

**INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE  
MAESTRÍA EN INFORMÁTICA APLICADA**

**Reconocimiento de validez oficial de estudios según acuerdo SEP No. 984179 del 24 de Agosto de 1998**



**PLAN PARA ADMINISTRAR EL PROYECTO Y DESARROLLO DE LOS  
REQUERIMIENTOS PARA GENERAR LOS REPORTES DE LA EVALUACIÓN DEL  
DESEMPEÑO DOCENTE DEL SISTEMA DE INFORMACIÓN DE LA UNIVERSIDAD  
ITESO.**

**TESIS  
QUE PARA OBTENER EL GRADO DE  
MAESTRÍA EN INFORMÁTICA APLICADA**

**PRESENTA:  
FRANCISCO ÁLVAREZ PARTIDA**

**ASESOR:  
DR. FRANCISCO RIVERA MARTÍNEZ**

**Tlaquepaque, Jalisco,**

**Agosto de 2003**

**A mis esposa Lola**

**A mis hijos María Lola y Juan Diego**

## AGRADECIMIENTOS

En primer lugar a las personas en esta universidad que me dieron la oportunidad de seguir creciendo como ser humano, en el estudio que tanto me gusta.

A mi asesor de tesis el Dr. Francisco Rivera Martínez por sus valiosas orientaciones que evitaron que me perdiera en la investigación durante incontables ocasiones.

A la Mtra. Gisel Hernández co-asesora y al Mtro. Carlos Fernández sobre cuya metodología de desarrollo de requerimientos me apoyé para realizar este trabajo.

Al Lic. Francisco Núñez de la Peña y al Lic. Pedro Ramírez mis jefes inmediatos; por sus análisis y aportaciones al proceso de la Evaluación del Desempeño Docente y Académico del Iteso.

A los alumnos de Ingeniería en Sistemas Computacionales Aitana Díaz y José Luis Navarro quienes prestaron su servicio social colaborando en la obtención de los primeros requerimientos y en el desarrollo del primer prototipo.

A Francisco Santana alumno de Ingeniería en Sistemas Computacionales, becario que colaboró en la obtención de la bibliografía para el marco teórico. También a los alumnos de la misma carrera Raúl Alejandro Estrella y Mario Alberto Ruiz quienes prestaron su servicio social desarrollando los requerimientos para el segundo prototipo del módulo.

A todo el personal de esta universidad que desde las Direcciones, Departamentos, Centros y Oficinas que han colaborado con la Evaluación del desempeño docente, por su esfuerzo cotidiano para mejorar nuestros métodos, procesos y la calidad educativa de nuestra universidad, gracias por su entusiasmo y al ánimo que me han transmitido durante todos estos años.

Francisco Álvarez

Agosto 2003

## CONTENIDO

<b>1</b>	<b>INTRODUCCIÓN.....</b>	<b>9</b>
<b>1.1</b>	<b>Objetivo .....</b>	<b>10</b>
<b>1.2</b>	<b>Metodología.....</b>	<b>10</b>
<b>1.3</b>	<b>Alcance .....</b>	<b>11</b>
<b>1.4</b>	<b>Estructura de la tesis.....</b>	<b>11</b>
<b>2</b>	<b>MARCO DE REFERENCIA .....</b>	<b>12</b>
<b>2.1</b>	<b>Ingeniería de Sistemas de Información. ....</b>	<b>12</b>
<b>2.1.1</b>	<b>Conceptos básicos.....</b>	<b>12</b>
2.1.1.1	Sistema. ....	12
2.1.1.2	Ingeniería.....	13
2.1.1.3	Ingeniería de sistemas. ....	14
2.1.1.4	Funciones de un ingeniero en sistemas. ....	15
2.1.1.5	Información .....	15
2.1.1.6	Sistemas de Información. ....	16
<b>2.2</b>	<b>Ingeniería de software.....</b>	<b>17</b>
<b>2.2.1</b>	<b>Conceptos básicos.....</b>	<b>17</b>
2.2.1.1	Ingeniería de Software .....	17
<b>2.2.2</b>	<b>Proceso de desarrollo de Software.....</b>	<b>18</b>
<b>2.3</b>	<b>Ingeniería de requerimientos. ....</b>	<b>19</b>
<b>2.3.1</b>	<b>Conceptos básicos.....</b>	<b>19</b>
2.3.1.1	Definición de requerimientos de software. ....	19
2.3.1.2	Definición de Ingeniería de requerimientos. ....	20
<b>2.3.2</b>	<b>Relación de los Requerimientos con otros procesos.....</b>	<b>22</b>
<b>2.3.3</b>	<b>Clasificación de requerimientos.....</b>	<b>23</b>
2.3.3.1	Clasificación por niveles de requerimientos. ....	23
2.3.3.2	Clasificación por tipos de requerimientos. ....	23

2.3.3.3	Subdivisión por tipos de requerimientos.....	24
<b>2.3.4</b>	<b>Profesionales de la IR .....</b>	<b>28</b>
2.3.4.1	Analista.....	29
2.3.4.2	Analista de requerimientos de sistemas. ....	29
2.3.4.3	Analista de requerimientos de software. ....	29
2.3.4.4	Trabajo en equipo.....	29
<b>2.3.5</b>	<b>Proceso de Ingeniería de Requerimientos.....</b>	<b>30</b>
2.3.5.1	Subproceso de Desarrollo de Requerimientos de Sistema.....	33
2.3.5.2	Subproceso de Diseño de la Arquitectura del Sistema.....	47
2.3.5.3	Subproceso de Desarrollo de Requerimientos de Software .....	48
<b>2.3.6</b>	<b>Métodos de la Ingeniería de Requerimientos. ....</b>	<b>57</b>
2.3.6.1	Conceptos básicos. ....	57
2.3.6.2	La disciplina en los proyectos de Ingeniería de Requerimientos.....	57
2.3.6.3	El enfoque estructurado.....	58
2.3.6.4	El lenguaje de patrones. ....	60
2.3.6.5	El enfoque orientado a objetos. ....	61
2.3.6.6	El UML .....	65
2.3.6.7	Diseño Centrado en el Usuario.....	67
2.3.6.8	Modelo de madurez para el desarrollo de software (CMM).....	68
<b>2.3.7</b>	<b>Técnicas del proceso de requerimientos.....</b>	<b>69</b>
2.3.7.1	Lluvia de ideas. ....	70
2.3.7.2	Entrevistas .....	70
2.3.7.3	Encuestas.....	71
2.3.7.4	Prototipos .....	72
<b>2.3.8</b>	<b>Herramientas de la Ingeniería de Requerimientos. ....</b>	<b>75</b>
<b>2.3.9</b>	<b>Mejora de procesos de requerimientos. ....</b>	<b>76</b>
2.3.9.1	Los objetivos de los procesos de mejora.....	76
2.3.9.2	Fundamentos de mejora de procesos.....	76
2.3.9.3	Ciclo de mejora de procesos.....	77
<b>2.4</b>	<b>Administración de Proyectos de Requerimientos.....</b>	<b>78</b>
<b>2.4.1</b>	<b>Conceptos básicos.....</b>	<b>78</b>
2.4.1.1	Proyecto.....	78
2.4.1.2	Proceso. ....	79

2.4.1.3	Administración.....	79
2.4.1.4	Administración de requerimientos.....	79
2.4.1.5	Modelos de ciclo de vida.....	79
2.4.1.6	Identificar el modelo del ciclo de vida.....	80
<b>2.4.2</b>	<b>El contexto de la administración de proyectos.....</b>	<b>82</b>
2.4.2.1	Involucrados en el proyecto.....	82
2.4.2.2	Influencias organizacionales.....	83
2.4.2.3	Influencias Generales.....	83
<b>2.4.3</b>	<b>Áreas de conocimiento de la administración de proyectos.....</b>	<b>84</b>
2.4.3.1	Integración.....	84
2.4.3.2	Alcance.....	84
2.4.3.3	Tiempos.....	86
2.4.3.4	Costos.....	86
2.4.3.5	Calidad.....	87
2.4.3.6	Personal.....	88
2.4.3.7	Comunicaciones.....	88
2.4.3.8	Riesgos.....	89
2.4.3.9	Procuración.....	90
<b>2.5</b>	<b>El Sistema de Información del Iteso.....</b>	<b>91</b>
<b>2.5.1</b>	<b>El Iteso.....</b>	<b>91</b>
2.5.1.1	Datos generales.....	91
2.5.1.2	Reseña histórica.....	91
2.5.1.3	Organigrama.....	92
<b>2.5.2</b>	<b>La Oficina de Servicios de Información del Iteso.....</b>	<b>93</b>
2.5.2.1	Datos generales.....	93
2.5.2.2	Reseña histórica.....	93
2.5.2.3	Organigrama.....	94
<b>2.5.3</b>	<b>El Sistema de Información del Iteso.....</b>	<b>94</b>
2.5.3.1	Conceptos básicos.....	94
2.5.3.2	Áreas del Sistema de Información del Iteso.....	94
<b>2.6</b>	<b>El módulo de reportes históricos del desempeño docente del Iteso.....</b>	<b>95</b>
<b>2.6.1</b>	<b>La evaluación del desempeño académico.....</b>	<b>95</b>

2.6.1.1	Conceptos básicos. ....	95
2.6.1.2	Paradigmas didácticos y evaluación docente. ....	97
2.6.1.3	Indicadores más usados en la evaluación de la práctica docente. ....	98
2.6.1.4	Confiabilidad de las encuestas de evaluación del desempeño docente. ....	99
2.6.1.5	Validez de las encuestas de evaluación del desempeño docente. ....	100
<b>2.6.2</b>	<b>La evaluación del desempeño académico en el Iteso. ....</b>	<b>101</b>
2.6.2.1	Datos generales. ....	101
2.6.2.2	Reseña histórica. ....	103
2.6.2.3	El módulo de reportes históricos de la evaluación A3. ....	105
2.6.2.4	Estado actual. ....	105
<b>3</b>	<b>PROYECTO DE INGENIERÍA DE REQUERIMIENTOS. ....</b>	<b>107</b>
<b>3.1</b>	<b>El Plan de administración del proyecto de Ingeniería de Requerimientos. ....</b>	<b>107</b>
<b>3.1.1</b>	<b>Plan de integración. ....</b>	<b>107</b>
3.1.1.1	Previos. ....	107
3.1.1.2	Desarrollo del plan del proyecto. ....	107
3.1.1.3	Control general de cambios. ....	108
<b>3.1.2</b>	<b>Plan de administración del alcance. ....</b>	<b>109</b>
3.1.2.1	Justificación. ....	109
3.1.2.2	Metodología. ....	110
3.1.2.3	Alcance. ....	110
3.1.2.4	Productos a entregar. ....	110
3.1.2.5	Estructura desglosada de trabajos. ....	111
3.1.2.6	Criterios de éxito. ....	111
3.1.2.7	Factores de éxito. ....	112
3.1.2.8	Verificación del alcance. ....	112
3.1.2.9	Control de cambios del alcance. ....	112
<b>3.1.3</b>	<b>Plan de administración del tiempo. ....</b>	<b>113</b>
3.1.3.1	Diagrama de Red. ....	113
3.1.3.2	Diagrama de Gantt. ....	116
<b>3.1.4</b>	<b>Plan de administración del costo. ....</b>	<b>117</b>
3.1.4.1	Requerimiento de recursos. ....	117
3.1.4.2	Estimación de costos. ....	118

3.1.4.3	Plan de administración de la variación de costos.....	118
<b>3.1.5</b>	<b>Plan de administración de calidad.....</b>	<b>119</b>
3.1.5.1	Políticas de calidad.....	119
3.1.5.2	Estándares o modelos a aplicar.....	119
3.1.5.3	Actividades de calidad.....	119
<b>3.1.6</b>	<b>Plan de administración del personal.....</b>	<b>119</b>
3.1.6.1	Asignaciones de roles y responsabilidades.....	119
3.1.6.2	Plan de administración del personal.....	123
3.1.6.3	Organigrama.....	123
3.1.6.4	Directorio del equipo de proyecto.....	123
<b>3.1.7</b>	<b>Plan de administración de comunicaciones.....</b>	<b>124</b>
<b>3.1.8</b>	<b>Plan de administración de riesgos.....</b>	<b>126</b>
3.1.8.1	Matriz de Estimación de Riesgos.....	127
3.1.8.2	Matriz de Administración de Riesgos.....	128
<b>3.1.9</b>	<b>Plan de administración de la procuración.....</b>	<b>131</b>
<b>3.2</b>	<b>Requerimientos del proyecto.....</b>	<b>132</b>
<b>3.2.1</b>	<b>Requerimientos del Sistema.....</b>	<b>132</b>
3.2.1.1	Requerimientos Funcionales del Sistema.....	132
3.2.1.2	Requerimientos no Funcionales del Sistema.....	133
<b>3.2.2</b>	<b>Diseño del sistema.....</b>	<b>135</b>
3.2.2.1	Diagrama de Flujo de Datos - 0.....	135
3.2.2.2	Diagrama de Flujo de Datos - 1.....	136
3.2.2.3	Diagrama de Flujo de Datos - 2.....	137
3.2.2.4	Diagrama de Flujo de Datos - 3.....	138
3.2.2.5	Diagrama de flujo de Datos - 4.....	139
<b>3.2.3</b>	<b>Requerimientos de software.....</b>	<b>141</b>
3.2.3.1	Requerimientos Funcionales de Software.....	141
3.2.3.2	Requerimientos no Funcionales de Software.....	142
<b>3.2.4</b>	<b>Casos de Uso.....</b>	<b>147</b>
3.2.4.1	CU1RepNumA3.....	147
3.2.4.2	CU2RepComA3.....	149
3.2.4.3	CU3RepHistA3.....	151
3.2.4.4	CU4RepDepA3.....	153



3.2.4.5	CU5RepProgA3 .....	155
<b>3.2.5</b>	<b>Modelado de Casos de Uso .....</b>	<b>157</b>
<b>3.2.6</b>	<b>Formatos .....</b>	<b>160</b>
<b>3.3</b>	<b>Conclusiones.....</b>	<b>161</b>
<b>BIBLIOGRAFÍA .....</b>		<b>163</b>
<b>ANEXOS.....</b>		<b>168</b>

## 1 INTRODUCCIÓN

El proceso<sup>1</sup> de desarrollo de requerimientos para los sistemas de información se encuentra descrito y conformado de diferentes maneras, sin que exista un consenso generalizado, este trabajo es un estudio sobre ingeniería de requerimientos entendida como desarrollo y administración de requerimientos, para sistemas de información, basado en una metodología compuesta por los estándares, las organizaciones y los autores encontrados sobre el tema.

El área de sistemas de información emplea una gran cantidad de recursos humanos y materiales para satisfacer las necesidades cambiantes y en aumento de las organizaciones, pero sus esfuerzos no siempre resultan exitosos y es frecuente que los usuarios se muestran decepcionados con los productos que reciben, porque no satisfacen adecuadamente sus necesidades debido a factores técnicos, administrativos y de comportamiento de los cuales el aspecto técnico es el menos problemático. En realidad el problema estriba en que los usuarios no entienden sus necesidades y la gente de sistemas por desconocimiento, salvo contadas excepciones, no los pueden guiar para determinar con precisión qué es lo que debe hacer el sistema. (Towfighian 1998).

El proceso de desarrollo de requerimientos es quizás el más difícil, crítico y propenso a errores del proceso de desarrollo de software. Esta tesis propone un proceso para desarrollar requerimientos de software y un plan para administrar este desarrollo, aplicado a un módulo del sistema informativo de la Universidad Iteso, con dos enfoques principales:

---

<sup>1</sup> El término proceso se encuentra definido en la sección 2.4.1.3

- a) Utilizar los estándares y las mejores prácticas encontradas que mejor se adapten al contexto de la Universidad ITESO
- b) Desarrollo de requerimientos con una participación activa de los usuarios.

En base al proceso y al plan propuesto se obtienen los requerimientos para desarrollar el software del programa de reportes históricos de la Evaluación del Desempeño Docente (EDD), de la encuesta de evaluación de los alumnos/as a los profesores/as en la actividad de dar clases (A3), que forma parte del Sistema de Evaluación del Desempeño Académico (SEDA) y este a su vez del Sistema de Información del Iteso.

En los hallazgos y conclusiones se mencionan los problemas y aciertos obtenidos al aplicar el proceso y el plan, que servirán de base para una propuesta de estandarización del proceso de desarrollo de requerimientos para los proyectos de desarrollo de software de nuestra universidad.

## 1.1 Objetivo

Generar un proceso estandarizado de desarrollo de requerimientos de software, para lograr los beneficios descritos por Wiegers (Wiegers 1999) y por el estándar IEEE830 (IEEE830 1998).

- a) Lograr mayor satisfacción de los clientes en cuanto a funcionalidad y usabilidad, estableciendo bases para acuerdos con los clientes y los proveedores sobre lo que el software debe hacer.
- b) Reducir esfuerzo y costos de desarrollo, mantenimiento y soporte.
- c) Mejorar la calidad de los requerimientos del proyecto en una etapa temprana del ciclo de desarrollo lo cual reduce el retrabajo y mejora la productividad.
- d) Alcanzar los objetivos programados controlando el deslizamiento del alcance y los cambios en los requerimientos, proveyendo una línea base para validación y verificación.

## 1.2 Metodología.

Esta tesis es un estudio que selecciona las mejores prácticas propuestas por los estándares, las organizaciones y los autores encontrados que han escrito sobre la materia; que **se adaptan** al contexto de nuestra Universidad.

La búsqueda bibliográfica se realizó en Internet, en la base de datos EBSCOHost, en el acervo de la biblioteca de la Universidad Iteso y en el material bibliográfico disponible de los profesores y compañeros de la maestría.

También se realizó una búsqueda de campo sobre los sistemas de evaluación de otras universidades de la Cd. de México y locales.

La conformación del proceso se realizó con los pasos que aparecían mencionados en forma constante en la bibliografía, más los que según el análisis se determinó que era conveniente incluir.

Las dos áreas de conocimiento, con las que se inicia el desarrollo del marco teórico -Ingeniería de Sistemas e Ingeniería de Software- son los medios en los que está inserta y se relacionan directamente con esta investigación.

### **1.3 Alcance**

- a) El desarrollo de los requerimientos.
- b) El plan para administrar el desarrollo de los requerimientos anteriores.

Ambos alcances están referidos los reportes históricos de las evaluaciones A3 de la EDD para profesores de licenciatura, del sistema de información de la Universidad Iteso.

### **1.4 Estructura de la tesis.**

El capítulo 2 es la revisión bibliográfica dividida en seis grandes temas:

- a) La Ingeniería de Sistemas de Información.
- b) La Ingeniería de Software.
- c) La Ingeniería de Requerimientos.
- d) La Administración de Proyectos de Ingeniería de Requerimientos de Software.
- e) El Sistema de Información del Iteso.
- f) El módulo A3 de la evaluación del desempeño académico en el Iteso.

El capítulo 3 muestra los resultados de la tesis.

- a) El plan para administrar el proyecto de Ingeniería de Requerimientos.
- b) Los requerimientos.
- c) Las conclusiones.

## 2 MARCO DE REFERENCIA

### 2.1 Ingeniería de Sistemas de Información.

#### 2.1.1 Conceptos básicos.

##### 2.1.1.1 Sistema.

Los teóricos actuales de los Sistemas de información han definido un “Sistema” de diferentes maneras: Senn dice que “es un conjunto de componentes que interactúan entre sí para un cierto objetivo. sus elementos son: el medio circundante o entorno, límites o fronteras, entradas, salidas y componentes” (Senn 1990). Hellriegel dice que “es una asociación de partes interrelacionadas e interdependientes” (Hellriegel 1998). Para Murdick “Es un conjunto de elementos organizados que se encuentran en interacción, que busca alguna meta o metas comunes, operando para ello sobre datos o información, sobre energía o materia y organismos en una referencia temporal para producir como salida información o energía u organismos (Murdick 1998). Stair dice que “es un conjunto de elementos o componentes que interactúan entre sí para cumplir metas .Los sistemas poseen ; entradas, procesamiento, mecanismos, salidas y retroalimentación” (Stair 2000).

En base a lo anterior tenemos que un **Sistema es un conjunto de componentes que interactúan entre sí para un cierto objetivo y que están definidos por un medio circundante, tienen límites y constan de entradas, mecanismos, procesamiento, salidas y retroalimentación.**

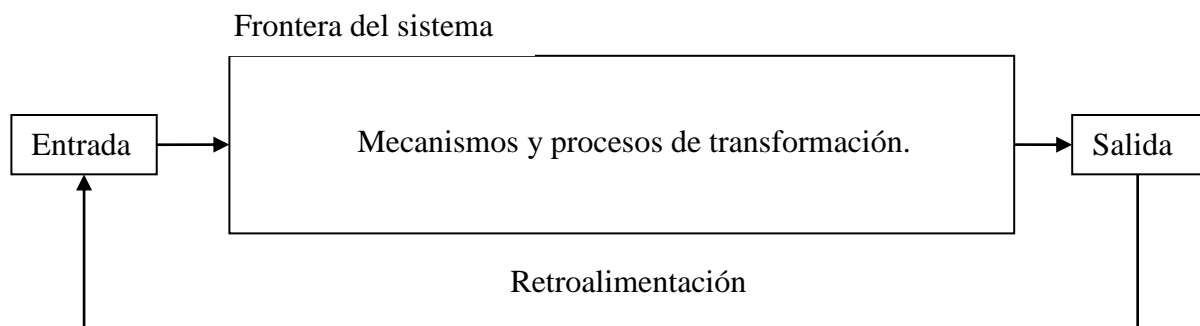


Fig. 2.1.1.1 Modelo de Sistema.<sup>2</sup>

<sup>2</sup> Nota: En este trabajo, cuando no se indica explícitamente la fuente, la tabla es original del autor.

Entre de los procesos que conforman el sistema debe existir un medio de comparación entre los niveles de desempeño actuales y los niveles aceptables de desempeño denominados estándares. La información proporcionada al hacer esta comparación junto con el proceso de reportar las diferencias, recibe el nombre de retroalimentación.

#### **2.1.1.1.1 Dimensiones que conforman el marco de trabajo de un sistema.**

- a) Capacidad: su efectividad para lograr el objetivo.
- b) Funciones: lo que hace el sistema.
- c) Políticas: las reglas básicas que tiene el sistema.
- d) Fases: la naturaleza temporal (dinámica) del sistema.
- e) Medio ambiente: qué cosas fuera del sistema lo afectan.
- f) Recursos: de que está compuesto el sistema.
- g) Costo x tiempo: el precio del sistema.

Si estamos al tanto de estas dimensiones podemos discernir más rápidamente cuáles elementos del sistema interactúan más fuertemente y tener pistas sobre qué investigar primero, también nos ayuda con los defectos arquitectónicos (Ej. la mala elección de un candidato para una de las dimensiones). (Westerman 2001)

#### **2.1.1.2 Ingeniería**

La Ingeniería puede ser vista como un circuito cerrado de retroalimentación, que consta de procesos o actividades relacionadas realizadas en respuesta a un acuerdo de necesidades, estos procesos consumen recursos y producen productos. Para administrar o mejorar los procesos hay que ejercer control. Los controles son mecanismo de toma de decisiones que consideran las metas y las restricciones en la formulación de acciones intencionadas a dirigir o modificar el proceso. La decisión de tomar la acción se basa en medidas, evidencia cuantitativa con respecto al estado del proceso. Las medidas pueden ser hechas de las condiciones interiores del proceso, los productos del proceso y de la satisfacción de los usuarios de los productos. (Moore 1998)

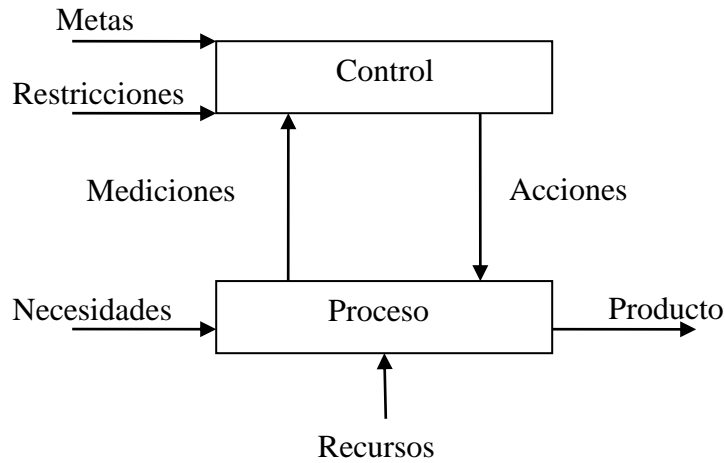


Fig. 2.1.1.2 Un modelo de ingeniería (Moore 1998)

### 2.1.1.3 Ingeniería de sistemas.

Es el proceso que identifica las características técnicas y las reglas de operación de un sistema que mejor logran sus objetivos.(Westerman 2001)

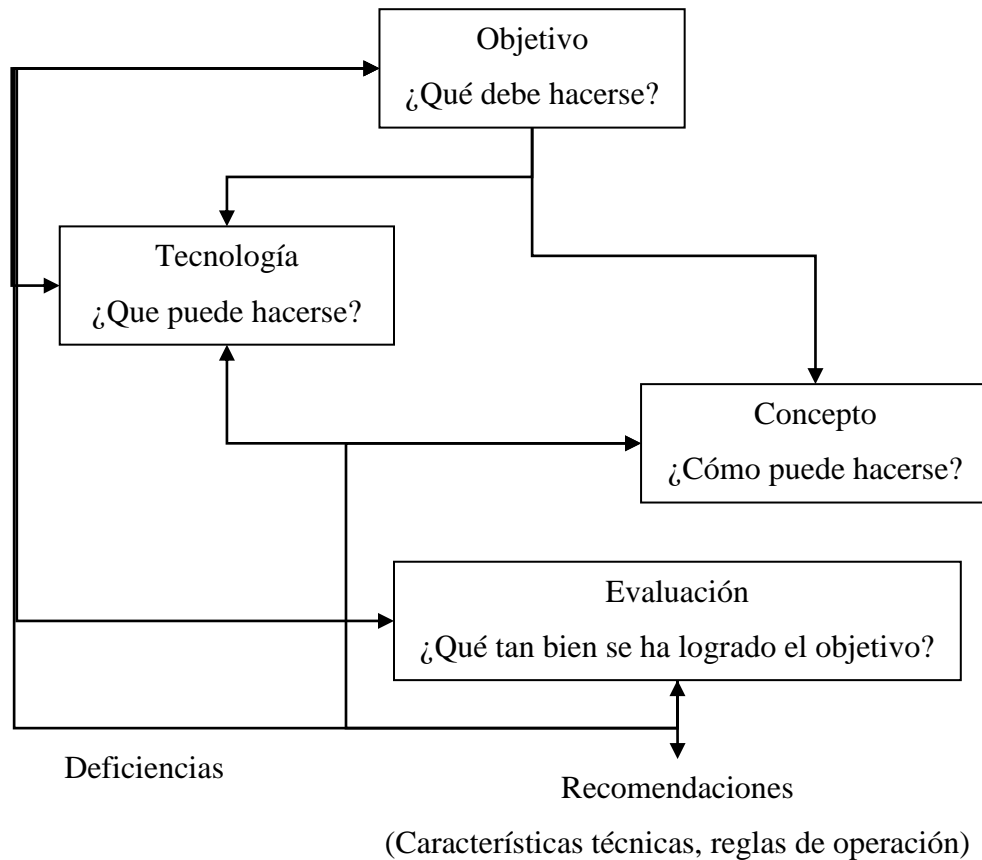


Fig. 2.1.1.3 Proceso de alto nivel de Ingeniería de Sistemas (Westerman 2001)

Esta definición implica un punto de partida en los objetivos del sistema y abarca una solución tecnológica y la evaluación de esta tecnología no solamente en términos de las posibles alternativas de personas, hardware o software, sino también en términos de como se desempeñan todos esos elementos tomados en forma conjunta. Para hacer esto último debemos saber como interactúan estos elementos; esto es que debemos tener un concepto de trabajo de lo que hace el sistema para alcanzar su objetivo.

La evaluación implica la existencia de criterios o medidas de que tan bien se ha alcanzado el objetivo. Existen dos salidas del proceso, una va fuera de él, en forma de recomendaciones para un diseño detallado del sistema y la otra salida es interna al proceso para retroalimentar sus deficiencias y mejorarlo.

Leffingwell (Leffingwell 2000) la define como un enfoque y medios interdisciplinarios para capacitar la realización de sistemas exitosos. Hace énfasis en la definición temprana en el ciclo de vida de las necesidades del cliente y la funcionalidad requerida; en la documentación de los requerimientos, en el procesamiento una síntesis de diseño y una validación de sistema mientras se considera el problema completo.

#### **2.1.1.4 Funciones de un ingeniero en sistemas.**

- a) Definir los objetivos del sistema y sus métricas.
- b) Proponer conceptos de sistemas
- c) Seleccionar la tecnología
- d) Evaluar todos los pasos de la ingeniería de sistemas
- e) Analizar las evaluaciones
- f) Recomendar cursos de acción.

#### **2.1.1.5 Información**

Senn dice que es “un conjunto de datos que se presentan en forma inteligible al receptor. Tiene un valor real o percibido para el usuario y se agrega a lo que ya conocía respecto a un suceso o área de interés y debe ser relevante para la situación en la cual se aplicará” (Senn 1990). Stair dice que “es un conjunto de datos organizados de tal manera que adquieren valor adicional más allá del que poseen por sí mismos” (Stair 2000). Posteriormente Haag dio una definición más concisa “son datos que tienen un significado particular, en un contexto específico” (Haag 2002).

Entonces tenemos que **Información es un conjunto de datos organizados de tal manera que adquieren valor adicional real o percibido, en un contexto específico, para un receptor.**

#### **2.1.1.6 Sistemas de Información.**

Para Senn es “un conjunto de personas, datos y procedimientos que funcionan en conjunto. Recibe datos de fuentes internas o externas de la empresa como elementos de entrada. Después actúa sobre los datos para producir información, los procedimientos determinan como se elabora dicha información” (Senn 1990). Alter dice que son “sistemas que utilizan tecnología de información para capturar, transmitir, almacenar, recuperar, manipular o desplegar información utilizada en una o más procesos de negocios” (Alter 1999). Para Stair son “un conjunto de elementos o componentes interrelacionados para recolectar (entrada), manipular (procesamiento) y diseminar (salida) datos e información, que cuenta además con un mecanismo de retroalimentación para el cumplimiento de un objetivo” (Stair 2000). Finalmente Haag los definió como “los encargados de la planeación, el desarrollo, la administración y el uso de herramientas de tecnologías de información para realizar tareas relacionadas con el procesamiento y administración de la información” (Haag 2002).

Conforme a las definiciones anteriores, tenemos que, un **Sistema de información (SI), es un conjunto de personas, datos y procedimientos interrelacionados, que se encargan de la planeación, el desarrollo, la administración y el uso de herramientas de tecnologías de información, para capturar, transmitir, almacenar, recuperar, manipular o desplegar información utilizada en una o más procesos de negocios, que cuenta además con un mecanismo de retroalimentación para el cumplimiento de un objetivo.**



## 2.2 Ingeniería de software.

### 2.2.1 Conceptos básicos.

#### 2.2.1.1 Ingeniería de Software

El IEEE 610.12 define la Ingeniería de Software (IS) como:

Acercamiento sistemático, disciplinado y cuantificable, aplicado al desarrollo, operación y mantenimiento del software. (Moore 1998)

En el ámbito del desarrollo de la Ingeniería de software, se puede hablar de una Ingeniería de requerimientos, tema de este trabajo.

##### 2.2.1.1.1 Relación de la Ingeniería de Software con otras disciplinas.

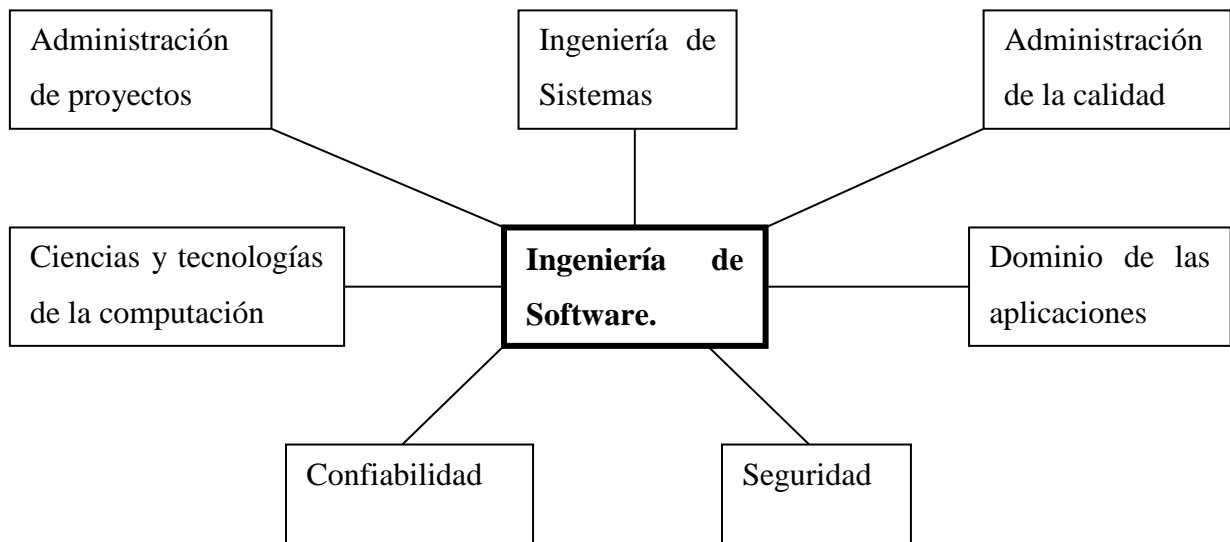


Fig. 2.2.1.1.1 Relación de la Ingeniería de Software con otras disciplinas. (Moore 1998)

## 2.2.2 Proceso de desarrollo de Software.

El proceso de desarrollo de Software se puede subdividir en varias tareas que denominaremos fases, cada fase soluciona diferentes problemas del proceso, tiene sus propios modelos conceptuales, notaciones y heurísticas para ayudar a construir estos modelos.

Existe un amplio acuerdo en la naturaleza de estas fases, pero los nombres y los límites están en disputa. Estas fases aparecen en los diferentes modelos de ciclos de vida que se discuten más adelante en el punto 2.4.2.

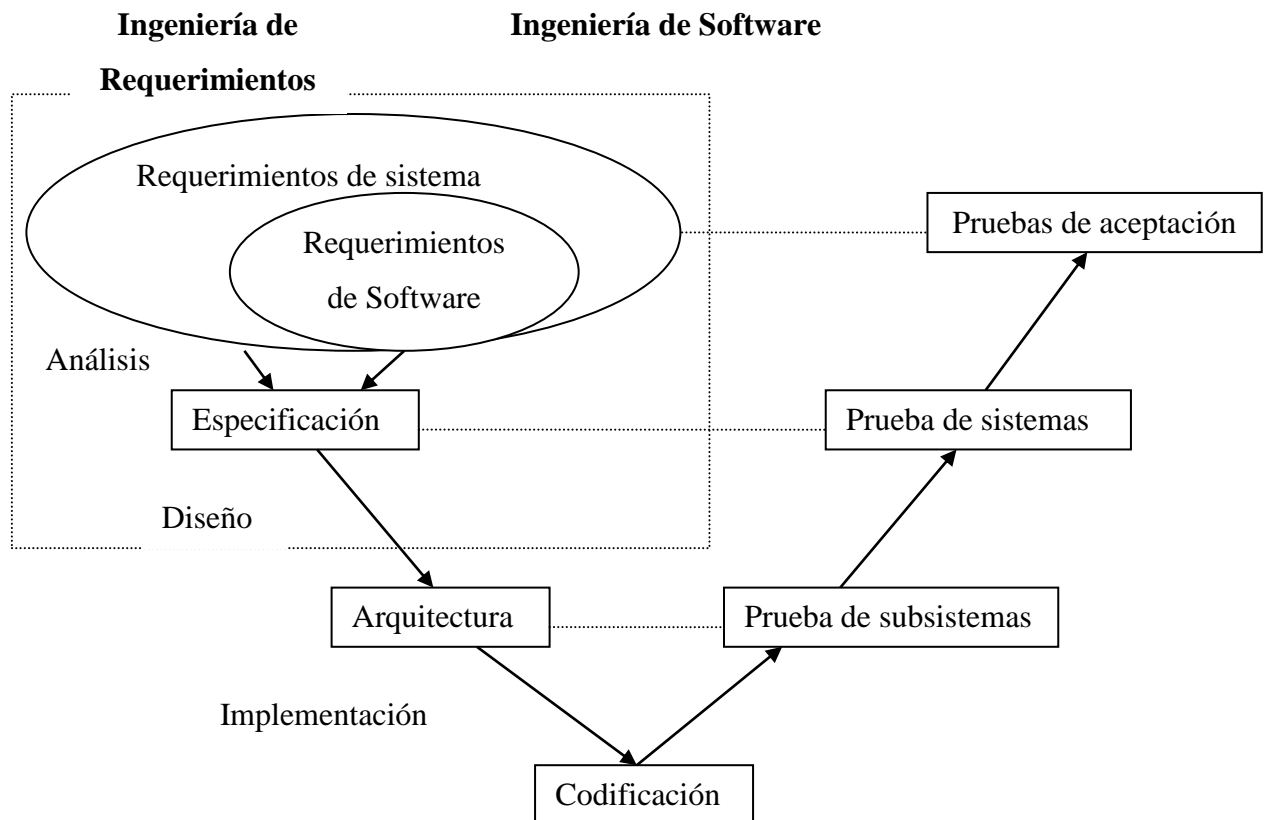


Fig. 2.2.2 Diagrama V de Ingeniería de software.

El ovalo representa los requerimientos originales para el sistema a ser producido, que a menudo no están bajo el control del desarrollador de software sino que deben ser suministrados por los clientes, las cajas representan las salidas de las diferentes fases de desarrollo. Los tiempos de producción corren de izquierda a derecha, el flujo hacia abajo en la izquierda muestra las etapas desde los requerimientos hasta la codificación (producción), y el flujo hacia arriba en la derecha muestra la codificación siendo integrada y probada antes de ser entregada (ensamblado). Las líneas punteadas indican que los resultados de la fase de integración a la derecha están ajustados y deben satisfacer los requerimientos a la izquierda. (Coleman 1994). El área punteada a la izquierda representa el ámbito de la Ingeniería de Requerimientos.

## **2.3 Ingeniería de requerimientos.**

### **2.3.1 Conceptos básicos.**

#### **2.3.1.1 Definición de requerimientos de software.**

Son los problemas de un cliente o del mercado que necesitan resolverse (Whitten 1995)

Los requerimientos son las capacidades o condiciones establecidas por el cliente o el usuario y pueden ser funciones, restricciones u otras propiedades, que deben ser provistas alcanzadas o satisfechas por el sistema (Sodhi 1992).

Los requerimientos son cualquier criterio que un artefacto a ser diseñado debe tener para ser considerado exitoso. La ingeniería es esencialmente salvar el hueco entre los requerimientos y la materia disponible. (Kovitz 1999).

El IEEE<sup>3</sup> Standard Glossary of Software Engineering Terminology (IEEE610.12 1990) define un requerimiento enfatizando tres aspectos:

1. Condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo.
2. Condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, especificación u otro documento impuesto formalmente.
3. Una representación documentada de una condición o capacidad como en 1 y 2.

---

<sup>3</sup>Institute of Electrical and Electronics Engineers, Inc.

### 2.3.1.2 Definición de Ingeniería de requerimientos.

Dentro del contexto de la Ingeniería de sistemas y de la IS, tenemos a la Ingeniería de Requerimientos (IR), Según Thayer (Thayer 1997) es ciencia y la disciplina a la cual le concierne el establecimiento y la documentación de los requerimientos de software. Consiste en el hallazgo, análisis, especificación y administración de los requerimientos.

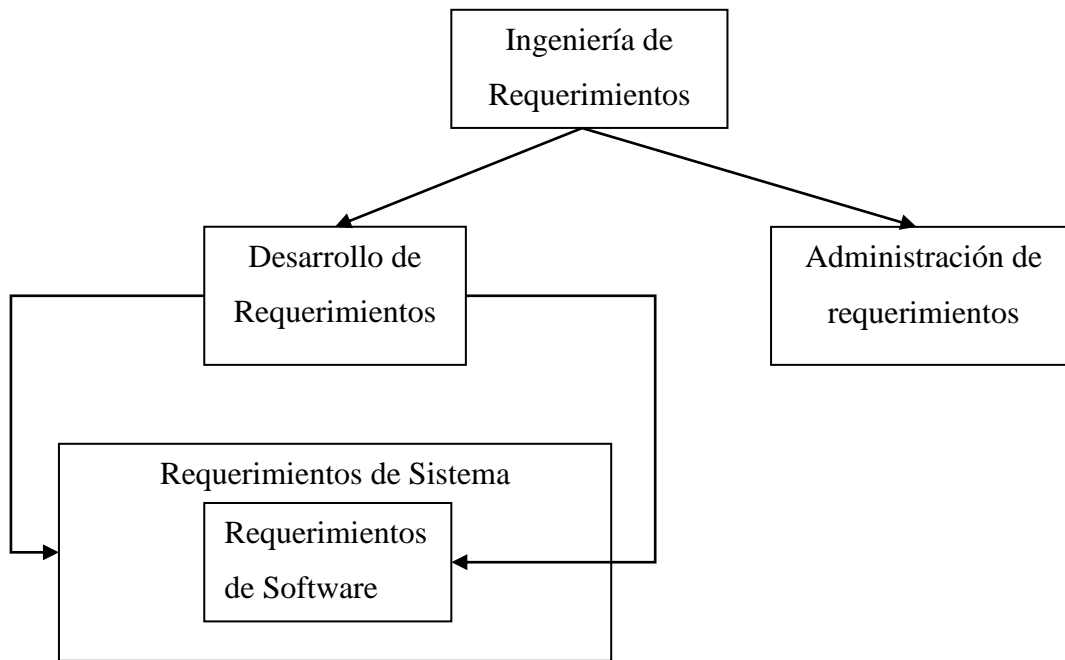


Fig. 2.3.1.2.1 Descomposición jerárquica del dominio de la Ingeniería de Requerimientos.

Wiegiers (Wiegiers 1999) y otros autores dividen la IR en dos: Desarrollo de requerimientos (DR) y Administración de Requerimientos (AR), Faulk citado en (Dorfman 1997) dividen a su vez el DR en dos: Requerimientos de Sistema y Requerimientos de Software, que se describen en los puntos 2.3.1.2.1 y 2.3.1.2.2 respectivamente, la necesidad de esta división se explica en el punto 2.3.3.1. La AR se tratará en el punto 2.4

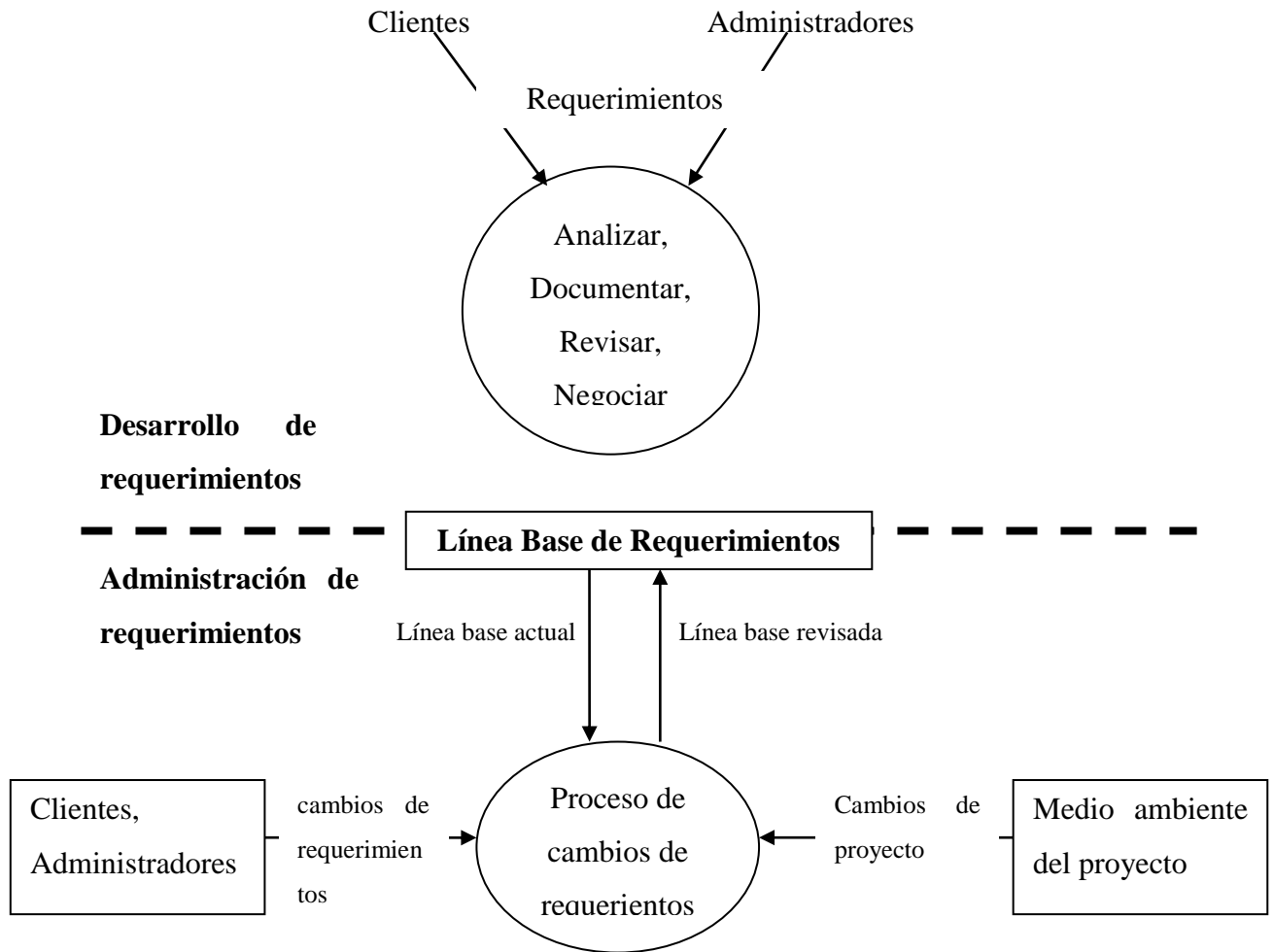


Fig.2.3.1.2.2 Límites entre desarrollo y administración de requerimientos en un Sistema de Información.

### 2.3.1.2.1 Ingeniería de Requerimientos de Sistema.

La Ingeniería de Requerimientos de Sistema (IRS), se encarga de descubrir, analizar y describir funciones y capacidades de los sistemas; requerimientos del negocio, organizacionales y del usuario; requerimientos de protección, seguridad, ingeniería de factores humanos (ergonómicos), interfaz, operaciones y de mantenimiento; restricciones de diseño y requerimientos de calidad. (Hernández 2001)

Cuando el software es creado en el contexto de un contexto de un gran sistema de hardware y software, se definen primero los requerimientos del sistema y luego los de diseño. (Thayer 1997)

### 2.3.1.2.2 Ingeniería de Requerimientos de Software

La Ingeniería de Requerimientos de Software (IRSW) es el arte de explotar la configurabilidad extraordinaria de las computadoras, o el arte de inventar caminos útiles para configurarlas (Kovitz 1999).

Le concierne específicamente la obtención, recolección, el análisis, documentación y verificación de los requerimientos de software. Incluye la transformación de los requerimientos del sistema en una descripción de requerimientos de software, parámetros de desempeño y configuración del sistema, a través del uso iterativo de un proceso de definición, análisis, estudios costo / beneficio y prototipos (Hernández 2001)

### 2.3.2 Relación de los Requerimientos con otros procesos.

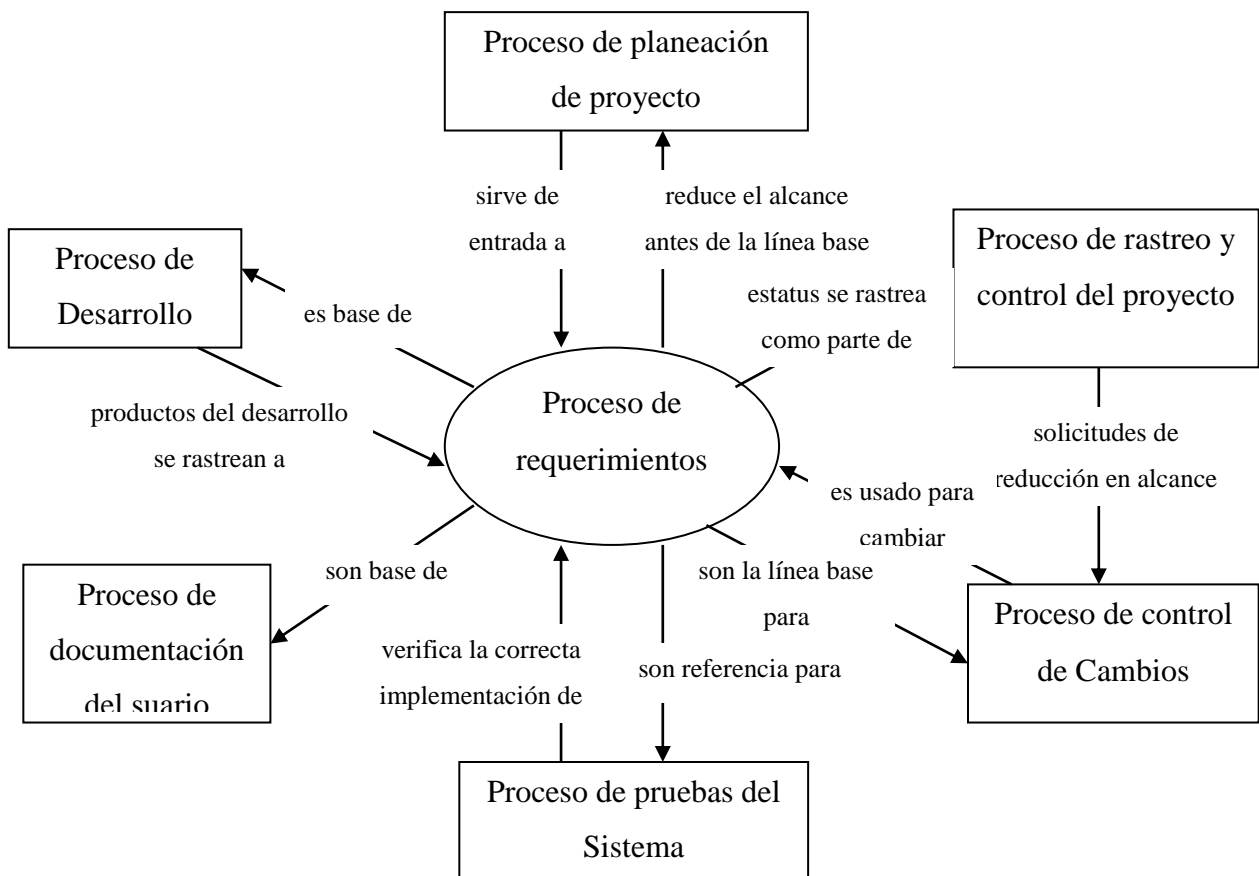


Fig. 2.3.2 Relación de los Requerimientos con otros procesos del proyecto. Modelo adaptado de Wiegers (Wiegers 1999).

Los requerimientos son el corazón de los proyectos de software, le dan soporte a muchas de las otras actividades técnicas y administrativas. Las interfaces entre los requerimientos y los otros procesos se describirán más adelante en este trabajo.

### **2.3.3 Clasificación de requerimientos.**

Existen varias formas de hacer una taxonomía de los requerimientos, se exponen las que se encontraron de uso más frecuente en la actualidad, con anotaciones sobre las variaciones de los términos utilizados.

#### **2.3.3.1 Clasificación por niveles de requerimientos.**

Los niveles están considerados según la jerarquía de los objetivos que representan, por lo descrito en Faulk y Kirakowski citados por Hernández (Hernández 2001), Wiegers (Wiegers 1999) y Sommerville (Sommerville 2002) se pueden distinguir 3 niveles:

1. Requerimientos del negocio u Organizacionales. Representan los objetivos de más alto nivel de la organización o de los clientes, que se requieren del sistema o producto. Se capturan en un documento que describe la visión y alcance del proyecto. También pueden describir la factibilidad del sistema desde un punto de vista comercial.
2. Requerimientos de los usuarios y especificación funcional. Describen las tareas que los usuarios tienen que llevar a cabo con el producto. Incluyendo las funciones requeridas para soportar estas funciones, las interfases usuario sistema, soporte a los usuarios, equipamiento y hardware. También incluye las metas de usabilidad que se tienen que alcanzar y el enfoque para la instalación del sistema. Se capturan en casos de uso o descripciones de escenarios.
3. Requerimientos técnicos del sistema. Especifican cómo el sistema logrará las funciones requeridas y la estructura de los datos que tienen que estar disponibles para que el procesamiento interno sea exitoso. También especifican las restricciones técnicas tales como la velocidad mínima de comunicación en la red.

#### **2.3.3.2 Clasificación por tipos de requerimientos.**

La clasificación por tipos de requerimientos no implica jerarquía. Faulk citado en (Dorfman 1997) diferencia dos tipos de especificaciones:

- 1.- De sistema. Que son los requerimientos funcionales y capacidades del sistema, requerimientos del negocio y del usuario; protección seguridad ingeniería de factores humanos

(ergonómicos) interfaz, operaciones y requerimientos de mantenimiento, restricciones diseño y requerimientos de calidad, que se refieren al sistema en general y se describen en la sección 2.3.3.2.

2.- De software. Que son parte de los requerimientos anteriores, pero que se refieren exclusivamente a los componentes de software del sistema.

Gunter (Gunter 2000) y otros separan los requerimientos de software en dos categorías:

- a) Especificaciones de requerimientos para los usuarios. Para aquellos que encargan la comisión, pagan o la utilizan.
- b) Especificaciones de requerimientos de software para los programadores.

La distinción de ambos tipos de especificaciones de sistema y de software es importante porque compete a dos especialistas diferentes:

- a) Ingenieros en sistemas, generalmente con limitados conocimientos de software.
- b) Ingenieros en software, que generalmente rechazan involucrarse en ingeniería en sistemas.

De aquí que sea necesario preparar los Ingenieros de Requerimientos en ambos campos para que puedan cumplir adecuadamente sus funciones.

### **2.3.3.3 Subdivisión por tipos de requerimientos.**

Ambos tipos de especificaciones se dividen a su vez según el tipo de requerimientos. Ross citado en Sodhi (Sodhi 1992) dice que hay cuatro tipos de requerimientos:

1. Funcionales, que especifican la función que el software debe de ser capaz de realizar, que pueden ser vistos desde una perspectiva dinámica o desde una estática.
2. No funcionales, los relacionados con el desempeño, confiabilidad, seguridad, mantenimiento, disponibilidad, exactitud, manejo de error, capacidad, habilidad para ser utilizados por usuarios específicos, etc.
3. Inversos, describen las restricciones de un comportamiento permitido.
4. Restricciones de diseño e implementación, son las condiciones límite, con las que tiene que ser construido e implementado el software.



Faulk citado en Thayer (Thayer 1997) había dicho, que la clasificación de requerimientos en funcionales y no funcionales ofrece poca ayuda en la comprensión de atributos comunes de diferentes tipos de requerimientos.

Por ejemplo un requerimiento del usuario sobre seguridad podría parecer uno no funcional. Sin embargo cuando se desarrolla en detalle, conduce a otros requerimientos que son claramente funcionales, como la necesidad de incluir en el sistema los recursos para la autorización del usuario. (Sommerville 2002)

Faulk prefirió hacer la clasificación como:

3.- Requerimientos de comportamiento: que incluyen toda la información necesaria para determinar si el comportamiento en el tiempo de corrido de una implementación dada es aceptable. Los requerimientos de comportamiento definen todas las restricciones en las salidas del sistema (por ejemplo: valores, exactitud y tiempo), y estados del sistema resultantes para todas las posibles entradas y estados actuales del sistema. Con esta definición la seguridad, el desempeño, el tiempo y la tolerancia a los errores son todos requerimientos de comportamiento.

4.- Requerimientos de desarrollo de calidad: incluyen cualquier restricción en los atributos de la construcción estática del sistema, tales como su capacidad de ser probados, cambiados, mantenidos y reutilizados.

En el estado actual de la práctica la clasificación se sigue haciendo en funcionales y no funcionales y se debe recordar que ésta es una distinción artificial.

Nissen y otros (Nissen 1996) hablan de la necesidad de capturar requerimientos de fuentes múltiples de información, en donde los conflictos entre las perspectivas de diferentes involucrados, se utilicen productivamente para obtener otros requerimientos.

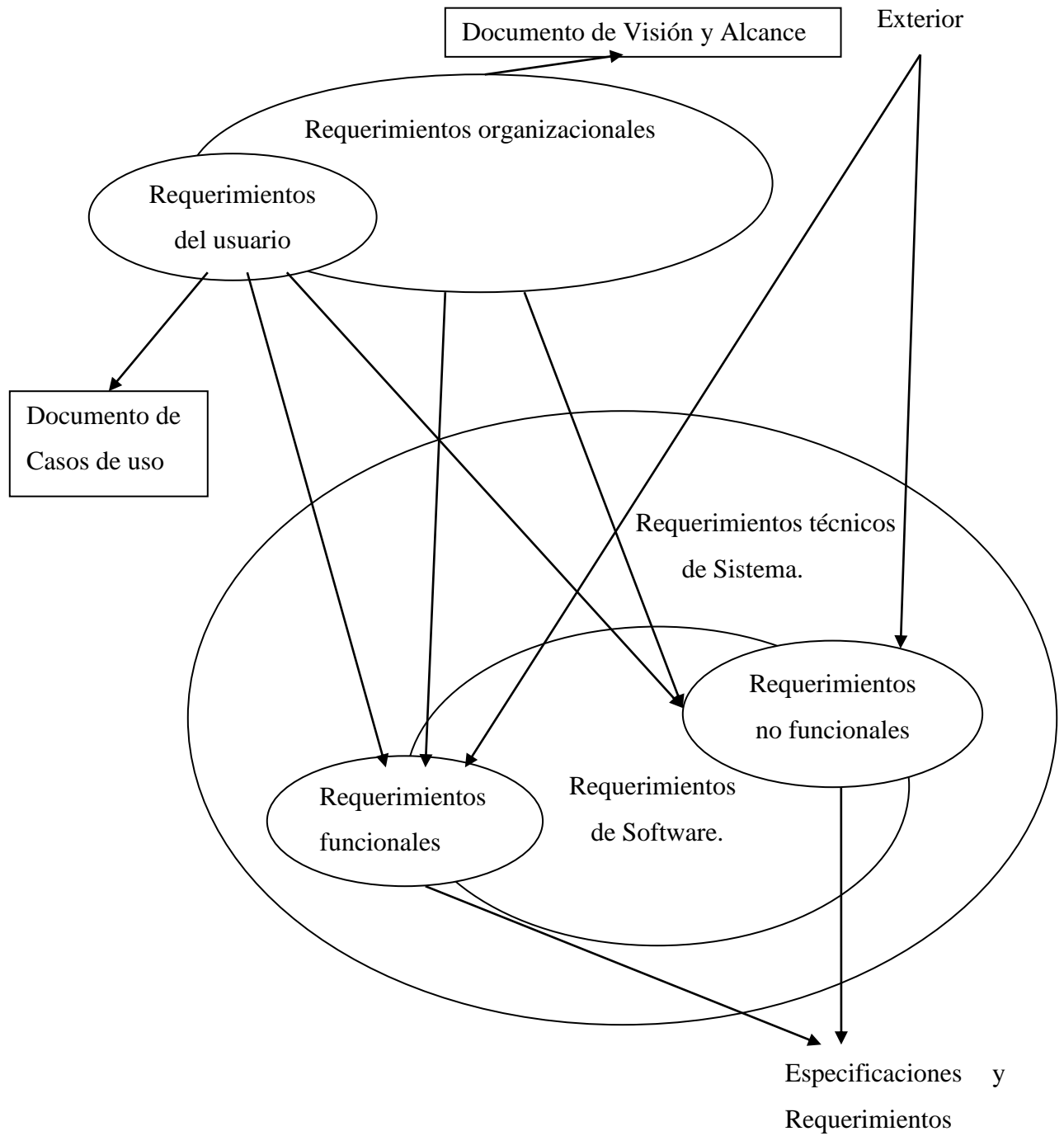


Fig. 2.3.3.2.1 Relación de varias clasificaciones de los Requerimientos.

Sommerville (Sommerville 2002) describe los requerimientos no funcionales más a detalle y los subdivide en tres: requerimientos del proceso (producto), requerimientos de la organización y requerimientos externos

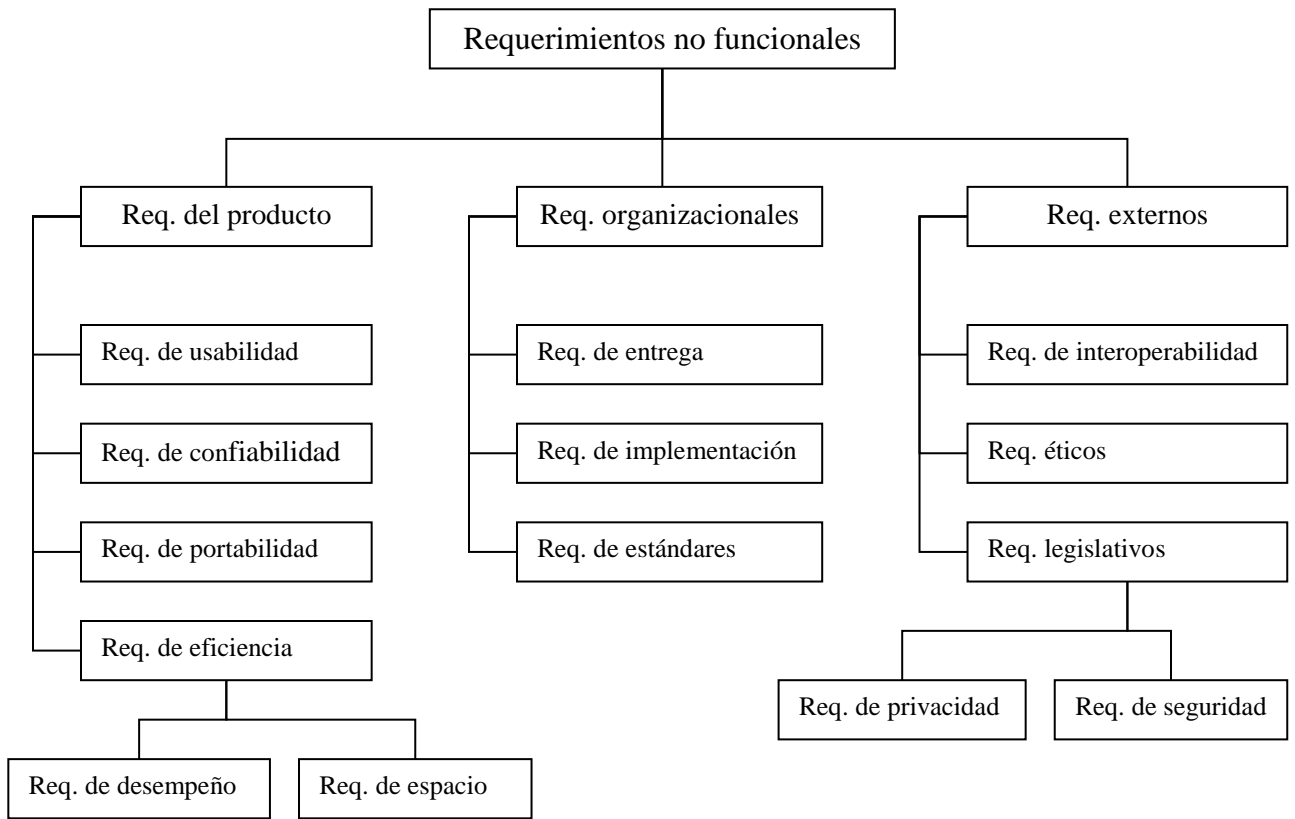


Fig. 2.3.3.2.2 Tipos de requerimientos no funcionales (Sommerville 2002)

- a) Los requerimientos del producto: son los que resultan de la necesidad de entregar un producto de una forma particular. Entre los requerimientos del producto se encuentran:
  - a. Requerimientos de usabilidad: relacionados con el esfuerzo para usar el software, tanto durante su aprendizaje como durante su operación. Ejemplos: familiarización, preparación de entradas, ejecución, interpretación de salidas e interfaz consistente.
  - b. Requerimientos de confiabilidad: expresan el alcance por el cual el software se ejecutará sin fallas dentro de un periodo de tiempo específico, como por ejemplo una tasa aceptable de fracaso.
  - c. Requerimientos de portabilidad: expresan si el software va a ser usado en otro medio, entorno operativo, configuración de hardware o entorno de sistema de software.

- d. Requerimientos de eficiencia: incluyen los requerimientos de desempeño, que se refieren a los factores que describen tanto cualidades objetivas como subjetivas del sistema que expresan medidas de éxito y restricciones bajo las cuales el sistema no puede fallar (como por ejemplo cuán rápido debe ser el sistema al ejecutarse).
- b) Requerimientos organizacionales: son consecuencia de políticas y procedimientos organizacionales. Ejemplos: Estándares de procesos que deben usarse, requerimientos de implementación, tales como lenguaje de programación o métodos de diseño o requerimientos de entrega que se especifiquen cuándo el producto y la documentación serán liberados. Los requerimientos organizacionales pueden ser derivados tanto de la organización cliente, como de la desarrolladora.
- c) Requerimientos externos: Cubren todos los requerimientos que se derivan de los factores externos al sistema y sus procesos de desarrollo. Entre ellos se encuentran:
  - a. Requerimientos de interoperabilidad: definen cómo los sistemas interactúan con otros sistemas en otras organizaciones.
  - b. Requerimientos éticos: los que garantizan la aceptación de los usuarios y del público en general.
  - c. Requerimientos legislativos: los que deben seguirse para garantizar que las operaciones del sistema estén dentro de la ley. Entre ellos tenemos a los requerimientos de privacidad y a los requerimientos de seguridad, que pueden ser físicos (lugares para guardar dispositivos de almacenamiento externos con discos, colocación de cables, desconexión de terminales, etc.) u operacionales (métodos de cifrado o cualquier otro aspecto que afecte la disponibilidad de información sensible)

#### **2.3.4 Profesionales de la IR**

Chang (Chang 1996) menciona cuatro aspectos del perfil de un profesional de la IR:

- a) Habilidades para organizar los requerimientos, para presentarlos en formatos bien estructurados, apropiados para que sean utilizados durante el inicio del proceso de diseño y para que sean comprensibles por el usuario final.
- b) Procesos, debe administrar la suave evolución de los requerimientos
- c) Comunicación, debe saber escuchar y considerar justamente tanto las opciones técnicas como las no técnicas a las que Siddiqui (Siddiqui 1994) denomina perspectiva social, que requiere fundamentar las observaciones en el mundo real.
- d) Tecnología, debe poseer un entendimiento orgánico del dominio de la aplicación.

Están conformados en un equipo multidisciplinario, el número de personas que lo conforman depende del tipo y complejidad del proyecto, de manera general habrá un administrador del proyecto, y varios analistas.

A continuación se describe el papel de los analistas y en la sección 2.4 las actividades que realicen otros miembros del equipo.

#### **2.3.4.1 Analista**

Los analistas son quienes desarrollan los procesos técnicos de requerimientos, se dedican a la recolección (investigación) de datos, al análisis, a la elaboración de las especificaciones y a la verificación. Una vez más, se distinguirán dos tipos: analistas de requerimientos de sistemas y analistas de requerimientos de software.

#### **2.3.4.2 Analista de requerimientos de sistemas.**

Un analista de sistemas estudia un sistema para saber cómo trabaja y dónde es necesario efectuar mejoras. Como normalmente no trabaja en el dominio del problema del usuario, no tiene los mismos conocimientos que ellos y lo primero que debe hacer es comprenderlo. Si tiene experiencia en un área particular y conoce sistemas similares puede anticipar requerimientos para el nuevo sistema. Estudia al sistema actual y descubre requerimientos auxiliándose de técnica, métodos y herramientas.

#### **2.3.4.3 Analista de requerimientos de software.**

Un analista de software se debe basar en los requerimientos del sistema, si éstos existen, para analizar y documentar los requerimientos de software. Debe tener suficientes conocimientos técnicos para poderle comunicar a los diseñadores qué es lo que deben hacer. Esta habilidad no la requieren los analistas de sistemas.

#### **2.3.4.4 Trabajo en equipo.**

Humphrey (Humphrey 1990) dice que la historia del desarrollo de software es de un aumento de escala. Inicialmente unos pocos individuos podían elaborar pequeños programas; el trabajo pronto creció más allá de ellos, entonces se utilizaron equipos de una o dos docenas de individuos, pero con éxito estaba mezclado. Mientras algunas organizaciones han resuelto estos problemas de pequeños sistemas, la escala de nuestro trabajo continúa creciendo. Hoy grandes proyectos requieren típicamente del trabajo coordinado de varios equipos.

Reifer (Reifer 2000) dice que los Ingenieros de Software pueden y deben influenciar los requerimientos y sus cambios, por medio de la participación en el equipo que los define. Para hacer que sus ideas sean aceptadas, los Ingenieros de Software necesitan cambiar su imagen de especialistas por la de generalistas cuyas ideas complementan aquellas de los otros miembros.

### **2.3.5 Proceso de Ingeniería de Requerimientos.**

Para definir el proceso de Ingeniería de Requerimientos se tiene que analizar primero el modelo de ciclo de vida del proyecto de desarrollo de software del sistema de información, este punto se trata más adelante en la sección 2.4.1.5 Este modelo se subdividirá de la siguiente manera:

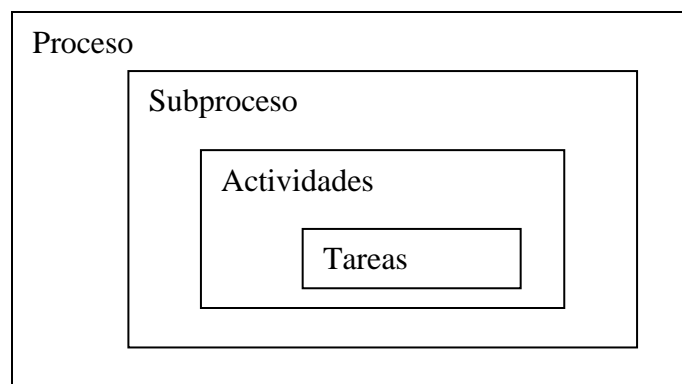


Fig. 2.3.5.1 Subdivisión del proceso

El proceso de Ingeniería de Requerimientos (PIR), utilizado en este trabajo, se apoya principalmente en los procesos descritos por Kovitz (Kovitz 1999), Sommerville (Sommerville 1997), Wieggers (Wieggers 1999) y Hernández (Hernández 2001b)

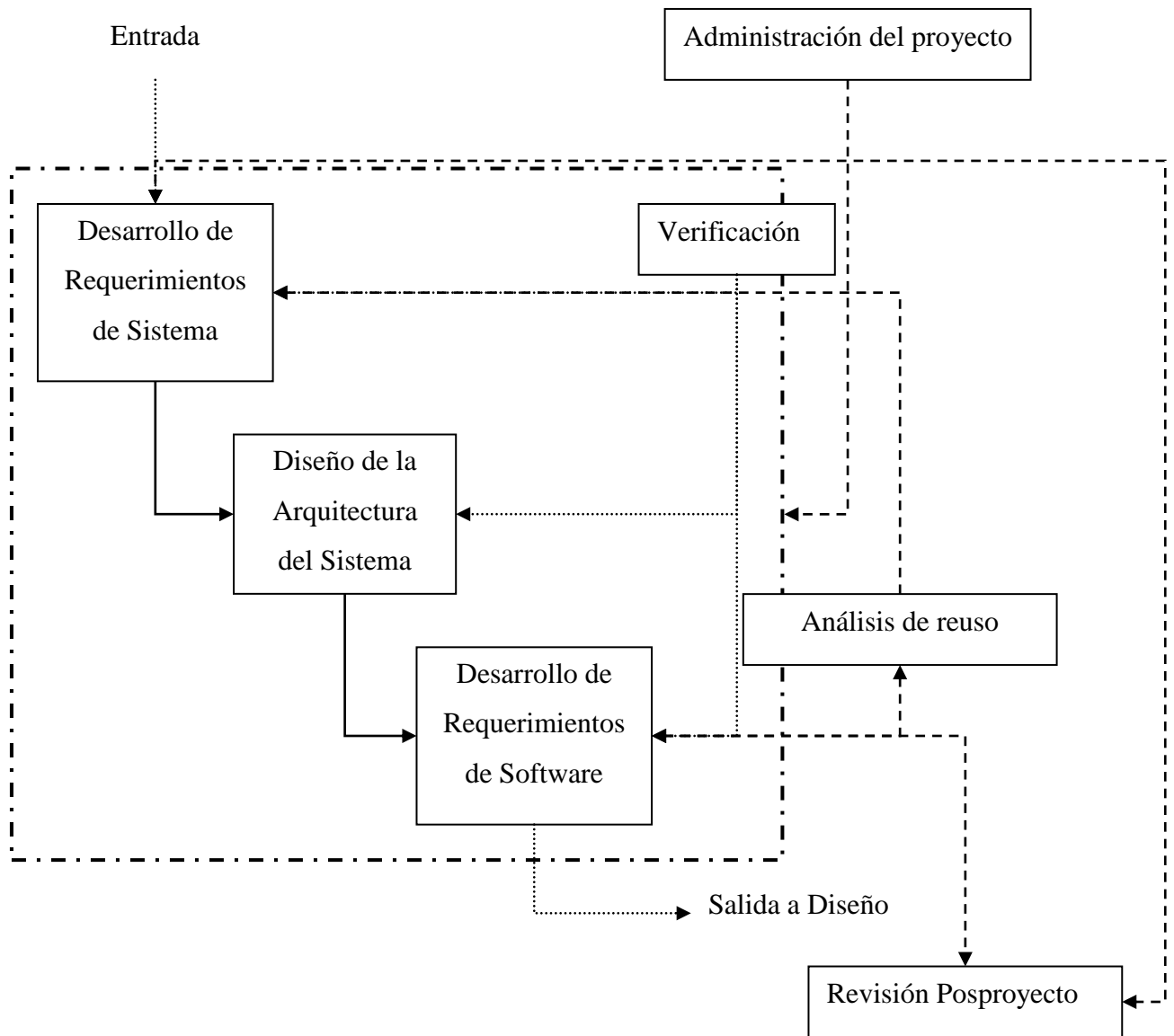


Fig. 2.3.5.2 Proceso de la Ingeniería de Requerimientos.

Los Subprocesos de Desarrollo de Requerimientos fig.13, son un juego de actividades estructuradas que se siguen para obtener<sup>4</sup>, validar y mantener un documento de requerimientos de sistema (Sommerville 1997). Este subproceso es doble porque se divide en dos partes, una para los requerimientos de sistema y otra para los requerimientos de software.

La obtención, el análisis, la especificación y la verificación no tienen lugar en una secuencia lineal clara: estos subprocesos se traslapan y tienen un modelo de ciclo de vida incremental e iterativo, aunque se describan en un modelo en cascada.

<sup>4</sup> Derive

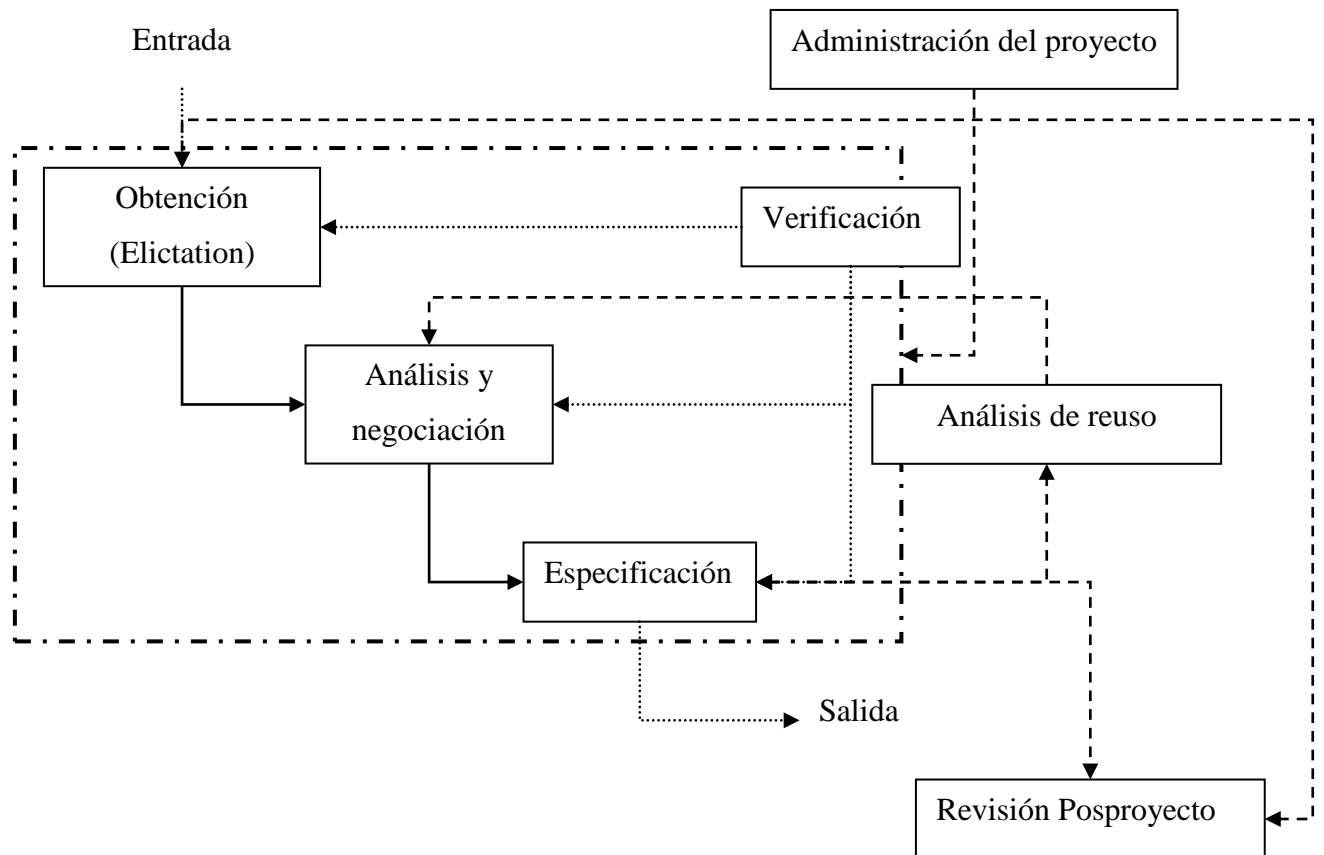


Fig. 2.3.5.3 Subproceso de Desarrollo de Requerimientos.

La descripción completa de cada subproceso incluirá:

- a) El Objetivo
- b) Las entradas
- c) Los participantes (roles)
- d) Actividades y tareas organizadas en un procedimiento, incluyendo los métodos herramientas y técnicas recomendadas.
- e) Salidas.

Se deben definir los activos con que contará la organización para facilitar su proceso de requerimientos. Un activo es una colección entidades mantenidas por una organización para ser usadas por los proyectos en el desarrollo, adaptación, mantenimiento e implementación de sus procesos de software. Estos activos típicamente incluyen: listas de chequeo, ejemplos, políticas, procedimientos, descripción de procesos, plantillas, bases de datos de procesos, etc. (Hernández 2001b)



### **2.3.5.1 Subproceso de Desarrollo de Requerimientos de Sistema.**

#### **2.3.5.1.1 Objetivo.**

Descubrir, analizar y describir las funciones y capacidades del sistema bajo estudio, sus requerimientos del negocio, organizativos, de usuarios, de protección, de seguridad, ergonómicos, de interfaz, de procesos y operacionales, de mantenimiento, así como restricciones de diseño y requerimientos de calidad.

#### **2.3.5.1.2 Entradas.**

- a) Solicitud de la propuesta del cliente.
- b) Procesos de trabajo del negocio.
- c) Estándares.
- d) Plantilla de especificación de requerimientos del sistemas.
- e) Planes de administración del proyecto.

#### **2.3.5.1.3 Participantes (roles).**

Líder del proyecto, analistas, clientes y usuarios

#### **2.3.5.1.4 Procedimiento.**

- a) Obtención de Requerimientos de Sistema.
  - a. Definir el problema.
  - b. Describir el dominio del problema.
  - c. Identificar las fuentes de requerimientos.
  - d. Seleccionar métodos técnicas y herramientas para el desarrollo de requerimientos de sistema.
  - e. Escribir la visión, alcance y las fases de la Estructura desglosada del trabajo (EDT)<sup>5</sup>.
  - f. Identificar las clases de usuarios, caracterizarlos y seleccionar representantes de cada clase.
  - g. Identificar a los que toman las decisiones de requerimientos y sus procesos de toma de decisiones.

---

<sup>5</sup> Work Breakdown Structure

- h. Recolectar requerimientos según tareas colocadas en el EDT del plan de búsqueda y recolección y aplicando los métodos, técnicas y herramientas seleccionados en el paso “d” y que fueron especificados en dicho plan.
- b) Análisis y negociación.
  - a. Dibujar el diagrama de contexto del sistema.
  - b. Crear prototipos de interfase con usuarios.
  - c. Analizar la factibilidad de los requerimientos.
  - d. Asignar prioridades y negociarlas con los clientes.
  - e. Crear un diccionario de datos.
- c) Especificaciones.
  - a. Registrar reglas del negocio.
  - b. Adoptar plantillas de documentos.
  - c. Adoptar un esquema para la identificación de requerimientos.
  - d. Señalar las fuentes de donde se tomaron los requerimientos.
  - e. Documentar requerimientos de sistema funcionales y no funcionales.
  - f. Escribir una Especificación de Requerimientos de Sistema.
  - g. Documentar la arquitectura del sistema y la asignación de componentes del mismo al hardware, software o procesos manuales.
- d) Verificación.
  - a. Definir criterios de aceptación (éxito) para cada requerimiento en interacción con el cliente.
  - b. Realizar inspecciones informales en cada una de las actividades.
  - c. Revisar documentos de requerimientos que se hayan producido.
  - d. Escribir pruebas de caso de los requerimientos.
  - e. Escribir el manual del usuario correspondiente a este subproceso.
- e) Reuso.
  - a. Actualizar biblioteca de requerimientos reusables.

#### **2.3.5.1.4.1 Obtención de Requerimientos de Sistema.**

La actividad de Obtención<sup>6</sup> (Búsqueda y recolección) es quizás la actividad, más difícil crítica y propensa a errores del desarrollo de requerimientos del sistema. Sólo tiene éxito si existe una excelente relación entre los clientes y los desarrolladores. (Hernández 2001b)

##### **2.3.5.1.4.1.1 Definición del problema.**

En Andriole (Andriole 1996), esta tarea aparece como análisis de la solicitud, en la que se responde una pregunta sencilla ¿De que se trata el proyecto?. En este paso se decide si el proyecto está alineado con los objetivos de la organización y con sus estrategias tecnológicas.

Kovitz (Kovitz 1999) dice que para definir problemas, para que la gente los resuelva programando una computadora, requiere proveer toda la información que los programadores y los diseñadores de interfase necesitan para que una computadora genere efectos fuera de ella. También necesitamos entender que hacen los programadores para resolver estos problemas en la vida real.

La definición rigurosa de los requerimientos solo es posible en relación con un patrón específico de diseño. Correspondiendo a cada patrón de diseño, existe un juego de preguntas sobre el tipo de problemas que resuelve el patrón. Un documento de requerimientos responde a estas preguntas.

Todos los problemas de software son de la siguiente forma:

1.- Configura la máquina M para producir los efectos R en el dominio D.

La maquina M es la computadora a ser programada, incluyendo sus mecanismo de entradas y salidas, los efectos R son los requerimientos. El Dominio D es una parte necesaria de la definición del problema, porque es la parte del mundo en términos en los cuales el requerimiento está definido, y porque la máquina raramente puede producir los efectos deseados por sí sola.

El software no es un artefacto tangible, es una configuración particular de la computadora, un estado posible del hardware.

La forma general de todos los problemas de software sugiere tres preguntas fundamentales a responder para escribir requerimientos de software:

- 1.- Qué tipo de máquina se quiere configurar.
- 2.- Qué efectos se quiere que produzca la configuración

---

<sup>6</sup> Elicitation.

3.- Cuáles son las propiedades del mundo exterior que la máquina puede explotar para producir estos efectos.

Algunas veces los requerimientos se contrastan con el diseño, en donde diseño se entiende como la elección de los medios para llevar a cabo un efecto deseado. Los requerimientos son el efecto deseado y las interfaces, código de programa, etc. son los medios para llevarlos a cabo, esto es desafortunado porque los requerimientos son tanto ellos como el diseño.

La invención de los requerimientos es una cuestión de inventar un problema bien definido para resolver. Un problema bien definido es una serie de criterios de acuerdo a soluciones propuestas que o bien resuelve definitivamente el problema o bien definitivamente fallan al resolverlo, junto con información complementaria, tal como que materiales están disponibles para resolver el problema.

Sin embargo los mismos requerimientos no necesariamente se miden contra criterios estrictos de éxito o fracaso. En la ingeniería de requerimientos se diseña contra un problema abierto. Un problema abierto es una situación en la que creemos que alguna mejora es posible, pero en la que no tenemos criterios definitivos para medir la mejora. Descubrir buenos criterios es en si mismo parte del problema. En ingeniería de requerimientos se empieza con un problema abierto y se termina con un problema bien definido.

#### **2.3.5.1.4.1.2 Dominio del problema.**

Los requerimientos definen el problema a ser resuelto por el software, no describen el software que lo resuelve. El dominio del problema es la parte del mundo en donde la computadora va a producir el efecto deseado, en conjunto con los medios disponibles para producirlo directa o indirectamente (Kovitz 1999).

#### **2.3.5.1.4.1.3 Identificar las fuentes de los requerimientos.**

Los requerimientos de sistemas pueden venir de diferentes partes, dependiendo de la naturaleza del producto y del medio ambiente de desarrollo. A continuación se describen algunas de las posibles fuentes de requerimientos:

- a) Usuarios potenciales.
- b) Productos actuales o similares.

- c) Reportes de problemas y solicitudes de ampliación para el sistema actual.
- d) Biblioteca de requerimientos reusables.

#### **2.3.5.1.4.1.4 Selección de métodos, técnicas y herramientas**

Se seleccionan dependiendo del tipo de proyecto, se pueden escoger algunos del punto 2.3.7 Se destacan:

- a) Entrevista. Yourdon (Yourdon 1989) menciona las directrices para conducir una entrevista:
  - a. Desarrollar un plan de entrevista general.
  - b. Estar seguro de que se cuenta con la aprobación para hablar con los usuarios.
  - c. Planificar la entrevista para hacer efectivo el uso del tiempo.
  - d. Utilizar herramientas automatizadas apropiadas, pero sin abusar.
  - e. Tratar de juzgar en información está más interesado el usuario
  - f. Utilizar un estilo de entrevista apropiado.
- b) Encuesta.
- c) Observación.
- d) Métodos de trabajo en grupo. Leffingwell (Leffingwell 2000) dice que son quizás la técnica más poderosa para obtener requerimientos, reúne a todos los involucrados en un periodo corto pero intensivo y enfocado de trabajo. Utiliza un facilitador externo experimentado en la administración de requerimientos, que ayuda a asegurar el éxito del taller. Destaca cuatro tipos:
  - a. Lluvia de ideas<sup>7</sup>: involucra la generación de ideas y su reducción, las ideas más creativas e innovadoras resultan a menudo de combinaciones múltiples de ideas aparentemente no relacionadas, se pueden utilizar varias técnicas para priorizar las ideas creadas. Aunque se prefiere la lluvia de ideas en vivo, la basada en Internet puede ser una alternativa viable en algunos casos.
  - b. Panel de historias<sup>8</sup>: su propósito es obtener en una fase temprana las reacciones “Sí, pero”, pueden ser pasivos, activos o interactivos. En los paneles de historias se identifica a los participantes, se explica lo qué les sucede y cómo les sucede.
  - c. Casos de uso: identifican quién, qué y cómo del comportamiento de un sistema. Describen las interacciones entre el usuario y el sistema, se enfocan en lo que el

---

<sup>7</sup> Brainstorming.

<sup>8</sup> Storyboarding.

sistema hace para el usuario. El modelo de casos de uso describe la totalidad del comportamiento funcional del sistema.

- d. Actuación de roles<sup>9</sup>: permite al equipo de desarrolladores experimentar el mundo del usuario desde la perspectiva del usuario. Las cartas de colaboración de responsabilidad de clase<sup>10</sup> (CRC) utilizadas a menudo en el análisis orientado a objetos, se derivan de la actuación de roles.

#### **2.3.5.1.4.1.5 Visión y alcance.**

La visión describe el tema del proyecto y lo que el producto podría llegar finalmente a ser en un mundo perfecto. El alcance describe lo que el producto en realidad es, así como lo que no va a incluir. (Wiegiers 1999). En este apartado también deberán definirse las restricciones externas que impacten en las soluciones de diseño o de implementación (como por ejemplo leyes y estándares internacionales).

La EDT es una forma de agrupamiento, orientada a entregables, de los elementos del proyecto que organiza y define el alcance total del proyecto (PMI 2000).

#### **2.3.5.1.4.1.6 Usuarios.**

Según Wiegiers (Wiegiers 1999) los usuarios pueden clasificarse entre otras maneras por:

- a) Frecuencia con la usan el producto.
- b) Dominio de la aplicación y pericia en los sistemas computacionales.
- c) Características de uso.
- d) Proceso de negocio que realizan.
- e) Localización geográfica.
- f) Nivel de privilegio de acceso.

Cada proyecto debe incluir un pequeño número de participantes clave de la comunidad de usuarios, a esta selección de usuarios la denominaremos “Campeones del producto”, ellos serán la interfase primaria entre los usuarios y los desarrolladores. Los mejores campeones del producto tienen una visión clara del nuevo sistema y son altamente entusiastas porque ellos anticipan la manera en la que el nuevo producto los beneficiará a ellos y a sus colegas. Los campeones deben ser comunicadores eficientes, que sean respetados por su grupo de colegas.

---

<sup>9</sup> Role Playing.

<sup>10</sup> Class-Responsibility-Collaboration cards

#### **2.3.5.1.4.1.7 Toma de decisiones.**

En general, las decisiones deben hacerse tan abajo en la organización jerárquica como sea posible, por la gente en contacto cercano con los ítems a decidir y que este bien informada sobre ellos. Según Wieggers (Wieggers 1999), se debe favorecer una toma de decisiones participativa sobre una toma de decisiones en consenso, ya que alcanzar el consenso es ideal, pero no se puede detener el avance del proyecto mientras se espera que todos los involucrados se alineen en cada uno de los ítems.

A continuación algunas formas sugeridas de manejar algunas situaciones de toma de decisiones que pudieran presentarse en los proyectos:

- a) Si los usuarios individuales desacuerdan en los requerimientos, los campeones del producto deciden.
- b) Si diferentes clases de usuarios tienen necesidades incompatibles, decidir cual clase de usuario es más importante a satisfacer.
- c) Todos los diferentes clientes corporativos pueden demandar que el producto sea diseñado para satisfacer sus preferencias. De nuevo hay que utilizar los objetivos del negocio para con el proyecto para determinar cuales clientes son los más importantes a satisfacer.
- d) Los requerimientos expresados por los administradores algunas veces se conflictúan con los expresados por los usuarios actuales en los departamentos. Se debe observar que los primeros estén alineados con los requerimientos del negocio y que no sean requerimientos de funcionalidad, si los administradores no están en contacto directo con el producto.
- e) Cuando los desarrolladores del producto piensen que entrarán en conflicto con lo que los clientes dicen que quieren, normalmente los clientes deberán tomar la decisión.

#### **2.3.5.1.4.1.8 Planeación de la búsqueda y recolección de requerimientos**

Las tareas de la EDT correspondientes a esta fase deberán ser como:

- a) Definir los escenarios de operación del sistema actual y sus ambientes de utilización (interacciones esperadas con el entorno y otros sistemas, tareas humanas, interconexiones físicas con productos, plataformas, sistemas, etc.). Documentar escenarios con casos de uso generales o casos de uso detallados (Hernández 2001b). Los casos de uso se describen en el punto 2.3.5.1.4.1.4.

- b) Definir las métricas de efectividad, que reflejen expectativas y satisfacción del cliente relacionadas con requerimientos de desempeño, seguridad, usabilidad, mantenibilidad, tiempo, costo, carga de trabajo, desempeño humano, etc. (Hernández 2001b).
- c) Reutilizar requerimientos (Wiegiers 1999).

#### **2.3.5.1.4.2 Análisis y negociación de Requerimientos de Sistema.**

Consiste en la investigación, determinación, clasificación e interpretación de hechos, diagnóstico de problemas y empleo de la información para recomendar mejora al sistema. Su producto fundamental es la Especificación del Requerimientos del Sistema, que se describe en el punto 2.3.5.1.4.3.

La negociación se refiere a lograr acuerdos con los clientes sobre la aceptación de un juego de Requerimientos, la prioridad, y la definición de una línea base (Sommerville 1997). Es importante porque:

- a) Los clientes tienen grandes expectativas del sistema y solicitan más de los que se puede hacer en una primera versión.
- b) El tiempo es reducido.
- c) Los recursos son limitados.

En la asignación de prioridades participan:

- a) Líder del proyecto.
- b) Representante del cliente.
- c) Representante del desarrollador.

#### **2.3.5.1.4.2.1 Dibujar el diagrama de contexto del sistema.**

Es un modelo simple que define los límites y las interfaces entre el sistema que está siendo desarrollado y las entidades externas al sistema. También identifica el flujo de información y materiales a lo largo de las interfaces (Wiegiers 1999). Es el diagrama funcional de más alto nivel, del cual se descomponen todos los demás y en él no se pueden mostrar conexiones fuera del dominio del sistema. (Kovitz 1999).



El sistema entero se representa como un simple proceso de caja negra, mostrado como un círculo o burbuja, también se muestran las actividades externas o terminadores que se conectan al sistema y los flujos de datos o materiales entre el sistema y los terminadores. Los flujos entre los elementos en el diagrama de contexto representan a menudo complejas estructuras de datos, que se definen en un diccionario de datos. Ver al apartado 3.2.2

#### **2.3.5.1.4.2.2 Crear prototipos de interfase con usuarios.**

Es la mejor manera para que el cliente entienda la interfase del usuario y le sean más tangibles los conceptos y posibilidades (Kovitz 1999). Los usuarios pueden evaluar el prototipo para que los participantes en el proyecto alcancen un mayor entendimiento mutuo del problema a ser resuelto... También sirven cuando encontramos conflictos entre los requerimientos escrito y el prototipo(Wiegers 1999). Los prototipos pueden ayudar a reducir el volumen y la magnitud de los cambios de las especificaciones de requerimientos, que de no hacerse serían necesarios más tarde en el proceso de desarrollo de software(Whitten 1995).

Los prototipos se pueden dividir en cuanto a su función en (Wiegers 1999):

- a) Prototipos horizontales: también llamados de comportamiento o maquetas, permiten explorar algunos comportamientos específicos del sistema, con la meta de refinar los requerimientos. Son como un escenario en el que se muestran las fachadas de las pantallas de interfases de los usuarios, permitiendo alguna navegación entre ellas, pero con muy poca o nula funcionalidad.
- b) Prototipos verticales: también llamados estructurales o de prueba de concepto, implementan una porción de la funcionalidad de la aplicación. Se utilizan cuando no se tiene la certeza de que la propuesta arquitectónica sea sólida, o cuando se quiere optimizar algoritmos, o evaluar un esquema de base de datos propuesta, o probar requerimientos críticos de tiempo.

Wiegers (Wiegers 1999) dice que los prototipos se pueden dividir en cuanto a su ciclo de vida en:

- a) Prototipos desechables: también llamados exploratorios, se utilizan para responder preguntas, resolver incertidumbres y mejorar la calidad de los requerimientos.
- b) Prototipos evolutivos: proveen los fundamentos de una arquitectura sólida para construir el producto incrementalmente mientras los requerimientos se vuelven más claros.

#### **2.3.5.1.4.2.3 Analizar la factibilidad de los requerimientos.**

Evaluar la factibilidad de la implementación de cada requerimiento a un desempeño y costo aceptable en un medio ambiente dado. Entender los riesgos asociados con la implementación de cada requerimiento, incluyendo los conflictos con otros requerimientos, dependencias o factores externos y obstáculos técnicos (Wiegiers 1999).

#### **2.3.5.1.4.2.4 Asignar prioridades y negociarlas con los clientes.**

Aplicar un acercamiento analítico para determinar la prioridad de implementación relativa, de los casos de uso, características del producto, o requerimientos individuales. Basados en ésta prioridad, determinar que versión del producto contendrá que característica o juego de requerimientos. Cuando los cambios de requerimientos sean aceptados habrá que colocarlos en una versión futura específica e incorporar el esfuerzo necesario para alcanzar ese cambio en la planeación de esa versión (Wiegiers 1999).

#### **2.3.5.1.4.2.5 Crear un diccionario de datos.**

El diccionario de datos es un repositorio central las definiciones de todos los ítems de datos y las estructuras utilizadas por el sistema. Sirve para asegurar que los desarrolladores que trabajan en componentes relacionados utilicen definiciones de datos consistentes. En la etapa de requerimientos, el diccionario de datos, deberá definir al menos los ítems de datos del cliente, para asegurar que el cliente y el equipo de desarrolladores utilicen las mismas definiciones y terminología.

#### **2.3.5.1.4.3 Especificaciones de Requerimientos de Sistema.**

La Especificación (Síntesis) de Requerimientos de Sistema (ERSi) describe las funciones y capacidades del sistema, requerimientos del negocio, organizacionales y del usuario; protección seguridad, ingeniería de factores humanos (ergonómicos), interfaces, operaciones y requerimientos de mantenimiento; restricciones de diseño y requerimientos de calidad

##### **2.3.5.1.4.3.1 Registrar las reglas del negocio.**

Las reglas del negocio son los principios operativos acerca del producto, tales como: quién puede tomar que acciones y bajo que circunstancias. Hay que documentar estas reglas en una sección especial de la ERSi o en un documento por separado. Algunas reglas del negocio, llevarán a requerimientos funcionales que las refuercen, estos requerimientos deberán de poder ser rastreados hasta sus correspondientes reglas del negocio.

#### **2.3.5.1.4.3.2 Adoptar plantillas de documentos.**

Definir una plantilla estándar que provea una estructura consistente para registrar tanto requerimientos de sistema como requerimientos de software y sus correspondientes requerimientos funcionales y no funcionales. Se utilizará una plantilla modificada del Standard 830-1998 (IEEE830 1998), originalmente está diseñada en función de requerimientos de software.

#### **2.3.5.1.4.3.3 Adoptar un esquema para la identificación de requerimientos.**

Definir una convención para proveer a cada requerimiento individual con una etiqueta de identificación única. La convención debe ser lo suficientemente robusta para resistir, adiciones, borrados y cambios de los requerimientos a través del tiempo. El etiquetado de los requerimientos permite la rastreabilidad de los mismos y facilita el mantener los registros de los cambios hechos en los requerimientos y establecer metas para el estatus y los cambios de actividad en los mismos.

#### **2.3.5.1.4.3.4 Señalar las fuentes de donde se tomaron los requerimientos.**

Para asegurar que todos los involucrados en el proyecto sepan porque están y a quién pertenecen los requerimientos del sistema, hay que rastrearlos hasta sus orígenes, por medio del etiquetado.

#### **2.3.5.1.4.3.5 Escribir una Especificación de Requerimientos de Sistema.**

Para la ERSi se adaptan algunas plantillas que aparecen en estándar 1233 de la IEEE Guía para del desarrollar Especificaciones de requerimientos de sistema (IEEE1233 1998), que incluyen la identificación, organización, presentación y modificación de los requerimientos, además de las condiciones para incorporar los conceptos operacionales, restricciones de diseño y los requerimientos de configuración de diseño.

En la ERSi es necesario documentar la arquitectura del sistema y la asignación de componentes del mismo al hardware, software o procesos manuales.

Kovitz (Kovitz 1999) dice que los principales beneficios de la documentación son:

- a) Extender lo que la mente puede asir y recordar.
- b) Dar la misma historia a cada miembro del equipo.
- c) Introducir en el proyecto a los nuevos miembros del mismo.
- d) Proteger la equidad intelectual.
- e) Ayudar al escritor a entender mejor el problema.

En cuanto al contenido y el estilo, el mismo Kovitz señala que, para la organización del contenido primero debe hacerse una lista no ordenada de los contenidos: grandes tópicos, pequeños tópicos, conceptos, proposiciones, ideas para gráficas y cualquier otra cosa que se quiera incluir en el documento. Posteriormente de debe establecer una secuencia tomando en cuenta que:

1.- El problema es más fundamental que la solución. Los hechos que no están dentro de nuestro control para escoger, son más fundamentales que los que podemos escoger

2.- Las cosas son mas fundamentales que sus atributos, las relaciones entre ellas y otras cosas, y las acciones que pueden hacer o se les pueden hacer.

3.- Un caso normal es más fundamental que los casos excepcionales.

4.- Lo que una cosa es, es más fundamental que cualquier rol que ella jueguen situaciones diferentes o como sea utilizada.

5.- Una descripción de agentes, usualmente deberá de preceder a una descripción de los agentes pasivos sobre los que actúan.

Para distinguir los elementos principales del contenido hay que hacer énfasis, éste se puede lograr de las siguientes maneras:

1.- Énfasis gráfico.

2.- Lo que sea que aparezca primero se enfatiza automáticamente.

3.- Lo que sea referido muchas veces se enfatiza.

4.- Rodear un contenido con espacio en blanco lo enfatiza.

5.- Tomar más espacio enfatiza.

6.- Cualquier tipo de contraste enfatiza los elementos contrastados.

7.- Las itálicas son el último recurso.

En el estándar 12207 de la IEEE/EIA<sup>11</sup> Ciclos de vida de los procesos de Software (IEEE/EIA12207.0 1995; IEEE/EIA12207.1 1997; IEEE/EIA12207.2 1997), que provee una guía para registrar los datos del ciclo de vida resultantes del proceso de ciclo de vida.

#### **2.3.5.1.4.4 Verificación de Requerimientos de Sistema.**

---

<sup>11</sup> Electronics Industries Alliance.

Las actividades de los requerimientos aseguran que las declaraciones de los requerimientos sean exactas, completas y demuestran las características de calidad deseadas.

#### **2.3.5.1.4.4.1 Definir criterios de aceptación (éxito) para cada requerimiento en interacción con el cliente.**

Pedir a los usuarios que describan como determinarán que el producto satisface sus necesidades y que está listo para su uso. Basar las pruebas de aceptación en escenarios de uso o en casos de uso.

#### **2.3.5.1.4.4.2 Realizar inspecciones en cada una de las actividades.**

Se puede utilizar la técnica de Caminar a través<sup>12</sup> definido en el estándar 610.12 de la IEEE (IEEE610.12 1990) y en Wiegers (Wiegers 1999), como una técnica de análisis estático en la que un diseñador o programador guía a los miembros del equipo de desarrollo y a otros interesados, a través de un segmento de documentación o código, y los participantes hacen preguntas y comentarios sobre posibles errores, violaciones de desarrollo y estándares y otros problemas.

En nuestro caso estaríamos aplicando esta técnica para verificar el desarrollo de requerimientos de sistema.

#### **2.3.5.1.4.4.3 Revisar documentos de requerimientos que se hayan producido.**

La inspección formal de los documentos de requerimientos es una de las prácticas de calidad de software disponibles de más alto valor. Se debe reunir a un pequeño grupo de inspectores que representen diferentes perspectivas para que examinen cuidadosamente la ERSi los modelos relacionados tratando de encontrar los defectos (Wiegers 1999).

#### **2.3.5.1.4.4.4 Escribir pruebas de caso de los requerimientos.**

Derivar casos de prueba de caja negra (funcionales) de los casos de uso, para documentar el comportamiento esperado del producto bajo condiciones especificadas, también se puede utilizar la técnica de Caminar a través de los casos de prueba con los clientes para asegurarse de que el producto refleja el comportamiento deseado. Rastrear los casos de prueba hasta los requerimientos funcionales para asegurarse de que ningún requerimiento ha sido pasado por alto y de que todos tienen un caso de prueba correspondiente (Wiegers 1999).

#### **2.3.5.1.4.4.5 Escribir el manual del usuario.**

Hacer un borrador del manual del usuario en una etapa temprana del proceso de desarrollo de requerimientos y utilizarla como especificación de requerimientos o como ayuda en el análisis de

requerimientos. Describir todas las funciones visibles al usuario en un lenguaje de fácil comprensión (Wieggers 1999).

### **2.3.5.1.5 Reuso de Requerimientos de Sistema.**

Actualizar biblioteca de requerimientos reusables.

### **2.3.5.1.6 Salidas**

- a) Especificación de Requerimientos de Sistema. El propósito es especificar los requerimientos para un sistema o subsistema y los métodos a ser utilizados para asegurar que cada requerimiento haya sido cumplido, y debe incluir:
- a. Información de especificación genérica (fecha, estado, alcance, contexto, glosario, etc.)
  - b. Identificación del sistema y vista general.
  - c. Modos y estados requeridos.
  - d. Requerimientos para las funciones y desempeño del sistema.
  - e. Requerimientos del negocio, de la organización y del usuario.
  - f. Requerimientos de protección<sup>13</sup>, seguridad y protección de privacidad.
  - g. Requerimientos de ingeniería factores humanos (ergonomía).
  - h. Requerimientos de operación y mantenimiento.
  - i. Requerimientos de interfase externa del sistema.
  - j. Requerimientos ambientales del sistema.
  - k. Requerimientos de calificación y restricciones de diseño.
  - l. Requerimientos de recursos computacionales.
    - i. Requerimientos de hardware de computadora.
    - ii. Requerimientos de software de computadora.
    - iii. Requerimientos de computación de computadora.
  - m. Características de calidad del sistema.
  - n. Requerimientos de datos internos.
  - o. Requerimientos de datos dependientes de instalación.
  - p. Requerimientos físicos.
  - q. Requerimientos de personal de entrenamiento y logística.
  - r. Requerimientos de empaquetado.

---

<sup>12</sup> Walk through.

<sup>13</sup> Safety.

- s. Prioridad de requerimientos.
- t. Razón del requerimiento
- b) Biblioteca de Requerimientos Reusables actualizada.

### **2.3.5.2 Subproceso de Diseño de la Arquitectura del Sistema.**

#### **2.3.5.2.1 Objetivos.**

Establecer una arquitectura de alto nivel que muestre y describa a los componentes del sistema y sus interrelaciones y asignar requerimientos a los componentes de hardware, software y operaciones manuales.

#### **2.3.5.2.2 Entradas.**

- a) Especificaciones de requerimientos de sistema.
- b) Plan del proyecto.
- c) Plan de administración de riesgos.

#### **2.3.5.2.3 Participantes.**

Líder del proyecto, analistas, clientes y usuarios.

#### **2.3.5.2.4 Procedimiento.**

- a) Diseñar la arquitectura del sistema.
- b) Identificar los elementos de hardware, software y operaciones manuales.
- c) Documentar la arquitectura del sistema y asignación de requerimientos.
  - a. Estructura estática, representada mediante Diagramas de estructura o DFD de alto nivel donde se señale cada porción de la arquitectura.
  - b. Estructura dinámica de la arquitectura, representada con Diagramas de Transición de Estados.
- d) Evaluar la arquitectura y asignación de requerimientos considerando:
  - a. Rastreabilidad para los requerimientos del sistema.
  - b. Consistencia de los Requerimientos del Sistema.
  - c. Lo apropiado de los estándares de diseño y métodos empleados.
  - d. Factibilidad de que los elementos de software cumplan con los requerimientos asignados.
  - e. Factibilidad de operación y mantenimiento.
- e) Documentar los resultados de la evaluación.

- f) Aprobar en junta con clientes la Arquitectura y considerarla línea base para el proceso de Administración de la Configuración y para el subproceso de Análisis de Requerimientos de Software.

#### **2.3.5.2.5 Salidas.**

- a) Arquitectura del Sistema y Descripción de la Asignación de Requerimientos. El propósito es describir el diseño de la arquitectura de un sistema o subsistema. y debe incluir:
  - a. Información de descripción genérica (fecha, estado, alcance, contexto, glosario, etc.)
  - b. Vista general del sistema e identificación.
  - c. Identificación de elementos de hardware.
  - d. Identificación de elementos de software.
  - e. Identificación de operaciones manuales.
  - f. Conexiones y restricciones de los componentes de los incisos c, d y e.
  - g. Necesidades de los involucrados en el sistema.
  - h. Razones para la asignación de elementos de hardware, elementos de software y operaciones manuales.
  - i. Directrices o reglas acerca de cómo se deben adicionar los nuevos componentes a la arquitectura y cómo podría evolucionar la misma.
  - j. Referencias a estándares o métodos a los que está atada la arquitectura o que asume la misma, o a los que obligan los involucrados o que se escogen como principios fundamentales.
  - k. Una descripción del comportamiento dinámico de alto nivel del sistema, mostrando cómo los componentes trabajarían juntos y cómo sincronizan su trabajo para satisfacer los escenarios del usuario final (transiciones de estado de alto nivel).
- b) Resultados de la evaluación de la arquitectura.

### **2.3.5.3 Subproceso de Desarrollo de Requerimientos de Software**

#### **2.3.5.3.1 Objetivo.**

Especificar los requerimientos para los componentes de software del sistema, de forma tal que sirvan como base para el diseño del software, para la generación de casos de prueba y para la verificación final del alcance de los objetivos del funcionamiento y calidad del producto.



### **2.3.5.3.2 Entradas.**

- a) Especificaciones de Requerimientos de Sistema.
- b) Arquitectura del Sistema.
- c) Descripción de la Asignación de Requerimientos.
- d) Resultados de la evaluación de la Arquitectura.
- e) Plan del proyecto.
- f) Planes de administración del proyecto.

### **2.3.5.3.3 Participantes.**

Líder del proyecto, analistas, clientes y usuarios.

### **2.3.5.3.4 Procedimiento.**

Tomar cada elemento de la Arquitectura del Sistema y Descripción de la Asignación de Requerimientos correspondiente al software y:

- a) Obtención.
  - a. Seleccionar métodos técnicas y herramientas.
  - b. Revisar las descripciones de Casos de Uso Detallados de la ERSi que correspondan al software. Emplear una lista de verificación y realizar las correcciones necesarias.
  - c. Recolectar información sobre los atributos de calidad y otros requerimientos no funcionales de los usuarios.
- b) Análisis y negociación.
  - a. Detallar los requerimientos funcionales y sus modelos asociados (DFD u otros) de la ERSi, que correspondan al software, para clarificar la comprensión de los participantes en la recolección de las porciones de requerimientos de software. Con esta actividad se obtiene un borrador de la Especificación de Requerimientos de Software (ERS).
  - b. Desarrollar los prototipos de interfaces de usuario para ayudar a visualizar requerimientos que no son comprendidos claramente.
  - c. Evaluar los prototipos. Ésta y la anterior son tareas iterativas.
- c) Especificaciones.
  - a. Documentar requerimientos de software funcionales y no funcionales.
  - b. Documentar los prototipos de interfaces de usuario para ayudar a visualizar requerimientos que no son comprendidos claramente.
  - c. Escribir una Especificación de Requerimientos de Software.
- d) Verificación.

- a. Revisar documentos de requerimientos que se hayan producido.
  - b. Desarrollar casos de prueba conceptuales a partir de Casos de Uso Detallados.
  - c. Usar los Casos de Prueba Conceptuales para verificar los Casos de Uso Detallados, los requerimientos funcionales, los modelos de análisis y los prototipos.
  - d. Revisar la ERS.
  - e. Aprobar la ERS en junta con los clientes.
  - f. Pasar ERS de línea base a proceso de Administración de la Configuración (Administración de Requerimientos) y al proceso de Diseño.
  - g. Elaborar el manual del usuario correspondiente a este subproceso.
- e) Reuso.
- a. Actualizar biblioteca de requerimientos reusables.

#### **2.3.5.3.4.1 Obtención de Requerimientos de Software.**

Algunas de las tareas de esta actividad ya se realizaron en el subproceso de Desarrollo de Requerimientos de Sistema 2.3.5.1, por lo que se hace referencia a las mismas para evitar redundancia.

##### **2.3.5.3.4.1.1 Seleccionar métodos técnicas y herramientas.**

Ver punto 2.3.5.1.4.1.4.

##### **2.3.5.3.4.1.2 Revisar las descripciones de Casos de Uso Detallados de la ERSi que correspondan al software.**

Un caso de uso es la descripción de un caso del programa que esta siendo utilizado, es un camino a través de la funcionalidad del sistema, envolviendo un dialogo entre el sistema y alguna entidad externa que interactúa con él, que muestra cada acción iniciada por el usuario, pieza o hardware u otro software (llamados colectivamente actores), y las respuestas del programa (Kovitz 1999). Los casos de uso emergieron del mundo del desarrollo orientado a objetos.

Ver punto 2.3.5.1.4.1.8 inciso a.

Wiegers (Wiegers 1999), citando a Ham y a Larman menciona que los casos de uso de se pueden identificar utilizando varios acercamientos:

- a) Primero identificar a los actores y sus roles, luego identificar los procesos del negocio en los que participa cada uno para identificar los casos de uso.
- b) Identificar los eventos externo a los cuales debe responder el sistema, luego relacionar estos eventos con los actores participantes y los casos de uso específicos.

- c) Expresar los procesos del negocio o las actividades diarias en términos de escenarios específicos, derivar casos de uso de los escenarios, identificar los actores involucrados en cada caso de uso. Los escenarios son instancias específicas del caso de uso y pueden ser de dos tipos:
  - a. El curso normal o principal.
  - b. Cursos alternativos, que también resultan en completar satisfactoriamente la tarea, pero que representan variaciones en tareas específicas o el camino utilizado para completar la tarea.
- d) Derivar casos de uso probables de las declaraciones existentes de los requerimientos funcionales.

Wieggers también dice que los beneficios de utilizar casos de uso son:

- a) Se centran en el usuario y en las tareas.
- b) Previenen funcionalidades huérfanas, al relacionar usuarios con tareas y eliminar las innecesarias.
- c) Los desarrolladores que utilizan métodos de diseño orientados a objetos pueden convertir los casos de uso en modelos de objetos.

#### **2.3.5.3.4.1.3 Recolectar información sobre los atributos de calidad y otros requerimientos no funcionales de los usuarios.**

Hay que explorar las características de calidad y otros requerimientos no funcionales que contribuirán a que nuestro producto alcance o exceda las expectativas del cliente.

Ver puntos 2.3.3.2 y 2.3.5.1.4.1.8.

#### **2.3.5.3.4.2 Análisis y negociación de Requerimientos de Software.**

El análisis incluye el refinamiento y el escrutinio de los requerimientos de software obtenidos, para asegurarse de que todos los involucrados entienden lo que significan y encontrar conflictos, traslapes, errores, omisiones, inconsistencias u otras deficiencias.

Para la negociación ver el punto 2.3.5.1.4.2

#### **2.3.5.3.4.2.1 Detallar los requerimientos funcionales y no funcionales, y sus modelos asociados de la ERSi, que correspondan al software.**

- a) Funcionales
  - a. Interfases externas a los elementos de software.

- b. Operación de usuarios y ejecución.
  - c. Ingeniería de factores humanos como interacciones hombre- equipo y restricciones de acceso sobre el personal.
  - d. Definición de datos y bases de datos.
  - e. Aceptación de los productos de software
- 
- b) No funcionales
    - a. Del producto como: requerimientos de desempeño.
    - b. De Ingeniería de factores humanos como: operaciones manuales, áreas que necesitan atención concentrada del humano.
    - c. Organizacionales como: requerimientos de estándares y de aceptación
    - d. Externos como: requerimientos de seguridad, incluyendo aquellos relacionados con los métodos de operación y mantenimiento, influencias del medio y daños personales.

Los modelos se describen en el punto

Los modelos deben utilizarse para aumentar la comprensión del sistema complementando al lenguaje los métodos para modelar pueden ser (Wieggers 1999):

- a) Diagramas de flujo de datos (DFD). Identifican los procesos de transformación de un sistema, la colección o almacenamientos de datos o de material que manipula el sistema y los flujos de datos o material entre los procesos, almacenes y el mundo exterior. Los DFD hacen una descomposición jerárquica para abordar el análisis del sistema.
- b) Diagramas de entidad relación (DER). Muestran las relaciones de datos del sistema, representando grupos lógicos de información del dominio del problema y de sus conexiones.
- c) Diagramas de estado- transición (DET). Los sistemas de tiempo real y las aplicaciones de control de procesos pueden existir en uno de un limitado número de estados posibles a un tiempo dado. Un cambio en el estado solamente puede tener lugar cuando se satisfacen criterios bien definidos. Muchos sistemas de información trabajan con objetos del negocio que pasan por complejos ciclos de vida que pueden estudiarse con DETs.
- d) Mapas de dialogo. Representan la arquitectura de la interfase del usuario a un alto nivel de abstracción. Los usuarios y los desarrolladores pueden estudiar el mapa de

dialogo para alcanzar una visión común de cómo el usuario podría interactuar con el sistema para realizar una tarea.

- e) Diagramas de clase. Son una manera gráfica de mostrar las clases que se encontraron durante el análisis orientado a objetos y las relaciones entre ellas. Una clase es un conjunto de objetos agrupados en plantillas genéricas.

#### **2.3.5.3.4.2.2 Desarrollar los prototipos de interfaces de usuario.**

Para ayudar a visualizar aquellos que no son comprendidos claramente.

Sobre los prototipos ver el punto 2.3.5.1.4.2.2.

#### **2.3.5.3.4.2.3 Evaluar los prototipos.**

Esta tarea y la anterior son iterativas.

Wiegiers (Wiegiers 1999) dice que para evaluar los prototipos se debe observar si:

- a) El prototipo implementa la funcionalidad en la manera que se esperaba.
- b) Falta alguna funcionalidad.
- c) Se pueden condiciones de error, que el prototipo no contempla.
- d) Se presentan funciones innecesarias.
- e) La navegación parece lógica y completa.
- f) Se puede pensar en maneras más fáciles de realizar esta tarea.

#### **2.3.5.3.4.3 Especificaciones de Requerimientos de Software.**

La Especificación de Requerimientos de Software (ERS) tiene el propósito de documentar los requerimientos de manera consistente accesible y fácil de revisar, son parte de la ERSi, pero se separan porque están dirigidas a los desarrolladores y tienen un lenguaje técnico específico.

Ver punto 2.3.5.1.4.3.

##### **2.3.5.3.4.3.1 Documentar requerimientos de software funcionales y no funcionales.**

Ver punto 2.3.5.3.4.2.1.

##### **2.3.5.3.4.3.2 Documentar los prototipos de interfaces de usuario.**

Para ayudar a visualizar requerimientos que no son comprendidos claramente.

Ver punto 2.3.5.1.4.2.2.

##### **2.3.5.3.4.3.3 Escribir una Especificación de Requerimientos de Software.**

Para la ERS se adaptan algunas plantillas que aparecen en el estándar IEEE 830-1998 (IEEE830 1998), que se dividen en tres partes:

- a) Introducción
  - a. Propósito.
  - b. Alcance.
  - c. Definiciones, siglas y abreviaturas.
  - d. Referencias
  - e. Vista General.
- b) Descripción general.
  - a. Perspectiva del producto.
  - b. Funciones del producto.
  - c. Características de los usuarios.
  - d. restricciones.
  - e. Supuestos y dependencias.
- c) Requerimientos específicos.

Apéndices.

Índice.

En cuanto al contenido y estilo ver el punto 2.3.5.1.4.3.5. y tomar en cuenta además los siguientes elementos mencionados por Sommerville (Sommerville 1997):

- a) Los requerimientos son leídos más a menudo de lo que son escritos, por lo que hay que invertir tiempo y esfuerzo al escribirlos, además escribir clara y concisamente no es fácil.
- b) Los lectores de los requerimientos tienen formación y conocimientos diferentes.

#### **2.3.5.3.4.4 Verificación de Requerimientos de Software.**

Después de que el documento de requerimientos se ha escrito, los requerimientos deben verificarse (validarse) formalmente. Este proceso tiene que ver con la verificación de omisiones, conflictos y ambigüedades para asegurar que los requerimientos sigan los estándares de calidad (Sommerville 1997).

##### **2.3.5.3.4.4.1 Revisar documentos de requerimientos que se hayan producido.**

Ver punto 2.3.5.1.4.4.3.

#### **2.3.5.3.4.4.2 Desarrollar casos de prueba conceptuales.**

Los casos de prueba que se pueden desarrollar a partir de casos de uso detallados o bien de los requerimientos funcionales (Wiegiers 1999), utilizando técnicas analíticas tales como gráficos causa-efecto y sirven para hacer tangible el comportamiento del sistema a los participantes en el proyecto.

#### **2.3.5.3.4.4.3 Usar los Casos de Prueba Conceptuales para verificar los Casos de Uso Detallados, los requerimientos funcionales, los modelos de análisis y los prototipos.**

Se deben probar todos los productos para revelar ambigüedades e inconsistencias, cada requerimiento se deberá localizar en al menos un caso de prueba, para evitar que se pase por alto algún comportamiento del sistema.

#### **2.3.5.3.4.4.4 Revisar la ERS.**

Ver punto 2.3.5.1.4.4.3.

#### **2.3.5.3.4.4.5 Aprobar la ERS en junta con los clientes.**

En esta actividad se fija la línea base, en la cual se hace la transición de una ERS en desarrollo a una que ha sido revisada y aprobada.

#### **2.3.5.3.4.4.6 Pasar ERS de línea base a proceso de Administración de la Configuración (Administración de Requerimientos) y al proceso de Diseño.**

Ver figura 7.

#### **2.3.5.3.4.4.7 Elaborar el manual del usuario correspondiente a este subproceso.**

Ver punto 2.3.5.1.4.4.5.

#### **2.3.5.3.4.5 Reuso de Requerimientos de Software.**

Actualizar biblioteca de requerimientos reusables

#### **2.3.5.3.5 Salidas.**

- a) Especificación de Requerimientos de Software (ERS). El propósito es especificar los requerimientos para un producto de software y los métodos a ser utilizados para asegurar que cada requerimiento ha sido logrado, y debe incluir:
  - a. Información de especificación genérica (fecha, estado, alcance, contexto, glosario, etc.)

- b. Identificación del sistema y vista general (explícitamente o una referencia a otro documento que ya lo contenga).
  - c. Funcionalidad del Software, incluyendo:
    - i. Requerimientos de desempeño.
    - ii. Características físicas.
    - iii. Condiciones del medio ambiente.
  - d. Requerimientos para las interfaces externas al software.
  - e. Requerimientos de calidad.
  - f. Especificaciones de protección (safety), incluyendo aquellas relacionadas con métodos de operación y mantenimiento, influencias ambientales, y daños personales.
  - g. Especificaciones de seguridad y privacidad, incluyendo aquellas relacionadas con compromisos de información sensible.
  - h. Requerimientos de ingeniería de factores – humanos (ergonomía), incluyendo aquellos para:
    - i. Manual de operaciones.
    - ii. Interacciones equipo – humano.
    - iii. Restricciones de personal.
- b) Casos de prueba conceptuales.



## **2.3.6 Métodos de la Ingeniería de Requerimientos.**

### **2.3.6.1 Conceptos básicos.**

#### **2.3.6.1.1 Enfoque.**

Significa dirigir la atención o el interés hacia un asunto o problema desde unos supuestos previos, para tratar de resolverlo acertadamente. Es la manera en que se plantea o estudia un problema.

#### **2.3.6.1.2 Herramienta.**

Es el medio con el cual se aplica de forma práctica un método, con el cuál se aplica una técnica.

#### **2.3.6.1.3 Método.**

Es el modo de decir o hacer con orden una cosa, es el procedimiento que se sigue en las ciencias para hallar la verdad y enseñarla; puede ser de dos maneras: analítico y sintético. Procedimiento sistemático y ordenado para hacer algo.

#### **2.3.6.1.4 Metodología.**

Es un conjunto de métodos que se emplean en una investigación o ejecución.

#### **2.3.6.1.5 Técnica.**

Es el conjunto de procedimientos o recursos de que se sirve una ciencia o un arte. Pericia o habilidad para usar de esos procedimientos y recursos. Es la aplicación práctica de un método.

### **2.3.6.2 La disciplina en los proyectos de Ingeniería de Requerimientos.**

Disciplina es el acto de alentar un patrón deseado de comportamiento y el factor más grande que puede hacer o romper un proyecto de desarrollo de software es el grado de disciplina que ejerza el liderazgo del proyecto (Whitten 1995).

Todas las personas quieren y necesitan saber el patrón de comportamiento aceptable que se espera de ellos. Para los miembros de un proyecto es importante saber: cual es la misión, entender las tareas, entender como van a ser medidos contra un desempeño, saber como van a ser reconocidos y premiados por sus logros, saber que esperar de sus lideres, creer que los líderes del proyecto hacen un esfuerzo genuino para comprender a su gente y mantener una buena comunicación bidireccional, creer que los lideres del proyecto tomaran las mejores decisiones para el éxito del proyecto.

La disciplina impacta sobre la moral, la productividad, la calidad, las programaciones y los costos.

Para implementar disciplina es necesario:

- Establecer metas realistas: bien indicadas, entendidas por todos y medibles.
- Obtener compromisos.
- Rastrear el progreso contra los planes.
- Hacer cumplir los compromisos

### **2.3.6.3 El enfoque estructurado.**

Está basado en la descomposición funcional o en la descomposición de eventos y en el tradicional ciclo de vida de cascada, cambiando el énfasis de la construcción y prueba, al diseño. El enfoque estructurado se puede dividir en cinco técnicas (Yourdon 1988):

- a) Análisis Estructurado.
- b) Diseño Estructurado.
- c) Programación Estructurada.
- d) Desarrollo Arriba – Abajo<sup>14</sup>.
- e) Equipos de programación.
- f) Caminar a través estructurados<sup>15</sup>

Algunas ventajas de utilizar el enfoque estructurado son (Kendall 1996):

- a) Problemas grandes se parten en pequeños, más fácilmente manejables.
- b) Utiliza gráficos para ilustrar problemas.
- c) Todos los aspectos del proceso y del sistema están documentados.
- d) Los defectos se descubren más pronto, por medio de los diagramas propuestos.

Todos estos métodos hoy en día tienen severas limitaciones, sobre todo con la proliferación de los lenguajes orientados a objetos, el crecimiento en tamaño y complejidad de los sistemas, la proliferación de las computadoras personales y los sistemas de ventanas manejados por eventos (Hernández 2001). El método de diseño estructural es directo pero obliga a los programadores a concentrarse en operaciones, prestando poca atención a la estructura de los datos. Los diseños suelen

---

<sup>14</sup> Top-down development.

<sup>15</sup> Structgured walkthroughs.

traducirse en más código y menos datos, porque la organización de datos se deriva de los procesos y de sus necesidades de ínter actuación. Además, el método de diseño no está bien ligado; es probable que diferentes analistas trabajando en el mismo diseño obtengan resultados diferentes (Winblad 1993).

#### **2.3.6.3.1 Análisis Estructurado.**

Entre estos métodos podemos mencionar:

- a) El “Análisis Estructurado de Sistemas” de Gane y Sarson (Gane 1979), que propone el establecimiento de estándares y procedimientos como el emplear relaciones de tercera forma normalizada en la construcción de los almacenes de datos.
- b) El “Análisis Estructurado y la especificación de sistemas” de Demarco (Demarco 1979), que hace énfasis en introducir la tarea de “codificación” en la actividad de análisis para traducir los requerimientos del sistema a documentos de trabajo como diagramas de flujo, diccionario de datos, etc. en lugar de la mera descripción narrativa de lo que se requiere. El “Análisis Estructurado Moderno” de Yourdon (Yourdon 1989), propuso cambios en la terminología, en las herramientas, el modelado, la inclusión de diagramas entidad relación para modelar datos y la partición de eventos.

Una herramienta de este método son los diagramas de Warnier, que son fundamentalmente una serie de corchetes utilizados con un número pequeño de otros símbolos, para representar un problema (Orr 1977). Otras herramientas están descritas en el “Análisis Estructurado” de Wienberg (Wienberg 1978) y en “La guía práctica al diseño estructurado de sistemas” (Page-Jones 1988): diagramas de flujo de datos, diagrama de estructura de datos, diagramas de entidad- relación, tabla de decisiones, lenguaje estructurado, árbol de decisiones, tablas estructuradas y diccionario de datos.

#### **2.3.6.3.2 Diseño estructurado.**

Entre estos métodos podemos mencionar:

- a) El “Diseño Estructurado” de Yourdon y Constantine (Yourdon 1979) quien lo definió como: que componentes interconectados de qué manera resolverán un problema bien especificado. Sus componentes son: Técnicas de documentación, criterios de evaluación de diseño, heurísticas de diseño y estrategias de diseño.
- b) El diseño estructurado para los sistemas en tiempo real de Hatley y Pirbhai.

- c) El análisis y diseño estructurado “Suceso respuesta” presentado en 1984 por Stephen McMenamin y John Palmer y adaptado posteriormente para sistemas en tiempo real por Stephen Mellor y Paul Ward (Winblad 1993).

#### **2.3.6.3.3 Programación Estructurada.**

El término fue acuñado por Edsger Dijkstra, para quien algo estructurado tiene que cumplir con dos requisitos: estar organizado jerárquicamente y que las piezas de cada función estén relacionadas una a otra, por una secuencia, o una alternación, o una repetición (Orr 1977).

En 1966, Böhm y Jacopini probaron formalmente la teoría básica de la programación estructurada, de que cualquier programa podía ser escrito utilizando solamente tres constructos lógicos (Wienberg 1978):

- b) El proceso: un mecanismo con una declaración imperativa simple.
- c) La decisión: un mecanismo binario de toma de decisiones.
- d) El ciclo: un mecanismo de ciclado.

#### **2.3.6.4 El lenguaje de patrones.**

Es la manera en como la ingeniería realmente trabaja y puede ser de las dos formas siguientes (Kovitz 1999) :

- 1.- Los ingenieros aplican una ligera variación de un diseño que ya existía y había sido probado, o bien.
- 2.- Los ingenieros se enfocan a una exploración desestructurada de nuevos diseños y nuevas formas de reunir viejos diseños.

Consiste en catalogar numerosos patrones que se encuentran comúnmente, de la misma manera que poseer un amplio vocabulario capacita a la gente a escribir bien, poseer un rico vocabulario de diseño de patrones nos capacita para diseñar bien. Los patrones son ideas reutilizables.

En general un patrón tiene cuatro elementos esenciales (Gamma 1995):

- a) El nombre del patrón.
- b) El problema.
- c) La solución.

- d) Las consecuencias.

Los patrones en enfoque orientado a objetos, ver punto 2.3.6.5, se pueden catalogar en tres tipos(Gamma 1995):

- a) Creacionales
- b) Estructurales
- c) De comportamiento.

### **2.3.6.5 El enfoque orientado a objetos.**

Sus raíces pueden encontrarse en Noruega a finales de los años 60 en conexión con un lenguaje llamado Simula67, desarrollado por Kristen Nygaard y Ole-Johan Dahl., en le Centro de Cálculo Noruego. Este programa demostró el poder de modelación de un programa basado en clases, así como la idea de que los datos y las operaciones debían almacenarse juntos y que se podía ahorrar esfuerzo de programación si las propiedades comunes de los objetos podían ser preprogramadas.

Posteriormente, a mitad de la década de los 70, los científicos del Centro de investigación Palo Alto de Xerox (Xerox PARC) crearon el leguaje Smalltalk, primer lenguaje orientado a objetos consistente y complejo.

En la década de los 80 Peter Chen popularizo el modelado de la información, que se considera el precursor más próximo del análisis y diseño orientado a objetos. El primer material publicado sobre el análisis orientado a objetos provino de Sally Shlaer y Stephen Mellor en 1988 (Winblad 1993).

Sus mecanismo básicos son (Winblad 1993; Jacobson 1994):

- a) **Objeto** o mejor dicho una instancia de una clase, tiene un estado interno que es conciente de su propia clase. El estado del objeto se llama información del objeto. Un objeto puede ser utilizado por otros objetos. Cuando un objeto utiliza a otro objeto se dice que, le envía estímulos. El objeto receptor acepta el estímulo y realiza una operación, lo cual significa que el objeto inspecciona sus valores, altera su propio estado o utiliza otros objetos.
- b) **Clase** es una plantilla para todos los objetos, que define para que pueden ser utilizados estos objetos, la clase describe las operaciones que el objeto puede ofrecer, las actividades varias que pertenecen a cada operación, y los diferentes atributos que tiene cada objeto, el estado interno de un objeto está determinado por el valor de sus atributos

y asociaciones. Un objeto realiza una operación obedeciendo la definición de operación de su clase.

- c) **Modelo de objeto** consiste en un juego de clases y sus asociaciones, éstas pueden ser de dos tipos: de “conocimiento” y de “comunicación”
- d) **Herencia de clase** significa reutilizar la definición de la clase, en términos de operaciones, asociaciones y atributos.

Sus conceptos clave son (Winblad 1993):

- a) **Encapsulación**, el acceso a los datos de un objeto se permite solamente a través de los propios métodos del objeto.
- b) **Abstracción**, factores abstractos de comportamiento común entre los objetos para agruparlos en superclases.
- c) **Polimorfismo**, el mismo mensaje puede originar acciones completamente diferentes al ser recibido por diferentes objetos.
- d) **Persistencia**, distinción entre objetos creados solamente para el tiempo de duración de la ejecución y aquellos pensados para el almacenamiento permanente.

Sus términos técnicos relacionados son (Winblad 1993):

- a) **Ligadura dinámica**, las conexiones apropiadas entre sus componentes ocurren cuando el programa se está ejecutando.
- b) **Programación visual**, que provee al usuario de funcionalidad orientada a objetos.
- c) **Gran Objeto Binario** (Blob), es parte de una jerarquía de clases, tiene capacidad de gestión de mensajes, sirve para almacenar grandes cantidades de datos sin diferenciar.

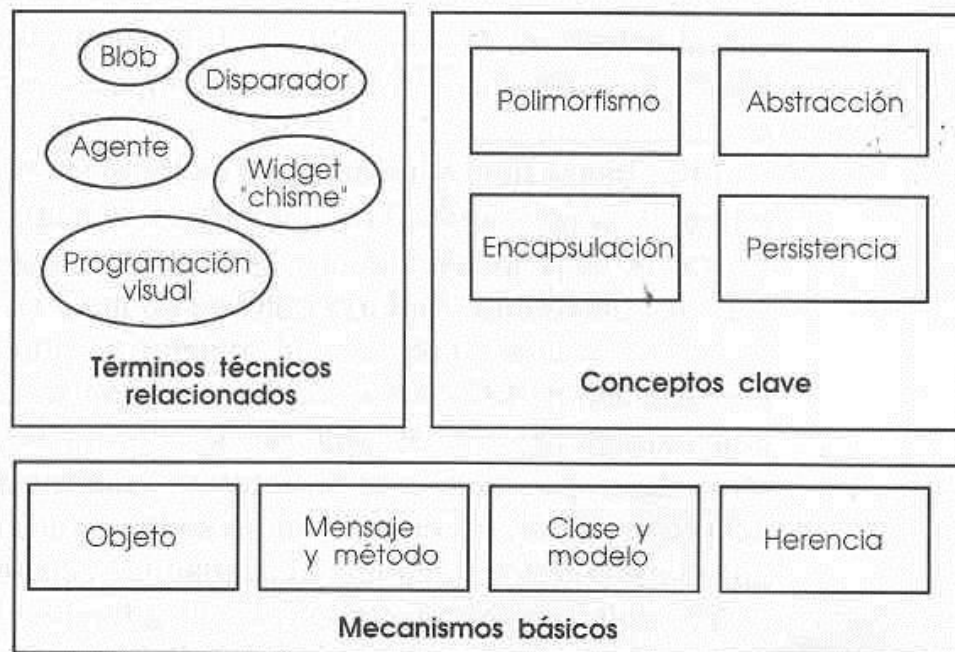


Fig. 2.3.6.5.1. Los pilares básicos de la orientación a objetos tomado de Winblad (Winblad 1993).

Jacobson (Jacobson 1994) propone el enfoque orientado a objetos porque se parece más a la realidad y por que sus abstracciones se entienden más fácilmente por la gente en comparación con otros métodos como el “Análisis y diseño estructurados” (SA/DA)<sup>16</sup>, la “Definición integrada de manufacturación ayudada por computadora” (IDEF), o la “Técnica de diseño y análisis estructurado” (SADT); que vienen del mundo de la computación y se basan en el modelado de los sistemas de información. Este punto se ve reforzado por Coad y Yourdon (Coad 1991) que haciendo referencia a la teoría de la clasificación que aparece en la enciclopedia Británica, mencionan que el ser humano aprende utilizando tres métodos de organización:

- a) Diferenciación de objetos.
- b) Distinción de objetos.
- c) Formación y distinción entre diferentes clases de objetos.

Con el enfoque orientado a objetos la visión actual de que un programa de computadora es una secuencia de instrucciones por la de un sistema de objetos interactivos (Winblad 1993), en el nivel más bajo de abstracción. A un nivel medio de abstracción se puede aplicar a la organización para hacer abstracciones de actividades a lo cual se llama “Análisis de dominio” (Hernández 2001).

<sup>16</sup> Ver punto 2.3.6.3

Ventajas del enfoque orientado a objetos (Winblad 1993; Jacobson 1994):

- a) Exhaustiva, es posible descomponer todas las clases jerárquicamente.
- b) Comprensible, el negocio se describe en términos de objetos que a menudo tienen ligas directas con ocurrencias en el mundo real.
- c) Modificable, los cambios usualmente son locales para una clase dada. Por lo tanto pueden ser introducidos sin afectar otras clases en el modelo. Como los protocolos entre los objetos están, bien definidos en las respectivas clases de objetos un objeto puede ser intercambiado por otro siempre y cuando tengan el mismo protocolo.
- d) Adaptable, la especialización es posible con ayuda de los mecanismos de herencia de las clases existentes.
- e) Reusable, las clases se pueden construir y manejar como componentes, cuyas propiedades pueden ser reutilizadas.
- f) Aumenta la capacidad de programación del usuario final.
- g) Permite administrar la complejidad.

La Orientación a Objetos exige procedimientos y estándares comunes para poder utilizar librerías de objetos y para ensamblar bibliotecas personalizadas, de aquí que “El rey de la industria del software será la compañía que controle las interfaces de programación (APIs) para la comunicación de mensajes a los objetos.

La misma autora señala que cuando los entornos sean completamente orientados a objetos, cambiará también el ciclo de vida del desarrollo de software, este cambio también afecta a la Ingeniería de Requerimientos de software. La confección de prototipos será sustituida por la simulación. Los usuarios o programadores construirán modelos que simularán su aplicación en tiempo real. Modificarán los modelos de aplicación hasta que se comporten correctamente y después solo necesitarán una orden de ejecución para obtener la solución final (Winblad 1993).

Algunos métodos orientados a objetos son:

- a) El Racional de Booch (Booch 1996) que divide los proyectos de software según el foco de atención por el que se manejan por el:
  - Calendario.
  - Requerimientos
  - Documentación.



- Calidad.
- Arquitectura.

Los proyectos orientados a requerimientos observan típicamente el siguiente proceso:

- Enumeración de todas las funciones del sistema.
- Diseño de los componentes.
- Implementación de los componentes.

Las actividades principales del macro proceso que se refieren al desarrollo de requerimientos son: la conceptualización, que no siempre es necesaria; el análisis, que introduce el uso de escenarios, también denominados Casos de Uso (Hernández 2001), como parte esencial de la captura de requerimientos; y el diseño. Esta metodología se describe más ampliamente en “Análisis y Diseño Orientado a Objetos (Booch 1994) .

- b) El de Fusión de Coleman et alli (Coleman 1994), que integra otros métodos como el de Boock, el de la “Colaboración responsabilizada de clase” (CRC), los métodos formales y las técnicas del modelado de objetos.
- c) El enfoque integral del Diseño de Sistemas Orientados a Objetos de Yourdon (Yourdon 1994).

La orientación a objetos pertenece exclusivamente a la estructura del programa y ni los requerimientos, ni las especificaciones describen la estructura del programa, por lo que no existen los requerimientos orientados a objetos (Kovitz 1999).

### **2.3.6.6 El UML**

El UML<sup>17</sup> es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura de decisiones y conocimiento de los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener y controlar la información sobre tales sistemas (Rumbauch 1999).

Fue desarrollado en 1994 en un esfuerzo por simplificar y consolidar el gran número de métodos de desarrollo orientado a objetos que habían surgido. En 1996 la OMG<sup>18</sup> publicó un petición de propuestas para un enfoque estándar sobre el modelado orientado a objetos y Booch, Jacobson y Rumbauch autores del UML empezaron a trabajar con metodólogos y desarrolladores de otras

---

<sup>17</sup> Unified Modeling Language o Lenguaje Unificacdo de Modelado

compañías para generar una propuesta atractiva a los miembros de la OMG, misma que fue adoptada unánimemente en noviembre de 1997

### **2.3.6.6.1 El UML y patrones.**

Los diseñadores expertos en orientación a objetos van formando un amplio repertorio de principios generales y de expresiones que los guían al crear software, a este repertorio le asignamos el nombre de patrones. El patrón es una descripción de un problema y su solución que recibe un nombre y que puede aplicarse a otros contextos.

Larman (Larman 1999) describe el uso de los patrones para asignar responsabilidades (GRASP) para evitar la gran variación en la calidad presentada en la interacción de los objetos y la asignación de responsabilidades necesarios para que los objetos se envíen mensajes entre ellos y lleven a cabo las operaciones. Los patrones GRASP son:

- a) Experto. Asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad. Responde al problema ¿Cuál es el principio fundamental en virtud del cual se asignan las responsabilidades en el diseño orientado a objetos?
- b) Creador. Asignarle a la clase B la responsabilidad de crear una instancia de clase A. Responde al problema ¿Quién debería ser responsable de crear una nueva instancia de alguna clase?
- c) Alta cohesión. Asignar una responsabilidad de modo que la cohesión siga siendo alta. Responde al problema ¿Cómo manejar la complejidad dentro de límites manejables?
- d) Bajo acoplamiento. Asignar una responsabilidad para mantener bajo acoplamiento. Responde al problema ¿Cómo dar soporte a una dependencia escasa y a un aumento de la reutilización?
- e) Controlador. Asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase. Responde al problema ¿Quién debería encargarse de atender un evento de un sistema?
- f) Polimorfismo. ¿Cómo manejar las alternativas basadas en el tipo?, ¿De qué manera crear componentes de software conectables?
- g) Fabricación pura. ¿A quién asignar la responsabilidad cuando uno está desesperado y no quiere violar los patrones Alta Cohesión y Bajo Acoplamiento?

- h) Indirección. ¿A quién asignarán las responsabilidades a fin de evitar acoplamiento directo?, ¿De qué manera se desacoplarán los objetos de modo que se obtenga un Bajo Acoplamiento y se conserve un alto potencial de reutilización?
- i) No hables con extraños. ¿A quién asignar las responsabilidades para evitar conocer la estructura de los objetos indirectos?

### **2.3.6.7 Diseño Centrado en el Usuario.**

El concepto tiene dos acepciones, una considera al usuario como los técnicos que van a trabajar con los requerimientos para desarrollar un proyecto (Gilmore 1994), y la otra acepción que tiene que ver con permitir al usuario final o cliente su interacción, incorporación y retroalimentación en el diseño (McConnell 1997).

#### **2.3.6.7.1 Beneficios del desarrollo orientado al cliente.**

- a) Mejora de la eficiencia.
- b) Menor trabajo repetido
- c) Reducción de riesgos.
- d) Ausencia de fricción.

#### **2.3.6.7.2 Métodos orientados al cliente**

- a) Planificación
  - a. Selección de un modelo de ciclo de vida apropiado, que le ofrezca al cliente signos tangibles del progreso. Para los cuales pueden servir el modelo en espiral, la entrega evolutiva, el prototipado evolutivo y la entrega por etapas.
  - b. Identificación del cliente real, al jefe o a la persona que toma las decisiones.
  - c. Establecer un método eficiente de interactuar con el cliente. De ser posible insistir en tratar con un solo cliente.
  - d. Crear un proyecto satisfactorio para todos. Utilizando la Teoría –W, para identificar condiciones de victoria para todas las partes involucradas.
  - e. Gestión de riesgos. Prestar especial atención a los riesgos relacionados con los clientes.
- b) Análisis de requerimientos.
  - a. Utilizar métodos de recopilación de requerimientos que ayuden a los clientes a representar qué es lo que quieren, como, por ejemplo prototipado de la interfaz de usuario, prototipado evolutivo y sesiones de desarrollo conjunto de aplicaciones.

- b. Crear grupos distintos que expongan los distintos puntos de vista sobre lo que quiere el cliente.
- c. Grabación en video de los clientes, utilizando el software.
- c) Diseño.
  - a. Utilizar métodos de diseño que permitan a los clientes cambiar de opinión ocasionalmente.
- d) Construcción. (No incluida en el alcance de este trabajo).
  - a. Empleo de métodos de implementación que creen un código legible y modificable, que le permita mejorar su capacidad para responder a cambios solicitados por los clientes.
  - b. Utilizar métodos de monitorización del progreso, como hitos miniatura que le permitan informar al cliente sobre el progreso.
  - c. Selección de un modelo de ciclo de vida que ofrezca al cliente signos de progreso tangible.

#### **2.3.6.8 Modelo de madurez para el desarrollo de software (CMM).**

El Modelo de Madurez y Capacidad “Capability Maturity Model” o CMM para el desarrollo de software describe los principios fundamentales de la madurez del proceso de desarrollo de software e intenta ayudar a las organizaciones que producen software para mejorar la madurez de sus procesos de desarrollo proponiendo un camino que evoluciona desde un proceso ad hoc a veces caótico a procesos maduros y disciplinados de producción (Paulk 1994). El CMM esta organizado en cinco niveles de madurez. Un nivel de madurez es un plano evolutivo muy bien definido hacia el logro de procesos maduros más maduros de software. Cada nivel de madurez provee una capa para la mejora continua del proceso.

Otra ventaja del CMM descrita por Card (Card 2000) es que utiliza al “Prevención de defectos” que consiste en el análisis de los defectos que fueron encontrados en el pasado y tomar acciones específicas para prevenir la ocurrencia de esos mismos defectos en el futuro. Los defectos pudieron haber sido identificados en otros proyectos, tanto como en las etapas tempranas del proyecto actual y son un mecanismo para transmitir lecciones aprendidas entre proyectos.

### **2.3.6.8.1 Los requerimientos en CMM.**

Para alcanzar el nivel 2 de CMM, una organización debe demostrar que satisface las metas en seis áreas de procesos clave<sup>19</sup> del desarrollo y administración del software. La administración de los requerimientos es una de estas áreas y sus metas son las siguientes(Wiegers 1999):

- a) Los requerimientos de software están controlados al establecer una línea base para la ingeniería de software y para el uso administrativo.
- b) La planeación del software, los productos y las actividades, se mantienen consistentes con los requerimientos de software.

### **2.3.6.8.2 Guía de buenas prácticas de la Ingeniería de Requerimientos.**

La Guía de buenas prácticas de la Ingeniería de Requerimientos<sup>20</sup> (REGPG) es una variante del CMM propuesta por Sawyer, Sommerville y Viller (Sawyer 1999), tiene tres niveles de madurez en vez de cinco: inicial, repetible y definido. En este método no se busca la certificación, por lo que es menos riguroso, pero con un enfoque hacia los requerimientos.

## **2.3.7 Técnicas del proceso de requerimientos.**

El estándar 1233 de la IEEE (IEEE1233 1998) identifica las siguientes técnicas para identificar requerimientos:

1. Talleres de trabajo estructurados.
2. Lluvia de ideas o sesiones para resolver problemas.
3. Entrevistas.
4. Encuestas o cuestionarios.
5. Observación de patrones de trabajo (Ej. estudios de tiempos y movimientos).
6. Observación de los sistemas organizacionales, políticos y del medio ambiente (Ej. sociogramas).
7. Revisión de la documentación técnica.
8. Análisis de mercado.
9. Evaluación de los sistemas competitivos.
10. Ingeniería inversa.
11. Simulaciones.
12. Prototipos.

---

<sup>19</sup> Key process areas (KPA's)

<sup>20</sup> Requirements Engineering Good Practice Guide.

### 13. Comparación de sistemas y procesos.

#### **2.3.7.1 Lluvia de ideas.**

Es uno de los tantos enfoques de grupo, probablemente el más viejo y mejor conocido. La idea es permitirle a la gente juntarse e inspirarse unos a los otros en la fase creativa de generación de ideas del proceso de solución del problema. Es usado para generar nuevas ideas, liberando la mente para aceptar cualquier idea que se sugiera, y así se da libertad a la creatividad. Este enfoque ha sido ampliamente usado en el diseño. Es usado en las primeras etapas en el proceso de recolección para la generación rápida de ideas y problemas en un dominio específico y está enfocado a la cantidad de respuestas. Los métodos de agrupamiento pueden emplearse después de la tormenta de ideas para mejorar la salida de la sesión de grupo.

Normalmente se aplica en las primeras fases del desarrollo cuando se conoce poco del diseño real y se necesitan ideas.

El proceso de grupo es provechoso en sí mismo y crea un sentimiento de propiedad de los resultados. Todos los del grupo tienen crédito por las buenas ideas. Incluso cuando hay expertos presentes, tiene la ventaja de evitar las críticas tempranas de las ideas y reduce los comentarios evaluativos negativos en los comienzos del diseño.

Los recursos humanos son las más importantes en el éxito de este enfoque. Se obtienen mejores resultados, mientras más experiencia tiene la gente en el campo. Pueden participar de 5 a 12 personas y se necesita que la sesión no tome más de un hora.

#### **2.3.7.2 Entrevistas**

Se emplean para reunir información proveniente de personas o grupos. Durante la misma se conversa, no se interroga. A menudo es la mejor fuente de información cualitativa (opiniones, políticas, descripciones subjetivas de actividades y problemas). Permiten descubrir áreas mal comprendidas, expectativas poco realistas e incluso indicadores de resistencia hacia el sistema propuesto.

Un entrevistador les hace preguntas a los expertos del dominio para obtener conocimientos del mismo. No son tan simples como parecen. Pueden ser de tres tipos: no estructuradas, semi-estructuradas y estructuradas. Hay autores que sólo las clasifican en estructuradas y no estructuradas. El tipo, detalle y validez de los datos recolectados varía con el tipo de entrevista y experiencia del entrevistador.

Es el método que más se usa y del que más se abusa para encontrar qué es lo que quieren los usuarios.

La aparente simplicidad de la entrevista no estructurada descansa en el hecho de que la mayoría de los adultos cree tener esta habilidad debido a su experiencia de conversación social. Utilizan un formato pregunta-respuesta y son apropiados cuando el analista desea adquirir información general sobre un sistema. Anima a los usuarios a compartir sentimientos, ideas, creencias.

La semi-estructurada es útil en situaciones donde se tienen que comprender asuntos muy amplios, pero el rango de reacciones de la persona que responde no es conocido o se sospecha incompleto.

La estructurada debería ser llevada a cabo sólo en situaciones donde el rango de réplicas de los demandados es casi tan bien conocida y hay una necesidad de juzgar la fortaleza de cada opinión. Utiliza preguntas estándar en un formato de respuesta cerrada. Utiliza un conjunto anticipado de respuestas.

Sirve para identificar posibles áreas de análisis más detallado. Los datos recogidos dan información sobre reglas generales y principios. Es más rápida que los métodos de observación. Son útiles para investigar eventos que ocurren esporádicamente. Son populares, conocidas y ampliamente aceptadas.

Tienen la característica, extremadamente positiva, de dar al entrevistado la sensación de que sus entradas al proceso se están tomando en cuenta. El entrevistado adquiere intereses en el asunto y se da una buena imagen de la compañía que le hace la entrevista. Puede ser de especial utilidad para reunir información de personas que no se comunican por escrito de forma adecuada o que no disponen de tiempo para llenar un cuestionario.

Tiene como limitación que las preguntas pueden ser formuladas de muchas formas y las respuestas interpretadas de muchas maneras. El entrevistador debe adquirir conocimientos del dominio del problema para saber que preguntar. Frecuentemente lo que la gente dice es diferente de lo que hace.

Es un método que requiere considerable preparación para parte del entrevistador.

### **2.3.7.3 Encuestas**

Es la aplicación de cuestionarios estándar a muestras grandes de población. Pueden ayudar a determinar las preferencias de los consumidores, prácticas de trabajo y actitudes. Hay dos tipos: las abiertas (el que responde es libre de expresar lo que quiera) y las cerradas (se le pide al que responde que seleccione de entre las respuestas disponibles).

En la encuesta abierta se pueden descubrir sentimientos, opiniones y experiencias generales para explorar un proceso o problema. La cerrada controla el marco de referencia al limitar las respuestas, por lo que se recomienda aplicar para recoger información basada en hechos reales.

Las encuestas abiertas se aplican en los mismos tipos de situaciones que las entrevistas semi-estructuradas. Deben ser usadas de preferencia en situaciones donde el grupo de usuarios está

diseminado geográficamente o donde el esfuerzo-persona es pequeño. A menos que el grupo de usuarios esté especialmente motivado, la proporción de respuestas del 20% o menos es común en los cuestionarios. Por el contrario, si hay muchas personas disponibles para recolectar requerimientos y el grupo de usuarios es accesible, mejor se recomiendan las entrevistas.

Son relativamente rápidas y baratas de aplicar. Los resultados pueden estar sujetos a análisis estadísticos. Al anonimato de los encuestados puede conducir a respuestas más honestas.

En su contra podemos argumentar que:

- El diseño de cuestionarios no es fácil.
- No permite al analista observar las expresiones o reacciones de los encuestados.
- La respuesta puede ser limitada ya que es posible que no tenga mucha importancia para el encuestado llenar el cuestionario.

El costo de uso depende mucho de la complejidad de la encuesta y del número de personas a las que se les requiere aplicar. Los costos de distribución son probablemente los más grandes, así como los de teléfono, si se realiza un seguimiento. El diseño requiere de cierta experiencia y puede ser costoso si un personal inexperto intenta usarlo por primera vez.

#### **2.3.7.4 Prototipos**

Existen varios tipos de prototipos entre los que se encuentran los prototipos en papel, prototipos rápidos, prototipos de video, a continuación se mencionan algunos prototipos.

##### **2.3.7.4.1.1 Prototipo Mundo de Oz.**

Involucra a un usuario interactuando con un sistema computacional que es realmente operado por un desarrollador oculto (referido como el mago). El “mago” procesa entradas de un usuario y simula las salidas del sistema. Durante este proceso al usuario se le deja creer que está interactuando directamente con el sistema. Este tipo de prototipo es beneficioso en los inicios del diseño y es un medio para estudiar las expectativas y requerimientos. Este enfoque es particularmente apropiado para explotar posibilidades de diseño en sistemas que urge implementar.

Se emplea en “interfaces inteligentes” que caracterizan agentes, asesores y/o procesamiento de lenguaje natural. En aplicaciones de multimedia y telemática. Apple lo ha adoptado.

Permite que los requerimientos de usabilidad y problemas sean explorados en una etapa temprana del proceso de diseño, particularmente en sistemas que van más allá de la disponibilidad de



una tecnología. El miembro del equipo que hace el papel de “mago” puede lograr una valiosa penetración al involucrarse cercanamente con la actividad del usuario.

Lo limita el hecho que, la persona que juega el papel de “mago”, tiene que apreciar la funcionalidad del sistema propuesto para realizar una representación convincente. Requiere un compromiso mayor de recursos que otros enfoques de prototipos, como aquellos que se realizan sólo sobre papel.

Puede ser difícil llevarlo a cabo en situaciones donde haya un gran desarrollo gráfico en la interfaz.

Su aplicación implica el uso de dos computadoras, una para el usuario y otra para el “mago”. Se requieren dos personas del equipo para conducir la evaluación: una es el “mago” y la otra instruye al usuario y registra la sesión. El “mago” debe ser un experto del equipo de diseño, de manera que las respuestas del sistema sean lógicas y no fuera de las posibilidades reales. El tiempo que tome depende del dominio de las tareas y de la cantidad de usuarios expuestos al prototipo.

No requiere entrenamiento obligatorio. Sin embargo, muchos reportes sobre el enfoque se originan en trabajos de investigación donde los detalles de la técnica se esquematizan. El enfoque requiere experiencia y/o entrenamiento para alcanzar competencia.

#### **2.3.7.4.1.2 Prototipos en papel**

Este método destaca el uso de materiales y equipos simples para crear una simulación en papel de una interfaz o sistema. Son un medio valioso y no costoso para evaluar e iterar opciones de diseño antes que un equipo este listo para la implementación. Los elementos de la interfaz como menús, ventanas diálogos e iconos, se pueden esquematizar en el papel o realizar creaciones más avanzadas usando tarjetas, acetatos, plumas. El resultado es a veces llamado “prototipo de baja fidelidad”. Cuando se ha preparado, un miembro del equipo de diseño se sienta frente al usuario y juega el papel de computadora, moviendo los elementos de la interfaz en respuesta a las acciones de los usuarios. El usuario realiza selecciones y activa los elementos de la interfaz usando su dedo como si fuera un ratón y escribiendo las entradas que serían tecleadas. Otra persona facilita la sesión dando instrucciones para las tareas e incitando al usuario a expresar sus pensamientos e impresiones. Puede haber otros observadores o pudiera ser filmado.

#### **2.3.7.4.1.3 Prototipos en video**

Permite a los diseñadores crear una simulación, basada en video de la funcionalidad de la interfaz, usando materiales y equipos simples. Los elementos de la interfaz son creados usando papel, plumas, acetatos. Los usuarios no pueden interactuar directamente con el prototipo aunque pueden ver

y comentar sobre la simulación del video. Tiene la ventaja de poder detectar desde etapas tempranas algunos problemas de usabilidad. Su limitación principal es que requiere que el personal esté familiarizado con el sistema que se pretende crear y no se puede interactuar, como se dijo anteriormente.

#### **2.3.7.4.1.4 Prototipos Rápidos**

Relacionado con el desarrollo de diferentes conceptos propuestos a través de prototipos de hardware y software y su evaluación. El desarrollo de una simulación o prototipo del futuro sistema puede ser muy útil, permitiéndoles a los usuarios visualizar el sistema y dar retroalimentación sobre éste. Así, este puede ser usado para clasificar las opciones de requerimientos de los usuarios.

El prototipo rápido es descrito como un método basado en computadora que ayuda a reducir el ciclo de desarrollo iterativo. Los prototipos interactivos se desarrollan y pueden ser rápidamente reemplazados o cambiados en línea con la retroalimentación de diseño. Esta retroalimentación puede derivarse de los colegas o de la experiencia de los usuarios.

Dentro de los círculos de la ingeniería de software, el método está cercanamente asociado con los sistemas de administración de interfaces de usuarios y varias herramientas de soporte al diseño. Las últimas herramientas ofrecen al diseñador, bibliotecas de procedimientos y elementos de interfaz gráfica para definir la estructura lógica del software y su “look and feel”. El título se refiere a un enfoque adoptado por los desarrolladores de software en el cual los prototipos exhiben una mayor fidelidad con el producto final, que aquellos creados como parte de otros métodos, tal como los prototipos en papel.

El método requiere recursos técnicos más sofisticados que en los métodos de prototipos de baja fidelidad. Un costo adicional de su uso es el nivel de experiencia requerido para dominar las herramientas de desarrollo, unido al tiempo necesario para implementar un prototipo de software. Si las pruebas basadas en usuarios son llevadas a cabo con el prototipo de desarrollo, entonces también se necesitan recursos para esta actividad.

Para ver otras descripciones se puede consultar el Anexo A de la tesis Hernández (Hernández 2001), o bien la página de Internet de Métodos para la especificación de requerimientos orientados al usuario de la organización europea Respect (Kirakowski 1997).

Goguen citado en Siddiqui (Siddiqui 1996) dice que los métodos actuales para obtener información tácita, tales como cuestionarios, entrevistas, introspección, grupos enfocados, son inadecuados. En su lugar el propone la “etnometodología”. Con este acercamiento el analista recoge

información en situaciones de ocurrencia natural, en donde los participantes están ocupados en actividades ordinarias de la vida diaria. Más aún el analista no impone las llamadas categorías “objetivas” preconcebidas para explicar lo que está ocurriendo, sino las categorías que los mismos participantes utilizan para comunicarse.

### **2.3.8 Herramientas de la Ingeniería de Requerimientos.**

El uso de las herramientas en el proceso de requerimientos se hace necesario apoyar esta fase de la producción de software o de sistemas en la que tradicionalmente se emplean pocos recursos. Palmer (Palmer 1992) dice que los estudios muestran, que el 5% del total del costo de un sistema se destinan al diseño y al desarrollo; sin embargo se requiere 100 veces más esfuerzo para corregir errores descubiertos en los requerimientos, del que se necesitaría para corregir los mismos errores si hubieran sido descubiertos durante la ingeniería de requerimientos. Leffingwell citado por Wiegers (Wiegers 1999) dice que entre un 40 y un 60% de los defectos encontrados en un proyecto de software pueden rastrearse como errores cometidos durante la etapa de requerimientos.

A continuación se listan algunas de las herramientas encontradas en el mercado actual:

La QARCC de Boehm e In es una herramienta basada en la administración del conocimiento que puede ayudar a los usuarios, desarrolladores y clientes a analizar los requerimientos y a identificar conflictos entre ellos.(Boehm 1996).

Concept Base (Nissen 1996) es una herramienta de metamodelado con facilidades avanzadas de consulta, ayuda en la captura, monitoreo y resolución de perspectivas múltiples de los requerimientos. Los metamodelos son modelos de modelos: mientras los requerimientos de modelos son representaciones abstractas de un mundo real existente o deseado, los metamodelos son representaciones abstractas de modelos de requerimientos existentes o deseados, así como de sus interrelaciones.

Traceability of Object-Oriented Requirements (TOOR) (Pinheiro 1996), para rastrear los requerimientos ayuda a verificar las características del sistema contra las especificaciones de requerimientos, identificar las fuentes de error y lo más importante, administrar los cambios.

Rational Rose(IBM 2003) es una herramienta de diseño de software de lenguaje de modelado unificado orientado a objetos, para modelado visual y construcción de componentes de aplicación de

software a nivel empresarial. Se utiliza para crear modelos visuales de marcos de trabajo para aplicaciones utilizando clases, usuarios, casos de uso, objetos y mensajes o relaciones entre ellos, en diagramas secuenciales utilizando símbolos de “agarrar y soltar”.

Rational Rose documenta los diagramas mientras están siendo construidos y genera código en lenguaje de preferencia del diseñador con: C++, Visual Basic, Java, Oracle8, CORBA o Lenguaje de definición de datos. Mediante un módulo que se vende por separado también puede generar código en Delphi. Las dos ventajas más populares de Rational Rose son su aplicabilidad a desarrollo iterativos o evolutivos y segundo hace ingeniería de “viaje completo”, retrocediendo y actualizando el resto del modelo hasta asegurarse de que el código permanece consistente.

En esta tesis se utilizó la metodología descrita en el Cuaderno de trabajo y el Cuaderno de prácticas de Modelado con UML editados en esta universidad (Hernández 2003; Hernández 2003b), que emplea el Rational Rose Enterprise Edition 2002 como herramienta de trabajo.

### **2.3.9 Mejora de procesos de requerimientos.**

#### **2.3.9.1 Los objetivos de los procesos de mejora.**

Según Humphrey (Humphrey 1990) los objetivos de los procesos de mejora son dos:

- 1.- Corregir los problemas que se han encontrado en proyectos previos o actuales.
- 2.- Anticipar y prevenir problemas que se pudieran encontrar en proyectos futuros.

#### **2.3.9.2 Fundamentos de mejora de procesos.**

- 1.- La mejora del proceso debe ser evolutiva, continua y cíclica.
- 2.- La gente y las organizaciones cambian solamente cuando tienen un incentivo para hacerlo.
- 3.- Los cambios del proceso deben estar orientados a metas.
- 4.- Tratar las actividades de mejora como mini proyectos.

### 2.3.9.3 Ciclo de mejora de procesos

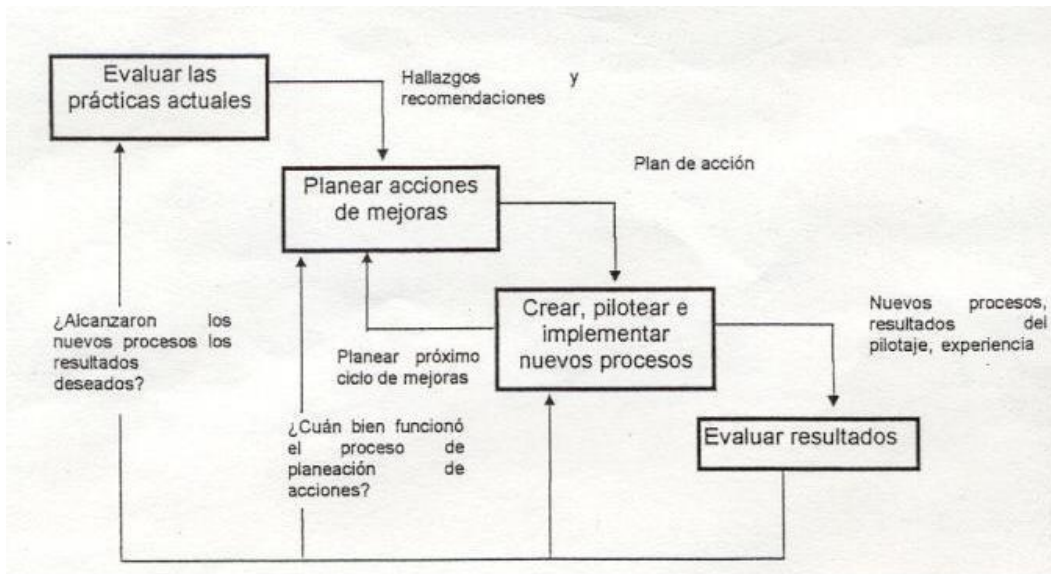


Fig 2.3.9.3.1 Ciclo de mejora de procesos

- 1.- Evaluar las prácticas actuales.
- 2.- Planear acciones de mejora
- 3.- Crear, pilotear e implementar nuevos procesos. Aún los equipos motivados y receptivos tienen una capacidad limitada para absorber el cambio, por lo que no hay que colocar demasiadas expectativas nuevas en un proyecto o grupo a la vez.
- 4.- Evaluar los resultados. Tomar en cuenta la curva de aprendizaje en la mejora de procesos.

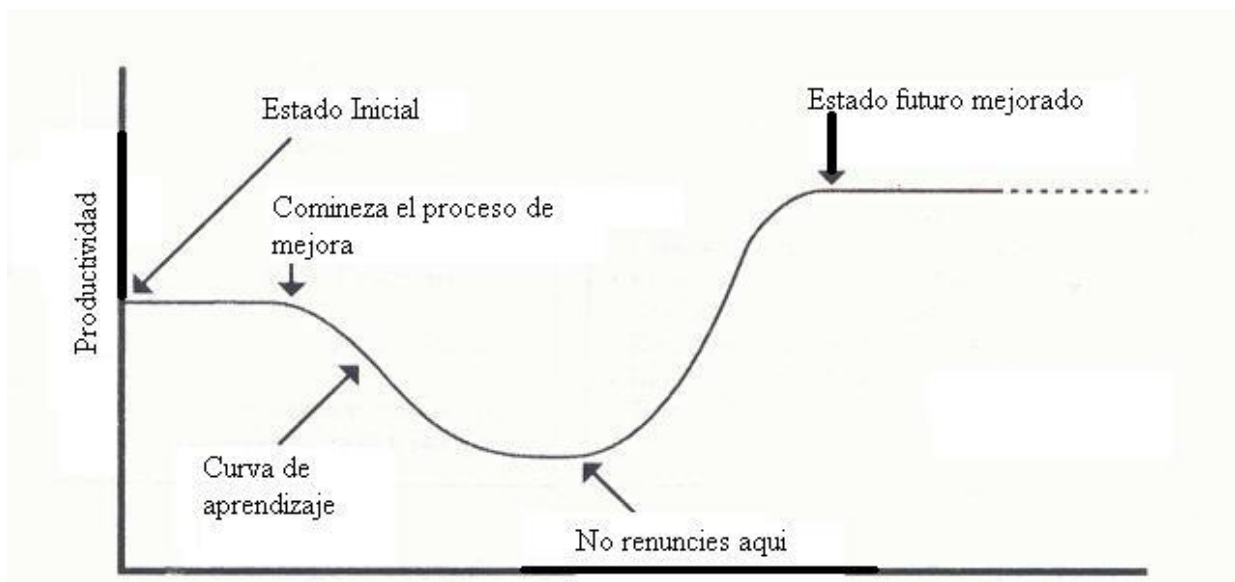


Fig. 2.3.9.3.1

### **2.3.9.3.1 Marco de trabajo de la mejora de procesos.**

El mejoramiento de los procesos de software debe seguir una estrategia y ésta a su vez, debe estar basada en un marco de trabajo. El marco de trabajo propuesto por Zahran (Zahran 1998) consta de 4 componentes:

- a) Infraestructura
- b) Programa (roadmap)
- c) Método de evaluación
- d) Plan de mejoramiento.

La infraestructura se refiere tanto a la infraestructura de administración como técnica y soporta y apoya al proceso. El programa es el modelo que caracteriza al proceso de software y da un enfoque paso a paso hacia los procesos efectivos de software; especifica etapas y niveles a ser alcanzados y criterios para lograrlo (como CMM o Trillium). EL método de evaluación (Bootstrap o ISO/IEC 15504) permite caracterizar el estado actual del proceso, sus prácticas e infraestructura y sus resultados deben incluir: fortalezas, debilidades y recomendaciones para mejorar. El plan de mejoramiento contiene acciones específicas para las mejoras.

En nuestro caso el área a cubrir es la de requerimientos y para su mejora se deberá determinar cada uno de los cuatro elementos anteriores del marco de trabajo, así como la estrategia para las mejoras.

## **2.4 Administración de Proyectos de Requerimientos.**

### **2.4.1 Conceptos básicos.**

#### **2.4.1.1 Proyecto.**

Un proyecto es un esfuerzo<sup>21</sup> temporal, emprendido para crear u producto o servicio único. Temporal significa que el producto o servicio tienen un principio y un fin definidos. Único significa que el producto o servicio es diferente de alguna manera distinguible de todos los productos o servicios similares (PMI 2000).

---

<sup>21</sup> Endeavor.

### **2.4.1.2 Proceso.**

Un proceso puede definirse simplemente como un acercamiento sistemático que se realiza para alcanzar un propósito específico. En el contexto de los procesos de desarrollo de software un proceso se define como el juego de actividades ordenadas, que después de ser completadas resultan en un software “producto” (Whitten 1995).

El PMI define un proceso como “una serie de acciones que conllevan un resultado. Los procesos de administración de proyectos tienen que ver con la descripción y organización del trabajo del proyecto. Los procesos orientados a la producción tienen que ver con especificar y crear el producto del proyecto” (PMI 2000).

### **2.4.1.3 Administración.**

La administración es la planeación, organización, dirección y control de las personas que trabajan en una organización (o en un proyecto) y de la serie de tareas y actividades que desempeñen (Hellriegel 1998).

### **2.4.1.4 Administración de requerimientos.**

Es un enfoque sistemático para obtener, organizar y documentar los requerimientos de un sistema, y los procesos que establecen y mantienen acuerdos entre el cliente y el equipo del proyecto en el cambio de los requerimientos del sistema (Leffingwell 2000).

### **2.4.1.5 Modelos de ciclo de vida**

Como el emprender un proyecto que es único envuelve un cierto grado de incertidumbre, se suele dividir en fases para proveer un mejor control administrativo y los vínculos apropiados a las operaciones en curso. Colectivamente las fases del proyecto se conocen como el ciclo de vida del proyecto.

Algunos ejemplos de modelos de ciclo de vida más mencionados por los autores encontrados son:

Codificar y arreglar.

Cascada

Incremental

Iterativo

Siddiqui (Siddiqui 1996) dice que la mayoría de los profesionales de la ingeniería de software creen que la fase de los requerimientos de un proyecto tiene su propio ciclo de vida, que a su vez se divide en otras fases, a las que se les ha dado diferentes etiquetas.

En los 80's Herb Krasner propuso 5 fases:

- a) Identificación de la necesidad y análisis del problema.
- b) Determinación de los requerimientos.
- c) Especificación de los requerimientos.
- d) Realización de los requerimientos.
- e) Administración de cambio de los requerimientos.

Más recientemente Matthias Jarke y Klaus Pohl propusieron un ciclo de tres fases:

- a) Obtención.
- b) Expresión.
- c) Validación.

Colin Potts y colegas optaron por un ciclo iterativo con tres fases, en el que usan escenarios de uso de los involucrados para identificar y validar los requerimientos.

- a) Documentación.
- b) Discusión.
- c) Evolución.

#### **2.4.1.6 Identificar el modelo del ciclo de vida.**

Complejidad del producto.

Tamaño del proyecto.

El grado en el que los requerimientos del producto están documentados y comprendidos.

Necesidad temprana de la disponibilidad de la función del producto.

Necesidad del involucramiento de los clientes durante el desarrollo.

Cliente único o múltiple.



<b>Capacidades del modelo de ciclo de vida</b>	<b>Casacada pura</b>	<b>Codificar y corregir</b>	<b>Espiral</b>	<b>Cascadas modificadas</b>	<b>Prototipado Evolutivo</b>
Trabaja con poca identificación de los requerimientos	Malo	Malo	Excelente	Medio a Excelente	Excelente
Trabaja con poca comprensión sobre la arquitectura	Malo	Malo	Excelente	Medio a Excelente	Malo a medio
Genera un sistema altamente fiable	Excelente	Malo	Excelente	Excelente	Medio
Genera un sistema con amplio desarrollo	Excelente	Malo a medio	Excelente	Excelente	Excelente
Gestionar riesgos	Malo	Malo	Excelente	Medio	Medio
Estar sometido a una planificación predefinida	Medio	Malo	Medio	Medio	Malo
Requiere poco tiempo de gestión	Malo	Excelente	Medio	Excelente	Medio
Permite modificaciones a medio camino	Malo	Malo a excelente	Medio	Medio	Excelente
Ofrece a los clientes signos visibles de progreso	Malo	Medio	Excelente	Medio	Excelente
Ofrece a la directiva signos visibles de progreso	Medio	Malo	Excelente	Medio a Excelente	Medio
Requiere poca sofisticación para los directivos y desarrolladores	Medio	Excelente	Malo	Malo a medio	Malo

Tabla 2.4.1.6.1 Fortalezas de algunos Modelos de ciclos de Vida (McConnell 1997)

<b>Capacidades del modelo de ciclo de vida</b>	<b>Entrega por etapas</b>	<b>Entrega evolutiva</b>	<b>Diseño por planificación</b>	<b>Diseño por herramientas</b>	<b>Software comercial disponible</b>
Trabaja con poca identificación de los requerimientos	Malo	Medio a excelente	Malo a medio	Medio	Excelente
Trabaja con poca comprensión sobre la arquitectura	Malo	Malo	Malo	Malo	Malo
Genera un sistema altamente fiable	Excelente	Medio a excelente	Medio	Malo a excelente	Malo a excelente
Genera un sistema con amplio desarrollo	Excelente	Excelente	Medio a excelente	Malo	N/A
Gestionar riesgos	Medio	Medio	Medio a excelente	Malo a medio	N/A
Estar sometido a una planificación predefinida	Medio	Medio	Excelente	Excelente	Excelente
Requiere poco tiempo de gestión	Medio	Medio	Medio	Medio a excelente	Excelente
Permite modificaciones a medio camino	Malo	Medio a excelente	Malo a medio	Excelente	Malo
Ofrece a los clientes signos visibles de progreso	Medio	Excelente	Medio	Excelente	N/A
Ofrece a la directiva signos visibles de progreso	Excelente	Excelente	Excelente	Excelente	N/A
Requiere poca sofisticación para los directivos y desarrolladores	Medio	Medio	Malo	Malo	Medio

Tabla 2.4.1.6.2 Fortalezas de algunos Modelos de ciclos de Vida (McConell 1997)

## **2.4.2 El contexto de la administración de proyectos.**

### **2.4.2.1 Involucrados en el proyecto.**

Son los individuos y organizaciones que están involucrados activamente en el proyecto, o cuyos intereses pueden ser afectados positiva o negativamente por los resultados de la ejecución del proyecto o de la terminación satisfactoria del mismo, administrador del proyecto, clientes, organización en donde se desarrolla el producto, el patrocinador, etc.

### **2.4.2.2 Influencias organizacionales.**

Las influencias organizacionales son dos: estructura organizacional y cultura organizacional.

#### **2.4.2.2.1 Estructura organizacional**

Sistema formal de relaciones de trabajo para la distinción e integración de las tareas (la determinación de quien hará qué y cómo se combinarán los esfuerzos).

##### **2.4.2.2.1.1 Elementos de la Estructura Organizacional**

- a) **Especialización:** Proceso de identificación de tareas particulares y su asignación a individuos o equipos capacitados para desempeñarlas.
- b) **Estandarización:** Uniformación y sistematización de los procedimientos que deben seguir los empelados en el desempeño de sus labores.
- c) **Coordinación:** Procedimientos formales e informales para la integración de las actividades desempeñadas por distintos individuos, equipos y departamentos de una organización.
- d) **Autoridad:** derecho a decidir y actuar.

#### **2.4.2.2.2 Cultura Organizacional**

Patrón propio de supuestos, valores y normas compartidos que modela las actividades de socialización, lenguaje, símbolos, historia y prácticas de una organización.

Las implicaciones son: desempeño organizacional, satisfacción individual, satisfacción del cliente, otras.

### **2.4.2.3 Influencias Generales**

Incluye los factores externos que generalmente influyen en la totalidad o la mayoría de las organizaciones. El tipo de sistema económico(de libre empresa, socialista o de demanda planeada, por ejemplo) y las condiciones económicas(ciclos de expansión y recesión y nivel general de vida); tipo de sistema político(democracia, dictadura o monarquía); condición del ecosistema(grado de contaminación del suelo, el agua y el aire); demografía(edad, género, raza, origen étnico y nivel de estudios de la población), y sistema cultural(valores, creencias, idioma e influencias religiosas). Estos aspectos del entorno general producen las fuerzas culturales, competitivas, tecnológicas y político-legales con efectos en las organizaciones.

### **2.4.3 Áreas de conocimiento de la administración de proyectos.**

A continuación se describen las partes principales de los planes de acción de las áreas de conocimiento de la administración de proyectos que intervienen en este proyecto en particular. Las entradas, herramientas, técnicas y salidas de cada uno de los planes se puede consultar en el anexo B Planes.

#### **2.4.3.1 Integración.**

Incluye los procesos requeridos para asegurar que todos los elementos del proyecto estén coordinados apropiadamente.

##### **2.4.3.1.1 Desarrollo del plan de proyecto.**

Utiliza las salidas de otros procesos de planeación para crear un documento coherente y consistente, que puede ser utilizado para guiar tanto la ejecución del proyecto como el control del mismo.

Se utiliza para:

- Guía de ejecución del proyecto.
- Documentar los supuestos de la planeación del proyecto.
- Documentar las decisiones de la planeación del proyecto, relativas a las alternativas escogidas.
- Facilitar la comunicación entre los involucrados en el proyecto.
- Definir revisiones administrativas clave, así como su contenido extensión y tiempo.

##### **2.4.3.1.2 Control general de cambios.**

- Influir en los factores que crean los cambios, para asegurarse que los cambios son benéficos.
- Determinar que un cambio ha ocurrido.
- Administrar los cambios actuales cuando y como ocurran.

#### **2.4.3.2 Alcance.**

El alcance se puede referir a:

- Alcance del producto – las características y las funciones que tienen que ser incluidas en un producto o servicio, su terminación es medida contra los requerimientos.
- Alcance del proyecto – el trabajo que debe ser hecho para entregar un producto con características y funciones definidas, su terminación es medida contra el plan.

#### **2.4.3.2.1 Iniciación:**

Comprometer a la organización para iniciar la siguiente fase del proyecto, es el proceso de reconocer formalmente que la existencia de un proyecto o de que un proyecto que ya existía deberá continuar a su siguiente fase.

La autorización de los proyecto se produce por: demanda del mercado, necesidades del negocio, requerimiento de los clientes, avance en la tecnología o de un requerimiento legal.

#### **2.4.3.2.2 Planeación del alcance:**

Desarrollar una declaración escrita del alcance como la base para decisiones futuras del proyecto, incluyendo en particular los criterios utilizados para determinar si el proyecto o la fase han sido completados satisfactoriamente.

#### **2.4.3.2.3 Definición del alcance:**

Subdividir las entregas principales del proyecto, en componentes más pequeños y más manejables, para mejorar la exactitud de las estimaciones de costos, tiempos y recursos, definir la línea de base de la medición y control de los rendimientos, facilitar una asignación de responsabilidades clara.

#### **2.4.3.2.4 Verificación del alcance:**

Formalizar la aceptación del alcance del proyecto, por los interesados, es diferente de control de calidad, porque está concernida por la aceptación de los resultados del trabajo y no es si estos resultados son correctos.

#### **2.4.3.2.5 Control de cambios del alcance:**

Se ocupa de influenciar a los factores que crean cambios en el alcance, para asegurarse de que los cambios son benéficos, determinar que un cambio en los alcances ha ocurrido y administrar los cambios actuales siempre y cuando se presenten.

### **2.4.3.3 Tiempos.**

#### **2.4.3.3.1 Definición de actividades.**

La definición de actividad comprende la identificación y documentación de las actividades específicas que deben ser realizadas para producir los entregables y subentregables identificados en la estructura de trabajos desglosada (ETD)

#### **2.4.3.3.2 Secuencia de las actividades**

#### **2.4.3.3.3 Estimación de la duración de las actividades**

Involucra la evaluación del número de periodos de trabajo más probables de ser necesitados para completar cada actividad especificada.

- 

#### **2.4.3.3.4 Desarrollo de la programación**

Significa determinar las fechas de inicio y terminación para las actividades del proyecto.

- 

#### **2.4.3.3.5 Control de la programación**

Involucra: a) la influencia de los factores que crean cambios en la programación, para asegurarse que estos cambios sean benéficos, b) determinar si la programación ha cambiado y c) administrar los cambios actuales cuando y como ellos ocurran.

### **2.4.3.4 Costos.**

#### **2.4.3.4.1 Planeación de recursos.**

Comprende determinar que recursos físicos (personas, equipamiento y material) y en que cantidades de cada una, deberán ser utilizadas para realizar las actividades del proyecto.

#### **2.4.3.4.2 Estimación de costos.**

Comprende desarrollar una aproximación (estimación) del costo de los recursos necesarios para completar las actividades del proyecto. Si el proyecto se va a realizar bajo contrato, se debe tener cuidado en distinguir la estimación del costo de la de precio alzado, mientras que la primera es el

resultado de una evaluación cuantitativa, la otra es producto de una decisión de negocios e implica un cargo por el producto o servicio.

#### **2.4.3.4.3 Presupuesto del costo**

Comprende la asignación de la estimación general de costo a los elementos de trabajo individuales, para establecer una línea de costo base con que medir el desempeño del proyecto.

#### **2.4.3.4.4 Control de costo**

Comprende: monitorear el desempeño del costo para detectar variaciones del plan, asegurarse de que todos los cambios apropiados están correctamente grabados en línea de costo base, prevenir cambios incorrectos, inapropiados o no autorizados, de ser incluidos en la línea de costo base e informar a todos los interesados de los cambios autorizados.

#### **2.4.3.5 Calidad.**

##### **2.4.3.5.1 Introducción.**

Calidad es una propiedad o conjunto de propiedades inherentes a una cosa, que permiten apreciarla como igual, mejor o peor, que las restantes de su especie.

Según el PMBook: Calidad es la totalidad de las características de una entidad, que se refieren a su capacidad de satisfacer necesidades implícitas o establecidas.

Son los procesos requeridos para asegurarse de que el proyecto satisfará las necesidades para las que fue emprendido.

##### **2.4.3.5.2 Planeación de la calidad.**

Implica la identificación de los estándares de calidad que serán relevantes al proyecto y a la determinación de cómo satisfacerlos.

##### **2.4.3.5.3 Aseguramiento de la calidad.**

Son todas las actividades planeadas y sistemáticas, implementadas con el plan de calidad para dar confianza de que el proyecto satisfará los estándares de calidad relevantes.

##### **2.4.3.5.4 Control de calidad.**

Involucra el monitoreo de resultados específicos del proyecto para determinar si cumplen con los estándares de calidad relevantes e identificar caminos para eliminar la causas de resultados insatisfactorios.

## **2.4.3.6 Personal.**

### **2.4.3.6.1 Introducción.**

Existe mucha literatura al respecto que puede ser consultada, y debe haber mucha sensibilidad en cuanto a cómo este conocimiento se aplicará al proyecto

La naturaleza temporal de los proyectos significa que las relaciones entre el personal y la organización, generalmente serán ambas temporales y nuevas.

La naturaleza y número de los involucrados en el proyecto cambiarán frecuentemente cuando el proyecto se mueva de una fase a otra de su ciclo de vida. Como resultado de esto, las técnicas que eran efectivas en una fase pueden no serlo en otra.

Las actividades administrativas de recursos humanos, rara vez son responsabilidad directa del equipo de administración del proyecto.

### **2.4.3.6.2 Planeación organizacional.**

Involucra la identificación, documentación y asignación de los roles, responsabilidades y relaciones de reporte (jerarquía) del proyecto, que pueden ser asignadas a individuos o a grupos y que deben ser revisadas regularmente a lo largo del mismo y que seguidamente están fuertemente ligadas a la planeación de las comunicaciones.

### **2.4.3.6.3 Contratación de personal.**

Involucra obtener el personal necesarios, asignados y trabajando en el proyecto.

### **2.4.3.6.4 Desarrollo de Equipo.**

Incluye tanto el mejoramiento de las habilidades de los involucrados para contribuir como individuos, así como el mejoramiento de las habilidades del equipo para funcionar como equipo.

## **2.4.3.7 Comunicaciones.**

### **2.4.3.7.1.1 Planeación de comunicaciones.**

Involucra la determinación de que información y comunicaciones son necesarias para los involucrados en el proyecto: quién necesita, qué información, cuándo la necesitarán y cómo les será dada.



#### **2.4.3.7.2 Distribución de la información.**

Involucra hacer que la información necesaria este disponible para los involucrados en el proyecto de una manera oportuna.

#### **2.4.3.7.3 Reporte de desempeño.**

Involucra coleccionar y diseminar la información de desempeño de manera que se provea a los involucrados en el proyecto con la información acerca de cómo están siendo utilizados los recursos para alcanzar los objetivos del proyecto. Este proceso incluye: reporte del estatus, reporte del progreso y predicciones.

#### **2.4.3.7.4 Cierre administrativo**

El proyecto o la fase, después de haber alcanzado sus objetivos o de haber sido terminada por otras razones, requiere un cierre. El cierre administrativo consiste en verificar y documentar los resultados del proyecto para formalizar la aceptación del producto o del proyecto por el patrocinador o cliente.

### **2.4.3.8 Riesgos.**

#### **2.4.3.8.1 Identificación de riesgo.**

Consiste en determinar que riesgos son los más propensos a afectar el proyecto, y en determinar cuales son las características de cada uno, no es un evento único, sino algo que debe ser ejecutado regularmente a lo largo del proyecto.

Se deben definir riesgos internos (los que el equipo puede controlar o sobre los que puede tener influencia) y los riesgos externos (fuera del control y la influencia del equipo).

#### **2.4.3.8.2 Cuantificación de riesgo.**

Involucra la evaluación de riesgos, y las interacciones entre ellos para evaluar el rango posible de resultados del proyecto.

### **2.4.3.8.3 Desarrollo de la respuesta al riesgo**

Involucra la definición de los pasos de mejora para las oportunidades y la respuesta a las amenazas, la respuesta a las amenazas generalmente cae dentro de tres categorías: evitar, mitigar y aceptar.

### **2.4.3.8.4 Control de la respuesta al riesgo**

Involucra la ejecución del plan para la administración del riesgo de manera que se responda a los eventos de riesgo durante el curso del proyecto.

- 

## **2.4.3.9 Procuración.**

### **2.4.3.9.1 Planeación de la procuración.**

Es el proceso de identificar cuales necesidades del proyecto pueden cubrirse mejor con la procuración de bienes y servicios afuera del proyecto de la organización. Involucra la consideración de sí, cómo, que, cuánto y cuando procurar.

### **2.4.3.9.2 Planeación de la solicitud.**

Involucra la preparación de los documentos necesarios para soportar la solicitud.

- 

### **2.4.3.9.3 Solicitud**

Involucra el obtener información (Ofertas y cotizaciones) de probables vendedores y cómo se pueden alcanzar las necesidades del proyecto.

### **2.4.3.9.4 Selección de fuentes**

Involucra la recepción de ofertas o propuestas y la aplicación de los criterios de evaluación para seleccionar a un proveedor.

## **2.5 El Sistema de Información del Iteso.**

### **2.5.1 El Iteso.**

#### **2.5.1.1 Datos generales.**

El nombre oficial es Instituto Tecnológico y de Estudios Superiores de Occidente (2003). Se localiza al sur de la zona metropolitana de Guadalajara en el Municipio de Tlaquepaque, Jal., su dirección es Periférico Sur Manuel Gómez Morín No. 8585. Se ubica en una superficie de 40.59 Ha., tiene una superficie construida de 4.116 Ha.(Acuña 2003) Para más datos acerca la ubicación de la Universidad se puede consultar la página:

<http://www.iteso.mx/campus/mapa/planta/ubicacion.htm>

Una versión resumida de la misión de la universidad es: “La búsqueda de la verdad para formar universitarios, hombres y mujeres libres, críticos, responsables, competentes y decididos a poner su ser y su profesión para transformar la sociedad mexicana en una más justa y más humana. Para consultar la misión y las orientaciones fundamentales del Iteso, se tiene la siguiente dirección de internet:

<http://www.iteso.mx/inst/cat/index.htm>

En el periodo escolar de primavera 2003 cuenta con: 22 Licenciaturas, 11 Maestrías y 2 Doctorados, además ofrece Diplomados y Cursos. Para mayor información consultar la página:

<http://www.iteso.mx/acad/depacad.htm>

En el mismo periodo escolar tiene 8,302 alumnos inscritos, atendidos por 970 profesores, que dan clases a 2344 grupos (SSE 2003).

En el Iteso laboran 604 personas de apoyo (administrativo y de servicios).

#### **2.5.1.2 Reseña histórica.**

En 1957 un grupo de profesionales y empresarios tapatíos constituyó el Instituto Tecnológico y de Estudios Superiores de Occidente, Asociación Civil (ITESO, AC), "con el fin primordial de promover y desarrollar en la región la educación superior de inspiración cristiana

Para cumplir con este fin, ITESO, AC fundó, en agosto de 1957, el Instituto Tecnológico y de Estudios Superiores de Occidente (ITESO) y solicitó la cooperación de la Compañía de Jesús. Luego

de algunos años se constituyó el Centro de Cultura Superior, Asociación Civil (Cecus, AC), compuesto por profesores e investigadores miembros de la Compañía de Jesús que laboran en el ITESO, que se formalizó el 13 de junio de 1967.

El ITESO es una universidad privada, propiedad de ITESO, AC, cuyos estudios de educación superior gozan de plena validez oficial en la República Mexicana, de conformidad con el acuerdo de reconocimiento de la Secretaría de Educación Pública número 15,018 y la legislación federal vigente en la materia.

Para más datos acerca de la historia del Iteso consultar “Acontecer histórico del Iteso” de Alberto Gutiérrez Formoso (Gutierrez 1997; Gutierrez 1998; Gutierrez 2000). Sobre el pasado y presente del Iteso consultar la página:

<http://www.iteso.mx/inst/cat/crono.htm>

### 2.5.1.3 Organigrama.

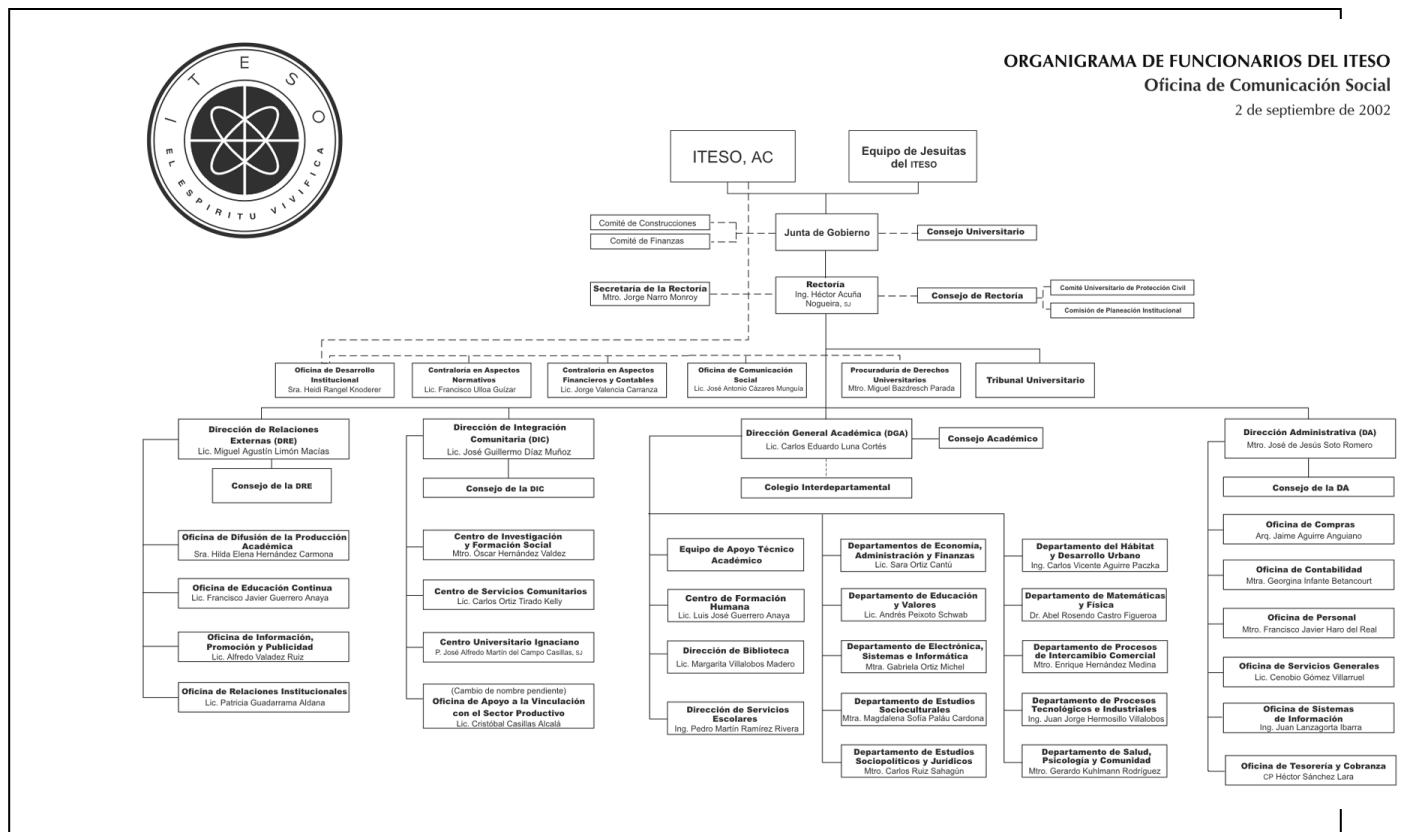


Fig 2.5.1.3 Organigrama del ITESO

## **2.5.2 La Oficina de Servicios de Información del Iteso.**

### **2.5.2.1 Datos generales.**

La Oficina de Servicios de Información (OSI) tiene como labor sustantiva promover la cultura computacional basada en la informática haciendo accesible el uso de las computadoras a todos los miembros de la comunidad universitaria, porque el cómputo es parte de la cultura, auxiliar en la educación, medio de comunicación y factor estratégico para la universidad (Narro 2000).

La oficina se encuentra situada en el Edificio “G” del campus.

<http://www.iteso.mx/campus/mapa/planta/index.htm>

En el 2002 el Iteso contaba con 1,942 computadoras; de ellas, 1,020 estaban asignadas a empleados o dependencias, 856 a salas y laboratorios, y 66 tenían funciones de servidor (Acuña 2003).

### **2.5.2.2 Reseña histórica.**

Sus antecedentes se extrajeron del reporte de la “Historia de Cómputo del Iteso” (Morales 1997), excepto cuando se indique otra referencia.

En Abril de 1977 se inaugura el Centro de Computo Concepción Carroll de Urrea. En agosto de 1981 se planea desarrollar un sistema de registros escolares y nómina. En Julio 1982 se cuenta con una microcomputadora en la División de Ingeniería para el Sistema Escolar, se utiliza para el control de alumnos hasta registro de servicio social. En noviembre de 1983 llegan terminales para instalarse en las Direcciones y Coordinaciones de las Divisiones Académicas. En febrero de 1987 se firma un convenio con IBM, se aportará equipo de cómputo, mantenimiento de programas y recursos humanos, el Iteso aportará instrucción al personal, proyectos de investigación y servicio editorial para difundir resultados. En marzo de 1989 surge un preproyecto de la red Administrativa del Iteso, el cual contempla las siguientes oficinas: Contabilidad, Tesorería, Créditos Educativos, Servicios Generales, Servicios Escolares y Recursos Humanos.

Con la Reforma Organizativa del Iteso 1995 se crea la Oficina de Servicios Computacionales (OSC), cuyas labores sustantivas eran: hacer posible que toda la comunidad universitaria pueda hacer uso del cómputo; y mejorar continuamente los servicios computacionales del Iteso (Vázquez 1995). La

OSC depende directamente de la Dirección Administrativa. En el plan trianual 1996-1999 la OSC se divide en: Organización y Métodos, Atención a usuarios, Redes, Telecomunicaciones y Software.

En 1997 se inicia el cambio de la plataforma operativa DOS con el Sistema de Información Iteso en Clipper 87, conformado por 78 módulos conteniendo un total aproximado de 1,150 programas que funcionan diariamente, a la nueva plataforma de computo en Linux, utilizando Oracle como motor de base de datos, Linux en los servidores de intermedios de Web, para hacer transacciones vía Internet y Delphi como herramienta de desarrollo para ambiente Windows.

Algunos ejemplos de módulos son: el escolar, tesorería, nóminas, evaluación del desempeño docente, etc., estos módulos a su vez están compuestos de varios programas.

En el plan trienal 2000 – 2002 cambia a su nombre actual: Oficina de Servicios de Información (OSI) (Narro 2000).

### 2.5.2.3 Organigrama.

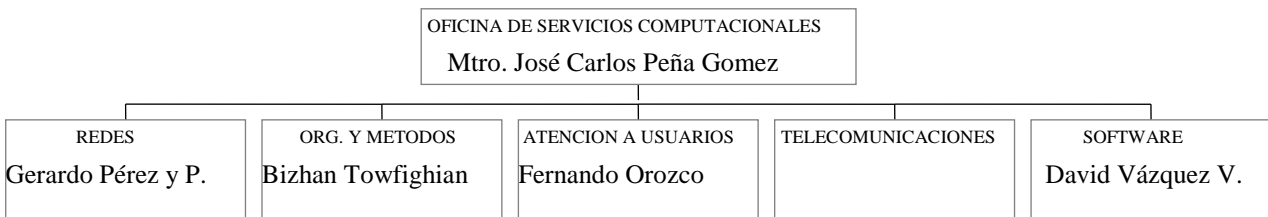


Fig 2.5.2.3 Organigrama de la OSI

## 2.5.3 El Sistema de Información del Iteso.

### 2.5.3.1 Conceptos básicos.

Entendemos como un Sistema de Información del Iteso (SII) el conjunto de personas, recursos, métodos de trabajo, procesos e información que interactúan entre sí para ayudar a la toma de decisiones en función de los objetivos institucionales.

### 2.5.3.2 Áreas del Sistema de Información del Iteso.

El SII actualmente se encuentra dividido en 4 grandes áreas:

- a) Proyectos.
- b) Financiero.

- c) Escolar.
- d) Administrativo.

Las áreas se dividen en sistemas y éstos a su vez en módulos. La EDD es un sistema del área Escolar, los reportes históricos de la evaluación de los alumnos de licenciatura a los profesores son un módulo del anterior.

## **2.6 El módulo de reportes históricos del desempeño docente del Iteso.**

### **2.6.1 La evaluación del desempeño académico.**

#### **2.6.1.1 Conceptos básicos.**

##### **2.6.1.1.1 Evaluar.**

Señalar el valor de una cosa. Estimar, apreciar, calcular el valor de una cosa. Estimar los conocimientos, aptitudes y rendimiento de los alumnos (RAE 1999).

##### **2.6.1.1.2 Evaluación del desempeño.**

La evaluación del desempeño es el proceso de evaluación sistemática de las cualidades laborales, necesidades de desarrollo y avances en el cumplimiento de metas de cada empleado y de la posterior determinación de medios para la evaluación del desempeño. Las metas deben ser claras y específicas; y deben poseer un alto grado de dificultad, a pesar de lo cual deben hallarse al alcance de los individuos (Hellriegel 1998).

##### **2.6.1.1.3 Uso de las evaluaciones del desempeño.**

Hellriegel (Hellriegel 1998) define cuatro usos:

- a) Decisiones de retribuciones.
- b) Desplazamiento de personal.
- c) Retroalimentación sobre el desempeño
- d) Necesidades de capacitación.

##### **2.6.1.1.4 Uso de las evaluaciones del desempeño docente.**

Marsh citado por Rueda (Rueda 2000) resume los propósitos de los cuestionarios de evaluación, de la siguiente manera:

- a) Diagnóstico y retroalimentación a los profesores sobre su desempeño o efectividad docente.
- b) Medida de efectividad docente para ser empleada como información para las decisiones sobre el otorgamiento de la definitividad a los profesores y su promoción.
- c) Información para que los alumnos puedan seleccionar cursos e instructores.
- d) Investigación sobre los resultados y procesos docentes.

#### **2.6.1.1.5 Problemas de las evaluaciones del desempeño.**

Hellriegel (Hellriegel 1998) menciona cuatro problemas:

- a) Características del evaluador.
- b) Indulgencia, ocurre cuando un individuo evalúa por encima del nivel merecido.
- c) Efecto aureola, ocurre cuando la información del evaluador acerca del desempeño en una dimensión determina la evaluación en todas las demás.
- d) Tendencia central, es un error de evaluación que ocurre cuando se asignan calificaciones promedio, aún cuando el desempeño varíe.

#### **2.6.1.1.6 Metodologías de evaluación del desempeño.**

##### **2.6.1.1.6.1 Clasificación.**

Consiste en evaluar el desempeño mediante la comparación de empleados que realizan un trabajo igual o similar.

##### **2.6.1.1.6.2 Escala Gráfica.**

Consiste en evaluar a los empleados en relación con una serie de medidas de cumplimientos, por lo general en una escala de cinco o siete puntos.

##### **2.6.1.1.6.3 Clasificación del comportamiento.**

Describe conductas laborales específicas, a lo largo de un continuum ascendente, con la ventaja de que es posible describir objetiva y detalladamente las conductas relativas al desempeño.

##### **2.6.1.1.6.4 Evaluación de 360°.**

Es el proceso para obtener información sobre un individuo de una gran diversidad de fuentes. Con esta información se retroalimenta al individuo para ayudarlo a promover su desarrollo y mejoramiento continuos. La información puede provenir de: la persona que está siendo evaluada (autoevaluación), el gerente a quién dicha persona reporta, otros gerentes (especialmente en los



sistemas matriciales), subordinados directos en su propio equipo (si lo tienen), colegas y pares, clientes, proveedores (France 1997).

Una de las formas de autoevaluación que les permite a los docentes registrar y mejorar su trabajo son los Portafolios (Martin-Kniep 1999), que son colecciones de trabajos especializados y orientados hacia un objetivo, que captan un proceso imposible de apreciar plenamente. Los portafolios profesionales han sido aprobados por la Junta Nacional de Estándares para la Enseñanza Profesional, por el estado de Texas y por numerosos distritos escolares en EUA, con el fin de reconocer y certificar la idoneidad docente.

### **2.6.1.2 Paradigmas didácticos y evaluación docente.**

La actividad docente en su sentido más amplio: la impartición de clases, la asesoría a estudiantes, la tutoría, la dirección de proyectos terminales o tesis, la continua revisión o actualización de planes y programas de estudio, la organización de seminarios, etc. conforman el área más importante del quehacer universitario. El profesor es la figura central en el proceso de enseñanza – aprendizaje, los principales enfoques empleados para el estudio de este rol son (ANUIES 2000):

- a) Presagio – Producto, considera que la eficacia de la enseñanza es efecto directo de las características físicas y psicológicas de la personalidad del profesor.
- b) Conductual, análisis conducente a identificar los rasgos que un profesor debiera tener y está basado en el concepto de “competencias” docentes (explicadas más adelante). No resuelve el problema básico de la elección, por parte del profesor, de las competencias y comportamientos adecuados a las situaciones tan diversas que plantea la práctica cotidiana de la docencia.
- c) Proceso – Producto, en este paradigma el método es el elemento central en el proceso de aprendizaje. Se basa en la identificación de patrones estables de comportamiento en el establecimiento de correlaciones entre éstos y el rendimiento académico.
- d) Mediacional, es el paradigma de la enseñanza y el aprendizaje. Reconoce y utiliza dimensiones adicionales como la formación previa y las experiencias personales del profesor y del alumno, el contexto escolar y social y el contexto del aula, como esenciales para determinar el comportamiento de los actores del proceso educativo.
- e) Ecológico, visualiza todas las acciones recíprocas del proceso enseñanza – aprendizaje. Considera el aula es un complejo sistema de relaciones e intercambios. Pondera la eficacia del profesor respecto a sus alumnos, en el contexto específico.

El término “competencias” no tiene una definición consensuada (Reynaga 1999), Grootings (citado por Reynaga) dice que son los conocimientos, destrezas y aptitudes necesarios para ejercer una profesión, y que la persona competente es la que puede resolver los problemas de forma autónoma, flexible y que está capacitado para colaborar en su entorno profesional y en la organización del trabajo. Reynaga hace énfasis en la distinción entre competencias y cualificaciones, las primeras implican posibilitar el crecimiento del sujeto en todas sus dimensiones; mientras que las segundas abarcarían únicamente la dimensión laboral.

### **2.6.1.3 Indicadores más usados en la evaluación de la práctica docente.**

Estos indicadores son susceptibles de ser evaluados a través de encuestas a los alumnos, que serían el caso de la encuesta A3 de la evaluación del desempeño docente del Iteso. Fueron propuestos por García Ramos citado en ANUIES (ANUIES 2000).

- a) Competencia docente:
  - a. Dominio de los contenidos de la materia que enseña.
  - b. Buena preparación previa de las clases o sesiones.
  - c. Orden en la exposición de los temas.
  - d. Seguimiento de una secuencia lógica en el orden del temario.
  - e. Vinculación de los nuevos conocimientos con lo visto anteriormente.
  - f. Elaboración de síntesis o resúmenes de lo revisado y de lo que se va a explicar.
  - g. Verificación del nivel de comprensión alcanzado por los alumnos al término de las sesiones.
  - h. Claridad expositiva.
  - i. Uso de medios variados de apoyo al aprendizaje.
- b) Atención y dedicación hacia el alumno:
  - a. Cumplimiento del horario de clase y de tutorías establecido.
  - b. Motivación para asistir a tutorías y resolver dudas.
  - c. Respeto a los juicios y opiniones de los demás.
  - d. Dedicación del tiempo que sea necesario fuera de clase.
  - e. Atención individual a los alumnos que la solicitan.
  - f. Trato respetuoso a los estudiantes.
- c) Grado de participación del alumno en clase, fomentado o aceptado por el profesor:
  - a. Participación de los alumnos en la elaboración y exposición de los temas.
  - b. Motivación a los alumnos para preguntar y participar en clase.
  - c. Impulso al trabajo en grupo.

- d. Fomento del diálogo, reflexión y debate sobre los temas tratados.
- d) Calidad de la evaluación realizada por el profesor:
  - a. Conocimiento de los criterios de evaluación por parte de los alumnos.
  - b. Equidad en las evaluaciones.
  - c. Uso de diferentes instrumentos de evaluación según los objetivos a evaluar.
  - d. Oportunidad en la entrega de resultados de las evaluaciones.
  - e. Retroalimentación a los alumnos sobre los problemas detectados en la evaluación.
- e) Planificación – Programación del profesor:
  - a. Entrega oportuna del programa de la materia, del cronograma, de los criterios de evaluación y del significado de la calificación.
  - b. Distribución del tiempo y de las actividades para cumplir todos los objetivos del curso o unidad de enseñanza aprendizaje.
- f) Logro de los objetivos y metas de las unidades de enseñanza aprendizaje:
  - a. Percepción de los estudiantes respecto al logro de los objetivos.
  - b. Percepción sobre la importancia de lo aprendido en la unidad de enseñanza aprendizaje.
- g) Asiduidad y Puntualidad:
  - a. Porcentaje de asistencia del profesor.
  - b. Porcentaje de sesiones iniciadas puntualmente.
  - c. Porcentaje de sesiones terminadas puntualmente.

Existen otros estudios como el de Feldman citado por Rueda (Rueda 2000) que identificó 28 dimensiones relacionadas con la efectividad docente, que pueden complementar las anteriores.

#### **2.6.1.4 Confiabilidad de las encuestas de evaluación del desempeño docente.**

La confiabilidad es la consistencia de los cuestionarios de evaluación a través del tiempo y su correlación con sus equivalentes, en el caso de la evaluación del desempeño docente, generalmente se determina a partir del análisis de las correlaciones del grupo de elementos que conforman un instrumento y del nivel de acuerdo entre los alumnos de un mismo grupo. Se ha demostrado (ANUIES 2000; Rueda 2000) que la consistencia depende del número de alumnos que la respondan, de aquí que las aplicaciones en grupos menores de 10 alumnos pueden ocasionar problemas de confiabilidad, tampoco es fiable ni justo comparar evaluaciones de grupos grandes con las de grupos pequeños. La autopercepción de los profesores sobre su desempeño dista de la de los jueces externos, en especial de la de sus alumnos. La estabilidad de las puntuaciones no suele ser alta cuando se analiza el desempeño

del profesor en los diferentes cursos que imparte o en un mismo curso impartido por varios profesores. De aquí que no sea conveniente juzgar el desempeño de un profesor por los resultados solamente de un curso o de un periodo académico en el que imparte cursos diferentes, sino que se recomienda evaluar el desempeño de un profesor a lo largo de varios periodos académicos, basándose en el gran promedio de todos sus cursos como el indicador de su desempeño, ya que la estabilidad de estas puntuaciones tiene una alta correlación.

### **2.6.1.5 Validez de las encuestas de evaluación del desempeño docente.**

La validez se refiere (Rueda 2000) a que las evaluaciones del desempeño docente midan realmente lo que pretenden medir (validez de contenido) y sean a su vez capaces de predecir la efectividad docente del profesor y el rendimiento académico de los alumnos (validez predictiva). La validez también se puede analizar desde otras dos perspectivas: la validez del proceso instruccional y la validez de los resultados de la instrucción.

En cuanto al proceso de instrucción, se considera al alumno como un consumidor que debe manifestar su satisfacción con el servicio que el profesor brinda. Por lo tanto se considera al alumno como un ente capaz de evaluar el servicio que recibe, adoptándose una postura más democrática dentro del proceso de enseñanza – aprendizaje. En esta perspectiva los estudios se han topado con el problema del efecto “Doctor Fox” o de “Seducción Académica”, en el que el nivel de expresividad o de seducción del profesor tiene un alto impacto en las evaluaciones del profesor pero no en su logro académico, y el contenido del curso tiene un alto impacto en el logro académico de los alumnos pero poco impacto en sus evaluaciones.

Sobre la validez de los cuestionarios de evaluación en relación con los productos de la instrucción, estudios recientes han demostrado que los alumnos son jueces competentes de qué tanto aprendieron en el curso, aunque no saben si lo que están aprendiendo es apropiado o no para cubrir los objetivos del curso o del currículum.

En el aspecto de la validez también se debe considerar que según el “Paradigma de Aplicabilidad” estas evaluaciones son aplicables en instituciones educativas de otros países o culturas.

Otro factor a considerar es la “Multidimensionalidad” de las encuestas, que puede lograrse mediante tres estrategias:

- a) Deducir los ítems de un análisis lógico de los contenidos de la efectividad docente con base en la retroalimentación de profesores y alumnos.

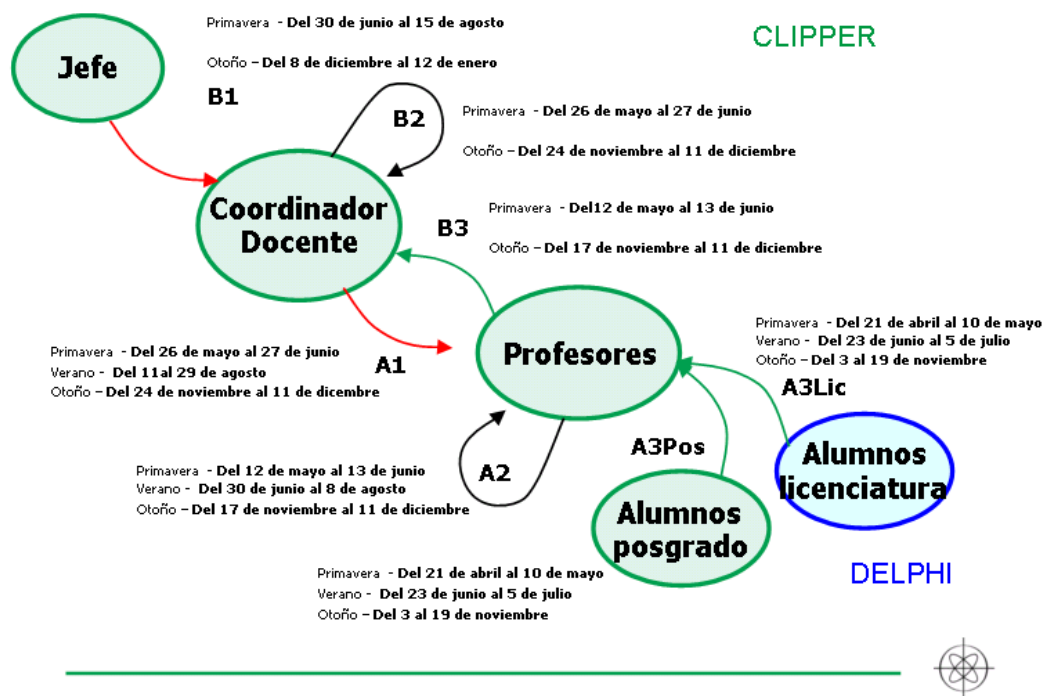
- b) Deducir los factores e ítems de determinada teoría de la enseñanza o del aprendizaje.
- c) Realizar un análisis factorial con el propósito de validar los factores del instrumento.

## 2.6.2 La evaluación del desempeño académico en el Iteso.

### 2.6.2.1 Datos generales.

#### 2.6.2.1.1 Esquema General de la Evaluación del Desempeño Académico.

Fig 2.6.2.1.1 Esquema General de la evaluación del desempeño académico



#### 2.6.2.1.2 Objetivos de la Evaluación del Desempeño Docente

Los tres objetivos de la evaluación del desempeño docente son (2002):

- a) Mejorar el desempeño de los profesores mediante la retroalimentación oportuna y sistemática de los resultados y/o productos de su trabajo.
- b) Identificar oportunidades de desarrollo académico de los profesores, tanto en lo individual como en lo grupal.

- c) Generar información para sustentar las decisiones institucionales en materia de promociones y distinciones académicas, aplicación del factor de desempeño en el tabulador, reubicaciones y bajas, de acuerdo con las normas aplicables en cada caso.

### **2.6.2.1.3 Criterios de la Evaluación del Desempeño Docente.**

El 20 de febrero de 2002 el Consejo Académico aprobó los siguientes criterios de la evaluación del desempeño docente (2002):

- a) El referente de la evaluación del desempeño es el conjunto de actividades académicas definidas como tales en el Reglamento del Personal Académico (RPA), y concretadas en el encargo de los profesores según su tipo de relación con el ITESO: de planta, de asignatura, temporales y visitantes.
- b) La evaluación del desempeño de los profesores de planta es anual e integra el conjunto de actividades definidas en su encargo de trabajo. La evaluación del desempeño de los profesores de asignatura tiene como objeto las actividades de enseñanza previstas para esta categoría del personal, y es por el término del periodo escolar para el que son contratados.
- c) El desempeño de los profesores temporales y visitantes se realiza con referencia a las actividades y plazos especificados en el contrato o acuerdo de trabajo específico.
- d) La evaluación del desempeño incluye tanto el proceso como los resultados y/o productos definidos en el encargo, conforme sus especificaciones e indicadores de calidad. Se basa en los insumos de información que presentan el propio profesor (autoevaluación), sus pares en el contexto grupal de actividad, los usuarios o beneficiarios del trabajo y el jefe de departamento o centro académico, o jefe inmediato en el caso de los profesores de asignatura.
- e) La evaluación es sancionada por el jefe de departamento o centro académico en diálogo con el profesor, y queda registrada por escrito como elemento de contexto para las decisiones consecuentes. En el caso de los profesores de asignatura, el jefe cuenta con los apoyos necesarios dentro de la organización interna del trabajo de la dependencia académica.
- f) Cada una de las actividades académicas cuenta, según su índole, con su propia instrumentación. En todos los casos la instrumentación debe incluir: los componentes de la actividad que son considerados en la evaluación; las descripciones o rúbricas de los comportamientos y resultados esperados para distintos niveles de desempeño; las evidencias del desempeño (programas de curso, materiales elaborados, productos de aprendizaje de los estudiantes, informes, publicaciones, etc.); las guías, formatos e instrumentos para obtener la información de las diferentes fuentes (el profesor, los pares...), así como para el registro final de la evaluación; los tiempos del proceso; los parámetros y procedimientos numéricos que permitan la expresión cuantitativa de la evaluación.

- g) La instrumentación de la evaluación en lo que se refiere a sus componentes, descripciones de desempeño y evidencias es coherente con los marcos institucionales relacionados con las actividades académicas (Criterios y procedimientos para las promociones y distinciones académicas, Marco para el desarrollo institucional de la investigación, políticas editoriales, etc.)
- h) La instrumentación de la evaluación es tan sencilla como sea posible, de tal manera que propicia la participación de todos los actores involucrados y facilita el procesamiento oportuno y ágil de la información y su registro.
- i) El profesor es responsable de obtener y registrar las evidencias ordinarias de su desempeño, así como de presentar los informes y datos necesarios en el proceso de evaluación.
- j) Para el caso de la evaluación docente y de la coordinación docente, la información proveniente de los usuarios (estudiantes y profesores, respectivamente), es recogida y procesada con base en instrumentos estandarizados y únicos.
- k) Los profesores están formalmente enterados de los objetivos, criterios e instrumentación puntual de la evaluación, y tienen acceso a sus resultados documentados y sancionados.

Los criterios anteriores se basan en los criterios y disposiciones que aparecen en el Marco de la Docencia Iteso: Disposiciones Generales (2001), y en las actuaciones básicas del profesor en la preparación, conducción, seguimiento y evaluación de los procesos de enseñanza-aprendizaje, que se mencionan en el Marco de la Docencia en el Iteso: Elementos básicos del desempeño y de la Coordinación Docentes (2001).

#### **2.6.2.2 Reseña histórica.**

Las fechas más importantes de este proceso, fueron tomadas de Collignon (Collignon 1996).

En 1970 se registra el primer intento de evaluación a los académicos, cuando el maestro Alberto Hernández Medina, entonces director de la escuela de Relaciones Industriales del Iteso, diseñó un cuestionario para conocer algunos aspectos de su clase macroeconomía.

En 1974 el Dr. Jorge Mata Murillo realiza algunas modificaciones al cuestionario con la idea de utilizarlo en varias carreras y le incluye indicadores de control, asimismo organiza el perfil del docente en cuatro rubros: actitudes, conocimientos, comportamiento en el aula y cualidades humanas.

En 1975 el Consejo Académico aprueba el cuestionario y se comienza aplicar a todo el alumnado de todas las carreras de la universidad, hasta 1986, el cuestionario fue revisado continuamente, modificado en su forma y se aplicó sistemáticamente cada semestre escolar.

En 1986 el proceso se había burocratizado y faltaba una real participación de las divisiones académicas en el proceso, y la Secretaría Académica propone al Comité Académico apoyar la propuesta de descentralización del proceso de evaluación de profesores.

En 1996 la Dirección General Académica vuelve a centralizar la aplicación del instrumento que ahora es parte de un proceso de tres partes denominado de Evaluación del Desempeño Docente EDD, que se aplica y procesa a través del sistema de cómputo de la universidad.

Para 1999 se hicieron las siguientes propuestas (Alvarez 1997):

1.- Propuestas aspectos cualitativos:

a) Insistir en la participación voluntaria y honesta de los alumnos en el proceso de la evaluación pero, al mismo tiempo regulada por la institución. En el caso de los docentes, incluir los contratos que la evaluación es un requisito institucional. Para los encargados de área, incluir el instrumento como un elemento de apoyo para la renovación educativa.

b) Para el mejoramiento de la práctica docente, completar el proceso de evaluación de la práctica docente. Dirigir el sentido hacia la evaluación del proceso mismo de enseñanza - aprendizaje. Redefinir criterios básicos y compatibles a todo el cuerpo docente universitario que sean revisados, compartidos y aceptados por todos.

c) Las decisiones sobre el personal deben ser tomadas considerando el proceso completo de evaluación.

2.- Propuestas aspectos cuantitativos:

Contar con un instrumento confiable y válido.

En febrero de 2002 entre en vigor el nuevo sistema de evaluación del Desempeño Académico del Iteso (Alvarez 2002), con nuevos instrumentos de evaluación y nuevas formas de preguntar, en la que hay descriptores del comportamiento, con las que se espera que los evaluadores tengan más elementos de juicio. Aunque cambiaron las preguntas en las encuestas, tienen equivalencias que se utilizarán para comparar la información histórica. Aún no tenemos datos sobre la validez de la información de acuerdo a la participación al contestar las encuestas, sabemos que mientras más participación haya,



mayor será la representatividad de la muestra. No podemos afirmar que los que acuden a contestar las encuestas son los van a evaluar muy bajo o muy alto.

### **2.6.2.3 El módulo de reportes históricos de la evaluación A3.**

Los antecedentes de este módulo se pueden encontrar en los reportes de la Comisión Coordinadora de la Evaluación del Desempeño Docente (Alvarez 1997; Alvarez 1998; Alvarez 1998b) y en los reportes de periodos escolares siguientes, aunque ya no se han publicado. En estos reportes hay tablas y gráficos de: porcentajes de participación por Departamento y por periodo; promedios departamentales por área y subárea del instrumento de evaluación por Departamento y por periodo, las gráficas incluyen la media del Iteso; y el ordenamiento de los profesores en base al puntaje obtenido en el área 200 del instrumento. Posteriormente este ordenamiento se hizo en base al promedio de 4 ítems que se consideraron los más importantes.

### **2.6.2.4 Estado actual.**

El SEDA conjunta una serie de procesos en una relación sistémica con otros procesos educativos y administrativos de la universidad. Este sistema actualmente depende de la DSE, se viene aplicando desde el periodo escolar agosto – noviembre 1996. Desde 1998 soy el encargado de la organización y la logística de la aplicación, de la generación de reportes para las diferentes instancias de la universidad, así como de implementar los cambios que se le hagan.

Para dar soporte al SEDA se cuenta con un módulo en el Sistema de Información (SI) del ITESO, dependiente de la Oficina de Servicios de Información (OSI), desarrollado en una plataforma DOS en lenguaje Clíper, con capacidad de almacenamiento de la información de un año, que comprende tres ciclos escolares, pasado este lapso de tiempo, se hace un respaldo se cierra el sistema y se borran todos los datos para poder almacenar y capturar los del siguiente año.

En cada ciclo escolar se generan reportes que se hacen llegar a los profesores, se archivan en sus expedientes y se guardan en archivos DBF. Estos reportes se requieren para tomar decisiones de recontractación y la promoción de los profesores, así como para hacer análisis estadísticos sobre el desempeño académico del Iteso.

Es frecuente que estos reportes se extravíen y sea necesario regenerarlos, además la explotación de su potencial de información se reduce a reportes de porcentaje de participación y al análisis general de los datos numéricos de los profesores o de los departamentos del Iteso, como el ordenamiento de los profesores en base a los promedios obtenidos en algunos campos, el promedio de la apreciación estudiantil de los profesores del departamento, agrupación por grupos de la misma materia, etc.

El personal de la OSI puede recuperar los archivos de datos DBF generados en clíper, para regenerar en Access, los reportes ya mencionados, por el momento tenemos el 90% de los archivos respaldados, pero su consulta en Word y Excel es lenta y laboriosa.

En un principio los archivo de DBF comprendían solamente las evaluaciones de los alumnos de licenciatura a los profesores, paulatinamente se han hecho modificaciones a las encuesta de la evaluación de los alumnos y se han agregado más reportes como la evaluación de los alumnos de posgrado a los profesores, la Auto- evaluación de los profesores y la Evaluación de los coordinadores docentes.

La plataforma operativa del SI del Iteso se está emigrando de DOS a Linux, utilizando Oracle como motor de base de datos, Linux en los servidores de intermedios de Web, para hacer transacciones vía Internet y Delphi como herramienta de desarrollo para ambiente Windows.

En la nueva plataforma de cómputo toda la información de los reportes del SEDA deberá estar en línea y podrá ser consultada por cualquier persona con atribuciones para ello desde el propio profesor, los coordinadores docentes, jefes de departamento, etc. Estas personas podrán seleccionar, organizar y procesar los datos para generar la información específica que requieran y que no este contenida en los reportes estándares que se generan.

Experiencias anteriores con proyectos piloto en otros módulos del SI han demostrado que el proceso de cambio es complejo, han tomado más tiempo del planeado y han necesitado de retrabajo en correcciones y ajustes, el desarrollo de los módulos en la nueva plataforma operativa requiere establecer procesos estandarizados para lograr una transición más rápida y eficiente.

### **3 Proyecto de Ingeniería de Requerimientos.**

#### **3.1 El Plan de administración del proyecto de Ingeniería de Requerimientos.**

##### **3.1.1 Plan de integración.**

###### **3.1.1.1 Previos.**

###### **3.1.1.1.1 Formatos.**

El formato general de los documentos relacionados con el proyecto es tamaño carta o un múltiplo de este. Los formatos particulares de los procedimientos de la administración del proyecto se encuentran en el anexo “Procedimientos para los planes de la administración del proyecto”

###### **3.1.1.1.2 Diccionario.**

Las definiciones que se utilizan están descritos en los apartados “Conceptos básicos” de los capítulos anteriores de este trabajo. Para cualquier otro concepto no incluido se utilizará la definición que aparece en el estándar IEEE 610.12

##### **3.1.1.2 Desarrollo del plan del proyecto.**

El plan del proyecto de IR del módulo de reportes históricos del instrumento A3 de la Evaluación del Desempeño Docente Iteso se conformará con los siguientes planes:

- a) Plan de administración del alcance.
- b) Plan de administración del tiempo.
- c) Plan de administración del costo.
- d) Plan de administración de calidad.
- e) Plan de administración del personal.
- f) Plan de administración de comunicaciones.
- g) Plan de administración de riesgos.
- h) Plan de administración de la procuración.

### 3.1.1.3 Control general de cambios.

#### 3.1.1.3.1 Responsabilidades.

La instancia de la Universidad ITESO que solicitó el proyecto, nombrará a un encargado del mismo, quién junto con el líder del proyecto serán los único punto de contacto para la administración del control general de cambios. Cada uno será responsable del control de los cambios, uno ante su instancia y el otro ante la OSI. La solicitud de los cambios debe ser aprobada por ambas partes previamente a la investigación y a la implementación. El encargado y el líder del proyecto son responsables de la identificación y el análisis de impacto de los cambios.

#### 3.1.1.3.2 Proceso del control general de cambios.

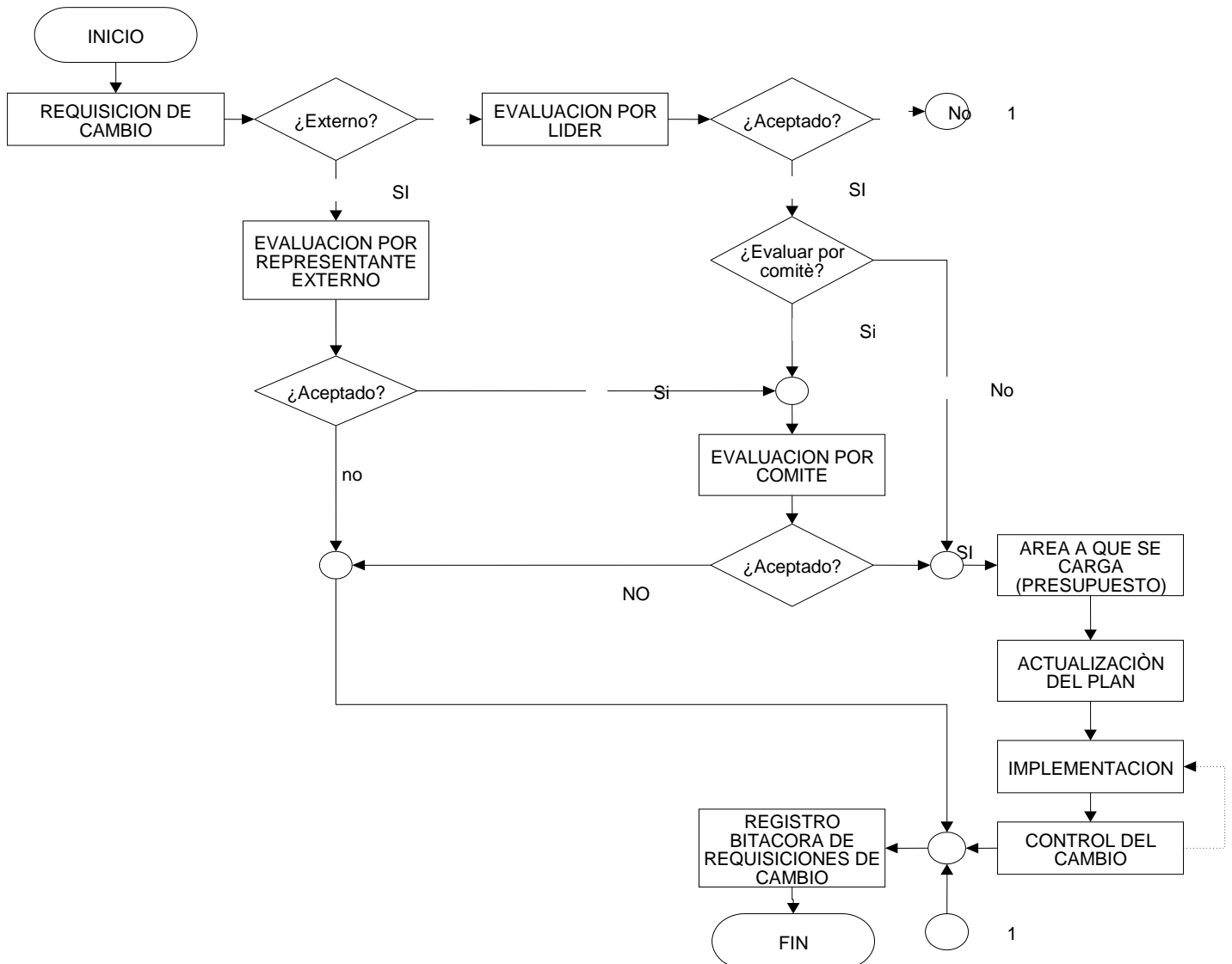


Fig 3.1.1.3.2 Proceso del control general de cambios

### **3.1.1.3.3 Formatos del control general de cambios.**

Los formatos se encuentran el anexo de los procedimientos de la administración del proyecto.

### **3.1.2 Plan de administración del alcance.**

#### **3.1.2.1 Justificación.**

“Consolidación: evaluar y transformar” fue el lema institucional del informe 2003 del rector Ing. Héctor Manuel Acuña Nogueira S.J. En su informe pide la mejora de nuestros procesos de evaluación, desde los más simples y cotidianos, como la aplicación de exámenes a los estudiantes, hasta los más sofisticados, como la revisión de las estrategias institucionales. Estos procesos de evaluación deben estar marcados por la participación, la honestidad, la claridad en los alcances, la transparencia, la congruencia y, sobre todo, la verdad, con el objetivo de dar a toda la institución y, más aún a todos los que la formamos, mayores posibilidades de desarrollo en las capacidades individuales y colectivas. Este objetivo es la transformación basada en la evaluación reflexionada sobre lo que se ha logrado y dejado de hacer.

El resultado de la evaluación y mejora continua de la calidad educativa de nuestra universidad nos permite además acreditarlos ante instituciones particulares de educación superior que han alcanzado altos estándares de calidad.

Los objetivos de la EDD, propuestos en el nuevo sistema, (ver apartado 2.6.2.1.2) atienden a una sistematización de los resultados para identificar oportunidades de desarrollo académico de los profesores y a la generación de información para sustentar decisiones institucionales.

La nueva plataforma de cómputo de la universidad trajo una mejora en los sistemas de información de la misma, abriendo nuevas posibilidades de manejo de la información que se deben aprovechar en la sistematización de resultados y la consiguiente generación de información.

En el módulo de reportes históricos de la evaluación de los alumnos de licenciatura a los profesores se deberán completar los reportes faltantes y conjuntar la información de tres encuestas diferentes que se han aplicado desde 1996 a la fecha, y volverlos más dinámicos, para que sean los usuarios de la EDD, los que estructuren la información de acuerdo a sus propios requerimientos y puedan establecer los análisis, las comparaciones, los promedios, las correlaciones, y las inferencias necesarias para su trabajo y que además ayuden a transformar nuestros procesos de evaluación para darles validez y hacerlos más confiables.

### **3.1.2.2 Metodología.**

Como esta pedido por el rector la metodología será participativa de los usuarios en el desarrollo de requerimientos, con un ciclo de vida incremental del proyecto del Sistema de la EDD, en donde una etapa de desarrollo es el módulo de reportes históricos. El módulo en particular tiene un desarrollo evolutivo entendido en dos sentidos extensión y versión. En cuanto a extensión se hará primero a nivel departamental y luego a nivel de la universidad. En cuanto a versión se desarrollarán primero los requerimientos considerados como prioritarios.

Por metodología participativa se entiende la involucración activa de los usuarios en el proceso de desarrollo, interactuando en equipos multidisciplinarios.

El lenguaje de modelado del sistema será UML, utilizaremos el CASE System Architect como herramienta de apoyo, para documentar los modelos, especificaciones y requerimientos del proyecto.

La metodología de IR descrita en el marco de referencia y la administración del proyecto siguiendo la propuesta adaptada del PMBOK que también aparece en el marco de referencia.

Para levantar los requerimientos se utilizarán las siguientes técnicas:

- a) Entrevistas.
- b) Cuestionarios.
- c) Tormenta de ideas.
- d) Discusiones de grupo.
- e) Prototipos en papel y de software.

### **3.1.2.3 Alcance.**

El alcance es el desarrollo de los requerimientos del módulo de reportes históricos de la evaluación de los alumnos de licenciatura a los profesores, del Sistema de la EDD, para todos los Departamentos y Centros dependientes de la Dirección General Académica.

### **3.1.2.4 Productos a entregar.**

Los productos a entregar incluyen:

- Especificaciones de requerimientos de sistema.
- Diseño de la arquitectura de sistema.
- Especificaciones de requerimientos de software.
- Modelado de software.

- Diseño de pantallas.
- Diagramas de flujo.
- Diagramas de objetos.
- Diagramas E/R.

### **3.1.2.5 Estructura desglosada de trabajos.**

Se presenta en el plan para la administración del tiempo. Ver punto 3.1.3

### **3.1.2.6 Criterios de éxito.**

Considerando que los procesos de evaluación institucional forman parte de la cultura del Iteso, el proyecto se considerará exitoso si contribuye a:

- a) Crear una cultura de evaluación académica medida por la participación de los alumnos y de los académicos.
- b) La calidad y el nivel de la enseñanza de la institución se ve paulatinamente impactado por los resultados emanados del proceso de la EDD. Medidos por la satisfacción de los alumnos con el profesorado y por las competencias adquiridas por los alumnos a su paso por esta universidad.
- c) Se consolida un modelo de evaluación sistémico, que complementa los procesos institucionales.

Para la EDD se considerarán los siguientes criterios de éxito:

- a) Se cuenta con información histórica válida y confiable de las evaluaciones de los alumnos de licenciatura a los profesores.
- b) Se hace investigación sobre los resultados de la evaluación y los procesos docentes que permiten establecer indicadores claros aceptados institucionalmente para comparación de resultados de la evaluación de los profesores.
- c) Se toman decisiones de permanencia, mejora y promoción de los docentes en base a varios insumos uno de los cuales son los reportes históricos de la EDD.
- d) Se definen correlaciones que permitan elaborar juicios más estructurados sobre los resultados de las evaluaciones.

Para la OSI se tienen los siguientes criterios de éxito:

- a) El proyecto se realizó de acuerdo a un plan establecido.

- b) Se documentó este proceso.
- c) Se analizaron los resultados del proceso seguido y se tomaron medidas y realizaron correcciones para aplicar en los siguientes procesos.

#### **3.1.2.7 Factores de éxito.**

Los siguientes factores serán clave para el éxito del proyecto:

- a) La institucionalización del proceso de evaluación.
- b) La integración de los departamentos y centros del Iteso, como un equipo en el proceso de la EDD.
- c) La comunicación activa entre los involucrados en el proyecto.

#### **3.1.2.8 Verificación del alcance.**

Una vez terminada la declaración del alcance deberá ser presentada por el líder del proyecto, a la instancia de la universidad que generó la solicitud del producto, para determinar si está de acuerdo a sus expectativas. Una vez aceptado el documento, deberá firmarse por el encargado y el líder del proyecto para pasar a la siguiente fase de desarrollo de requerimientos.

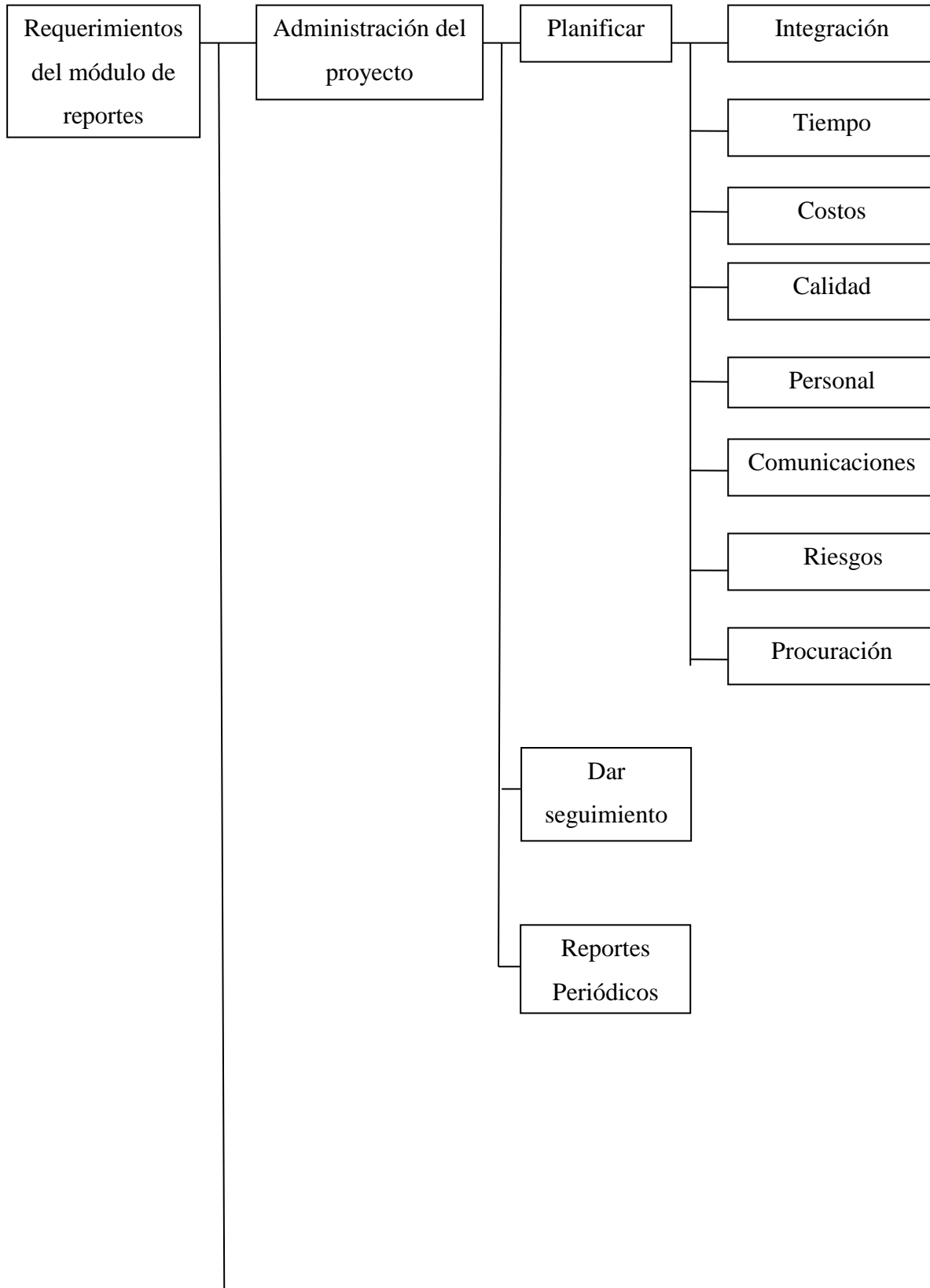
#### **3.1.2.9 Control de cambios del alcance.**

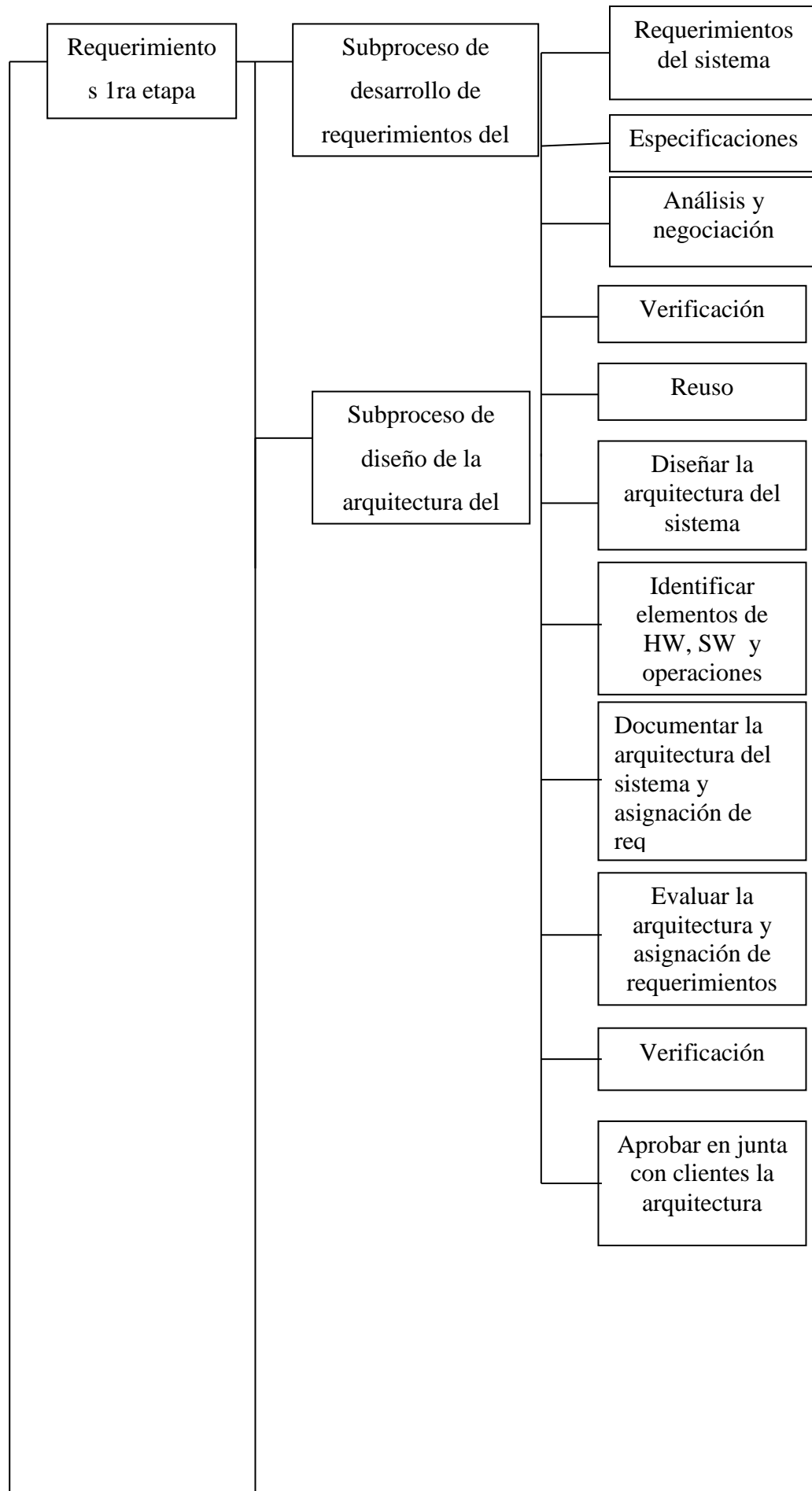
El control de cambios de alcance deberá seguir el procedimiento descrito en el punto 3.1.1.3. “Control general de cambios”.



### 3.1.3 Plan de administración del tiempo.

#### 3.1.3.1 Diagrama de Red.





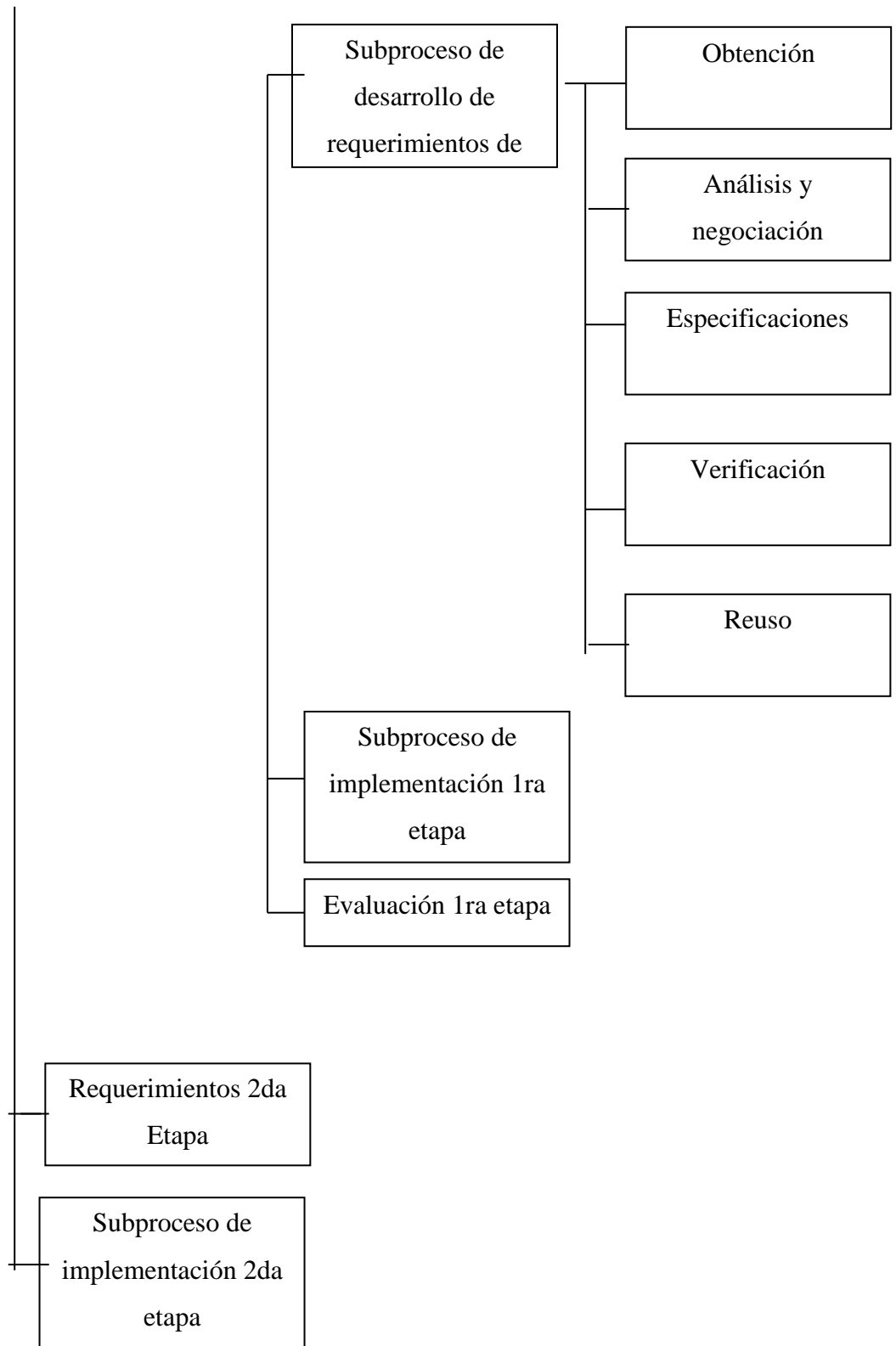


Fig 3.1.3.1 Diagrama de Red

### 3.1.3.2 Diagrama de Gantt.

Id	Nombre de tarea	Duración	1er semestre		2º semestre		1er semestre		2º semestre		1er semestre		2º semestre	
			tri 1	tri 2	tri 3	tri 4	tri 1	tri 2	tri 3	tri 4	tri 1	tri 2	tri 3	
1	<b>Requerimientos del módulo de reportes históricos ASLic.</b>	<b>588 días?</b>												
2	<b>Administración del proyecto</b>	<b>588 días</b>												
3	<b>Planificar</b>	<b>13 días</b>												
4	Integración	4 días												
5	Tiempo	4 días												
6	Costos	9 días												
7	Calidad	5 días												
8	Personal	5 días												
9	Comunicaciones	6 días												
10	Riesgos	3 días												
11	Procuración	3 días												
12	Dar seguimiento	575 días												
13	Reportes periódicos	44 días												
14	<b>Requerimientos 1ra. Etapa</b>	<b>90 días</b>												
15	<b>Subproceso de desarrollo de requerimientos de sistema</b>	<b>90 días</b>												
16	<b>Requerimientos de sistema</b>	<b>34 días</b>												
17	Definir el problema	4 días												
18	Describir el dominio del sistema	5 días												
19	Identificar las fuentes de requerimientos	1 día												
20	Métodos, técnicas y herramientas para el desarrollo c	1 día												
21	Visión, alcance y fases de la estructura desglosada d	3 días												
22	Identificar las clases de usuarios, caracterizarlos y s	1 día												
23	Identificar a los que toman las decisiones de requerir	1 día												
24	Recolectar requerimientos según tareas colocadas e	30 días												

Id	Nombre de tarea	Duración	1er semestre		2º semestre		1er semestre		2º semestre		1er semestre		2º semestre	
			tri 4	tri 1	tri 2	tri 3	tri 4	tri 1	tri 2	tri 3	tri 4	tri 1	tri 2	tri 3
25	Señalar las fuentes de donde se tomaron los requerim													
26	<b>Análisis y negociación</b>													
27	Hacer los casos de uso de los requerimientos													
28	Dibujar el diagrama de contexto del sistema													
29	Crear prototipos de interfase con usuarios													
30	Analizar la factibilidad de los requerimientos													
31	Asignar prioridades y negociarlas con los clientes													
32	Crear un diccionario de datos													
33	<b>Especificaciones</b>													
34	Registrar reglas del negocio													
35	Adoptar plantillas de documentos													
36	Adoptar un esquema para la identificación de requeri													
37	Documentar requerimientos de sistema funcionales y													
38	Escribir una especificación de requerimientos de sist													
39	Escribir el manual del usuario correspondiente a este													
40	Documentar la arquitectura del sistema y la asignació													
41	<b>Verificación</b>													
42	Definir criterios de aceptación para cada requerimier													
43	Realizar inspecciones informales en cada una de las													
44	Revisar documentos de requerimientos que se hayan													
45	Escribir pruebas de caso de los requerimientos													
46	<b>Reuso</b>													
47	Actualizar la biblioteca de requerimientos reusables													
48	<b>Subproceso de diseño de la arquitectura de sistema</b>													
49	Diseñar la arquitectura del sistema													
50	Identificar elementos de HWSW y operaciones manuales													
51	<b>Documentar la arquitectura del sistema y asignación de re</b>													
52	Estructura estática, representada por DFD de alto nivel													
53	<b>Evaluar la arquitectura y asignación de requerimientos</b>													
54	Rastreabilidad para los requerimientos del sistema													

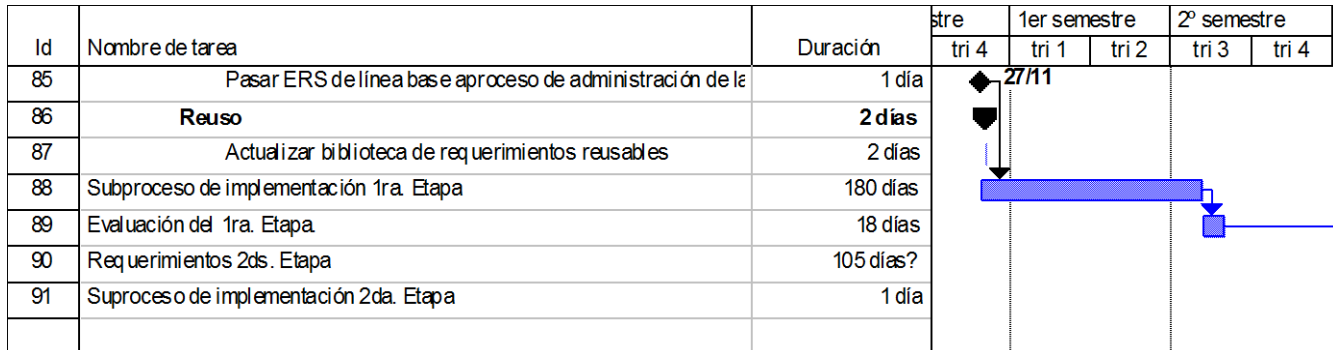
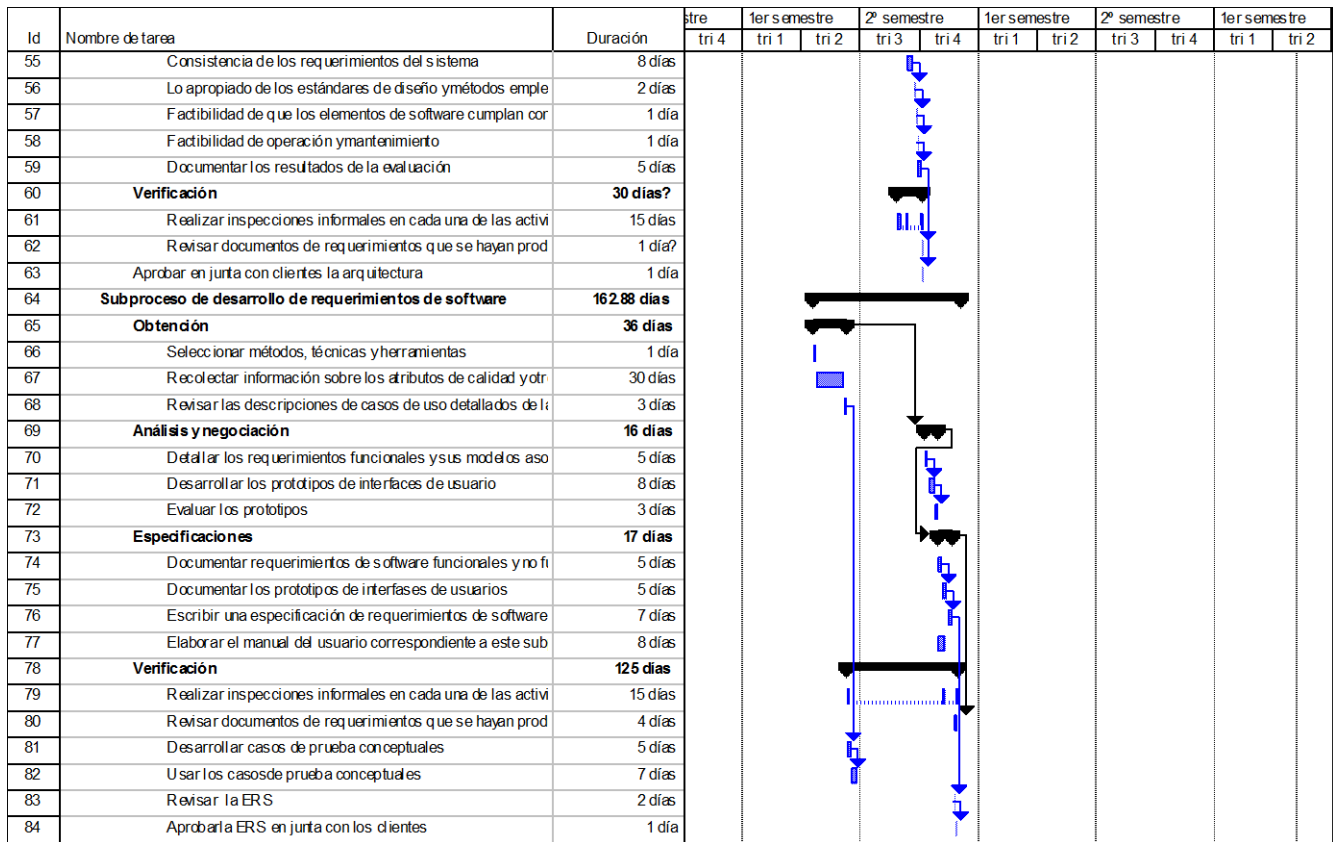


Fig 3.1.3.2 Diagrama de Gantt

### 3.1.4 Plan de administración del costo.

#### 3.1.4.1 Requerimiento de recursos.

En base a la experiencia de proyectos anteriores, confirmados en la primera etapa de este trabajo, se requiere el siguiente personal para la ingeniería de requerimientos de este proyecto.

<b>Personal</b>	<b>Cantidad</b>	<b>Tiempo</b>
Líder de proyecto	1	Completo
Analista	3	Completo
Documentador	1	Parcial
Probador	1	Parcial

### **3.1.4.2 Estimación de costos.**

Tomando en cuenta la duración estimada del proyecto de 586 días y el costo por hora del personal dedicado, tenemos:

<b>Personal.</b>	<b>Costo/Hr.</b>	<b>Hr.</b>	<b>Días</b>	<b>Total</b>
Líder del proyecto.	\$100.00	8	586	\$468,800
Analista 1	\$75.00	8	586	\$351,600
Analista 2	\$50.00	8	586	\$234,400
Analista 3	\$50.00	8	586	\$234,400
Documentador	\$50.00	4	586	\$117,200
Probador	\$50.00	4	586	\$117,200
				\$1'532,600

Los gastos indirectos no se cargan directamente al proyecto, si no a las dependencias encargadas del proyecto, estos gastos serían: depreciación de equipo de cómputo, depreciación de mobiliario de oficina, papelería, energía eléctrica, teléfono, etc.

### **3.1.4.3 Plan de administración de la variación de costos.**

Las variaciones de costos en el proyecto se manejarán de dos maneras.

- a) Las variaciones que generen un cambio en la línea base de costos menores al 20%, podrán ser autorizados directamente por el líder del proyecto.
- b) Las variaciones que generen un cambio en la línea base de costos mayor al 20% deberán ser autorizados por el comité de evaluación de cambios.

### **3.1.5 Plan de administración de calidad.**

#### **3.1.5.1 Políticas de calidad.**

Adoptar un ciclo de vida evolutivo, en el que se ve reflejado el ciclo de Deming (Planeación, Ejecución, Acción y Revisión) de mejora continua a través del tiempo. Además cada una de las etapas del proceso termina con una verificación de la misma para revisar que cumpla con las expectativas del cliente y con los estándares mencionados en el siguiente punto.

Las necesidades y expectativas del cliente, con respecto a la calidad están definidas en el documento de requerimientos, en la sección 3.2 Requerimientos del proyecto.

#### **3.1.5.2 Estándares o modelos a aplicar.**

Los diccionarios del proyecto estarán en concordancia con el IEEE610.12 (1990). IEEE Standard Glossary of Software Engineering Terminology.

Los documentos y manuales del usuario que se generen estarán en concordancia con el IEEE Standard for Software User Documentation.

Las especificaciones del sistema estarán de acuerdo con el IEEE1233 (1998). IEEE Guide for Developing System Requirements Specifications.

Las especificaciones de software estarán de acuerdo con el IEEE830 (1998). IEEE Recommended Practice for Software Requirements Specifications.

Los casos de uso y el modelado del sistema serán orientados a objetos.

#### **3.1.5.3 Actividades de calidad.**

Para este proyecto se consideran importantes dos actividades:

- a) Participación activa de los involucrados.
- b) Procesos de verificación en cada una de las etapas del proyecto.

### **3.1.6 Plan de administración del personal.**

#### **3.1.6.1 Asignaciones de roles y responsabilidades.**

	Líder de proyecto	Analista 1	Analista 2	Analista 3	Documentador	Probador
<b>Requerimientos del módulo de reportes históricos A3Lic.</b>						
<b>Administración del proyecto</b>						
<b>Planificar</b>						
Integración	A					
Tiempo	A					
Costos	A					
Calidad	A					
Personal	A					
Comunicaciones	A					
Riesgos	A					
Procuración	A					
Dar seguimiento	A	P	P	P		
Reportes periódicos	A	P	P	P		
<b>Requerimientos 1ra. Etapa</b>						
<b>Subproceso de desarrollo de requerimientos de sistema</b>						
<b>Requerimientos de sistema</b>						
Definir el problema	A	P	P	P		
Describir el dominio del sistema	A					
Identificar las fuentes de requerimientos	A					
Métodos, técnicas y herramientas para el desarrollo de req.	A	P	P	P		
Visión, alcance y fases de la estructura desglosada del trabajo	A				P	
Identificar las clases de usuarios, caracterizarlos y seleccionar	A					
Identificar a los que toman las decisiones de requerimientos	A					
Recolectar requerimientos según tareas colocadas en el EDT		A	A	A		
Señalar las fuentes de donde se tomaron los requerimientos		A	P	P		
<b>Análisis y negociación</b>						
Hacer los casos de uso de los requerimientos					A	
Dibujar el diagrama de contexto del sistema		A	P	P		
Crear prototipos de interfase con usuarios			A			P
Analizar la factibilidad de los requerimientos						A
Asignar prioridades y negociarlas con los clientes	A					
Crear un diccionario de datos					A	
<b>Especificaciones</b>						
Registrar reglas del negocio					A	



Adoptar plantillas de documentos		A			P	
Adoptar un esquema para la identificación de requerimientos	A	P	P	P		
Documentar requerimientos de sistema func. y no func.		A			P	
Escribir una especificación de requerimientos de sistema					A	
Escribir el manual del usuario correspondiente a subproceso					A	A
Documentar la arquitectura del sistema, hardware, software					A	
<b>Verificación</b>						
Definir criterios de aceptación para cada requerimiento	A					P
Realizar inspecciones informales en cada actividad						A
Revisar documentos de requerimientos producidos					A	A
Escribir pruebas de caso de los requerimientos					A	
<b>Reuso</b>						
Actualizar la biblioteca de requerimientos reusables					A	
<b>Subproceso de diseño de la arquitectura de sistema</b>						
Diseñar la arquitectura del sistema	A					
Identificar elementos de HW, .SW y operaciones manuales		A				
<b>Documentar la arquitectura del sistema y asignación de req.</b>						
Estructura estática, representada por DFD de alto nivel				A	P	
<b>Evaluar la arquitectura y asignación de requerimientos</b>						
Rastreabilidad para los requerimientos del sistema						I
Consistencia de los requerimientos del sistema		A				
Lo apropiado de los estándares de diseño y métodos	A					
Elementos de software cumplan con los requerimientos		A				
Factibilidad de operación y mantenimiento			A			
Documentar los resultados de la evaluación					A	
<b>Verificación</b>						
Realizar inspecciones informales en cada una de las act.						A
Revisar documentos de requerimientos producidos					A	A
Aprobar en junta con clientes la arquitectura	S					
<b>Subproceso de desarrollo de requerimientos de software</b>						
<b>Obtención</b>						
Seleccionar métodos, técnicas y herramientas	A					
Recolectar información sobre atributos de calidad req.		A	A	A		
Revisar las descripciones de casos de uso					A	
<b>Análisis y negociación</b>						
Detallar los requerimientos funcionales y sus modelos			A			
Desarrollar los prototipos de interfaces de usuario				A		
Evaluar los prototipos						

**Especificaciones**

Documentar req. de software funcionales y no funcionales

A

Documentar los prototipos de interfases de usuarios

A

Escribir una especificación de requerimientos de software

A

Elaborar el manual del usuario correspondiente a subproceso

A

**Verificación**

Realizar inspecciones informales en cada actividad

A

Revisar documentos de requerimientos producidos

Desarrollar casos de prueba conceptuales

A

Usar los casos de prueba conceptuales

A

Revisar la ERS

A

Aprobarla ERS en junta con los clientes

S

Pasar ERS de línea base a proceso de admón. de la config.

A

**Reuso**

Actualizar biblioteca de requerimientos reusables

A

Subproceso de implementación 1ra. Etapa

Evaluación del 1ra. Etapa.

Requerimientos 2ds. Etapa

Subproceso de implementación 2da. Etapa

### 3.1.6.2 Plan de administración del personal.

El personal asignado a este proyecto estará contratado por horas, limitadas por la jornada laboral completa o medio tiempo. Los periodos en los que se requiere su participación están indicados en la gráfica de Gantt del proyecto. (Ver apartado 3.1.3.2).

El personal asignado a este proyecto podrá participar en otros proyectos simultáneamente, para establecer la disponibilidad de tiempos, los líderes de proyectos deberán acordarlos de común acuerdo con el personal. Las horas laboradas destinadas a cada proyecto se registrarán en la bitácora personal de cada empleado.

El personal asignado a este proyecto también podrá intercambiar roles y responsabilidades de acuerdo a sus capacidades y habilidades y a las necesidades del proyecto.

### 3.1.6.3 Organigrama.

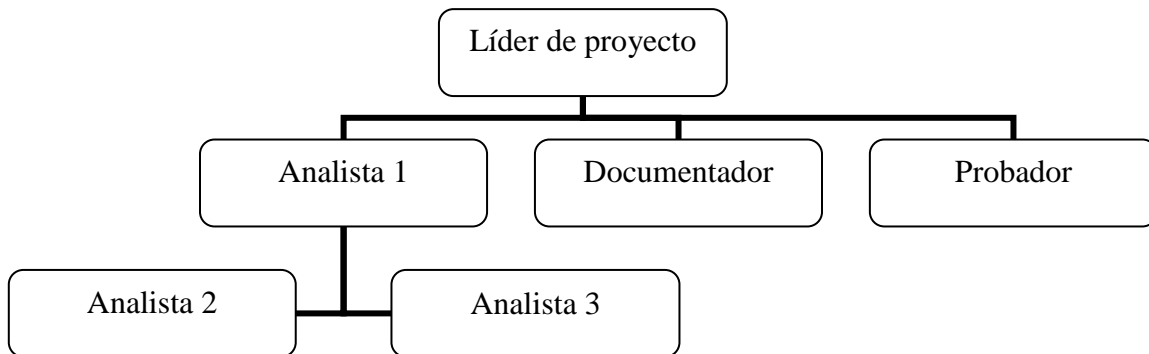


Fig 3.1.6.3 Organigrama equipo de ingeniería de requerimientos

### 3.1.6.4 Directorio del equipo de proyecto.

Puesto	Nombre	Ext.	Correo-e	Móvil
Líder de proyecto	Francisco Álvarez	3255	falvarez@iteso.mx	31 66 25 52
Analista 1	Francisco Santana	3255		
Analista 2	José Luis Navarro		is32393@iteso.mx	
Analista 3	Aitana Díaz			
Documentador	Raúl Estrella	3255		
Probador	Mario Ruiz	3255		

### **3.1.7 Plan de administración de comunicaciones.**

- a) Previos.- La comunicación clara, oportuna y enviada de manera regular a los participantes a lo largo de la duración del proyecto es vital para que no decaiga el esfuerzo. La comunicación debe permitir despertar y mantener el interés en el proyecto, desarrollar el sentido de pertenencia de los involucrados, cohesionar al equipo base, por la parte interna; así como lograr la participación de los involucrados en el proyecto.
- b) Objetivos.- El propósito de este plan de comunicación es:
  - a. Mantener vivo el interés del personal involucrado a lo largo del proyecto.
  - b. Desarrollar el sentido de pertenencia de los involucrados al programa.
  - c. Cohesionar al equipo base.
  - d. Informar a la Dirección General Académica, a la Subdirección de Servicios Escolares y a la Oficina de Sistemas de Información de los avances o retrasos y problemas del proyecto.
  - e. Citar a juntas y reuniones.
- c) Mensajes.- Los mensajes a enviar son:
  - a. De compromiso de las Direcciones y las Oficinas.
  - b. De inicio del proyecto.
  - c. De las bondades del proyecto.
  - d. De los logros (hitos) del proyecto.
  - e. De continuidad del proyecto.
  - f. De solicitudes derivadas del proyecto como: ayuda, información, etc.
  - g. Problemas/Retrasos.
  - h. Citas a reuniones.
  - i. De cambios al proyecto y su autorización correspondiente.
- d) Medios a utilizar.-
  - a. Carta personal (Cp)
  - b. Correo electrónico (Ce)
  - c. Teléfono (Te)
  - d. Juntas de información (Ji)
  - e. Juntas de seguimiento del proyecto (Js)
  - f. Bitácora del proyecto (Bi).
  - g. Notice de la Universidad (No).

- e) Auditorio.- Los grupos que podemos identificar a los que irá dirigida la comunicación son:
- a. Directores y Jefes de oficina.
  - b. Colegio de jefes.
  - c. Equipo base.
  - d. Coordinadores docentes.
  - e. Coordinadores de programa.
  - f. Encargados administrativos.
- f) Políticas de comunicación.
- a. Para que sea efectiva, la comunicación debe ser bidireccional (acuse de recepción, verificación de comprensión del mensaje)
  - b. La comunicación debe ser oportuna.
  - c. Debe mantenerse informado a todo el personal involucrado.
- g) Esquema de comunicación.-

	<b>Auditorio</b>					
<b>Información</b>	Directores y Jefes	Colegio de Jefes	Equipo base	Coordinadores docentes	Coordinadores de programa	Encargados administrativos
Compromiso	--	--	--	CE o Ce	CE o Ce	CE o Ce
Inicio	No	No	No	No	No	No
Bondades	--	--	--	Ji	Ji	Ji
Logros	--	Js	--	Js	Js	Js
Continuidad	--	--	--	Js	Js	Js
Solicitudes	CP / Js	CP / Js	--	--	--	--
Problemas/Retraso	Js	Js / Bi	Js / Bi			
Citas				CE o Ce / Te	CE o Ce / Te	CE o Ce / Te
Cambios	Js	Js	Js	--	--	--

### **3.1.8 Plan de administración de riesgos.**

- a) Metodología.- Para identificar los riesgos y para administrarlos se seguirá la metodología participativa descrita en el apartado 3.1.2.2 y se emplearan las matrices de los apartados 3.1.8.1 y 3.1.8.2.
- b) Roles y responsabilidades.- Son los definidos en el apartado 3.1.6.1
- c) Tiempos.- El proceso de la administración de los riesgos se realizará en conjunto con el proceso de verificación de cada una de las etapas del proyecto.
- d) Formatos de reporte.- Están en el anexo “Procedimientos para los planes de la administración del proyecto”
- e) Rastreo.- Se llevará un registro de las actividades del manejo de riesgos.

### 3.1.8.1 Matriz de Estimación de Riesgos.

Área de Riesgo.	Subárea de riesgo	Descripción del riesgo	Probabilidad de ocurrencia	Gravedad para el proyecto.	Impacto en el proyecto.
Proyecto	Novedad de un proyecto de ingeniería de requerimientos	Apatía, desinterés, incomprensión hacia el proyecto.	A	A	A
		Participación del usuario.	M	A	A
		Los requerimientos no son tomados en cuenta por la OSI	M	A	A
Requerimientos de sistema, diseño, requerimientos de software.	Metodología	Muy complicada para el contexto del Iteso. Que consuma mucho tiempo o recursos.	M	A	A
	DGS/SE	Apoyo insuficiente.	B	A	A
		Indefinición cambios, falta de consenso.	A	A	A
	Departamentos	Apoyo insuficiente.	A	A	A
	OSI	Fuera de alcance por tiempo y presupuesto.	M	A	M
		Entorno cerrado.	A	A	A
	Usuarios	Apatía, desinterés, incomprensión del proyecto.	A	A	A
		Falta de conocimiento de sus necesidades reales.	A	A	A

### 3.1.8.2 Matriz de Administración de Riesgos.

Área de Riesgo.	Subárea de riesgo	Descripción del riesgo	Acciones preventivas	Acciones correctivas.	Resp.
Proyecto	Novedad de un proyecto de Ingeniería de requerimientos	Apatía, desinterés, incompreensión hacia el proyecto.	Campaña de información y promoción.	Mostrar casos de éxito del proyecto.	Líder de proyecto
		Participación del usuario.	Incorporar dentro de sus encargos. Motivar.	Acudir a DGA. Definir e instrumentar acciones de recuperación de la experiencia en el proyecto.	Líder de proyecto
		Los requerimientos no son tomados en cuenta por la OSI	Dialogar con la OSI	Acudir a DAF	Líder de proyecto
Requerimientos de sistema, diseño, requerimientos de software.	Metodología	Muy complicada para el contexto del Iteso. Que consume mucho tiempo o recursos.	Hacer un diagnóstico previo sobre el tipo de metodología a aplicar	Realizar cambios en las prioridades establecidas.	Líder de proyecto.



<b>Área de Riesgo.</b>	<b>Subárea de riesgo</b>	<b>Descripción del riesgo</b>	<b>Acciones preventivas</b>	<b>Acciones correctivas.</b>	<b>Resp.</b>
	DGS/SE	Apoyo insuficiente.	Mostrar los impactos esperados del proyecto. Explicitar las necesidades de apoyo.	Acudir con los directores.	Directores.
		Indefinición cambios, falta de consenso.	Generar una visión común.	Acudir con los directores. Establecer puntos de acuerdos.	Directores.
	Departamentos	Apoyo insuficiente.	Incluir el proyecto dentro de los encargos.	Acudir con los jefes de los Departamentos	Jefes de departamentos
	OSI	Fuera de alcance por tiempo y presupuesto.	Planear en conjunto los alcances. Obtener presupuesto.	Acudir con el director de la DAF	Líder de proyecto.
		Entorno cerrado.	Dialogar sobre los objetivos del proyecto	Acudir con el jefe de la OSI	Líder de proyecto.

<b>Área de Riesgo.</b>	<b>Subárea de riesgo</b>	<b>Descripción del riesgo</b>	<b>Acciones preventivas</b>	<b>Acciones correctivas.</b>	<b>Resp.</b>
	Usuarios	Apatía, desinterés, incomprensión del proyecto.	Motivarlos, platicar con ellos, capacitarlos	Acudir con sus jefes.	Líder de proyecto.
		Falta de conocimiento de sus necesidades reales.	Platicar con ellos, cuestionarlos, hacer que reflexionen.	Trabajar con versiones en las que se irán incorporando las nuevas necesidades detectadas.	Líder de proyecto.

### **3.1.9 Plan de administración de la procuración.**

Para este proyecto se emplearán becarios y prestadores de servicio social, dirigidos por académicos de planta, que tienen entre sus encargos institucionales, la realización del mismo.

En caso de ser necesaria la contratación del personal para este proyecto, las contrataciones se harán de acuerdo al “Procedimiento de Contratación por Honorarios Profesionales y Honorarios Asimilables a Sueldos” emitido por la Oficina de Personal de la Dirección Administrativa y de Finanzas del Iteso. Para las contrataciones se deberá llenar el formato de la “Solicitud de contratación por Servicios Profesionales Independientes bajo los esquemas de honorarios”, emitido por la misma oficina.

Para este proyecto se utilizarán el equipo y los materiales ya existentes en el Iteso, contemplados durante los ejercicios presupuestales 2001 a 2003, en que se llevará a cabo el proyecto, pertenecientes a la EDD y las dependencias involucradas.

En caso de ser necesaria la compra de equipo o material adicional, realización de viajes, etc., para la realización del proyecto, las compras se harán de acuerdo a los siguientes procedimientos: “Procedimiento de Planeación y Adquisición de Equipo y Programas de Cómputo”, “Procedimiento de Requisición de compra y suministro de productos y servicios comerciales” y “Procedimiento de pago y comprobación de servicios de viaje y hospedaje”.

Para cualquier otro suministro no incluido en este plan, se seguirán los procedimientos institucionales establecidos según el caso.

## 3.2 Requerimientos del proyecto.

### 3.2.1 Requerimientos del Sistema.

#### 3.2.1.1 Requerimientos Funcionales del Sistema.

<b>ID</b>	<b>Requerimiento</b>	<b>Est.</b>	<b>Pri.</b>	<b>Dep.</b>	<b>Comentarios</b>
RSIF001	La plataforma de desarrollo y presentación será Windows 98, NT o XP	Alta	Alta		
RSIF002	Para el proyecto se utilizará Delphi en su versión 6 como herramienta de desarrollo	Alta	Alta		
RSIF003	El manejador de base de datos será Oracle versión 8.1	Alta	Alta		
RSIF004	El sistema operativo es Debian versión 3	Alta	Alta		
RSIF005	Deberá correr en una red Ethernet con topología de estrella, cable categoría 5 o fibra óptica, con velocidad mínima de 10Mbps	Alta	Alta		
RSIF006	El programa se podrá ejecutar en cualquier computadora en red dentro del campus del Iteso.	Alta	Alta		
RSIF007	El programa correrá en un esquema cliente servidor	Alta	Alta		
RSIF008	El programa correrá en procesadores Pentium II o superiores	Alta	Alta		
RSIF043	El usuario podrá mandar a imprimir un reporte a impresora local o de red.	Alta	Alta	RSWOI016	

<b>ID</b>	<b>Requerimiento</b>	<b>Est.</b>	<b>Pri.</b>	<b>Dep.</b>	<b>Comentarios</b>
RSIF055	Dar capacidad de almacenar información de las A3Lic histórica y permitir acceder a esta información. Los periodos pueden ser semestre otoño, semestre verano o semestre primavera.				

### 3.2.1.2 Requerimientos no Funcionales del Sistema.

#### 3.2.1.2.1 Requerimientos de usabilidad del Sistema.

<b>ID</b>	<b>Requerimiento</b>	<b>Est</b>	<b>Pri</b>	<b>Dep.</b>	<b>Comentarios</b>
RSIPU010	Interfase sencilla, sin una gran carga de menús, solamente los indispensables. Para realizar cualquier operación evitar muchos rodeos y tener las opciones más utilizadas lo más intuitiva y práctica posible.	Media	Media		
RSIPU011	Utilizará los Hotkeys más utilizados en otros programas (Control+C -> copiar; Control +z Undo; etc.)	Media	Baja	RSWOI012	
RSIPU013	Ventana principal contará con una barra en la parte superior que será el "Menú Principal", este a su vez contendrá submenús para opciones más especializadas.	Alta	Alta	RSWOI014	
RSIPU015	Considerando la opción de implementar el sistema en MS-DOS, se haría una sucesión de ventanas de modo texto, donde se filtrarían las opciones obteniendo finalmente un reporte concreto. Estas ventanas podrán regresarse a fin de corregir algún detalle del reporte o gráfica.	Baja	Media	RSWOI015	

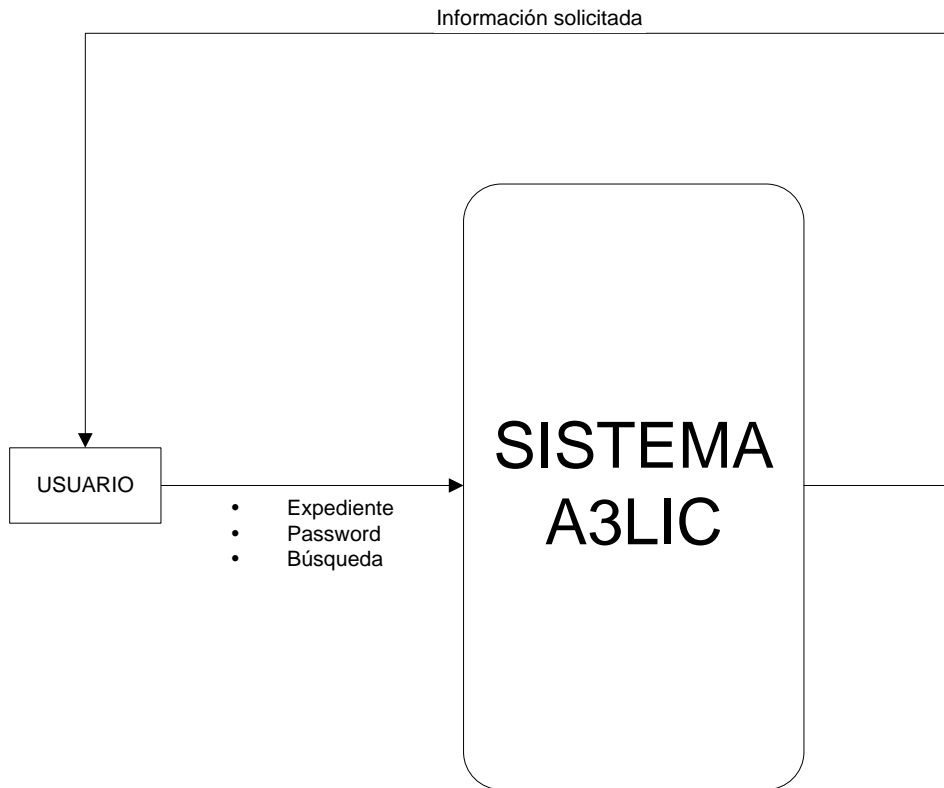
<b>ID</b>	<b>Requerimiento</b>	<b>Est</b>	<b>Pri</b>	<b>Dep.</b>	<b>Comentarios</b>
RSIPU019	Las ventanas podrán minimizarse dentro de la ventana principal para poder tener disponibles varias búsquedas. Sin que estas se confundan o entorpezcan el trabajo.	Media	Media		
RSIF052	El sistema tiene opción de cambiar los colores dentro de las gráficas.	Media	Media	RSWOI005	
RSIF046	El usuario podrá personalizar los reportes, es decir que el usuario podrá definir el contenido de los mismos (pie de página, título del reporte).	Media	Media	RSWOI005	

### 3.2.1.2.2 Requerimientos del entregable

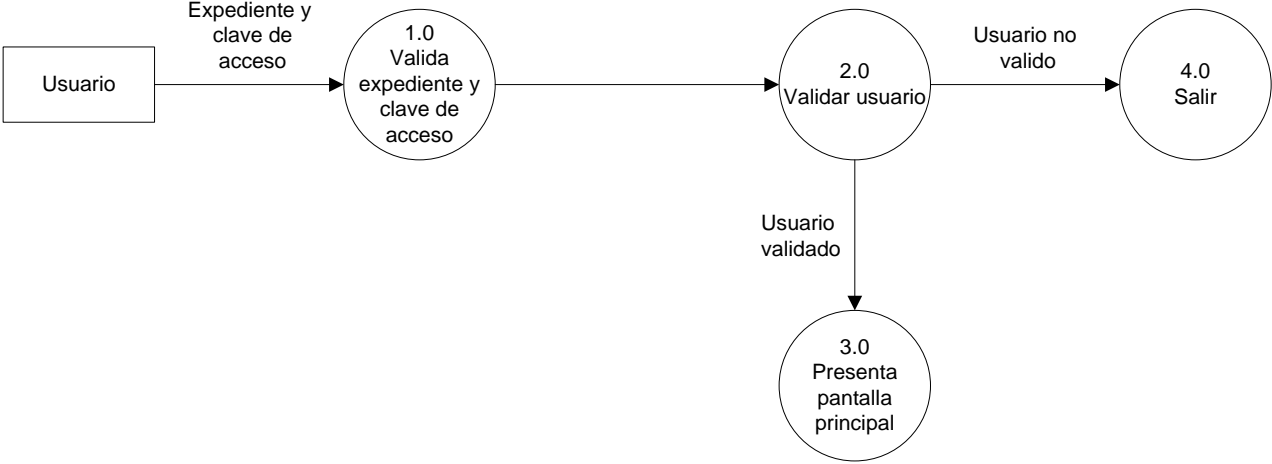
<b>ID</b>	<b>Requerimiento</b>	<b>Est</b>	<b>Pri</b>	<b>Dep.</b>	<b>Comentarios</b>
RSIOE001	El sistema contendrá ayuda en línea.	Media	Media		
RSIOE002	Todos los iconos tendrán tool tips.	Media	Baja	RSWOI017	
RSIOE004	Se entregará un manual técnico orientado a los administradores del sistema, donde se especifica: <ul style="list-style-type: none"> <li>- Estructura de tablas y bases de datos</li> <li>- Información para la creación de tablas especializadas para maestros</li> <li>- Programas del sistema</li> <li>- Anexo del manual de usuario</li> <li>- Explicación y ejemplos del uso de opciones para administradores.</li> <li>- Problemas y soluciones comunes.</li> <li>- Requerimientos de Hardware.</li> </ul>	Media	Media		

### 3.2.2 Diseño del sistema.

#### 3.2.2.1 Diagrama de Flujo de Datos - 0

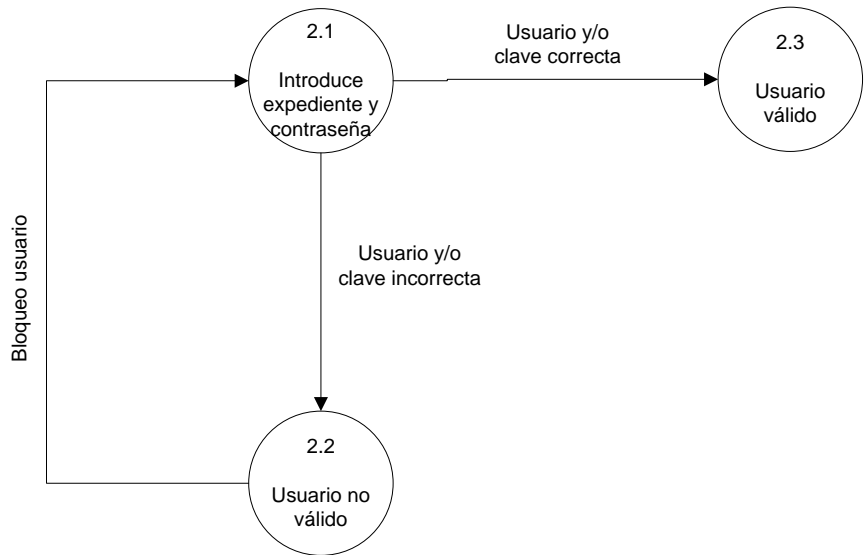


**3.2.2.2 Diagrama de Flujo de Datos - 1**





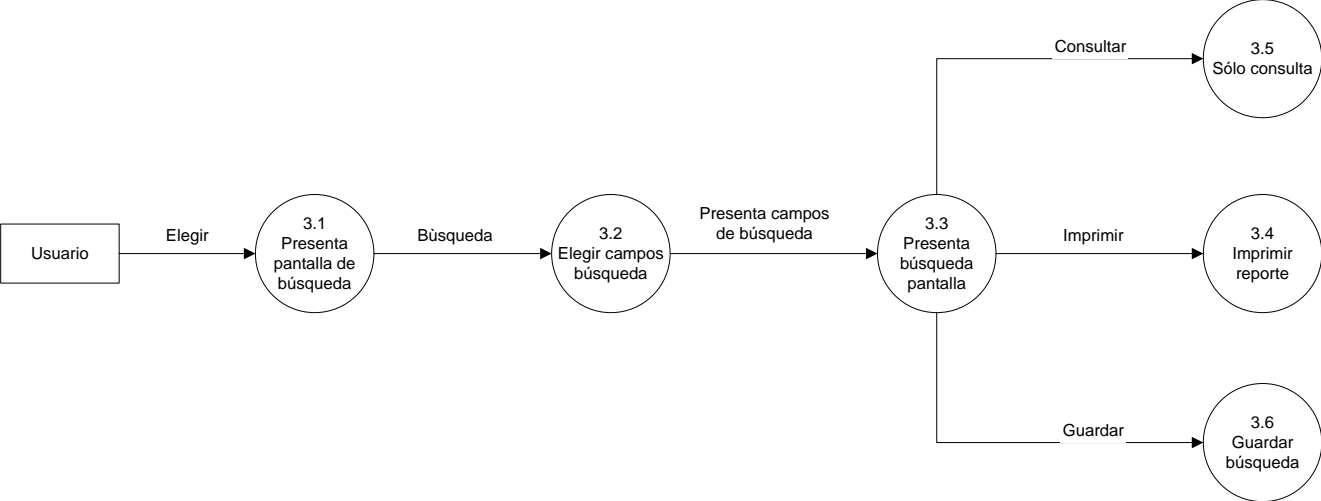
### 3.2.2.3 Diagrama de Flujo de Datos - 2



#### **Bloquear Usuario:**

- Si el usuario introduce 3 veces mal su password, el sistema lo bloqueará por cuestiones de seguridad. Para habilitar al suario nuevamente, tendrá que acudir con la persona indicada

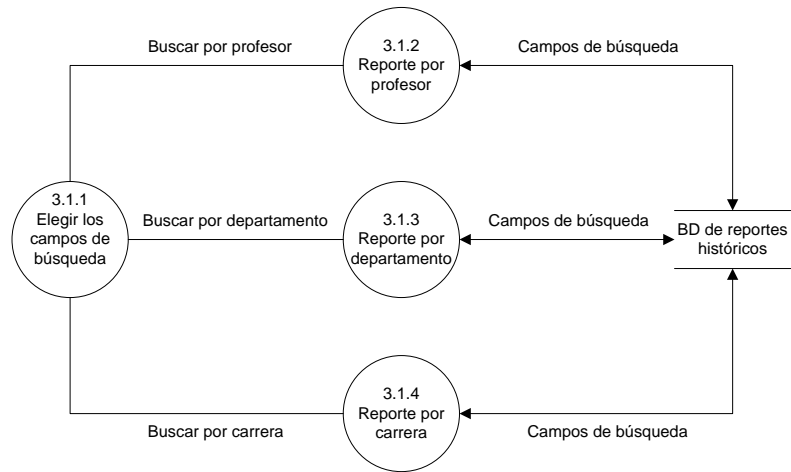
### 3.2.2.4 Diagrama de Flujo de Datos - 3



La Búsqueda será definida mediante los siguientes campos:

- Periodo.
- Número de expediente.
- Departamento.

### 3.2.2.5 Diagrama de flujo de Datos - 4



#### Reporte por departamento

La búsqueda contendrá los siguientes campos:

- Periodo
- Nombre profesor
- Expediente profesor
- Departamento
- UAB
- Academia
- Nombre materia
- Clave materia
- Turno

#### Reporte por profesor

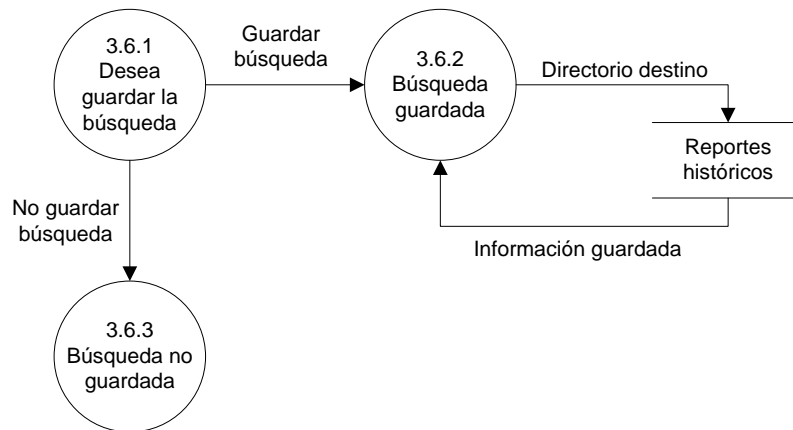
La búsqueda podrá contener uno o más de los siguientes campos:

- Periodo
- Número de expediente
- Departamento

#### Reporte por carrera presentará en pantalla las siguientes opciones de búsqueda:

- Periodo
- Presentar por rango de créditos cursados

### 3.2.2.5.1 Diagrama de flujo de datos - 5



**Directorio destino:**

Se refiere al directorio de la carpeta donde el usuario guardará su búsqueda

### 3.2.3 Requerimientos de software.

#### 3.2.3.1 Requerimientos Funcionales de Software.

ID	Requerimiento	Est	Pri	Dep.	Comentarios
RSWF001	En los reportes numéricos y por departamento el sistema deberá obtener los promedios para cada <i>Item y área de la encuesta</i>	Alta	Alta		
RSWF016	El sistema deberá proveer al usuario la capacidad de buscar los comentarios de un profesor en una materia determinada, en un periodo determinado.	Media	Alta		
RSWF018	El sistema debe proveer al usuario la capacidad de consultar el A3Lic para cada maestro dentro de un periodo especificado. Es decir, obtener un reporte de cada materia con la correspondiente información en cada <i>Área e Ítem</i> , para ese profesor en ese periodo.	Alta	Alta		
RSWF023	El usuario podrá ver los resultados de un profesor para una determinada materia, en todas sus Áreas o Ítems de la evaluación.	Media	Media	RSIF022, RSWOI001	
RSWF044	Podrá hacerse una consulta histórica desde evaluaciones pasadas al periodo actual, y que esa información esté respaldada	Media	media		
RSWF045	Cuando se realice una búsqueda esta se grabará como Script de las búsquedas, serán grabados en la red, con un identificador de usuario, para que solamente ese usuario pueda utilizarlos.	Baja	Alta		

<b>ID</b>	<b>Requerimiento</b>	<b>Est</b>	<b>Pri</b>	<b>Dep.</b>	<b>Comentarios</b>
RSWF049	El programa cliente podrá guardar archivos que contengan búsquedas, para que el usuario pueda posteriormente ejecutarlas sin necesidad de establecer las opciones desde el principio.	Media	Media		
RSWF048	El programa cliente podrá acceder a los archivos que contengan búsquedas anteriores o recientes, para que el usuario pueda nuevamente ejecutarlas sin necesidad de establecer las opciones desde el principio.	Media	Media		

### 3.2.3.2 Requerimientos no Funcionales de Software.

<b>ID</b>	<b>Requerimiento</b>	<b>Est</b>	<b>Pri</b>	<b>Dep.</b>	<b>Comentarios</b>
RSWPU009	El sistema abrirá una ventana para cada búsqueda si necesidad de tener que borrar búsquedas recientes, y tenerlas en una ventana.	Media	Media		
RSWPU016	En los reportes no aparecerá el número del ítem que se está imprimiendo, sino el aspecto que se evalúa, por ejemplo: Si puntualidad fuera el <i>Ítem No. 1</i> , en los reportes aparecerá " <i>Puntualidad</i> ", no <i>Ítem 1</i> .	Alta	Media	RSWOI016	
RSWPU018	El historial será un vaciado completo de resultados de todos los maestros, en todos los ítems de todas las materias. Se presentan a manera de tablas.	Media	Media		

ID	Requerimiento	Est	Pri	Dep.	Comentarios
RSWOE003	Se entregará un manual de usuario, orientado a las personas que utilizaran el programa, generaran reportes y gráficas, el manual les dirá los pasos a seguir, explicará las opciones y también asesorará sobre el uso de la ayuda en línea.	Media	Media		
RSWELS001	Las búsquedas serán personales y privadas, el almacenamiento se describe en RI018.	Media	Alta	RSIF049, RSIF045	
RSWELS002	Las tablas que lleguen al usuario final estarán seleccionadas con anterioridad por un administrador o encargado de departamento. En RI019 se especifica como puede realizarse eso.	Alta	Alta		
RSWPP001	Que el software sea estable, con los menos errores posibles.	Alta	Alta		Se hará una serie de pruebas a fin de garantizar un buen desempeño del producto final. Para poder determinar si el software es estable se analizará el tiempo promedio de uso, así como la cantidad de veces que el SW es utilizado, a fin de obtener parámetros válidos para evaluar el número de fallas vs. Tiempo ó número de ejecuciones.

ID	Requerimiento	Est	Pri	Dep.	Comentarios
RSWOI003	Los ítems y materias podrán ser seleccionados con check Box's que permitan un acceso más rápido y sencillo.	Media	Media		El software de desarrollo se podrían implementar árboles que agilicen la búsqueda. Estos árboles categorizarían los empleados por departamentos y materias. Y las materias se acomodarían por departamentos.
RSWOI009	El sistema podrá guardar las búsquedas realizadas por el usuario de la información de los A3.	Baja	Media	RSWOI008	De ser el almacenamiento en red, es posible que por seguridad y espacio no se guarden más que los scripts para generar los reportes y gráficas. (El resultado de los reportes, o gráficas). A diferencia de las búsquedas que son a manera de Scripts estas serán las gráficas o reportes completos.
RSWOI010	El sistema será capaz de generar múltiples ventanas por cada búsqueda. Se generará una forma principal para cada búsqueda.	Media	Media		



<b>ID</b>	<b>Requerimiento</b>	<b>Est</b>	<b>Pri</b>	<b>Dep.</b>	<b>Comentarios</b>
RSWOI012	Los Hotkeys que se implementarán incluyen: Guardar, cerrar, Abrir, Seleccionar Todo,	Media	Media		Basándose en programas de mayor uso como: Word, Explorer, Netscape Etc. para escoger la combinación de las teclas.
RSWOI014	Se implementará una barra de menú principal con submenús si fuera necesario.	Alta	Alta		
RSWOI016	En los reportes, búsquedas o gráficas, no aparecerá el número del ítem que se está imprimiendo, sino la descripción textual del ítem.	Alta	Media		Por ejemplo: Si puntualidad fuera el Ítem No. 1, en los reportes aparecerá "Puntualidad", no Ítem 1. Es decir el programa desplegará el campo, no un identificador que el usuario no pueda entender claramente o no le sea útil. Este requerimiento se refiere a que el sistema hará una liga entre el identificador del campo y un descriptor que sea entendible por los usuarios.
RSWOI017	Los botones de menú tendrán un tool tip que oriente sobre el uso de la opción.	Alta	Baja		Esto lo permite el software de desarrollo.

ID	Requerimiento	Est	Pri	Dep.	Comentarios
RSWOI018	Se hará un formato de Script para repetir búsquedas, estas podrán ser cargadas por un usuario y repetir esa búsqueda para generar reportes o gráficas	Baja	Alta		
RSWOI019	Los Scripts que permiten replicar búsquedas estarán almacenados en la cuenta del usuario, de tal manera que sólo el tenga acceso a ellos. El almacenamiento del Script se hará en una estructura de datos que se guardará en un archivo de texto.	Media	Alta	RSWOI018	Se decidirá por una de las dos opciones, siendo más viable la segunda.

### 3.2.4 Casos de Uso

#### 3.2.4.1 CU1RepNumA3

**IDENTIFICACIÓN:** CU1RepNumA3

**Prioridad:** Alta

**NOMBRE:** Reporte numérico A3 tipos: Lic1, Lic2, Lic3 y Pos

**CREADO POR:** Raúl y Mario

**ÚLTIMA ACTUALIZACIÓN POR:** Raúl

**FECHA CREACIÓN:** 01/07/03

**FECHA ÚLTIMA ACTUALIZACIÓN:** 20/08/03

**ACTORES:** Profesores, coordinadores docentes, coordinadores de programa, coordinadores de UAB, tutores, secretarías, encargados administrativos, jefes, servicios escolares, DGA

**DESCRIPCIÓN:** Los usuarios tipo Profesor tendrán acceso a los reportes numéricos que sean únicamente suyos por cada período.

Los usuarios tipo coordinadores docentes, coordinadores de programa, coordinadores de UAB, tutores, secretarías tendrán acceso a los reportes numéricos que sean de su departamento o centro por cada período y por expediente de profesor.

Los usuarios tipo encargados administrativos, jefes, servicios escolares, DGA, tendrán acceso a los reportes numéricos de todo el ITESO por cada período, por cada departamento y por expediente de profesor.

**FRECUENCIA DE USO:** Cada vez que necesita información de algún profesor y al final del semestre.

### CURSO NORMAL DE EVENTOS

#### ACCIONES DE ACTORES

- 1.-Doble clic en el icono del programa
- 3.-Introducir expediente y clave de acceso
- 6.-Seleccionar tipo de reporte
- 8.-Seleccionar campos para realizar búsqueda
- 11.-Guarda búsqueda
- 15.-Sale del sistema

#### RESPUESTA DEL SISTEMA

- 2.-Presenta pantalla de Id y clave de acceso
- 4.-Validar exp y clave.
- 5.-Muestra menú principal.
- 7.-Presenta pantalla reporte seleccionado.
- 9.-Realiza búsqueda
- 10.-Presenta búsqueda
- 12.- Pregunta Directorio
- 13.- Guarda archivo
- 14.- Regresa a pantalla de búsqueda
- 16.- Cierra sesión usuario.

## CURSOS ALTERNATIVOS DE EVENTOS

### ACCIONES DE ACTORES

#### Alternativa 1: “No quiere guardar búsqueda”

11.- No guardar búsqueda

13.- Confirma opción

#### Alternativa 2: “Imprimir reporte”

11.- Mandar imprimir reporte

13.- Elige opción de impresión

17.- Sale del sistema

### RESPUESTAS DEL SISTEMA

12.- Pregunta si está seguro

14.-Regresa pantalla búsqueda

12.- Presenta opciones de impresión

14.- Captura opción de impresión

15.- Manda imprimir reporte

16.- Regresa a pantalla de búsqueda.

18.- Cierra sesión usuario

## EXCEPCIONES

Error en la búsqueda: el usuario se equivoca de campo y el sistema devuelve una búsqueda no deseada.

El usuario deberá regresar a la pantalla de búsqueda y volver a solicitar una búsqueda.

Error al inicio de sesión: el usuario introdujo mal su expediente o clave de acceso. El sistema mandará aviso de error y permitirá al usuario intentar de nuevo entrar al sistema.

### 3.2.4.2 CU2RepComA3

**IDENTIFICACIÓN:** CU2RepComA3

**Prioridad:** Alta

**NOMBRE:** Reporte de comentarios A3 tipos: Lic1, Lic2, Lic3 y Pos

**CREADO POR:** Raúl y Mario

**ÚLTIMA ACTUALIZACIÓN POR:** Raúl

**FECHA CREACIÓN:** 01/07/03

**FECHA ÚLTIMA ACTUALIZACIÓN:** 20/08/03

**ACTORES:** Profesores, coordinadores docentes, coordinadores de programa, coordinadores de UAB, tutores, secretarías, encargados administrativos, jefes, servicios escolares, DGA

**DESCRIPCIÓN:** Los usuarios tipo Profesor tendrán acceso a los reportes de comentarios que sean únicamente suyos por cada período.

Los usuarios tipo coordinadores docentes, coordinadores de programa, coordinadores de UAB, tutores, secretarías tendrán acceso a los reportes de comentarios que sean de su departamento o centro por cada período y por expediente.

Los usuarios tipo encargados administrativos, jefes, servicios escolares, DGA, tendrán acceso a los reportes numéricos de comentarios de todo el ITESO por cada período, por cada departamento y por expediente.

**FRECUENCIA DE USO:** Cada vez que necesita información de algún profesor y al final del semestre.

### CURSO NORMAL DE EVENTOS

#### ACCIONES DE ACTORES

- 1.-Doble clic en el icono del programa
- 3.-Introducir expediente y clave de acceso
- 6.-Seleccionar tipo de reporte
- 8.-Seleccionar campos para realizar búsqueda
- 11.-Guarda búsqueda
- 15.-Sale del sistema

#### RESPUESTA DEL SISTEMA

- 2.-Presenta pantalla de Id y clave de acceso
- 4.-Validar exp y clave.
- 5.-Muestra menú principal.
- 7.-Presenta pantalla reporte seleccionado.
- 9.-Realiza búsqueda
- 10.-Presenta búsqueda
- 12.- Pregunta Directorio
- 13.- Guarda archivo
- 14.- Regresa a pantalla de búsqueda
- 16.- Cierra sesión usuario.

## CURSOS ALTERNATIVOS DE EVENTOS

### ACCIONES DE ACTORES

#### Alternativa 1: “No quiere guardar búsqueda”

- 11.- No guardar búsqueda
- 13.- Confirma opción

#### Alternativa 2: “Imprimir reporte”

- 11.- Mandar imprimir reporte
- 13.- Elige opción de impresión

- 17.- Sale del sistema

### RESPUESTAS DEL SISTEMA

- 12.- Pregunta si está seguro
- 14.-Regresa pantalla búsqueda
- 12.- Presenta opciones de impresión
- 14.- Captura opción de impresión
- 15.- Manda imprimir reporte
- 16.- Regresa a pantalla de búsqueda.
- 18.- Cierra sesión usuario

## EXCEPCIONES

Error en la búsqueda: el usuario se equivoca de campo y el sistema devuelve una búsqueda no deseada.

El usuario deberá regresar a la pantalla de búsqueda y volver a solicitar una búsqueda.

Error al inicio de sesión: el usuario introdujo mal su expediente o clave de acceso. El sistema mandará aviso de error y permitirá al usuario intentar de nuevo entrar al sistema.

### 3.2.4.3 CU3RepHistA3

**IDENTIFICACIÓN:** CU3RepHistA3

**Prioridad:** Alta

**NOMBRE:** Reporte histórico por profesor.

**CREADO POR:** Raúl y Mario

**ÚLTIMA ACTUALIZACIÓN POR:** Raúl

**FECHA CREACIÓN:** 01/07/03

**FECHA ÚLTIMA ACTUALIZACIÓN:**20/08/03

**ACTORES:** Profesores, coordinadores docentes, coordinadores de programa, coordinadores de UAB, tutores, secretarias, encargados administrativos, jefes, servicios escolares, DGA

**DESCRIPCIÓN:** Los usuarios tipo Profesor tendrán acceso a los reportes históricos que sean únicamente suyos.

Los usuarios tipo coordinadores docentes, coordinadores de programa, coordinadores de UAB, tutores, secretarias tendrán acceso a los reportes históricos que sean de su departamento o centro por expediente.

Los usuarios tipo encargados administrativos, jefes, servicios escolares, DGA, tendrán acceso a los reportes históricos de todo el ITESO por cada departamento y por expediente.

**FRECUENCIA DE USO:** Cada vez que necesita información de algún profesor y al final del semestre.

#### CURSO NORMAL DE EVENTOS

##### ACCIONES DE ACTORES

- 1.-Doble clic en el icono del programa
- 3.-Introducir expediente y clave de acceso
- 6.-Seleccionar tipo de reporte
- 8.-Seleccionar campos para realizar búsqueda
- 11.-Guarda búsqueda
- 15.-Sale del sistema

##### RESPUESTA DEL SISTEMA

- 2.-Presenta pantalla de Id y clave de acceso
- 4.-Validar exp y clave.
- 5.-Muestra menú principal.
- 7.-Presenta pantalla reporte seleccionado.
- 9.-Realiza búsqueda
- 10.-Presenta búsqueda
- 12.- Pregunta Directorio
- 13.- Guarda archivo
- 14.- Regresa a pantalla de búsqueda
- 16.- Cierra sesión usuario.

#### CURSOS ALTERNATIVOS DE EVENTOS

## **ACCIONES DE ACTORES**

### **Alternativa 1: “No quiere guardar búsqueda”**

11.- No guardar búsqueda

13.- Confirma opción

### **Alternativa 2: “Imprimir reporte”**

11.- Mandar imprimir reporte

13.- Elige opción de impresión

17.- Sale del sistema

## **RESPUESTAS DEL SISTEMA**

12.- Pregunta si está seguro

14.-Regresa pantalla búsqueda

12.- Presenta opciones de impresión

14.- Captura opción de impresión

15.- Manda imprimir reporte

16.- Regresa a pantalla de búsqueda.

18.- Cierra sesión usuario

## **EXCEPCIONES**

Error en la búsqueda: el usuario se equivoca de campo y el sistema devuelve una búsqueda no deseada.

El usuario deberá regresar a la pantalla de búsqueda y volver a solicitar una búsqueda.

Error al inicio de sesión: el usuario introdujo mal su expediente o clave de acceso. El sistema mandará aviso de error y permitirá al usuario intentar de nuevo entrar al sistema.



### 3.2.4.4 CU4RepDepA3

**IDENTIFICACIÓN:** CU4RepDepA3

**Prioridad:** Alta

**NOMBRE:** Reporte departamental A3 tipos: Lic1, Lic2, Lic3 y Pos.

**CREADO POR:** Raúl y Mario

**ÚLTIMA ACTUALIZACIÓN POR:** Raúl

**FECHA CREACIÓN:** 01/07/03

**FECHA ÚLTIMA ACTUALIZACIÓN:**20/08/03

**ACTORES:** Coordinadores docentes, coordinadores de programa, coordinadores de UAB, tutores, secretarias, encargados administrativos, jefes, servicios escolares, DGA

**DESCRIPCIÓN:** Los usuarios tipo coordinadores docentes, coordinadores de programa, coordinadores de UAB, tutores, secretarias tendrán acceso a los reportes departamentales que sean de su departamento o centro por cada período escolar.

Los usuarios tipo encargados administrativos, jefes, servicios escolares, DGA, tendrán acceso a los reportes departamentales de todo el ITESO por cada período escolar y por cada departamento.

**FRECUENCIA DE USO:** Cada vez que necesita información de algún profesor y al final del semestre.

#### **CURSO NORMAL DE EVENTOS**

##### **ACCIONES DE ACTORES**

##### **RESPUESTA DEL SISTEMA**

1.-Doble clic en el icono del programa

2.-Presenta pantalla de Id y clave de acceso

3.-Introducir expediente y clave de acceso

4.-Validar exp y clave.

5.-Muestra menú principal.

6.-Seleccionar tipo de reporte

7.-Presenta pantalla reporte seleccionado.

8.-Seleccionar campos para realizar búsqueda

9.-Realiza búsqueda

10.-Presenta búsqueda

11.-Guarda búsqueda

12.- Pregunta Directorio

13.- Guarda archivo

14.- Regresa a pantalla de búsqueda

15.-Sale del sistema

16.- Cierra sesión usuario.

#### **CURSOS ALTERNATIVOS DE EVENTOS**

##### **ACCIONES DE ACTORES**

##### **RESPUESTAS DEL SISTEMA**

**Alternativa 1: “No quiere guardar búsqueda”**

11.- No guardar búsqueda

13.- Confirma opción

**Alternativa 2: “Imprimir reporte”**

11.- Mandar imprimir reporte

13.- Elige opción de impresión

17.- Sale del sistema

12.- Pregunta si está seguro

14.-Regresa pantalla búsqueda

12.- Presenta opciones de impresión

14.- Captura opción de impresión

15.- Manda imprimir reporte

16.- Regresa a pantalla de búsqueda.

18.- Cierra sesión usuario

### **EXCEPCIONES**

Error en la búsqueda: el usuario se equivoca de campo y el sistema devuelve una búsqueda no deseada.

El usuario deberá regresar a la pantalla de búsqueda y volver a solicitar una búsqueda.

Error al inicio de sesión: el usuario introdujo mal su expediente o clave de acceso. El sistema mandará aviso de error y permitirá al usuario intentar de nuevo entrar al sistema.

### 3.2.4.5 CU5RepProgA3

**IDENTIFICACIÓN:** CU5RepProgA3

**Prioridad:** Media

**NOMBRE:** Reporte por programa A3 tipos: Lic1, Lic2, Lic3 y Pos.

**CREADO POR:** Raúl y Mario

**ÚLTIMA ACTUALIZACIÓN POR:** Raúl

**FECHA CREACIÓN:** 01/07/03

**FECHA ÚLTIMA ACTUALIZACIÓN:**20/08/03

**ACTORES:** Coordinadores docentes, coordinadores de programa, coordinadores de UAB, tutores, secretarias, encargados administrativos, jefes, servicios escolares, DGA

**DESCRIPCIÓN:** Los usuarios tipo coordinadores docentes, coordinadores de programa, coordinadores de UAB, tutores, secretarias tendrán acceso a los reportes por programa que sean de su departamento o centro por cada período escolar.

Los usuarios tipo encargados administrativos, jefes, servicios escolares, DGA, tendrán acceso a los reportes departamentales de todo el ITESO por cada período escolar y por cada departamento.

**FRECUENCIA DE USO:** Cada vez que necesita información de algún profesor y al final del semestre.

#### **CURSO NORMAL DE EVENTOS**

##### **ACCIONES DE ACTORES**

##### **RESPUESTA DEL SISTEMA**

1.-Doble clic en el icono del programa

2.-Presenta pantalla de Id y clave de acceso

3.-Introducir expediente y clave de acceso

4.-Validar exp y clave.

5.-Muestra menú principal.

6.-Seleccionar tipo de reporte

7.-Presenta pantalla reporte seleccionado.

8.-Seleccionar campos para realizar búsqueda

9.-Realiza búsqueda

10.-Presenta búsqueda

11.-Guarda búsqueda

12.- Pregunta Directorio

13.- Guarda archivo

14.- Regresa a pantalla de búsqueda

15.-Sale del sistema

16.- Cierra sesión usuario.

#### **CURSOS ALTERNATIVOS DE EVENTOS**

##### **ACCIONES DE ACTORES**

##### **RESPUESTAS DEL SISTEMA**

**Alternativa 1: “No quiere guardar búsqueda”**

11.- No guardar búsqueda

13.- Confirma opción

**Alternativa 2: “Imprimir reporte”**

11.- Mandar imprimir reporte

13.- Elige opción de impresión

17.- Sale del sistema

12.- Pregunta si está seguro

14.-Regresa pantalla búsqueda

12.- Presenta opciones de impresión

14.- Captura opción de impresión

15.- Manda imprimir reporte

16.- Regresa a pantalla de búsqueda.

18.- Cierra sesión usuario

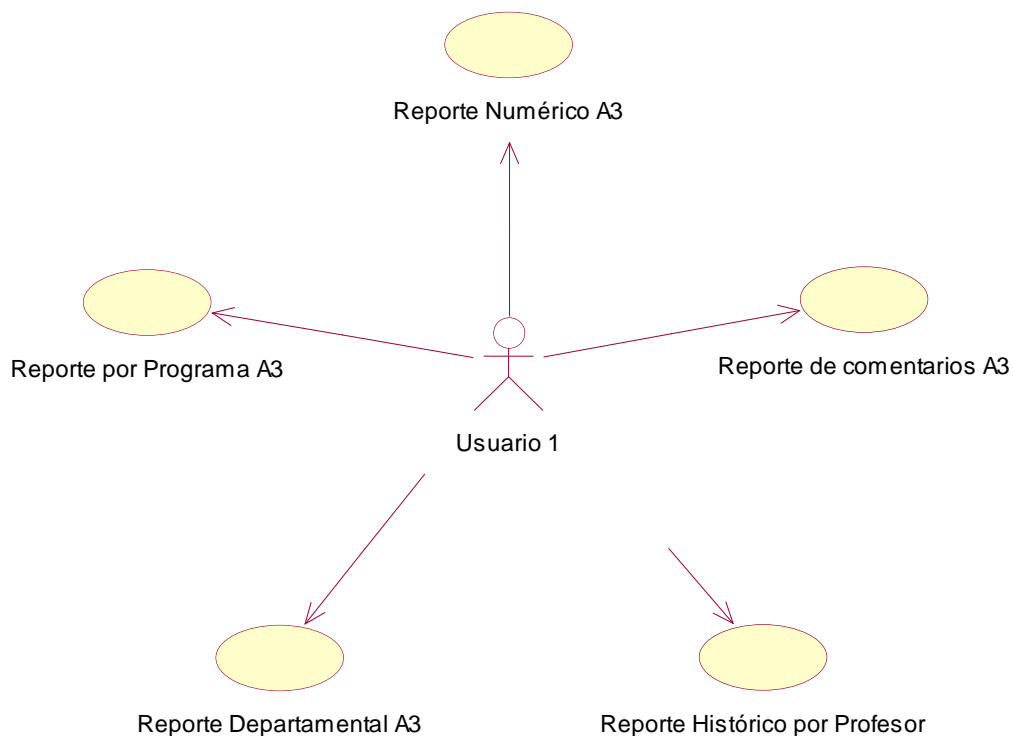
### **EXCEPCIONES**

Error en la búsqueda: el usuario se equivoca de campo y el sistema devuelve una búsqueda no deseada.

El usuario deberá regresar a la pantalla de búsqueda y volver a solicitar una búsqueda.

Error al inicio de sesión: el usuario introdujo mal su expediente o clave de acceso. El sistema mandará aviso de error y permitirá al usuario intentar de nuevo entrar al sistema.

### 3.2.5 Modelado de Casos de Uso



#### Usuario 1

Se refiere a los usuarios siguientes:

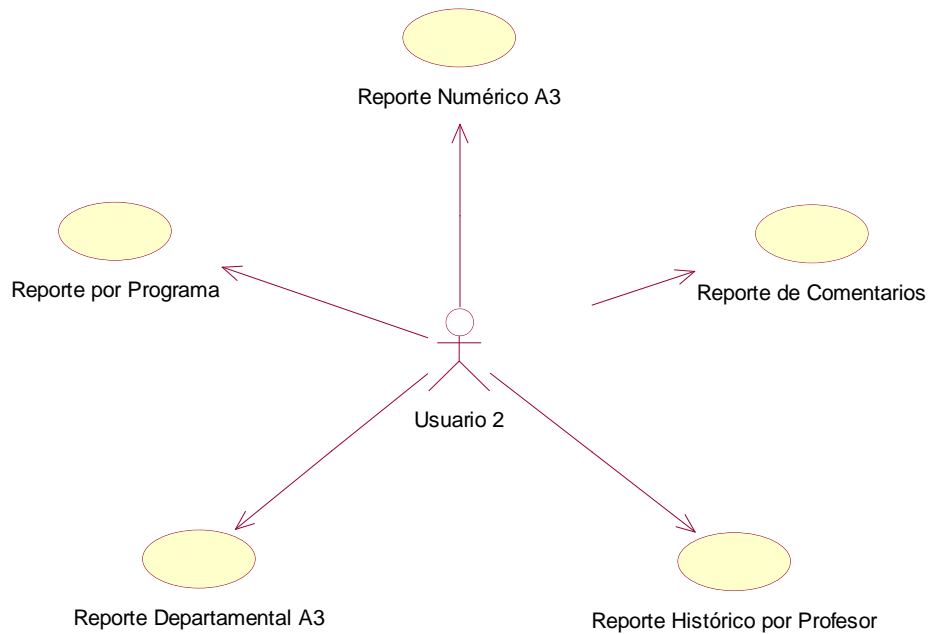
- Encargados administrativos
- Jefes
- Servicios escolares
- DGA

Este tipo de usuario no tiene ningún tipo de restricción en el sistema

Cada uno de los reportes tiene los siguientes tipos:

Los tipos están en función del periodo y del grado en el que fueron aplicados

- Reporte numérico A3: Lic 1, Lic 2, Lic 3.
- Reporte de Comentarios A3: Lic 1, Lic 2, Lic 3.
- Reporte Departamental A3: Lic 1, Lic 2, Lic 3.
- Reporte *por Programa*: Lic 1, Lic 2, Lic 3.



## Usuario 2

Se refiere a los usuarios siguientes:

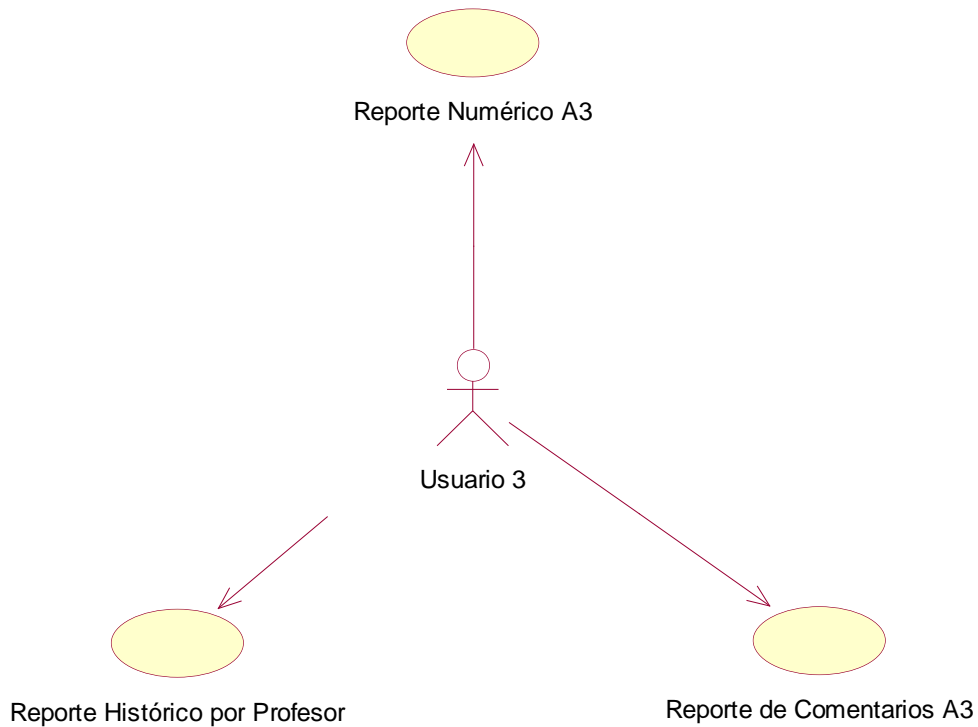
- Coordinadores Docentes
- Coordinadores de Programa
- Coordinadores de UAB
- Tutores
- Secretarias

Este tipo de usuario solamente tendrá acceso a ver solo su Departamento o Centro.

Cada uno de los reportes tiene los siguientes tipos:

Los tipos están en función del periodo y del grado en el que fueron aplicados

- *Reporte numérico A3*: Lic 1, Lic 2, Lic 3.
- *Reporte de Comentarios A3*: Lic 1, Lic 2, Lic 3.
- *Reporte Departamental A3*: Lic 1, Lic 2, Lic 3.
- *Reporte por Programa*: Lic 1, Lic 2, Lic 3.



### **Usuario 3**

Se refiere a los usuarios siguientes:

- Profesores

Este tipo de usuario tendrá restringido a consultar solamente los reportes sobre él.

Cada uno de los reportes tiene los siguientes tipos:

Los tipos están en función del periodo y del grado en el que fueron aplicados

- *Reporte numérico A3*: Lic 1, Lic 2, Lic 3.
- *Reporte de Comentarios A3*: Lic 1, Lic 2, Lic 3.

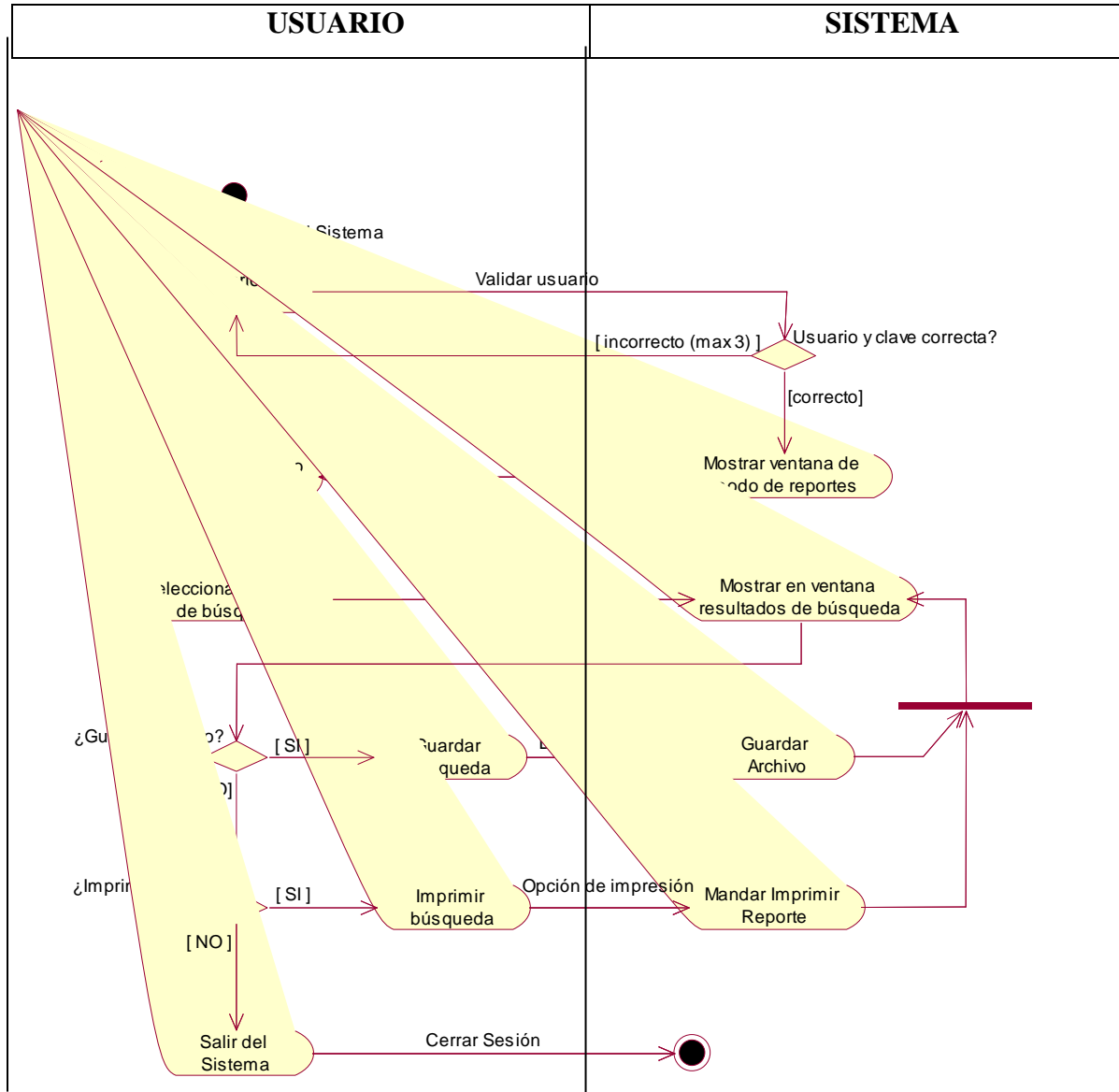


Diagrama de Actividades de los Casos de Uso

### 3.2.6 Formatos

Los formatos de los reportes del sistema así como sus descripciones, se pueden consultar en el anexo D Formatos.



### 3.3 Conclusiones.

En base a la investigación del marco teórico se definió una metodología de Ingeniería de Requerimientos, para aplicarla en el desarrollo de los requerimientos de un módulo del sistema administrativo de información de la Universidad ITESO. Se utilizaron diferentes técnicas y herramientas que se complementaron entre sí, e hicieron más claros y entendibles los requerimientos.

De acuerdo al marco teórico la IR se divide en dos: requerimientos de sistema (IRSi) y requerimientos de software (IRSw). Un sistema es el conjunto de: hardware, software, datos, gente, instalaciones y procedimientos, un requerimiento desde el punto de vista de un usuario es la condición o capacidad necesaria para el usuario para resolver un problema o alcanzar un objetivo. La diferencia entre los dos tipos de requerimientos descansa en sus orígenes, los requerimientos de sistema se obtienen directamente del usuario, transformando una necesidad operacional en una descripción, parámetros de desempeño y configuración de sistema; mientras que los de software se obtienen de la especificación de los requerimientos de sistema, dividiendo los requerimientos de sistema en subsistemas y tareas, y transformándolos a parámetros de software.

El ciclo de vida adoptado fue un modelo evolutivo pero en las fases internas se puede observar un sub-ciclo en cascada, al analizar las actividades en el tiempo, algunas están traslapadas. Los dos tipos de requerimientos siguen el mismo proceso de desarrollo: obtención, análisis, especificación y verificación, para administrar el proyecto de desarrollo se aplicaron las áreas de conocimiento del PMBOK generándose planes y formas de control del proceso. En medio de las dos fases de desarrollo de requerimientos esta la fase de de la arquitectura del sistema, en este caso ya había una existente que se complementó cuando fue necesario y a la cual se adaptó el proyecto.

Sobre el proceso de seguido en este trabajo, lo más difícil fue acotar el tema de la tesis, luego adquirir la disciplina de la investigación y poner en práctica los hallazgos, aplicar una metodología y evaluar los resultados en cada una de las etapas.

También la etapa final de la modelación de los requerimientos fue de trabajo arduo, porque cada cambio afectaba a otras partes del modelo.

Con gusto recuerdo un punto del proceso en el que de pronto se aclararon los contenidos y la estructura. El trabajo por sí solo iba pidiendo las lecturas y los temas siguientes, y estaba tan inmerso en la investigación y la aplicación, que me costó trabajo detenerme para entregar.

En cuanto a los resultados del trabajo, al aplicar la metodología de administración del proyectos de IR, se facilitó el tener el control del proyecto y evitar dejar pasar cosas por alto.

Al aplicar una metodología de trabajo participativo encontramos a muchos usuarios deseosos de colaborar, de ser escuchados y tomados en cuenta, sus aportaciones fueron muy valiosas.

Se hizo una selección de personas para obtener los requerimientos, de esta lista se entrevistaron algunas personas hasta que los resultados obtenidos fueron repetitivos y no aportaban nueva información para los casos.

La estrategia para el manejo de los requerimientos fue dejar los básicos o prioritarios y posibilitar al usuario para obtener la información más fina que habían solicitado, mediante el pos-proceso de los reportes, utilizando herramientas de Office como: Access, Word ,o Excel.

Los requerimientos prioritarios fueron establecidos por los encargados del SEDD a nivel directivo, en un proceso de toma de decisiones vertical pero cuyas propuestas partieron de la base.

El uso de herramientas de software para el modelado de los requerimientos fue de mucha ayuda, y ahorro mucho tiempo de trabajo, sobre todo con los cambios constantes conforme se iba avanzando en la concepción del proyecto.

Espero que la experiencia contribuya a mejorar nuestros procesos de Ingeniería de Requerimientos

## BIBLIOGRAFÍA

- (2001). Marco de la docencia en el ITESO: Disposiciones generales. DGA, ITESO: 5.
- (2001). Marco de la docencia en el ITESO: Elementos básicos del desempeño y la coordinación docentes. DGA, ITESO: 5.
- (2002). Evaluación del Desempeño del Personal Académico. Actividades de impartición de clases y de Coordinación Docente. DGA, ITESO: 10.
- (2003). Estatuto orgánico del ITESO: 39.
- Acuña, H. (2003). Informe del Rector 2002. Tlaquepaque, Jal., Méx., Iteso.
- Alter, S. (1999). Information Systems: a Management Perspective, Addison-Wesley.
- Alvarez, F. e. a. (1997). Evaluación del Desempeño Docente ITESO, Informe valorativo. Tlaquepaque, Jal., Méx., Iteso: 64.
- Alvarez, F. e. a. (1998). Evaluación del Desempeño Docente ITESO, Informe valorativo. Tlaquepaque, Jal., Méx., Iteso: 38.
- Alvarez, F. e. a. (1998b). Evaluación del Desempeño Docente ITESO, Informe valorativo. Tlaquepaque, Jal., Méx., Iteso: 65.
- Alvarez, F. y. C. C. (2002). Nuevo Sistema de Evaluación del desempeño Académico, "II Foro para la innovación educativa". Tlaquepaque, Jal., Méx., Iteso.
- Andriole, S. (1996). Managing Systems Requirements: Methods, Tools and Cases, McGraw - Hill.
- ANUIES (2000). Evaluación del desempeño del personal académico. Análisis y propuesta de metodolgia básica. México D.F., ANUIES.
- Boehm, B. W. a. I. H. (1996). Identifying Quality Requirements Conflicts. IEEE Software: 25-35.
- Booch, G. (1994). Object-Oriented Analysis and Design with Applications. 390 Bridge Parkway, redwood City, California 94065, The Benjamin/Cummings Publishing company, Inc.
- Booch, G. (1996). Object Solutions. Managing the object-oriented project. 2725 Sand Hill Road, Menlo Park, CA 94025, Addison-Wesley.
- Card, D. N. (2000). Sorting Out Six Sigma and the CMM. IEEE Software.
- Coad, P. a. E., Yourdon (1991). Object-Oriented Analysis. Englewood Cliffs, New Jersey 07632, Prentice-Hall.
- Coleman, D. e. a. (1994). Object - Oriented development, The fusion method, Prentice Hall.

- Collignon, M. M. (1996). El grupo y la construcción de significados a propósito de la evaluación de profesores. Departamento de Educación y Valores, Maestría en Educación. Tlaquepaque, Jal., Méx., Iteso: 157.
- Chang, C. (1996). Blueprint for the Ideal Requirements Engineer. IEEE Software.
- Demarco, T. (1979). Structured Analysis and System Specification. Englewood cliffs, New Jersey 07632, Prentice-Hall.
- Dorfman, M. e. a. (1997). Software Engineering, IEEE Computer Society.
- France, S. (1997). Evaluación de 360°. México D.F., Panorama Editorial S.A. de C.V.
- Gamma, E. e. a. (1995). Design Patterns. Elements of Reusable Object-Oriented Software. One Jacob Way, Reading, Massachusetts 01867, Addison-Wesley Publishing Company, Inc.
- Gane, C. a. S., Trish (1979). Structured Systems Analysis: tools and techniques. Englewood Cliffs, New Jersey 07632, Prentice-Hall, Inc.
- Gilmore, D. e. a. (1994). User-Centred Requirements for software Engineering Environments, NATO, ASI.
- Gunter, C. e. a. (2000). A Reference Model for Requirements and Specifications. IEEE Software.
- Gutierrez, A. (1997). Acontecer Histórico del Iteso 1957-1997, Iteso.
- Gutierrez, A. (1998). Acontecer Histórico del Iteso 1957-1997. Lo organizacional administrativo., Iteso.
- Gutierrez, A. (2000). Acontecer Histórico del Iteso 1957-1997. Lo académico desde los documentos oficiales., Iteso.
- Haag, S. C., Maeve & McCubbrey, Donald J. (2002). Management Information Systems for the Information Age. New York, McGraw-Hill.
- Hellriegel, D. S., John W. (1998). Administración. México, Thomson.
- Hernández, G. (2001). Mejoras a la Ingeniería de Requerimientos en DISA SA de CV. CUCEI. Guadalajara, Univerisidad de Guadalajara.
- Hernández, G. (2001b). Metodología para desarrollo y administración de requerimientos. Tlaquepaque, Jal., Méx., Iteso: 13.
- Hernández, G. e. a. (2003). Modelado con UML Cuaderno de Trabajo Ver. 2.2. Guadalajara, Univerisidad Iteso.
- Hernández, G. e. a. (2003b). Modelado con UML Cuaderno de Prácticas. Guadalajara, Univerisidad Iteso.
- Humphrey, W. S. (1990). Managing the Software Process, Addison - Wesley.

IBM (2003). Rational Rose family, IBM. **2003**.

IEEE610.12 (1990). IEEE Standard Glossary of Software Engineering Terminology. 345 Est 47th Street, New York 10017-2394, EUA, The Institute of electrical and Electronics Engineers, Inc.

IEEE830 (1998). IEEE Recommended Practice for Software Requirements Specifications. 345 East 47th Street, New York 10017-2394, USA, IEEE.

IEEE1233 (1998). IEEE Guide for Developing System Requeriments Specifications. 345 East 47th Street, New York, NY 10017-2394, EUA, The Institute of Electrical and Electronics Engineers, Inc.

IEEE/EIA12207.0 (1995). IEEE/EIA Standard for Information Technology- Software life cycle processes. 345 East 47th Street, New York, NY 10017-2394, EUA, The Institute of Electrical and Electronics Engineers, Inc.

IEEE/EIA12207.1 (1997). IEEE/EIA Software life cycle processes- Life cycle data. 345 East 47th Street, New York, NY 10017-2394, EUA, The Institute of Electrical and Electronics Engineers, Inc.

IEEE/EIA12207.2 (1997). IEEE/EIA Software life cycle processes- Implementation consideratios. 345 East 47th Street, New York, NY 10017-2394, EUA, The Institute of Electrical and Electronics Engineers, Inc.

Jacobson, I. e. a. (1994). The Object Advantage Business process reengineering with object technology, ACM Press.

Kendall, P. A. (1996). Introduction to systems Analysis & Design: A Structured Approach, Irwin/McGraw-Hill.

Kirakowski, J. (1997). Methods for User-Orientated Requeriments Specification, Requeriments Engineering and Specification in Telematics. **2003**.

Kovitz, B. L. (1999). Practical Software requirements, Manning Publications Co.

Larman, C. (1999). UML y Patrones. 53370 Naucalpan de Juárez, Edo. de México, Prentice Hall Hispanoamericana, S.A.

Leffingwell, D. a. W. D. (2000). Managing Software Requirements. One Lake Street, Upper Saddle River, NJ 07458, Addison Wesley.

Martin-Kniep, G. O. (1999). Portfolios del desempeño de maestros, profesores y directivos. La sabiduría de la práctica. Argentina, Paidós.

McConnell, S. (1997). Desarrollo y gestión de proyectos informáticos. Edificio Valrealty, 1ra. planta. Basauri,17. 28023 Aravaca, Madrid, España, McGraw - Hill / Interamericana de España, S.A.U.

Moore, J. W. (1998). Software Engineering Standards, IEEE Computer Society.

Morales, D. (1997). Historia de computo del Iteso. Tlaquepaque, Jal., Méx., Iteso: 10.

- Murdick, R. G. (1998). Sistemas de Información Administrativa, Prentice Hall.
- Narro, J. (2000). Agenda institucional de planeación. Tlaquepaque, Iteso.
- Nissen, H. W. e. a. (1996). Managing Multiple Requirements Perspectives with Metamodels. IEEE Software: 37-48.
- Orr, K. T. (1977). Structured Systems Development. Englewood Cliffs, New Jersey 07632, Prentice-Hall, Inc.
- Page-Jones, M. (1988). The practical guide to structured systems design. Englewood cliffs, New Jersey 07632, Prentice-Hall, Inc.
- Palmer, J. D. a. F. N. A. (1992). An Integrated Environment for Requirements Engineering. IEEE Software: 80-85.
- Paulk, M. C. e. a. (1994). The Capability Maturity Model, Carnegui Mellon University.
- Pinheiro, F. A. C. a. G. J. A. (1996). An Object-Oriented Tool for Tracing Requirements. IEEE Software: 52-64.
- PMI (2000). A Guide to the Project Management Body of Knowledge. Four Campus Boulevard, Newton Square, PA 19073-3299 USA, Project Management Institut.
- RAE (1999). Diccionario de la Lengua Española, Espasa Calpe S.A.
- Reifer, D. J. (2000). Requirements Management: The Search for Nirvana. IEEE Software.
- Reynaga, S. (1999). Competencias... ¿Hacia la construcción de un nuevo mito? Tlaquepaque, Jal., México.: 17.
- Rueda, M. y. F. D. (2000). Evaluación de la docencia. Perspectivas actuales. México, Paidós.
- Rumbauch, J. e. a. (1999). El Lenguaje Unificado de Modelado, Manual de referencia. Madrid España, Addison Wesley.
- Sawyer, P. S. I. a. V. S. (1999). Capturing the Benefits of Requirements Engineering. IEEE Software: 78-85.
- Senn, J. A. (1990). Sistemas de información para la administración, Grupo Editorial Iberoamericana.
- Siddiqui, J. (1994). Challenging Universal Truths of Requeriments Engineering. IEEE Software.
- Siddiqui, J. a. H. S. (1996). Requirements Engineering: The Emerging Wisdom. IEEE Software: 15-19.
- Sodhi, J. (1992). Software Requirements Analysis and Specification, McGraw - Hill, Inc.
- Sommerville, I. (2002). Ingeniería de Software, Addison Wesley.

- Sommerville, I. S. P. (1997). Requirements Engineering, a good practice guide. Baffins Lane, Chichester, West Sussex PO 19 1UD, England, John Wiley & Sons Ltd.
- SSE (2003). Informe de Indicadores Académico - Escolares. Tlaquepaque, Jal. , Méx., Iteso: 53.
- Stair, R. M. R., George W. (2000). Principios de Sistemas de Información. México, Thomson.
- Thayer, R. H. e. a. (1997). Software Engineering Project Management, IEEE Computer Society.
- Thayer, R. H. e. a. (1997). Software Requirements Engineering, IEEE Computer Society.
- Towfighian, B. (1998). Análisis de factores que intervienen en la implementación de un sistema de información administrativo. Maestría en administración de sistemas de información. Zapopan, Jal., Instituto Tecnológico y Estudios Superiores de Monterrey: 145.
- Vázquez, A. (1995). La reforma oranzativa del Iteso a partir de 1995. Tlaquepaque, Iteso.
- Westerman, H. R. (2001). Systems engineering principles and practice, Artec house, Inc.
- Whitten, N. (1995). Managing Sftware Development Projects, John Wiley & Sons, Inc.
- Wieggers, K. E. (1999). Software Requirements. Redmond, Washington, EUA., Microsoft Press.
- Wienberg, V. (1978). Structured Analysis. New York, N.Y., Yourdon Press.
- Winblad, A. L. e. a. (1993). Software orientado a objetos, Addison-Weley Iberoamericana, S.A.
- Yourdon, E. (1988). Managing the System Life Cycle. Englewood Cliffs, New Jersey 07632, Yourdon Press.
- Yourdon, E. (1989). Modern Structured Analysis. Englewood Cliffs, New Jersey 07632, Yourdon Press.
- Yourdon, E. (1994). Object-Oriented Systems Design an Integrated Approach. Englewood Cliffs, New Jersey 07632, Prentice-Hall, Inc.
- Yourdon, E. a. C., Larry L. (1979). Structured Design. Fundamentas of a Discipline of Computer Program and Systems Design. Englewood Cliffs, N.J. 07632, Prentice-Hall, Inc.
- Zahran, S. (1998). Software Porcess Improvement. Edinburg Gate; Harlow, Essex CM20 2JE, England, Addison Wesley Longman.

**ANEXOS**