concepto en el que se acompañe el comportamiento resultante. Específicamente, la estrategia debe ser coherente con el comportamiento.
- ✓ **HTML:** “Hyper Text Markup Language”, lenguaje de marcas, con que se programan las páginas Web. Brinda facilidades para mostrar imágenes, textos hipervínculos, tablas, etc., y es interpretado por los navegadores Web.
- ✓ **IEEE:** corresponde a las siglas de The Institute of Electrical and Electronics Engineers, el Instituto de Ingenieros Eléctricos y Electrónicos, una asociación técnico-profesional mundial dedicada a la estandarización, entre otras cosas.
- ✓ **Interactividad:** Expresión extensiva que en una serie de intercambios comunicacionales implica que el último mensaje se relaciona con mensajes anteriores a su vez relativos a otros previos.
- ✓ **Metodología:** La metodología se entenderá aquí como la parte del proceso de investigación que sigue a la propedéutica y permite sistematizar los métodos y las técnicas necesarios para llevarla a cabo.
- ✓ **Modelos de Casos de Uso:** Su especificación permite definir la arquitectura de aplicaciones informáticas y precisar la funcionalidad de los procesos de negocio.
- ✓ **Norma:** En tecnología, una norma o estándar es una especificación que reglamenta procesos y productos para garantizar la interoperabilidad.
- ✓ **Ontología:** La Ontología se ocupa de la definición del ser y de establecer las categorías fundamentales o modos generales de ser de las cosas a partir del estudio de sus propiedades.
- ✓ **Sistema:** Un sistema es una colección de unidades organizadas para cumplir un propósito en específico. Un sistema puede ser descrito por uno o más modelos y posiblemente desde diferente.

- ✓ **Terminología:** Es una ciencia interdisciplinar que se nutre de un conjunto específico de conocimientos conceptualizado en otras disciplinas.
- ✓ **Trazabilidad:** Es un conjunto de medidas, acciones y procedimientos que permiten registrar e identificar cada producto desde su origen hasta su destino final.
- ✓ **Usabilidad:** El modelo conceptual de la usabilidad, proveniente del diseño centrado en el usuario, no está completo sin la idea utilidad.