



TESIS DE MASTER

Sistemas Experto de Ayuda para la Selección del Modelo de Ciclo de Vida

Autor: Lic. Bibiana D. Rossi

Dirigida por:

DRA. Almudena Sierra
Dr. Ramón García Martínez

Madrid, Buenos Aires, Septiembre 2001

A mis hijos

Gastón La Valle y Verónica La Valle

AGRADECIMIENTOS

A las actuales autoridades del *Instituto Tecnológico de Buenos Aires*, República Argentina,
por llevar adelante una política consistente en programas de formación en RR.HH

A la *Universidad Tecnológica Nacional, Facultad Regional Buenos Aires*, República Argentina
por ser mi primer Alma Mater en donde aprendí a amar mi profesión

A la *Universidad Nacional de Luján*, República Argentina
por ser mi segunda Alma Mater en donde encontré una vía de crecimiento formativo

A la *Facultad de Informática de la Universidad Politécnica de Madrid*
y al apoyo prestado especialmente por Natalia Juristo Juzgado, directora del Magíster

Al Dr. *Edmundo Noé Gramajo*
por iniciar el proyecto y por confiar en mi desempeño para participar en él

A mi primer maestro y jefe Lic. *Carlos A. Uhalde*
que acompañó y fortaleció mi crecimiento profesional

A mi directora Dra. *Almudena Sierra*, por el tiempo dedicado,
la claridad y buena predisposición ante las consultas realizadas

A mi director Dr. *Ramón García Martínez*, por su especial dedicación,
con la que permanentemente abona mi crecimiento profesional

Al equipo de expertos del dominio, Prof. *Gregorio Perichinsky*,
Lic. *Carlos Beltrami*, Lic. *Carlos Leone*, Ing. M. *Florencia Pollo Cattaneo*,
Lic. *Laura Lucchini*, Ing. *Mariano Weschler*, Lic. *Enrique Fernández*
Por el compromiso que asumieron, para que el proyecto llegara a buen término

A mi hermana, amiga, colega y socia Lic. *Paola Britos*
por su amor, tolerancia, apoyo logístico y continua compañía

A mis alumnos Ing. *Alberto Patrón* e Ing. *Sebastián Corbat*
por su afecto y continuo apoyo en el dominio de la herramienta Kappa.

A los alumnos Carlos Coto, Andrea Cribelli y Santiago Gallo, por su colaboración

A todos mis alumnos del Magíster quienes
alentaron, compartieron y enriquecieron mi propio trabajo de tesis

A mis amigos Lic. *Adriana Fachal*, A.S. *Marcelo Castro*, Osvaldo Rapetti,
Daniel Castro, Luis Donadío
quienes con su cariño y aliento acompañaron el avance de este trabajo

A mis compañeras de trabajo Cecilia Cardozo y Carolina Procopio, por su paciencia
y colaboración permanente en la tarea diaria y en este trabajo en particular

A mi ahijado Maxi, por regalarme sus sonrisas, abrazos, travesuras y amor

A mis hijos Verónica La Valle y Gastón La Valle
por su paciencia, tolerancia, compañía, comprensión, apoyo y su gran amor

INDICE**Capítulo 1.**

1. Introducción	11
-----------------	----

Capítulo 2.

2. Dominio de la Aplicación	17
2.1 Introducción	17
2.2 Concepto de Ciclo de Vida	19
2.3 Ciclo de vida en Cascada	20
2.4 Ciclo de vida en Espiral	26
2.4.1 Concepto de Riesgo	26
2.4.2 Fases del ciclo de vida en Espiral	28
2.5 Ciclo de vida Orientado a objetos	31
2.5.1 Conceptos básicos	32
2.5.2 Fases del ciclo de vida Orientado a objetos	35
2.5.3 Modelos básicos del CV Orientado a objetos	42
2.5.3.1 Modelo de objetos	43
2.5.3.2 Modelo dinámico	44
2.5.3.3 Modelo funcional	45

Capítulo 3.

3. Definición del problema	49
3.1 Planteamiento	49
3.2 Objetivo del trabajo	50
3.3 Alcance	50
3.4 Ambito del proyecto	50
3.5 Metodología IDEAL	52
3.6 Gestión del Proyecto	56
3.6.1 Planificación	56
3.6.2 Estimación de recursos	57
3.6.3 Control de Configuración	57
3.6.4 Evaluación del proyecto	57

Capítulo 4.

4. Estudio de Viabilidad	65
4.1 Test de Viabilidad	65

4.2 Análisis de Viabilidad para el proyecto	68
4.2.1 Justificación de la dimensión Plausibilidad	68
4.2.2 Justificación de la dimensión Justificación	69
4.2.3 Justificación de la dimensión Adecuación	71
4.2.4 Justificación de la dimensión Exito	73
4.3 Cálculo de Viabilidad para el proyecto	77
4.4 Evaluación de Viabilidad para el proyecto	87

Capítulo 5.

5. Adquisición de Conocimientos	91
5.1 El proceso de Adquisición de Conocimientos	91
5.2 Adquisición de Conocimientos del presente trabajo	92
5.3 Técnicas para Adquisición de Conocimientos utilizadas en el presente trabajo	94
5.4 Adquisición de conocimientos a partir de un Equipo de expertos	95
Sesión C.1	95
Sesión C.2	118
Sesión C.3	128
5.5 Teoría de la Construcción Personal o Emparrillado	136
Sesión D.1	136
Sesión A.4	142
5.5 Conclusión de la Fase Primaria de Adquisición de Conocimientos	145

Capítulo 6.

6. Conceptualización de Conocimientos	149
6.1 Proceso de Conceptualización	149
6.2 Análisis de Conocimientos	150
6.2.1 Paso 1: Identificación, comparación y categorización de conceptos	151
6.2.2 Paso 2: Identificación de las relaciones entre conceptos	166
6.2.3 Paso 3: Identificación de los conocimientos estratégicos	167
6.2.3.1: Descomposición funcional modular	167
6.2.3.2: Comprobación de los conocimientos estratégicos	180
6.2.4: Paso 4: Identificación de los conocimientos tácticos	180
6.2.4.1: Análisis de los conocimientos tácticos, seudorreglas	180
6.2.4.2: Comprobación de los conocimientos tácticos	215

6.2.5 Paso 5: Identificación de los conocimientos fácticos	215
6.2.5.1: Análisis de los conocimientos fácticos	216
6.2.5.2: Comprobación de los conocimientos fácticos	252
6.3 Síntesis de conocimientos	252
6.3.1 Modelo de procesos, modelo dinámico	252
6.3.2 Modelo estático	253
6.3.3 Mapa de Conocimiento	267
6.4 Comprobación de la Conceptualización	268

Capítulo 7.

7. Formalización de Conocimientos	281
7.1 Selección de Formalismos	281
7.2 Formalización de los Conocimientos en Reglas de Producción	281
7.3 Formalización de los Conocimientos en Marcos	281
7.4 Formalización de los Conocimientos en Procedimientos	293

Capítulo 8.

8. Implementación del sistema	299
8.1 Selección de la Herramienta	299
8.1.1 Los Objetos en Kappa	299
8.1.2 Las Reglas en Kappa	300
8.2 Implementación en la Herramienta	301
8.2.1 Estructura general de los Objetos	302
8.3 Desarrollo de un caso ejemplo	304

Capítulo 9.

9. Evaluación	325
9.1 Evaluación de la Adquisición de Conocimientos	325
9.2 Evaluación de la Conceptualización de Conocimientos	326
9.3 Evaluación de la Formalización de Conocimientos	326
9.4 Evaluación de la Implementación de Conocimientos	327
9.4.1 Caso Ejemplo 1.1: Facturación Telefónica	327
9.4.2 Caso Ejemplo 1.2: Facturación Telefónica	332
9.4.3 Caso Ejemplo 2: Control de Stock	339

9.4.4 Caso Ejemplo 3: Control de Morosos	344
9.4.5 Caso Ejemplo 4: Técnica emparrillado	349
9.4.6 Caso Ejemplo 5: Inscripción Universitaria	355
9.4.7 Caso Ejemplo 6: Organizar Comisiones	364
9.4.8 Caso Ejemplo 7: Administración Consorcio	370
9.4.9 Caso Ejemplo 8: Registración Contable	374
9.4.10 Caso Ejemplo 9: Portal Empleos	377
9.4.11 Caso Ejemplo 10: Pedidos Drogería	383
9.5 Resultados de la Evaluación del Sistema Experto	390

Capítulo 10.

10. Conclusiones y Futuras Líneas de Investigación	395
10.1 Conclusiones de trabajo	395
10.2 Líneas de Investigación y Desarrollo	396
10.3 Líneas de Investigación Complementarias	397

Capítulo 11

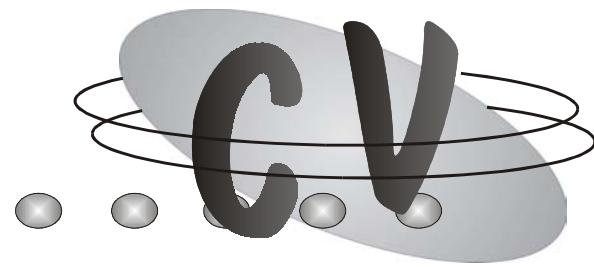
11. Bibliografía	401
11.1 Referencias bibliográficas	401
11.2 Abreviaturas	403

Capítulo 12

12. Anexos	407
12.1 Anexos del capítulo 3: Gestión de Configuración	407
12.2 Anexos del capítulo 5: Sesiones de Adquisición de Conocimientos	419
A.1-A.2-A.3 Sesiones de entrevistas con el experto	419
B.1 Sesión de extracción de conocimientos	429

Capítulo 1

Introducción



En los últimos años el proceso software se ha presentado como el eje central en la escala de los niveles propuestos por el Modelo de Madurez del SEI (Software Engineering Institute), [Paulk, M. 1993]. Como punto de partida frente a la resolución de problemas reales de software, el ingeniero debe establecer una estrategia de desarrollo que acompañe al proceso. Esta estrategia implica seleccionar un modelo de proceso (ciclo de vida) según la naturaleza del proyecto, de la aplicación a desarrollar, de los métodos y herramientas a utilizar y los controles y entregas que se requieren.

Existen varios modelos de ciclo de vida, entre los que se destacan: cascada, prototipado de usar y tirar, incremental, emisión gradual, mejora iterativa, ensamblaje de componentes, espiral, prototipado operativo y prototipado rápido. No existe un modelo de ciclo de vida que funcione para cualquier proyecto.

El Ingeniero en software debe estudiar las características del proyecto y seleccionar el modelo de ciclo de vida que más se adapte a ellas. Las bases para determinar el ciclo de vida más adecuado son: la cultura de la organización, la disponibilidad para correr riesgos, el dominio de la aplicación, la volatilidad de los requisitos y cuánto se comprenden dichos requisitos.

Esta selección se realiza al inicio del proyecto y se deben considerar un conjunto de variables de análisis que requiere en su mayor parte de la experiencia del ingeniero en software, ya que al inicio del proyecto subyace una gran cuota de subjetividad en casi todos los aspectos del proyecto.

En este contexto, se ha considerado conveniente disponer de la asistencia de un Sistema Experto que colabore en la selección del modelo de ciclo de vida más adecuado para el proyecto. El sistema asistirá al Ingeniero en software en la selección del ciclo de vida más adecuado para un proyecto de desarrollo en particular, colaborando en el análisis de las características más relevantes a tener en cuenta.

Se observa que el conocimiento requerido para confeccionar el Sistema Experto es de laboriosa obtención. Sin embargo, vista la importancia del tema, se propone considerar la solución postulando una clasificación de ciclos de vida en familias genéricas. Las que fueron consideradas para el presente trabajo son: cascada, espiral y orientadas a objetos. En este contexto se propone encuadrar el desarrollo teniendo en cuenta los aspectos más sobresalientes de cada familia.

Este trabajo esta organizado en capítulos cuyo contenido es el siguiente:

- En el capítulo II, Domino del problema, se describe el dominio de aplicación del Proyecto. Se presenta el concepto de Ciclo de Vida (CV), su importancia y vigencia. Luego se describen los ciclos de vida que se consideran en este proyecto: el CV en Cascada, el CV en espiral y el CV Orientado a objetos. Del CV en Cascada se describen sus fases. Del CV en Espiral se explica uno de los conceptos principales de ese ciclo, que es el riesgo, y se describen las fases que lo constituyen. Del CV Orientado a objetos se hace una breve introducción de los conceptos básicos, se describen las fases y se presentan los modelos básicos que se usan.
- En el capítulo III, Definición del problema, se presenta el objetivo y alcance del proyecto especificando las características del problema a resolver y los participantes. Se describe la metodología IDEAL que es la seleccionada para desarrollar el sistema. Como cierre del capítulo se presenta un plan del proyecto que es la base sobre la cual se cumplen las actividades de gestión.
- En el capítulo IV, Estudio de Viabilidad, se describe el Test de Viabilidad que propone la Metodología IDEAL. Luego se detalla el análisis de viabilidad para el proyecto, justificando los valores definidos en el test. A continuación se evalúa la viabilidad del proyecto.
- En el capítulo V, Adquisición del Conocimiento se describe el proceso de adquisición de conocimientos y las técnicas usadas en el mismo. Como inicio del proceso se presentan las primeras sesiones de entrevistas con el experto principal para completar el estudio de viabilidad. Luego se documenta el proceso de extracción de conocimientos en el cual se utilizó el Análisis Estructural de Textos. Se detalla el proceso de educación de conocimientos realizado al equipo de expertos para el cual se aplicaron el Método Delphi y la Técnica Nominal de Grupo. Y finalmente se presenta la educación de conocimientos con el experto principal, el desarrollo de las entrevistas y la técnica de Emparrillado.
- En el capítulo VI, Conceptualización de Conocimientos se presenta el resultado del desarrollo de la fase de Conceptualización. En primer lugar se documenta el resultado del Análisis de los conocimientos adquiridos organizados en Conocimientos Estratégicos, Tácticos y Fácticos. Luego se muestra el resultado de la Síntesis de los conocimientos adquiridos y analizados documentado en el Modelo estático, el Modelo dinámico y la conformación final en el Mapa de Conocimientos. Como cierre del capítulo se presenta la Comprobación del proceso de Conceptualización.

- En el capítulo VII, Formalización de Conocimientos se especifica cómo los conocimientos que en la fase de Conceptualización fueron expresados en representaciones externas intermedias: árbol de descomposición funcional, seudorreglas y tablas de concepto-atributo-valor son representados en representaciones compatibles con la tecnología computacional. Se seleccionan los formalismos a utilizar. Luego se formalizan los conocimientos que serán posteriormente implementados en la herramienta seleccionada.
- En el capítulo VIII, Implementación del sistema se describen en primer lugar las características de la herramienta que se usará para implementar el sistema experto. Luego usando un caso como ejemplo se presenta el sistema en sí, la interfaz y su operatoria general.
- En el capítulo IX, Evaluación, se presentan los casos de prueba con los que se ha evaluado el sistema. Se documenta la verificación y validación que se ha realizado y se mencionan los ítem que documentan la evaluación realizada a lo largo del desarrollo del sistema.

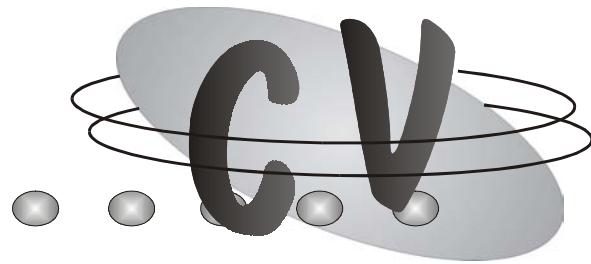
El objetivo de la etapa de evaluación es garantizar la calidad del sistema experto. La calidad esta asociada con el funcionamiento correcto del sistema y que el sistema responda a las expectativas del usuario. La evaluación no es, una fase concreta de la ingeniería del conocimiento [Gómez, A. y otros 1997] sino un conjunto de actividades que ser realizaran a lo largo de cada fase de desarrollo del sistema. Cada fase del proceso de desarrollo requiere una evaluación diferente aunque es conveniente utilizar los mismos casos de prueba a lo largo de todas las fases.

- En el capítulo X, Conclusiones finales y Futuras Líneas de Investigación, se consideran tres aspectos importantes en relación al trabajo realizado y a su continuidad. En la parte de Conclusiones se detallan los aportes originales que el trabajo de tesis ha generado en el dominio de conocimiento de la Selección del Modelo de Ciclo de Vida más adecuado. En la segunda parte se mencionan aspectos y líneas de investigación que se pueden considerar para ampliar el presente desarrollo. En la tercera parte se mencionan temas de investigación complementarios.
- En el capítulo XI, Bibliografía, se presenta la bibliografía que se ha usado en el presente trabajo y el detalle de las abreviaturas.
- En el capítulo XII, Anexos, se detallan los anexos. En primer lugar se presentan los anexos del capítulo 3, Definición del problema: el control de configuración. En segundo lugar se presentan los anexos del capítulo 5, Adquisición de Conocimientos: las sesiones de entrevista realizadas al

experto y las sesiones de extracción de conocimientos realizadas sobre el análisis de la bibliografía del dominio de aplicación del presente trabajo.

Capítulo 2

Dominio de la Aplicación



En este capítulo se describe brevemente el dominio de aplicación del Proyecto. En primer lugar se presenta el concepto de Ciclo de Vida (CV), su importancia y vigencia. Luego se describen los ciclos de vida que se consideran en este proyecto el CV en Cascada, el CV en Espiral y el CV Orientado a objetos.

Del CV en Cascada se describen sus fases. Del CV en Espiral se explica uno de los conceptos principales de ese ciclo, que es el riesgo, y se describen las fases que lo constituyen. Del CV Orientado a objetos se hace una breve introducción de los conceptos básicos, se describen las fases y se presentan los modelos básicos que se usan.

2.1 INTRODUCCION

Todos los sistemas de información están conformados por componentes estructurales: datos, procesos, controles, entradas, salidas, modelos, tecnología. A estos componentes se le suman los requerimientos: de sistemas, de procesamiento de datos, de integración, de costo y eficacia, de factibilidad. También se suman variables del negocio como: capacidad competitiva, calidad y utilidad de la información, factores organizacionales, factores humanos.

Todos estos elementos no hacen un sistema de información a menos que conformen una unidad con un propósito determinado. La tarea de desarrollar un sistema de información es emprendida por los profesionales de sistemas que emplean una metodología de desarrollo de sistemas, como el camino para realizar su trabajo. La metodología es una guía desde el análisis hasta el mantenimiento. El primer paso de la tarea consiste en definir el modelo de ciclo de vida, es decir, qué camino guía se selecciona.

Existen varios tipos de ciclos de vida entre los cuales seleccionar. Es natural que existan varios tipos de ciclos de vida ya que existe también una gran variedad de aplicaciones para las cuales se construyen productos software con diversas características particulares.

Las técnicas y metodologías de desarrollo de sistemas se confunden con frecuencia con el ciclo de vida [Whitten, J. 1996]. El propósito del ciclo de vida es planear, ejecutar y controlar el proyecto de desarrollo de un sistema. El ciclo de vida define las fases y las tareas esenciales para el desarrollo de sistemas, sin importar el tipo o la envergadura del sistema que se intenta construir. Por ejemplo, siempre se debe estudiar y analizar el sistema actual (en el grado de detalle adecuado) antes de definir las necesidades de los usuarios y fijar las prioridades.

Algunos autores y profesionales presentan el concepto de ciclo de vida como parte del pasado, que ha perdido vigencia en la actualidad frente a las nuevas filosofías de análisis y diseño de sistemas de software. El mito de la muerte del ciclo de vida se relaciona con la popularidad de las técnicas y metodologías [Whitten, J. 1996].

Una técnica es un método que aplica herramientas y reglas específicas para completar una o más actividades del ciclo de vida del desarrollo de sistemas. Las técnicas en su mayoría son solo aplicables a una parte del ciclo de vida. Por ejemplo la programación estructurada es una técnica aplicable a las fases de codificación y soporte de sistemas. No da apoyo a las fases de planificación, análisis o diseño. Por lo tanto es necesario combinar la programación estructurada con otras técnicas de desarrollo para cubrir todas las fases del ciclo de vida.

Una metodología es una versión particular e individual de un ciclo de vida completo para el desarrollo de sistemas [Burch, J. y Grudnitski, G. 1994], que incluye:

- Tareas paso a paso para cada fase. Determina el orden de las fases del proceso software.
- Funciones desempeñadas en cada tarea.
- Productos resultantes y normas de calidad para cada tarea.
- Técnicas de desarrollo que se utilizarán en cada tarea.

Una auténtica metodología debe acompañar al ciclo de vida completo del desarrollo de sistemas. La mayor parte de las metodologías modernas incluye el uso de varias técnicas de desarrollo con sus herramientas asociadas. Los términos técnica y metodología suelen utilizarse de forma indistinta tanto en las publicaciones informáticas como en la industria. Con frecuencia se llama metodología al análisis estructurado cuando, en realidad, es una técnica. Dicha técnica forma parte de varias metodologías conocidas [Whitten, J. 1996].

Así pues, puede afirmarse que el ciclo de vida sigue vigente. Simplemente se ha confundido con la forma de populares metodologías disponibles comercialmente, las cuales contienen técnicas muy conocidas y aplicadas.

Independientemente del número o nombres de las fases del ciclo de vida, éste racionaliza y asigna una rutina al proceso de construcción de sistemas de información. La meta principal del ciclo de vida de sistemas es reducir los inicios

falsos, reclamamiento indebido y callejones sin salida. Además aumenta la probabilidad de que el sistema que se construya e instale finalmente, sea el que los usuarios desean y necesitan.

2.2 CONCEPTO DE CICLO DE VIDA

El proceso de construcción de software puede verse como una cadena de tareas. Las cadenas de tareas son planes idealizados de qué acciones deben realizarse y en qué orden. El software obtenido tras el proceso puede ser visto como el “producto” que entra al proceso, se transforma (a lo largo de la cadena de tareas) y que sale del proceso hasta obtener el producto deseado. Desde esta perspectiva del producto, se pueden establecer los estados por los que va pasando el producto en un proceso software: la entrada al proceso es una necesidad, que una vez estudiada se convierte en una especificación de requisitos, que posteriormente se transforma en un diseño del sistema, para pasar más adelante a ser un código y finalmente un sistema software completo e integrado. Este enfoque orientado al producto, focalizado en el producto transformado (en lugar del proceso que lo transforma) se llama ciclo de vida. Es decir, el ciclo que el producto software sufre a lo largo de su vida, desde que nace (o se detecta la necesidad) hasta que muere (o se retira el sistema). [Juristo Juzgado, N. b 1996].

La figura 2-1 muestra la relación entre el proceso de construcción del software y el ciclo de vida.

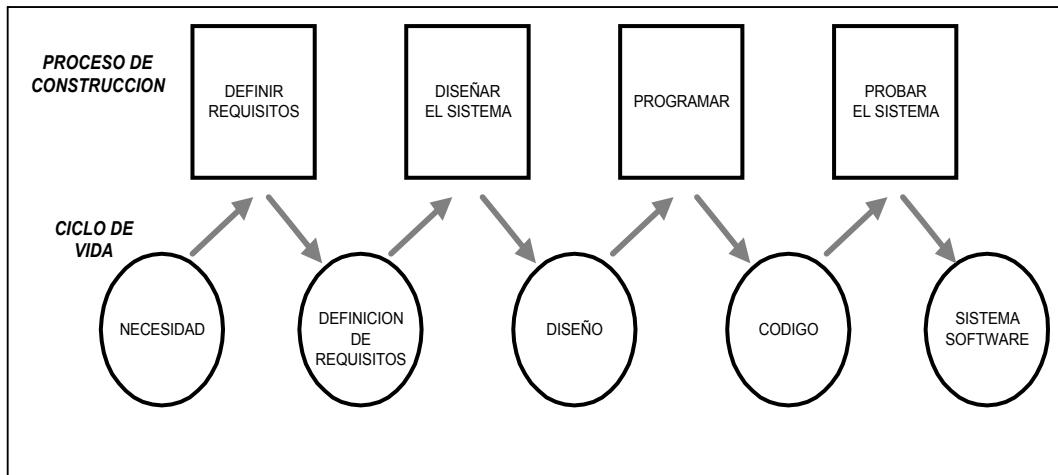


FIGURA 2-1: Relación entre Proceso de construcción de software y el Ciclo de Vida.

El cambio de perspectiva del proceso al producto es muy sutil. Es más, proceso y producto, producto y proceso, están intrínsecamente relacionados, son

inseparables. Cuando la perspectiva se centra en el proceso, el objetivo se focaliza en resolver un caso del mundo real, el proceso refleja como se usa la experiencia humana en la construcción de software y cómo se aplica a un dominio concreto. Cuando la perspectiva se centra en el producto, el objetivo se focaliza en detallar el conjunto específico de productos (documentos) que se requieren en cada paso del proceso de desarrollo de software.

La norma IEEE 1074 (Estándar IEEE del Ciclo de Vida para el Proceso de Desarrollo de Software) [IEEE, 1991] define ciclo de vida como: “*una aproximación lógica de la adquisición, el suministro, el desarrollo, la explotación y el mantenimiento del software*”.

La norma ISO 12207-1 (Proceso del Ciclo de Vida del Software) [ISO, 1994], define ciclo de vida como: “*un marco de referencia, que contiene los procesos, las actividades y tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto software, abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso*”.

2.3 CICLO DE VIDA EN CASCADA

Fue presentado por primera vez por Royce en 1970 [Royce, W.W. 1970]. En este CV la evolución del producto software procede a través de una secuencia ordenada de transiciones de una fase a la siguiente según un orden lineal. Se presenta frecuentemente como un simple modelo con forma de cascada como muestra la Figura 2-2.

A continuación se presenta una breve descripción de cada una de sus fases :

Planificación de sistemas: el ámbito de la planificación puede ser toda la empresa, una división de la misma o cualquier otro tipo de sus unidades organizativas. El propósito es identificar y establecer las prioridades sobre aquellas aplicaciones de los sistemas de información cuyo desarrollo reporte máximos beneficios para la empresa considerada en su conjunto. Sus entradas son las “misiones de la empresa” y cualquier tipo de “detalle o limitación de los sistemas existentes”. Sus salidas son los “planes de sistemas de empresa y de información” y “los proyectos planificados de desarrollo de aplicaciones”. Es así como el proyecto planificado avanza hacia la fase siguiente.

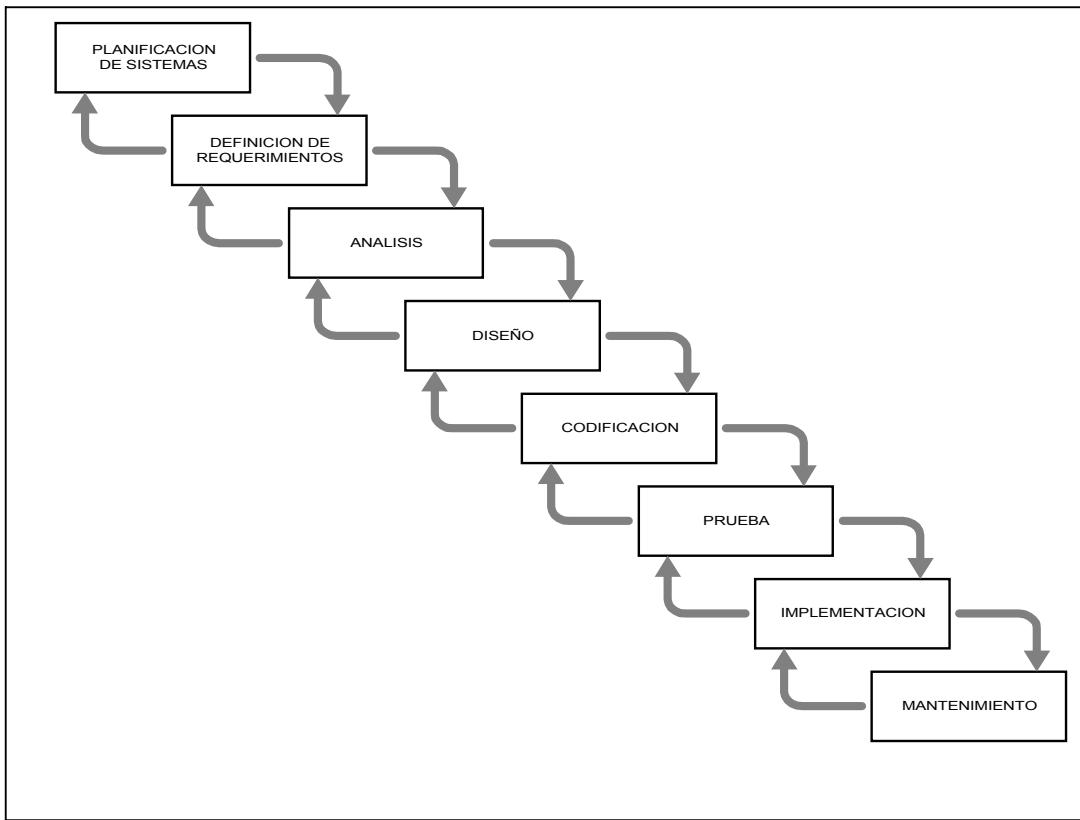


FIGURA 2-2: Ciclo de Vida en Cascada

Definición de Requerimientos: el ámbito de la definición de requisitos es un único proyecto de sistemas de información. Su propósito es analizar el problema o la situación de la empresa y definir las necesidades con respecto a la creación o perfeccionamiento de un sistema de información. Las entradas posibles son:

- El proyecto planificado de desarrollo de aplicaciones (procedente de la fase anterior)
- Un proyecto no planificado de desarrollo de aplicaciones (que responde a un problema, una oportunidad o una norma imprevista)
- Detalles y limitaciones de los sistemas existentes.
- Hechos y necesidades relacionados con la empresa.

El producto de esta fase es un Informe de Requisitos en el que se detalla lo que precisan los usuarios y no cómo se proyectan, diseñan o implementan dichas necesidades.

Análisis: el dominio de aplicación es un único proyecto de desarrollo. Consiste en el estudio del sistema actual de la empresa y establecer prioridades en las necesidades de información manifestadas por los usuarios para la construcción del nuevo sistema de información. Entre sus

principales actividades está el estudio de la viabilidad del proyecto, el estudio y modelado del actual sistema en funcionamiento, la definición de los requisitos y establecimiento de prioridades y la especificación de las posibles alternativas de solución.

El proceso de análisis se centra especialmente en el dominio del problema a resolver. Para comprender la naturaleza del sistema a construir el analista del software debe comprender el dominio de información, la función requerida, el comportamiento, el rendimiento y la interconexión. Los requisitos, se documentan y se revisan con el cliente.

Las entradas son el Plan del proyecto y el Informe de requerimientos.

Las salidas son el Estudio de viabilidad, y el Informe del sistema actual.

Diseño: el proceso de diseño traduce requisitos en una representación centrada en la computadora que se pueda evaluar antes de que comience la generación de código. El diseño también se documenta y forma parte del sistema de configuración.

El diseño de software es un proceso complejo que se centra en:

- La estructura de los datos. Se definen archivos y bases de datos. Es habitual definir un diagrama de estructuras de datos.
- Los métodos y procedimientos de proceso (algoritmos). Es habitual definir diagramas de flujos.
- La arquitectura del software, la estructura de la red informática.
- Representaciones de interfaz. Diseño de entradas, salidas, pantallas.

Codificación: el diseño se debe traducir en una forma legible por la máquina. El paso de generación de código lleva a cabo esta tarea. Si se lleva a cabo el diseño de una forma detallada, la generación de código se realiza con mayor facilidad. Esta fase puede incluir también las actividades de construcción de las redes y bases de datos a usar por el nuevo sistema.

La entrada a esta fase son las Especificaciones de diseño. El resultado de esta fase son los programas informáticos no instalados aún para producción.

Prueba: generado el código, comienzan las pruebas del programa, para encontrar errores en el software. El proceso de prueba se centra en los procesos lógicos internos del software, asegurando que todas las sentencias estén probadas y en las funciones externas, buscando que la entrada definida produzca resultados reales de acuerdo con los resultados requeridos.

Las pruebas de unidades aseguran que los programas de aplicaciones funcionen de forma adecuada cuando se prueban de forma aislada con

respecto a otros programas de aplicación. La entrada es la especificación de procesos y la especificación del programa

Las pruebas de sistemas aseguran que los programas de aplicaciones escritos de forma aislada funcionen adecuadamente cuando se integran en el sistema global. La entrada es la especificación de requerimientos del sistema

La salida de esta fase es el sistema instalado y probado.

Implementación: se desarrollan todas las actividades necesarias para poner operativo el sistema desarrollado, retirar el sistema en uso (si lo hubiera) y capacitar a los usuarios.

Las actividades consisten en procurar una transición suave entre el antiguo sistema y el nuevo, ayudar a los usuarios a resolver los problemas normales de arranque, escribir manuales de operación, cargar los archivos y las bases de datos.

Las entradas de esta fase son: el sistema instalado en la fase anterior, la documentación para el usuario final, la formación del usuario final.

La salida es el sistema de información en producción.

Mantenimiento: el software sufrirá cambios después de ser entregado al cliente. Se producen cambios porque se han encontrado errores, porque el software debe adaptarse para acoplarse a los cambios en su entorno externo, porque se requiere un cambio por un software de base o un hardware nuevo, o porque el cliente requiere de mejoras funcionales o de rendimiento.

El mantenimiento vuelve a aplicar cada una de las fases a un sistema o subsistema ya existente o a uno nuevo.

El modelo de ciclo de vida en Cascada, permite iteraciones durante el desarrollo, ya sea dentro de una misma fase, ya sea de un estado hacia otro anterior. La mayor iteración se produce cuando una vez terminado el desarrollo y cuando se ha visto el software producido se decide comenzar de nuevo y redefinir los requisitos del usuario. A menudo, durante el desarrollo, se pueden tomar decisiones que den lugar a diferentes alternativas. El modelo en cascada no reconoce esta situación. Por ejemplo, dependiendo del análisis de requisitos se puede implementar el sistema desde cero o adoptar uno ya existente, o comprar un paquete que proporcione las funcionalidades requeridas.

Entre las características del modelo en Cascada pueden mencionarse:

- Cada fase empieza cuando se ha terminado la fase anterior. [Hawryszkiewycz, I.T. 1990]

- Para que un proyecto tenga éxito, en cualquier caso, todos las fases señalados en el modelo en cascada deben ser desarrollados.
- Para pasar de una fase a otra es necesario conseguir todos los objetivos de la etapa previa. [Böehm, B.W. 1981]
- Cualquier desarrollo en diferente orden de las fases dará un producto de inferior calidad. Sin embargo, existen ciertos proyectos para los cuales este orden es inviable. Esta ha sido una de las principales razones para definir otros modelos.
- Las etapas están organizadas de un modo lógico. Es decir, si una etapa no puede llevarse a cabo hasta que se hayan tomado ciertas decisiones de más alto nivel, debe esperar hasta que esas decisiones estén tomadas. Así el diseño espera a los requisitos, el código espera a que el diseño esté terminado, etc.
- Al final de cada fase, tanto los desarrolladores como los usuarios, tienen la oportunidad de revisar el proyecto.
- Cada etapa incluye cierto proceso de revisión y se necesita una aceptación del producto antes de que la salida de la etapa pueda usarse. Este ciclo de vida está organizado de modo que se pase el menor número de errores de una etapa a la siguiente.
- Facilita la gestión de control del progreso del desarrollo del sistema, de las fechas de entrega y de los costos esperados.
- El ciclo es iterativo. A pesar de que el flujo básico es de arriba hacia abajo, el ciclo de vida en cascada reconoce, qué problemas encontrados en etapas inferiores afectan a las decisiones de las etapas superiores.
- El modelo en cascada asume que los requisitos de un sistema pueden ser congelados antes de comenzar el diseño. Esto para sistemas totalmente nuevos, es poco realista. Congelar los requisitos requiere seleccionar el hardware. La terminación de un gran proyecto puede llevar años. Dada la velocidad de obsolescencia de la tecnología es bastante probable que el software final utilice un hardware obsoleto.
- No refleja el proceso real de desarrollo de software. Los proyectos rara vez siguen el flujo secuencial, puesto que siempre hay iteraciones. Aunque en este modelo la iteración está permitida en etapas contiguas [Macro, A. 1990], en la práctica real la iteración abarca más de una etapa.
- El sistema en funcionamiento no está disponible hasta las fases finales del proyecto. Esto significa que la mayor parte del feedback del cliente sobre sus necesidades se obtiene una vez que se han consumido los recursos.

Existe una perspectiva alternativa del modelo de ciclo de vida en cascada, que enfatiza la validación de los productos. Ver figura 2-3. Esta variante del modelo en Cascada presenta una descomposición en fases, que se visualiza en

la línea superior, consiste en: la definición de los requisitos del sistema global y la especificación de los requisitos del software. Estos últimos llevan al diseño preliminar de múltiples funciones, cada una de las cuales se expande en el

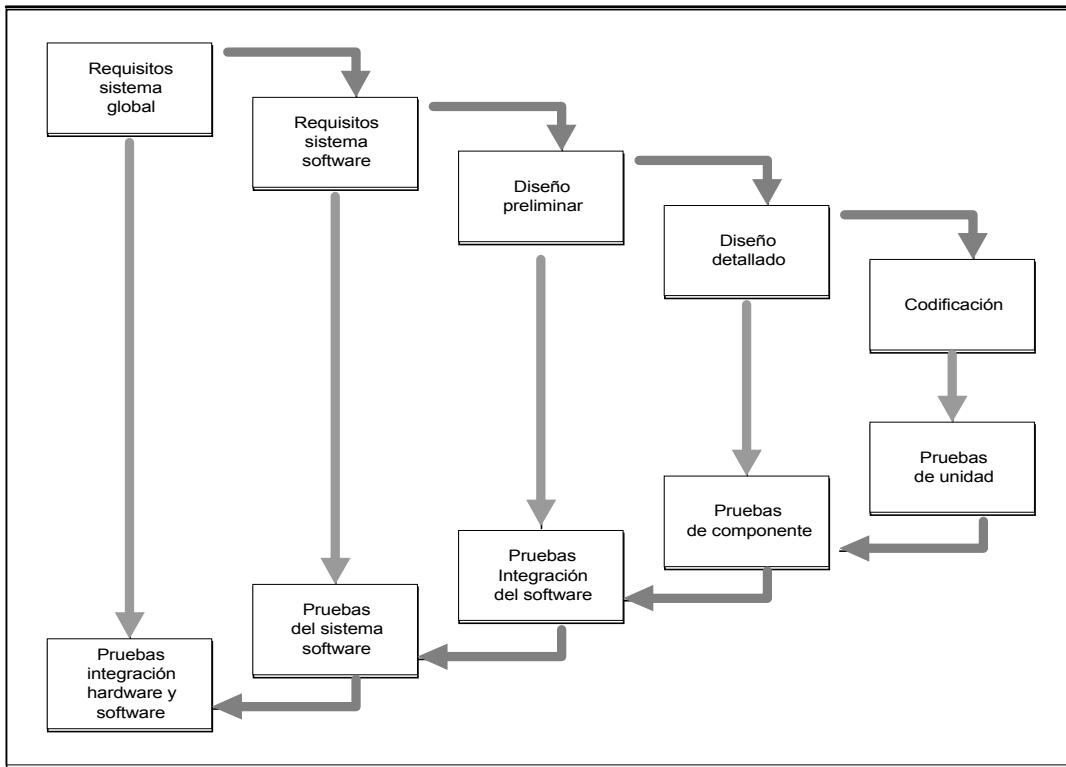


FIGURA 2-3: Ciclo de Vida en Cascada, validación de productos.

diseño detallado, que, a su vez, evoluciona en un número mayor de programas unitarios. El ensamblaje del producto final se produce en el sentido contrario, dentro de un proceso de síntesis o composición. Primero se aceptan los programas unitarios probados. Entonces estos se agrupan en módulos que a su vez, deben ser probados y aceptados. Los módulos se agrupan para certificar que el grupo formado por todos ellos incluye las funcionalidades deseadas. Finalmente, el software es integrado con el hardware hasta formar un único sistema informático que satisface los requisitos globales.

La perspectiva de la figura 2-3 destaca que el producto obtenido en cada etapa no sólo determina qué debe hacerse en la fase siguiente del proceso, sino que también establece los criterios para determinar si el producto compuesto y ensamblado satisface los objetivos de la etapa.

2.4 CICLO DE VIDA EN ESPIRAL

El modelo en espiral propuesto por Boehm [Boehm, B.W. 1987], consta de una serie de ciclos que se recorren por iteraciones sucesivas hasta cumplir cierto hito o condiciones prefijadas. El modelo surge como un modelo no operativo de producción de software que tiende a poner énfasis en el riesgo a asumir en cada etapa y el control de ese riesgo. Esta característica ha motivado un creciente uso de este modelo en grandes proyectos, que son, cada vez más permeables al riesgo.

Como el riesgo es uno de los fundamentos característicos de este modelo se introduce una breve explicación del concepto.

2.4.1 CONCEPTO DE RIESGO

Cada vez que se inicia un proyecto de desarrollo de software aparecen áreas de incertidumbre del tipo:

- ¿Se entienden realmente las necesidades del cliente?
- ¿Se pueden implementar las funciones a desarrollar para la fecha fijada?
- ¿Se encontrarán problemas técnicos de difícil solución que no son visibles al inicio del proyecto?
- Los cambios que se producen en el proyecto, ¿en cuánto afectarán la planificación?

En muchas ocasiones las estrategias de riesgo que se implementan en los proyectos de desarrollo de software son reactivas, es decir, se supervisa el proyecto en previsión de posibles riesgos. Lo más frecuente es que no se haga nada respecto a los riesgos hasta que algo sale mal. El equipo que lleva adelante el proyecto corre entonces, para corregir el problema rápidamente (“estrategia del bombero”). En ese momento cuando se produce el problema, “la gestión de riesgo” entra en acción y el proyecto se encuentra en peligro real.

Una estrategia mejor para el control de riesgo es proactiva, es decir, comienza antes de comenzar los trabajos técnicos. Se identifican riesgos potenciales, se valoran su probabilidad y su impacto y se establece una prioridad según su importancia. Después el equipo de desarrollo establece un plan para controlar el riesgo. El primer objetivo es evitar el riesgo, pero como no se pueden evitar todos los riesgos, el equipo trabaja para desarrollar un plan de contingencia que le permita responder de una manera eficaz y controlada.

El riesgo implica dos características:

- Incertidumbre: el acontecimiento que caracteriza al riesgo puede o no ocurrir. No hay riesgos con un 100 % de probabilidad.
- Pérdida: si el riesgo se convierte en una realidad ocurrirán consecuencias no deseadas o pérdidas.

Cuando se analizan los riesgos es importante cuantificar el nivel de incertidumbre y el grado de pérdida asociado con cada riesgo.

El Análisis del riesgo consta de cuatro actividades:

1. *Identificación del riesgo*: es un intento sistemático para especificar las amenazas al proyecto. Un método para identificar riesgos es crear una lista de comprobación de elementos de riesgo. Esta lista puede organizarse de diferente manera considerando la categoría genérica de los riesgos (del proyecto, técnicos, del negocio) o se puede enfocar en las características más relevantes de los siguientes factores: tamaño del producto, impacto en el negocio, características del cliente, definición del proceso, entorno de desarrollo, tecnología a usar, complejidad del sistema, tamaño y experiencia del equipo de desarrollo.
2. *Proyección del riesgo*: implica estimar dos aspectos del riesgo, la probabilidad de que el riesgo se concrete y las consecuencias de los problemas asociados al riesgo, si ocurriera. Para ello se realizan cuatro actividades de proyección de riesgo: establecer una escala que refleje la probabilidad de ocurrencia del riesgo, definir las consecuencias del riesgo, estimar el impacto del riesgo en el proyecto y en el producto y documentar la proyección del riesgo. Es una técnica sencilla que puede implementarse en una hoja de cálculo y que facilita la proyección en la tabla de riesgo [Pressman, R.S. 1997].
3. *Evaluación del riesgo*: durante la evaluación del riesgo se continúa examinando la exactitud de las estimaciones que se hicieron en el paso previo, se priorizan los riesgos que no se habían cubierto y se analizan las formas de controlar y/o impedir la ocurrencia de los riesgos más probables. Para que esta evaluación sea efectiva se debe definir un nivel de referencia de riesgo [Charette, R.N. 1992], que permite definir el punto de ruptura en el que las decisiones de seguir el proyecto o dejarlo son igualmente aceptables.
4. *Gestión del riesgo*: se toma como base el análisis realizado respecto de los riesgos y se desarrolla un plan que detalla los pasos o actividades a realizar para gestionar cada riesgo. Estas actividades consumen recursos y por lo tanto, generan un costo adicional en el proyecto. Una parte de la gestión del riesgo consiste en la evaluación de los beneficios conseguidos con los propios pasos de la gestión de riesgo, comprobando que tengan un mayor peso que los costos asociados a su implementación, es decir, se realiza un análisis costo-beneficio. La experiencia demuestra que el 80% de los problemas

posibles del proyecto se deben al 20% de los riesgos identificados [Pressman, R.S. 1997]. El análisis que se toma como base sirve para determinar qué riesgos se encuentran en ese 20%. Es posible entonces que algunos de los riesgos que hayan sido identificados, calculados y evaluados no se encuentren reflejados en el plan de gestión de riesgos porque no pertenecen al 20% crítico.

2.4.2 FASES DEL CICLO DE VIDA EN ESPIRAL

El modelo en espiral puede recorrerse de distinta forma según el caso en estudio. El recorrido básico del modelo es por ciclos, sobre las sendas determinadas por la espiral (ver figura 2-4) siguiendo el sentido de las agujas del reloj. Sus actividades se agrupan en fases. Una vez completadas las fases del primer ciclo se vuelve a empezar el ciclo siguiente.

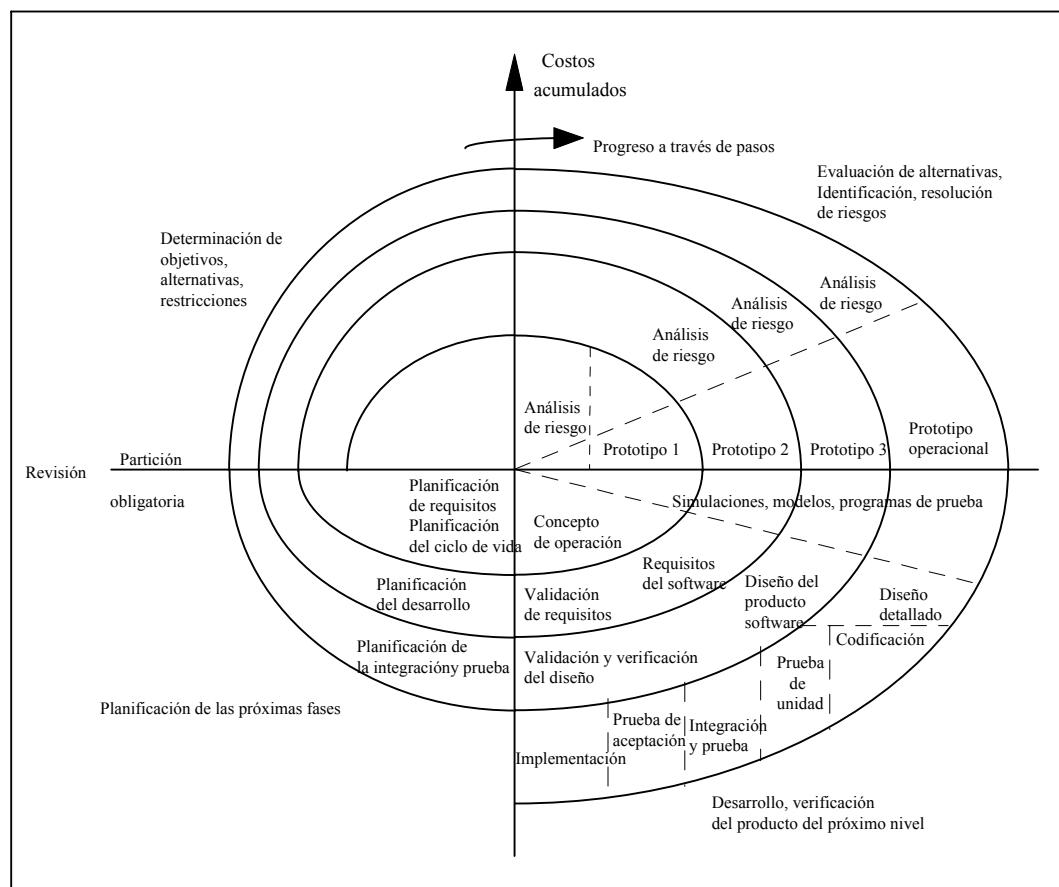


FIGURA 2-4: Ciclo de Vida en Espiral

Primera fase. Consiste en identificar:

- Los objetivos del sistema o subsistema a desarrollar. Se considera el rendimiento, funcionalidad, flexibilidad para los cambios, etc.
- Las alternativas significativas para implementar este sistema o subsistema. Puede considerarse diseñar A, diseñar B, reusar, comprar, etc.
- Las restricciones impuestas sobre la implementación de las alternativas. Como costos, calendarios, interfaces, etc.

Segunda fase. Consiste en:

- Evaluar las diferentes alternativas identificadas en el paso previo con relación a los objetivos y las restricciones.
- Identificar áreas de incertidumbre que son causa de riesgos del proyecto. Se identifican entonces los riesgos del proyecto.
- Formular estrategias para resolver las causas de mayor riesgo. Para ello se utilizan técnicas de resolución de riesgos, tales como: prototipado, simulación, cuestionarios, modelización analítica, o combinaciones de éstas u otras técnicas.
- Determinar cómo se continúa con la fase siguiente. Como resultado del paso previo es posible que algunos riesgos identificados aún persistan. El próximo paso se decide en relación con los riesgos que aún existen. Considerando las características de estos riesgos el modelo puede recorrerse de múltiples formas.

Tercera fase. Consiste en seguir alguna de las siguientes alternativas:

- *Alternativa 1.* Recorrer todas las fases de la espiral en secuencia.
- *Alternativa 2.* Recorrer un subconjunto de todas las fases de la espiral: las fases encerradas dentro del ángulo que contienen los prototipos y las simulaciones.
- *Alternativa 3.* Recorrer un subconjunto de todas las fases de la espiral: todas las fases menos las fases encerradas dentro del ángulo que contiene los prototipos y las simulaciones.
- *Alternativa X.* Recorrer un subconjunto X de todas las fases de la espiral, que determinará el líder de proyecto.

Alternativa 1:

Si los riesgos de interfaz de usuario, de estrategia de control o de rendimiento dominan fuertemente el desarrollo del sistema, el próximo paso puede ser un desarrollo tipo evolutivo. La espiral se recorre en todas sus fases secuencialmente: un mínimo esfuerzo para especificar la naturaleza general del producto (Concepto de operación); un plan para el próximo nivel de prototipo (Planificación de requisitos, Planificación de ciclo de vida); hasta el

desarrollo de un prototipo de mayor detalle (Prototipo 2) para continuar resolviendo los principales riesgos emergentes (Simulaciones, modelos...); y así sucesivamente hasta completar los ciclos de la espiral.

Alternativa 2:

Si el prototipo desarrollado es operacionalmente útil y lo suficientemente robusto como para ser la base, con bajo riesgo, para la evolución del producto, los siguientes pasos para el manejo de los riesgos remanentes pueden ser el desarrollo de series de prototipos evolutivos. La espiral se recorre dirigiéndose hacia la derecha de la figura 2-4, completando las fases encerradas dentro del ángulo de líneas punteadas.

Alternativa 3:

Si los riesgos de rendimiento y de interfaz de usuario fueron resueltos y el desarrollo del sistema se encuentra dominado por los riesgos de la estrategia de control, los siguientes pasos pueden resolverse con un modelo semejante al modelo en cascada. La espiral se recorre completando las fases (Concepto de Operación, Requisitos de Software, Diseño del producto Software, etc.) con excepción de las fases encerradas dentro del ángulo de líneas punteadas. Cada fase es seguida por una fase de validación y de preparación de planes para el siguiente ciclo de la espiral.

Alternativa X:

El modelo en espiral permite acomodar cualquier modelo de desarrollo de software o cualquier combinación de ellos, considerando las características de los riesgos y la efectividad relativa de las diversas técnicas de resolución de riesgos.

En la planificación de las siguientes fases puede considerarse, por ejemplo, subdividir el sistema en partes y encarar el desarrollo de esas partes ya sea por desarrollo incremental o que las partes sean desarrolladas por distintos grupos, organizaciones o personas individuales.

En ese caso es posible visualizar el modelo en espiral agregando la tercera dimensión al modelo. Es decir considerar una serie de círculos paralelos de la espiral, un círculo por cada componente. Por ejemplo, cada espiral individual se asocia al desarrollo de cada uno de los subsistemas o componentes, siguiendo su propia forma de evolución según las características de esos subsistemas o componentes. Cada espiral puede ser asignada, según el componente a desarrollar a un programador, diseñador, cliente, usuario, etc.

Una característica importante del modelo en espiral es que cada ciclo se completa con una revisión en la que participan las principales personas u organizaciones que tienen relación con el producto. Esta revisión cubre todos los

productos desarrollados durante el ciclo anterior, incluyendo los planes para el siguiente y los recursos necesarios para llevarlos a cabo. El objetivo principal de la revisión sirve para asegurar que todas las partes involucradas están de acuerdo respecto al método de trabajo para la siguiente fase.

2.5 CICLO DE VIDA ORIENTADO A OBJETOS

El paradigma de Orientación a Objetos (OO) se caracteriza por conceptualizar el desarrollo de software empleando modelos cuya unidad básica es el “objeto” que combina las estructuras de datos con los comportamientos en una entidad única [Rumbaugh, J. y otros 1996].

En primer lugar, se construye un modelo de análisis para abstraer los aspectos esenciales del dominio de la aplicación sin tener en cuenta la implementación. En esta etapa se toman decisiones importantes que después se completan para optimizar la implementación. En segundo lugar, los objetos del dominio de la aplicación constituyen el marco de trabajo del modelo de diseño, pero se implementan en términos de objetos del dominio de la computadora. Por último, el modelo de diseño se implementa en algún lenguaje de programación, base de datos y hardware.

Los objetos del dominio de la aplicación y del dominio de la computadora se pueden modelar, diseñar e implementar utilizando los mismos conceptos y la misma notación orientada a objetos. Esta misma notación se usa desde el análisis hasta la implementación pasando por el diseño, de forma tal que la información añadida en la fase de desarrollo no necesita perderse, ni ser traducida, para la próxima fase.

La esencia del desarrollo orientado a objetos es la identificación y organización de conceptos (objetos) del dominio de la aplicación. Sólo cuando se han identificado, organizado y comprendido los conceptos inherentes de la aplicación se pueden tratar en forma efectiva los detalles de las estructuras de datos y de las funciones. Es una premisa básica que los errores de las primeras fases del proceso de desarrollo tienen mucha influencia sobre el producto final y también sobre el tiempo requerido para finalizar.

El beneficio principal del desarrollo orientado a objetos no es reducir el tiempo de desarrollo; el desarrollo orientado a objetos puede requerir más tiempo que el desarrollo convencional porque se pretende que promueva la reutilización futura y la reducción de los posteriores errores y el futuro mantenimiento. El tiempo transcurrido hasta que el código se completa por primera vez es

posiblemente el mismo que el transcurrido en una aproximación convencional o, quizás ligeramente mayor. El beneficio del desarrollo orientado a objetos consiste en que las iteraciones subsiguientes son más rápidas y más fáciles que empleando un desarrollo convencional porque las revisiones están más localizadas. La práctica muestra que suelen ser necesarias menos iteraciones porque se descubren y se corrigen más problemas durante el desarrollo. [Rumbaugh, J. y otros 1996].

2.5.1 CONCEPTOS BASICOS

Hay varios conceptos que son propios de la orientación a objetos y otros inherentes a la tecnología. Aunque no todos son exclusivos de los sistemas orientados a objetos están bien apoyados por el paradigma.

Orientado a Objetos: significa que el software se organiza como una colección de objetos discretos que contienen tanto estructuras de datos como comportamiento.

Identidad: los datos están cuantificados en entidades discretas y distinguibles denominadas objetos. Cada objeto posee su propia identidad inherente. Pueden ser ejemplos de objetos, *un párrafo de un documento, la reina blanca del juego de ajedrez, una silla*. En otras palabras: dos objetos serán distintos aún cuando los valores de todos sus atributos (tales como el nombre y el tamaño) sean idénticos. Por ejemplo en un conjunto de 6 sillas de un mismo juego, los valores de los atributos son los mismos y cada una de las sillas tiene su propia identidad.

Identificación: en el mundo real los objetos se limitan a existir, pero dentro de un entorno de computación cada objeto posee una identificación mediante la cual se puede hacer alusión a él de modo exclusivo. La identificación se puede implementar de distintas maneras que pueden ser como una dirección, un índice de una matriz, o un valor exclusivo de un atributo.

Clasificación: significa que los objetos con la misma estructura de datos (atributos) y comportamiento (operaciones) se agrupan, bajo alguna condición, para formar una clase. Son ejemplos de clases: *párrafo, pieza de ajedrez*.

Clase: es una abstracción que describe propiedades importantes para una aplicación y que ignora el resto. La selección de clases es arbitraria y depende de la aplicación. Una clase contiene el molde (estructura, esquema) a partir del cual se crean los objetos que pertenecen a ella y el código que debe ejecutarse cada vez que un objeto de la clase recibe un mensaje. Una clase contiene la

descripción de las características comunes de todos los objetos que pertenecen a ella: la especificación del comportamiento, la definición de la estructura interna y la implementación de los métodos. Por ejemplo la clase "gato" es diferente si la aplicación es para un sistema de gestión de una veterinaria que si es un sistema de películas de dibujos animados.

Instancia: se dice que cada objeto es una instancia de su clase. Toda clase describe un conjunto posiblemente finito de objetos individuales. Toda instancia de la clase posee su propio valor para cada uno de los atributos pero comparte los nombres de los atributos y las operaciones con las demás instancias de la clase. Todo objeto contiene una referencia implícita a su propia clase: "sabe la clase de cosa que es". Los objetos contienen los valores de los atributos (que lo distinguen de otros objetos) y una identidad. Si continuamos con el ejemplo del sistema de películas de dibujos animados y de la clase gato, pueden considerarse como instancias: el gato Tom (Tom & Jerry), el gato Silvestre (amigo de Piolín), el gato Félix.

Operación: es una acción o una transformación que se lleva a cabo, o que se aplica a un objeto. *Maullar, Saltar, Arañar* son ejemplos de operaciones que se aplican a un objeto gato, en el marco del sistema películas dibujos animados.

Método: es una implementación específica de una operación ejecutable por una cierta clase. Codificar las operaciones como Maullar, Arañar en algún lenguaje.

Polimorfismo: significa que una operación puede comportarse de modos distintos en distintas clases teniendo el mismo nombre de método. La operación *mover* se puede comportar distinto en las clases *párrafo* y *pieza de ajedrez*. En términos prácticos, el polimorfismo permite referirse a objetos de diferentes clases, por medio del mismo elemento de programa y realizar la misma operación de formas diferentes, de acuerdo al objeto a que se hace referencia en cada momento. En el mundo real una operación es simplemente, una abstracción de comportamiento análogo entre distintas clases de objetos. Cada objeto sabe llevar a cabo sus propias operaciones. Sin embargo, en un lenguaje orientado a objetos es éste el que selecciona automáticamente el método correcto para implementar una operación basándose en el nombre de la operación y en la clase del objeto que está siendo afectado. El usuario de una operación no necesita ser consciente del número de métodos que existen para implementar una cierta operación polimórfica. Se pueden añadir clases sin modificar el código existente, siempre y cuando se proporcione métodos para todas las operaciones aplicables a las nuevas clases. Por ejemplo para la clase gato, saltar puede diferenciarse de la operación saltar para la clase perro.

Herencia es compartir atributos y operaciones entre clases tomando como base una relación jerárquica. En términos generales se puede definir una clase que después se irá refinando sucesivamente para producir subclases. Todas las subclases poseen o heredan todas y cada una de las propiedades de su superclase y añaden, además, sus propiedades exclusivas. No es necesario repetir las propiedades de las superclases en cada subclase. Por ejemplo, *ventana de desplazamiento* y *ventana fija* son subclases de *ventana*. Ambas subclases heredan las propiedades de *ventana* tales como una región visible en la pantalla. La *ventana de desplazamiento* añade una *barra de desplazamiento* y un *botón ascensor*. La capacidad de sacar factor común a las propiedades de varias clases en una superclase común y de heredar las propiedades de la superclase puede reducir muchísimo la repetición en el diseño y en los programas siendo una de las ventajas principales de un sistema orientado a objetos. Otro ejemplo de subclases puede ser *gato siamés* y *gato de angora* de la clase *gato*.

Abstracción: consiste en centrarse en los aspectos esenciales de una entidad e ignorar sus propiedades accidentales. En el desarrollo de sistemas esto significa centrarse en lo que es y lo que hace un objeto antes de decidir cómo debería ser implementado. La capacidad de utilizar herencia y polimorfismo proporciona una potencia adicional. El uso de la abstracción durante el análisis significa tratar solamente conceptos del dominio de la aplicación y no tomar decisiones de diseño o de implementación antes de haber comprendido el problema. Un uso adecuado de la abstracción permite utilizar el mismo modelo para el análisis, diseño de alto nivel, estructura del programa, estructura de una base de datos y documentación. Un estilo de diseño independiente del lenguaje pospone los detalles de programación hasta la fase final, relativamente mecánica, del desarrollo.

Encapsulamiento: denominado también *ocultamiento de información* consiste en separar los aspectos externos del objeto, a los cuales pueden acceder otros objetos, de los detalles internos de implementación del mismo, que quedan ocultos para los demás. El encapsulamiento evita que el programa sea tan interdependiente que un pequeño cambio tenga efectos secundarios masivos. La implementación de un objeto se puede modificar sin afectar a las aplicaciones que la utilizan. Quizás sea necesario modificar la implementación de un objeto para mejorar el rendimiento, corregir un error, consolidar el código o para hacer un transporte a otra plataforma. El encapsulamiento no es exclusivo de los lenguajes orientados a objetos pero la capacidad de combinar la estructura de datos y el comportamiento en una única entidad hace que el encapsulamiento sea más potente y claro que en los lenguajes convencionales que separan las estructuras de datos y el comportamiento.

Combinación de datos y comportamientos: el enfoque orientado a objetos tiene una sola jerarquía, jerarquía de clases. Unifica la jerarquía de estructuras de datos y jerarquía de procedimientos que presentan los enfoques convencionales. Cuando un objeto invoca una operación no necesita considerar cuántas implementaciones existen de una operación dada. El polimorfismo de operadores traslada la carga de decidir que implementación hay que utilizar llevándola, del código que hace la llamada, a la jerarquía de clases. El mantenimiento es más sencillo porque el código que hace la llamada no necesita ser modificado cuando se añade una clase nueva. En el contexto de un sistema orientado a objetos la jerarquía de estructuras de datos es idéntica a la jerarquía de herencia de operaciones.

Reutilización: la herencia tanto de estructuras de datos como de comportamientos, permite compartir una estructura común entre varias subclases similares sin redundancia. Una de las principales ventajas de los lenguajes orientados a objetos es compartir código empleando la herencia. Esto reduce el número de clases distintas que es preciso comprender y analizar.

El desarrollo orientado a objetos no sólo permite compartir información dentro de una aplicación sino que, además, ofrece la perspectiva de volver a utilizar diseños y códigos en futuros proyectos. Aún cuando esta posibilidad se ha hecho resaltar excesivamente como justificación de la tecnología orientada a objetos, el desarrollo orientado a objetos proporciona herramientas tales como la abstracción, encapsulado y herencia para construir bibliotecas de componentes reutilizables. Se debe tener presente que la reutilización no sucede, debe ser planeada pensando más allá de la aplicación inmediata y se debe invertir un esfuerzo adicional en lograr un diseño mas general.

2.5.2 FASES DEL CICLO DE VIDA ORIENTADO A OBJETOS

Los sistemas tienden a evolucionar con el tiempo, por esto, un modelo evolutivo de proceso acoplado con un enfoque que fomenta la reutilización de componentes es el mejor paradigma para ingeniería de software OO. El proceso OO se mueve a través de una espiral evolutiva que puede considerar el desarrollo incremental e iterativo de prototipos.

Es posible aplicar conceptos orientados a objetos a lo largo del todo el ciclo de vida de desarrollo del sistema, desde el análisis hasta la implementación pasando por el diseño. Se pueden traspasar las mismas clases de una etapa a otra sin modificar la notación aunque ganarán detalles adicionales de implementación en las etapas posteriores. Los mismos conceptos orientados a

objetos de identidad, clasificación, polimorfismo y herencia son aplicables a todo el ciclo de desarrollo completo.

Existen diversas propuestas de metodologías de desarrollo de sistemas de información basadas en el paradigma de OO. Las metodologías orientadas a objetos más conocidas son:

- OMT (Object Modelling Technique) de Rumbaugh [Rumbaugh, J. y otros 1996].
- El micro y macroproceso de Booch [Booch, G. 1996].
- OOSE (Object Oriented Software Engineering) de Jacobson [Jacobson, I. 1992].
- También existe el método Proceso Unificado (Unified Software Development) [Jacobson, I. y otros 1999], nacido de la fusión de los 3 métodos mencionados.

En la figura 2-5 se muestra el ciclo de vida Orientado a objetos.

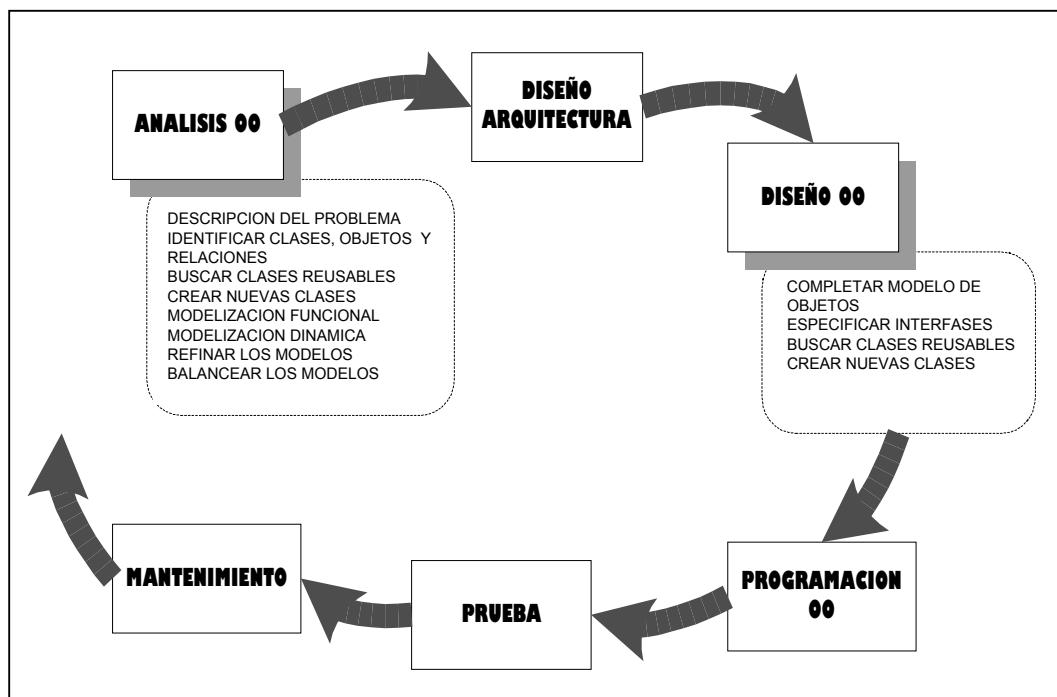


FIGURA 2-5: Ciclo de Vida Orientado a Objetos

Análisis OO: El análisis estudia el comportamiento del sistema en su totalidad. El objetivo de la actividad de análisis consiste en exponer y comprender el problema y conceptos del dominio de la aplicación, dejando de lado conceptos de implementación. El modelo de análisis es una abstracción de lo *que* debe hacer el sistema deseado y no de la *forma* en que se hará. Un buen modelo podrá ser

comprendido y revisado por expertos de la aplicación que no sean programadores [Rumbaugh, J. y otros 1996]. El modelo de análisis no debe contener ninguna decisión de implementación, los objetos se describirán en términos de atributos y operaciones que son visibles para el usuario.

El comienzo de la fase de análisis es una descripción inicial del problema que hay que resolver. La definición de este problema no suele ser completa ya que es generada por el cliente y, por lo tanto, no debe tomarse como invariable. No es más que un punto de partida que debe servir como base para refinar los requisitos reales. No existe ninguna razón para esperar que la definición del problema que se haya preparado sin un análisis completo sea la correcta.

El primer paso consiste en establecer los requisitos básicos. Estos deben ser comunicados por el cliente al Ingeniero de Software. Es imprescindible la comprensión en su totalidad del problema de tal forma que el Ingeniero de Software debe trabajar con el solicitante hasta refinar los requisitos y lograr que los mismos representen la verdadera intención del usuario.

El resultado del análisis es un modelo formal que representa en forma concisa el problema y permite construir una solución. Se genera una representación del mundo real compuesto por distintos modelos en los que se analizan la estructura estática (objetos y sus relaciones), secuencia de acciones (dinámica de las funciones del sistema) y transformación de datos (funciones o procesos).

Las actividades de esta fase son:

- Definir y revisar las clases, y la jerarquía de clases
- Definir y revisar los atributos y las operaciones asociadas a una clase.
- Establecer las relaciones entre clases.
- Crear y revisar un modelo de comportamiento
- Marcar clases reutilizables
- Definir los modelos funcionales y dinámicos.

La OO enfatiza la reutilización de objetos, por lo tanto, una actividad importante del análisis del sistema es buscar clases en una biblioteca (de clases OO existentes) antes de construir otras.

Es poco probable definir las clases necesarias para un gran sistema o producto en una sola iteración. El análisis OO y los modelos de diseño van evolucionando y surge la necesidad de clases adicionales. El proceso iterativo de la fase se muestra en la figura 2-5 dentro de un cuadro de líneas punteadas.

Diseño Arquitectura: El diseño de sistema es la primera fase del diseño donde se selecciona la aproximación básica para resolver el problema decidiendo la estructura y el estilo global.

Existe un cierto número de estilos de arquitectura, cada uno de los cuales es adecuado para ciertas clases de aplicaciones. La arquitectura del sistema está constituida por su descomposición en subsistemas, su concurrencia inherente, la

asignación de subsistemas a hardware y a software, la administración de datos, la coordinación de recursos globales, la implementación del control de software, las condiciones de contorno y las prioridades de compensación. Cada uno de los puntos mencionados son características que el diseñador de sistemas deberá decidir.

Un sistema se puede dividir en un pequeño número de componentes principales denominados subsistemas. Los mismos comparten alguna propiedad común y a su vez pueden descomponerse en subsistemas propios aún más pequeños. Los subsistemas de más bajo nivel se denominan módulos. Se asignan los subsistemas a procesadores y a tareas. Cada subsistema debe ser asociado a una unidad de hardware, a un procesador de propósito general o a una unidad funcional especializada.

Es posible utilizar almacenes de datos para separar subsistemas distintos dentro de una arquitectura y para dar a los datos de la aplicación un cierto grado de permanencia. Estos almacenes se pueden implementar mediante estructuras de datos en memoria, archivos y/o bases de datos.

El diseñador de sistemas debe identificar los recursos globales y tiene que determinar mecanismos para controlar el acceso a los mismos. Existen dos clases de flujos de control: Control externo y Control interno.

El control interno es el flujo de control dentro de un proceso, mientras que el control externo es el flujo de los sucesos externamente visibles entre los objetos del sistema. Además existen 3 clases de control para sucesos externos: secuencial controlado por procedimientos, secuencial controlado por sucesos y concurrente.

La mayoría del diseño trata del comportamiento estacionario, no obstante es preciso considerar también las condiciones de contorno: Iniciación, Terminación y Fallos.

Un aspecto esencial de la arquitectura del sistema es establecer prioridades de compensación entre tiempo y espacio, hardware y software, sencillez y generalidad y eficiencia y mantenibilidad. Para esto no debe considerarse sólo el software en sí mismo sino también el proceso de desarrollo.

Diseño OO: El diseñador de objetos construye un modelo de diseño basándose en el modelo de análisis que lleven incorporados detalles de implementación; es decir durante esta fase se elaboran los modelos de análisis, se refinan y finalmente se optimizan para producir un diseño práctico.

Durante esta fase existe un desplazamiento del foco de atención desde los conceptos de la aplicación hasta los conceptos de la computadora haciendo hincapié en las estructuras de datos y los algoritmos necesarios para implementar cada una de las clases.

No se parte de cero, sino que las operaciones identificadas durante el análisis deben expresarse en forma de algoritmos sencillos optimizando así la estructura

del modelo de objetos con el objetivo de lograr una implementación eficiente y optimizando medidas importantes de rendimiento.

Tanto los objetos del dominio de la aplicación como los objetos del dominio de la computadora se describen utilizando unos mismos conceptos y una misma notación orientados a objetos aún cuando existan en planos conceptuales diferentes.

Algunas clases no forman parte del análisis sino que se presentan como parte del diseño o de la implementación. Por ejemplo, las estructuras de datos tales como árboles, tablas de dispersión y listas enlazadas no suelen estar presentes en el mundo real. Se presentan para que presten su apoyo a algoritmos concretos durante el diseño. Estos objetos de estructuras de datos se utilizan para implementar objetos dentro de la computadora y no derivan directamente sus propiedades del mundo real.

Los pasos que debe llevar a cabo el diseñador son:

- El diseñador debe transformar las acciones y los procesos del modelo funcional en operaciones asociadas a las clases del modelo de objetos haciendo corresponder la estructura lógica del modelo de análisis en una organización física de un programa
- Se seleccionan algoritmos que muestran cómo se hace la operación especificada durante el análisis teniendo en cuenta complejidad computacional, facilidad de implementación y comprensibilidad, flexibilidad y ajuste fino del modelo de objetos
- Seleccionar estructuras de datos que organicen la información del modelo de análisis de forma cómoda para los algoritmos que la utilizan. Entre estas estructuras de datos se cuentan las matrices, listas, colas, pilas, conjuntos, bolsas, diccionarios, asociaciones, árboles y variaciones de todas ellas
- Definir nuevas clases y operaciones internas necesarias y que no fueron directamente mencionadas en la descripción del problema por parte del cliente
- Añadir asociaciones redundantes para minimizar el tiempo de acceso.
- Se implementa el modelo dinámico teniendo en cuenta tres aproximaciones básicas: sistema controlado por procedimientos, sistema controlado por sucesos y utilización de tareas concurrentes.
- Se ajustan las definiciones de las clases y las operaciones con el objetivo de incrementar la herencia. Se reorganizan clases y operaciones teniendo en cuenta que todas las operaciones deben tener la misma interfaz, la misma semántica y la misma signatura previo a poder utilizar la herencia
- Se diseña la implementación de asociaciones. Cada asociación se implementa como un objeto por separado o bien añadiendo atributos de enlace cuyos valores sean objetos a una de las clases de la asociación (o a las dos)

- Se empaquetan las clases y las asociaciones en módulos. El empaquetamiento implica ocultar la información interna a los ojos del exterior, coherencia de entidades y construcción de módulos físicos.

Es importante considerar que el diseño orientado a objetos es un proceso iterativo.

La documentación del diseño de objetos es un conjunto de modelos entre los que se destacan: el modelado de objetos detallado, el modelado dinámico detallado y el modelo funcional detallado, tanto en forma gráfica (diagrama del modelo de objetos), como en forma textual (descripción de clases).

Programación OO: Las clases de objetos y las relaciones desarrolladas durante el diseño se traducen finalmente a un lenguaje de programación concreto, a una base de datos o a una implementación en hardware. La programación debería ser una parte relativamente pequeña del ciclo de desarrollo y fundamentalmente mecánica porque todas las decisiones importantes deberán hacerse durante el diseño. El lenguaje de destino influye en cierta medida sobre las decisiones de diseño pero éste no debería depender de la estructura final de un lenguaje de programación.

La manera más directa de implementar un diseño OO es usando un lenguaje orientado a objetos, pero incluso los lenguajes orientados a objetos ofrecen distintos grados de apoyo para los conceptos OO. Cada lenguaje supone un compromiso entre la potencia conceptual, eficiencia y compatibilidad con los trabajos anteriores. El uso de un lenguaje no orientado a objetos exige mayor cuidado y disciplina para mantener la estructura de objetos en el programa. Para implementar un diseño OO en un lenguaje OO deben tenerse en cuenta [Rumbaugh, J. y otros 1996]:

- Definición de clases
- Creación de objetos
- Llamadas a operaciones
- Uso de la herencia
- Implementación de asociaciones

Algunos sistemas de bases de datos OO integran un lenguaje OO con una base de datos como un todo, sin discontinuidades. Se pueden definir operaciones para cada clase de objetos, pero el programador no necesita leer y escribir explícitamente en un almacenamiento persistente. Aun cuando los sistemas de base de datos OO prometen un redimiendo mejor y mas facilidad de uso a largo plazo todavía no son tan maduros como los sistemas convencionales de bases de datos relacionales y además pueden plantearse problemas para integrarlos con aplicaciones convencionales ya existentes.

Durante la codificación es posible que sea necesario [Booch, G. 1996]:

- Añadir una nueva clase o una nueva colaboración entre clases

- Cambiar el diseño de una clase
- Cambiar la representación de una clase
- Reorganizar la estructura de clases
- Cambiar la interfaz de una clase

La implementación de un diseño OO en un lenguaje no OO requiere que el programador haga corresponder los conceptos de objetos con el lenguaje. Los pasos necesarios son [Rumbaugh, J. y otros 1996]

- Traducir las clases a estructuras de datos
- Pasar argumentos a los métodos
- Reservar espacio para los objetos
- Implementar la herencia en estructuras de datos
- Implementar la resolución de los métodos
- Implementar las asociaciones
- Resolver la concurrencia
- Encapsular los detalles internos de las clases

Los buenos programas siguen reglas de diseño tienen mayores posibilidades de ser correctos, reutilizables, extensibles y fáciles de corregir. Esas reglas deben tener en cuenta la Reutilización, Extensibilidad, Robustez y la Programación Industrial [Rumbaugh, J. y otros 1996].

PRUEBA: La prueba debe ser una actividad continua durante el proceso de desarrollo. Debe centrarse en el comportamiento externo del sistema. Un propósito secundario de la prueba es tantear los límites del sistema con el fin de comprender como falla bajo ciertas condiciones. Debe abarcar al menos tres dimensiones:

- Prueba unitaria: Implica la prueba de clases y mecanismos individuales.
- Prueba del subsistema: Implica probar una categoría completa o subsistema. Pueden utilizarse como pruebas de regresión para cada versión del prototipo, verificando hasta qué grado se satisface los puntos funcionales asignados.
- Prueba del sistema: implica probar el sistema como un todo. También se pueden usar como pruebas de regresión cuando se ensamblan nuevas versiones.

El control de calidad del software involucra un conjunto de actividades sistemáticas que proporcionan evidencia de la aptitud de uso del producto completo. El control de calidad busca ofrecer medidas cuantificables de bondad para la calidad del un sistema software, esas medidas son conocidas como "métricas". Algunas métricas tradicionales son aplicables directamente a los sistemas OO pero otras han tenido que ser particularmente definidas para OO [Ierache, J. 1999].

Se completa con éxito esta fase cuando la funcionalidad y calidad de las versiones son suficientes para expedir el producto.

MANTENIMIENTO: Es la actividad de gestionar la evolución del producto. Es en gran medida una reiniciación del ciclo de las fases anteriores. Se realizan cambios localizados en el sistema, se añaden nuevos requisitos y se eliminan errores persistentes. Inmediatamente tras la entrega del sistema en producción, sus desarrolladores y usuarios finales tendrán probablemente una serie de mejoras o modificaciones que les gustaría realizar. A medida que hay más usuarios usando el sistema, se descubrirán nuevos errores y patrones de uso que el control de calidad no pudo anticipar.

Si se ha hecho un buen trabajo previo el añadido de funcionalidad nueva o la modificación de algún comportamiento existente se hará en forma natural. Las actividades propias de esta fase son:

- Asignar prioridad a las peticiones de mejoras básicas o informes de errores que denotan problemas del sistema, y estimar el costo de volver a desarrollarlo.
- Establecer una colección significativa de estos cambios y tratarlos como puntos funcionales para la nueva versión.
- Gestionar la siguiente evolución de la versión. Esto implica versiones de producción y de depuración de errores.

Se reconoce que sé esta efectuando el mantenimiento del sistema si la arquitectura sigue siendo flexible al cambio. Se reconoce que se precisa un nuevo sistema cuando la respuesta a nuevas mejoras comienza a requerir recursos de desarrollo excesivo y el costo de continuidad del sistema es mayor que desarrollar uno nuevo.

2.5.3 MODELOS BASICOS DEL CV ORIENTADO A OBJETOS

Un modelo es una abstracción de algo, cuyo objetivo es comprenderlo antes de construirlo. Dado que los modelos omiten los detalles no esenciales es más sencillo manipularlos que manipular la entidad original. La abstracción permite enfrentarse a la complejidad. Para construir sistemas complejos, el desarrollador debe abstraer distintas vistas del sistema, construir modelos utilizando notaciones precisas, verificar que los modelos satisfacen los requisitos del sistema y añadir, gradualmente, detalles para transformar los modelos en una implementación.

El paradigma de objetos [Booch, G. 1996], [Rumbaugh, J. y otros 1996], [Martin, J. y Odell, J. 1994], emplea básicamente tres clases de modelos para describir el sistema: el Modelo de Objetos que describe los objetos del sistema y sus relaciones; el Modelo Dinámico que describe las interacciones existentes

entre objetos del sistema; y el Modelo Funcional (o Casos de Uso) que describe las transformaciones de datos del sistema. Todos los modelos son aplicables en la totalidad de las fases del desarrollo y van adquiriendo detalles de implementación a medida que progresá el desarrollo. Un procedimiento típico de software contiene estos tres aspectos:

- utiliza estructuras de datos (modelo de objetos),
- secuencia las operaciones en el tiempo (modelo dinámico) y
- transforma valores (modelo funcional).

El enfoque orientado a objetos se centra primordialmente en identificar objetos procedentes del dominio de la aplicación ajustándoles después los procedimientos porque es necesario describir “qué” está cambiando o transformándose, antes de describir “cuándo” y “cómo” cambia.. Soporta mejor las evoluciones de los requisitos porque está basado en el entorno subyacente del dominio de la aplicación en si, más que en los requisitos funcionales ad-hoc de un único problema.

2.5.3.1 MODELO DE OBJETOS

Describe la estructura estática (de datos), de los objetos del sistema (identidad, atributos y operaciones) y también sus relaciones. El modelo de objetos contiene diagramas de objetos. Un diagrama de objetos es un grafo cuyos nodos son clases de objetos y cuyos arcos son relaciones entre las clases. El diagrama contiene clases de objetos organizados en jerarquías que comparten una estructura y comportamiento comunes y que están asociadas a otras clases. Estas clases definen los atributos que lleva cada instancia de objeto y las operaciones que efectúa o sufre cada uno. En cada instancia de la clase se guardan los valores de esos atributos.

Para la construcción del modelo de objetos los pasos pueden ser:

Primer paso	Se identifican las clases de objetos relevantes en el dominio de la aplicación para lo cual se definen clases de objetos tentativas y luego con el fin de retener las clases correctas se eliminarán todas aquellas consideradas triviales y/o erróneas.
Segundo paso	Se prepara un diccionario de datos, se describe el alcance de la clase dentro del problema estudiado, las asociaciones (relaciones), atributos y operaciones (procesos).
Tercer paso	Se identifican asociaciones entre clases reteniendo aquellas que resulten adecuadas.

	Se reconocen atributos de los objetos descartando aquellos que resulten innecesarios e incorrectos.
Cuarto paso	Se organizan y simplifican las clases de objetos utilizando la herencia para compartir una estructura común.
Quinto paso	Se itera y refina el modelo de objetos En caso de presentarse algún error se debe volver a etapas anteriores, si es necesario, para poder corregirlo. Es poco común que un modelo de objetos sea correcto a la primera pasada.
Sexto paso	Se deben agrupar las clases y asociaciones fuertemente acopladas en módulos Un módulo es un conjunto de clases que captura algún subconjunto lógico del modelo completo.

2.5.3.2 MODELO DINÁMICO

Describe los aspectos de comportamiento (de control) de un sistema, que cambian con el tiempo. El modelo dinámico se utiliza para especificar e implementar los aspectos del control del sistema. Los modelos dinámicos contienen diagramas de estados. Un diagrama de estados es un grafo cuyos nodos son estados y cuyos arcos son transiciones entre estados causadas por sucesos o eventos.

Se especifican en este modelo la temporización y secuencia de operaciones (sucesos que marcan los cambios, secuencias de sucesos, estados que definen el contexto para los sucesos), y la organización de sucesos y de estados. El modelo dinámico captura el control, aquel aspecto de un sistema que describe las secuencias de operaciones que se producen sin tener en cuenta lo que hagan las operaciones, aquello a lo que afecten o la forma en la que estén implementadas. Las acciones de los diagramas de estado se corresponden con funciones procedentes del modelo funcional; los sucesos de un diagrama de estado pasan a ser operaciones que se aplican a objetos dentro del modelo de objetos. Para construir un modelo dinámico se deben seguir los siguientes pasos:

Primer paso	Se preparan escenarios de sesiones típicas y excepcionales: Se comienza preparando escenarios para casos "normales" luego se consideran los casos "especiales" y por último los casos en los que el usuario comenta algún error. Un escenario es una secuencia de sucesos. Éstos se producen siempre que se intercambia información entre un objeto del sistema y un agente externo.
Segundo Paso	Identificación de sucesos que actúen entre objetos y seguimientos

	<p>de sucesos para cada escenario: Entre los sucesos externos contamos todas las señales, entradas, decisiones, interrupciones, transacciones y acciones procedentes o destinadas al usuario o a dispositivos externos.</p> <p>Se deben agrupar con el mismo nombre aquellos sucesos que tengan el mismo efecto sobre el flujo de control, aún cuando difieren los valores de sus parámetros.</p> <p>Un diagrama de flujo de sucesos resume los sucesos habidos entre clases sin tener en cuenta la secuencia e incluyendo sucesos procedentes de todos los escenarios y los sucesos de error.</p>
Tercer Paso	<p>Construcción de un diagrama de estado: Se construye un diagrama de estados para cada clase de un objeto que tenga un comportamiento dinámico no trivial. El mismo deberá mostrar las tramas de sucesos que reciba y envíe junto con las acciones que lleve a cabo.</p>
Cuarto Paso	<p>Correspondencia de sucesos entre objetos: Una vez que los diagramas de estado para todas las clases están completos hay que comparar sucesos entre los mismos para comprobar la congruencia en el nivel del sistema.</p> <p>Las consideraciones a tener en cuenta son:</p> <ul style="list-style-type: none"> • Todo suceso debería tener un emisor y un receptor • Son sospechosos los estados sin predecesores o sucesores • Si se produce una entrada en un momento inoportuno se debe tener cuidado con errores de sincronización <p>"El conjunto de diagramas de estado es lo que constituye el modelo dinámico".</p>

2.5.3.3 Modelo Funcional

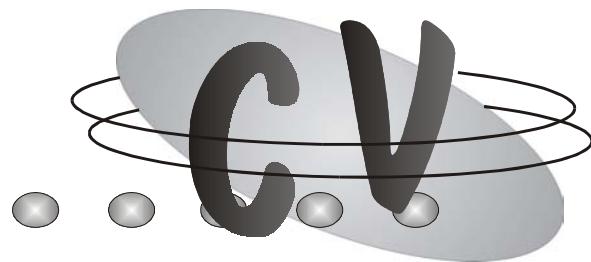
Describe las transformaciones, de valores de datos que ocurren dentro del sistema, captura lo que hace el sistema, independientemente de cuando se haga o de la forma en que se haga. El modelo funcional contiene diagramas de flujo de datos. Un diagrama de flujo de datos es un grafo cuyos nodos son procesos y cuyos arcos son flujos de datos, se muestra las dependencias entre los valores y el cálculo de valores de salida a partir de los de entrada, sin considerar cuándo y dónde se ejecutan las funciones, ni siquiera si llegan a ejecutarse. Las funciones se invocan como acciones en el modelo dinámico y se muestran como operaciones que afectan a objetos en el modelo de objetos.

Para mostrar dependencias funcionales resultan útiles los diagramas de flujo de datos. En un diagrama de flujo de datos sus procesos se corresponden con actividades de los diagramas de estado de las clases y sus flujos de datos con valores de atributos de un diagrama de objetos. Los pasos a seguir para construir un modelo funcional son:

Primer paso	Identificar valores de entrada y de salida: Se comienza por enumerar los valores de entrada y de salida que son los parámetros de los sucesos que se intercambian entre el sistema y el mundo exterior.
Segundo Paso	Construir diagramas de flujo de datos que muestren las dependencias funcionales: Se construye un diagrama de flujo de datos que muestra la forma en que se calcula cada valor de salida a partir de los de entrada. Generalmente los diagramas de flujo de datos suelen construirse por capas y contienen objetos internos de almacenamiento que retienen valores entre iteraciones.
Tercer Paso	Descripción de funciones: Una vez que el Diagrama de flujo de datos se ha refinado lo suficiente se debe describir cada función centrándose en lo que hace y no en la forma de implementarla. Las descripciones especifican las relaciones entre los valores de entrada y los de salida. El propósito es especificar el algoritmo de transformación.

Capítulo 3

Definición del problema



En este capítulo se presenta una definición del objetivo y alcance del proyecto especificando las características del problema a resolver y los participantes. Se describe brevemente la metodología IDEAL que es la seleccionada para desarrollar el sistema. Como cierre del capítulo se presenta un plan del proyecto que es la base sobre la cual se cumplen las actividades de gestión.

3.1 PLANTEAMIENTO DEL PROBLEMA

Una de las tareas iniciales de todo proyecto de software consiste en seleccionar el modelo de ciclo de vida que guiará el desarrollo del proyecto. Esta tarea presenta una serie de dificultades:

- Existen diversos modelos de ciclo de vida: cascada, prototipado de usar y tirar, incremental, emisión gradual, mejora iterativa, ensamblaje de componentes, espiral, prototipado operativo, prototipado rápido, etc. No existe un modelo de ciclo de vida que funcione para cualquier proyecto.
- Algunos de estos modelos tienen características similares.
- El ingeniero software debe estudiar las características del proyecto y seleccionar el modelo de ciclo de vida que más se adapte a ellas. Las bases para determinar el ciclo de vida más adecuado son: la cultura de la organización, la disponibilidad para correr riesgos, el dominio de aplicación, la volatilidad de los requisitos y cuánto se comprenden dichos requisitos.
- Esta selección se realiza al inicio del proyecto y se deben considerar un conjunto de variables de análisis que requiere en su mayor parte de la experiencia del ingeniero en software, ya que al inicio del proyecto subyace una gran cuota de subjetividad en casi todos los aspectos del proyecto.
- La bibliografía disponible, con relación al tema de selección de ciclos de vida, si bien menciona el conjunto de variables a analizar, no presenta un conocimiento sistemáticamente organizado y no especifica cómo debe realizarse el proceso para seleccionar el ciclo de vida para el proyecto. [Juristo Juzgado, N. a 1996], [Juristo Juzgado, N. b 1996], [IEEE, 1991], [ISO, 1994], [Royce, W.W. 1970], [Böehm, B.W. 1987], [Pressman, R.S. 1997], [Rumbaugh, J y otros 1996], [Booch, G. 1996], [Jacobson, I. 1992], [Jacobson, I. y otros 1999]

3.2 OBJETIVO DEL TRABAJO

El objetivo principal de este trabajo consiste en desarrollar un prototipo de Sistema Experto (SE) que asista al Ingeniero en Software en la selección del ciclo de vida más adecuado para el desarrollo de un proyecto de software en particular, colaborando en el análisis de las características mas relevantes a tener en cuenta.

Para alcanzar este objetivo se plantean metas que son:

- Usar las técnicas de Ingeniería del Conocimiento para conocer y documentar los criterios y experiencia utilizados por un grupo de expertos para resolver el problema de selección del modelo de Ciclo de Vida.
- Iniciar el proceso para establecer un marco ordenado de variables a tener en cuenta para decidir el ciclo de vida más conveniente para el desarrollo de un proyecto.
- Utilizar la Metodología Ideal en el desarrollo del Sistema.
- Implementar el sistema usando la herramienta KAPPA-PC.

3.3 ALCANCE

Existen muchos modelos de ciclos de vida disponibles tanto en la bibliografía como de uso particular de alguna organización. Dada la diversa cantidad de modelos y la falta de sistematización y documentación de las variables que deben ser consideradas en la elección del ciclo de vida más adecuado, es necesario fijar una cota en el desarrollo del Sistema Experto, tanto en los modelos como en las variables a considerar.

Se consideran los siguientes ciclos de vida: Cascada, Modelo en Espiral y Modelo Orientado a Objetos. Se encuadra el desarrollo del Sistema Experto teniendo en cuenta los aspectos más sobresalientes de cada modelo. Respecto de las variables, se determinarán las más significativas que resulten del proceso de Adquisición de Conocimientos con los expertos.

3.4 AMBITO DEL PROYECTO

El desarrollo del Sistema Experto involucra a los siguientes participantes:

- *Experto principal:* el experto recomendado para el proyecto es el Prof. Gregorio Perichinsky, Doctor en Ingeniería informática por la Universidad de Buenos Aires (UBA), Director del Laboratorio de Ingeniería de Software y Director del Departamento de Ciencias de la Computación de la Facultad de Ingeniería de la UBA.
- *Grupo de Expertos:* son profesionales de sistemas que lideran el desarrollo de proyectos de software y docentes universitarios que dictan materias en las cuales uno de los temas es ciclos de vida y metodologías para análisis y diseño de sistemas de información.
 - *Lic. Carlos Beltrami:* Gerente de Sistemas de una fábrica de ropa con sedes en todo el país. Titular de la cátedra Sistemas de Información de la Universidad Nacional de Catamarca.
 - *Lic. Carlos Leone:* Consultor y asesor en Informática en el área Servicios Profesionales. Titular de la cátedra Diseño de Sistemas de la Universidad Tecnológica Nacional.
 - *Ing. María Florencia Pollo Cattaneo:* Secretaria técnica del Departamento de Sistemas. Prof. Asociada de la cátedra de Análisis de Sistemas en la Universidad Tecnológica Nacional.
 - *Lic. Laura Lucchini:* Jefe de División Sistemas y Profesora adjunta de la cátedra Seminario de Sistemas en la Universidad Nacional de Luján.
 - *Ing. Mariano Weschler:* Socio gerente de una empresa consultora que desarrolla software para e-business. Docente auxiliar en la cátedra de Análisis de Sistemas en la Universidad Tecnológica Nacional.
 - *Lic. Enrique Fernández:* Jefe de Programadores en la Gerencia de Sistemas del Banco de la Ciudad de Buenos Aires. Auxiliar de cátedra de Seminario de Sistemas en la Universidad Nacional de Luján.
- *Usuarios:* los usuarios son los líderes de proyecto de desarrollo de software. Docentes y alumnos universitarios de las carreras de Informática.
- *Ambito de uso del Sistema Experto:* el área de sistemas de una empresa, empresas de desarrollo de software, cátedras de la universidad.
- *Ambito de desarrollo del Sistema Experto:* el plan de tesis de Magíster de CAPIIS (Carrera de Posgrado en Ingeniería del Software) por acuerdo

entre el ITBA (Instituto Tecnológico de Buenos Aires) y la UPM (Universidad Politécnica de Madrid).

- *Evaluación:* se hará con los usuarios del sistema y con el grupo de expertos. Se cuenta con casos de prueba que aportan los expertos y los propios usuarios.
- *Herramienta:* el prototipo se desarrollará para ser usado en computadoras personales.

3.5 METODOLOGÍA IDEAL

La metodología IDEAL, presenta un ciclo de vida troncocónico en tres dimensiones [Gómez, A. y otros 1997]. Su base es un modelo en espiral [Böehm, B.W. 1987] y la tercera dimensión representa el mantenimiento perfectivo una vez implementado el SE.

El desarrollo del sistema se basa en la filosofía del modelo en espiral en donde cada fase finaliza con un prototipo que conduce a la fase siguiente del ciclo. El mantenimiento perfectivo involucra la incorporación sistemática de nuevos conocimientos (adquiridos por el uso del sistema) típica cualidad de un SE. Los ejes de la base del cono representan el costo y el tiempo.

El eje de la calidad se representa de abajo hacia arriba; va de mayor diámetro, o sea conocimientos menos específicos y de menor calidad, a menor diámetro, conocimientos más exactos y de mayor calidad. Al inicio del funcionamiento del sistema se obtienen grandes cantidades de conocimientos de distinta calidad pero a medida que el sistema se usa el conocimiento se refina, se obtienen menos conocimientos, pero de mayor calidad.

La metodología IDEAL presenta las siguientes fases y etapas [Gómez, A. y otros 1997]:

FASE I: Identificación de la tarea.

I.1. Plan de requisitos y adquisición de conocimientos. Definir con los usuarios cuáles son los objetivos cuantitativos, cualitativos y filosóficos del sistema a desarrollar. Definir las limitaciones de costo y tiempo, tecnología disponible. Sirve de base para el estudio de viabilidad. Para completar el Plan de Requisitos es necesario comenzar con la Adquisición de conocimientos, entrevistando directivos, expertos y usuarios. La Adquisición de conocimientos más profunda corresponde a la Fase II.

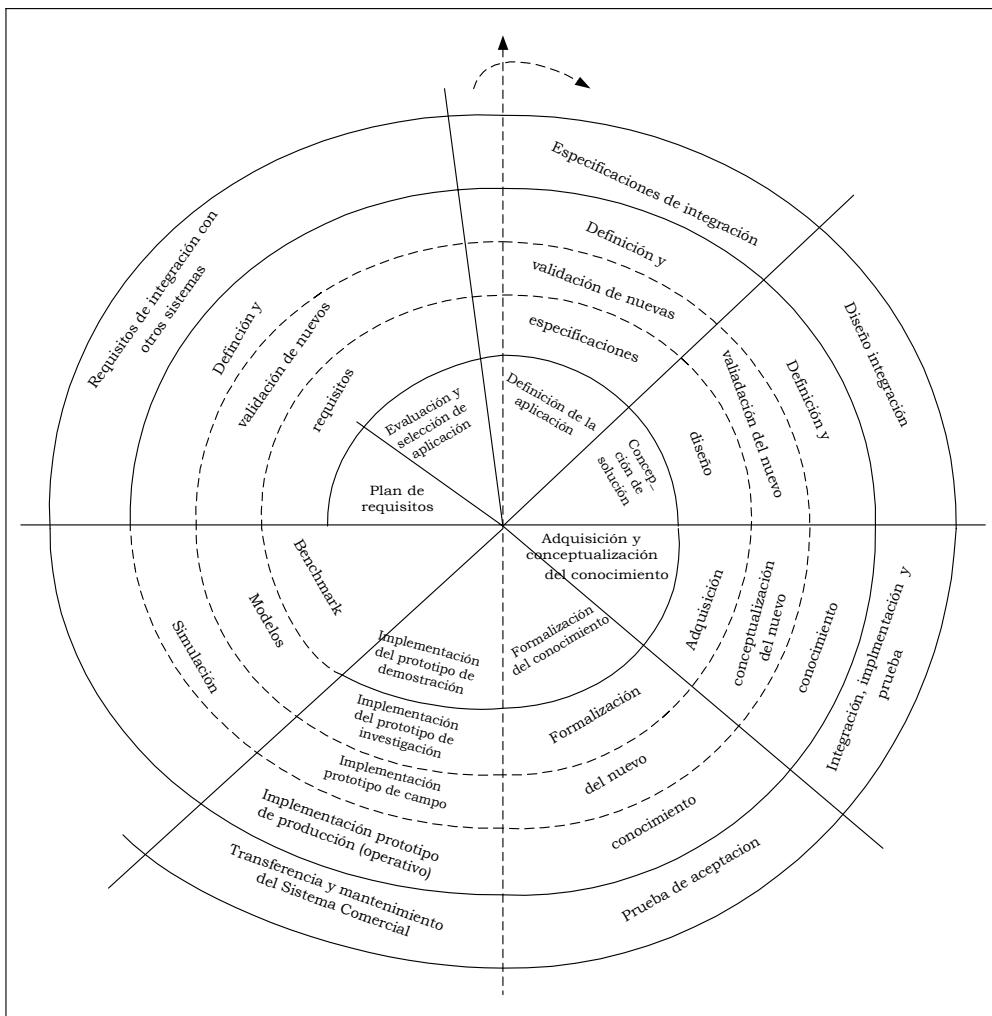


FIGURA 3-1 Modelo troncocónico de la metodología IDEAL- Base

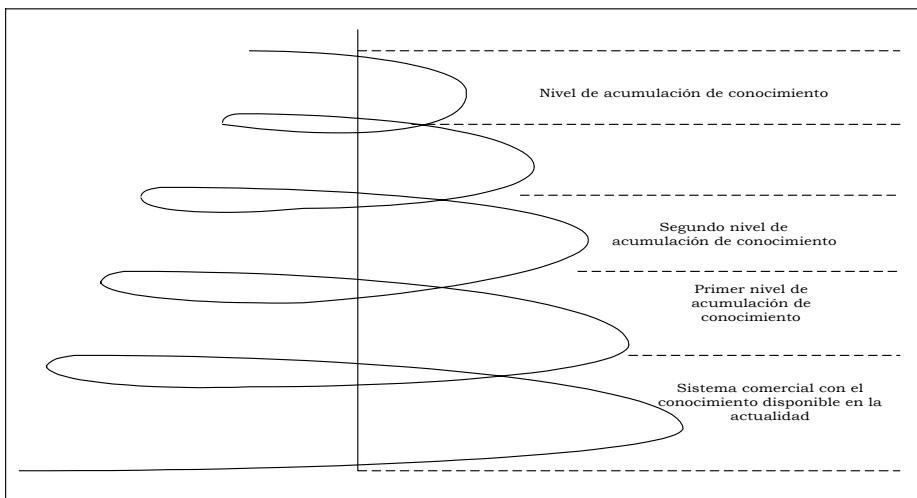


FIGURA 3-2 Modelo troncocónico de la metodología IDEAL- Eje de calidad

I.2. Evaluación y selección de la tarea: Se realiza un estudio de viabilidad del proyecto, desde la perspectiva de la INCO cuantificando la tarea para determinar el grado de dificultad que presenta.

I.3 Definición de las características del sistema: Especificar técnica y formalmente los requisitos del usuario, se pasa de una descripción informal de los requisitos del usuario a una especificación técnica completa hecha por el IC. Se definen los requisitos Funcionales, Operativos, de Interfaz y de Soporte. Se definen criterios de éxito, casos de prueba o ensayo, recursos necesarios, análisis de costo/beneficio, evaluación de riesgos, calendario.

FASE II: Desarrollo de los prototipos.

Las especificaciones iniciales de los sistemas suelen ser incompletas, imprecisas, inconsistentes, contradictorias. Obtener mayor grado de precisión requiere el desarrollo de prototipos que permiten refinar con mayor precisión los requerimientos. Se desarrollan paulatinamente los prototipos de: Demostración, Investigación, Campo y Operación, siendo cada uno refinamiento del anterior. Para ello es necesario realizar ciertas actividades existiendo ligeras diferencias entre las etapas del prototipo de Demostración y los otros.

II.1. Concepción de la solución: Descomposición en Subproblemas y Determinación de analogías. Se produce un diseño general del prototipo. Engloba dos actividades principales, el desarrollo del DFD (diagrama de flujo de datos) y el diseño arquitectónico del sistema.

II.2. Adquisición y Conceptualización de los conocimientos: Se alterna la extracción de los conocimientos públicos de sus fuentes (libros, documentos, manuales de procedimientos, etc.), la educación de los conocimientos privados de los expertos y la conceptualización para modelar el comportamiento del experto.

II.3. Formalización de los conocimientos y Definición de la Arquitectura: Hay dos actividades fundamentales. Una es seleccionar los formalismos para representar en la máquina los conocimientos que conforman la conceptualización obtenida en la etapa anterior. Esta formalización está ligada con los tipos de conocimientos más apropiado y las herramientas disponibles en su desarrollo.

La otra es realizar un diseño detallado, hay que establecer los módulos que definen el motor de inferencias, la base de conocimientos y las interfaces con el usuario y con otros sistemas.

II.4 Selección de la Herramienta e Implementación: Si la herramienta de desarrollo previamente seleccionada es adecuada la implementación es inmediata y automática, caso contrario es similar a cualquier implementación de software, en donde será necesario realizar la programación.

II.5. Validación y evaluación del prototipo: La fiabilidad de los resultados es una de los puntos principales de un SE. Estos sistemas dependen en gran medida del contexto y eso dificulta la determinación de las validaciones.

Se definen casos de prueba, juegos de ensayo o ensayo en paralelo. Se examina la interfaz de usuario.

II.6. Definición de nuevos requisitos y diseño: Es la definición de los requisitos, especificaciones y diseño del prototipo siguiente, para ello deben repetirse las etapas II.1 a II.6.

FASE III: Ejecución de la construcción del sistema integrado.

Los SE forman parte de sistemas software con los que interactúan.

III.1. Requisitos y diseño de la integración con otros sistemas. Es el estudio y diseño de interfaces con otros sistemas, tanto en hardware como en software.

III.2 Implementación y evaluación del Sistema Integrado: Se implementa la integración del SE con los otros sistemas existentes hardware y software para conseguir un sistema final.

III.3. Aceptación del sistema por el usuario: Es la última prueba por parte de los usuarios y expertos, que debe satisfacer todos los requerimientos, tanto en su fiabilidad como en su eficiencia.

FASE IV: Actuación para conseguir el mantenimiento perfectivo.

IV.1. Definir el mantenimiento del sistema global: Se emplean las técnicas habituales de IS para establecer el mantenimiento correctivo (corrección de errores) y el mantenimiento perfectivo (aumento de funcionalidad del sistema).

IV.2. Definir el mantenimiento de las Bases de Conocimientos: Considera la incorporación de nuevos conocimientos, que se generan por el propio uso del sistema. El análisis de protocolos, como forma de adquisición de conocimientos es imprescindible. En consecuencia, es necesario mentalizar a los expertos y usuarios finales del sistema en el uso de dicho protocolos.

V.3. Adquisición de nuevos conocimientos y Actualización del sistema: El mantenimiento perfectivo es esencial ya que además del aumento de funcionalidades efectúa la incorporación de nuevos conocimientos que se generan por el propio uso del SBC. Este mantenimiento se ve reflejado en la tercera dimensión de la espiral troncocónica. Hay que establecer métodos para actualizar el sistema incorporando los conocimientos adquiridos.

FASE V: Lograr una adecuada transferencia tecnológica.

Esto es especialmente crítico en los SBC en donde se han detectado diferencias considerables cuando los manejan sus diseñadores con respecto a su uso rutinario por los usuarios finales.

V.1. Organizar la transferencia tecnológica: Definir sesiones de entrenamiento de los diseñadores con los usuarios, tanto para explicar el manejo del propio sistema como para manejar y entender la documentación.

V.2. Completar la documentación del sistema: Realizar un manual de usuario amigable.

3.6 GESTION DEL PROYECTO

La gestión del proyecto implica actividades de planificación, estimación de recursos, control de configuración y evaluación del proyecto.

3.6.1 PLANIFICACION

La planificación consiste en definir las actividades a realizar, la duración de las actividades, la concurrencia y solapamiento de las mismas a través de la red de actividades.

Se adjunta al final de este capítulo el Plan General de Gestión del Proyecto donde se presentan las actividades a ser desarrolladas a lo largo del proyecto. El desarrollo del presente trabajo de Tesis de Master abarca todas las Fases de la metodología IDEAL, con excepción de las etapas III.1: Requisitos y diseño de la integración con otros sistemas y III.2: Implementación y evaluación del Sistema Integrado, ya que no existe un sistema al que se deba integrar.

3.6.2 ESTIMACION DE RECURSOS

La estimación es predecir los recursos humanos, técnicos y económicos para cumplir con las actividades planificadas.

Los recursos humanos son:

- El Experto principal: Prof. Perichinsky
- El Grupo de Expertos: Lic. Beltrami, Lic. Leone, Ing. Pollo Cattaneo, Lic. Lucchini, Ing. Weschler, Lic. Fernández
- Ingeniero en Conocimiento: Lic. Rossi

Los recursos técnicos son:

- Computador personal: Pentium, Ram 160, Disco Rígido 3Gb, CD 24X, Impresora Deskjet 680
- Software a usar : Kappa PC

Los recursos económicos son:

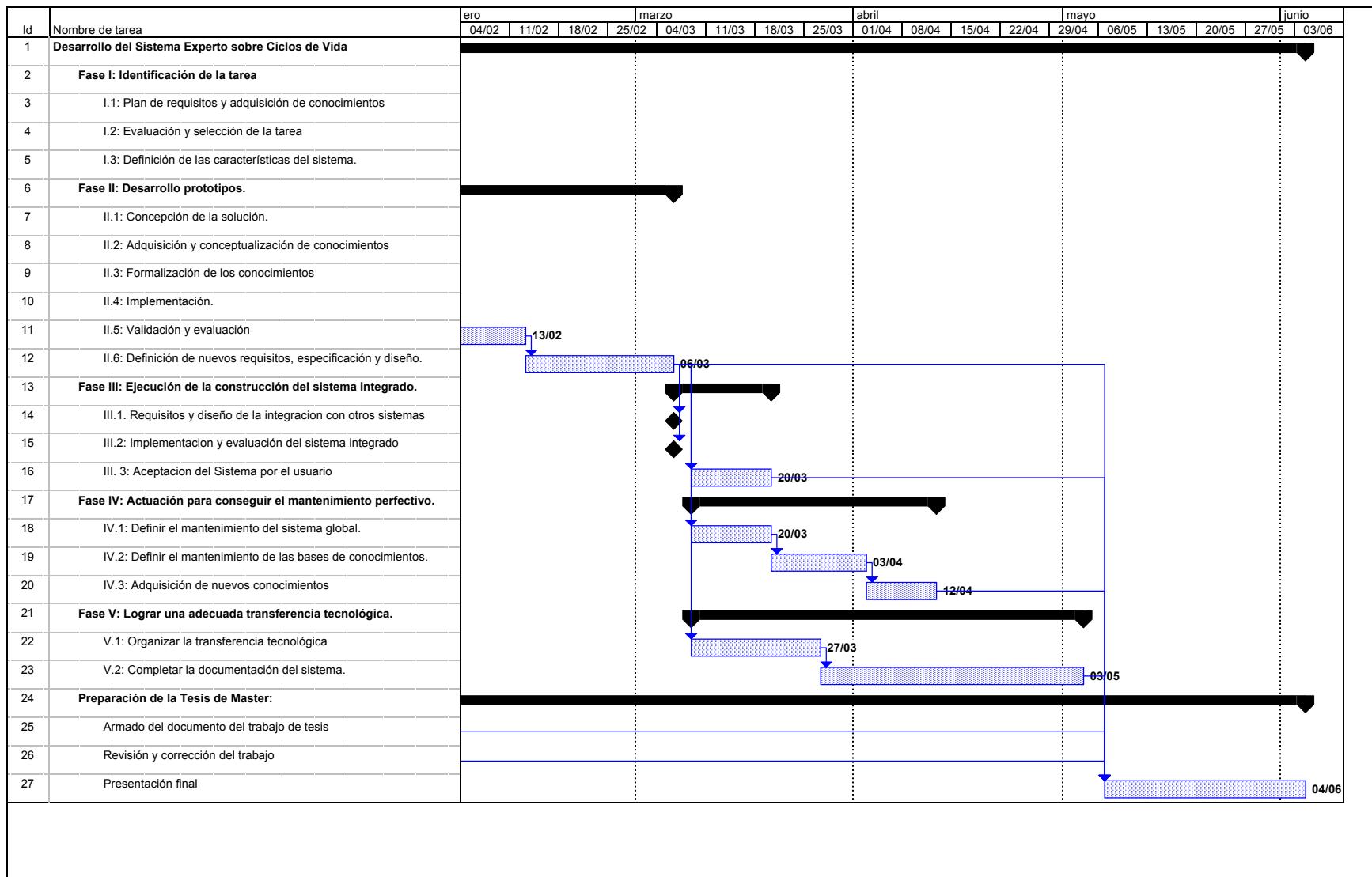
- Financiamiento parcial del proyecto por el Instituto Tecnológico de Buenos Aires, con un monto asignado en horas de trabajo equivalente a 8760 dólares.

3.6.3 CONTROL DE CONFIGURACION

Es una actividad cuya misión es controlar la evolución de un sistema software, y por lo tanto se debe realizar a lo largo de todo el ciclo de vida desde la Identificación de la tarea, incluyendo el Mantenimiento perfectivo, hasta que el producto se retira. En el capítulo 12 se documenta el Control de configuración definido para el presente proyecto.

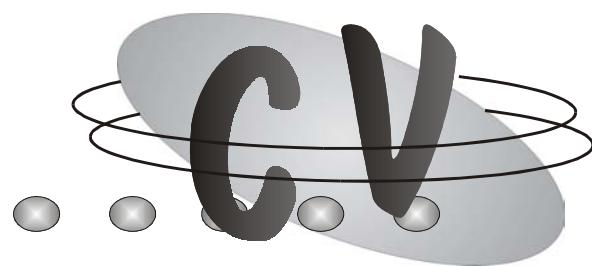
3.6.4 EVALUACION DEL PROYECTO

La evaluación del proyecto es un conjunto de actividades que se realizan conjuntamente a medida que se van completando las fases de desarrollo establecidas en la Planificación. El proceso de Evaluación se encuentra descripto detalladamente en el capítulo 9, Evaluación.



Capítulo 4

Estudio de Viabilidad



En el presente capítulo se describe el Test de Viabilidad que propone la Metodología IDEAL. Luego se detalla el análisis de viabilidad para el proyecto, justificando los valores definidos en el test. A continuación se evalúa la viabilidad del proyecto

4.1 TEST DE VIABILIDAD

El estudio de viabilidad permite determinar si la solución de un problema es viable con sistemas informáticos basados en conocimiento, o no.

La metodología IDEAL propone un test de viabilidad en el que se analizan 4 factores de viabilidad. El desarrollo de un SE debe considerar si es posible, está justificado, es apropiado y si va a tener éxito.

- *Plausibilidad:* determina si se cuenta con los medios necesarios para poder abordar el problema desde la Ingeniería del Conocimiento (INCO). Para ello se analizan dos aspectos básicos: las características del experto y las características de la tarea que lleva adelante el experto.
Del experto se analiza si existe, si es reconocido como tal por sus colegas, si es cooperativo y si es capaz de articular sus métodos y procedimientos de trabajo. Es importante que el experto haya resuelto el problema con suficiente frecuencia y que esté realmente interesado en el desarrollo del sistema.
De la tarea se analiza su grado de dificultad, si está adecuadamente estructurada y qué tipo de habilidades se requieren para realizarla.
- *Justificación:* determina si se justifica utilizar sistemas expertos. Se analizan los aspectos de la necesidad de la experiencia y de la inversión a realizar.
De la experiencia se analizan las características del entorno en donde hay que realizar la tarea (peligrosidad, distancia) y la escasez o posible pérdida de la experiencia.
De la inversión se analizan los costos del sistema, la recuperación de la inversión, el valor de la tarea a resolver y si existen soluciones alternativas.
- *Adecuación:* determina si la INCO permite abordar el problema. Existen problemas que no son adecuados para ser resueltos por la INCO ya que son resolubles por algoritmos convencionales o que requieren de sentido común. Se analiza la naturaleza de la tarea, la complejidad de la tarea y el tipo de tarea.

- **Éxito:** determina si existe colaboración y predisposición por parte del experto para el desarrollo del sistema, convicción de los directivos para afrontar los costos del desarrollo y predisposición de los usuarios en general.

Para estudiar la viabilidad del proyecto se utiliza el método propuesto en la metodología IDEAL. El método propuesto consiste en completar una tabla donde se valoran un conjunto de variables para luego estimar si el sistema experto es posible, está justificado, es adecuado y tendrá éxito. El método es de tipo métrico, usa ponderaciones, como métrica utiliza la media armónica e incorpora la manipulación de valores lingüísticos mediante intervalos difusos.

Los valores que pueden asignarse a las variables son de diferente tipo:

- **Numéricos:** Valores discretos en un intervalo [1,10]
- **Booleanos:** Valores: Sí – No. Su representación mediante intervalos difusos es:

Sí	10	10	10	10
No	0	0	0	0

- **Lingüísticos:** Valores: Nada, poco, regular, mucho, todo. Son traducidos a valores difusos para poder realizar los cálculos correspondientes, de acuerdo con la siguiente tabla.

Nada	0	0	1.2	2.2
Poco	1.2	2.2	3.4	4.4
Regular	3.4	4.4	5.6	6.6
Mucho	5.6	6.6	7.8	8.8
Todo	7.8	8.8	10	10

El procedimiento para aplicar la técnica es el siguiente:

Paso 1: completar la columna VALOR en la tabla.

- En las filas se detallan cada una de las características consideradas para establecer la viabilidad del proyecto.
- Los valores de las columnas, excepto el último (que corresponde a la columna VALOR), están predeterminados en función de la experiencia de los autores que proponen el método. Puede modificarse alguno de

ellos si el INCO lo cree necesario. Las columnas se dividen de la siguiente manera:

- ✓ **CATEGORIA:** especifica con quien está relacionada la característica. Valores EXPERTO, TAREA, DIRECTIVO/USUARIO
- ✓ **DIMENSION:** referencia con qué grupo básico está relacionada la característica. Valores Pn (Posible), Jn (Justificado), An (Adecuado), En (Exitoso)
- ✓ **PESO:** valora la importancia relativa de esa característica con relación al test, valores 1 a 10. Valora si la característica favorece o disminuye el grado de interés en el desarrollo del SE, valores +, -. Estos valores los fija el INCO.
- ✓ **TIPO:** referencia la importancia de la característica en el proyecto. Valores ESENCIAL, DESEABLE.
- ✓ **NATURALEZA:** referencia el tipo de valores de la característica. DIFUSA, NUMÉRICA, o BOOLEANA.
- ✓ **UMBRAL:** referencia al valor mínimo que es necesario superar para esa característica, si es esencial.
- ✓ **VALOR:** asignado por quien está evaluando el proyecto.

Paso 2: calcular el valor de cada dimensión. Para las dimensiones de plausibilidad, adecuación y éxito se usa la fórmula:

$$VC_i = \frac{1}{2} \frac{\sum_{k=1}^{r_i} P_{ik}}{\sum_{k=1}^{r_i} V_{ik}} + \frac{1}{2} \frac{\sum_{k=1}^{r_i} P_{ik} V_{ik}}{\sum_{k=1}^{r_i} P_{ik}}$$

VC_j : Valor global de la aplicación para un factor dado.

V_{ik} : Valor de la característica k en la dimensión i .

P_{ik} : Peso de la característica k en la dimensión i .

r_i : Número de la característica en la dimensión i .

Para la dimensión de justificación se toma el valor máximo de los valores asignados a cada una de las características de ese factor al multiplicar por el peso asociado y luego se calcula la aproximación numérica de los intervalos difusos y se toma el máximo.

Paso 3: calcular el valor final. Consiste en el cálculo de la media aritmética ponderada del resultado obtenido para cada dimensión. Los valores de ponderación son:

Plausibilidad	8
Adecuación	8
Justificación	3
Exito	5

$$V_f = \frac{\sum_{i=1}^4 P_i V_i}{\sum_{i=1}^4 P_i}$$

El desarrollo del Sistema Experto se considera viable si el valor final es igual o mayor a 6.

4.2 ANALISIS DE VIABILIDAD PARA EL PROYECTO

A continuación se enumeran todas las características del test de viabilidad y se acompaña una breve justificación del valor asignado a cada una.

4.2.1 JUSTIFICACIÓN DE LA DIMENSION PLAUSIBILIDAD

P1: *Existen expertos, están disponibles y son cooperativos.*

Valor: SI

Existe un grupo de expertos en el ámbito empresarial y académico, que se desempeñan laboralmente, liderando proyectos de desarrollo de sistemas de información y de investigación aplicada, interesados en la construcción del sistema, como apoyo en su tarea de gestión al inicio del proyecto.

Existe un grupo de expertos, que se desempeñan en el ámbito universitario, en las cátedras de análisis y diseño de sistemas de información interesados en la construcción del sistema, como apoyo para la práctica del tema.

Existe un experto principal Prof. Perichinsky dispuesto a cooperar en forma directa durante el desarrollo del sistema con quien se ha establecido ya un cronograma semanal de encuentros.

P2: *El experto es capaz de estructurar sus métodos y procedimientos de trabajo.*

Valor: MUCHO

Los expertos han demostrado ser ordenados en sus procedimientos de trabajo, a lo que se suma la habilidad como docente en la didáctica que manifiestan en sus explicaciones. El grupo de expertos universitarios tiene habilidades para facilitar

las explicaciones. El grupo de expertos empresariales es ordenado en sus procedimientos de trabajo.

Se ha considerado particularmente en la evaluación de esta característica el grado de dificultad que se presenta al ser un grupo de expertos los consultados.

P3: La tarea está bien estructurada y se entiende.

Valor: REGULAR

La tarea es clara, se entiende y está poco estructurada. Existe bastante información disponible de las variables a considerar, pero no están estructuradas con precisión. Es justamente uno de los puntos que interesa especialmente al conjunto de expertos: que el desarrollo del sistema introducirá una propuesta para estructurar con mayor precisión el conjunto de variables a evaluar al seleccionar un ciclo de vida.

P4: Existen suficientes casos de prueba y sus soluciones asociadas.

Valor: 10

El conjunto de expertos cuenta con suficientes casos de prueba. Por un lado un amplio conjunto de proyectos de desarrollo que pueden aportar los líderes de proyecto, tanto aquellos que ya han sido desarrollados, como los que recién están en sus inicios. Por otro lado un amplio conjunto de casos de prueba y estudio que pueden aportar los docentes. En el capítulo 9, Evaluación del sistema se detallan los casos de prueba facilitados por los expertos.

P5: La tarea sólo depende de los conocimientos y no del sentido común.

Valor: 8

La tarea depende del análisis y estimación de un conjunto de características que presenta un proyecto de desarrollo de sistemas, al momento de inicio. Los conocimientos radican en cuáles son las variables a analizar y en la estimación del valor que se les asigna, que es justamente resultado de la experiencia de haber realizado la tarea en varias oportunidades y con diferentes proyectos.

4.2.2 JUSTIFICACION DE LA DIMENSION JUSTIFICACION

J1: Resuelve una tarea útil y necesaria.

Valor: MUCHO

Los profesionales a cargo de proyectos, deben decidir el ciclo de vida a seguir en un momento inicial del desarrollo, donde la información acerca del proyecto es incompleta, la herramienta colabora en revisar el conjunto de variables a tener en cuenta para tomar la decisión.

Los docentes contarán con una herramienta que facilita el entrenamiento y comprensión del tema por parte de los estudiantes.

J2: Se espera una alta tasa de recuperación de la inversión.

Valor: 7

La recuperación de la inversión se espera en términos de una mejor productividad, dado que seleccionando el ciclo de vida adecuado de acuerdo con las características del proyecto, contribuye a un mejor plan de desarrollo y, por lo tanto, a un ajuste más adecuado de plazos y esfuerzos previstos. La herramienta contribuye a revisar cuáles son las características a considerar.

J3: Hay escasez de experiencia humana.

Valor: MUCHO

No es frecuente encontrar expertos en la selección de ciclos de vida. Se requiere personas que hayan trabajado como líderes de proyecto en una gran cantidad de proyectos de software, que puedan estructurar con precisión las características a evaluar.

J4: Hay necesidad de tomar decisiones en situaciones críticas o ambientes hostiles, penosos y/o poco gratificantes.

Valor: REGULAR

El trabajo se realiza en ambientes adecuados, las situaciones no son hostiles ni penosas. Puede considerarse situación crítica ya que la decisión hay que tomarla en un momento inicial cuando aún no se dispone de toda la información respecto del proyecto, o la información que se dispone es incompleta.

J5: Hay necesidad de distribuir los conocimientos.

Valor: MUCHO

Se espera que el sistema pueda ser utilizado por la comunidad informática tanto empresarial como académica.

J6: Los conocimientos pueden perderse de no realizarse el sistema.

Valor: REGULAR

El sistema se basa parte en bibliografía, artículos y sobre la base de la experiencia de un grupo de expertos. Dado que el conocimiento no se encuentra precisamente estructurado se pierde la oportunidad de definir con mayor claridad los conocimientos para realizar la tarea.

J7: No existen soluciones alternativas.

Valor: SI

En la actualidad la tarea se resuelve sobre la base de experiencia humana. Por las características de la tarea se intenta emular el comportamiento humano al tomar una decisión. En este contexto las técnicas tradicionales de desarrollo de software son inadecuadas.

4.2.3 JUSTIFICACION DE LA DIMENSION ADECUACION

A1: *La transferencia de experiencia entre humanos es factible.*

Valor: MUCHO

La transferencia de conocimientos entre humanos es factible. El grupo de expertos docentes tiene habilidad y experiencia en la transferencia de conocimientos. El grupo de expertos que se desempeña liderando proyectos, cuenta con habilidad para la transferencia de conocimientos por ser ésta una de las actividades propias de la coordinación. El experto principal es capaz de transmitir los conocimientos.

A2: *La tarea requiere “experiencia”.*

Valor: TODO

En la actualidad la tarea se resuelve sobre la base de experiencia humana. Se requieren personas que hayan trabajado como líderes de proyecto en una gran cantidad de proyectos de software y que puedan estructurar con precisión las características a evaluar.

A3: *Los efectos de la introducción del SE no pueden preverse.*

Valor: POCO

Se espera que la introducción del SE no depare efectos no deseados. La tarea que realizará será asistir a los profesionales al inicio del proyecto aportando confiabilidad en su trabajo. No existen impedimentos éticos, ni legales que impidan el desarrollo del sistema.

A4: *La tarea requiere razonamiento simbólico.*

Valor: MUCHO

La tarea requiere de razonamiento simbólico para su realización.

A5: *La tarea requiere el uso de “heurísticas” para acotar el espacio de búsqueda.*

Valor: MUCHO

Los conocimientos necesarios son estimaciones en algunos casos subjetivas, por lo que se necesitan heurísticas para precisar las características a analizar y evitar un exhaustivo análisis que entorpezca o confunda al usuario en el momento de decidir el ciclo de vida más adecuado.

A6: *La tarea es de carácter práctico y más táctica que estratégica.*

Valor: SI

La tarea es táctica, está orientada a definir los valores estimados de las características que presenta un proyecto de desarrollo de sistemas de información en particular.

A7: *Se espera que la tarea continúe sin cambios significativos durante un largo período de tiempo.*

Valor: MUCHO

Se espera que la tarea en su esencia no cambie a corto plazo. El concepto de ciclo de vida fue presentado en la década de los 70. A lo largo de los años han aparecido otras propuestas de ciclo de vida, pero la tarea no ha cambiado en esencia. Los ciclos de vida que se han presentado posteriormente satisfacen características relacionadas con el incremento en la variedad de la demanda de proyectos informáticos. Justamente el Sistema Experto desarrollado con la metodología IDEAL considera el mantenimiento perfectivo lo que favorece la incorporación de nuevas características a analizar.

A8: *Se necesitan varios niveles de abstracción en la resolución de la tarea.*

Valor: POCO

No son necesarios demasiados niveles de abstracción para resolver la tarea. Los conocimientos necesarios no son complejos, los temas son estudiados en los primeros años de una carrera universitaria.

A9: *El problema es relativamente simple o puede descomponerse en subproblemas.*

Valor: MUCHO

El problema se puede descomponer en subproblemas. Es posible establecer un agrupamiento de las características a evaluar y analizar cada subgrupo de variables. Es justamente una de las primeras metas de la fase de adquisición del conocimiento el establecer conjuntos de características por área.

A10: *El experto no sigue un proceso determinista en la resolución del problema.*

Valor: SI

El proceso de razonamiento implica analizar y evaluar las características de cada proyecto en particular. Cada proyecto es un problema diferente y su resolución demanda un nuevo análisis.

A11: *La tarea acepta la técnica de prototipado gradual.*

Valor: SI

Dado que el problema puede descomponerse en subproblemas y que los requisitos iniciales no están del todo claros es adecuado considerar el desarrollo del sistema con un prototipo incremental.

A12: *El experto resuelve el problema a veces con información incompleta o incierta.*

Valor: TODO

El experto suele resolver el problema contando con información incompleta o incierta, puesto que la decisión hay que tomarla al inicio del proyecto, cuando los requerimientos, lo que se espera del sistema y los recursos, aún no se encuentran definidos con precisión. Es decir, todas las decisiones del proyecto deben tomarse con posterioridad a la selección del ciclo de vida, lo que implica que la información con que cuenta el experto es incompleta o incierta.

A13: *Es conveniente justificar las soluciones adoptadas.*

Valor: TODO

Es muy importante justificar las soluciones propuestas por el sistema, explicando al usuario el razonamiento aplicado y el fundamento de la conclusión.

A14: *La tarea requiere investigación básica.*

Valor: NO

La tarea no requiere de investigación básica.

A15: *El sistema funcionará en “tiempo real” con otros programas o dispositivos.*

Valor: NADA

El sistema no trabajará en tiempo real con otros programas o dispositivos.

4.2.4 JUSTIFICACION DE LA DIMENSION EXITO

E1: *Existe una ubicación idónea para el SE.*

Valor: TODO

El sistema estará ubicado en el área de desarrollo de sistemas de información, particularmente a disposición de los líderes de proyecto.

En el ámbito académico el área son las cátedras de análisis y diseño de sistemas, preparación y evaluación de proyectos, gestión y control de proyectos informáticos, para ser usados por docentes y alumnos.

E2: *Problemas similares se han resuelto mediante INCO.*

Valor: SI

Otros problemas similares, de apoyo a la tarea del líder de proyecto en las primeras actividades del desarrollo de sistemas, están siendo resueltos por SE.

E3: *El problema es similar a otros en los que resultó imposible aplicar esta tecnología.*

Valor: NO

Se desconoce la existencia de antecedentes que demuestren haber intentado resolver problemas similares mediante la INCO y que se haya fracasado.

E4: *La continuidad del proyecto está influenciada por vaivenes políticos.*

Valor: POCO

No se estima que la continuidad del proyecto esté influenciada por cambios políticos.

E5: *La inserción del sistema se efectúa sin traumas, es decir, apenas se interfiere en la rutina cotidiana.*

Valor: MUCHO

El sistema no interfiere ni genera cambios en la estructura del trabajo cotidiano. La herramienta colabora orientando en una de las primeras decisiones por lo que es considerada como una colaboración. De todas formas hay que considerar que alguna adaptación siempre es necesaria.

E6: *Se dispone de experiencia en INCO.*

Valor: REGULAR

No existe experiencia previa directa, pero se cuenta con el asesoramiento de los tutores y con conocimientos en el manejo de las técnicas y la metodología de desarrollo.

E7: *Se dispone de los recursos humanos, hardware y software necesarios para el desarrollo e implementación del sistema.*

Valor: TODO

Se cuenta con recursos de hardware y software adecuados, material bibliográfico y la disposición del personal universitario y de los expertos que colaboran gustosamente, como así también de los tutores asesores. También se cuenta con la colaboración de programadores en el uso de la herramienta Kappa.

E8: *El experto resuelve el problema en la actualidad.*

Valor: TODO

Los expertos tanto del área académica como empresarial resuelven el problema cotidianamente.

E9: La solución del problema es prioritaria para la institución.

Valor: MUCHO

La solución del problema es parte de una rama de investigación que se ha comenzado y se desea continuar con futuros trabajos que asistan a los desarrolladores de sistemas informáticos a lo largo de las actividades del proyecto.

E10: Las soluciones son explicables.

Valor: MUCHO

El sistema debe explicar cada solución adoptada ya que un buen nivel de trazabilidad ayuda al usuario a continuar el proceso y mejorar la calidad del SE.

E11: Los objetivos del sistema son claros y evaluables.

Valor: MUCHO

Los objetivos del sistema están claramente establecidos y pueden ser evaluados mientras se desarrolla el sistema y al final del desarrollo. Los objetivos son profundizados y especificados con mayor precisión a lo largo del desarrollo del sistema, considerando que se ha optado para su construcción por prototipado incremental.

E12: Los conocimientos están repartidos entre un conjunto de individuos.

Valor: REGULAR

Los conocimientos se encuentran repartidos entre un conjunto de expertos, no porque esté parcializado sino porque no está estructurado con precisión y se necesita justamente de la experiencia conjunta para proponer una estructura organizativa básica. Todos los expertos poseen un nivel similar de conocimientos. Se cuenta además con un experto principal que tendrá la última palabra en casos de discrepancia.

E13: Los directivos, usuarios, expertos e IC están de acuerdo en las funcionalidades del SE.

Valor: MUCHO

Todos los integrantes del proyecto están de acuerdo en las funciones principales que cumplirá el Sistema Experto. La funcionalidad será definida con mayor precisión a lo largo del desarrollo del sistema, considerando que se ha optado para su construcción por prototipado incremental.

E14: La actitud de los expertos ante el desarrollo del sistema es positiva y no se sienten amenazados por el proyecto.

Valor: TODO

Los expertos esperan con agrado el desarrollo del sistema. Estiman que será un aporte para mejorar su tarea. En el ámbito educativo el sistema permitirá implementar más y mejores casos de práctica. No consideran que el Sistema Experto signifique una amenaza sino por el contrario permitirá realizar la tarea con una base más estructurada.

E15: *Los expertos convergen en sus soluciones y métodos.*

Valor: REGULAR

Los expertos poseen una formación similar y adoptan soluciones convergentes. Se espera que existan puntos de diferencia que enriquezcan la estructura básica de conocimiento a definir, y todos aceptan la opinión final del experto principal como válida en casos de discrepancia.

E16: *Se acepta la planificación del proyecto propuesta por el IC.*

Valor: SI

Fue aceptada y consensuada con todos los involucrados.

E17: *Existen limitaciones estrictas de tiempo en la realización del sistema.*

Valor: POCO

No existen limitaciones estrictas y rígidas, pero es deseable finalizarlo en el tiempo previsto.

E18: *La dirección y usuarios apoyan los objetivos y directrices del proyecto.*

Valor: MUCHO

La dirección apoya firmemente el proyecto por tratarse de parte de un proyecto de investigación en el cual el objetivo es desarrollar Sistemas Expertos que apoyen la tarea de los profesionales informáticos en el desarrollo de sistemas de información. Los usuarios están interesados en el proyecto porque el SE introduciría una estructura más ordenada que facilite la decisión a tomar y permitiría mayor calidad de práctica para los estudiantes.

E19: *El nivel de formación requerido por los usuarios del sistema es elevado.*

Valor: MUCHO

Está dirigido a ingenieros en software o con experiencia similar.

E20: *Las relaciones IC-Experto son fluidas.*

Valor: MUCHO

Las relaciones con el grupo de expertos son fluidas, se garantiza al menos un encuentro semanal. La relación con el experto principal es directa y continua, están previstos encuentros frecuentes, para los momentos más intensos del proceso de Adquisición y Conceptualización de Conocimientos.

E21: *El proyecto forma parte de un camino crítico con otros sistemas.*

Valor: NO

El proyecto no forma parte de un camino crítico con otros sistemas.

E22: *Se efectuará una adecuada transferencia tecnológica.*

Valor: MUCHO

Está previsto un plan de capacitación, los manuales para el uso del sistema, y un período de seguimiento en los primeros meses de uso.

E23: *Lo que cuenta en la solución es la calidad de la respuesta.*

Valor: SI

La calidad de la respuesta es muy importante de tal modo que el usuario encuentre satisfactorios los resultados que el SE le presente.

4.3 CALCULO DE VIABILIDAD PARA EL PROYECTO

Paso 1: completar la columna VALOR en las tablas.

- En la tabla 4.3.1 se presentan los valores asignados al Factor Plausibilidad.
- En la tabla 4.3.2 se presentan los valores asignados al Factor Justificación.
- En la tabla 4.3.3 se presentan los valores asignados al Factor Adecuación.
- En la tabla 4.3.4 se presentan los valores asignados al Factor Éxito.

Paso 2: calcular el valor para cada dimensión.

- En la tabla 4.3.5 se presenta el cálculo del Factor de Plausibilidad.
- En la tabla 4.3.6 se presenta el cálculo del Factor de Justificación.
- En la tabla 4.3.7 se presenta el cálculo del Factor de Adecuación.
- En la tabla 4.3.8 se presenta el cálculo del Factor de Éxito.

Paso 3: calcular el valor final. Consiste en el cálculo de la media aritmética ponderada del resultado obtenido para cada dimensión.

- En la tabla 4.3.9 se presentan el cálculo del Valor Final

TABLA 4.3.1 DIMENSION DE PLAUSIBILIDAD

Denominación de la Característica	Categoría	Dimensión	Peso	Tipo	Naturaleza	Umbral	Valor
Existen expertos, están disponibles y son cooperativos.	Experto	P1	10	Esencial	Booleana	Sí (sí)	SI
El experto es capaz de estructurar sus métodos y procedimientos de trabajo.	Experto	P2	7	Deseable	Difusa	No	MUCHO
La tarea está bien estructurada y se entiende.	Tarea	P3	8	Deseable	Difusa	No	REGULAR
Existen suficientes casos de prueba y sus soluciones asociadas.	Tarea	P4	10	Esencial	Numérica	Sí (8)	10
La tarea sólo depende de los conocimientos y no del sentido común.	Tarea	P5	9	Deseable	Numérica	No	8

TABLA 4.3.2 DIMENSION DE JUSTIFICACIÓN

Denominación de la Característica	Categoría	Dimensión	Peso	Tipo	Naturaleza	Umbral	Valor
Resuelve una tarea útil y necesaria.	Tarea	J1	8	Deseable	Difusa	No	MUCHO
Se espera una alta tasa de recuperación de la inversión	Directivos / Usuarios	J2	7	Deseable	Numérica	No	7
Hay escasez de experiencia humana.	Experto	J3	6	Deseable	Difusa	No	MUCHO
Hay necesidad de tomar decisiones en situaciones críticas o ambientes hostiles, penosos y/o pocos gratificantes.	Tarea	J4	10	Deseable	Difusa	No	REGULAR
Hay necesidad de distribuir los conocimientos.	Tarea	J5	10	Deseable	Difusa	No	MUCHO
Los conocimientos pueden perderse de no realizarse el sistema.	Experto	J6	10	Deseable	Difusa	No	REGULAR
No existen soluciones alternativas.	Tarea	J7	8	Esencial	Booleana	Sí (sí)	SI

TABLA 4.3.3 DIMENSION DE ADECUACION

Denominación de la Característica	Categoría	Dimensión	Peso	Tipo	Naturaleza	Umbral	Valor
La transferencia de conocimientos entre humanos es factible.	Tarea	A1	7	Deseable	Difusa	No	MUCHO
La tarea requiere "experiencia".	Tarea	A2	10	Deseable	Difusa	No	TODO
Los efectos de la introducción del SE no pueden preverse.	Tarea	A3	-2	Deseable	Difusa	No	POCO
La tarea requiere razonamiento simbólico.	Tarea	A4	5	Deseable	Difusa	No	MUCHO
La tarea requiere el uso de heurística para acotar el espacio de búsqueda.	Tarea	A5	7	Deseable	Difusa	No	MUCHO
La tarea es de carácter práctico y más táctica que estratégica.	Tarea	A6	8	Deseable	Booleana	No	SI
Se espera que la tarea continúe sin cambios significativos durante un largo período de tiempo.	Tarea	A7	8	Esencial	Difusa	Sí (mucho)	MUCHO
Se necesitan varios niveles de abstracción en la resolución de la tarea.	Tarea	A8	8	Deseable	Difusa	No	POCO
El problema es relativamente simple o puede descomponerse en subproblemas.	Tarea	A9	6	Deseable	Difusa	No	MUCHO
El experto no sigue un proceso determinista en la resolución del problema.	Experto	A10	3	Deseable	Booleana	No	SI
La tarea acepta la técnica de prototipado gradual.	Tarea	A11	8	Deseable	Booleana	No	SI
El experto resuelve el problema a veces con información incompleta o incierta.	Experto	A12	3	Deseable	Difusa	No	TODO
Es conveniente justificar las soluciones adoptadas.	Tarea	A13	3	Deseable	Difusa	No	TODO
La tarea requiere investigación básica.	Tarea	A14	-10	Esencial	Booleana	Sí (No)	NO
El sistema funcionara en tiempo real con otros programas o dispositivos.	Tarea	A15	-6	Deseable	Difusa	No	NADA

TABLA 4.3.4 DIMENSIÓN DE ÉXITO

Denominación de la Característica	Categoría	Dimensión	Peso	Tipo	Naturaleza	Umbral	Valor
Existe una ubicación idónea para el SE.	Directivos / Usuarios	E1	7	Deseable	Difusa	No	TODO
Problemas similares se han resuelto con INCO.	Tarea	E2	8	Deseable	Booleana	No	SI
El problema es similar a otros en lo que resulto imposible aplicar esta tecnología.	Tarea	E3	-5	Deseable	Booleana	No	NO
La continuidad del proyecto está influenciada por vaivenes políticos.	Directivos / Usuarios	E4	-9	Esencial	Difusa	Sí (poco)	POCO
La inserción del sistema se efectúa sin traumas, es decir, apenas se interfiere en la rutina cotidiana.	Directivos / Usuarios	E5	8	Deseable	Difusa	No	MUCHO
Se dispone de experiencia en INCO.	Tarea	E6	7	Deseable	Difusa	No	REGULAR
Se dispone de los recursos humanos, hardware y software necesarios para el desarrollo e implementación del sistema.	Tarea	E7	4	Deseable	Difusa	No	TODO
El experto resuelve el problema en la actualidad.	Experto	E8	4	Deseable	Difusa	No	TODO
La solución del problema es prioridad para la institución.	Directivos / Usuarios	E9	8	Esencial	Difusa	Sí (mucho)	MUCHO
Las soluciones son explicables.	Tarea	E10	5	Deseable	Difusa	No	MUCHO
Los objetivos del sistema son claros y evaluables.	Tarea	E11	6	Deseable	Difusa	No	MUCHO
Los conocimientos están repartidos entre un conjunto de individuos.	Experto	E12	-7	Deseable	Difusa	No	REGULAR
Los directivos, usuarios, experto e IC están de acuerdo en las funcionalidades del SE.	Directivos / Usuarios	E13	4	Esencial	Difusa	Sí (mucho)	MUCHO
La actitud de los expertos ante el desarrollo del sistema es positiva y no se sienten amenazados por el proyecto.	Experto	E14	8	Deseable	Difusa	No	TODO

TABLA 4.3.4 DIMENSION DE ÉXITO (continuación)

Denominación de la Característica	Categoría	Dimensión	Peso	Tipo	Naturaleza	Umbral	Valor
Los expertos convergen en sus soluciones y métodos.	Experto	E15	5	Deseable	Difusa	No	REGULAR
Se acepta la planificación del proyecto propuesta por el IC.	Directivos / Usuarios	E16	8	Esencial	Booleana	Sí (sí)	SI
Existen limitaciones estrictas de tiempo en la realización del sistema.	Tarea	E17	-6	Deseable	Difusa	No	POCO
La dirección y usuarios apoyan los objetivos y directrices del proyecto.	Directivos / Usuarios	E18	7	Esencial	Difusa	Sí (mucho)	MUCHO
El nivel de formación requerido por los usuarios del sistema es elevado.	Directivos / Usuarios	E19	-2	Deseable	Difusa	No	MUCHO
Las relaciones IC - Experto son fluidas.	Experto	E20	4	Deseable	Difusa	No	MUCHO
El proyecto forma parte de un camino critico con otros sistemas.	Tarea	E21	-6	Deseable	Booleana	No	NO
Se efectuará una adecuada transferencia tecnológica.	Directivos / Usuarios	E22	8	Esencial	Difusa	Sí (mucho)	MUCHO
Lo que cuenta en la solución es la calidad de la respuesta.	Experto	E23	5	Deseable	Booleana	No	SI

TABLA 4.3.5 CALCULO DE LA DIMENSION DE PLAUSIBILIDAD

F	Peso	Valor	Valores Difusos				Peso / Valor				Peso * Valor			
			10	10	10	10	1	1	1	1	100	100	100	100
P1	10	Sí	10	10	10	10	1	1	1	1	100	100	100	100
P2	7	Mucho	5,6	6,6	7,8	8,8	1,25	1,06	0,9	0,8	39,2	46,2	54,6	61,6
P3	8	Regular	3,4	4,4	5,6	6,6	2,35	1,82	1,43	1,21	27,2	35,2	44,8	52,8
P4	10	10	10	10	10	10	1	1	1	1	100	100	100	100
P5	9	8	8	8	8	8	1,13	1,13	1,13	1,13	72	72	72	72
	44						6,72	6,00	5,45	5,13	338,4	353,4	371,4	386,4

Rtdo	7,12	7,68	8,26	8,67
------	------	------	------	------

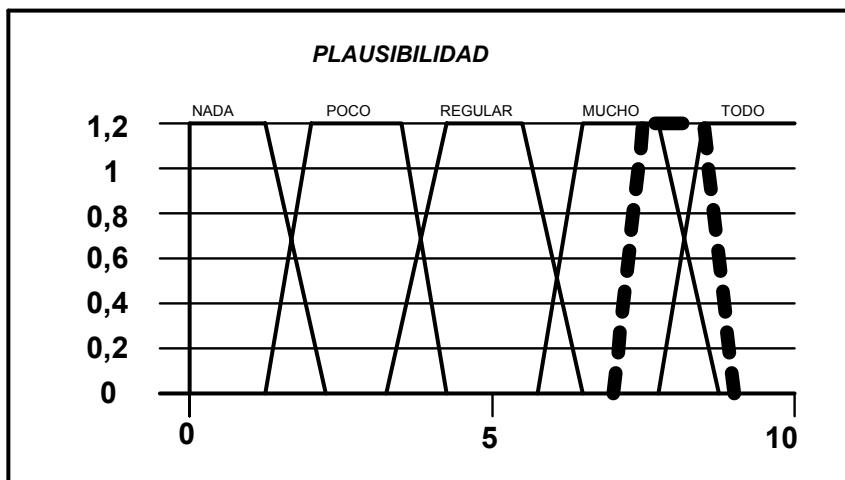


TABLA 4.3.6 CALCULO DE LA DIMENSION DE JUSTIFICACION

F	Peso	Valor	Valores Difusos				Peso * Valor				Aprox. Numérica
J1	8	Mucho	5,6	6,6	7,8	8,8	44,8	52,8	62,4	70,4	57,6
J2	7	7	7	7	7	7	49	49	49	49	49
J3	6	Mucho	5,6	6,6	7,8	8,8	33,6	39,6	46,8	52,8	43,2
J4	10	Regular	3,4	4,4	5,6	6,6	34	44	56	66	50
J5	10	Mucho	5,6	6,6	7,8	8,8	56	66	78	88	72
J6	10	Mucho	5,6	6,6	7,8	8,8	56	66	78	88	72
J7	8	Sí	10	10	10	10	80	80	80	80	80
		59					331,4	375,4	428,2	472,2	

Valor Máximo: 80

Rtdo	7,8	8,8	10	10	TODA
------	-----	-----	----	----	------

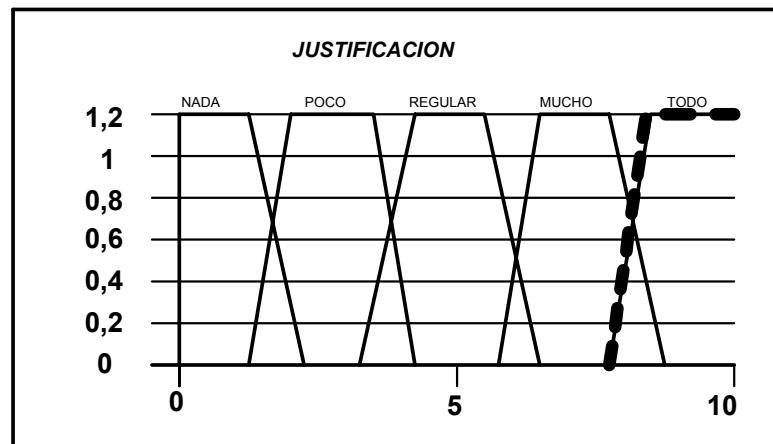


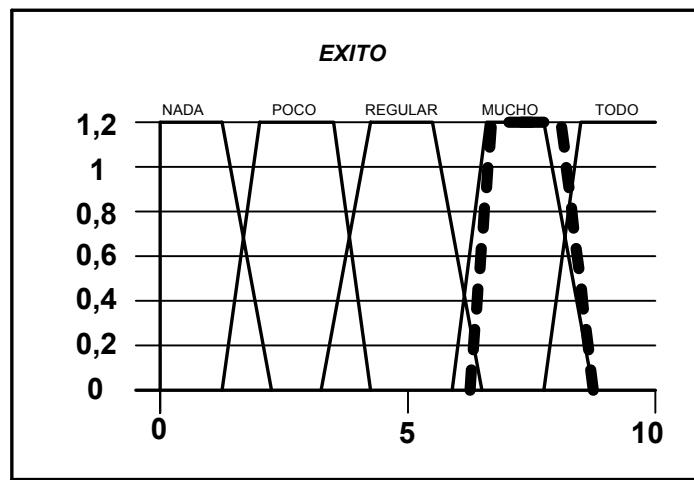
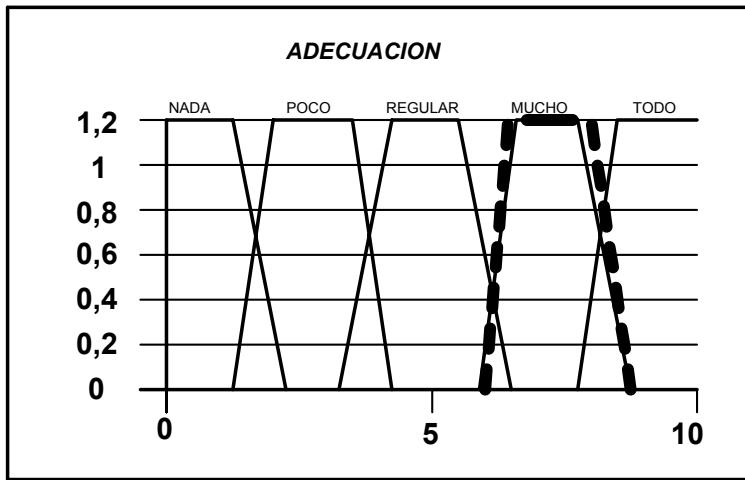
TABLA 4.3.7 CALCULO DE LA DIMENSION DE ADECUACION

F	Peso	Valor	Valores Difusos				Peso / Valor				Peso * Valor			
			5,6	6,6	7,8	8,8	1,25	1,061	0,897	0,795	39,2	46,2	54,6	61,6
A1	7	Mucho	5,6	6,6	7,8	8,8	1,25	1,061	0,897	0,795	39,2	46,2	54,6	61,6
A2	10	Todo	7,8	8,8	10	10	1,282	1,136	1	1	78	88	100	100
A3	-(2)	(Poco)	5,6	6,6	7,8	8,8	0,357	0,303	0,256	0,227	11,2	13,2	15,6	17,6
A4	5	Mucho	5,6	6,6	7,8	8,8	0,893	0,758	0,641	0,568	28	33	39	44
A5	7	Mucho	5,6	6,6	7,8	8,8	1,25	1,061	0,897	0,795	39,2	46,2	54,6	61,6
A6	8	Sí	10	10	10	10	0,8	0,8	0,8	0,8	80	80	80	80
A7	8	Mucho	5,6	6,6	7,8	8,8	1,429	1,212	1,026	0,909	44,8	52,8	62,4	70,4
A8	8	Poco	1,2	2,2	3,4	4,4	6,667	3,636	2,353	1,818	9,6	17,6	27,2	35,2
A9	6	Mucho	5,6	6,6	7,8	8,8	1,071	0,909	0,769	0,682	33,6	39,6	46,8	52,8
A10	3	Sí	10	10	10	10	0,3	0,3	0,3	0,3	30	30	30	30
A11	8	Sí	10	10	10	10	0,8	0,8	0,8	0,8	80	80	80	80
A12	3	Todo	7,8	8,8	10	10	0,385	0,341	0,3	0,3	23,4	26,4	30	30
A13	3	Todo	7,8	8,8	10	10	0,385	0,341	0,3	0,3	23,4	26,4	30	30
A14	-(10)	(No)	10	10	10	10	1	1	1	1	100	100	100	100
A15	-(6)	(Nada)	7,8	8,8	10	10	0,769	0,682	0,6	0,6	46,8	52,8	60	60
	58						18,64	14,34	11,94	10,9	667,2	732,2	810,	853,
													2	2

Rtdo	6,07	7,17	8,25	8,85
-------------	-------------	-------------	-------------	-------------

TABLA 4.3.8 CALCULO DE LA DIMENSION DE EXITO

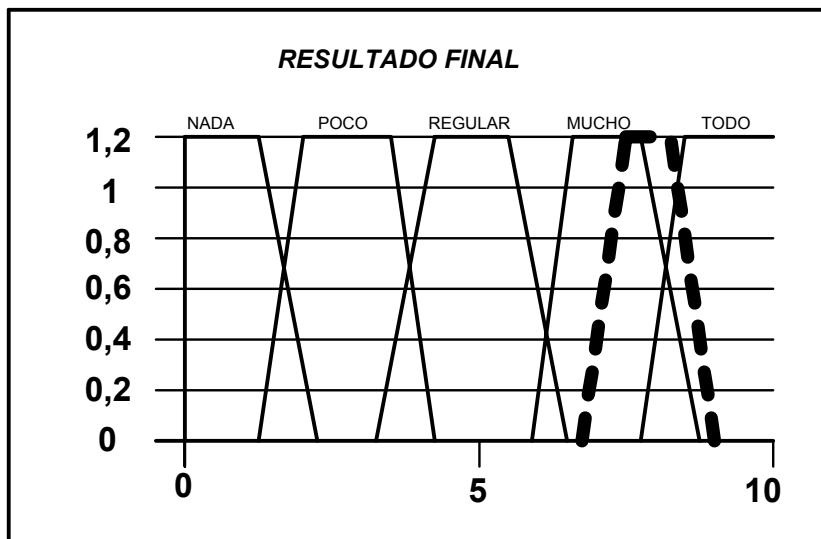
F	Peso	Valor	Valores Difusos				Peso / Valor				Peso * Valor			
E1	7	Todo	7,8	8,8	10	10	0,897	0,795	0,7	0,7	54,6	61,6	70	70
E2	8	Sí	10	10	10	10	0,8	0,8	0,8	0,8	80	80	80	80
E3	-(5)	(No)	10	10	10	10	0,5	0,5	0,5	0,5	50	50	50	50
E4	-(9)	(Poco)	5,6	6,6	7,8	8,8	1,067	1,364	1,154	1,023	50,4	59,4	70,2	79,2
E5	8	Mucho	5,6	6,6	7,8	8,8	1,429	1,212	1,026	0,909	44,8	52,8	62,4	70,4
E6	7	Regular	3,4	4,4	5,6	6,6	2,059	1,591	1,25	1,061	23,8	30,8	39,2	46,2
E7	4	Todo	7,8	8,8	10	10	0,513	0,455	0,4	0,4	31,2	35,2	40	40
E8	4	Todo	7,8	8,8	10	10	0,513	0,455	0,4	0,4	31,2	35,2	40	40
E9	8	Mucho	5,6	6,6	7,8	8,8	1,429	1,212	1,026	0,909	44,8	52,8	62,4	70,4
E10	5	Mucho	5,6	6,6	7,8	8,8	0,893	0,758	0,641	0,568	28	33	39	44
E11	6	Mucho	5,6	6,6	7,8	8,8	1,071	0,909	0,769	0,682	33,6	39,6	46,8	52,8
E12	-(7)	(Regular)	3,4	4,4	5,6	6,6	23,8	30,8	39,2	46,2	23,8	30,8	39,2	46,2
E13	4	Mucho	5,6	6,6	7,8	8,8	0,714	0,606	0,513	0,455	22,4	26,4	31,2	35,2
E14	8	Todo	7,8	8,8	10	10	1,026	0,909	0,8	0,8	62,4	70,4	80	80
E15	5	Regular	3,4	4,4	5,6	6,6	1,471	1,136	0,893	0,758	17	22	28	33
E16	8	Sí	10	10	10	10	0,8	0,8	0,8	0,8	80	80	80	80
E17	-(6)	(Poco)	5,6	6,6	7,8	8,8	1,071	0,909	0,769	0,682	33,6	39,6	46,8	52,8
E18	7	Mucho	5,6	6,6	7,8	8,8	1,25	1,061	0,897	0,795	39,2	46,2	54,6	61,6
E19	-(2)	(Mucho)	1,2	2,2	3,4	4,4	1,667	0,909	0,588	0,455	2,4	4,4	6,8	8,8
E20	4	Mucho	5,6	6,6	7,8	8,8	0,714	0,606	0,513	0,455	22,4	26,4	31,2	35,2
E21	-(6)	(No)	10	10	10	10	0,6	0,6	0,6	0,6	60	60	60	60
E22	8	Mucho	5,6	6,6	7,8	8,8	1,429	1,212	1,026	0,909	44,8	52,8	62,4	70,4
E23	5	Sí	10	10	10	10	0,5	0,5	0,5	0,5	50	50	50	50
	71				25,01	20,09	17,81	16,22	930,4	1039	1170	1256		
		Rtdo.		6,118	7,061	8,107	8,801							

**TABLA 4.3.9 CALCULO DEL VALOR FINAL**

Factor	Coeficiente	Valores Difusos				Peso*Valor			
		7,12	7,68	8,26	8,63	56,96	61,44	66,08	69,04
Plausibilidad	8	7,12	7,68	8,26	8,63	23,4	26,4	30	30
Justificación	3	7,8	8,8	10	10	48,56	57,36	66	70,8
Adecuación	8	6,07	7,17	8,25	8,85	30,6	35,3	40,55	44
Éxito	5	6,12	7,06	8,11	8,8	159,5	180,5	202,6	213,8
	24								

Valor Final	6,65	7,52	8,44	8,91
--------------------	-------------	-------------	-------------	-------------

Media General	7,9
----------------------	------------



4.4 EVALUACION DE VIABILIDAD PARA EL PROYECTO

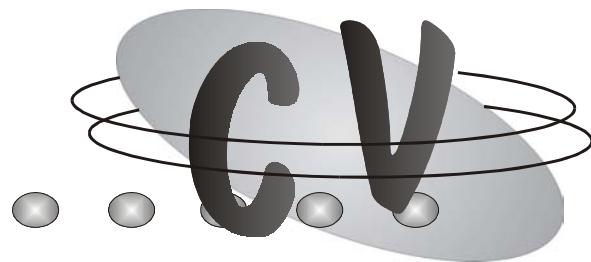
Los resultados obtenidos en cada dimensión son:

- **Plausibilidad:** 7,12; 7,68; 8,26; 8,67.
- **Justificación:** 7,8; 8,8; 10; 10.
- **Adecuación:** 6,07; 7,17; 8,25; 8,85
- **Éxito:** 6,12; 7,06; 8,11; 8,8
- **Conclusión Final:** 6,65; 7,52; 8,44; 8,91
- **Media General:** 7,9

El sistema planteado es viable, la media general es mayor a 6, desde la perspectiva de la Ingeniería del Conocimiento.

Capítulo 5

Adquisición de Conocimientos



En este capítulo se describe el proceso de adquisición de conocimientos y las técnicas usadas en el mismo. Como inicio del proceso se presentan las primeras sesiones de entrevistas con el experto principal para completar el estudio de viabilidad. Luego se documenta el proceso de extracción de conocimientos en el cual se utilizó el Análisis Estructural de Textos. Se detalla el proceso de educación de conocimientos realizado al equipo de expertos para el cual se aplicaron el Método Delphi y la Técnica Nominal de Grupo. Finalmente se presenta la educación de conocimientos con el experto principal, el desarrollo de las entrevistas y la técnica de Emparrillado.

5.1. EL PROCESO DE ADQUISICIÓN DE CONOCIMIENTOS

En las primeras fases del desarrollo de un sistema basado en conocimiento el proceso de adquisición de conocimientos consume el 80% del tiempo disponible. Durante esta etapa la relación con el experto se intensifica y se afianza. Si esta etapa está bien lograda tanto en los aspectos técnicos como así también la empatía con el experto, las posibilidades de éxito se incrementan.

Se entiende por adquisición de conocimientos al proceso de recolección de información, a partir de cualquier fuente necesaria para construir un SBC. La adquisición de conocimientos no es un paso en la metodología de desarrollo de un SBC, sino que es una tarea que se produce en paralelo a todas las etapas de construcción (identificación, conceptualización, formalización, validación, mantenimiento, etc). [Gómez, A. y otros 1997].

Teniendo en cuenta cuál es el tipo de fuente de conocimientos, escrita (incluye cualquier tipo de documentación, video, etc.) o humana, el proceso de adquisición se identifica como Extracción de conocimientos o Educación de Conocimientos respectivamente.

En la metodología IDEAL se propone básicamente el siguiente esquema general a seguir en el proceso de adquisición:

1. Primeras reuniones y evaluación de viabilidad
2. Extracción de conocimientos
3. Educación de conocimientos
 - 3.1. Interrogatorio inicial
 - 3.2. Investigación profunda.

Primeras reuniones y evaluación de viabilidad: estas reuniones son con el experto, los usuarios del SE y los directivos del proyecto. Estas reuniones

permiten conocer las necesidades y expectativas de los usuarios, y realizar el estudio de viabilidad del proyecto. En estas reuniones se busca lograr familiaridad con la terminología del dominio. El objetivo es alcanzar conocimientos generales (grano grueso).

Extracción de conocimientos: este paso es para que el IC se familiarice con el dominio tanto como sea posible antes de comenzar las sesiones con el experto. El objetivo es también información de tipo general, tener una visión general del dominio (grano grueso).

Educción de conocimientos: este es un ciclo en el que se produce la interacción con el experto humano y sus conocimientos. En el *Interrogatorio inicial* se explica y acuerda con el experto cómo se desarrollará el proceso de educación. Se logra una visión de alto nivel del dominio, áreas de aplicación, problemas, conceptos, procedimientos y se bajarán gradualmente a detalles del rea del dominio. En el ciclo de *Investigación profunda* se estudian los pasos para la resolución de la tarea, los datos para resolverla y se conoce el proceso de razonamiento del experto.

5.2 ADQUISICIÓN DE CONOCIMIENTOS DEL PRESENTE TRABAJO

Siguiendo el esquema de adquisición de conocimientos propuesto por la metodología IDEAL el esquema que se aplicó en el presente trabajo fue el siguiente:

1. Primeras reuniones y evaluación de viabilidad
2. Extracción de conocimientos
3. Educción de conocimientos a un equipo de expertos
4. Educción de conocimientos a un único experto

1. Primeras reuniones y evaluación de viabilidad: estas reuniones con el experto (que se encuentran documentadas en el capítulo 12, anexo A, sesiones A.1, A.2 y A.3) han permitido:

- Determinar los objetivos del SE
- Precisar el alcance y ámbito del proyecto
- Obtener información para realizar el estudio de viabilidad del proyecto.
- Realizar y revisar el estudio de viabilidad del proyecto
- Lograr familiaridad con la terminología del dominio
- Determinar la estrategia para la extracción de conocimientos
- Definir el compromiso de participación del grupo de expertos

- Comprometer la participación del experto principal en la preparación y evaluación de las sesiones de educación de conocimientos con el equipo de expertos

2. Extracción de conocimientos: las sesiones de extracción de conocimientos (que se encuentran documentadas en el capítulo 12, anexo B, sesión B.1 del presente capítulo) han permitido:

- Revisar sistemáticamente la bibliografía sugerida por el experto
- Incrementar la familiaridad con la terminología del dominio
- Obtener información para preparar las sesiones de educación de conocimientos con el equipo de expertos

3. Educación de conocimientos a un equipo de expertos: estas reuniones con el grupo de expertos (que se encuentran documentadas en las sesiones C.1, C.2 y C.3 del presente capítulo) han permitido:

- Analizar exhaustivamente los resultados obtenidos de la extracción de conocimientos
- Detallar las características a evaluar de un proyecto para seleccionar el ciclo de vida
- Definir áreas y agrupar las características en esas áreas
- Determinar la necesidad de un primer prototipo del SE con la información obtenida hasta el momento
- Reconocer áreas de ampliación y refinamiento del SE para futuras versiones

4. Educación de conocimientos: este proceso (que se encuentra documentado en las sesiones D.1, y A.4 del presente capítulo) han permitido:

- Analizar la consistencia de la información obtenida en el proceso de educación de conocimientos con el grupo de expertos
- Corroborar el grado de correspondencia entre el modelo mental del experto y la representación obtenida

5.3 TECNICAS PARA ADQUISICIÓN DE CONOCIMIENTOS UTILIZADAS EN EL PRESENTE TRABAJO

Las técnicas usadas en el proceso de adquisición del conocimiento del presente proyecto son:

1. Primeras reuniones y evaluación de viabilidad
 - Entrevistas No Estructuradas
 - Entrevistas Estructuradas
2. Extracción de conocimientos
 - Análisis Estructural de Textos
3. Educción de conocimientos a un equipo de expertos
 - Método Delphi
 - Técnica Nominal del Grupo.
4. Educción de conocimientos al experto principal
 - Emparrillado
 - Entrevistas Estructuradas

Cada una de las sesiones se ha documentado siguiendo el siguiente ciclo de educación [Gómez, A. y otros 1997]:

1. Preparación de la sesión:
 - Información a tratar
 - Amplitud, profundidad
 - Técnica adecuada
 - Preparación de preguntas
2. Sesión
 - Repaso del análisis de la ultima sesión
 - Explicación al experto de los objetivos de la nueva sesión
 - Educación
 - Resumen y comentarios del experto
3. Transcripción de la sesión
4. Análisis de la sesión
 - Lectura para obtención de una visión general
 - Extracción de conocimientos concretos
5. Evaluación de la sesión
 - ¿Se han conseguido los objetivos?
 - ¿Es necesario volver sobre el mismo objetivo?
 - Número y tipo de sesiones necesaria para cubrir el área

A continuación se presenta la documentación de las principales sesiones de adquisición de conocimientos. Se han seleccionado las sesiones realizadas al equipo de expertos utilizando el método Delphi y las sesiones donde se utiliza la técnica de Emparrillado para documentar en el presente capítulo. El resto de las sesiones se encuentran en el capítulo 12.

El orden cronológico en que fueron realizadas las sesiones es el siguiente:

- El grupo de sesiones identificadas con A.x se corresponde con las entrevistas realizadas al experto. Las sesiones A.1, A.2, y A.3 se encuentran en el capítulo 12, anexo A. La sesión A4 que revisa los resultados del Emparrillado se encuentra documentada en el presente capítulo.
- El grupo de sesiones identificadas con B.x se corresponde con las sesiones de extracción de conocimiento, particularmente del análisis de la bibliografía. La sesión B.1 se encuentra en el capítulo 12, anexo B.
- El grupo de sesiones identificadas con C.x se corresponde con las sesiones realizadas al equipo de expertos. Las sesiones C.1, C.2 y C.3 se encuentran documentadas en el presente capítulo.
- El grupo de sesiones identificadas con D.x se corresponde con las sesiones de técnicas indirectas, particularmente Emparrillado. La sesión D.1 se encuentra documentada en el presente capítulo.

5.4 ADQUISICION DE CONOCIMIENTOS A PARTIR DE UN EQUIPO DE EXPERTOS

SESION C.1

C.1.1 Preparación de la Sesión I

- *Información a tratar:* Identificar las características a tener en cuenta en un proyecto en particular para seleccionar un ciclo de vida en Cascada, Orientado a objetos y/o Espiral.
- *Técnica utilizada:* Método Delphi.

- *Preparación del cuestionario.*

El cuestionario se define con una presentación en la que se indican los objetivos y se adjunta al mismo la tabla obtenida como resultado de la extracción de conocimientos (solamente con las columnas de Ventajas y Desventajas) para que se realice la identificación de las características sobre la información resultante del análisis bibliográfico.

En el diseño de la tabla se presentan todas las ventajas y desventajas obtenidas del análisis de textos, pero no se identifican a qué ciclos de vida están asociadas, para que los expertos recurran a su propia experiencia en la selección de las características asociadas con los ciclos de vida a seleccionar.

Se presenta una muestra del cuestionario que se entregó en el ítem C.1.5.

C.1.2 Realización de la Sesión I

- *Sesión realizada el:* 16 de febrero de 2000.
- *Expertos asistentes:* Prof. Perichinsky, Lic. Beltrami, Lic. Leone, Ing. Pollo Cattaneo, Lic. Lucchini, Ing. Weschler, Lic. Fernández.
- *Ingeniero del conocimiento:* Bibiana Rossi.
- *Lugar:* Oficina del experto. Salón de conferencias con mesas de trabajo.
- *Tiempo:* 18 a 21.00hs.
- *Objetivos:* Identificar las características a tener en cuenta en un proyecto en particular para seleccionar un ciclo de vida en Cascada, Orientado a objetos y/o Espiral.
- *Procedimiento:*
 - El IC explica básicamente el proyecto. El experto principal ha tenido reuniones previas de preparación con el equipo de expertos.
 - El IC presenta el método Delphi.
 - El IC presenta el cuestionario y el documento adjunto. Explica el procedimiento que se siguió para la elaboración de la tabla.
 - El IC explica el objetivo de la sesión. Responde preguntas aclaratorias.
 - Los expertos responden el cuestionario individualmente sin interactuar entre ellos.

C.1.3 Tabulación y Análisis de los Resultados de la Sesión I

- *Conocimientos extraídos y tabulados:*

Características a tener en cuenta en un proyecto en particular para seleccionar un ciclo de vida en Cascada, Orientado a objetos y/o Espiral.

El documento tabulado resultante de la síntesis de las respuestas individuales de cada experto, se presenta en el ítem C.1.6.

- *Conocimientos a educir en próximas sesiones:*

Revisar los resultados de la primera sesión del método Delphj.

Identificar si existe una forma de agrupar las características por área que facilite el análisis del proyecto para el cual se desea seleccionar el ciclo de vida.

C.1.4 Evaluación de la Sesión I

- *¿Se han logrado los objetivos?*

Si se han cumplido. Cada experto entregó un documento donde agrupo las características más relevantes a tener en cuenta de un proyecto en particular para seleccionar los ciclos de vida Cascada, Orientado a objetos y Espiral.

- *¿Es necesario volver sobre lo mismo?*

Es necesario refinar la información obtenida y agruparla en áreas.

- *Número y tipo de sesiones para cubrir el área*

Se estima:

- 3 sesiones con el experto principal para la tabulación de análisis y resultados.
- 2 sesiones con el grupo de expertos.
- 3 sesiones con el experto principal para completar la fase de inicial de adquisición de los conocimientos.

C.1.5 Cuestionario de la sesión C.1

Acerca del Proyecto

- **Proyecto:** Identificar las características a tener en cuenta en un proyecto en particular para seleccionar un ciclo de vida en Cascada, Orientado a objetos y/o Espiral.
- **Etapa del Proyecto:** Adquisición de Conocimientos.
- **Técnicas Utilizadas:**

Técnica usada	Resultado
Entrevistas	Estudio de viabilidad
Análisis de textos	Tabla de Ventajas y Desventajas de los modelos de ciclos de vida
Método Delphi	Características a analizar de un proyecto para seleccionar un modelo de ciclo de vida entre Cascada, Orientado a objetos y/o Espiral

- **Método Delphi - Pasos a seguir:**

1. Elaboración y refinamiento del cuestionario
2. Sesión I - Respuestas al cuestionario
3. Tabulación y análisis de los resultados 1er.ciclo
4. Sesión II - Respuestas y refinamiento
5. Tabulación y análisis de los resultados 2do. Ciclo
6. Sesión III - Respuestas y refinamiento
7. Tabulación y análisis de los resultados finales

Acerca de la Sesión I

- **Propósito:** Tomando como punto de partida la Tabla de ventajas y desventajas de los modelos de ciclos de vida, identificar las características a analizar de un proyecto para seleccionar el ciclo de vida en Cascada, Orientado a objetos y/o Espiral.
- **Material que se adjunta:**
 - Información acerca del proyecto
 - Información acerca de la Sesión I
 - Lista a completar para Modelo en Cascada
 - Lista a completar para Modelo Orientado a objetos
 - Lista a completar para Modelo en Espiral
 - Tabla de ventajas y desventajas de los modelos de ciclos de vida
- **Procedimiento:**
 1. Leer la Tabla de ventajas y desventajas
 2. Identificar todas las características relevantes de un proyecto para seleccionar el modelo de ciclo de vida en Cascada (considerando ítem 5). Escribirlas en la lista correspondiente.
 3. Identificar todas las características relevantes de un proyecto para seleccionar el modelo de ciclo de vida en Orientado a objetos (considerando ítem 5). Escribirlas en la lista correspondiente.
 4. Identificar todas las características relevantes de un proyecto para seleccionar el modelo de ciclo de vida en Espiral (considerando ítem 5). Escribirlas en la lista correspondiente.
 5. Analizar en cada caso:
 - ✓ ¿Es posible modificar algo?
 - ✓ ¿Es posible añadir?
 - ✓ ¿Es posible eliminar?
 - ✓ ¿Es posible sustituir un concepto por otro?
 - ✓ ¿Es posible combinar elementos entre sí?
 - ✓ ¿Se puede tomar la idea en sentido opuesto?
 - ✓ ¿Está asociada a uno a dos o a los tres modelos de CV a seleccionar?

Sesión I: 16 de febrero de 2000

Experto:

Lista a completar para Modelo en CASCADA

Sesión I: 16 de febrero de 2000

Experto:

Lista a completar para Modelo en ORIENTADO A OBJETOS

Sesión I: 16 de febrero de 2000

Experto:

Lista a completar para Modelo en ESPIRAL

TABLA DE VENTAJAS /DESVENTAJAS DE LOS MODELOS DE CICLOS DE VIDA

Modelo	Ventajas	Desventajas
Modelo 1	<ul style="list-style-type: none"> ➤ No está limitado a proyectos chicos también puede manejar proyectos grandes (1) ➤ Sirve para la formular los requerimientos de un sistema de Software cuando los requerimientos del usuario son vagos, incompletos o inestables (1) ➤ Puede servir como herramienta para experimentar con nuevas e innovadoras ideas de diseño (1) ➤ Puede servir como factor de seguridad en desarrollos con alto factor de riesgo (1) ➤ Puede servir como forma de reaccionar ante potenciales cambios organizacionales (1) ➤ Puede servir como forma de promover al cliente a participar del proceso de desarrollo (1) ➤ Facilita un ambiente de enseñanza para usuarios finales potenciales durante el desarrollo (1) ➤ Puede facilitar la introducción gradual de un sistema de computación en una organización (1) ➤ Es usado cuando hay un gran nivel de incertidumbre (1) ➤ Es usado cuando hay varias opciones de diseño e implementación (1) ➤ Es usado cuando hay dificultades en formular las especificaciones (1) ➤ Es usado cuando no hay experiencia previa en el desarrollo con una técnica específica (1) ➤ Cuando se necesita un método para producir el sistema en forma gradual (1) 	

Modelo	Ventajas	Desventajas
Modelo 2	<ul style="list-style-type: none"> ➤ Sirve para formular los requerimientos de un sistema de Software cuando los requerimientos del usuario son vagos, incompletos o inestables (5) 	<ul style="list-style-type: none"> ➤ Se puede caer en el error de prolongar en el desarrollo final, decisiones ineficientes utilizadas en el prototipo (5)
Modelo 3	<ul style="list-style-type: none"> ➤ Los incrementos se pueden planear para gestionar riesgos técnicos ➤ Se ajusta a entornos de alta incertidumbre, por no tener la necesidad de poseer un conjunto exhaustivo de requisitos, (especificaciones, diseños, etc.), al comenzar el sistema, ya que cada refinamiento amplía los requisitos y las especificaciones derivadas de la fase anterior. ➤ Es particularmente útil cuando la dotación de personal no está disponible para una implementación completa en cuanto a la fecha límite de gestión que se ha establecido (5) 	<ul style="list-style-type: none"> ➤ Falta de planificación a largo plazo ➤ Tentación de caer en el modelo de codificar y corregir ➤ Existe el problema de determinar si los requisitos propuestos son válidos, se detectan tarde. ➤ Puede encomendar demasiados recursos a una solución errónea ➤ Es difícil para más de una persona trabajar en un único prototipo ➤ Aunque permite el cambio continuo de requisitos existe el problema de determinar si los requisitos propuestos son válidos. ➤ Los errores en los requisitos se detectan tarde y su corrección resulta tan costosa como en el modelo en cascada.
Modelo 4		<ul style="list-style-type: none"> ➤ Descuida el análisis apropiado ➤ Recae en la intuición de los desarrolladores ➤ Lleva a decisiones de diseño prematuras ➤ La adquisición de conocimientos y el análisis se vuelven manejados por la implementación ➤ Se hace difícil de mantener ya que favorece la falta de documentación del análisis y diseño.

Modelo	Ventajas	Desventajas
Modelo 5	<ul style="list-style-type: none"> ➤ Ayuda a prevenir que se sobrepasen las fechas de entrega y los costos esperados. ➤ Al final de cada fase el personal técnico y los usuarios tienen la oportunidad de revisar el progreso del proyecto. ➤ Da facilidades a los gestores para controlar el progreso de los sistemas. ➤ Reconocimiento de ciclos de realimentación entre etapas (2) ➤ Es mejor que un enfoque hecho al azar (5) 	<ul style="list-style-type: none"> ➤ Enfoque secuencial para construir un producto en el cual la iteración no es evidente. ➤ Se tarda mucho tiempo en pasar por todo el ciclo, dado que hasta que no se finalice una fase no se pasa a la siguiente. ➤ El enfoque top-down necesita ser matizado con un paso de reciclado (vuelta para atrás) para cubrir temas como riesgo y reuso de módulos de software (2) ➤ No maneja adecuadamente aspectos concernientes al desarrollo de familias de programas y de organización de software para permitir cambios (2) ➤ Asume progresión relativamente uniforme en los pasos de elaboración (2) ➤ El énfasis en elaborados documentos como criterio de finalización de las fases de requerimientos y diseño no funciona bien para muchas clases de software, particularmente las aplicaciones interactivas. ➤ No contempla la clase de desarrollo evolutivo que presenta el prototipado rápido y los lenguajes de 4ta. generación. (2) ➤ Los proyectos reales rara vez siguen el modelo secuencial que propone el modelo (5). ➤ No contempla los posibles modos de desarrollo de software futuros, asociados con las capacidades de la programación automática, transformación de programas y asistentes de software basados en el conocimiento. (2) ➤ A menudo es difícil que el cliente explice todos los requisitos (5) ➤ Asume una especificación de requisitos perfecta. ➤ Detección de errores tardía

Modelo	Ventajas	Desventajas
		<ul style="list-style-type: none"> ➤ No contempla los posibles modos de desarrollo de software futuros, asociados con las capacidades de la programación automática, transformación de programas y asistentes de software basados en el conocimiento. (2) ➤ Una versión de trabajo del programa no estará disponible hasta que el proyecto esté muy avanzado (5) ➤ No refleja el proceso real de desarrollo de software. Los proyectos reales raramente siguen este flujo secuencial, puesto que siempre hay iteraciones. Aunque en este modelo la iteración esté permitida en etapas contiguas (MACRO,1990),en la vida real la iteración abarca mas de una etapa. Un caso típico es la redefinición de los requisitos cuando se está codificando la aplicación. ➤ Acentúa el fracaso de la industria del software con el usuario final. En este caso, el usuario debe tener paciencia, ya que el sistema en funcionamiento no estará disponible hasta las fases finales del proyecto. ➤ Agravado para KBS (incompleto y con especificaciones cambiantes) ➤ Administradores y usuarios tienen poca idea de como quedará el sistema cuando se especifican los requerimientos

Modelo	Ventajas	Desventajas
Modelo 6	<ul style="list-style-type: none"> ➤ En el desarrollo interno existe gran flexibilidad y libertad para ajustarse a los acuerdos etapa por etapa, para aplazar acuerdos de opciones específicas, para establecer miniespirales para resolver caminos críticos, para ajustar niveles de esfuerzo, o para acomodar prácticas como prototipado desarrollo evolutivo y uso de métodos de diseño ajustado al costo. ➤ Alienta el desarrollo de especificaciones que no son necesariamente uniformes, exhaustivas o formales, al diferir la elaboración detallada de los elementos de software de bajo riesgo, y evita roturas innecesarias en sus diseños, hasta que los elementos de alto riesgo del diseño sean establecidos. ➤ La revisión de los principales objetivos sirve para asegurar que todas las partes involucradas están de acuerdo respecto al método de trabajo para la siguiente fase. ➤ La identificación de riesgos asociados con cada una de las alternativas y las diferentes maneras de resolverlos son el centro del modelo. Con los métodos tradicionales, es habitual dejar las partes, difíciles para el final y empezar con las más fáciles y de menor riesgo, obteniendo así la ilusión de un gran avance. ➤ La división de los proyectos en ciclos, cada uno con un acuerdo al final de cada ciclo, implica que existe un acuerdo para los cambios a realizar o para terminar el proyecto, en función de lo que se ha aprendido desde el inicio del proyecto. 	<ul style="list-style-type: none"> ➤ Trabaja bien en los desarrollos internos, pero necesita un ajuste posterior para adaptarlo a la subcontratación de software. ➤ Necesidad de expertos en evaluación de riesgos para identificar y manejar las fuentes de riesgos de un proyecto. ➤ Requiere una considerable habilidad para la evaluación del riesgo y depende de ella para el éxito (5) ➤ Deposita una gran cantidad de confianza en la habilidad de los desarrolladores de software para identificar y manejar las fuentes de riesgo del proyecto. (5) ➤ En general, los pasos del proceso necesitan una elaboración adicional para asegurar que todos los participantes de un desarrollo de software estén operando en un contexto consistente. (5) ➤ Puede demostrar dificultad para ejecutar especificaciones con una adecuada performance ➤ En el desarrollo de software bajo contrato no existe esta flexibilidad y libertad, por lo que es necesario mucho tiempo para definir los contratos, ya que los entregables no están previamente definidos de forma clara. ➤ Todavía no encaja en el mundo de la adquisición de software por contrato. ➤ A no ser que se realice una inspección por expertos, en este tipo de proyecto se tendrá la ilusión de progresar por un período, y sin embargo, se encuentra dirigido directamente hacia el desastre. ➤ Las personas pueden encontrarlo difícil para dirigir transformaciones

Modelo	Ventajas	Desventajas
	<ul style="list-style-type: none"> ➤ Contesta la pregunta “cuánto es suficiente?” para cada una de las fuentes de la actividad del proyecto y para el manejo de los recursos. ➤ Puede aplicarse a lo largo de la vida del proyecto (5) ➤ Existe un reconocimiento explícito de las diferentes alternativas para alcanzar los objetivos de un proyecto ➤ El modelo se adapta a cualquier tipo de actividad, incluidas algunas que no existen en otros métodos (por ejemplo, consulta de asesores expertos o investigadores ajenos) que son muy útiles para la consecución de los objetivos de un proyecto. ➤ Acomoda una mezcla apropiada de enfoques orientados a especificaciones, orientados a prototipos, orientados a simulaciones, orientados a transformación automática, y otros (2) ➤ Es aplicable tanto a esfuerzos de desarrollo como de mejora (enhacement). (2) ➤ Promueve el desarrollo de especificaciones que no son necesariamente uniformes, exhaustivas o formales en las que posterga la elaboración detallada de elementos de software de bajo riesgo, evitando un corte innecesario en su diseño (2) ➤ Incorpora prototipado como una opción de reducción de riesgo en cualquier etapa de desarrollo (2) ➤ Mejora la estimación y reduce el costo de corregir ➤ Permite vueltas atrás en etapas tempranas del espiral al identificar alternativas más atractivas o necesitar la resolución de un nuevo riesgo. 	

Modelo	Ventajas	Desventajas
	<ul style="list-style-type: none"> ➤ Permite explícitamente estrategias para desarrollar familias de programas y para reusar software existente (2) ➤ Permite la evolución del ciclo de vida, crecimiento y cambios en el producto de software. (2) ➤ Provee mecanismos para incorporar objetivos de calidad del software en el proceso de desarrollo del producto de software. (2) ➤ Se focaliza en eliminar errores y alternativas poco atractivas en forma temprana (2) ➤ Permite iteraciones, vueltas atrás y terminación prematura de proyectos no viables (2) ➤ Puede soportar y ser soportado por ambientes de desarrollo de software avanzados (2) ➤ No implica procedimientos separados para desarrollo y mejoramiento de software. (2) ➤ Provee un marco viable para el desarrollo de sistemas de soft o hardware integrados. (2) 	
Modelo 7	<ul style="list-style-type: none"> ➤ La tecnología OO pretende acelerar el desarrollo de sistemas de una manera iterativa e incremental. ➤ La ventaja principal de estos modelos es que permiten fijar hitos más frecuentemente, realizando entregas de sistemas que son operativos cada dos o tres meses para recibir retroalimentación del cliente lo antes posible e ir adaptando la aplicación según cambien las necesidades y se refinen los requisitos 	<ul style="list-style-type: none"> ➤ Otro aspecto importante en la tecnología OO es "generalizar" los componentes para que sean reutilizables, lo que incrementa los costos de desarrollo entre un 10 y 50%, por lo que resulta imprescindible un desarrollo que optimice esta inversión. ➤ El inconveniente que presentan es la dificultad de gestionar de manera formal los proyectos que siguen estos ciclos de vida aunque, como se ha señalado, este problema se puede paliar diferenciando el "micro" del "macroproceso".
Modelo 8	<ul style="list-style-type: none"> ➤ Produce un diseño limpio y fácil de comprender, que resulta más fácil de probar, mantener y extender. 	<ul style="list-style-type: none"> ➤ La distinción entre análisis y diseño podrá parecer a veces arbitraria y confusa

Modelo	Ventajas	Desventajas
	<ul style="list-style-type: none"> ➤ Da al proceso de desarrollo una base más estable, y permite utilizar un único concepto unificador de software a lo largo del proceso: el concepto de objeto, de tal forma que la información registrada durante el análisis no se pierde ni se transforma cuando se produce el diseño y la implementación. ➤ La organización de un sistema en torno de objetos da al proceso de desarrollo una estabilidad de la cual carecen las aproximaciones orientadas a funciones. ➤ No hay discontinuidad en los modelos, una notación de una fase sea sustituida por otra notación distinta en otra fase ➤ Los cambios de funcionalidad se admiten con facilidad en el diseño orientado a objetos ➤ El paradigma del mundo real formado por objetos y relaciones proporciona el contexto para comprender el comportamiento dinámico y funcional ➤ La misma notación hace más sencillo repetir los pasos de desarrollo con grados de detalle cada vez más finos. Cada iteración añade o clarifica características, en lugar de modificar un trabajo que ya se había hecho. Hay menos oportunidades para introducir incongruencias y errores ➤ Es más flexible al cambio y más extensible ➤ Los sistemas son más fáciles de entender, esto hace que el diseño sea más intuitivo y simplifica la seguibilidad entre los requisitos y el código de software. El diseño resulta sea más coherente para personas que no fueran parte del equipo original del diseño 	

Modelo	Ventajas	Desventajas
	<ul style="list-style-type: none"> ➤ Se incrementa la reutilización de los componentes de un proyecto para el siguiente ➤ Integra mejor las bases de datos con el código ➤ La descomposición de software orientada a objetos modela más de cerca la percepción que de la realidad tiene la persona. Por tanto, no es sorprendente que el software desarrollado resulte más comprensible, extensible y mantenable. ➤ El diseño resultante es más limpio y más adaptable, los cambios futuros serán mucho más sencillos 	
Modelo 9	<ul style="list-style-type: none"> ➤ La tecnología de objetos pretende acelerar el desarrollo de sistemas de una manera iterativa e incremental. ➤ Permite fijar hitos más frecuentemente, realizando entregas de sistemas que son operativos cada dos o tres meses, para recibir retroalimentación del cliente lo antes posible e ir adaptando la aplicación según cambian las necesidades y se refinan los requisitos. ➤ El enfoque evolutivo que el macroproceso adopta para el desarrollo significa que hay oportunidades para identificar problemas en momentos tempranos del ciclo de vida y responder convenientemente a estos riesgos antes de que comprometan el éxito del proyecto (7) ➤ La gestión de proyectos OO, en estado estables, provoca una reducción en la cantidad total de recursos que se necesitan y un desplazamiento en el ritmo de su despliegue respecto a métodos más tradicionales (7) 	<ul style="list-style-type: none"> ➤ La tecnología de objetos pretende <i>generalizar</i> los componentes para que sean reutilizables, lo que incrementa los costos de desarrollo entre un 10 y un 50%, por lo que resulta imprescindible un desarrollo que optimice esta inversión. ➤ Dificultad de gestionar de manera formal los proyectos que siguen estos ciclos de vida aunque, como se ha señalado, este problema se puede paliar diferenciando el <i>micro</i> del <i>macroproceso</i> ➤ En algunos casos tiene un alto costo de puesta en marcha (7) ➤ Tiene problemas de eficacia en ciertos casos particulares (7) ➤ El microproceso del desarrollo orientado a objetos es inestable de forma innata y requiere una dirección activa para forzar su conclusión (7)

Modelo	Ventajas	Desventajas
	<ul style="list-style-type: none"> ➤ Alienta la reutilización de componentes de software (7) ➤ Los costos del ciclo de vida son con frecuencia menores que los de un enfoque tradicional porque el producto resultante tiende a ser de mucha mejor calidad y mas flexible ante el cambio (7) ➤ Lleva a sistemas más flexibles al cambio (7) ➤ Reduce el riesgo de desarrollo ➤ La integración incremental tiende a reducir el riesgo de desarrollo porque acelera el descubrimiento de problemas de arquitectura y eficacia en etapas tempranas del desarrollo ➤ El macroproceso funciona muy bien para asegurar la calidad porque permite una recolección de datos continua sobre la tasa de descubrimiento de errores. La integración sucede incrementalmente a lo largo de todo el ciclo de vida en vez de ocurrir en un evento explosivo ➤ Un desarrollo OO explota la potencia expresiva de los lenguajes de programación orientados a objetos (7) ➤ El enfoque de desarrollo incremental es extremadamente apropiado para el paradigma orientado a objetos (7) 	
Modelo 10	<ul style="list-style-type: none"> ➤ Maneja adecuadamente aspectos concernientes al desarrollo de familias de programas y a la organización de software para acomodar cambios (2) 	<ul style="list-style-type: none"> ➤ No ha sido completamente elaborado para ver como cubre aspectos tales como prototipado y reusabilidad (2)
Modelo 11	<ul style="list-style-type: none"> ➤ Contiene procesos de abstracción separados hasta que se obtiene una especificación formal (2) 	<ul style="list-style-type: none"> ➤ Tiene retos en acomodar reusabilidad de software, familias de programas .
Modelo 12		<ul style="list-style-type: none"> ➤ Tienen unos límites del sistema bien definidos, así que puede resultar difícil extender un diseño SA/SD hasta unos nuevos límites

Modelo	Ventajas	Desventajas
Modelo 13	<ul style="list-style-type: none"> ➤ Facilita el logro de objetivos de desarrollo de soft para muchos proyectos (3) ➤ Es práctico en el uso del enfoque top-down step-wise. 	<ul style="list-style-type: none"> ➤ No es fácil de aplicar cuando el proyecto es de gran tamaño(3) ➤ Requiere que el problema y su solución estén bien comprendidos
Modelo 14	<ul style="list-style-type: none"> ➤ Incorpora al ciclo de vida un marco conceptual para incorporar capacidades de programación automática, transformación de programas y asistentes basados en conocimiento (2) 	<ul style="list-style-type: none"> ➤ Tiene problemas en escalar a sistemas muy grandes (2) ➤ Tiene problemas en acomodar familias de programas (2) ➤ Tiene problemas en manejar opciones entre nuevas y antiguas capacidades (2)
Modelo 15	<ul style="list-style-type: none"> ➤ Los requerimientos en prosa pueden ser leídos y aprobados directamente por los clientes ➤ El prototipado es posible por iteración del ciclo de desarrollo entero ➤ Provee de puntos útiles de chequeo (milestones) (4) 	<ul style="list-style-type: none"> ➤ Los requerimientos informales son notorios por su ambigüedad ➤ El testeo y la prueba de correctitud son muy difíciles ➤ Es difícil que soporte automatización completa (4) ➤ La descomposición top-down es difícil y riesgosa (4)
Modelo 16	<ul style="list-style-type: none"> ➤ Los requerimientos formales pueden ser procesados por la maquina La especificación ejecutable provee resultados tempranos. (4) ➤ Dispone prototipado rápido automáticamente (4) ➤ El testing o verificación es evitado derivando la implementación de la especificación usando solo transformaciones y mapeos que han sido probados (4) ➤ Maneja objetos formales, lo que implica que se pueden desarrollar templete y herramientas (4) ➤ Los usuarios tienen un modelo del sistema para interiorizarse y evaluarlo 	<ul style="list-style-type: none"> ➤ Los requerimientos formales son inaccesibles para los usuarios finales y otras personas no técnicas (4) ➤ Presenta peligro de decisiones de diseño prematuras (4) ➤ Puede resultar difícil ejecutar especificaciones con performance adecuada ➤ Es una tecnología nueva y subdesarrollada (4) ➤ La implementación transformacional es nueva y no desarrollada tecnológicamente ➤ Las personas pueden encontrarlo difícil para dirigir transformaciones

C.1.6 Resultados de la sesión C.1

Resultados Tabulados de la Sesión I

Modelo en CASCADA

Modelo ORIENTADO A OBJETOS

ITEM	CARACTERISTICA DEL PROYECTO
17. O:	<i>No se han definido claramente los límites del sistema</i>
18. O:	<i>No se han definido claramente los requisitos del sistema</i>
19. O:	<i>Se puede asumir una progresión NO uniforme y secuencial en las fases del ciclo de vida</i>
20. O:	<i>Se presume la necesidad de retroalimentación en las fases del desarrollo del proyecto</i> <ul style="list-style-type: none"> • <i>Se presume independencia entre las fases del desarrollo del proyecto</i> • <i>Se presume poca dependencia entre las fases de desarrollo del proyecto</i> • <i>Se presume mucha dependencia entre las fases de desarrollo del proyecto</i>
21. O:	<i>El proyecto requiere el descubrimiento de problemas de arquitectura en etapas tempranas del desarrollo</i>
22. O:	<i>El proyecto requiere el descubrimiento de problemas de eficacia en etapas tempranas del desarrollo</i>
23. O:	<i>El sistema a diseñar es innovador</i>
24. O:	<i>El sistema se basa en componentes de fuerte contenido matemático</i>
25. O:	<i>El sistema se basa en componentes de fuerte contenido gráfico</i>
26. O:	<i>El sistema tiene subsistemas que presentan comportamiento dinámico</i>
27. O:	<i>El tipo de aplicación presenta comportamiento fuertemente interactivo</i>
28. O:	<i>El tipo de aplicación presenta comportamiento en Tiempo Real</i>
29. O:	<i>Se prevé una alta variabilidad (modificaciones en las funciones) en los procesos del sistema</i>
30. O:	<i>El sistema es explícitamente sensible a cambios, extensiones y/o ampliaciones.</i>
31. O:	<i>El tipo de aplicación es desarrollo de software de base</i>
32. O:	<i>El tipo de aplicación es un SSBCC sistema basado en conocimientos</i>
33. O:	<i>El sistema presenta subsistemas complejos</i> <ul style="list-style-type: none"> • <i>Es necesario descomponer el sistema en varios niveles para su mejor comprensión</i> • <i>Es necesario descomponer el sistema en varios niveles para su mejor manipulación</i>
34. O:	<i>El proyecto requiere una gestión de proyecto medianamente formal.</i>
35. O:	<i>El proyecto requiere una gestión de proyecto poco formal.</i>
36. O:	<i>Se prevé una fuerte necesidad de reutilizar los componentes de un proyecto para el siguiente</i>
37. O:	<i>Es un subsistema de un sistema desarrollado en Objetos</i>
38. O:	<i>Es una ampliación de un sistema desarrollado en Objetos</i>
39. O:	<i>Es una modificación de un sistema desarrollado en Objetos</i>
40. O:	<i>Se prevé una fuerte necesidad de reutilizar los componentes de un software ya existente (desarrollado en Objetos)</i>
41. O:	<i>Es necesario disponer de una versión temprana del software desarrollado</i>
42. O:	<i>Se prevé la necesidad de usar metodologías medianamente probadas</i>
43. O:	<i>Se presume riesgo en el desarrollo del sistema.</i>

Modelo en ESPIRAL

ITEM	CARACTERISTICA DEL PROYECTO
44. E: El entorno del sistema es de alta incertidumbre	
45. E: Hay gran nivel de incertidumbre en la especificación de los requisitos	<ul style="list-style-type: none"> • El cliente no ha explicitado la mayoría de los requisitos • Las especificaciones no son uniformes, exhaustivas o formales • Los límites del sistema no se definen claramente.
46. E: Se puede asumir una progresión NO uniforme y secuencial en las fases del ciclo de vida	
47. E: Se presume la necesidad de retroalimentación en las fases del desarrollo del proyecto	<ul style="list-style-type: none"> • Se presume independencia entre las fases del desarrollo del proyecto • Se presume poca dependencia entre las fases de desarrollo del proyecto • Se presume mucha dependencia entre las fases de desarrollo del proyecto
48. E: El tipo de aplicación a diseñar es innovador	
49. E: El sistema a desarrollar puede resolverse por varias opciones de diseño	
50. E: El sistema a desarrollar puede resolverse por varias opciones de implementación	
51. E: El proyecto requiere el descubrimiento de problemas de arquitectura del sistema en etapas tempranas del desarrollo	
52. E: El proyecto requiere el descubrimiento de problemas de eficacia (en la arquitectura del sistema) en etapas tempranas del desarrollo	
53. E: El tipo de aplicación requiere el desarrollo fuertemente integrado de hardware y software	
54. E: El tipo de sistema a desarrollar es un SSBBCC sistema basado en conocimientos	
55. E: El tipo de sistema a desarrollar es algorítmico	
56. E: Se prevé en el desarrollo del sistema crecimientos y cambios del producto	<ul style="list-style-type: none"> • El tipo de sistema a desarrollar es software de base
57. E: El sistema presenta subsistemas complejos	<ul style="list-style-type: none"> • Es necesario descomponer el sistema en varios niveles para su mejor comprensión • Es necesario descomponer el sistema en varios niveles para su mejor manipulación
58. E: El tipo de aplicación presenta características fuertemente interactivas	
59. E: El tipo de aplicación presenta características de Tiempo Real	
60. E: El tipo de aplicación a desarrollar es un sistema nuevo totalmente	
61. E: Es necesario facilitar la introducción gradual del sistema para los usuarios	
62. E: Es necesario contar con una fuerte participación del cliente en el Desarrollo	
63. E: Se cuenta con habilidad para la evaluación del riesgo	<ul style="list-style-type: none"> • Se cuenta con métodos y técnicas para evaluar el riesgo
64. E: La gestión del proyecto requiere de acuerdos confirmados para los cambios a efectuarse durante el desarrollo del proyecto	
65. E: La gestión del proyecto requiere entregas parciales del sistema con funciones que se irán completando gradualmente.	
66. E: La gestión del proyecto requiere entregas tempranas del sistema con funciones que se irán completando gradualmente.	

<i>67. E: La gestión del proyecto prevé definir explícitamente estrategias para reusar software existente.</i>
<i>68. E : No hay experiencia previa en el sistema a desarrollar</i>
<i>69. E: Es necesario disponer de una versión temprana del software desarrollado</i>
<i>70. E: El desarrollo del sistema es responsabilidad de la organización, no se terceriza</i>
<i>71. E: Se dispone de software para prototipar</i>
<i>72. E: Es factible adquirir software para prototipar</i>
<i>73. E: Existen importantes dudas sobre la viabilidad del software</i>
<i>74. E: Se presume alto factor de riesgo en el desarrollo del sistema</i> <ul style="list-style-type: none"> • <i>Es necesario definir explícitamente estrategias para realizar análisis de riesgo</i> <ul style="list-style-type: none"> ✓ <i>Permite vuelta atrás a etapas anteriores cuando se requiere la resolución de algún tópico nuevo de riesgo</i> ✓ <i>Permite vuelta atrás a etapas anteriores de la espiral cuando son identificadas mejores alternativas</i> ✓ <i>Acomoda iteraciones, vuelta atrás y terminación prematura de proyectos no viables</i> ✓ <i>Identificar los riesgos asociados con cada una de las alternativas y las diferentes maneras de resolverlos.</i> ✓ <i>Es necesario definir mecanismos explícitos para incorporar objetivos de calidad en el desarrollo del producto (análisis de riesgo)</i>
<i>75. E: Se estiman riesgos técnicos en el desarrollo del sistema</i> <ul style="list-style-type: none"> • <i>Es necesario definir explícitamente estrategias para realizar análisis de riesgo</i> <ul style="list-style-type: none"> ✓ <i>Permite vuelta atrás a etapas anteriores cuando se requiere la resolución de algún tópico nuevo de riesgo</i> ✓ <i>Permite vuelta atrás a etapas anteriores de la espiral cuando son identificadas mejores alternativas</i> ✓ <i>Acomoda iteraciones, vuelta atrás y terminación prematura de proyectos no viables</i> ✓ <i>Identificar los riesgos asociados con cada una de las alternativas y las diferentes maneras de resolverlos.</i> ✓ <i>Es necesario definir mecanismos explícitos para incorporar objetivos de calidad en el desarrollo del producto (análisis de riesgo)</i>
<i>76. E: Se prevé la necesidad de los mismos procedimientos para desarrollo y mejoramiento.</i>
<i>77. E: Se prevé la conveniencia de los mismos procedimientos para desarrollo y mejoramiento</i>
<i>78. E: La gestión del proyecto considera la conveniencia de utilizar una metodología poco probada.</i>
<i>79. E: La gestión del proyecto considera la factibilidad de utilizar una metodología poco probada</i>
<i>80. E: Existe inexperiencia con las técnicas de ingeniería de software que se usarán</i>
<i>81. E: El usuario es muy exigente con los requerimientos del sistema respecto del producto final.</i>

SESION C.2

C.2.1 Preparación de la Sesión II

- *Información a tratar:* Agrupar en áreas las características para los ciclos de vida en Cascada, Orientado a objetos y Espiral.
- *Técnica utilizada:* Método Delphi.
- *Preparación del cuestionario.*
Se presenta una muestra del cuestionario que se entregó en el ítem C.2.5.

C.2.2 Realización de la Sesión II

- *Sesión realizada el:* 23 de febrero de 2000.
- *Expertos asistentes:* Prof. Perichinsky, Lic. Beltrami, Lic. Leone, Ing. Pollo Cattaneo, Lic. Lucchini, Ing. Weschler, Lic. Fernández.
- *Ingeniero del conocimiento:* Bibiana Rossi.
- *Lugar:* Oficina del experto. Salón con mesas de trabajo.
- *Tiempo:* 18 a 21.00hs.
- *Objetivos:*
 - Refinar y revisar los resultados de la Sesión I.
 - Agrupar en áreas las características a tener en cuenta en un proyecto en particular para seleccionar un ciclo de vida en Cascada, Orientado a objetos y/o Espiral.
- *Procedimiento:*
 - El IC explica brevemente cómo se realizó la tabulación y análisis de los resultados de la sesión I.
 - El IC presenta el cuestionario de la sesión II y el documento adjunto. El IC explica el objetivo de la sesión. Responde preguntas aclaratorias.
 - Los expertos responden el cuestionario individualmente sin interactuar entre ellos.

C.2.3 Tabulación y Análisis de los Resultados de la Sesión II

- *Conocimientos extraídos y tabulados:*

Características agrupadas por área para seleccionar un ciclo de vida en Cascada, Orientado a objetos y/o Espiral.

El documento tabulado resultante de la síntesis de las respuestas individuales de cada experto se presenta en el ítem C.2.6

- *Conocimientos a educir en próximas sesiones:*

Revisar y refinar los resultados de la primer y segunda sesión.

C.2.4 Evaluación de la Sesión II

- *¿Se han logrado los objetivos?*

Sí se han cumplido. Cada experto entregó un documento donde agrupó las características a tener en cuenta de un proyecto en particular para seleccionar los ciclos de vida Cascada, Orientado a objetos y Espiral, agrupadas a su vez en áreas.

- *¿Es necesario volver sobre lo mismo?*

Es necesario refinar la información obtenida.

- *Número y tipo de sesiones para cubrir el área*

Se estima

- 2 sesiones con el experto principal para la tabulación de análisis y resultados.
- 1 sesión con el grupo de expertos.
- 2 sesiones con el experto principal para completar la fase de adquisición de los conocimientos.

C.2.5 Cuestionario de la sesión C.2

Acerca del Proyecto

- **Proyecto:** Identificar las características a tener en cuenta en un proyecto en particular para seleccionar un ciclo de vida en Cascada, Orientado a objetos y/o Espiral.
- **Etapa del Proyecto:** Adquisición de Conocimientos.
- **Técnicas Utilizadas:**

Técnica usada	Resultado
Entrevistas	Estudio de viabilidad
Análisis de textos	Tabla de Ventajas y Desventajas de los modelos de ciclos de vida
Método Delphi	Características a analizar de un proyecto para seleccionar un modelo de ciclo de vida entre Cascada, Orientado a objetos y/o Espiral

- **Método Delphi - Pasos a seguir:**

1. Elaboración y refinamiento del cuestionario
2. Sesión I - Respuestas al cuestionario
3. Tabulación y análisis de los resultados 1er.ciclo
4. Sesión II - Respuestas y refinamiento
5. Tabulación y análisis de los resultados 2do. Ciclo
6. Sesión III - Respuestas y refinamiento
7. Tabulación y análisis de los resultados finales

Acerca de la Sesión II

- **Propósito:** Tomando como punto de partida los Resultados Tabulados de la Sesión I, identificar y agrupar las características en áreas para seleccionar el ciclo de vida en Cascada, Orientado a objetos y/o Espiral.
- **Material que se adjunta:**
 - Información acerca del proyecto
 - Información acerca de la Sesión II
 - Resultados Tabulados de la Sesión I.
 - Lista a completar para Areas de análisis
- **Procedimiento:**
 1. Leer los Resultados Tabulados de la Sesión I.
 2. Refinar los resultados analizando en cada caso:
 - a. ¿Es posible modificar algo?
 - b. ¿Es posible añadir?
 - c. ¿Es posible eliminar?
 - d. ¿Es posible sustituir un concepto por otro?
 - e. ¿Es posible combinar elementos entre sí?
 - f. ¿Se puede tomar la idea en sentido opuesto?
 - g. ¿Está asociada a uno a dos o a los tres modelos de CV a seleccionar?
 - h. ¿Corresponde efectivamente al modelo propuesto?
 3. Definir áreas en las que se puedan agrupar las características.
 4. Agrupar todas las características en las áreas definidas para seleccionar el modelo de ciclo de vida en Cascada (considerando ítem 5). Escribirlas en la lista correspondiente.
 5. Agrupar todas las características en las áreas definidas para seleccionar el modelo de ciclo de vida en Orientado a objetos (considerando ítem 5). Escribirlas en la lista correspondiente.
 6. Agrupar todas las características en las áreas definidas para seleccionar el modelo de ciclo de vida en Cascada (considerando ítem 5). Escribirlas en la lista correspondiente.

Sesión II: 28 de febrero de 2000

Experto:

Lista a completar por AREAS DE ANALISIS

Se adjuntan además los Resultados Tabulados de la Sesión I.

C.2.6 Resultados de la sesión C.2

RESULTADO TABULADO DE LA SESIÓN II

Sesión II: 28 de febrero de 2000

Experto: Equipo de expertos

MODELOS: CASCADA, ORIENTADO A OBJETOS y ESPIRAL

AREA: ESPECIFICACIÓN DE REQUERIMIENTOS

1. *C: El usuario ha explicitado la mayoría de los requerimientos al comienzo del proyecto.*
2. *C: El usuario ha explicitado los requisitos formalmente*
3. *C: El usuario ha explicitado los requisitos exhaustivamente*
4. *C: El usuario ha explicitado los requisitos uniformemente*
5. *C: Los límites del sistema están claramente definidos*

6. *O: No se han definido claramente los límites del sistema*
7. *O: No se han definido claramente los requisitos del sistema*

8. *E: El entorno del sistema es de alta incertidumbre*
9. *E: hay gran nivel de incertidumbre en la especificación de los requisitos*
 - *El cliente no ha explicitado la mayoría de los requisitos*
 - *Las especificaciones no son uniformes, exhaustivas o formales*
 - *Los límites del sistema no se definen claramente.*
10. *E: El usuario es muy exigente con los requerimientos del sistema respecto del producto final*

MODELOS: CASCADA, ORIENTADO A OBJETOS y ESPIRAL

AREA: CARACTERISTICAS DEL CICLO DE VIDA

11. *C: Se puede asumir una progresión relativamente uniforme y secuencial en las fases del ciclo de vida*
12. *C: Se presume la necesidad de retroalimentación en las fases del desarrollo del proyecto*
 - *Se presume independencia entre las fases del desarrollo del proyecto*
 - *Se presume poca dependencia entre las fases de desarrollo del proyecto*

13. *O: Se puede asumir una progresión NO uniforme y secuencial en las fases del ciclo de vida*

14.O: Se presume la necesidad de retroalimentación en las fases del desarrollo del proyecto

- Se presume independencia entre las fases del desarrollo del proyecto
- Se presume poca dependencia entre las fases de desarrollo del proyecto
- Se presume mucha dependencia entre las fases de desarrollo del proyecto

15.E: Se puede asumir una progresión NO uniforme y secuencial en las fases del ciclo de vida

16.E: Se presume la necesidad de retroalimentación en las fases del desarrollo del proyecto

- Se presume independencia entre las fases del desarrollo del proyecto
- Se presume poca dependencia entre las fases de desarrollo del proyecto
- Se presume mucha dependencia entre las fases de desarrollo del proyecto

MODELOS: CASCADA, ORIENTADO A OBJETOS y ESPIRAL

AREA: CARACTERISTICAS DEL TIPO DE APLICACION

17.C: El sistema presenta subsistemas medianamente complejos

- Es necesario descomponer el sistema en pocos niveles para su mejor comprensión
- Es necesario descomponer el sistema en pocos niveles para su mejor manipulación

18.C: El sistema presenta subsistemas poco complejos

- Es necesario descomponer el sistema en pocos niveles para su mejor comprensión
- Es necesario descomponer el sistema en pocos niveles para su mejor manipulación

19.C: el sistema tiene componentes de fuerte contenido algorítmico

20.C: el tipo de aplicación presenta un comportamiento predominantemente BATCH

21.O: El proyecto requiere el descubrimiento de problemas de arquitectura en etapas tempranas del desarrollo

22.O: El proyecto requiere el descubrimiento de problemas de eficacia en etapas tempranas del desarrollo

23.O: El sistema a diseñar es innovador

24.O: El sistema se basa en componentes de fuerte contenido matemático

25.O: El sistema se basa en componentes de fuerte contenido gráfico

26.O: El sistema tiene subsistemas que presentan comportamiento dinámico

27.O: El tipo de aplicación presenta comportamiento fuertemente interactivo

28.O: El tipo de aplicación presenta comportamiento en Tiempo Real

29.O: Se prevé una alta variabilidad (modificaciones en las funciones) en los procesos del sistema

30. O: El sistema es explícitamente sensible a cambios, extensiones y/o ampliaciones.
31. O: El tipo de aplicación es desarrollo de software de base
32. O: El tipo de aplicación es un SSBBCC sistema basado en conocimientos
33. O: El sistema presenta subsistemas complejos
- Es necesario descomponer el sistema en varios niveles para su mejor comprensión
 - Es necesario descomponer el sistema en varios niveles para su mejor manipulación
34. E: El tipo de aplicación a diseñar es innovador
35. E: El sistema a desarrollar puede resolverse por varias opciones de diseño
36. E: El sistema a desarrollar puede resolverse por varias opciones de implementación
37. E: El proyecto requiere el descubrimiento de problemas de arquitectura del sistema en etapas tempranas del desarrollo
38. E: El proyecto requiere el descubrimiento de problemas de eficacia (en la arquitectura del sistema) en etapas tempranas del desarrollo
39. E: El tipo de aplicación requiere el desarrollo fuertemente integrado de hardware y software
40. E: El tipo de sistema a desarrollar es un SSBBCC sistema basado en conocimientos
41. E: El tipo de sistema a desarrollar es algorítmico
42. E: Se prevé en el desarrollo del sistema crecimientos y cambios del producto
- El tipo de sistema a desarrollar es software de base
43. E: El sistema presenta subsistemas complejos
- Es necesario descomponer el sistema en varios niveles para su mejor comprensión
 - Es necesario descomponer el sistema en varios niveles para su mejor manipulación
44. E: El tipo de aplicación presenta características fuertemente interactivas
45. E: El tipo de aplicación presenta características de Tiempo Real
46. E: El tipo de aplicación a desarrollar es un sistema nuevo totalmente

MODELOS: CASCADA, ORIENTADO A OBJETOS y ESPIRAL

AREA: CARACTERISTICAS DE LA GESTION DE PROYECTO

47. C: No será necesario disponer de una versión temprana del software desarrollado hasta que el proyecto este avanzado.
48. C: Se terceriza el desarrollo y se requiere un ajustado control de la gestión del proyecto. (cascada, facilita el control del proyecto cuando se terceriza el desarrollo)
49. C: El proyecto requiere una gestión de proyecto muy formal.
50. C: No se dispone de software para prototipar, ni es factible adquirirlo.

51.C: Se prevé la necesidad de usar metodologías ampliamente probadas.

52.O: El proyecto requiere una gestión de proyecto medianamente formal.

53.O: El proyecto requiere una gestión de proyecto poco formal.

54.O: Se prevé una fuerte necesidad de reutilizar los componentes de un proyecto para el siguiente

55.O: Es un subsistema de un sistema desarrollado en Objetos

56.O: Es una ampliación de un sistema desarrollado en Objetos

57.O: Es una modificación de un sistema desarrollado en Objetos

58.O: Se prevé una fuerte necesidad de reutilizar los componentes de un software ya existente (desarrollado en Objetos)

59.O: Es necesario disponer de una versión temprana del software desarrollado

60.O: Se prevé la necesidad de usar metodologías medianamente probadas

61.O: Se presume riesgo en el desarrollo del sistema.

62.E: Es necesario facilitar la introducción gradual del sistema para los usuarios

63.E: Es necesario contar con una fuerte participación del cliente en el Desarrollo

64.E: Se cuenta con habilidad para la evaluación del riesgo

- Se cuenta con métodos y técnicas para evaluar el riesgo

65.E: La gestión del proyecto requiere de acuerdos confirmados para los cambios a efectuarse durante el desarrollo del proyecto

66.E: La gestión del proyecto requiere entregas parciales del sistema con funciones que se irán completando gradualmente.

67.E: La gestión del proyecto requiere entregas tempranas del sistema con funciones que se irán completando gradualmente.

68.E: La gestión del proyecto prevé definir explícitamente estrategias para reusar software existente.

69.E: No hay experiencia previa en el sistema a desarrollar

70.E: Es necesario disponer de una versión temprana del software desarrollado

71.E: El desarrollo del sistema es responsabilidad de la organización, no se terceriza

72.E: Se dispone de software para prototipar

73.E: Es factible adquirir software para prototipar

74.E: Existen importantes dudas sobre la viabilidad del software

75.E: Se presume alto factor de riesgo en el desarrollo del sistema

- Es necesario definir explícitamente estrategias para realizar análisis de riesgo

✓ Permite vuelta atrás a etapas anteriores cuando se requiere la resolución de algún tópico nuevo de riesgo

✓ Permite vuelta atrás a etapas anteriores de la espiral cuando son identificadas mejores alternativas.

✓ Acomoda iteraciones, vuelta atrás y terminación prematura de proyectos no viables.

✓ Identificar los riesgos asociados con cada una de las alternativas y las diferentes maneras de resolverlos.

✓ Es necesario definir mecanismos explícitos para incorporar objetivos de calidad en el desarrollo del producto (análisis de riesgo)

76.E: Se estiman riesgos técnicos en el desarrollo del sistema

- Es necesario definir explícitamente estrategias para realizar análisis de riesgo
 - ✓ Permite vuelta atrás a etapas anteriores cuando se requiere la resolución de algún tópico nuevo de riesgo
 - ✓ Permite vuelta atrás a etapas anteriores de la espiral cuando son identificadas mejores alternativas.
 - ✓ Acomoda iteraciones, vuelta atrás y terminación prematura de proyectos no viables.
 - ✓ Identificar los riesgos asociados con cada una de las alternativas y las diferentes maneras de resolverlos.
 - ✓ Es necesario definir mecanismos explícitos para incorporar objetivos de calidad en el desarrollo del producto (análisis de riesgo)

77.E: Se prevé la necesidad de los mismos procedimientos para desarrollo y mejoramiento.

78.E: Se prevé la conveniencia de los mismos procedimientos para desarrollo y mejoramiento

79.E: La gestión del proyecto considera la conveniencia de utilizar una metodología poco probada.

80.E: La gestión del proyecto considera la factibilidad de utilizar una metodología poco probada

81.E: Existe inexperiencia con las técnicas de ingeniería de software que se usarán

SESION C.3

C.3.1 Preparación de la Sesión III

- *Información a tratar:* Revisar los resultados obtenidos en las sesiones anteriores. Las características agrupadas por áreas para analizar en un proyecto para la selección entre los ciclos de vida en Cascada, Orientado a objetos y Espiral.
- *Técnica utilizada:* Método Delphi - Técnica Nominal de Grupo
- *Preparación de la sesión:*
No se ha preparado un cuestionario. Se presentan los resultados de la sesión anterior y se solicita que lo revisen.

C.3.2 Realización de la Sesión III

- *Sesión realizada el:* 9 de marzo de 2000.
- *Expertos asistentes:* Prof. Perichinsky, Lic. Beltrami, Lic. Leone, Ing. Pollo Cattaneo, Lic. Lucchini, Ing. Weschler, Lic. Fernández
- *Ingeniero del conocimiento:* Bibiana Rossi.
- *Lugar:* Oficina del experto. Salón de conferencias con mesas de trabajo.
- *Tiempo:* 18 a 21.00hs.
- *Objetivos:*
 - Refinar y revisar los resultados de la Sesión II.
 - Discutir alguna diferencia de opinión y semántica respecto de los resultados.
- *Procedimiento:*
 - El IC presenta el resultado de la sesión II. El IC explica el objetivo de la sesión.
 - Los expertos revisan los resultados individualmente y van registrando sus ideas para discutir.
 - Se discuten secuencialmente las ideas y se vota.
 - El IC toma nota del resultado de la votación.

- *Síntesis de los temas discutidos:*

- Se revisaron las áreas definidas:
 - ✓ Respecto del área Características del Ciclo de Vida, las características agrupadas no se correspondían con el ciclo de vida, que es justamente lo que se desea como respuesta del sistema, sino que se corresponde con características que se estiman respecto de la fase de desarrollo del proyecto.
 - ✓ Se cambia la redacción de las características que mencionan “ciclo de vida” por “fases de desarrollo del proyecto”.
 - ✓ Las características definidas en Características del Ciclo de Vida se incorporan al área Características del Tipo de Aplicación.
 - ✓ Desaparece el área Características del ciclo de vida.
 - ✓ Las áreas tal como quedan definidas: Especificación de Requerimientos, Tipo de Aplicación y Gestión de Proyecto se consideran correctas y el nombre asignado representativo.
- En ampliaciones posteriores sería conveniente agregar más información y seguramente se encontrarían más áreas de análisis respecto de un proyecto.
- La conveniencia de contar con el prototipo lo antes posible como forma de motivar la participación del equipo de expertos.
- Considerar la posibilidad de seleccionar un ciclo de vida en espiral pero usando técnicas de objetos: se dijo que era posible, que era necesario considerarlo para próximas ampliaciones del prototipo pero que resultaba más conveniente contar con un sistema funcionando antes de pensar en analizar más variables y sus combinaciones.

C.3.3 Tabulación y Análisis de los Resultados de la Sesión III

- *Conocimientos extraídos y tabulados:*

Características agrupadas por área para seleccionar un ciclo de vida en Cascada, Orientado a objetos y/o Espiral.
 El documento tabulado resultante de la presente sesión se presenta en el ítem C3.5.

- *Conocimientos a educir en próximas sesiones:*

Realizar el análisis de la información obtenida hasta el momento en la fase de adquisición de conocimientos y ver qué otras técnicas es posible aplicar (emparrillado, análisis de protocolo, etc.).

C.3.4 Evaluación de la Sesión III

- *¿Se han logrado los objetivos?*

Sí se han cumplido. Se ha refinado la información obtenida en las sesiones con el grupo de expertos.

- *¿Es necesario volver sobre lo mismo?*

No parece necesario, sólo cuando surjan dudas puntuales será necesario consultar al experto principal.

- *Número y tipo de sesiones para cubrir el área*

Se estima:

- 1 sesión con el experto principal para la tabulación de análisis y resultados.
- 2 sesiones con el experto principal para completar la fase de adquisición de los conocimientos.

C.3.5 Resultados de la sesión C.3

RESULTADO TABULADOS DE LA SESION III

Sesión III: 9 de marzo de 2000

Experto: Equipo de expertos

MODELOS: CASCADA, ORIENTADO A OBJETOS y ESPIRAL

AREA: ESPECIFICACIÓN DE REQUERIMIENTOS

1. *C: El usuario ha explicitado la mayoría de los requerimientos al comienzo del proyecto.*
2. *C: El usuario ha explicitado los requisitos formalmente*
3. *C: El usuario ha explicitado los requisitos exhaustivamente*
4. *C: El usuario ha explicitado los requisitos uniformemente*
5. *C: Los límites del sistema están claramente definidos*
6. *O: No se han definido claramente los límites del sistema*
7. *O: No se han definido claramente los requisitos del sistema*
8. *E: El entorno del sistema es de alta incertidumbre*
9. *E: hay gran nivel de incertidumbre en la especificación de los requisitos*
 - *El cliente no ha explicitado la mayoría de los requisitos*
 - *Las especificaciones no son uniformes, exhaustivas o formales*
 - *Los límites del sistema no se definen claramente.*
10. *E: El usuario es muy exigente con los requerimientos del sistema respecto del producto final*

MODELOS: CASCADA, ORIENTADO A OBJETOS y ESPIRAL

AREA: TIPO DE APLICACION

11. *C: El sistema presenta subsistemas medianamente complejos*
 - *Es necesario descomponer el sistema en pocos niveles para su mejor comprensión*
 - *Es necesario descomponer el sistema en pocos niveles para su mejor manipulación*
12. *C: El sistema presenta subsistemas poco complejos*
 - *Es necesario descomponer el sistema en pocos niveles para su mejor comprensión*
 - *Es necesario descomponer el sistema en pocos niveles para su mejor manipulación*

13. C: el sistema tiene componentes de fuerte contenido algorítmico
14. C: el tipo de aplicación presenta un comportamiento predominantemente BATCH
15. C: Se puede asumir una progresión relativamente uniforme y secuencial en las fases de desarrollo del proyecto
16. C: Se presume la necesidad de retroalimentación en las fases del desarrollo del proyecto
 - Se presume independencia entre las fases del desarrollo del proyecto
 - Se presume poca dependencia entre las fases de desarrollo del proyecto
17. O: El proyecto requiere el descubrimiento de problemas de arquitectura en etapas tempranas del desarrollo
18. O: El proyecto requiere el descubrimiento de problemas de eficacia en etapas tempranas del desarrollo
19. O: El sistema a diseñar es innovador
20. O: El sistema se basa en componentes de fuerte contenido matemático
21. O: El sistema se basa en componentes de fuerte contenido gráfico
22. O: El sistema tiene subsistemas que presentan comportamiento dinámico
23. O: El tipo de aplicación presenta comportamiento fuertemente interactivo
24. O: El tipo de aplicación presenta comportamiento en Tiempo Real
25. O: Se prevé una alta variabilidad (modificaciones en las funciones) en los procesos del sistema
26. O: El sistema es explícitamente sensible a cambios, extensiones y/o ampliaciones.
27. O: El tipo de aplicación es desarrollo de software de base
28. O: El tipo de aplicación es un SSBCC sistema basado en conocimientos
29. O: El sistema presenta subsistemas complejos
 - Es necesario descomponer el sistema en varios niveles para su mejor comprensión
 - Es necesario descomponer el sistema en varios niveles para su mejor manipulación
30. O: Se puede asumir una progresión NO uniforme y secuencial en las fases de desarrollo del proyecto
31. O: Se presume la necesidad de retroalimentación en las fases del desarrollo del proyecto
 - Se presume independencia entre las fases del desarrollo del proyecto
 - Se presume poca dependencia entre las fases de desarrollo del proyecto
 - Se presume mucha dependencia entre las fases de desarrollo del proyecto
32. E: El tipo de aplicación a diseñar es innovador
33. E: El sistema a desarrollar puede resolverse por varias opciones de diseño
34. E: El sistema a desarrollar puede resolverse por varias opciones de implementación
35. E: El proyecto requiere el descubrimiento de problemas de arquitectura del sistema en etapas tempranas del desarrollo

36. *E: El proyecto requiere el descubrimiento de problemas de eficacia (en la arquitectura del sistema) en etapas tempranas del desarrollo*
37. *E: El tipo de aplicación requiere el desarrollo fuertemente integrado de hardware y software*
38. *E: El tipo de sistema a desarrollar es un SSBBCC sistema basado en conocimientos*
39. *E: El tipo de sistema a desarrollar es algorítmico*
40. *E: Se prevé en el desarrollo del sistema crecimientos y cambios del producto*
 - *El tipo de sistema a desarrollar es software de base*
41. *E: El sistema presenta subsistemas complejos*
 - *Es necesario descomponer el sistema en varios niveles para su mejor comprensión*
 - *Es necesario descomponer el sistema en varios niveles para su mejor manipulación*
42. *E: El tipo de aplicación presenta características fuertemente interactivas*
43. *E: El tipo de aplicación presenta características de Tiempo Real*
44. *E: El tipo de aplicación a desarrollar es un sistema nuevo totalmente*
45. *E: Se puede asumir una progresión NO uniforme y secuencial en las fases de desarrollo del proyecto*
46. *E: Se presume la necesidad de retroalimentación en las fases del desarrollo del proyecto*
 - *Se presume independencia entre las fases del desarrollo del proyecto*
 - *Se presume poca dependencia entre las fases de desarrollo del proyecto*
 - *Se presume mucha dependencia entre las fases de desarrollo del proyecto*

MODELOS: CASCADA, ORIENTADO A OBJETOS y ESPIRAL

AREA: GESTION DE PROYECTO

47. *C: No será necesario disponer de una versión temprana del software desarrollado hasta que el proyecto este avanzado.*
48. *C: Se terceriza el desarrollo y se requiere un ajustado control de la gestión del proyecto. (cascada, facilita el control del proyecto cuando se terceriza el desarrollo)*
49. *C: El proyecto requiere una gestión de proyecto muy formal.*
50. *C: No se dispone de software para prototipar, ni es factible adquirirlo.*
51. *C: Se prevé la necesidad de usar metodologías ampliamente probadas.*
52. *O: El proyecto requiere una gestión de proyecto medianamente formal.*
53. *O: El proyecto requiere una gestión de proyecto poco formal.*
54. *O: Se prevé una fuerte necesidad de reutilizar los componentes de un proyecto para el siguiente*
55. *O: Es un subsistema de un sistema desarrollado en Objetos*
56. *O: Es una ampliación de un sistema desarrollado en Objetos*

- 57.O: Es una modificación de un sistema desarrollado en Objetos
 58.O: Se prevé una fuerte necesidad de reutilizar los componentes de un software ya existente (desarrollado en Objetos)
 59.O: Es necesario disponer de una versión temprana del software desarrollado
 60.O: Se prevé la necesidad de usar metodologías medianamente probadas
 61.O: Se presume riesgo en el desarrollo del sistema.
- 62.E: Es necesario facilitar la introducción gradual del sistema para los usuarios
 63.E: Es necesario contar con una fuerte participación del cliente en el Desarrollo
 64.E: Se cuenta con habilidad para la evaluación del riesgo
 - Se cuenta con métodos y técnicas para evaluar el riesgo
 65.E: La gestión del proyecto requiere de acuerdos confirmados para los cambios a efectuarse durante el desarrollo del proyecto
 66.E: La gestión del proyecto requiere entregas parciales del sistema con funciones que se irán completando gradualmente.
 67.E: La gestión del proyecto requiere entregas tempranas del sistema con funciones que se irán completando gradualmente.
 68.E: La gestión del proyecto prevé definir explícitamente estrategias para reusar software existente.
 69.E :No hay experiencia previa en el sistema a desarrollar
 70.E: Es necesario disponer de una versión temprana del software desarrollado
 71.E: El desarrollo del sistema es responsabilidad de la organización, no se terceriza
 72.E: Se dispone de software para prototipar
 73.E: Es factible adquirir software para prototipar
 74.E: Existen importantes dudas sobre la viabilidad del software
 75.E: Se presume alto factor de riesgo en el desarrollo del sistema
 - Es necesario definir explícitamente estrategias para realizar análisis de riesgo
 - ✓ Permite vuelta atrás a etapas anteriores cuando se requiere la resolución de algún tópico nuevo de riesgo
 - ✓ Permite vuelta atrás a etapas anteriores de la espiral cuando son identificadas mejores alternativas.
 - ✓ Acomoda iteraciones, vuelta atrás y terminación prematura de proyectos no viables.
 - ✓ Identificar los riesgos asociados con cada una de las alternativas y las diferentes maneras de resolverlos.
 - ✓ Es necesario definir mecanismos explícitos para incorporar objetivos de calidad en el desarrollo del producto (análisis de riesgo)
 76.E: Se estiman riesgos técnicos en el desarrollo del sistema
 - Es necesario definir explícitamente estrategias para realizar análisis de riesgo
 - ✓ Permite vuelta atrás a etapas anteriores cuando se requiere la resolución de algún tópico nuevo de riesgo
 - ✓ Permite vuelta atrás a etapas anteriores de la espiral cuando son identificadas mejores alternativas.

- ✓ Acomoda iteraciones, vuelta atrás y terminación prematura de proyectos no viables.
 - ✓ Identificar los riesgos asociados con cada una de las alternativas y las diferentes maneras de resolverlos.
 - ✓ Es necesario definir mecanismos explícitos para incorporar objetivos de calidad en el desarrollo del producto (análisis de riesgo)
- 77.E: Se prevé la necesidad de los mismos procedimientos para desarrollo y mejoramiento.
- 78.E: Se prevé la conveniencia de los mismos procedimientos para desarrollo y mejoramiento
- 79.E: La gestión del proyecto considera la conveniencia de utilizar una metodología poco probada.
- 80.E: La gestión del proyecto considera la factibilidad de utilizar una metodología poco probada
- 81.E: Existe inexperiencia con las técnicas de ingeniería de software que se usarán

5.5 TEORIA DE LA CONSTRUCCION PERSONAL O EMPARRILLADO

SESION D.1

D.1.1 Preparación de la Sesión I

- *Información a tratar:* Revisar la consistencia de los resultados obtenidos de la sesión III en el proceso de adquisición de conocimientos a partir de un equipo de expertos.
- *Técnica utilizada:* Emparrillado
- *Pasos a seguir:*
 - Identificación de los elementos
 - Identificación de las características
 - Diseño de la parrilla
 - Formalización
 - Análisis de resultados

D.1.2 Identificación de los elementos

El experto ha señalado como elementos los tres ciclos de vida que serán considerados en el primer prototipo del sistema experto.

- E1 Ciclo de vida en Cascada
- E2 Ciclo de vida Orientado a Objetos
- E3 Ciclo de vida en Espiral

D.1.3 Identificación de las características

El experto ha seleccionado las características para realizar el emparrillado, tomando como punto de partida los resultados finales obtenidos a partir de un equipo de expertos. El punto de partida son entonces el conjunto de características, organizadas por áreas, a tener en cuenta de un proyecto, para la selección del modelo de ciclo de vida más adecuado entre Cascada, Orientado a objetos y Espiral.

Dado que los valores de las características deben ser bipolares, se detalla a continuación cada una de ellas:

- C1 a- Los requerimientos están exhaustivamente explicitados al comienzo
 - b- Los requerimientos están incompletos al comienzo

- C2 a- Existe independencia en las fases de desarrollo
b- Existe dependencia en las fases de desarrollo
- C3 a- Los subsistemas componentes son complejos
b- Los subsistemas componentes son simples
- C4 a- La aplicación es predominantemente interactiva
b- La aplicación es predominantemente "batch"
- C5 a- La aplicación es de fuerte contenido matemático
b- La aplicación es de bajo contenido matemático
- C6 a- La aplicación es de fuerte contenido gráfico
b- La aplicación es de bajo contenido gráfico
- C7 a- La aplicación es desarrollo de software de base
b- La aplicación no es desarrollo de software de base
- C8 a- La aplicación es desarrollo de software basado en conocimiento
b- La aplicación no es desarrollo de software basado en conocimiento
- C9 a- Se dispone de software para prototipar
b- No se dispone de software para prototipar
- C10 a- Es necesario disponer de una versión temprana del software
b- No es necesario disponer de una versión temprana del software
- C11 a- Se presume alto factor de riesgo durante el desarrollo
b- Se presume bajo factor de riesgo durante el desarrollo
- C12 a- El desarrollo del sistema es responsabilidad de terceros
b- El desarrollo del sistema es responsabilidad de la organización

D.1.4 Diseño de la parrilla

Para el armado de la parrilla se definen cada uno de los elementos como las columnas de la parrilla y cada una de las características polarizadas como las filas de la parrilla, tal como se muestra en la figura 5-1.

De las tres formas posibles de construcción de la parrilla, dicotómica, clasificatoria o evaluativa, se ha seleccionado la evaluativa. El experto asigna valores entre 1 y 10 considerando para la característica analizada el grado de relación con cada uno de los elementos, tal como se muestra en la figura 5-1.

Los valores se corresponden de la siguiente manera: "*la característica Cx se encuentra n relacionada con el elemento Ey*". La relación es mayor cuanto mayor es el valor asignado entre 1 y 10, siendo para los valores extremos:

- 1: no relacionada
- 10: totalmente relacionada

	E1 CASCADA	E2 OBJETOS	E3 ESPIRAL	
C1: REQ. EXPLICITADOS	10	6	3	C1: REQ. INCOMPLETOS
C2: INDEPENDENCIA	8	4	2	C2: DEPENDENCIA
C3: SUB. COMPLEJOS	3	8	10	C3: SUB. SIMPLES
C4: APL. INTERACTIVA	2	10	6	C4: APL. BATCH
C5: FUERTE MATEMATICO	7	10	5	C5: BAJO MATEMATICO
C6: FUERTE GRAFICO	1	10	7	C6: BAJO GRAFICO
C7: SOFT. BASE	1	10	8	C7: NO SOFT. BASE
C8: SOFT. SSBCC	1	6	9	C8: NO SOFT. SSBCC
C9: PROT. DISPONIBLE	1	8	10	C9: PROT. NO DISPONIBLE
C10: VERSION TEMPRANA	1	7	10	C10: NO VERSION TEMPRANA
C11: ALTO RIESGO	1	6	10	C11: BAJO RIESGO
C12: RESP. TERCEROS	9	5	1	C12: RESP. ORGANIZACION

FIGURA 5 -1: *Parrilla Evaluada*

D.1.5 Formalización

La parrilla evaluada se estudia en dos direcciones, la clasificación de los elementos y la clasificación de las características. Se realiza el cálculo de la parrilla siguiendo el desarrollo propuesto por Gómez [Gómez, A. y otros 1997]. En ambos casos se ha aplicado la convención de distancia mínima.

D.1.5.1 Clasificación de los elementos

	E1	E2	E3
E1		69	82
E2			37
E3			

FIGURA 5 - 2: Parrilla de Distancias entre elementos

	E2-E3	E1
E2-E3		69
E1		

FIGURA 5 - 2: Matriz obtenida por Distancia Mínima en la Primer Iteración

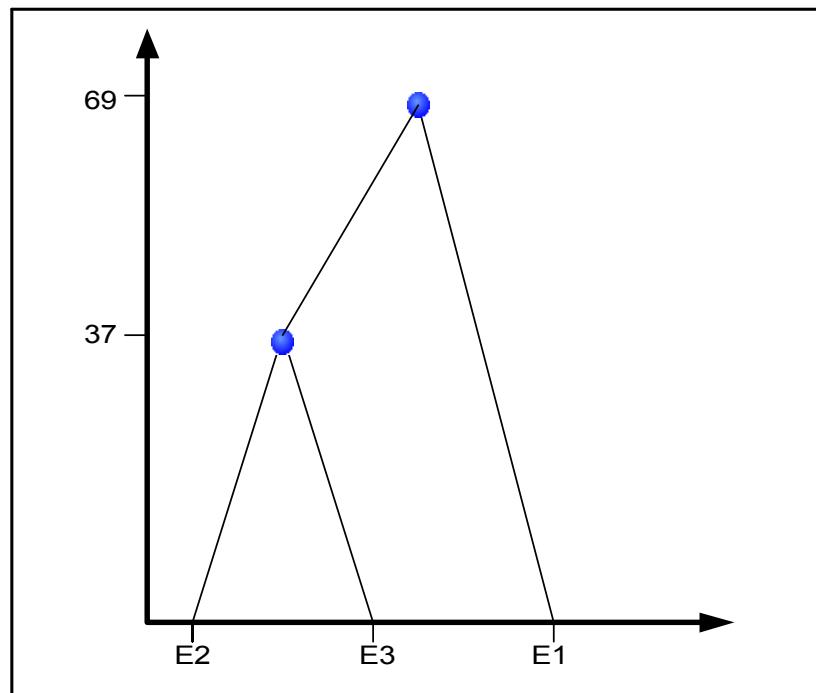


FIGURA 5 - 3: Arbol ordenado de la matriz de Distancias de los Elementos

D.1.5.2 Clasificación de las características

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
C1		5	16	15	9	17	18	15	18	17	16	4
C2	14		17	16	10	18	19	16	19	18	17	3
C3	7	2		7	11	7	6	5	2	3	4	18
C4	8	7	18		6	2	3	8	7	8	9	17
C5	14	11	12	11		8	9	14	13	14	15	11
C6	6	7	20	19	13		1	6	5	6	7	19
C7	5	6	21	20	14	22		5	4	5	6	20
C8	2	3	18	17	11	19	20		3	2	1	17
C9	5	4	21	20	14	22	23	20		1	2	20
C10	4	3	20	19	13	21	22	19	22		1	19
C11	3	4	19	18	12	20	21	18	21	20		18
C12	15	16	3	8	14	8	7	2	3	2	1	

FIGURA 5-4: Matriz de Distancias para las Características

	NC1	NC2	C3	C4	C5	C6	C7	C8	C9	C10	C11	NC12
NC1		5	7	8	9	6	5	2	5	4	3	4
NC2			2	7	10	7	6	3	4	3	4	3
C3				7	11	7	6	5	2	3	4	3
C4					6	2	3	8	7	8	9	8
C5						8	9	11	13	13	12	11
C6							1	6	5	6	7	8
C7								5	4	5	6	7
C8									3	2	1	2
C9										1	2	3
C10											1	2
C11												1
NC12												

FIGURA 5-5: Matriz Diagonal Superior de Características en su Primer Iteración.

	C6-C7	C8-C9-C10-C11-NC12	NC1	NC2	C3	C4	C5
C6-C7		4	5	6	6	2	8
C8-C9-C10-C11-NC12			2	3	2	7	11
NC1					5	7	8
NC2						2	7
C3							7
C4							6
C5							

FIGURA 5-6: Matriz Diagonal Superior de Características en su Segunda Iteración.

	[C6-C7] [C4]	[C8-C9-C10-C11-NC12] [NC1-NC2-C3]	C5
[C6-C7] [C4]		4	6
[C8-C9-C10-C11-NC12] [NC1-NC2-C3]			9
C5			

FIGURA 5-7: Matriz Diagonal Superior de Características en su Tercera Iteración.

	[C6-C7] [C4] [C8-C9-C10-C11-NC12] [NC1-NC2-C3]	C5
[C6-C7] [C4] [C8-C9-C10-C11-NC12] [NC1-NC2-C3]		6
C5		

FIGURA 5-8: Matriz Diagonal Superior de Características en su Cuarta Iteración.

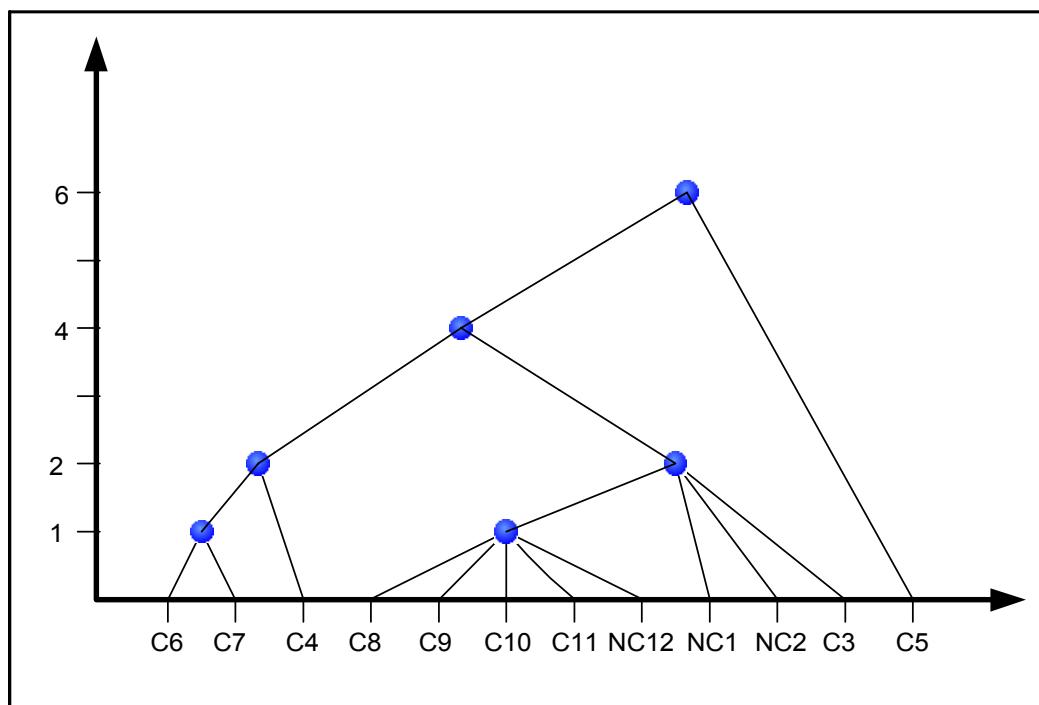


FIGURA 5-9: Árbol ordenado de la matriz de distancias de las características

D.1.6 Análisis de los resultados

La lectura e interpretación de los gráficos obtenidos permite encontrar relaciones entre los elementos y características que deben ser analizados y corroborados con el experto. Para ello se concreta una entrevista con el experto con el objetivo de analizar los resultados.

SESION A.4

A.4.1 Preparación de la Sesión IV

- *Información a tratar:* Evaluación de los árboles ordenados de elementos y características obtenidos durante la etapa de Formalización de la técnica de emparrillado.
- *Amplitud y Profundidad:* Corroborar el grado de correspondencia que existe entre el modelo mental del experto y la representación obtenida durante el emparrillado.
- *Técnica utilizada:* Entrevista estructurada.
- *Preparación de Preguntas:*
 - ✓ ¿Cuál de los tres modelos de ciclos de vida seleccionados para este proyecto es más rápidamente distinguible frente a las características particulares de un proyecto? (E2 - E3)
 - ✓ De los tres modelos de ciclos de vida seleccionados para este proyecto, ¿cuáles tienen más características comunes? (E2 - E3)
 - ✓ ¿Cuál es la relación entre una aplicación de fuerte contenido gráfico y que sea software de base? (C6-C7)
 - ✓ ¿Tiene sentido pensar en una aplicación que sea software de base, de fuerte contenido gráfico, y predominantemente interactiva? (C6-C7-C4)
 - ✓ ¿Es posible que una misma aplicación reúna las siguientes características: los requerimientos están incompletos al comienzo, existe dependencia en las fases de desarrollo y subsistemas componentes complejos? (NC1-NC2-C3)
 - ✓ ¿Cómo afecta a una aplicación si esta es de fuerte contenido matemático? (C5)
 - ✓ ¿Cuál es la relación entre que la aplicación sea un desarrollo de software basado en conocimiento, disponer de software para prototipar, que sea necesario disponer de una versión temprana del software, que se presuma

alto factor de riesgo y que el desarrollo del sistema sea responsabilidad de la organización? (C8-C9-C10-C11-NC12)

A.4.2 Realización de la Sesión IV

La entrevista se realiza en la oficina del experto, habiendo acordado previamente la hora de inicio y fin de la entrevista. Se le explica al experto el objetivo de la entrevista. El entrevistado se muestra muy dispuesto y manifiesta curiosidad por conocer los resultados obtenidos de la técnica de emparrillado. Se desarrolla la sesión de la entrevista.

A.1.3 Transcripción de la Sesión IV

Entrevista realizada el 23 de febrero de 2000

Experto: Dr. Gregorio Perichinsky

Ingeniero del Conocimiento: Bibiana Rossi

Lugar: oficina del experto

Tiempo: 14 a 15 hs.

Objetivos: Evaluar los resultados obtenidos con la técnica de emparrillado

I. *¿Cuál de los tres modelos de ciclos de vida seleccionados para este sistema experto es más rápidamente distinguible frente a las características particulares de un proyecto?*

E. El modelo en cascada es el que se identifica con mayores diferencias respecto de los otros dos. Tanto el modelo en espiral como el modelo de objetos prevén el uso de prototipos, tienen algunas similitudes entre ellos y se diferencian del ciclo en cascada que es más rígido en su concepción.

I. *De los tres modelos de ciclos de vida seleccionados para este proyecto ¿cuáles tienen más características comunes?*

E. Como dije recién el ciclo de vida en espiral y el ciclo de vida orientado a objetos comparten la filosofía de prototipado.

I. *¿Cuál es la relación entre una aplicación de fuerte contenido gráfico y que sea software de base? ¿Todo software de base es de alto contenido gráfico?*

E. El software de base puede o no ser de alto contenido gráfico, pero una aplicación de fuerte contenido gráfico es altamente probable que sea software de base o que haya sido desarrollada con software de base de alto contenido grafico. La relación por lo tanto no es recíproca.

Una aplicación de alto contenido grafico, es generalmente usada por profesionales que requieren que vistas y planos de objetos, es una aplicación de “servicio” a profesionales y es software de base. Un ejemplo típico de esto es por ejemplo el Autocad o el CAD-CAM, Si la aplicación a desarrollar es de fuerte contenido grafico requiere de software de base de fuerte contenido gráfico. Por ejemplo el software usado para efectos especiales cinematográficos o para dibujos animados.

IC. ¿Tiene sentido pensar en una aplicación que sea software de base, de fuerte contenido gráfico y predominantemente interactiva?

E. Absolutamente. Por la calidad de servicio que presta el software de base es justamente predominantemente interactivo. Los ejemplos que recién mencioné tienen claramente esa característica.

IC. ¿Es posible que una misma aplicación reúna las siguientes características: los requerimientos están incompletos al comienzo, existe dependencia en las fases de desarrollo y los subsistemas componentes son complejos?

E. Los requisitos claramente explícitos están relacionados fuertemente con el ciclo de vida en cascada. Pero esto no define que no pueda seleccionarse otro ciclo de vida, simplemente facilitará más el desarrollo del sistema. Si los requisitos no están claramente definidos entonces se hace necesario el uso de prototipos. Tanto el ciclo de vida en espiral como objetos consideran el uso de prototipos. En síntesis si los requisitos están incompletos puede ser conveniente realizar algunas versiones de prototipo o versiones tempranas para ayudar a definirlos y explicitarlos. Si los subsistemas son complejos es muy razonable que los requisitos no se puedan definir con facilidad y por eso están incompletos al comienzo.

IC. ¿Cómo afecta una aplicación si esta es de fuerte contenido matemático?

E. No mucho, puede o no ser de fuerte o bajo contenido matemático y eso no incide respecto de la complejidad de los subsistemas, ni de la necesidad de prototipación.

IC. ¿Cuál es la relación entre: la aplicación es un desarrollo de software basado en conocimiento, se dispone de software para prototipar, es necesario disponer de una versión temprana del software, que se presume alto factor de riesgo y que la responsabilidad del desarrollo sea de la organización?

E. Si una organización necesita un software basado en conocimiento, el alto factor de riesgo puede estar asociado a que este tipo de aplicaciones no son masivamente incluidas en las organizaciones excepto en aquellos casos que un problema no se pueda resolver por los desarrollos tradicionales y resulte costoso, eso hace que se asocie intuitivamente con alto factor de riesgo. En estos casos de riesgo una versión temprana del software que facilite el análisis de riesgo durante el desarrollo es lo mas aconsejable y por lo tanto el software para prototipar es necesario. Por otra parte la filosofía de desarrollo de este tipo de sistemas es justamente por prototipos y hay prototipadores de software fáciles de usar de un costo muy razonable disponibles en el mercado.

En cuanto a si el desarrollo es responsabilidad de la propia organización o de terceros, ambas opciones son aplicables para software basado en conocimiento o para aplicaciones de software conocidas como tradicionales. Todavía no es tradición que las empresas que se dedican a desarrollar software tengan sistemas basados en conocimiento como un posible producto. Suele entonces la empresa contratar especialistas (ingenieros en conocimiento) para desarrollar el sistema y desde ese punto de vista se responsabiliza del desarrollo.

A.4.4 Análisis de la Sesión IV

- *Conocimientos extraídos:*

Se ha podido evaluar el modelo conceptual de un subconjunto de características, y sus relaciones. Estas relaciones serán tenidas en cuenta en la definición de los modelos y de las inferencias del sistema.

A.4.5 Evaluación de la Sesión IV

- *¿Se han logrado los objetivos?*

Sí se han cumplido los objetivos.

- *¿Es necesario volver sobre lo mismo?*

En el caso del primer prototipo no resulta necesario en este momento. En el desarrollo de prototipos más completos puede ser conveniente presentar otros emparrillados considerando otros subconjuntos de características para cruzar contra la información obtenida.

5.6 CONCLUSIÓN DE LA FASE PRIMARIA DE ADQUISICIÓN DE CONOCIMIENTOS

Tal como se ha definido en el apartado 5.1 del presente capítulo el proceso de adquisición de conocimientos es una tarea que se produce en paralelo a todas las etapas de la construcción del sistema. En cuanto al presente trabajo el proceso de adquisición de conocimientos realizado entre el 20 de octubre de 1999 y el 23 de marzo de 2000 (unas 10 sesiones) permite contar con suficiente información para documentar formalmente la etapa de Conceptualización. El experto continuó prestando su apoyo a lo largo de todo el proyecto. Para las sesiones cumplidas con posterioridad a la fase de Adquisición de Conocimientos se usó la técnica de entrevistas no estructuradas, siguiendo las necesidades de información surgidas del momento.

La segunda etapa de adquisición de conocimientos se realizó entre el 27 de marzo del 2000 y el 12 de abril de 2001 (unas 16 sesiones) para la revisión de:

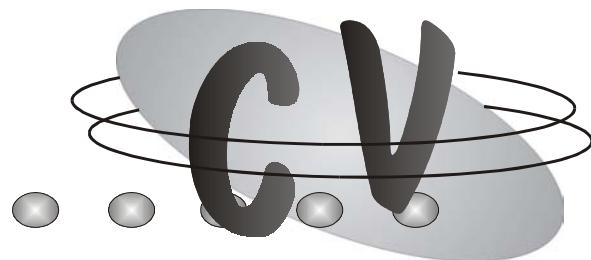
- **A.4.2, Sesión V:** el Glosario de Términos
- **A.4.3, Sesión VI y A.4.4, Sesión VII:** la Tabla Concepto Atributo Valor y Diccionario de Conceptos
- **A.4.5, Sesión VIII:** los atributos Factibilidad / Necesidad / Conveniencia de Metodologías ampliamente probadas- medianamente probadas- poco probadas

- **A.4.6, Sesión IX:** los atributos Necesidad / Conveniencia de mismos Procedimientos de Desarrollo y Mantenimiento. Primeros casos de prueba.
- **A.4.7, Sesión X:** comprobación de los conocimientos estratégicos (modelo de descomposición funcional)
- **A.4.8, Sesión XI:** comprobación de los conocimientos tácticos (seudo reglas)
- **A.4.9, Sesión XII:** comprobación de los conocimientos fácticos
- **A.4.10, Sesión XIII:** el modelo dinámico. Casos de prueba.
- **A.4.11, Sesión XIV:** el modelo estático
- **A.4.12, Sesión XV:** completar casos de prueba.
- **A.4.13, Sesión XVI:** comprobación del mapa de conocimientos
- **A.4.14, Sesión XVII:** completar casos de prueba
- **A.4.15, Sesión XVIII:** evaluación del funcionamiento del sistema
- **A.4.16, Sesión XIX:** evaluación del funcionamiento del sistema
- **A.4.17, Sesión XX:** revisar conclusiones y futuras líneas de investigación

La documentación de estas sesiones está formalizada en los capítulos siguientes de Conceptualización, Formalización , Implementación y Evaluación del sistema experto.

Capítulo 6

Conceptualización de Conocimientos



En este capítulo se presenta el resultado del desarrollo de la fase de Conceptualización. En primer lugar se documenta el resultado del Análisis de los conocimientos adquiridos organizados en Conocimientos Estratégicos, Tácticos y Fácticos. Luego se muestra el resultado de la Síntesis de los conocimientos adquiridos y analizados documentado en el Modelo estático, el Modelo dinámico y la conformación final en el Mapa de Conocimientos. Como cierre del capítulo se presenta la Comprobación del proceso de Conceptualización.

6.1 PROCESO DE CONCEPTUALIZACION

En el capítulo anterior, Adquisición de conocimientos, se describió y documentó el proceso de obtener información de los expertos. El resultado de ese proceso es un complejo paquete de conocimientos sobre el dominio de la aplicación, que son el punto de partida para el desarrollo del sistema. ¡¿Cómo pasar entonces de este torbellino de información a un sistema automatizado?! Como en cualquier desarrollo informático, es necesario definir en primera instancia modelos conceptuales que luego se traducen en modelos formales y finalmente se inicia y completa el proceso de implementación. Primer paso entonces y objetivo de este capítulo es definir el modelo conceptual de este proyecto.

Según Gómez et al. [Gómez, A. y otros 1997], la Conceptualización conlleva un proceso de estructuración de los conocimientos adquiridos que se representa en la figura 6-1. Este proceso tiene dos etapas, una primera de Análisis y un trabajo de Síntesis. La etapa de Análisis tiene como objetivo organizar los conocimientos en estratégicos, fácticos y tácticos. La etapa de Síntesis tiene como objetivo armar los modelos dinámico y estático en los cuales los conocimientos forman parte en mayor o menor medida (representado por línea llena y por línea punteada respectivamente) y esos modelos se integran en el Mapa de Conocimientos conformando así el Modelo Conceptual del sistema. El modelo estático y el modelo dinámico integrados en forma de mapa de conocimientos modelan el comportamiento del experto en la solución de los problemas. Estos dos modelos se corresponden respectivamente con la estructura y la funcionalidad conceptual del sistema.

El proceso de Conceptualización no es secuencial ni respecto de la fase de Adquisición de conocimientos, ni en sí mismo. Mientras el proceso de Conceptualización se está cumpliendo, continúa el proceso de adquisición con el experto que va revisando y refinando la organización de los conocimientos. De la misma forma el proceso de síntesis avanza junto con el proceso de análisis

intercalándose las actividades propias de una y otra etapa y retroalimentándose para su refinamiento.

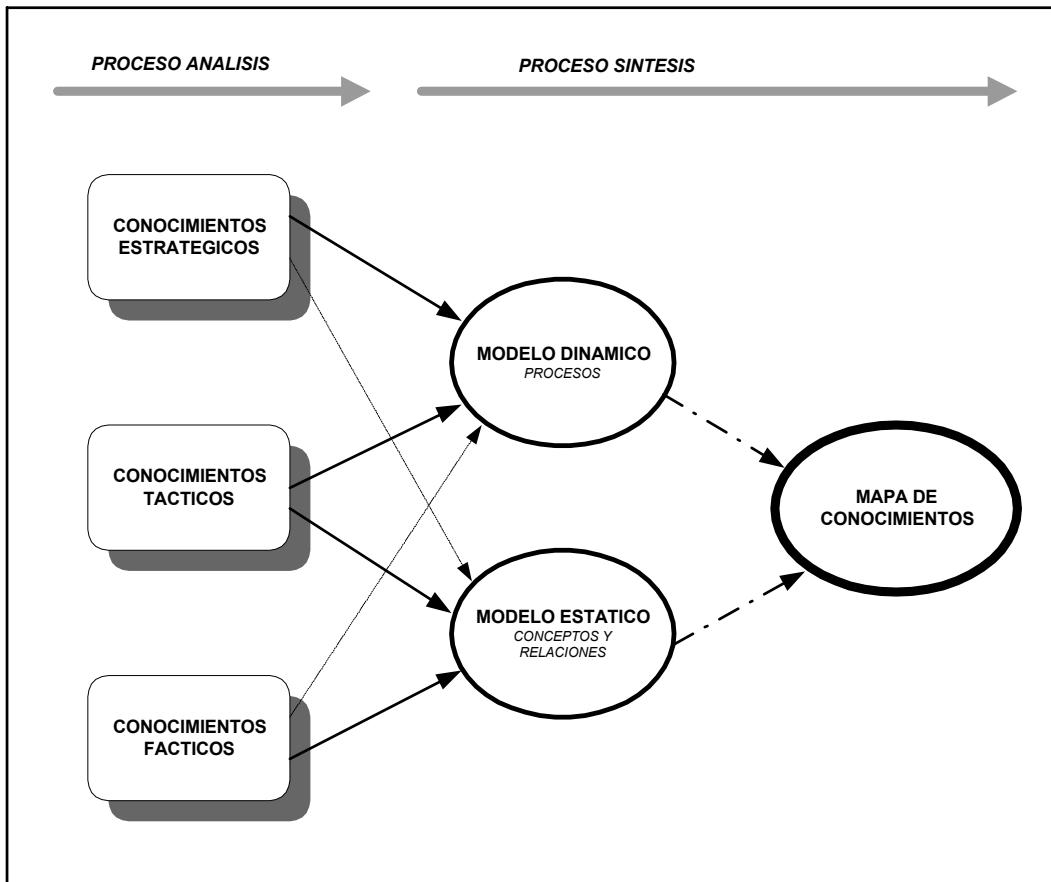


FIGURA 6-1 Proceso de Conceptualización

6.2 ANALISIS DE CONOCIMIENTOS

Se identifican y ordenan los conocimientos según la siguiente clasificación:

- **Estratégicos o de control:** especifican qué hacer, dónde y porqué hacerlo. Los conocimientos estratégicos fijan la secuencia de pasos que el SE debe seguir para ejecutar la tarea.
- **Tácticos, de acción u operativos:** especifican cómo y cuándo el SE puede añadir a sus conocimientos genéricos información actual acerca del caso.

- **Fácticos o declarativos:** especifican lo que es, o se cree que es verdad acerca del mundo en general y acerca del caso particular para el cual se está ejecutando la tarea.

En el presente trabajo se documenta la etapa de análisis de conocimientos en 5 pasos que usualmente se cumplen para completar el proceso de análisis. Se ha decidido presentarlos en esta forma porque facilitan la comprensión de la documentación ya que representa el proceso que realizan el ingeniero de conocimientos y el experto conjuntamente.

En cada paso se trabaja en mayor medida con alguno de los tipos de conocimientos:

- Estratégicos se identifican particularmente en el paso 3.
- Tácticos se identifican particularmente en el paso 4.
- Fácticos se identifican en los pasos 1, 2 y 5 .

6.2.1 PASO 1: IDENTIFICACION, COMPARACIÓN Y CATEGORIZACIÓN DE CONCEPTOS

El primer paso del proceso de Conceptualización es identificar los conceptos y registrar los atributos y valores asociados. En este primer paso se trabaja fundamentalmente con conocimientos fácticos y se describen ordenadamente completando:

- **Glosario de términos:** en la *Tabla 6-1* (pág. 152 a 155) se detallan el significado de los términos que usa el experto en la resolución de la tarea. La definición de estos términos se ha ido completando a lo largo de varias sesiones de conocimiento y el confeccionarlo ha contribuido a un mejor entendimiento del dominio del problema y a refinar el modelo conceptual.
- **Diccionario de conceptos:** en la *Tabla 6-2* (pág. 156 a 158) se identifican los conceptos funcionales del más alto nivel, especificando su utilidad o función, sinónimos, acrónimos, los atributos que lo definen y de donde pueden derivarse los datos.
- **Tabla de Conceptos-Atributos-Valores:** en la *Tabla 6-3* (pág. 159 a 165) se registran los atributos propios de cada concepto. Se entiende por atributo las propiedades o características de un concepto que se necesita conocer para modelar la tarea del experto.

TABLA 6-1: GLOSARIO DE TERMINOS	
TERMINO	DESCRIPCION
Algorítmico	Conjunto de operaciones que siguen un proceso predefinido para la solución de un problema. Principio opuesto a la heurística.
Ampliación	Se refiere a cuando a partir de nuevos requerimientos es necesario agregar funciones o procesos al sistema que está definido e implementado. Se agregan funciones a las ya existentes, aumenta el número de funciones.
Aplicación	Sistema o parte de un sistema que lleva a cabo un conjunto de actividades o tareas determinadas. En sentido más estricto conjunto de tareas desarrolladas por un sistema informático. Se usan como sinónimos Sistema y Proyecto.
Batch	Procesamiento diferido. Modalidad de procesamiento informático con carga y control de información por lotes y ejecución normalmente diferida
Cliente	Ver Usuario.
Comienzo del proyecto	Conjunto de actividades iniciales de un proyecto donde se establecen los objetivos, límites, alcances, información a brindar y alguna particularidad específica de la aplicación a desarrollar.
Comportamiento dinámico	Forma de procesamiento donde el ordenador recibe y/o procesa la información en el momento "inmediatamente" (sin diferir). Usualmente el comportamiento dinámico es interactivo
Comportamiento interactivo	Forma de procesamiento en que el usuario dialoga con un ordenador. Usualmente el comportamiento interactivo es dinámico.
Comprensión	Proceso de análisis del dominio de conocimiento de las partes componentes de un sistema
Emisión gradual	Es la nueva edición de una aplicación con modificaciones notables respecto de emisiones anteriores. También se conoce como una nueva versión de la aplicación. Abarca la aplicación en su totalidad.
Entorno del sistema	Se refiere a los sistemas o elementos externos a la aplicación que se vinculan con ella recibiendo o enviando información. Ver Límites del sistema.
Es conveniente	Se dice que algo es conveniente cuando es necesario y factible
Es factible	Se dice que algo es factible cuando es una alternativa posible de aplicar o seleccionar, considerando los aspectos políticos y de recursos
Es necesario	Se dice que algo es necesario cuando existen razones que justifican, condicionan su aplicación o selección considerando los recursos disponibles y los aspectos políticos
Extensión	Se refiere a cuando es necesario dar mayor funcionalidad a las funciones o procesos existentes

TABLA 6-1: GLOSARIO DE TERMINOS

TERMINO	DESCRIPCION
	definidos e implementados en un sistema. Se extienden las funciones ya existentes, no aumentan las funciones sino algún aspecto de alguna de ellas
Límites del sistema	Se usa como sinónimo de los alcances del sistema o aplicación. Se refiere genéricamente a los procesos que involucra, y particularmente con que otros sistemas (entidades externas) se relaciona recibiendo o enviando información y que información es la que intercambia con esas entidades.
Manipulación	Proceso de diseño de las partes componentes de un sistema o aplicación.
Pocas opciones de diseño	No mas de tres posibilidades de modelos formales o de diseño a partir de un modelo de análisis
Pocas opciones de implementación	No mas de dos posibilidades de modelos implementables computacionalmente.
Problemas de Arquitectura	Incumplimiento de Restricciones que se deben tener en cuenta al diseñar y configurar los componentes de un sistema. Se aplica al diseño de sistemas informáticos procesadores, aplicaciones de software y redes.
Problemas de Eficacia	Incumplimiento de Restricciones en el cumplimiento de los objetivos vinculadas a la arquitectura de un sistema.
Progresión no uniforme y secuencial de las fases	Las actividades que es necesario realizar durante el desarrollo pueden cumplirse de tal modo que cada actividad cumplida puede determinar retroceder a una etapa anterior o continuar con alguna otra sin mantener un ordenamiento. El grado de avance puede ser desparejo y con diferentes ritmos.
Progresión uniforme y secuencial de las fases	Las actividades que es necesario realizar durante el desarrollo pueden cumplirse en una sucesión ordenada de tal modo que cada actividad cumplida determina la siguiente con un avance gradual y parejo (sin grandes distorsiones)
Proyecto	Conjunto integrado de planes, tareas, actividades y operaciones necesarias para planificar y desarrollar un sistema o una aplicación. Ver Aplicación
Redefinición	Se refiere a cuando se cambian los procedimientos de una o varias funciones principales ya definidas e implementadas en un sistema
Requerimientos	Es la especificación de que información debe brindar el sistema o aplicación informática. Se usa como sinónimo de Requisitos. Ver Requisitos.
Requisitos	Es la especificación de cómo se desea que brinda la información el sistema o aplicación informática. Es un conjunto de condiciones que ha de cumplir un sistema de información, restricción que se impone a su desarrollo o a su funcionamiento. Se usa como sinónimo de Requerimientos. Ver Requerimientos

TABLA 6-1: GLOSARIO DE TERMINOS	
TERMINO	DESCRIPCION
Requisitos definidos desestructuradamente	Que se han definido las restricciones que debe cumplir el sistema a desarrollar en forma desordenada y despareja. Que no se ha observado alguna metodología que de estructura similar a la documentación de los requisitos.
Requisitos definidos exhaustivamente	Que se han definido en detalle y sin omisiones, las restricciones que debe cumplir el sistema a desarrollar.
Requisitos definidos formalmente	Que se han definido las restricciones que debe cumplir el sistema a desarrollar en forma expresa (en algún documento o informe) y con precisión. Que se ha observado alguna metodología de documentación.
Requisitos definidos Incompletos	Que se han definido genéricamente y con posibilidad de omisiones, las restricciones que debe cumplir el sistema a desarrollar.
Requisitos definidos informalmente	Que se han definido las restricciones que debe cumplir el sistema a desarrollar en forma imprecisa y que no están expresamente documentados. Que no se ha observado alguna metodología de documentación.
Requisitos definidos mayoritariamente	Cuando el usuario manifiesta que se encuentran definidos el 90% o mas de los requisitos del sistema y que el 10 % pueden ser detalles que no modifican la estructura de las restricciones ya definidas, ni de la aplicación a desarrollar.
Requisitos definidos parcialmente	Cuando el usuario manifiesta que se encuentran definidos menos del 90% de los requisitos del sistema y/o que los restantes pueden modificar la estructura de las restricciones ya definidas y de la aplicación a desarrollar.
Requisitos definidos uniformemente	Que se han definido las restricciones que debe cumplir el sistema a desarrollar en forma ordenada, pareja y semejante. Que se ha observado alguna metodología que ordena, facilita y da estructura similar a la documentación de los requisitos.
Sistema	Forma abreviada de referirse a Sistema de Información. Ver Aplicación
Sistema de Información	Conjunto integrado de las personas, procedimientos, medios materiales y otros recursos destinado a la captura, administración, proceso y distribución de información en el ámbito de una organización.
Software de base	Conjunto integrado de programas que se encargan de la gestión de algún servicio básico para el usuario del ordenador, por ejemplo, procesador de textos, planilla de cálculo, graficador, etc.
Subsistemas	Componentes o partes distinguibles de una aplicación.
Tiempo Real	Forma de procesamiento interactivo en que un ordenador ejecuta las instrucciones y procesa los datos, teniendo como restricción un limitado y ajustado margen de tiempo para producir las respuestas.

TABLA 6-1: GLOSARIO DE TERMINOS	
TERMINO	DESCRIPCION
Usuario	Persona que usa el sistema informático. Persona que contrata el desarrollo del sistema. Suele usarse Cliente como sinónimo.
Varias opciones de diseño	Cuatro o más posibilidades de modelos formales o de diseño a partir de un modelo de análisis
Varias opciones de implementación	Tres o más posibilidades de modelos formales implementables computacionalmente.

TABLA 6-2: DICCIONARIO DE CONCEPTOS

CONCEPTO	FUNCION	SINÓNIMOS/ ACRONIMOS	ATRIBUTOS	DERIVADO DE
Proyecto	Proyecto para el que se selecciona el CV. Registro para identificar cada uno de los casos en que se usa el SE, con fines estadísticos posteriores.	Aplicación /Sistema	Identificación Nombre del proyecto Líder del proyecto Objetivo Fecha inicio Fecha finalización CV propuesto por SE CV seleccionado	La carpeta del proyecto, en particular del Informe de requerimientos.
Requisitos	Registro de la estimación del líder de proyecto respecto de las características de la especificación de requisitos para el proyecto a desarrollar.	Requerimientos	Tipo de definición Grado de certidumbre Grado de cumplimiento Definición requisitos Definición límites	Estimación del Informe de requerimientos del proyecto y de las primeras entrevistas con el usuario.
Aplicación	Registro de la estimación del líder de proyecto respecto de las características de la aplicación (software) a desarrollar	Sistema	Niveles de Composición Complejidad Subsistemas Componentes Predominantes Comportamiento Predominante Orientación Progresión Fases desarrollo Relación Fases desarrollo Retroalimentación Fases desarrollo Integración HW-SW Factores diseño Existencia Aplicaciones Posibilidad diseño Posibilidad Implementación Modelado Prototipo Modelado Objetos Variabilidad Procesos Opcionalidad Tipo modificación	Estimación resultante de los requerimientos del proyecto y de las primeras entrevistas con el usuario.
Coordinación	Especifica los aspectos de gestión: políticos, técnicos, metodológicos	Gestión	Formalidad Entregas	Estimación resultante de los requerimientos del proyecto y

TABLA 6-2: DICCIONARIO DE CONCEPTOS

CONCEPTO	FUNCION	SINÓNIMOS/ ACRONIMOS	ATRIBUTOS	DERIVADO DE
	que influyen en la selección del CV.		Responsabilidad Control gestión Software para prototipar Factibilidad metodología Necesidad metodología Reúso aplicación existente Reúso aplicación OO Reúso aplicación futura Procedimientos cambios Procedimientos Desarrollo-Mantenimiento Conveniencia Metodología Factibilidad Prototipo Sistema OO Entrega Intermedia Aplicabilidad Espiral Aplicabilidad Prototipo Aplicabilidad OO Procedimientos D-M	de las primeras entrevistas con el usuario.
Usuario	Especifica el tipo de relación, participación, del usuario en el proyecto a desarrollar.	Cliente / operador	Participación Introducción gradual	
Equipo de proyecto	Especifica las características de los recursos humanos que participan en el desarrollo de la aplicación.		Experiencia técnicas de IS Experiencia previa	Estimación resultante de las aptitudes de los recursos humanos asignados, o de los recursos existentes, para el desarrollo de la aplicación.
Riesgos del proyecto	Especifica la estimación de varios ítems a tener en cuenta respecto de los posibles riesgos del proyecto que afectan en la		Viabilidad software Nivel de riesgo Análisis riesgo Técnicas AR	Estimación resultante de los requerimientos del proyecto y de las primeras entrevistas con el usuario

TABLA 6-2: DICCIONARIO DE CONCEPTOS

CONCEPTO	FUNCION	SINÓNIMOS/ ACRONIMOS	ATRIBUTOS	DERIVADO DE
	selección del CV.		Identificación Alternativas Riesgos Alternativa Categorías riesgo Objetivo calidad Terminación Proyectos Riesgo Cascada Riesgo Objetos Habilidad Riesgo Factor Riesgo Riesgo Espiral Riesgo	
Área Requisitos	Especifica el ciclo de vida propuesto de acuerdo a los factores estimados en Requisitos.		CV Propuesto Requisitos	Los valores estimados por el usuario del SE para los atributos de Requisitos.
Área Aplicación	Especifica el ciclo de vida propuesto de acuerdo a los factores estimados en Aplicación		CV Propuesto Aplicación	Los valores estimados por el usuario del SE para los atributos de Aplicación
Área Gestión Proyecto	Especifica el ciclo de vida propuesto de acuerdo a los factores estimados en Coordinación, Usuario, Equipo de Proyecto y Riesgos del Proyecto		CV Propuesto Gestión	Los valores estimados por el usuario del SE para los atributos de Coordinación, Usuario, Equipo de proyecto, Riesgos del proyecto.
CV Diagnóstico	Especifica el ciclo de vida propuesto de acuerdo a los ciclos de vida estimados en Área Requisitos, Área Aplicación, Área Gestión Proyecto		CV Propuesto del Proyecto	Los valores derivados de las áreas de Requisitos, aplicación y Gestión Proyecto.

TABLA 6-3: TABLA CONCEPTOS-ATRIBUTOS-VALORES

CONCEPTO	ATRIBUTO	VALOR
<i>Proyecto</i>	Identificación	<ul style="list-style-type: none"> • Código alfanumérico
	Nombre del proyecto	<ul style="list-style-type: none"> • Descripción (texto)
	Líder del proyecto	<ul style="list-style-type: none"> • Nombre y Apellido
	Objetivo	<ul style="list-style-type: none"> • Descripción (texto libre)
	Fecha inicio	<ul style="list-style-type: none"> • DD / MM /AAAA
	Fecha finalización	<ul style="list-style-type: none"> • DD / MM /AAAA
	CV propuesto por SE	<ul style="list-style-type: none"> • Cascada • Espiral • Objetos
	CV seleccionado	<ul style="list-style-type: none"> • Cascada • Espiral • Objetos • Otro

TABLA 6-3: TABLA CONCEPTOS-ATRIBUTOS-VALORES

CONCEPTO	ATRIBUTO	VALOR
<i>Requisitos</i> <i>Requerimientos</i>	Tipo de Definición	<ul style="list-style-type: none"> • Formalmente • Informalmente • Exhaustivamente • Incompleto • Uniformemente • Desestructuradamente
	Definición Requisitos (al inicio del proyecto)	<ul style="list-style-type: none"> • Definición Clara • Definición Incierta • Alta incertidumbre • Baja incertidumbre
	Definición Límites (al inicio del proyecto)	<ul style="list-style-type: none"> • Definición Clara • Definición Incierta • Alta incertidumbre • Baja incertidumbre
	Grado de Certidumbre al inicio del proyecto	<ul style="list-style-type: none"> • Mayoritariamente definidos ($> = 90\%$) • Parcialmente definidos ($< 90\%$)
	Grado de Cumplimiento en Producto final	<ul style="list-style-type: none"> • Usuario muy exigente • Usuario poco exigente
<i>Aplicación</i> <i>Sistema</i>	Componentes Predominantes	<ul style="list-style-type: none"> • Algorítmicos • Matemáticos • Gráficos
	Complejidad Subsistemas	<ul style="list-style-type: none"> • Baja • Alta
	Comportamiento Predominante	<ul style="list-style-type: none"> • Batch • Dinámico • Interactivo • Tiempo Real
	Orientación	<ul style="list-style-type: none"> • Software de Base • Sistema Basado en Conocimiento • Otros
	Niveles de Composición	<ul style="list-style-type: none"> • Pocos Subsistemas • Varios Subsistemas

TABLA 6-3: TABLA CONCEPTOS-ATRIBUTOS-VALORES

CONCEPTO	ATRIBUTO	VALOR
	Progresión Fases desarrollo	<ul style="list-style-type: none"> • Uniforme y Secuencial • No Uniforme y Secuencial
	Relación Fases desarrollo	<ul style="list-style-type: none"> • Poca Dependencia • Mucha Dependencia • Independencia
	Retroalimentación Fases desarrollo	<ul style="list-style-type: none"> • Baja • Alta
	Integración HW-SW	<ul style="list-style-type: none"> • Fuertemente • Medianamente • Levemente
	Factores Diseño (a considerar en etapas tempranas del desarrollo)	<ul style="list-style-type: none"> • Problemas de Arquitectura • Problemas de eficacia
	Existencia Aplicaciones (similares en el mercado)	<ul style="list-style-type: none"> • Innovador • Conocido
	Posibilidad Diseño	<ul style="list-style-type: none"> • Pocas opciones • Varias opciones
	Posibilidad Implementación	<ul style="list-style-type: none"> • Pocas opciones • Varias opciones
	Modelado Prototipo	<ul style="list-style-type: none"> • Conveniente • No conveniente
	Modelado Objetos	<ul style="list-style-type: none"> • Conveniente • No conveniente
	Variabilidad Procesos	<ul style="list-style-type: none"> • Alta • Baja
	Tipo modificación (explícitamente previstas en datos y procesos)	<ul style="list-style-type: none"> • Redefinición • Extensión • Ampliación • Emisión gradual
	Opcionalidad	<ul style="list-style-type: none"> • Múltiple • Simple
<i>Coordinación</i>	Formalidad	<ul style="list-style-type: none"> • Poco formal • Medianamente formal

TABLA 6-3: TABLA CONCEPTOS-ATRIBUTOS-VALORES

CONCEPTO	ATRIBUTO	VALOR
	Entregas	<ul style="list-style-type: none"> • Muy formal • Versión temprana • Versión completa • Versión gradual • Versión parcial
	Responsabilidad	<ul style="list-style-type: none"> • Terceros • Organización propia
	Control gestión	<ul style="list-style-type: none"> • Muy ajustado • Medianamente ajustado • Poco ajustado
	Software para prototipar	<ul style="list-style-type: none"> • No disponible • Hay disponible • Factible de adquirir • No adquirible
	Factibilidad Metodología	<ul style="list-style-type: none"> • Ampliamente probada • Medianamente probada • Poco probada
	Necesidad Metodología	<ul style="list-style-type: none"> • Ampliamente probada • Medianamente probada • Poco probada
	Reúso aplicación existente	<ul style="list-style-type: none"> • Estrategias explícitas • No estrategias explícitas
	Reúso aplicación futura	<ul style="list-style-type: none"> • Muy necesario • Medianamente necesario • Poco necesario
	Reúso aplicación OO	<ul style="list-style-type: none"> • Subsistema del existente • Ampliación del existente • Modificación del existente • Componentes del actual
	Procedimientos Cambios	<ul style="list-style-type: none"> • Acuerdos confirmados • No Acuerdos confirmados

TABLA 6-3: TABLA CONCEPTOS-ATRIBUTOS-VALORES		
CONCEPTO	ATRIBUTO	VALOR
	Procedimientos Desarrollo-Mantenimiento	<ul style="list-style-type: none"> • Es factible usar los mismos • Es factible usar diferentes • Es necesario usar los mismos • Es necesario usar diferentes
	Conveniencia Metodología	<ul style="list-style-type: none"> • Ampliamente probada • Medianamente probada • Poco probada
	Factibilidad Prototipo	<ul style="list-style-type: none"> • Factible • No factible
	Sistema OO	<ul style="list-style-type: none"> • Existe • No existe
	Entrega Intermedia	<ul style="list-style-type: none"> • Existe • No existe
	Aplicabilidad Espiral	<ul style="list-style-type: none"> • Aplicable • No aplicable
	Aplicabilidad Prototipo	<ul style="list-style-type: none"> • Aplicable • No aplicable
	Aplicabilidad OO	<ul style="list-style-type: none"> • Aplicable • No aplicable
	Procedimientos D-M	<ul style="list-style-type: none"> • Conveniente • No conveniente
<i>Usuario / cliente</i>	Participación	<ul style="list-style-type: none"> • Fuerte • Regular • Poca
	Introducción gradual	<ul style="list-style-type: none"> • Necesaria • No necesaria
Equipo de proyecto	Experiencia técnicas IS	<ul style="list-style-type: none"> • Hay • No hay
	Experiencia previa	<ul style="list-style-type: none"> • Hay • No hay

TABLA 6-3: TABLA CONCEPTOS-ATRIBUTOS-VALORES

CONCEPTO	ATRIBUTO	VALOR
<i>Riesgos del Proyecto</i>	Viabilidad software	<ul style="list-style-type: none"> • Hay certeza • Hay dudas
	Nivel de riesgo	<ul style="list-style-type: none"> • Alto riesgo • Mediano riesgo • Bajo riesgo • No hay riesgo
	Análisis riesgo	<ul style="list-style-type: none"> • Estrategias explícitas • No estrategias explícitas
	Técnicas AR	<ul style="list-style-type: none"> • Se dispone • No se dispone
	Identificación Alternativas	<ul style="list-style-type: none"> • Se identifican • No se identifican • Etapas anteriores
	Riesgos Alternativa	<ul style="list-style-type: none"> • Se identifican • No se identifican • Etapas anteriores
	Categorías riesgo	<ul style="list-style-type: none"> • Técnicos • Otros riesgos
	Objetivo calidad	<ul style="list-style-type: none"> • Mecanismos explícitos • No mecanismos explícitos
	Terminación Proyectos	<ul style="list-style-type: none"> • Prematura • En término.
	Riesgo Cascada	<ul style="list-style-type: none"> • Aceptable • No aceptable
	Riesgo Objetos	<ul style="list-style-type: none"> • Aceptable • No aceptable
	Habilidad Riesgo	<ul style="list-style-type: none"> • Existe • No existe
	Factor Riesgo	<ul style="list-style-type: none"> • Existe • No existe
	Riesgo Espiral	<ul style="list-style-type: none"> • Aceptable • No aceptable

TABLA 6-3: TABLA CONCEPTOS-ATRIBUTOS-VALORES		
CONCEPTO	ATRIBUTO	VALOR
	Riesgo	<ul style="list-style-type: none"> • Evaluable • No evaluabile
<i>Area Requisitos</i>	CV Propuesto Requisitos	<ul style="list-style-type: none"> • Cascada • Objetos <ul style="list-style-type: none"> • Espiral • No hay propuesta
<i>Area Aplicación</i>	CV Propuesto Aplicación	<ul style="list-style-type: none"> • Cascada • Objetos <ul style="list-style-type: none"> • Espiral • No hay propuesta
<i>Area Gestión proyecto</i>	CV Propuesto Gestión	<ul style="list-style-type: none"> • Cascada • Objetos <ul style="list-style-type: none"> • Espiral • No hay propuesta
<i>CV Diagnóstico</i>	CV Propuesto Proyecto	<ul style="list-style-type: none"> • Cascada • Objetos <ul style="list-style-type: none"> • Espiral • No hay propuesta

6.2.2 PASO 2: IDENTIFICACIÓN DE LAS RELACIONES ENTRE CONCEPTOS

El segundo paso del proceso de Conceptualización es identificar las relaciones entre los conceptos. En este paso se trabaja con conocimientos fácticos, se representa gráficamente el modelo mental que el experto tiene del aspecto estático del problema, usando el modelo entidad-relación de Chen [Chen, P.S. 1976]. En la figura 6-2 se detalla el modelo relacional del presente trabajo.

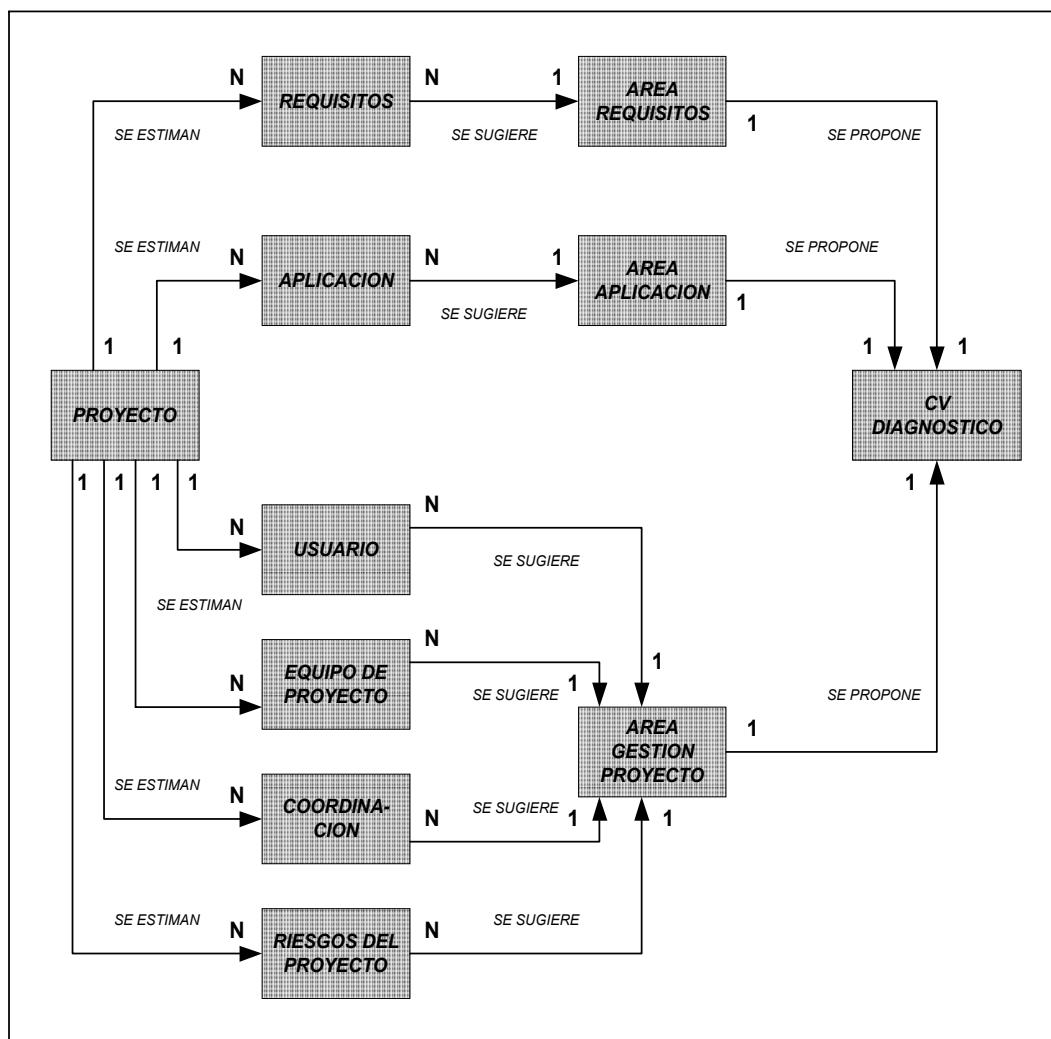


FIGURA 6-2: Modelo Relacional

6.2.3 PASO 3: IDENTIFICACIÓN DE LOS CONOCIMIENTOS ESTRATEGICOS

Una vez identificados los conceptos, sus atributos y sus relaciones el tercer paso del proceso de Conceptualización consiste en identificar las funciones del proceso de resolución del experto. Estos conocimientos son de tipo estratégico.

En primer lugar se presenta una representación gráfica en forma de árbol (figura 6-3) que ilustra los pasos modulares que completan la tarea del experto y el flujo de control del proceso de resolución. Los módulos se han identificado considerando su cohesión funcional y su independencia.

En segundo lugar se describen los módulos de acuerdo a la propuesta de Gómez [Gómez, A. y otros 1997]:

- Pasos de Alto Nivel:
Se corresponden con el nivel 1 y 2 de la figura 6-3: Árbol de Descomposición funcional del Problema.
- Subpasos de la Tarea:
Se corresponden con el nivel 3 de la figura 6-3: Árbol de Descomposición funcional del Problema.
- Subpasos de Bajo Nivel:
Se corresponden con el nivel 4 de la figura 6-3: Árbol de Descomposición funcional del Problema.

6.2.3.1 DESCOMPOSICION FUNCIONAL

Módulo 1. Selección del CV del proyecto

A partir de la decisión de construir un sistema de información una de las primeras actividades consiste en definir los requerimientos del sistema. Una vez definidos es necesario seleccionar el ciclo de vida guía del proceso de desarrollo.

En el marco del presente trabajo y siguiendo la propuesta del equipo de expertos, a partir del análisis de los textos, se han identificado tres grandes áreas de análisis respecto del proyecto software. Estas áreas: Especificación de Requerimientos, Tipo de Aplicación y Gestión de Proyecto no son exhaustivas y la identificación de módulos independientes facilita la incorporación de otras áreas en futuros prototipos.

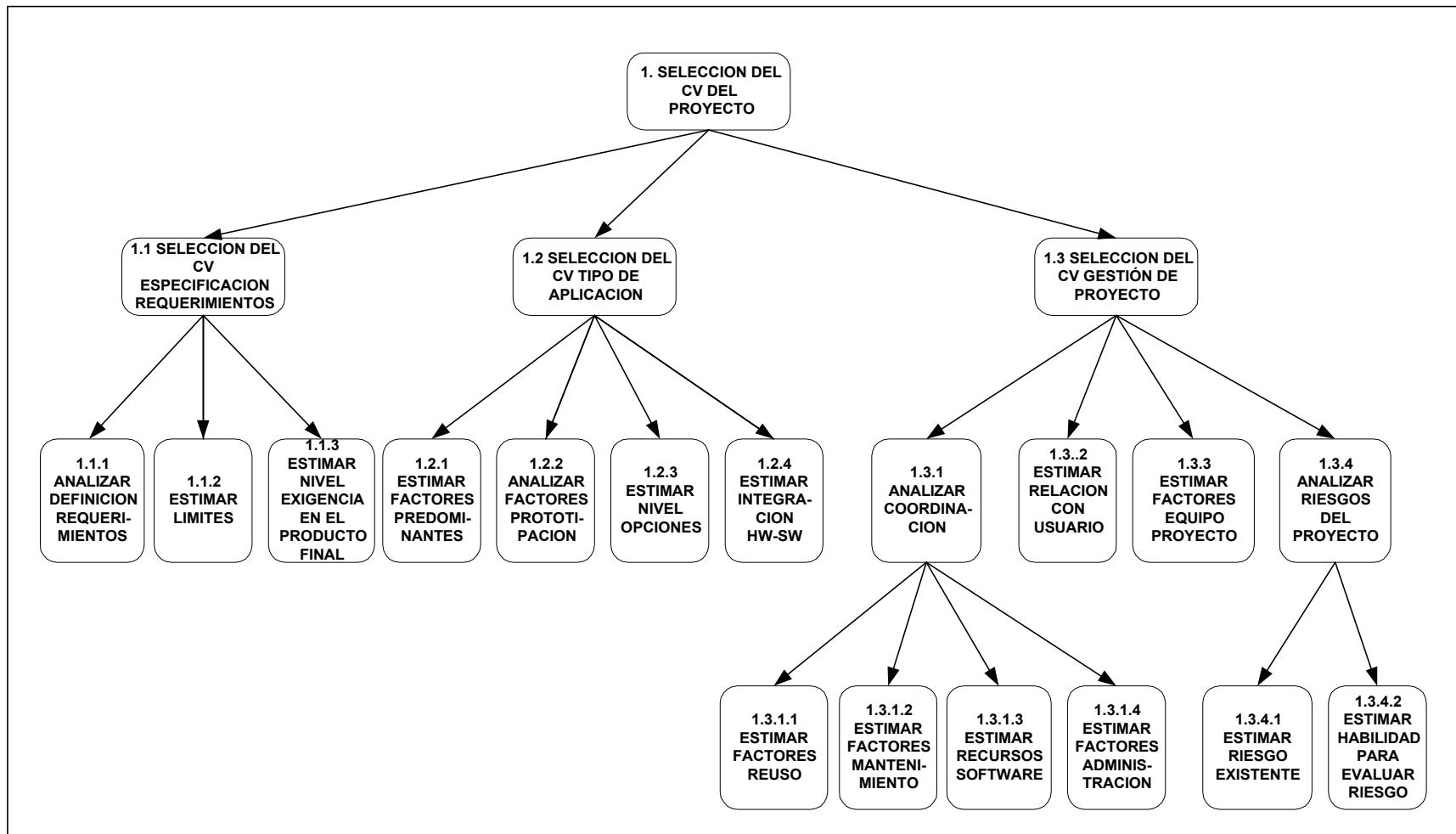


FIGURA 6-3 Arbol de Descomposición Funcional del Problema

Propósito:

Determinar el ciclo de vida propuesto para el desarrollo del proyecto.

Para llevar a cabo la tarea, el sistema debe realizar las siguientes funciones de alto nivel:

Módulo 1.1 Selección del CV Especificación de Requerimientos

Módulo 1.2 Selección del CV Tipo de Aplicación

Módulo 1.3 Selección del CV Gestión de Proyecto

Entrada- Origen de la Entrada:

Los resultados parciales obtenidos para cada uno de los subpasos.

Razonamiento:

Considerando el/los ciclos de vida posibles para cada área se determina el ciclo de vida más adecuado para el proyecto.

Salida- Destino de la Salida:

El nombre del ciclo de vida seleccionado para el proyecto. Su destino es la Base de Conocimientos y edición en pantalla de resultados.

Módulo 1.1 Selección del CV Especificación de Requerimientos

Se analizan algunos aspectos de la Especificación de Requerimientos para proponer el o los ciclos de vida posibles de acuerdo a esta área. Los aspectos analizados están encuadrados dentro del marco del presente trabajo y no son exhaustivos. La identificación en módulos independientes prevé la incorporación de otros aspectos a analizar en esta área en futuros prototipos.

Propósito:

Determinar el o los ciclos de vida posibles para el área de Especificación de Requerimientos.

Para llevar a cabo la tarea el sistema debe realizar las siguientes subtareas:

Módulo 1.1.1 Analizar Definición de Requerimientos

Módulo 1.1.2 Estimar Límites

Módulo 1.1.3 Estimar Nivel de exigencia en el producto final

Entrada- Origen de la Entrada:

Los valores estimados por el usuario del sistema experto, para los atributos vinculados al concepto Requisitos.

Razonamiento:

Si los requisitos y los límites están definidos claramente y explicitados al comienzo del proyecto se recomienda usar ciclo de vida en Cascada.

Si los requisitos y los límites están definidos con alta incertidumbre y no están explicitados al comienzo del proyecto es recomendable usar ciclo de vida en Orientado a objetos o en Espiral.

Si la exigencia del usuario respecto del cumplimiento de los requerimientos en el producto final es muy alta es recomendable usar ciclo de vida Orientado a objetos o en Espiral.

Salida- Destino de la Salida:

El nombre del ciclo de vida seleccionado para el área. Su destino es la Base de Conocimientos y edición en pantalla de resultados.

Módulo 1.2 Selección del CV Tipo de Aplicación

Se analizan algunos aspectos de la Aplicación a desarrollar para proponer el o los ciclos de vida posibles de acuerdo a esta área. Los aspectos analizados están encuadrados dentro del marco del presente trabajo y no son exhaustivos. La identificación en módulos independientes prevé la incorporación de otros aspectos a analizar en esta área en futuros prototipos.

Propósito:

Determinar el o los ciclos de vida posibles para el área de Tipo de Aplicación.

Para llevar a cabo la tarea el sistema debe realizar las siguientes subtareas:

Módulo 1.2.1 Estimar Factores Predominantes

Módulo 1.2.2 Analizar Factores Prototipación

Módulo 1.2.3 Estimar Nivel de opciones.

Módulo 1.2.4 Estimar Integración de Hardware y Software.

Entrada- Origen de la Entrada:

Los valores estimados por el usuario del sistema experto, para los atributos vinculados al concepto Aplicación.

Razonamiento:

Para recomendar el ciclo de vida en cascada es necesario considerar la complejidad de los subsistemas que lo conforman, la retroalimentación en las fases de desarrollo y los factores predominantes.

Para recomendar el ciclo de vida Orientado a objetos es necesario considerar si las técnicas de modelado de objetos son las más convenientes y si las técnicas de prototipación son convenientes.

Para recomendar el ciclo de vida en Espiral además de considerar si las técnicas de modelado de objetos o de prototipado son convenientes hay que tener en cuenta si el sistema es multiopcional y si se requiere una fuerte integración de hardware y software.

Salida- Destino de la Salida:

El nombre del ciclo de vida seleccionado para el área. Su destino es la Base de Conocimientos y edición en pantalla de resultados.

Módulo 1.3 Selección del CV Gestión de Proyecto

Se analizan algunos aspectos de la Gestión de Proyecto para proponer el o los ciclos de vida posibles de acuerdo a esta área. Los aspectos analizados están encuadrados dentro del marco del presente trabajo y no son exhaustivos. La

identificación en módulos independientes prevé la incorporación de otros aspectos a analizar en esta área en futuros prototipos.

Propósito:

Determinar el o los ciclos de vida posibles para el área de Gestión del Proyecto.

Para llevar a cabo la tarea el sistema debe realizar las siguientes subtareas:

Módulo 1.3.1 Analizar Coordinación

Módulo 1.3.2 Estimar Relación con el Usuario

Módulo 1.3.3 Estimar Factores del Equipo de desarrollo del Proyecto

Módulo 1.3.4 Analizar Riesgos del Proyecto

Entrada- Origen de la Entrada:

Los valores estimados por el usuario del sistema experto, para los atributos vinculados a los conceptos Coordinación, Usuario, Equipo de proyecto y Riesgos del Proyecto.

Razonamiento:

Para recomendar el ciclo de vida en Cascada es necesario analizar:

- Conveniencia de usar metodologías probadas
- Nivel de riesgo
- Si el modelo en cascada es aplicable
- Responsable de la gestión

Para recomendar el ciclo de vida Orientado a objetos es necesario analizar:

- Nivel de riesgo
- Necesidad de entregas intermedias
- Si existe un sistema anterior desarrollado en objetos
- Si el modelo de objetos es aplicable

Para recomendar el ciclo de vida en Espiral es necesario analizar:

- Nivel factor de riesgo evaluable
- Prototipación es aplicable
- Prototipación es factible
- Si el modelo en espiral es aplicable

Salida- Destino de la Salida:

El nombre del ciclo de vida seleccionado para el área. Su destino es la Base de Conocimientos y edición en pantalla de resultados.

Módulo 1.1.1 Analizar Definición Requerimientos

Propósito:

Estimar si los requisitos están claramente definidos al comienzo del proyecto.

Entrada- Origen de la Entrada:

Los valores ingresados por el usuario del sistema experto para los atributos Tipo de definición y Grado de Certidumbre al inicio del proyecto.

Razonamiento:

Si los requisitos se encuentran definidos formalmente, uniformemente y exhaustivamente puede decirse que los requisitos están claramente definidos y es conveniente el modelo cascada.

Si los requisitos se encuentran definidos informalmente, incompletos o desestructuradamente puede decirse que la incertidumbre en la definición de los requisitos es alta, en este caso es conveniente el modelo en espiral u objetos.

Si los requerimientos están mayoritariamente definidos al inicio del proyecto es conveniente el modelo cascada, caso contrario se recomienda alguno de los otros dos modelos.

Salida- Destino de la Salida:

Determinación de la claridad o incertidumbre de los requisitos. Su destino es la Base de Conocimientos.

Módulo 1.1.2 Estimar Límites

Propósito:

Estimar si los límites están claramente definidos al comienzo del proyecto.

Entrada- Origen de la Entrada:

Los valores ingresados por el usuario del sistema experto para los atributos Tipo de Definición y Grado de Certidumbre al inicio del proyecto.

Razonamiento:

Si los requisitos se encuentran definidos formalmente, uniformemente y exhaustivamente puede decirse que los límites están claramente definidos.

Si los requisitos se encuentran definidos informalmente, incompletos o desestructuradamente puede decirse que la incertidumbre en el entorno del sistemas es alta.

Salida- Destino de la Salida:

Determinación de la claridad o incertidumbre de los límites. Su destino es la Base de Conocimientos.

Módulo 1.1.3 Estimar Nivel Exigencia en el Producto Final

Propósito:

Estimar la exigencia del usuario respecto del cumplimiento de los requerimientos en el producto final.

Entrada- Origen de la Entrada:

Los valores ingresados por el usuario del sistema experto para los atributos grado de cumplimiento en producto final.

Razonamiento:

Si la exigencia del usuario es mucha es recomendable el ciclo de vida en espiral u orientado a objetos, pero no se recomienda el modelo en cascada porque su estructura no permite prototipación o entregas intermedias.

Salida- Destino de la Salida:

Valor ingresado por el usuario con destino a la Base de Conocimientos.

Módulo 1.2.1 Estimar Factores Predominantes*Propósito:*

Determinar el tipo de técnicas de modelado (objetos, prototipación o cascada) que mejor se adecuan a los factores predominantes de la aplicación a desarrollar.

Entrada- Origen de la Entrada:

Los valores ingresados por el usuario del sistema experto para los atributos Componentes Predominantes, Comportamiento Predominante y Orientación.

Razonamiento:

Las técnicas de modelado en cascada son adecuadas si los factores predominantes son:

- Fuerte contenido algorítmico
- Comportamiento batch

Las técnicas de modelado de objetos son adecuadas si los factores predominantes son:

- Fuerte contenido matemático
- Fuerte contenido gráfico
- Subsistemas con comportamiento dinámico
- Comportamiento fuertemente interactivo
- Comportamiento en tiempo real
- Desarrollo de software de base

Las técnicas de modelado por prototipo son adecuadas si los factores predominantes son:

- Sistema Basado en conocimiento

Salida- Destino de la Salida:

Valores ingresados por el usuario con destino a la Base de Conocimientos.

Módulo 1.2.2 Analizar Factores Prototipación*Propósito:*

Determinar si para el desarrollo de la aplicación es conveniente usar técnicas de prototipación.

Entrada- Origen de la Entrada:

Los valores ingresados por el usuario del sistema experto para los atributos Niveles de Composición, Progresión Fases desarrollo, Relación Fases desarrollo, Factores de Diseño a considerar en etapas tempranas del desarrollo, Existencia de Aplicaciones similares en el mercado y Tipo de Modificación explícitamente previstas.

Razonamiento:

Las técnicas de prototipación son adecuadas si los factores predominantes son:

- Alta variabilidad de los procesos
- Factores de diseño que inciden en etapas tempranas (arquitectura, eficacia)
- La aplicación es innovadora
- Subsistemas complejos
- Progresión no uniforme y secuencial en las fases de desarrollo
- Alta retroalimentación en las fases de desarrollo
- Progresión no uniforme y secuencial en las fases de desarrollo del proyecto

Salida- Destino de la Salida:

Valores ingresados por el usuario con destino a la Base de Conocimientos.

Módulo 1.2.3 Estimar Nivel Opciones

Propósito:

Determinar si la aplicación es multiopcional.

Entrada- Origen de la Entrada:

Los valores ingresados por el usuario del sistema experto para los atributos Posibilidad de Diseño, Posibilidad de Implementación.

Razonamiento:

Si la aplicación es multiopcional es conveniente aplicar el ciclo de vida en espiral.

Salida- Destino de la Salida:

Valores ingresados por el usuario con destino a la Base de Conocimientos.

Módulo 1.2.4 Estimar Integración HW-SW

Propósito:

Determinar nivel de integración del hardware y software que requiere la aplicación a desarrollar.

Entrada- Origen de la Entrada:

Los valores ingresados por el usuario del sistema experto para el atributo Integración entre hardware y software.

Razonamiento:

Si la aplicación requiere una fuerte integración hardware-software es conveniente aplicar el ciclo de vida en espiral.

Salida- Destino de la Salida:

Valores ingresados por el usuario con destino a la Base de Conocimientos.

Módulo 1.3.1 Analizar Coordinación

Propósito:

Analizar los factores relacionados con administración, mantenimiento y reusabilidad que inciden en las decisiones de coordinación en el desarrollo de un proyecto.

Para llevar a cabo la tarea el sistema debe realizar las siguientes subtareas:

Módulo 1.3.1.1 Estimar Factores Reúso

Módulo 1.3.1.2 Estimar Factores Mantenimiento

Módulo 1.3.1.3 Estimar Recursos Software

Módulo 1.3.1.4 Estimar Factores Administración

Entrada- Origen de la Entrada:

Los valores estimados por el usuarios del sistema experto, para los atributos vinculados al concepto Coordinación.

Razonamiento:

Los valores obtenidos permiten determinar

- Necesidad de Entregas intermedias
- Conveniencia de usar metodologías probadas
- Responsable de la gestión
- Aplicabilidad de modelos (cascada, objeto, espiral)
- Factibilidad de usar prototipación

Salida- Destino de la Salida:

Los valores obtenidos con destino a la Base de Conocimientos.

Módulo 1.3.2 Estimar Relación con Usuario

Propósito:

Determinar si el modelo por prototipo es aplicable.

Entrada- Origen de la Entrada:

Los valores ingresados por el usuario del sistema experto para los atributos Participación del cliente en el desarrollo, Introducción gradual del sistema a los usuarios.

Razonamiento:

El modelo por prototipo es aplicable cuando se requiere una fuerte participación del cliente en el desarrollo o es necesario facilitar la introducción gradual del sistema para los usuarios.

Salida- Destino de la Salida:

Los valores ingresados por el usuario con destino la Base de Conocimientos. Determinación de la aplicabilidad del prototipo con destino la Base de Conocimientos.

Módulo 1.3.3 Estimar Factores Equipo Proyecto

Propósito:

Determinar si el modelo por prototipo es aplicable.

Entrada- Origen de la Entrada:

Los valores ingresados por el usuario del sistema experto para los atributos Experiencia en el uso de las técnicas de IS, Experiencia en el tipo de aplicación.

Razonamiento:

El modelo por prototipo es aplicable cuando existe inexperiencia en el sistema a desarrollar o en las técnicas de IS a usar.

Salida- Destino de la Salida:

Los valores ingresados por el usuario con destino la Base de Conocimientos. Determinación de la aplicabilidad del prototipo con destino la Base de Conocimientos.

Módulo 1.3.4 Analizar Riesgos del Proyecto

Propósito:

Determinar el nivel de riesgo y su evaluabilidad.

Para llevar a cabo la tarea el sistema debe realizar las siguientes subtareas:

Módulo 1.3.4.1 Estimar Riesgo Existente

Módulo 1.3.4.2 Estimar Habilidad para Evaluar Riesgo

Entrada- Origen de la Entrada:

Los valores estimados por el usuario del sistema experto, para los atributos vinculados al concepto Riesgos del Proyecto.

Razonamiento:

Los valores obtenidos permiten determinar:

- Nivel de riesgo
- Nivel de factor de riesgo evaluable

Salida- Destino de la Salida:

Los valores obtenidos con destino a la Base de Conocimientos.

Módulo 1.3.1.1 Estimar Factores Reúso

Propósito:

Determinar si existe un sistema anterior desarrollado en objetos y si el modelo de objetos es aplicable.

Entrada- Origen de la Entrada:

Los valores ingresados por el usuario del sistema experto para los atributos Estrategia de reúso de aplicación existente, Reúso de aplicación existente en OO, Reúso de aplicación a desarrollar en aplicaciones futuras.

Razonamiento:

Existe un sistema anterior desarrollado en objetos si la aplicación a desarrollar es un subsistema, ampliación o modificación de un sistema desarrollado en objetos.

Si es necesario reusar software existente desarrollado en objetos o si es necesario un reúso posterior de componentes para el siguiente proyecto puede considerarse que existe un sistema anterior desarrollado en objetos.

Si se definen explícitamente estrategias de reúso de software existente el modelo de objetos es aplicable.

Salida- Destino de la Salida:

Los valores ingresados por el usuario con destino la Base de Conocimientos. Determinación de la aplicabilidad del modelo de objetos y existencia de sistema anterior desarrollado en objetos con destino la Base de Conocimientos.

Módulo 1.3.1.2 Estimar Factores Mantenimiento***Propósito:***

Determinar si el modelo en espiral es aplicable.

Entrada- Origen de la Entrada:

Los valores ingresados por el usuario del sistema experto para los atributos Procedimientos para cambios durante el desarrollo, Procedimientos para Desarrollo y Mantenimiento.

Razonamiento:

Si es factible y necesario usar los mismos procedimientos para desarrollo y mantenimiento o se requiere de acuerdos confirmados para los cambios durante el desarrollo, el modelo en espiral es aplicable.

Salida- Destino de la Salida:

Los valores ingresados por el usuario con destino la Base de Conocimientos. Determinación de la aplicabilidad del modelo en espiral con destino la Base de Conocimientos.

Módulo 1.3.1.3 Estimar Recursos Software***Propósito:***

Determinar si el modelo por prototipo es factible.

Entrada- Origen de la Entrada:

Los valores ingresados por el usuario del sistema experto para los atributos Software para prototipar.

Razonamiento:

El modelo por prototipo es factible si se dispone de software para prototipar o es posible adquirirlo.

Salida- Destino de la Salida:

Los valores ingresados por el usuario con destino la Base de Conocimientos. Determinación de la factibilidad del modelo por prototipo con destino la Base de Conocimientos.

Módulo 1.3.1.4 Estimar Factores Administración*Propósito:*

Determinar si existen entregas intermedias, el control y la formalidad de la gestión, y la conveniencia de usar metodologías.

Entrada- Origen de la Entrada:

Los valores ingresados por el usuario del sistema experto para los atributos Formalidad, Entregas, Responsabilidad del desarrollo, control de la Gestión, Factibilidad de usar una Metodología, Necesidad de usar una Metodología.

Razonamiento:

El ciclo de vida en cascada es recomendable en caso de una gestión de proyecto muy formal, con un ajustado control, si se terceriza el desarrollo y si es conveniente usar metodologías ampliamente probadas.

El modelo de objetos es aplicable en caso de que existan entregas intermedias, que la gestión de proyecto sea poco o medianamente formal, y que las metodologías a usar estén medianamente probadas.

El modelo por prototipo es aplicable si existen entregas intermedias.

El modelo en espiral es aplicable si las metodologías a usar están poco probadas, si la gestión de proyecto es poco o medianamente formal.

Salida- Destino de la Salida:

Los valores ingresados por el usuario con destino la Base de Conocimientos. Determinación de la entregas intermedias, conveniencia de usar modelos, aplicabilidad de los modelos (objetos, prototipo, espiral, cascada) con destino la Base de Conocimientos.

Módulo 1.3.4.1 Estimar Riesgo Existente*Propósito:*

Determinar Aceptabilidad del riesgo y la existencia del factor de riesgo y la aplicabilidad del modelo en espiral.

Entrada- Origen de la Entrada:

Los valores ingresados por el usuario del sistema experto para los atributos Viabilidad del software, Factor de riesgo, Categorías riesgo, Estrategias para incorporar calidad y Terminación de proyectos.

Razonamiento:

El modelo en espiral es aplicable si existe factor de riesgo, si se definen mecanismos explícitos para incorporar calidad, si se estima la terminación prematura de proyectos no viables.

Si hay dudas sobre la viabilidad del software o se estiman riesgos técnicos en el desarrollo del sistema puede decirse que existe factor de riesgo.

El riesgo es aceptable para el modelo en espiral si el factor de riesgo es mediano o alto.

El riesgo es aceptable para el modelo de objetos si el factor de riesgo es mediano o bajo.

El modelo en cascada es adecuado si el riesgo es bajo o inexistente.

Salida- Destino de la Salida:

Los valores ingresados por el usuario con destino la Base de Conocimientos. Determinación de la aceptabilidad del riesgo, de la aplicabilidad del modelo en espiral y de la existencia del factor de riesgo con destino la Base de Conocimientos.

Módulo 1.3.4.2 Estimar Habilidad para Evaluar Riesgo*Propósito:*

Determinar la evaluabilidad del riesgo, la aplicabilidad del modelo en espiral y la habilidad para evaluar el riesgo.

Entrada- Origen de la Entrada:

Los valores ingresados por el usuario del sistema experto para los atributos Estrategias para el Análisis de riesgo, Métodos y técnicas para Análisis de riesgo, Identificación de Alternativas, Identificación Riesgos para cada Alternativa.

Razonamiento:

El modelo en espiral es aplicable si se cuenta con habilidad para la evaluación del riesgo y si se definen explícitamente estrategias para analizar los riesgos. Si se identifican alternativas para resolver los riesgos y se identifican los riesgos asociados a cada una de esas alternativas, se cuenta entonces con habilidad para la evaluación del riesgo.

El factor de riesgo es evaluable si existe factor de riesgo y se cuenta con habilidad para resolver los riesgos y se cuenta con métodos y técnicas para evaluar el riesgo.

Salida- Destino de la Salida:

Los valores ingresados por el usuario con destino la Base de Conocimientos. Determinación de la evaluabilidad del riesgo, de la aplicabilidad del modelo en espiral y de la habilidad para evaluar riesgo con destino la Base de Conocimientos.

6.2.3.2 COMPROBACIÓN DE LOS CONOCIMIENTOS ESTRATÉGICOS

El modelo funcional ha sido validado por el experto. Se han realizado sesiones de entrevistas no estructuradas en las cuales el experto ha evaluado la representación de conocimientos para el desempeño de la tarea. La comprobación ha sido satisfactoria.

6.2.4 PASO 4: IDENTIFICACIÓN DE LOS CONOCIMIENTOS TÁCTICOS

En esta fase se documentan los conocimientos tácticos, es decir, como el experto usa los hechos conocidos y las hipótesis actuales sobre el caso para obtener nuevos hechos e hipótesis. Este análisis ha permitido producir una definición detallada de cada paso de razonamiento que deberá ejecutar el sistema experto.

Para representar los conocimientos tácticos se ha utilizado como representación intermedia las seudoreglas. Esta representación es muy familiar para el experto, por su actividad profesional en el área de desarrollo de sistemas informáticos. En particular, muchas de las seudoreglas fueron documentándose a medida que se realizaba la comprobación de los conocimientos estratégicos.

6.2.4.1 ANALISIS DE LOS CONOCIMIENTOS TÁCTICOS - SEUDORREGLAS

El conocimiento se representa completando una Hoja de Reglas siguiendo la propuesta de Gómez [Gómez, A. y otros 1997] con el siguiente diseño:

<i>Estado de la regla</i>	<i>Texto de la regla</i>
Palabras del experto	
Formulación externa de la regla	
Nombre de la regla	

La Hoja de Reglas se ha completado de la siguiente manera:

- Palabras del experto: a partir de las palabras del experto durante la fase de Adquisición y como descripción detallada de la descomposición funcional realizada en el paso 2 del proceso de Conceptualización se identificaron las reglas bajo el formato: “**SI** condición1,...,condición n **ENTONCES** acción1,..., acción n”, siendo este formato habitual para el razonamiento del experto.

- Formulación externa de la reglas: a partir de la identificación de reglas del experto se formalizan las reglas considerando la categorización de conceptos realizada en el primer paso del proceso de Conceptualización.
- Nombre de la regla: es un nombre que identifica a cada una de las reglas en la herramienta software (Kappa).

Las seudorreglas se han agrupado de acuerdo con las tres áreas definidas en la Adquisición de Conocimientos por el grupo de expertos:

- Área de Especificación de Requerimientos: tablas comprendidas entre la 6-4 y la 6-8.
- Área de Tipo de Aplicación: tablas comprendidas entre la 6-9 y la 6-39.
- Área de Gestión de Proyecto: tablas comprendidas entre la 6-40 y la 6-91.
- Ciclo de Vida del Proyecto: tablas comprendidas entre la 6-92 y la 6-118

Estado de la regla	Texto de la regla
Palabras del experto	<p><i>Si</i> <i>El usuario ha explicitado los requisitos formalmente y ha explicitado los requisitos exhaustivamente y ha explicitado los requisitos uniformemente</i> <i>Entonces</i> <i>Se han definido claramente los requisitos del sistema y Se han definido claramente los límites del sistema</i></p>
Formulación externa de la regla	<p><i>Si</i> <i>Tipo de Definición es igual a formalmente y exhaustivamente y uniformemente</i> <i>Entonces</i> <i>Definición Requisitos es definición clara y Definición Límites es definición clara.</i></p>
Nombre de la regla	REGLA ER-R1

Tabla 6-4: Selección del Ciclo de vida para el área Especificación de Requerimientos

Estado de la regla	Texto de la regla
Palabras del experto	<p><i>Si</i> <i>Se han definido claramente los requisitos del sistema y Se han definido claramente los límites del sistema y El usuario ha explicitado la mayoría de los requerimientos al comienzo del proyecto</i> <i>Entonces</i> <i>La Especificación de Requerimientos indica usar modelo en Cascada</i></p>
Formulación externa de la regla	<p><i>Si</i> <i>Definición Requisitos es definición clara y Definición Límites es definición clara y Grado de Certidumbre al inicio del proyecto es Mayoritariamente definidos</i> <i>Entonces</i> <i>CV Propuesto Requisitos es Cascada.</i></p>
Nombre de la regla	REGLA ER-R2

Tabla 6-5: Selección del Ciclo de vida para el área Especificación de Requerimientos

Estado de la regla	Texto de la regla
Palabras del experto	<p><i>Si</i> <i>El usuario ha explicitado los requisitos informalmente y ha explicitado los requisitos incompletos y ha explicitado los requisitos desestructuradamente.</i> <i>Entonces</i> <i>Hay gran nivel de incertidumbre en la especificación de requisitos y Hay alta incertidumbre en el entorno del sistema.</i></p>
Formulación externa de la regla	<p><i>Si</i> <i>Tipo de Definición es igual a informalmente e incompleto y desestructuradamente</i> <i>Entonces</i> <i>Definición Requisitos es definición incierta y Definición Límites es definición incierta.</i></p>
Nombre de la regla	REGLA ER-R3

Tabla 6-6: Selección del Ciclo de vida para el área Especificación de Requerimientos

Estado de la regla	Texto de la regla
Palabras del experto	<p><i>Si</i> <i>Hay gran nivel de incertidumbre en la especificación de requisitos y Hay alta incertidumbre en el entorno del sistema y El usuario NO ha explicitado la mayoría de los requerimientos al comienzo del proyecto</i> <i>Entonces</i> <i>La Especificación de Requerimientos indica usar modelo en Espiral y La Especificación de requerimientos indica usar modelo de Objetos.</i></p>
Formulación externa de la regla	<p><i>Si</i> <i>Definición Requisitos es definición incierta y Definición Límites es definición incierta y Grado de Certidumbre al inicio del proyecto es Parcialmente definidos</i> <i>Entonces</i> <i>CV Propuesto Requisitos es Espiral y Objetos.</i></p>
Nombre de la regla	REGLA ER-R4

Tabla 6-7: Selección del Ciclo de vida para el área Especificación de Requerimientos

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>El usuario es muy exigente con los requerimientos del sistema respecto del producto final.</i> <i>Entonces</i> <i>La Especificación de Requerimientos indica usar modelo en Espiral y La Especificación de requerimientos indica usar modelo de Objetos.</i></p>
Formulación externa de la regla	<p>Si <i>Grado de Cumplimiento en Producto final es Usuario muy exigente.</i> <i>Entonces</i> <i>CV Propuesto Requisitos es Espiral y Objetos.</i></p>
Nombre de la regla	REGLA ER-R5

Tabla 6-8: Selección del Ciclo de vida para el área Especificación de Requerimientos

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>Se presume independencia entre las fases del desarrollo del proyecto</i> <i>Entonces</i> <i>La necesidad de retroalimentación en las fases del desarrollo del proyecto es baja</i></p>
Formulación externa de la regla	<p>Si <i>Relación Fases desarrollo es igual a Independencia</i> <i>Entonces</i> <i>Retroalimentación Fases Desarrollo es baja</i></p>
Nombre de la regla	REGLA CV-R1

Tabla 6-9: Selección del Ciclo de vida para el área Tipo de Aplicación

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>Se presume poca dependencia entre las fases de desarrollo del proyecto</i> <i>Entonces</i> <i>La necesidad de retroalimentación en las fases del desarrollo del proyecto es baja</i></p>
Formulación externa de la regla	<p>Si <i>Relación Fases desarrollo es igual a Poca Dependencia</i> <i>Entonces</i> <i>Retroalimentación Fases Desarrollo es baja</i></p>
Nombre de la regla	REGLA CV-R2

Tabla 6-10: Selección del Ciclo de vida para el área Tipo de Aplicación

Estado de la regla	Texto de la regla
Palabras del experto	<p><i>Si</i> <i>Se presume mucha dependencia entre las fases de desarrollo del proyecto</i> <i>Entonces</i> <i>La necesidad de retroalimentación en las fases del desarrollo del proyecto es alta</i></p>
Formulación externa de la regla	<p><i>Si</i> <i>Relación Fases desarrollo es igual a Mucha Dependencia</i> <i>Entonces</i> <i>Retroalimentación Fases Desarrollo es alta</i></p>
Nombre de la regla	REGLA CV-R3

Tabla 6-11: Selección del Ciclo de vida para el área Tipo de Aplicación

Estado de la regla	Texto de la regla
Palabras del experto	<p><i>Si</i> <i>Se puede asumir una progresión relativamente uniforme y secuencial en las fases de desarrollo del proyecto y La necesidad de retroalimentación en las fases del desarrollo del proyecto es baja Y El sistema tiene componentes de fuerte contenido algorítmico</i> <i>Entonces</i> <i>Las características del Tipo de Aplicación indican usar modelo en Cascada.</i></p>
Formulación externa de la regla	<p><i>Si</i> <i>Progresión Fases Desarrollo es uniforme y secuencial y Retroalimentación Fases Desarrollo es baja y Componentes Predominantes es algorítmicos.</i> <i>Entonces</i> <i>CV Propuesto Aplicación es Cascada</i></p>
Nombre de la regla	REGLA CV-R4

Tabla 6-12: Selección del Ciclo de vida para el área Tipo de Aplicación

Estado de la regla	Texto de la regla
Palabras del experto	<p><i>Si</i> <i>Se puede asumir una progresión relativamente uniforme y secuencial en las fases de desarrollo del proyecto y La necesidad de retroalimentación en las fases del desarrollo del proyecto es baja Y El tipo de aplicación presenta un comportamiento predominantemente BATCH</i> <i>Entonces</i> <i>Las características del Tipo Aplicación indican usar modelo en Cascada</i></p>
Formulación externa de la regla	<p><i>Si</i> <i>Progresión Fases Desarrollo es uniforme y secuencial y Retroalimentación Fases Desarrollo es baja y Comportamiento Predominante es batch</i> <i>Entonces</i> <i>CV Propuesto Aplicación es Cascada</i></p>
Nombre de la regla	REGLA TA-R101

Tabla 6-13: Selección del Ciclo de vida para el área Tipo de Aplicación

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>Se puede asumir una progresión relativamente uniforme y secuencial en las fases de desarrollo del proyecto Y el sistema presenta subsistemas de baja complejidad y El sistema tiene componentes de fuerte contenido algorítmico</i> <i>Entonces</i> <i>El Tipo de Aplicación indica usar modelo en Cascada</i></p>
Formulación externa de la regla	<p>Si <i>Progresión Fases Desarrollo es uniforme y secuencial y Complejidad subsistemas es baja y Componentes Predominantes es algorítmico</i> <i>Entonces</i> <i>CV Propuesto Aplicación es Cascada</i></p>
Nombre de la regla	REGLA TA-R102

Tabla 6-14: Selección del Ciclo de vida para el área Tipo de aplicación

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>Se puede asumir progresión relativamente uniforme y secuencial en las fases de desarrollo del proyecto Y el sistema tiene subsistemas de baja complejidad y el tipo de aplicación tiene un comportamiento predominantemente BATCH</i> <i>Entonces</i> <i>El Tipo de Aplicación indica usar modelo en Cascada</i></p>
Formulación externa de la regla	<p>Si <i>Progresión Fases Desarrollo es uniforme y secuencial y Complejidad subsistemas es baja y Comportamiento Predominante es batch</i> <i>Entonces</i> <i>CV Propuesto Aplicación es Cascada</i></p>
Nombre de la regla	REGLA TA-R103

Tabla 6-15: Selección del Ciclo de vida para el área Tipo de Aplicación

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>Se puede asumir una progresión NO uniforme y secuencial en las fases de desarrollo del proyecto</i> <i>Entonces</i> <i>Las técnicas de prototipación son más convenientes.</i></p>
Formulación externa de la regla	<p>Si <i>Progresión Fases Desarrollo es No uniforme y secuencial</i> <i>Entonces</i> <i>Modelado Prototipo es Convenient</i></p>
Nombre de la regla	REGLA CV-R5

Tabla 6-16: Selección del Ciclo de vida para el área Tipo de Aplicación

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>La necesidad de retroalimentación en las fases del desarrollo del proyecto es alta</i> <i>Entonces</i> <i>Las técnicas de prototipación son más convenientes</i></p>
Formulación externa de la regla	<p>Si <i>Retroalimentación Fases Desarrollo es alta</i> <i>Entonces</i> <i>Modelado Prototipo es Conveniente</i></p>
Nombre de la regla	REGLA CV-R6

Tabla 6-17: Selección del Ciclo de vida para el área Tipo de Aplicación

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>Es necesario descomponer el sistema en pocos niveles para su mejor comprensión y Es necesario descomponer el sistema en pocos niveles para su mejor manipulación</i> <i>Entonces</i> <i>El sistema presenta subsistemas de baja complejidad</i></p>
Formulación externa de la regla	<p>Si <i>Niveles de Composición es Pocos Subsistemas</i> <i>Entonces</i> <i>Complejidad Subsistemas es Baja</i></p>
Nombre de la regla	REGLA TA-R1

Tabla 6-18: Selección del Ciclo de vida para el área Tipo de Aplicación

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>Es necesario descomponer el sistema en varios niveles para su mejor comprensión Y Es necesario descomponer el sistema en varios niveles para su mejor manipulación</i> <i>Entonces</i> <i>El sistema presenta subsistemas complejos</i></p>
Formulación externa de la regla	<p>Si <i>Niveles de Composición es Varios Subsistemas</i> <i>Entonces</i> <i>Complejidad Subsistemas es Alta</i></p>
Nombre de la regla	REGLA TA-R2

Tabla 6-19: Selección del Ciclo de vida para el área Tipo de Aplicación

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si El sistema se basa en componentes de fuerte contenido matemático Entonces Las técnicas de modelado OO son más convenientes</i>
Formulación externa de la regla	<i>Si Componentes Predominantes es matemáticos Entonces Modelado Objetos es Conveniente</i>
Nombre de la regla	REGLA TA-R4

Tabla 6-20: Selección del Ciclo de vida para el área Tipo de Aplicación

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si El sistema se basa en componentes de fuerte contenido gráfico Entonces Las técnicas de modelado OO son más convenientes</i>
Formulación externa de la regla	<i>Si Componentes Predominantes es gráficos Entonces Modelado Objetos es conveniente</i>
Nombre de la regla	REGLA TA-R5

Tabla 6-21: Selección del Ciclo de vida para el área Tipo de Aplicación

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si El sistema tiene subsistemas que presentan comportamiento dinámico Entonces Las técnicas de modelado OO son más convenientes</i>
Formulación externa de la regla	<i>Si Comportamiento Predominante es dinámico Entonces Modelado Objetos es Conveniente</i>
Nombre de la regla	REGLA TA-R3

Tabla 6-22: Selección del Ciclo de vida para el área Tipo de Aplicación

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si El tipo de aplicación presenta comportamiento fuertemente interactivo Entonces Las técnicas de modelado OO son más convenientes</i>
Formulación externa de la regla	<i>Si Comportamiento Predominante es interactivo Entonces Modelado Objetos es Conveniente</i>
Nombre de la regla	REGLA TA-R7

Tabla 6-23: Selección del Ciclo de vida para el área Tipo de Aplicación

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si El tipo de aplicación presenta comportamiento en Tiempo Real Entonces Las técnicas de modelado OO son más convenientes</i>
Formulación externa de la regla	<i>Si Comportamiento Predominante es tiempo real Entonces Modelado Objetos es Conveniente</i>
Nombre de la regla	REGLA TA-R8

Tabla 6-24: Selección del Ciclo de vida para el área Tipo de Aplicación

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si El tipo de aplicación es desarrollo de software de base Entonces Las técnicas de modelado OO son más convenientes</i>
Formulación externa de la regla	<i>Si Orientación es software de base Entonces Modelado Objetos es Conveniente</i>
Nombre de la regla	REGLA TA-R9

Tabla 6-25: Selección del Ciclo de vida para el área Tipo de Aplicación

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si El sistema es explícitamente sensible a cambios Entonces El sistema presenta una alta variabilidad en los procesos</i>
Formulación externa de la regla	<i>Si Tipo Modificación es Redefinición Entonces Variabilidad Procesos es Alta</i>
Nombre de la regla	REGLA TA-R11

Tabla 6-26: Selección del Ciclo de vida para el área Tipo de Aplicación

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si El sistema es explícitamente sensible a extensiones Entonces El sistema presenta una alta variabilidad en los procesos</i>
Formulación externa de la regla	<i>Si Tipo Modificación es Extensión Entonces Variabilidad Procesos es Alta</i>
Nombre de la regla	REGLA TA-R12

Tabla 6-27: Selección del Ciclo de vida para el área Tipo de Aplicación

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si El sistema es explícitamente sensible a ampliaciones. Entonces El sistema presenta una alta variabilidad en los procesos</i>
Formulación externa de la regla	<i>Si Tipo Modificación es Ampliación Entonces Variabilidad Procesos es Alta</i>
Nombre de la regla	REGLA TA-R13

Tabla 6-28: Selección del Ciclo de vida para el área Tipo de Aplicación

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si El proyecto requiere el descubrimiento de problemas de arquitectura en etapas tempranas del desarrollo Entonces Las técnicas de prototipación son más convenientes</i>
Formulación externa de la regla	<i>Si Factores Diseño es Problemas de Arquitectura Entonces Modelado Prototipo es Convenient</i>
Nombre de la regla	REGLA TA-R14

Tabla 6-29: Selección del Ciclo de vida para el área Tipo de Aplicación

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si El proyecto requiere el descubrimiento de problemas de eficacia en etapas tempranas del desarrollo Entonces Las técnicas de prototipación son más convenientes</i>
Formulación externa de la regla	<i>Si Factores Diseño es Problemas de eficacia Entonces Modelado Prototipo es Convenient</i>
Nombre de la regla	REGLA TA-R15

Tabla 6-30: Selección del Ciclo de vida para el área Tipo de Aplicación

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si El sistema presenta una alta variabilidad en los procesos Entonces Técnica Modelado es Prototipo son más convenientes</i>
Formulación externa de la regla	<i>Si Variabilidad Procesos es Alta Entonces Modelado Prototipo es Convenient</i>
Nombre de la regla	REGLA TA-R16

Tabla 6-31: Selección del Ciclo de vida para el área Tipo de Aplicación

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si El sistema a diseñar es innovador Entonces Las técnicas de prototipación son más convenientes</i>
Formulación externa de la regla	<i>Si Existencia Aplicaciones es Innovador Entonces Modelado Prototipo es Conveniente</i>
Nombre de la regla	REGLA TA-R17

Tabla 6-32: Selección del Ciclo de vida para el área Tipo de Aplicación

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si El tipo de aplicación es un SSBBCC sistema basado en conocimientos Entonces Las técnicas de prototipación son más convenientes</i>
Formulación externa de la regla	<i>Si Orientación es Sistema Basado en conocimiento Entonces Modelado Prototipo es Conveniente</i>
Nombre de la regla	REGLA TA-R18

Tabla 6-33: Selección del Ciclo de vida para el área Tipo de Aplicación

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si El sistema presenta subsistemas complejos Entonces Las técnicas de prototipación son más convenientes</i>
Formulación externa de la regla	<i>Si Complejidad Subsistemas es Alta Entonces Modelado Prototipo es Conveniente</i>
Nombre de la regla	REGLA TA-R19

Tabla 6-34: Selección del Ciclo de vida para el área Tipo de Aplicación

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si Las técnicas de modelado OO son más convenientes Y Las técnicas de prototipación son más convenientes Entonces El Tipo de Aplicación indica usar modelo de Objetos</i>
Formulación externa de la regla	<i>Si Modelado Objetos es conveniente y Modelado Prototipo es conveniente Entonces CV Propuesto Aplicación es Objetos</i>
Nombre de la regla	REGLA TA-R20

Tabla 6-35: Selección del Ciclo de vida para el área Tipo de Aplicación

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>Las técnicas de modelado OO son más convenientes Y Las técnicas de prototipación son más convenientes Y El tipo de aplicación requiere el desarrollo fuertemente integrado de hardware y software Entonces El Tipo de Aplicación indica usar modelo en Espiral</i></p>
Formulación externa de la regla	<p>Si <i>Modelado Objetos es conveniente Y Modelado Prototipo es conveniente Y Integración HW-SW es fuertemente Entonces CV Propuesto Aplicación es Espiral</i></p>
Nombre de la regla	REGLA TA-R21

Tabla 6-36: Selección del Ciclo de vida para el área Tipo de Aplicación

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>El sistema a desarrollar puede resolverse por varias opciones de diseño Entonces El sistema es multiopcional</i></p>
Formulación externa de la regla	<p>Si <i>Posibilidad Diseño es Varias opciones Entonces Opcionalidad es Múltiple</i></p>
Nombre de la regla	REGLA TA-R22

Tabla 6-37: Selección del Ciclo de vida para el área Tipo de Aplicación

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>El sistema a desarrollar puede resolverse por varias opciones de implementación Entonces El sistema es multiopcional</i></p>
Formulación externa de la regla	<p>Si <i>Posibilidad Implementación es Varias opciones Entonces Opcionalidad es Múltiple</i></p>
Nombre de la regla	REGLA TA-R23

Tabla 6-38: Selección del Ciclo de vida para el área Tipo de Aplicación

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>Las técnicas de modelado OO son más convenientes Y Las técnicas de prototipación son más convenientes Y El sistema es multiopcional</i> <i>Entonces</i> <i>El Tipo de Aplicación indica usar modelo en Espiral</i></p>
Formulación externa de la regla	<p>Si <i>Modelado Objetos es conveniente Y Modelado Prototipo es conveniente Y Opcionalidad es Múltiple</i> <i>Entonces</i> <i>CV Propuesto Aplicación es Espiral</i></p>
Nombre de la regla	REGLA TAR-24

Tabla 6-39: Selección del Ciclo de vida para el área Tipo de Aplicación

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>Es necesario usar metodologías ampliamente probadas Y Es factible usar metodologías ampliamente probadas</i> <i>Entonces</i> <i>Es conveniente usar metodologías ampliamente probadas</i></p>
Formulación externa de la regla	<p>Si <i>Necesidad Metodología es ampliamente probada Y Factibilidad Metodología es ampliamente probada</i> <i>Entonces</i> <i>Conveniencia Metodología es ampliamente probada</i></p>
Nombre de la regla	REGLA GP-R1

Tabla 6-40: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>El factor de riesgo es bajo</i> <i>Entonces</i> <i>El riesgo es aceptable para modelo en cascada</i></p>
Formulación externa de la regla	<p>Si <i>Nivel riesgo es bajo</i> <i>Entonces</i> <i>Riesgo Cascada es aceptable</i></p>
Nombre de la regla	REGLA GP-R2

Tabla 6-41: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>No hay factor de riesgo</i> <i>Entonces</i> <i>El riesgo es aceptable para modelo en cascada</i></p>
Formulación externa de la regla	<p>Si <i>Nivel riesgo es no hay riesgo</i> <i>Entonces</i> <i>Riesgo Cascada es aceptable</i></p>
Nombre de la regla	REGLA GP-R4

Tabla 6-42: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>Es conveniente usar metodologías ampliamente probadas Y No es necesario disponer de una versión temprana del software desarrollado hasta que el proyecto este avanzado Y El riesgo es aceptable para modelo en cascada</i> <i>Entonces</i> <i>La Gestión del Proyecto indica usar modelo en Cascada</i></p>
Formulación externa de la regla	<p>Si <i>Conveniencia Metodología es ampliamente probada Y Entregas es versión completa Y Riesgo Cascada es aceptable</i> <i>Entonces</i> <i>CV Propuesto Gestión es Cascada</i></p>
Nombre de la regla	REGLA GP-R5

Tabla 6-43: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>El proyecto requiere una gestión de proyecto muy formal Y No es necesario disponer de una versión temprana del software desarrollado hasta que el proyecto este avanzado Y El riesgo es aceptable para modelo en cascada.</i> <i>Entonces</i> <i>La Gestión del Proyecto indica usar modelo en Cascada</i></p>
Formulación externa de la regla	<p>Si <i>Formalidad es muy formal Y Entregas es versión completa Y Riesgo Cascada es aceptable</i> <i>Entonces</i> <i>CV Propuesto Gestión es Cascada</i></p>
Nombre de la regla	REGLA GP-R14

Tabla 6-44: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>No se dispone de software para prototipar y no es posible adquirir software para prototipar</i> <i>Entonces</i> <i>El modelo por prototipo no es factible</i></p>
Formulación externa de la regla	<p>Si <i>Software para prototipar es No disponible y No adquirible</i> <i>Entonces</i> <i>Factibilidad Prototipo es No factible</i></p>
Nombre de la regla	REGLA GP-R3

Tabla 6-45: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si El modelo por prototipo no es factible Y El riesgo es aceptable para modelo en cascada Entonces La Gestión del Proyecto indica usar modelo en Cascada</i>
Formulación externa de la regla	<i>Si Factibilidad Prototipo es No factible Y Riesgo Cascada es aceptable Entonces CV Propuesto Gestión es Cascada</i>
Nombre de la regla	REGLA GP-R16

Tabla 6-46: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si Se terceriza el desarrollo Y Se requiere un ajustado control de la gestión del proyecto Y No es necesario disponer de una versión temprana hasta que el proyecto este avanzado. Entonces La Gestión del Proyecto indica usar modelo en Cascada</i>
Formulación externa de la regla	<i>Si Responsabilidad es Terceros Y Control gestión es muy ajustado Y Entregas es versión completa Entonces CV Propuesto Gestión es Cascada</i>
Nombre de la regla	REGLA GP-R6

Tabla 6-47: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si Es un subsistema de un sistema desarrollado en Objetos Entonces Existe sistema anterior desarrollado en Objetos</i>
Formulación externa de la regla	<i>Si Reuso aplicación OO es Subsistema del existente Entonces Sistema OO es existe</i>
Nombre de la regla	REGLA GP-R7

Tabla 6-48: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si Es una ampliación de un sistema desarrollado en Objetos Entonces Existe sistema anterior desarrollado en Objetos</i>
Formulación externa de la regla	<i>Si Reuso aplicación OO es Ampliación del existente Entonces Sistema OO es existe</i>
Nombre de la regla	REGLA GP-R8

Tabla 6-49: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si Es una modificación de un sistema desarrollado en Objetos Entonces Existe sistema anterior desarrollado en Objetos</i>
Formulación externa de la regla	<i>Si Reúso aplicación OO es Modificación del existente Entonces Sistema OO es existe</i>
Nombre de la regla	REGLA GP-R9

Tabla 6-50: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si Se prevé una fuerte necesidad de reutilizar los componentes de un software ya existente desarrollado en Objetos Entonces Existe sistema anterior desarrollado en Objetos</i>
Formulación externa de la regla	<i>Si Reúso aplicación OO es Componentes del actual Entonces Sistema OO es existe</i>
Nombre de la regla	REGLA GP-R10

Tabla 6-51: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si Se prevé una fuerte necesidad de reutilizar los componentes de un desarrollo para el siguiente proyecto. Entonces Existe sistema anterior desarrollado en Objetos</i>
Formulación externa de la regla	<i>Si Reúso aplicación futura es muy necesario Entonces Sistema OO es existe</i>
Nombre de la regla	REGLA GP-R11

Tabla 6-52: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si Es necesario una versión temprana disponible del software desarrollado Entonces Existen entregas intermedias</i>
Formulación externa de la regla	<i>Si Entregas es Versión temprana Entonces Entrega intermedia es existe</i>
Nombre de la regla	REGLA GP-R24

Tabla 6-53: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si Es necesario disponer de una versión temprana del software desarrollado con funciones a completar gradualmente Entonces Existen entregas intermedias</i>
Formulación externa de la regla	<i>Si Entregas es Versión gradual Entonces Entrega intermedia es existe</i>
Nombre de la regla	REGLA GP-R25

Tabla 6-54: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si Es necesario disponer de una versión parcial del software desarrollado con funciones a completar gradualmente Entonces Existen entregas intermedias</i>
Formulación externa de la regla	<i>Si Entregas es Versión parcial Entonces Entrega intermedia es existe</i>
Nombre de la regla	REGLA GP-R28

Tabla 6-55: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si Existen entregas intermedias Entonces El modelo de Objetos es aplicable Y El modelo por prototipo es aplicable</i>
Formulación externa de la regla	<i>Si Entrega intermedia es existe Entonces Aplicabilidad OO es aplicable y Aplicabilidad prototipo es aplicable</i>
Nombre de la regla	REGLA GP-R29

Tabla 6-56: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si Se definen explícitamente estrategias de reúso de software existente Entonces El modelo de Objetos es aplicable</i>
Formulación externa de la regla	<i>Si Reúso aplicación existente es estrategias explícitas Entonces Aplicabilidad OO es aplicable</i>
Nombre de la regla	REGLA GP-R26

Tabla 6-57: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si No hay factor de riesgo Entonces El riesgo es aceptable para el modelo de Objetos</i>
Formulación externa de la regla	<i>Si Nivel riesgo es No hay riesgo Entonces Riesgo Objetos es aceptable</i>
Nombre de la regla	REGLA GP-R19

Tabla 6-58: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si El factor de riesgo es bajo Entonces El riesgo es aceptable para el modelo de Objetos</i>
Formulación externa de la regla	<i>Si Nivel riesgo es bajo riesgo Entonces Riesgo Objetos es aceptable</i>
Nombre de la regla	REGLA GP-R12

Tabla 6-59: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si El factor de riesgo es mediano Entonces El riesgo es aceptable para el modelo de Objetos</i>
Formulación externa de la regla	<i>Si Nivel riesgo es mediano riesgo Entonces Riesgo Objetos es aceptable</i>
Nombre de la regla	REGLA GP-R13

Tabla 6-60: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si Es necesario usar metodologías medianamente probadas Y Es factible usar metodologías medianamente probadas Entonces Es conveniente usar metodologías medianamente probadas</i>
Formulación externa de la regla	<i>Si Necesidad Metodología es medianamente probada Y Factibilidad Metodología es medianamente probada Entonces Conveniencia Metodología es medianamente probada</i>
Nombre de la regla	REGLA GP-R37

Tabla 6-61: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>El proyecto requiere una gestión de proyecto medianamente formal Y Es conveniente usar metodologías medianamente probadas</i> <i>Entonces</i> <i>El modelo de Objetos es aplicable</i></p>
Formulación externa de la regla	<p>Si <i>Formalidad es medianamente formal Y Conveniencia Metodología es medianamente probada</i> <i>Entonces</i> <i>Aplicabilidad OO es aplicable</i></p>
Nombre de la regla	REGLA GP-R15

Tabla 6-62: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>El proyecto requiere una gestión de proyecto poco formal Y Es conveniente usar metodologías medianamente probadas</i> <i>Entonces</i> <i>El modelo de Objetos es aplicable</i></p>
Formulación externa de la regla	<p>Si <i>Formalidad es poco formal Y Conveniencia Metodología es medianamente probada</i> <i>Entonces</i> <i>Aplicabilidad OO es aplicable</i></p>
Nombre de la regla	REGLA GP-R17

Tabla 6-63: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>El modelo de Objetos es aplicable Y Existe sistema anterior desarrollado en Objetos Y El riesgo es aceptable para el modelo de Objetos</i> <i>Entonces</i> <i>La Gestión del Proyecto indica usar modelo de Objetos</i></p>
Formulación externa de la regla	<p>Si <i>Aplicabilidad OO es aplicable Y Sistema OO es existe Y Riesgo Objetos es aceptable</i> <i>Entonces</i> <i>CV Propuesto Gestión es Objetos</i></p>
Nombre de la regla	REGLA GP—R18

Tabla 6-64: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si Es posible adquirir software para prototipar Entonces El modelo por prototipo es factible</i>
Formulación externa de la regla	<i>Si Software para prototipar es factible de adquirir Entonces Factibilidad prototipo es factible</i>
Nombre de la regla	REGLA GP-R20

Tabla 6-65: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si Se dispone de software para prototipar Entonces El modelo por prototipo es factible</i>
Formulación externa de la regla	<i>Si Software para prototipar es hay disponible Entonces Factibilidad prototipo es factible</i>
Nombre de la regla	REGLA GP-R21

Tabla 6-66: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si Es necesario facilitar la introducción gradual del sistema para los usuarios Entonces El modelo por prototipo es aplicable</i>
Formulación externa de la regla	<i>Si Introducción gradual es necesaria Entonces Aplicabilidad prototipo es aplicable</i>
Nombre de la regla	REGLA GP-R22

Tabla 6-67: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si Es necesario contar con una fuerte participación del usuario cliente en el Desarrollo Entonces El modelo por prototipo es aplicable</i>
Formulación externa de la regla	<i>Si Participación es fuerte Entonces Aplicabilidad prototipo es aplicable</i>
Nombre de la regla	REGLA GP-R23

Tabla 6-68: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p><i>Si</i> <i>La gestión del proyecto prevé definir explícitamente estrategias para reusar software existente.</i> <i>Entonces</i> <i>El modelo por prototipo es aplicable</i></p>
Formulación externa de la regla	<p><i>Si</i> <i>Reúso aplicación existente es estrategias explícitas</i> <i>Entonces</i> <i>Aplicabilidad prototipo es aplicable</i></p>
Nombre de la regla	REGLA GP-R29

Tabla 6-69: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p><i>Si</i> <i>Equipo de proyecto no tiene experiencia previa en el tipo de aplicación a desarrollar</i> <i>Entonces</i> <i>El modelo por prototipo es aplicable</i></p>
Formulación externa de la regla	<p><i>Si</i> <i>Experiencia previa es No hay</i> <i>Entonces</i> <i>Aplicabilidad prototipo es aplicable</i></p>
Nombre de la regla	REGLA GP-R27

Tabla 6-70: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p><i>Si</i> <i>Existe inexperiencia con las técnicas de ingeniería de software que se usarán</i> <i>Entonces</i> <i>El modelo por prototipo es aplicable</i></p>
Formulación externa de la regla	<p><i>Si</i> <i>Experiencia técnicas IS es No hay</i> <i>Entonces</i> <i>Aplicabilidad prototipo es aplicable</i></p>
Nombre de la regla	REGLA GP-R30

Tabla 6-71: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p><i>Si</i> <i>Es necesario Y factible usar los mismos procedimientos para desarrollo y mantenimiento del producto software</i> <i>Entonces</i> <i>Es conveniente usar los mismos procedimientos para desarrollo y mantenimiento del producto software</i></p>
Formulación externa de la regla	<p><i>Si</i> <i>Procedimientos Desarrollo-Mantenimiento es necesario usar los mismos y es factible usar los mismos.</i> <i>Entonces</i> <i>Procedimientos D-M es conveniente</i></p>
Nombre de la regla	REGLA GP-R31

Tabla 6-72: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>Es conveniente usar los mismos procedimientos para desarrollo y mantenimiento del producto software</i> <i>Entonces</i> <i>El modelo en Espiral es aplicable</i></p>
Formulación externa de la regla	<p>Si <i>Procedimientos D-M es conveniente</i> <i>Entonces</i> <i>Aplicabilidad Espiral es aplicable</i></p>
Nombre de la regla	REGLA GP-R32

Tabla 6-73: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>Se requiere de acuerdos confirmados para los cambios a efectuarse durante el desarrollo del proyecto.</i> <i>Entonces</i> <i>El modelo en Espiral es aplicable</i></p>
Formulación externa de la regla	<p>Si <i>Procedimientos Cambios es acuerdos confirmados</i> <i>Entonces</i> <i>Aplicabilidad Espiral es aplicable</i></p>
Nombre de la regla	REGLA GP-R33

Tabla 6-74: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>El desarrollo del sistema es responsabilidad de la organización (no se terceriza)</i> <i>Entonces</i> <i>El modelo en Espiral es aplicable</i></p>
Formulación externa de la regla	<p>Si <i>Responsabilidad es organización propia</i> <i>Entonces</i> <i>Aplicabilidad Espiral es aplicable</i></p>
Nombre de la regla	REGLA GP-R34

Tabla 6-75: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>Se identifican los riesgos asociados con cada una de las alternativas Y Se identifican las diferentes maneras de resolver los riesgos</i> <i>Entonces</i> <i>Se cuenta con habilidad para la evaluación del riesgo</i></p>
Formulación externa de la regla	<p>Si <i>Riesgos Alternativa es se identifican Y Identificación alternativas es se identifican</i> <i>Entonces</i> <i>Habilidad Riesgo es existe</i></p>
Nombre de la regla	REGLA GP-R38

Tabla 6-76: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p><i>Si</i> <i>Se cuenta con habilidad para la evaluación del riesgo Y</i> <i>Es posible volver a etapas anteriores en el desarrollo del sistema cuando se identifican mejoras alternativas</i> <i>Entonces</i> <i>El modelo en Espiral es aplicable</i></p>
Formulación externa de la regla	<p><i>Si</i> <i>Habilidad Riesgo es existe Y Identificación Alternativas es etapas anteriores</i> <i>Entonces</i> <i>Aplicabilidad Espiral es aplicable</i></p>
Nombre de la regla	REGLA GP-R39

Tabla 6-77: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p><i>Si</i> <i>Se cuenta con habilidad para la evaluación del riesgo Y</i> <i>Es necesario volver a etapas anteriores en el desarrollo del sistema para la resolución de algún tópico nuevo de riesgo</i> <i>Entonces</i> <i>El modelo en Espiral es aplicable</i></p>
Formulación externa de la regla	<p><i>Si</i> <i>Habilidad Riesgo es existe Y Riesgos Alternativa es etapas anteriores</i> <i>Entonces</i> <i>Aplicabilidad Espiral es aplicable</i></p>
Nombre de la regla	REGLA GP-R41

Tabla 6-78: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p><i>Si</i> <i>Se cuenta con habilidad para la evaluación del riesgo Y</i> <i>Se estima la posibilidad de la terminación prematura de proyectos no viables</i> <i>Entonces</i> <i>El modelo en Espiral es aplicable</i></p>
Formulación externa de la regla	<p><i>Si</i> <i>Habilidad Riesgo es existe Y Terminación Proyectos es prematura</i> <i>Entonces</i> <i>Aplicabilidad Espiral es aplicable</i></p>
Nombre de la regla	REGLA GP-R42

Tabla 6-79: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si Existen dudas sobre la viabilidad del software Entonces Existe factor de riesgo</i>
Formulación externa de la regla	<i>Si Viabilidad software es hay dudas Entonces Factor riesgo es existe</i>
Nombre de la regla	REGLA GP-R35

Tabla 6-80: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si Se estiman riesgos técnicos en el desarrollo del sistema Entonces Existe factor de riesgo</i>
Formulación externa de la regla	<i>Si Categorías riesgo es técnico Entonces Factor riesgo es existe</i>
Nombre de la regla	REGLA GP-R36

Tabla 6-81: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si El factor de riesgo es mediano Entonces El riesgo es aceptable para el modelo en Espiral</i>
Formulación externa de la regla	<i>Si Nivel riesgo es mediano riesgo Entonces Riesgo Espiral es aceptable</i>
Nombre de la regla	REGLA GP-R50

Tabla 6-82: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si El factor de riesgo es alto Entonces El riesgo es aceptable para el modelo en Espiral</i>
Formulación externa de la regla	<i>Si Nivel riesgo es alto riesgo Entonces Riesgo Espiral es aceptable</i>
Nombre de la regla	REGLA GP-R51

Tabla 6-83: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si El riesgo es aceptable para el modelo en Espiral Entonces Existe factor de riesgo</i>
Formulación externa de la regla	<i>Si Riesgo Espiral es aceptable Entonces Factor riesgo es existe</i>
Nombre de la regla	REGLA GP-R52

Tabla 6-84: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si Existe factor de riesgo Y Es necesario definir explícitamente estrategias para realizar análisis de riesgo Entonces El modelo en Espiral es aplicable</i>
Formulación externa de la regla	<i>Si Factor riesgo es existe Y Análisis riesgo es estrategias explícitas Entonces Aplicabilidad Espiral es aplicable</i>
Nombre de la regla	REGLA GP-R43

Tabla 6-85: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si Existe factor de riesgo Y Es necesario definir mecanismos explícitos para incorporar objetivos de calidad en el desarrollo del producto. Entonces El modelo en Espiral es aplicable</i>
Formulación externa de la regla	<i>Si Factor riesgo es existe Y Objetivo calidad es mecanismos explícitos Entonces Aplicabilidad Espiral es aplicable</i>
Nombre de la regla	REGLA GP-R44

Tabla 6-86: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si Es necesario usar metodologías poco probadas Y Es factible usar metodologías poco probadas Entonces Es conveniente usar metodologías poco probadas</i>
Formulación externa de la regla	<i>Si Necesidad Metodología es poco probada Y Factibilidad Metodología es poco probada Entonces Conveniencia Metodología es poco probada</i>
Nombre de la regla	REGLA GP-R53

Tabla 6-87: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si El proyecto requiere una gestión de proyecto medianamente formal Y Es conveniente usar metodologías poco probadas Entonces El modelo en Espiral es aplicable</i>
Formulación externa de la regla	<i>Si Formalidad es medianamente formal Y Conveniencia Metodología es poco probada Entonces Aplicabilidad Espiral es aplicable</i>
Nombre de la regla	REGLA GP-R54

Tabla 6-88: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<i>Si El proyecto requiere una gestión de proyecto poco formal Y Es conveniente usar metodologías poco probadas Entonces El modelo en Espiral es aplicable</i>
Formulación externa de la regla	<i>Si Formalidad es poco formal Y Conveniencia Metodología es poco probada Entonces Aplicabilidad Espiral es aplicable</i>
Nombre de la regla	REGLA GP-R55

Tabla 6-89: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>Existe factor de riesgo Y Se cuenta con habilidad para la evaluación del riesgo Y Se cuenta con métodos y técnicas para evaluar el riesgo</i> <i>Entonces</i> <i>El factor de riesgo es evaluable</i></p>
Formulación externa de la regla	<p>Si <i>Factor Riesgo es existe Y Habilidad riesgo es existe Y Técnicas AR es se dispone</i> <i>Entonces</i> <i>Riesgo es evaluable</i></p>
Nombre de la regla	REGLA GP-R56

Tabla 6-90: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>El modelo por prototipo es aplicable Y El modelo por prototipo es factible Y El factor de riesgo es evaluable Y El modelo en Espiral es aplicable</i> <i>Entonces</i> <i>La Gestión del Proyecto indica usar modelo en Espiral</i></p>
Formulación externa de la regla	<p>Si <i>Aplicabilidad Prototipo es aplicable Y Factibilidad prototipo es factible Y Riesgo es evaluable Y Aplicabilidad Espiral es aplicable</i> <i>Entonces</i> <i>CV Propuesto Gestión es Espiral</i></p>
Nombre de la regla	REGLA GP-R45

Tabla 6-91: Selección del Ciclo de vida para el área Gestión de Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>La Especificación de Requerimientos indica usar modelo en Cascada Y El Tipo de Aplicación indica usar modelo en Cascada Y La Gestión de Proyecto indica usar modelo en Cascada</i> <i>Entonces</i> <i>El Ciclo de Vida sugerido es Cascada</i></p>
Formulación externa de la regla	<p>Si <i>CV Propuesto Requisitos es Cascada Y CV Propuesto Aplicación es Cascada Y CV Propuesto Gestión es Cascada</i> <i>Entonces</i> <i>El CV Propuesto Proyecto es Cascada</i></p>
Nombre de la regla	REGLA R0

Tabla 6-92: Selección del Ciclo de vida para Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>La Especificación de Requerimientos indica usar modelo de Objetos Y El Tipo de Aplicación indica usar modelo de Objetos Y La Gestión de Proyecto indica usar modelo de Objetos</i> <i>Entonces</i> <i>El Ciclo de Vida sugerido es Objetos</i></p>
Formulación externa de la regla	<p>Si <i>CV Propuesto Requisitos es Objetos Y CV Propuesto Aplicación es Objetos Y CV Propuesto Gestión es Objetos Entonces</i> <i>El CV Propuesto Proyecto es Objetos</i></p>
Nombre de la regla	REGLA R1

Tabla 6-93: Selección del Ciclo de vida para el Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>La Especificación de Requerimientos indica usar modelo en Espiral Y El Tipo de Aplicación indica usar modelo en Espiral Y La Gestión de Proyecto indica usar modelo en Espiral</i> <i>Entonces</i> <i>El Ciclo de Vida sugerido es Espiral</i></p>
Formulación externa de la regla	<p>Si <i>CV Propuesto Requisitos es Espiral Y CV Propuesto Aplicación es Espiral Y CV Propuesto Gestión es Espiral</i> <i>Entonces</i> <i>El CV Propuesto Proyecto es Espiral</i></p>
Nombre de la regla	REGLA R2

Tabla 6-94: Selección del Ciclo de vida para el Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>La Especificación de Requerimientos indica usar modelo en Cascada Y El Tipo de Aplicación indica usar modelo en Cascada Y La Gestión de Proyecto indica usar modelo de Objetos</i> <i>Entonces</i> <i>El Ciclo de Vida sugerido es Cascada</i></p>
Formulación externa de la regla	<p>Si <i>CV Propuesto Requisitos es Cascada Y CV Propuesto Aplicación es Cascada Y CV Propuesto Gestión es Objetos</i> <i>Entonces</i> <i>El CV Propuesto Proyecto es Cascada</i></p>
Nombre de la regla	REGLA R3

Tabla 6-95: Selección del Ciclo de vida para el Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>La Especificación de Requerimientos indica usar modelo en Cascada Y El Tipo de Aplicación indica usar modelo de Objetos Y La Gestión de Proyecto indica usar modelo en Cascada</i> <i>Entonces</i> <i>El Ciclo de Vida sugerido es Cascada</i></p>
Formulación externa de la regla	<p>Si <i>CV Propuesto Requisitos es Cascada Y CV Propuesto Aplicación es Objetos Y CV Propuesto Gestión es Cascada</i> <i>Entonces</i> <i>El CV Propuesto Proyecto es Cascada</i></p>
Nombre de la regla	REGLA R4

Tabla 6-96: Selección del Ciclo de vida para el Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>La Especificación de Requerimientos indica usar modelo de Objetos Y El Tipo de Aplicación indica usar modelo en Cascada Y La Gestión de Proyecto indica usar modelo en Cascada</i> <i>Entonces</i> <i>El Ciclo de Vida sugerido es Cascada</i></p>
Formulación externa de la regla	<p>Si <i>CV Propuesto Requisitos es Objetos Y CV Propuesto Aplicación es Cascada Y CV Propuesto Gestión es Cascada</i> <i>Entonces</i> <i>El CV Propuesto Proyecto es Cascada</i></p>
Nombre de la regla	REGLA R6

Tabla 6-97: Selección del Ciclo de vida para el Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>La Especificación de Requerimientos indica usar modelo en Cascada Y El Tipo de Aplicación indica usar modelo en Cascada Y La Gestión de Proyecto indica usar modelo en Espiral</i> <i>Entonces</i> <i>El Ciclo de Vida sugerido es Cascada</i></p>
Formulación externa de la regla	<p>Si <i>CV Propuesto Requisitos es Cascada Y CV Propuesto Aplicación es Cascada Y CV Propuesto Gestión es Espiral</i> <i>Entonces</i> <i>El CV Propuesto Proyecto es Cascada</i></p>
Nombre de la regla	REGLA R7

Tabla 6-98: Selección del Ciclo de vida para el Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>La Especificación de Requerimientos indica usar modelo en Cascada Y El Tipo de Aplicación indica usar modelo en Espiral Y La Gestión de Proyecto indica usar modelo en Cascada Entonces El Ciclo de Vida sugerido es Cascada</i></p>
Formulación externa de la regla	<p>Si <i>CV Propuesto Requisitos es Cascada Y CV Propuesto Aplicación es Espiral Y CV Propuesto Gestión es Cascada Entonces El CV Propuesto Proyecto es Cascada</i></p>
Nombre de la regla	REGLA R8

Tabla 6-99: Selección del Ciclo de vida para el Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>La Especificación de Requerimientos indica usar modelo en Espiral Y El Tipo de Aplicación indica usar modelo en Cascada Y La Gestión de Proyecto indica usar modelo en Cascada Entonces El Ciclo de Vida sugerido es Cascada</i></p>
Formulación externa de la regla	<p>Si <i>CV Propuesto Requisitos es Espiral Y CV Propuesto Aplicación es Cascada Y CV Propuesto Gestión es Cascada Entonces El CV Propuesto Proyecto es Cascada</i></p>
Nombre de la regla	REGLA R10

Tabla 6-100: Selección del Ciclo de vida para el Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>La Especificación de Requerimientos indica usar modelo de Objetos Y El Tipo de Aplicación indica usar modelo de Objetos Y La Gestión de Proyecto indica usar modelo en Cascada Entonces El Ciclo de Vida sugerido es Objetos</i></p>
Formulación externa de la regla	<p>Si <i>CV Propuesto Requisitos es Objetos Y CV Propuesto Aplicación es Objetos Y CV Propuesto Gestión es Cascada Entonces El CV Propuesto Proyecto es Objetos</i></p>
Nombre de la regla	REGLA R11

Tabla 6-101: Selección del Ciclo de vida para el Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>La Especificación de Requerimientos indica usar modelo de Objetos Y El Tipo de Aplicación indica usar modelo en Cascada Y La Gestión de Proyecto indica usar modelo de Objetos</i> <i>Entonces</i> <i>El Ciclo de Vida sugerido es Objetos</i></p>
Formulación externa de la regla	<p>Si <i>CV Propuesto Requisitos es Objetos Y CV Propuesto Aplicación es Cascada Y CV Propuesto Gestión es Objetos</i> <i>Entonces</i> <i>El CV Propuesto Proyecto es Objetos</i></p>
Nombre de la regla	REGLA R12

Tabla 6-102: Selección del Ciclo de vida para el Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>La Especificación de Requerimientos indica usar modelo en Cascada Y El Tipo de Aplicación indica usar modelo de Objetos Y La Gestión de Proyecto indica usar modelo de Objetos</i> <i>Entonces</i> <i>El Ciclo de Vida sugerido es Objetos</i></p>
Formulación externa de la regla	<p>Si <i>CV Propuesto Requisitos es Cascada Y CV Propuesto Aplicación es Objetos Y CV Propuesto Gestión es Objetos</i> <i>Entonces</i> <i>El CV Propuesto Proyecto es Objetos</i></p>
Nombre de la regla	REGLA R14

Tabla 6-103: Selección del Ciclo de vida para el Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>La Especificación de Requerimientos indica usar modelo de Objetos Y El Tipo de Aplicación indica usar modelo de Objetos Y La Gestión de Proyecto indica usar modelo en Espiral</i> <i>Entonces</i> <i>El Ciclo de Vida sugerido es Objetos</i></p>
Formulación externa de la regla	<p>Si <i>CV Propuesto Requisitos es Objetos Y CV Propuesto Aplicación es Objetos Y CV Propuesto Gestión es Espiral</i> <i>Entonces</i> <i>El CV Propuesto Proyecto es Objetos</i></p>
Nombre de la regla	REGLA R15

Tabla 6-104: Selección del Ciclo de vida para el Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>La Especificación de Requerimientos indica usar modelo de Objetos Y El Tipo de Aplicación indica usar modelo en Espiral Y La Gestión de Proyecto indica usar modelo de Objetos Entonces El Ciclo de Vida sugerido es Objetos</i></p>
Formulación externa de la regla	<p>Si <i>CV Propuesto Requisitos es Objetos Y CV Propuesto Aplicación es Espiral Y CV Propuesto Gestión es Objetos Entonces El CV Propuesto Proyecto es Objetos</i></p>
Nombre de la regla	REGLA R16

Tabla 6-105: Selección del Ciclo de vida para el Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>La Especificación de Requerimientos indica usar modelo en Espiral Y El Tipo de Aplicación indica usar modelo de Objetos Y La Gestión de Proyecto indica usar modelo de Objetos Entonces El Ciclo de Vida sugerido es Objetos</i></p>
Formulación externa de la regla	<p>Si <i>CV Propuesto Requisitos es Espiral Y CV Propuesto Aplicación es Objetos Y CV Propuesto Gestión es Objetos Entonces El CV Propuesto Proyecto es Objetos</i></p>
Nombre de la regla	REGLA R18

Tabla 6-106: Selección del Ciclo de vida para el Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>La Especificación de Requerimientos indica usar modelo en Espiral Y El Tipo de Aplicación indica usar modelo en Espiral Y La Gestión de Proyecto indica usar modelo en Cascada Entonces El Ciclo de Vida sugerido es Espiral</i></p>
Formulación externa de la regla	<p>Si <i>CV Propuesto Requisitos es Espiral Y CV Propuesto Aplicación es Espiral Y CV Propuesto Gestión es Cascada Entonces El CV Propuesto Proyecto es Espiral</i></p>
Nombre de la regla	REGLA R19

Tabla 6-107: Selección del Ciclo de vida para el Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si</p> <p>La Especificación de Requerimientos indica usar modelo en Espiral Y El Tipo de Aplicación indica usar modelo en Cascada Y La Gestión de Proyecto indica usar modelo en Espiral</p> <p>Entonces</p> <p>El Ciclo de Vida sugerido es Espiral</p>
Formulación externa de la regla	<p>Si</p> <p>CV Propuesto Requisitos es Espiral Y CV Propuesto Aplicación es Cascada Y CV Propuesto Gestión es Espiral</p> <p>Entonces</p> <p>El CV Propuesto Proyecto es Espiral</p>
Nombre de la regla	REGLA R20

Tabla 6-108: Selección del Ciclo de vida para el Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si</p> <p>La Especificación de Requerimientos indica usar modelo en Cascada Y El Tipo de Aplicación indica usar modelo en Espiral Y La Gestión de Proyecto indica usar modelo en Espiral</p> <p>Entonces</p> <p>El Ciclo de Vida sugerido es Espiral</p>
Formulación externa de la regla	<p>Si</p> <p>CV Propuesto Requisitos es Cascada Y CV Propuesto Aplicación es Espiral Y CV Propuesto Gestión es Espiral</p> <p>Entonces</p> <p>El CV Propuesto Proyecto es Espiral</p>
Nombre de la regla	REGLA R22

Tabla 6-109: Selección del Ciclo de vida para el Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si</p> <p>La Especificación de Requerimientos indica usar modelo en Espiral Y El Tipo de Aplicación indica usar modelo en Espiral Y La Gestión de Proyecto indica usar modelo de Objetos</p> <p>Entonces</p> <p>El Ciclo de Vida sugerido es Espiral</p>
Formulación externa de la regla	<p>Si</p> <p>CV Propuesto Requisitos es Espiral Y CV Propuesto Aplicación es Espiral Y CV Propuesto Gestión es Objetos</p> <p>Entonces</p> <p>El CV Propuesto Proyecto es Espiral</p>
Nombre de la regla	REGLA R23

Tabla 6-110: Selección del Ciclo de vida para el Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>La Especificación de Requerimientos indica usar modelo en Espiral Y El Tipo de Aplicación indica usar modelo de Objetos Y La Gestión de Proyecto indica usar modelo en Espiral Entonces El Ciclo de Vida sugerido es Espiral</i></p>
Formulación externa de la regla	<p>Si <i>CV Propuesto Requisitos es Espiral Y CV Propuesto Aplicación es Objetos Y CV Propuesto Gestión es Espiral Entonces El CV Propuesto Proyecto es Espiral</i></p>
Nombre de la regla	REGLA R24

Tabla 6-111: Selección del Ciclo de vida para el Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>La Especificación de Requerimientos indica usar modelo de Objetos Y El Tipo de Aplicación indica usar modelo en Espiral Y La Gestión de Proyecto indica usar modelo en Espiral Entonces El Ciclo de Vida sugerido es Espiral</i></p>
Formulación externa de la regla	<p>Si <i>CV Propuesto Requisitos es Objetos Y CV Propuesto Aplicación es Espiral Y CV Propuesto Gestión es Espiral Entonces El CV Propuesto Proyecto es Espiral</i></p>
Nombre de la regla	REGLA R26

Tabla 6-112: Selección del Ciclo de vida para el Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>La Especificación de Requerimientos indica usar modelo en Cascada Y El Tipo de Aplicación indica usar modelo en Espiral Y La Gestión de Proyecto indica usar modelo en Objetos Entonces El Ciclo de Vida sugerido es Espiral</i></p>
Formulación externa de la regla	<p>Si <i>CV Propuesto Requisitos es Cascada Y CV Propuesto Aplicación es Espiral Y CV Propuesto Gestión es Objetos Entonces El CV Propuesto Proyecto es Espiral</i></p>
Nombre de la regla	REGLA R27

Tabla 6-113: Selección del Ciclo de vida para el Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>La Especificación de Requerimientos indica usar modelo en Cascada Y El Tipo de Aplicación indica usar modelo en Objetos Y La Gestión de Proyecto indica usar modelo en Espiral Entonces El Ciclo de Vida sugerido es Objetos</i></p>
Formulación externa de la regla	<p>Si <i>CV Propuesto Requisitos es Cascada Y CV Propuesto Aplicación es Objetos Y CV Propuesto Gestión es Espiral Entonces El CV Propuesto Proyecto es Objetos</i></p>
Nombre de la regla	REGLA R28

Tabla 6-114: Selección del Ciclo de vida para el Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>La Especificación de Requerimientos indica usar modelo en Objetos Y El Tipo de Aplicación indica usar modelo en Cascada Y La Gestión de Proyecto indica usar modelo en Espiral Entonces El Ciclo de Vida sugerido es Espiral</i></p>
Formulación externa de la regla	<p>Si <i>CV Propuesto Requisitos es Objetos Y CV Propuesto Aplicación es Cascada Y CV Propuesto Gestión es Espiral Entonces El CV Propuesto Proyecto es Espiral</i></p>
Nombre de la regla	REGLA R29

Tabla 6-115: Selección del Ciclo de vida para el Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>La Especificación de Requerimientos indica usar modelo en Objetos Y El Tipo de Aplicación indica usar modelo de Espiral Y La Gestión de Proyecto indica usar modelo en Cascada Entonces El Ciclo de Vida sugerido es Espiral</i></p>
Formulación externa de la regla	<p>Si <i>CV Propuesto Requisitos es Objetos Y CV Propuesto Aplicación es Espiral Y CV Propuesto Gestión es Cascada Entonces El CV Propuesto Proyecto es Espiral</i></p>
Nombre de la regla	REGLA R30

Tabla 6-116: Selección del Ciclo de vida para el Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>La Especificación de Requerimientos indica usar modelo en Espiral Y El Tipo de Aplicación indica usar modelo en Cascada Y La Gestión de Proyecto indica usar modelo de Objetos Entonces El Ciclo de Vida sugerido es Objetos</i></p>
Formulación externa de la regla	<p>Si <i>CV Propuesto Requisitos es Espiral Y CV Propuesto Aplicación es Cascada Y CV Propuesto Gestión es Objetos Entonces El CV Propuesto Proyecto es Objetos</i></p>
Nombre de la regla	REGLA R31

Tabla 6-117: Selección del Ciclo de vida para el Proyecto

Estado de la regla	Texto de la regla
Palabras del experto	<p>Si <i>La Especificación de Requerimientos indica usar modelo de Espiral Y El Tipo de Aplicación indica usar modelo de Objetos Y La Gestión de Proyecto indica usar modelo en Cascada Entonces El Ciclo de Vida sugerido es Objetos</i></p>
Formulación externa de la regla	<p>Si <i>CV Propuesto Requisitos es Espiral Y CV Propuesto Aplicación es Objetos Y CV Propuesto Gestión es Cascada Entonces El CV Propuesto Proyecto es Objetos</i></p>
Nombre de la regla	REGLA R32

Tabla 6-118: Selección del Ciclo de vida para el Proyecto

6.2.4.2 COMPROBACION DE LOS CONOCIMIENTOS TACTICOS

El experto ha completado, validado y revisado todas las seudorreglas, considerando los casos de prueba descriptos en el Capítulo 9 y se ha comprobado si el accionar del experto coincidía con las representaciones de los conocimientos tácticos. Esto permitió ajustar las seudorreglas que se relacionaban con el análisis del riesgo.

6.2.5 PASO 5: IDENTIFICACIÓN DE LOS CONOCIMIENTOS FACTICOS

Se completa la identificación de los conocimientos fácticos iniciada en el paso 1 y 2 (identificación, comparación, categorización y relaciones entre los conceptos). Los conocimientos fácticos del experto contienen información que el SE conocerá “a priori” acerca del área de aplicación y la información que el sistema obtendrá acerca del caso específico al ejecutar la tarea. En este paso se organiza la información recopilada acerca de cada atributo general de acuerdo al formato estándar propuesto por Gómez [Gómez, A y otros 1997].

6.2.5.1 ANALISIS DE LOS CONOCIMIENTOS FACTICOS

En el presente trabajo los atributos pueden clasificarse en tres grupos:

- Los valores ingresados por el usuario del SE, que identifican al proyecto analizado y no participan en el proceso de razonamiento.
- Los valores ingresados por el usuario del SE, a partir de su conocimiento respecto del proyecto informático a desarrollar.
- Los valores deducidos a partir de los valores ingresados por el usuario del sistema.

Estos atributos se describen en las tablas 6-119 a 6-192 que se presentan a continuación.

Información	Descripción
Nombre	<i>Identificación</i>
Concepto	<i>Proyecto</i>
Descripción	<i>Identifica al proyecto que se esta analizando.</i>
Tipo Valor	<i>Código Univoco</i>
Rango de valores	<i>Valor alfanumérico</i>
Nro. Valores por caso	<i>Mínimo 1</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>No hay método específico ya que es nombre de un archivo que guarda datos identificatorios del proceso.</i>
Confiabilidad de los datos de entrada	<i>El sistema verifica que no puedan ingresarse valores ya existentes en el directorio seleccionado.</i>
Uso	<i>Permite identificar el proyecto analizado.</i>
Formato de los resultados de salida	<i>Valor alfanumérico. No actualiza ningún atributo, no afecta el proceso de selección del CV.</i>
Material de soporte	—

Tabla 6-119: Descripción del atributo Identificación

Información	Descripción
Nombre	<i>Nombre del proyecto</i>
Concepto	<i>Proyecto</i>
Descripción	<i>Describe el nombre del proyecto.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Valores alfanuméricos</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>No hay método.</i>
Confiabilidad de los datos de entrada	—
Uso	<i>Identificar al proyecto por nombre.</i>
Formato de los resultados de salida	<i>Valor alfanumérico. No actualiza ningún atributo, no afecta el proceso de selección del CV.</i>
Material de soporte	—

Tabla 6-120: Descripción del atributo Nombre del Proyecto

Información	Descripción
Nombre	<i>Líder del proyecto</i>
Concepto	<i>Proyecto</i>
Descripción	<i>Nombre y apellido del líder del proyecto que se está analizando.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Valores alfanuméricos</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>No hay método.</i>
Confiabilidad de los datos de entrada	_____
Uso	<i>Identificar al líder del proyecto por nombre.</i>
Formato de los resultados de salida	<i>No actualiza atributos, no afecta el proceso de selección del CV.</i>
Material de soporte	_____

Tabla 6-121: Descripción del atributo Líder del Proyecto

Información	Descripción
Nombre	<i>Fecha inicio</i>
Concepto	<i>Proyecto</i>
Descripción	<i>Identifica la fecha en la que se inició el proyecto.</i>
Tipo Valor	<i>Numérico: DD/MM/AAAA</i>
Rango de valores	<i>DD 1-31; MM: 1-12; AAAA 1900 en adelante.</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>No hay método.</i>
Confiabilidad de los datos de entrada	<i>La fecha no puede ser mayor a la fecha del día.</i>
Uso	<i>Permite conocer la fecha de inicio del proyecto.</i>
Formato de los resultados de salida	<i>No actualiza atributos, no afecta el proceso de selección del CV.</i>
Material de soporte	_____

Tabla 6-122: Descripción del atributo Fecha inicio

Información	Descripción
Nombre	<i>Fecha finalización</i>
Concepto	<i>Proyecto</i>
Descripción	<i>Es la fecha en la que se terminó o se estima terminar el proyecto.</i>
Tipo Valor	<i>Numérico: DD/MM/AAAA</i>
Rango de valores	<i>DD 1-31; MM: 1-12; AAAA 1900 en adelante.</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>No hay método.</i>
Confiabilidad de los datos de entrada	<i>La fecha debe ser mayor a la fecha de entrada.</i>
Uso	<i>Permite conocer la fecha de finalización del proyecto.</i>
Formato de los resultados de salida	<i>No actualiza atributos, no afecta el proceso de selección del CV.</i>
Material de soporte	_____

Tabla 6-123: Descripción del atributo Fecha finalización

Información	Descripción
Nombre	<i>Objetivo</i>
Concepto	<i>Proyecto</i>
Descripción	<i>Describe el objetivo del proyecto desarrollar</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Valores alfanuméricos</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>No hay método.</i>
Confiabilidad de los datos de entrada	—
Uso	<i>Permite conocer el objetivo del proyecto a desarrollar.</i>
Formato de los resultados de salida	<i>No actualiza atributos, no afecta el proceso de selección del CV.</i>
Material de soporte	—

Tabla 6-124: Descripción del atributo Objetivo

Información	Descripción
Nombre	<i>CV propuesto por SE</i>
Concepto	<i>Proyecto</i>
Descripción	<i>Identifica el ciclo de vida propuesto por el Sistema Experto para el proyecto a desarrollar</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Cascada, Objetos, Espiral, No hay propuesta</i>
Nro. Valores por caso	<i>Mínimo 1</i>
Fuente	<i>Obtenido a partir de CV Propuesto Requisitos, CV Propuesto Aplicación, CV Propuesto Gestión</i>
Detalles acerca del método para obtener esta información	<i>De acuerdo con los de CV Propuesto Requisitos, CV Propuesto Aplicación, CV Propuesto Gestión se estima el ciclo de vida más adecuado para el Proyecto</i>
Confiabilidad de los datos de entrada	—
Uso	<i>Permite conocer el ciclo de vida más adecuado para las características del proyecto analizado propuesto por el SE.</i>
Formato de los resultados de salida	<i>Texto. Se actualiza cuando se actualiza el atributo CV Propuesto Proyecto. Se graba en archivo del proyecto. No afecta el proceso de selección del CV.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-125: Descripción del atributo CV propuesto por SE

Información	Descripción
Nombre	<i>CV seleccionado</i>
Concepto	<i>Proyecto</i>
Descripción	<i>Identifica el ciclo de vida seleccionado por el Líder del proyecto a desarrollar</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Cascada, Objetos, Espiral, Otros</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>No hay método.</i>
Confiabilidad de los datos de entrada	—
Uso	<i>Permite conocer el CV seleccionado por el líder del proyecto</i>
Formato de los resultados de salida	<i>No actualiza atributos, no afecta el proceso de selección del CV.</i>
Material de soporte	—

Tabla 6-126: Descripción del atributo CV seleccionado

Información	Descripción
Nombre	<i>Tipo de Definición</i>
Concepto	<i>Requisitos</i>
Descripción	<i>Describe la forma en la que se han definido los requisitos del proyecto a desarrollar</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Formalmente, Informalmente, Exhaustivamente, Incompleto, Uniformemente, Desestructuradamente</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 3</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos del proyecto a desarrollar y de las primeras entrevistas con los usuarios. Se analiza el grado de detalle, si existen omisiones, si están documentados con alguna metodología , si están descriptos en forma estructurada (ordenada, pareja, semejante) respectivamente.</i>
Confiabilidad de los datos de entrada	<i>El sistema experto verifica que no puedan ingresarse valores opuestos como formalmente e informalmente al mismo tiempo.</i>
Uso	<i>Permite determinar si los requisitos y los límites del sistema están claramente definidos</i>
Formato de los resultados de salida	<i>Texto. Actualiza los atributos Definición Requisitos y Definición Limites</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-127: Descripción del atributo Tipo de Definición

Información	Descripción
Nombre	<i>Grado de Certidumbre</i>
Concepto	<i>Requisitos</i>
Descripción	<i>Describe el nivel de certidumbre en el que se han definido los requisitos al inicio del proyecto a desarrollar</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Mayoritariamente definidos, Parcialmente definidos</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos del proyecto a desarrollar y de las primeras entrevistas con los usuarios. Se consideran mayoritariamente definidos si han sido definidos el 90 % o más de los requerimientos y parcialmente definidos si se han definido menos del 90 % al inicio del sistema para ambos casos.</i>
Confiabilidad de los datos de entrada	<i>El sistema experto verifica que sólo pueda ingresarse uno de los valores, ya que son opuestos.</i>
Uso	<i>Permite determinar junto con otros atributos la selección del ciclo de vida para el área Especificación de Requerimientos.</i>
Formato de los resultados de salida	<i>Texto. Actualiza, en combinación con los atributos Definición Requisitos y Definición Límites, el atributo CV Propuesto Requisitos</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-128 Descripción del atributo Grado de Certidumbre

Información	Descripción
Nombre	<i>Grado de Cumplimiento</i>
Concepto	<i>Requisitos</i>
Descripción	<i>Describe el nivel de exigencia en el cumplimiento de los requerimientos en el producto final, que espera el usuario que contrata el desarrollo de la aplicación.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Usuario muy exigente, Usuario poco exigente</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos del proyecto a desarrollar y de las primeras entrevistas con los usuarios.</i>
Confiabilidad de los datos de entrada	<i>El sistema experto verifica que sólo pueda ingresarse uno de los valores, ya que son opuestos.</i>
Uso	<i>Permite determinar la selección del ciclo de vida para el área Especificación de Requerimientos.</i>
Formato de los resultados de salida	<i>Texto. Actualiza el atributo CV Propuesto Requisitos.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-129 Descripción del atributo Grado de Cumplimiento

Información	Descripción
Nombre	<i>Definición Requisitos</i>
Concepto	<i>Requisitos</i>
Descripción	<i>Describe el nivel de certidumbre en la definición de los requisitos al inicio del proyecto.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Definición clara, Definición incierta, Alta incertidumbre, Baja incertidumbre</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Obtenido a partir del Tipo de Definición</i>
Detalles acerca del método para obtener esta información	<i>De acuerdo con los valores ingresados por el usuario en Tipo de Definición se estima si la definición es clara, incierta, alta o baja.</i>
Confiabilidad de los datos de entrada	<i>—</i>
Uso	<i>Permite determinar la selección del ciclo de vida para el área Especificación de Requerimientos.</i>
Formato de los resultados de salida	<i>Texto. Actualiza el atributo CV Propuesto Requisitos.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-130 Descripción del atributo Definición Requisitos

Información	Descripción
Nombre	<i>Definición Límites</i>
Concepto	<i>Requisitos</i>
Descripción	<i>Describe el nivel de certidumbre en la definición de los límites o contexto al inicio del proyecto..</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Definición clara, Definición incierta, Alta incertidumbre, Baja incertidumbre</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Obtenido a partir del Tipo de Definición</i>
Detalles acerca del método para obtener esta información	<i>De acuerdo con los valores ingresados por el usuario en Tipo de Definición se estima si la definición es clara, incierta, alta o baja</i>
Confiabilidad de los datos de entrada	<i>—</i>
Uso	<i>Permite determinar la selección del ciclo de vida para el área Especificación de Requerimientos.</i>
Formato de los resultados de salida	<i>Texto. Actualiza el atributo CV Propuesto Requisitos.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-131 Descripción del atributo Definición Límites

Información	Descripción
Nombre	<i>Niveles de Composición</i>
Concepto	<i>Aplicación</i>
Descripción	<i>Describe el nivel de composición necesario del sistema en subsistemas para alcanzar una mejor comprensión y manipulación del sistema a desarrollar.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Pocos subsistemas, Varios subsistemas</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos del proyecto a desarrollar y de las primeras entrevistas con los usuarios.</i>
Confiabilidad de los datos de entrada	<i>El sistema experto verifica que sólo pueda ingresarse uno de los valores, ya que son opuestos.</i>
Uso	<i>Permite determinar si los subsistemas que componen el sistema son de baja o alta complejidad.</i>
Formato de los resultados de salida	<i>Texto. Actualiza el atributo Complejidad subsistemas.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-132 Descripción del atributo Niveles de Composición

Información	Descripción
Nombre	<i>Complejidad Subsistemas</i>
Concepto	<i>Aplicación</i>
Descripción	<i>Describe el nivel de complejidad de los subsistemas componentes de la aplicación a desarrollar.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Baja, Alta</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Obtenido a partir de Niveles de Composición</i>
Detalles acerca del método para obtener esta información	<i>De acuerdo con los valores ingresados por el usuario en Niveles de Composición se estima si la complejidad es alta o baja.</i>
Confiabilidad de los datos de entrada	<i>—</i>
Uso	<i>Permite determinar junto con otros atributos (Progresión Fases Desarrollo Retroalimentación Fases Desarrollo, Componentes Predominantes y Comportamiento Predominante) si el modelo en cascada es adecuado para el área Tipo de Aplicación.</i>
Formato de los resultados de salida	<i>Texto. Participa en la actualización del atributo CV Propuesto Aplicación.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-133 Descripción del atributo Complejidad Subsistemas

Información	Descripción
Nombre	<i>Componentes Predominantes</i>
Concepto	<i>Aplicación</i>
Descripción	<i>Describe el componente predominante en la aplicación a desarrollar.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Algorítmicos, Matemáticos, Gráficos</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 3</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos del proyecto a desarrollar y de las primeras entrevistas con los usuarios.</i>
Confiabilidad de los datos de entrada	<i>No son opciones excluyentes.</i>
Uso	<i>Permite determinar si las técnicas de modelado orientadas a objetos son adecuadas para esos componentes. Permite determinar junto con otros atributos (Complejidad subsistemas, Progresión Fases Desarrollo Retroalimentación Fases Desarrollo) si el modelo en cascada es adecuado para el área Tipo de Aplicación.</i>
Formato de los resultados de salida	<i>Texto. Actualiza el atributo Modelado Objetos y participa en la actualización del atributo CV Propuesto Aplicación.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-134 Descripción del atributo Componentes Predominantes

Información	Descripción
Nombre	<i>Comportamiento Predominante</i>
Concepto	<i>Aplicación</i>
Descripción	<i>Describe el comportamiento predominante en la aplicación a desarrollar.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Batch, Dinámico, Interactivo, Tiempo Real</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos del proyecto a desarrollar y de las primeras entrevistas con los usuarios. Se considera Batch si la modalidad de procesamiento predominante es diferida y dinámico si no lo es. La modalidad es interactiva cuando hay dialogo entre el usuario y la aplicación, y si el tiempo de procesamiento es interactivo dentro de un ajustado y limitado margen de tiempo para producir las respuestas, se considera tiempo real.</i>
Confiabilidad de los datos de entrada	<i>No son opciones excluyentes, pero es conveniente que el usuario seleccione una de ellas como predominante. La opción Batch excluye a las otras tres y viceversa.</i>
Uso	<i>Permite determinar si las técnicas de modelado orientadas a objetos son adecuadas para ese comportamiento predominante. Permite determinar junto con otros atributos (Complejidad subsistemas, Progresión Fases Desarrollo Retroalimentación Fases Desarrollo) si el modelo en cascada es adecuado para el área Tipo de Aplicación.</i>
Formato de los resultados de salida	<i>Texto. Actualiza el atributo Modelado Objetos y participa en la actualización del atributo CV Propuesto Aplicación.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-135 Descripción del atributo Comportamiento Predominante

Información	Descripción
Nombre	<i>Relación Fases Desarrollo</i>
Concepto	<i>Aplicación</i>
Descripción	<i>Describe el grado de dependencia entre las fases de desarrollo del proyecto.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Mucha Dependencia, Poca Dependencia, Independencia</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos y de las primeras entrevistas con los usuarios.</i>
Confiabilidad de los datos de entrada	<i>El sistema experto verifica que sólo pueda ingresarse uno de los valores, ya que son excluyentes.</i>
Uso	<i>Permite determinar si la retroalimentación entre las fases de desarrollo del proyecto es alta o baja.</i>
Formato de los resultados de salida	<i>Texto. Actualiza el atributo Retroalimentación Fases Desarrollo.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-136 Descripción del atributo Relación Fases desarrollo

Información	Descripción
Nombre	<i>Retroalimentación Fases Desarrollo</i>
Concepto	<i>Aplicación</i>
Descripción	<i>Describe el grado de relación de retroalimentación que se estima entre las fases de desarrollo del proyecto.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Alta, Baja</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Obtenido a partir de Relación Fases Desarrollo</i>
Detalles acerca del método para obtener esta información	<i>De acuerdo con los valores ingresados por el usuario en Relación Fases Desarrollo se estima si la retroalimentación es alta o baja.</i>
Confiabilidad de los datos de entrada	—
Uso	<i>Permite determinar si el Modelado en Prototipo es Conveniente. Permite determinar junto con otros atributos (Progresión Fases Desarrollo, Complejidad Subsistemas, Componentes Predominantes y Comportamiento Predominante) si el modelo en cascada es adecuado para el área Tipo de Aplicación.</i>
Formato de los resultados de salida	<i>Texto. Participa en la actualización de los atributos, Modelado Prototipo y CV Propuesto Aplicación.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-137 Descripción del atributo Retroalimentación Fases desarrollo

Información	Descripción
Nombre	Orientación
Concepto	Aplicación
Descripción	Describe la orientación con la que se va a modelar y formalizar la aplicación a desarrollar.
Tipo Valor	Texto
Rango de valores	Software de base, Sistema Basado en Conocimiento, Otros
Nro. Valores por caso	Mínimo 0, Máximo 1
Fuente	Ingresada por el usuario
Detalles acerca del método para obtener esta información	Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos del proyecto a desarrollar y de las primeras entrevistas con los usuarios. Según el problema pueda resolverse por el desarrollo de métodos convencionales de tipo algorítmicos o sean problemas que requieren de heurísticas para alcanzar la solución se analiza si es un sistema basado en conocimiento. El sistema puede ser una aplicación que esta orientada a dar un servicio básico (como procesador de textos, prototipador, sistema operativo) una aplicación directa.
Confiabilidad de los datos de entrada	Son opciones excluyentes.
Uso	Permite determinar si las técnicas de modelado orientadas a objetos o de prototipación son adecuadas para esa orientación. Permite determinar junto con otros atributos (Complejidad subsistemas, Progresión Fases Desarrollo Retroalimentación Fases Desarrollo) si el modelo en cascada es adecuado para el área Tipo de Aplicación.
Formato de los resultados de salida	Texto. Actualiza los atributos Modelado Objetos y Modelado Prototipo y participa en la actualización del atributo CV Propuesto Aplicación.
Material de soporte	Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII

Tabla 6-138 Descripción del atributo Orientación

Información	Descripción
Nombre	Integración HW-SW
Concepto	Aplicación
Descripción	Describe el grado de integración entre los componentes de hardware y software que requiere la aplicación a desarrollar.
Tipo Valor	Texto
Rango de valores	Fuertemente, Medianamente, Levemente
Nro. Valores por caso	Mínimo 0, Máximo 1
Fuente	Ingresada por el usuario
Detalles acerca del método para obtener esta información	Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos y de las primeras entrevistas con los usuarios.
Confiabilidad de los datos de entrada	El sistema experto verifica que sólo pueda ingresarse uno de los valores, ya que son excluyentes.
Uso	Permite determinar junto con otros atributos (Modelado Objetos, Modelado Prototipo y Opcionalidad) si el modelo en espiral o el modelo de objetos es adecuado para el área Tipo de Aplicación.
Formato de los resultados de salida	Texto. Participa en la actualización del atributo CV Propuesto Aplicación.
Material de soporte	Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII

Tabla 6-139 Descripción del atributo Integración HW-SW

Información	Descripción
Nombre	<i>Progresión Fases Desarrollo</i>
Concepto	<i>Aplicación</i>
Descripción	<i>Describe el modo en el que se puede avanzar cumpliendo las etapas del ciclo de vida durante el desarrollo de la aplicación.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Uniforme y Secuencial, No uniforme y secuencial</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos y de las primeras entrevistas con los usuarios. Si el modo de avance es desparejo y con distintos ritmos, cumplida una etapa se puede volver a la anterior o continuar con otra sin seguir un ordenamiento entonces la progresión a través de las fases de desarrollo es No uniforme y secuencial. Si las fases se cumplen en una sucesión ordenada de modo que cada actividad determina la siguiente y el avance es gradual y parejo entonces la progresión se considera uniforme y secuencial.</i>
Confiabilidad de los datos de entrada	<i>El sistema experto verifica que sólo pueda ingresarse uno de los valores, ya que son excluyentes.</i>
Uso	<i>Permite determinar si las técnicas de prototipación son adecuadas para esa orientación. Permite determinar junto con otros atributos (Complejidad subsistemas, Progresión Fases Desarrollo Retroalimentación Fases Desarrollo) si el modelo en cascada es adecuado para el área Tipo de Aplicación.</i>
Formato de los resultados de salida	<i>Texto. Actualiza el atributo Modelado Prototipo y participa en la actualización del atributo CV Propuesto Aplicación.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-140 Descripción del atributo Progresión Fases desarrollo

Información	Descripción
Nombre	<i>Factores Diseño</i>
Concepto	<i>Aplicación</i>
Descripción	<i>Describe algunos problemas a considerar en etapas tempranas del desarrollo en los factores de diseño vinculados a la arquitectura y eficacia del sistema.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Problemas de Arquitectura, Problemas de Eficacia</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos y de las primeras entrevistas con los usuarios.</i>
Confiabilidad de los datos de entrada	<i>No se consideran valores excluyentes.</i>
Uso	<i>Permite determinar si las técnicas de prototipación son adecuadas para esa orientación.</i>
Formato de los resultados de salida	<i>Texto. Actualiza el atributo Modelado Prototipo.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-141 Descripción del atributo Factores Diseño

Información	Descripción
Nombre	<i>Existencia Aplicaciones</i>
Concepto	<i>Aplicación</i>
Descripción	<i>Describe si la aplicación a desarrollar es innovadora o ya existen aplicaciones similares.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Innovador, Conocido</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos y de las primeras entrevistas con los usuarios y de su propio conocimiento del mercado de productos de software</i>
Confiabilidad de los datos de entrada	<i>El sistema experto verifica que sólo pueda ingresarse uno de los valores, ya que son excluyentes.</i>
Uso	<i>Permite determinar si las técnicas de prototipación son adecuadas para esa orientación.</i>
Formato de los resultados de salida	<i>Texto. Actualiza el atributo Modelado Prototipo.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-142 Descripción del atributo Existencia Aplicaciones

Información	Descripción
Nombre	<i>Posibilidad Diseño</i>
Concepto	<i>Aplicación</i>
Descripción	<i>Describe las posibilidades de modelos formales o de diseño a considerar a partir de un mismo modelo de análisis.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Pocas opciones, Varias opciones</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos y de las primeras entrevistas con los usuarios y de su propia experiencia en el proceso de formalización. Si las posibilidades de desarrollo de modelos formales son tres o menos se considera que el valor del atributo puede ser pocas opciones, si las posibilidades son mas de tres entonces se consideran varias.</i>
Confiabilidad de los datos de entrada	<i>El sistema experto verifica que sólo pueda ingresarse uno de los valores, ya que son excluyentes.</i>
Uso	<i>Permite determinar si el sistema puede considerarse con optionalidades múltiples o simples.</i>
Formato de los resultados de salida	<i>Texto. Actualiza el atributo Opcionalidad.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-143 Descripción del atributo Posibilidad Diseño

Información	Descripción
Nombre	<i>Tipo Modificación</i>
Concepto	<i>Aplicación</i>
Descripción	<i>Describe las modificaciones que pueda tener la aplicación que se encuentren explícitamente prevista en los datos y los procesos.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Redefinición, Extensión, Ampliación, Emisión gradual</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 4</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<p><i>Valor estimado por el usuario del SE a partir del Informe de requerimientos, de las entrevistas con los usuarios y de su experiencia en el tipo de aplicación y el negocio.</i></p> <p><i>Se considera ampliación cuando por nuevos requerimientos se agregan funciones o procesos al sistema ya implementado. Se considera redefinición al cambio de los procedimientos de una o varias funciones principales ya implementadas en un sistema.</i></p> <p><i>Se considera extensión cuando es necesario dar mayor funcionalidad a los procesos existentes ya implementados. Se extienden, no aumentan las funciones sino algún aspecto de alguna de ellas.</i></p> <p><i>Emisión gradual es una nueva edición de una aplicación con modificaciones notables respecto de emisiones anteriores. Es una nueva versión de la aplicación. Abarca la aplicación en su totalidad.</i></p>
Confiabilidad de los datos de entrada	<i>No se consideran valores excluyentes.</i>
Uso	<i>Permite determinar si el sistema presenta alta variabilidad en los procesos y/o datos.</i>
Formato de los resultados de salida	<i>Texto. Actualiza el atributo Variabilidad Procesos</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-144 Descripción del atributo *Tipo Modificación*

Información	Descripción
Nombre	<i>Modelado Objetos</i>
Concepto	<i>Aplicación</i>
Descripción	<i>Describe si el modelado en objetos es aplicable en el desarrollo de la aplicación.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Conveniente, No conveniente</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Obtenido a partir de Orientación, Componentes Predominantes, Comportamiento Predominante.</i>
Detalles acerca del método para obtener esta información	<i>De acuerdo con los valores ingresados por el Orientación, Componentes Predominantes, Comportamiento Predominante se estima si el modelado por objetos es conveniente o no.</i>
Confiabilidad de los datos de entrada	—
Uso	<i>Permite determinar junto con otros atributos (Modelado Prototipo, Opcionalidad, Integración HW-SW) si el modelo en Objetos o Espiral es adecuado para el área Tipo de Aplicación.</i>
Formato de los resultados de salida	<i>Texto. Participa en la actualización del atributo CV Propuesto Aplicación.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-145 Descripción del atributo *Modelado Objetos*

Información	Descripción
Nombre	<i>Modelado Prototipo</i>
Concepto	<i>Aplicación</i>
Descripción	<i>Describe si el modelado en prototipo es aplicable en el desarrollo de la aplicación.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Conveniente, No conveniente</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Obtenido a partir de Complejidad Subsistemas, Retroalimentación Fases Desarrollo, Variabilidad Procesos, Progresión Fases Desarrollo, Existencia Aplicaciones, Factores Diseño, Orientación.</i>
Detalles acerca del método para obtener esta información	<i>De acuerdo con los valores ingresados por el usuario Complejidad Subsistemas, Retroalimentación Fases Desarrollo, Variabilidad Procesos, Progresión Fases Desarrollo, Existencia Aplicaciones, Factores Diseño, Orientación se estima si el modelado por prototipo es conveniente o no.</i>
Confiabilidad de los datos de entrada	—
Uso	<i>Permite determinar junto con otros atributos (Modelado Objetos, Opcionalidad, Integración HW-SW) si el modelo en Objetos o Espiral es adecuado para el área Tipo de Aplicación.</i>
Formato de los resultados de salida	<i>Texto. Participa en la actualización del atributo CV Propuesto Aplicación.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-146 Descripción del atributo Modelado Prototipo

Información	Descripción
Nombre	<i>Variabilidad Procesos</i>
Concepto	<i>Aplicación</i>
Descripción	<i>Describe el nivel de variabilidad esperado en los procesos de la aplicación a desarrollar.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Alta, Baja</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Obtenido a partir de Tipo Modificación</i>
Detalles acerca del método para obtener esta información	<i>De acuerdo con los valores ingresados por el usuario en Tipo Modificación se estima si la variabilidad es alta o baja.</i>
Confiabilidad de los datos de entrada	—
Uso	<i>Permite determinar junto con otros atributos (Complejidad Subsistemas, Retroalimentación Fases Desarrollo, Progresión Fases Desarrollo, Existencia aplicaciones y Factores Diseño) si el Modelado en Prototipo es Conveniente.</i>
Formato de los resultados de salida	<i>Texto. Participa en la actualización de los atributos, Modelado Prototipo.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-147 Descripción del atributo Variabilidad Procesos

Información	Descripción
Nombre	<i>Opcionalidad</i>
Concepto	<i>Aplicación</i>
Descripción	<i>Describe el tipo de opciones de diseño e implementación que se estima tiene la aplicación a desarrollar.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Múltiple, Simple</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Obtenido a partir de Posibilidad Diseño y Posibilidad Implementación.</i>
Detalles acerca del método para obtener esta información	<i>De acuerdo con los valores ingresados por el usuario en Posibilidad Diseño y Posibilidad Implementación se estima si la optionalidad es múltiple o simple.</i>
Confiabilidad de los datos de entrada	—
Uso	<i>Permite determinar junto con otros atributos (Modelado Objetos, modelado Prototipo, Integración HW-SW) si el modelo en Objetos o Espiral es adecuado para el área Tipo de Aplicación.</i>
Formato de los resultados de salida	<i>Texto. Participa en la actualización del atributo, CV Propuesto Aplicación.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-148 Descripción del atributo Opcionalidad

Información	Descripción
Nombre	<i>Posibilidad Implementación</i>
Concepto	<i>Aplicación</i>
Descripción	<i>Describe las posibilidades de modelos implementables computacionalmente a partir de un modelo de diseño y de un mismo modelo de análisis.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Pocas opciones, Varias opciones</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos y de las primeras entrevistas con los usuarios y de su propio conocimiento de proceso de codificación y formalización. Si las posibilidades de desarrollo de modelos programados son tres o menos se considera que el valor del atributo puede ser pocas opciones, si las posibilidades son mas de tres entonces se consideran varias.</i>
Confiabilidad de los datos de entrada	<i>El sistema experto verifica que sólo pueda ingresarse uno de los valores, ya que son excluyentes.</i>
Uso	<i>Permite determinar si el sistema puede considerarse con optionalidad es múltiples o simples.</i>
Formato de los resultados de salida	<i>Texto. Actualiza el atributo Opcionalidad.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-149 Descripción del atributo Posibilidad Implementación

Información	Descripción
Nombre	<i>Formalidad</i>
Concepto	<i>Coordinación</i>
Descripción	<i>Describe el grado de formalidad que se requiere para la gestión del proyecto a desarrollar.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Poco formal, Medianamente formal, Muy formal</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>Valor estimado por el usuario del SE a partir del Informe de requerimientos y de las primeras entrevistas con los usuarios y de su propio conocimiento del tipo de aplicación y del negocio.</i>
Confiabilidad de los datos de entrada	<i>El sistema experto verifica que sólo pueda ingresarse uno de los valores, ya que son excluyentes.</i>
Uso	<i>Permite determinar junto con otros atributos la aplicabilidad el modelo de objetos, espiral o cascada.</i>
Formato de los resultados de salida	<i>Texto. Actualiza junto con otros atributos el atributo Aplicabilidad OO, Aplicabilidad Espiral y Cv Propuesto Gestión.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-150 Descripción del atributo Formalidad

Información	Descripción
Nombre	<i>Entregas</i>
Concepto	<i>Coordinación</i>
Descripción	<i>Describe si es necesario gestionar la entrega de versiones tempranas o versiones parciales o graduales de la aplicación a desarrollar.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Versión temprana, no versión temprana, versión gradual, versión parcial</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos y de las primeras entrevistas con los usuarios y de su propio conocimiento de proceso del tipo de aplicación y del negocio.</i>
Confiabilidad de los datos de entrada	<i>El sistema experto verifica que sólo pueda ingresarse uno de los valores, ya que son excluyentes.</i>
Uso	<i>Permite determinar junto con otros atributos la aplicabilidad el modelo en cascada y la existencia de entregas intermedias.</i>
Formato de los resultados de salida	<i>Texto. Actualiza junto con otros atributos el atributo Entrega Intermedia y Cv Propuesto Gestión.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-151 Descripción del atributo Entregas

Información	Descripción
Nombre	<i>Responsabilidad</i>
Concepto	<i>Coordinación</i>
Descripción	<i>Describe quien tiene la responsabilidad del desarrollo de la aplicación. Es decir si el desarrollo se terceriza o no.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Terceros, Organización propia</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos y de las primeras entrevistas con los usuarios.</i>
Confiabilidad de los datos de entrada	<i>El sistema experto verifica que sólo pueda ingresarse uno de los valores, ya que son excluyentes.</i>
Uso	<i>Permite determinar junto con otros atributos la aplicabilidad el modelo de objetos, espiral o cascada.</i>
Formato de los resultados de salida	<i>Texto. Actualiza junto con otros atributos el atributo, Aplicabilidad Espiral y Cv Propuesto Gestión.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-152 Descripción del atributo Responsabilidad

Información	Descripción
Nombre	<i>Control gestión</i>
Concepto	<i>Coordinación</i>
Descripción	<i>Describe el grado de control que se requiere para la gestión del proyecto a desarrollar.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Muy ajustado, Medianamente ajustado, Poco ajustado</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos y de las primeras entrevistas con los usuarios y de su propio conocimiento de proceso del tipo de aplicación y del negocio.</i>
Confiabilidad de los datos de entrada	<i>El sistema experto verifica que sólo pueda ingresarse uno de los valores, ya que son excluyentes.</i>
Uso	<i>Permite determinar junto con otros atributos la aplicabilidad el modelo cascada.</i>
Formato de los resultados de salida	<i>Texto. Actualiza junto con otros atributos el atributo Cv Propuesto Gestión.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-153 Descripción del atributo Control gestión

Información	Descripción
Nombre	<i>Software para prototipar</i>
Concepto	<i>Coordinación</i>
Descripción	<i>Describe la disponibilidad de software para prototipar para ser usado en el proyecto a desarrollar.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>No disponible, Hay disponible, Factible de adquirir, No adquirible</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 2</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos y de las primeras entrevistas con los usuarios y de su propio conocimiento de proceso del tipo de aplicación y del negocio.</i>
Confiabilidad de los datos de entrada	<i>No se consideran excluyentes.</i>
Uso	<i>Permite determinar junto con otros atributos la factibilidad de usar prototipos en el desarrollo de la aplicación.</i>
Formato de los resultados de salida	<i>Texto. Actualiza el atributo Factibilidad Prototipo.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-154 Descripción del atributo Software para prototipar

Información	Descripción
Nombre	<i>Factibilidad Metodología</i>
Concepto	<i>Coordinación</i>
Descripción	<i>Describe el grado o nivel de prueba, de conocimiento general acerca de la metodología que es posible usar para el desarrollo de la aplicación.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Ampliamente probada, Medianamente probada, Poco probada</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos y de las primeras entrevistas con los usuarios y de su propio conocimiento de proceso del tipo de aplicación y del negocio.</i>
Confiabilidad de los datos de entrada	<i>El sistema experto verifica que sólo pueda ingresarse uno de los valores, ya que son excluyentes.</i>
Uso	<i>Permite determinar junto con otros atributos la conveniencia de usar metodologías según haya sido probada previamente en el desarrollo de otras aplicaciones..</i>
Formato de los resultados de salida	<i>Texto. Actualiza el atributo Conveniencia Metodología.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-155 Descripción del atributo Factibilidad Metodología

Información	Descripción
Nombre	<i>Necesidad Metodología</i>
Concepto	<i>Coordinación</i>
Descripción	<i>Describe el grado o nivel de prueba, de conocimiento general acerca de la metodología que es necesario usar para el desarrollo de la aplicación.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Ampliamente probada, Medianamente probada, Poco probada</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos y de las primeras entrevistas con los usuarios y de su propio conocimiento de proceso del tipo de aplicación y del negocio.</i>
Confiabilidad de los datos de entrada	<i>El sistema experto verifica que sólo pueda ingresarse uno de los valores, ya que son excluyentes.</i>
Uso	<i>Permite determinar junto con otros atributos la conveniencia de usar metodologías según haya sido probada previamente en el desarrollo de otras aplicaciones..</i>
Formato de los resultados de salida	<i>Texto. Actualiza el atributo Conveniencia Metodología.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-156 Descripción del atributo Necesidad Metodología

Información	Descripción
Nombre	<i>Reúso aplicación OO</i>
Concepto	<i>Coordinación</i>
Descripción	<i>Describe si el sistema a desarrollar esta relacionado con algún sistema previo desarrollado según el paradigma de orientación a objetos.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Subsistema del existente, Ampliación del existente, Modificación del existente, Componentes del actual</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 4</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos y de las primeras entrevistas con los usuarios y de su propio conocimiento de proceso del tipo de aplicación y del negocio.</i>
Confiabilidad de los datos de entrada	<i>No se consideran excluyentes.</i>
Uso	<i>Permite determinar junto con otros atributos si existe un sistema anterior desarrollado en objetos que se encuentre relacionado con el desarrollo del proyecto actual.</i>
Formato de los resultados de salida	<i>Texto. Actualiza el atributo Sistema OO.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-157 Descripción del atributo Reúso aplicación OO

Información	Descripción
Nombre	<i>Reúso aplicación existente</i>
Concepto	<i>Coordinación</i>
Descripción	<i>Describe si en los requerimientos se definen explícitamente estrategias para el reúso del software existente.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Estrategias explícitas, No estrategias explícitas</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos y de las primeras entrevistas con los usuarios y de su propio conocimiento de proceso del tipo de aplicación y del negocio.</i>
Confiabilidad de los datos de entrada	<i>El sistema experto verifica que sólo pueda ingresarse uno de los valores, ya que son excluyentes.</i>
Uso	<i>Permite determinar junto con otros atributos la aplicabilidad del modelo de Objetos y la aplicabilidad del prototipo.</i>
Formato de los resultados de salida	<i>Texto. Actualiza el atributo Aplicabilidad OO y Aplicabilidad Prototipo.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-158 Descripción del atributo Reúso aplicación existente

Información	Descripción
Nombre	<i>Reúso aplicación futura</i>
Concepto	<i>Coordinación</i>
Descripción	<i>Describe si el grado de necesidad existente de reutilizar componentes del desarrollo de la aplicación actual en proyectos subsiguientes.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Muy necesario, Medianamente necesario, Poco necesario</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos y de las primeras entrevistas con los usuarios y de su propio conocimiento de proceso del tipo de aplicación y del negocio.</i>
Confiabilidad de los datos de entrada	<i>El sistema experto verifica que sólo pueda ingresarse uno de los valores, ya que son excluyentes.</i>
Uso	<i>Permite determinar junto con otros atributos la si existe un sistema desarrollado bajo el paradigma de Objetos que afecte o condicione desarrollos subsiguientes.</i>
Formato de los resultados de salida	<i>Texto. Actualiza el atributo Sistema OO.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-159 Descripción del atributo Reúso aplicación futura

Información	Descripción
Nombre	<i>Procedimientos Cambios</i>
Concepto	<i>Coordinación</i>
Descripción	<i>Describe si en los requerimientos se requiere de acuerdos confirmados para gestionar cambios durante el desarrollo de la aplicación. Si se requiere de un procedimiento de control de cambios en la gestión de configuración.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Acuerdos confirmados, No acuerdos confirmados</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos y de las primeras entrevistas con los usuarios y de su propio conocimiento de proceso del tipo de aplicación y del negocio.</i>
Confiabilidad de los datos de entrada	<i>El sistema experto verifica que sólo pueda ingresarse uno de los valores, ya que son excluyentes.</i>
Uso	<i>Permite determinar junto con otros atributos la aplicabilidad del modelo en espiral .</i>
Formato de los resultados de salida	<i>Texto. Actualiza el atributo Aplicabilidad Espiral.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-160 Descripción del atributo Procedimientos Cambios

Información	Descripción
Nombre	<i>Procedimientos Desarrollo-Mantenimiento</i>
Concepto	<i>Coordinación</i>
Descripción	<i>Describe la factibilidad y necesidad de usar los mismos procedimientos para el desarrollo y mantenimiento de la aplicación a desarrollar.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Es factible usar los mismos, Es factible usar diferentes, Es necesario usar los mismos, Es necesario usar diferentes</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 2</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos y de las primeras entrevistas con los usuarios y de su propio conocimiento de proceso del tipo de aplicación y del negocio.</i>
Confiabilidad de los datos de entrada	<i>No se consideran excluyentes.</i>
Uso	<i>Permite determinar junto con otros atributos la conveniencia de usar los mismos procedimientos para el desarrollo y mantenimiento de la aplicación a desarrollar.</i>
Formato de los resultados de salida	<i>Texto. Actualiza el atributo Procedimientos D-M.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-161 Descripción del atributo Procedimientos Desarrollo-Mantenimiento

Información	Descripción
Nombre	<i>Conveniencia Metodología</i>
Concepto	<i>Coordinación</i>
Descripción	<i>Describe de acuerdo con el grado de factibilidad y necesidad que tipo de metodología es conveniente considerar en la gestión del proyecto de la aplicación a desarrollar.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Ampliamente probada, Medianamente probada, Poco probada</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Obtenido a partir de Factibilidad Metodología y Necesidad Metodología</i>
Detalles acerca del método para obtener esta información	<i>De acuerdo con los valores ingresados por el usuario en Factibilidad Metodología y Necesidad Metodología se estima la conveniencia de la metodología a aplicar</i>
Confiabilidad de los datos de entrada	<i>—</i>
Uso	<i>Permite determinar junto con otros atributos la aplicabilidad de la Orientación a Objetos, de la Espiral y/o del modelo en Cascada para el área Gestión de Proyectos.</i>
Formato de los resultados de salida	<i>Texto. Participa en la actualización de los atributos, Aplicabilidad OO, Aplicabilidad Espiral y CV Propuesto Gestión.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-162 Descripción del atributo Conveniencia Metodología

Información	Descripción
Nombre	<i>Factibilidad Prototipo</i>
Concepto	<i>Coordinación</i>
Descripción	<i>Describe de acuerdo con disponibilidad o posibilidad de adquirir software para prototipar el grado de factibilidad de usar prototipos para el desarrollo de la aplicación</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Factible, No factible</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Obtenido a partir de Software para Prototipar</i>
Detalles acerca del método para obtener esta información	<i>De acuerdo con los valores ingresados por el usuario en Software para Prototipar se estima la factibilidad de usar prototipos</i>
Confiabilidad de los datos de entrada	<i>—</i>
Uso	<i>Permite determinar junto con otros atributos la aplicabilidad de la del modelo en Cascada o del modelo en Espiral para el área Gestión de Proyectos.</i>
Formato de los resultados de salida	<i>Texto. Participa en la actualización del atributo CV Propuesto Gestión.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-163 Descripción del atributo Factibilidad Prototipo

Información	Descripción
Nombre	<i>Sistema OO</i>
Concepto	<i>Coordinación</i>
Descripción	<i>Describe si existe algún sistema anterior o esta prevista uno posterior orientado a objetos</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Existe, No existe</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Obtenido a partir de Reúso Aplicación OO y Reúso Aplicación Futura.</i>
Detalles acerca del método para obtener esta información	<i>De acuerdo con los valores ingresados por el usuario en Reúso Aplicación OO y Reúso Aplicación Futura si existe o esta previsto un sistema orientado a objetos</i>
Confiabilidad de los datos de entrada	—
Uso	<i>Permite determinar junto con otros atributos la aplicabilidad del Modelo de Objetos para el área Gestión de Proyectos.</i>
Formato de los resultados de salida	<i>Texto. Participa en la actualización del atributo CV Propuesto Gestión.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-164 Descripción del atributo Sistema OO

Información	Descripción
Nombre	<i>Entrega Intermedia</i>
Concepto	<i>Coordinación</i>
Descripción	<i>Describe el acuerdo con el tipo de entregas que espera el usuario si existe o no la necesidad de entregas intermedias de la aplicación a desarrollar.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Existe, No existe</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Obtenido a partir de Entregas</i>
Detalles acerca del método para obtener esta información	<i>De acuerdo con los valores ingresados por el usuario en Entregas se estima la necesidad de entregas intermedias</i>
Confiabilidad de los datos de entrada	—
Uso	<i>Permite determinar junto con otros atributos la aplicabilidad del Prototipo, de la Orientación a Objetos, y/o del modelo en Cascada para el área Gestión de Proyectos.</i>
Formato de los resultados de salida	<i>Texto. Participa en la actualización de los atributos, Aplicabilidad OO, Aplicabilidad Prototipo y CV Propuesto Gestión.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-165 Descripción del atributo Entrega Intermedia

Información	Descripción
Nombre	<i>Aplicabilidad Espiral</i>
Concepto	<i>Coordinación</i>
Descripción	<i>Describe la posibilidad de aplicar la modelización en Espiral en la aplicación a desarrollar.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Aplicable, No aplicable</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Obtenido a partir de Terminación Proyectos, Factor Riesgo, Objetivo Calidad, Análisis Riesgo, Responsabilidad, procedimientos Cambios, formalidad, Habilidad Riesgo, Procedimientos D-M, Riesgos Alternativa, Conveniencia Metodología e Identificación Alternativas</i>
Detalles acerca del método para obtener esta información	<i>De acuerdo con los valores ingresados por el usuario en Terminación Proyectos, Factor Riesgo, Objetivo Calidad, Análisis Riesgo, Responsabilidad, procedimientos Cambios, formalidad, Habilidad Riesgo, Procedimientos D-M, Riesgos Alternativa, Conveniencia Metodología e Identificación Alternativas se estima la aplicabilidad de la modelización en Espiral</i>
Confiabilidad de los datos de entrada	—
Uso	<i>Permite determinar junto con otros atributos la aplicabilidad del modelo en Espiral para el área Gestión de Proyectos.</i>
Formato de los resultados de salida	<i>Texto. Participa en la actualización del atributo CV Propuesto Gestión.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-166 Descripción del atributo Aplicabilidad Espiral

Información	Descripción
Nombre	<i>Aplicabilidad Prototipo</i>
Concepto	<i>Coordinación</i>
Descripción	<i>Describe la posibilidad de aplicar la modelización de prototipos en la aplicación a desarrollar.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Aplicable, No aplicable</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Obtenido a partir de Entrega Intermedia, Experiencia previa, Participación, Experiencia Técnicas IS, Reúso Aplicación existente, Introducción gradual</i>
Detalles acerca del método para obtener esta información	<i>De acuerdo con los valores ingresados por el usuario Entrega Intermedia, Experiencia previa, Participación, Experiencia Técnicas IS, Reúso Aplicación existente, Introducción gradual se estima la aplicabilidad de prototipos</i>
Confiabilidad de los datos de entrada	—
Uso	<i>Permite determinar junto con otros atributos la aplicabilidad del modelo en Espiral para el área Gestión de Proyectos.</i>
Formato de los resultados de salida	<i>Texto. Participa en la actualización del atributo CV Propuesto Gestión.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-167 Descripción del atributo Aplicabilidad Prototipo

Información	Descripción
Nombre	<i>Aplicabilidad OO</i>
Concepto	<i>Coordinación</i>
Descripción	<i>Describe la posibilidad de aplicar la modelización de objetos en la aplicación a desarrollar.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Aplicable, No aplicable</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Obtenido a partir de Conveniencia Metodología, entrega Intermedia, Formalidad, Reúso Aplicación existente</i>
Detalles acerca del método para obtener esta información	<i>De acuerdo con los valores ingresados por el usuario en de Conveniencia Metodología, entrega Intermedia, Formalidad, Reúso Aplicación existente se estima la aplicabilidad de objetos</i>
Confiabilidad de los datos de entrada	—
Uso	<i>Permite determinar junto con otros atributos la aplicabilidad del modelo en Objetos para el área Gestión de Proyectos.</i>
Formato de los resultados de salida	<i>Texto. Participa en la actualización del atributo CV Propuesto Gestión.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-168 Descripción del atributo Aplicabilidad OO

Información	Descripción
Nombre	<i>Procedimientos D-M</i>
Concepto	<i>Coordinación</i>
Descripción	<i>Describe de acuerdo con el grado de factibilidad y necesidad si es conveniente considerar los mismos procedimientos para el desarrollo y para el mantenimiento.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Conveniente, No conveniente</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Obtenido a partir de Procedimientos Desarrollo-Mantenimiento</i>
Detalles acerca del método para obtener esta información	<i>De acuerdo con los valores ingresados por el usuario en Procedimientos Desarrollo-Mantenimiento se estima la conveniencia o no de los mismos procedimientos.</i>
Confiabilidad de los datos de entrada	—
Uso	<i>Permite determinar junto con otros atributos la aplicabilidad de la Espiral.</i>
Formato de los resultados de salida	<i>Texto. Participa en la actualización del atributo Aplicabilidad Espiral.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-169 Descripción del atributo Procedimientos D-M

Información	Descripción
Nombre	<i>Participación</i>
Concepto	<i>Usuario</i>
Descripción	<i>Describe el grado de participación del cliente en el durante el proceso de desarrollo de la aplicación.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Fuerte, Regular, Poca</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos y de las primeras entrevistas con los usuarios y de su propio conocimiento de proceso del tipo de aplicación y del negocio.</i>
Confiabilidad de los datos de entrada	<i>El sistema experto verifica que sólo pueda ingresarse uno de los valores, ya que son excluyentes.</i>
Uso	<i>Permite determinar la aplicabilidad del modelo de prototipo.</i>
Formato de los resultados de salida	<i>Texto. Actualiza el atributo Aplicabilidad Prototipo.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-170 Descripción del atributo Participación

Información	Descripción
Nombre	<i>Introducción gradual</i>
Concepto	<i>Usuario</i>
Descripción	<i>Describe si es necesario facilitar la introducción gradual del sistema a los usuarios.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Necesaria, No necesaria</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos y de las primeras entrevistas con los usuarios y de su propio conocimiento de proceso del tipo de aplicación y del negocio.</i>
Confiabilidad de los datos de entrada	<i>El sistema experto verifica que sólo pueda ingresarse uno de los valores, ya que son excluyentes.</i>
Uso	<i>Permite determinar la aplicabilidad del modelo de prototipo.</i>
Formato de los resultados de salida	<i>Texto. Actualiza el atributo Aplicabilidad Prototipo.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-171 Descripción del atributo Introducción gradual

Información	Descripción
Nombre	<i>Experiencia técnicas IS</i>
Concepto	<i>Equipo de Proyecto</i>
Descripción	<i>Describe si el equipo que participará en el desarrollo del proyecto tiene experiencia con las técnicas de ingeniería del software a usar.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Hay, No hay</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos y de las primeras entrevistas con los usuarios y de su propio conocimiento de proceso del tipo de aplicación y del negocio.</i>
Confiabilidad de los datos de entrada	<i>El sistema experto verifica que sólo pueda ingresarse uno de los valores, ya que son excluyentes.</i>
Uso	<i>Permite determinar la aplicabilidad del modelo de prototipo.</i>
Formato de los resultados de salida	<i>Texto. Actualiza el atributo Aplicabilidad Prototipo.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-172 Descripción del atributo Experiencia técnicas IS

Información	Descripción
Nombre	<i>Experiencia previa</i>
Concepto	<i>Equipo de Proyecto</i>
Descripción	<i>Describe si el equipo que participará en el desarrollo del proyecto tiene experiencia previa en el tipo de sistema a desarrollar.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Hay, No hay</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos y de las primeras entrevistas con los usuarios y de su propio conocimiento de proceso del tipo de aplicación y del negocio.</i>
Confiabilidad de los datos de entrada	<i>El sistema experto verifica que sólo pueda ingresarse uno de los valores, ya que son excluyentes.</i>
Uso	<i>Permite determinar la aplicabilidad del modelo de prototipo.</i>
Formato de los resultados de salida	<i>Texto. Actualiza el atributo Aplicabilidad Prototipo.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-173 Descripción del atributo Experiencia previa

Información	Descripción
Nombre	<i>Viabilidad software</i>
Concepto	<i>Riesgos del Proyecto</i>
Descripción	<i>Describe si existen dudas acerca de la viabilidad de la aplicación a desarrollar.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Hay certeza, Hay dudas</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos y de las primeras entrevistas con los usuarios y de su propio conocimiento de proceso del tipo de aplicación y del negocio.</i>
Confiabilidad de los datos de entrada	<i>El sistema experto verifica que sólo pueda ingresarse uno de los valores, ya que son excluyentes.</i>
Uso	<i>Permite determinar la existencia de factor de riesgo.</i>
Formato de los resultados de salida	<i>Texto. Actualiza el atributo Factor Riesgo.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-174 Descripción del atributo Viabilidad software

Información	Descripción
Nombre	<i>Nivel de riesgo</i>
Concepto	<i>Riesgos del Proyecto</i>
Descripción	<i>Describe si el grado de riesgo que se estima existe para el desarrollo del proyecto.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Alto riesgo, Mediano riesgo, Bajo riesgo, No hay riesgo</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos y de las primeras entrevistas con los usuarios y de su propio conocimiento de proceso del tipo de aplicación y del negocio.</i>
Confiabilidad de los datos de entrada	<i>El sistema experto verifica que sólo pueda ingresarse uno de los valores, ya que son excluyentes.</i>
Uso	<i>Permite determinar cuan aceptable es el riesgo estimado para cada tipo de CV considerado cascada, objetos, espiral. Para el modelo en espiral el riesgo aceptable es de nivel mediano o alto. Para el modelo en objetos el riesgo aceptable es mediano, bajo o sin riesgo. Para el modelo en cascada el riesgo aceptable es bajo o sin riesgo.</i>
Formato de los resultados de salida	<i>Texto. Actualiza los atributos Riesgo Cascada, Riesgo Objetos y Riesgo Espiral.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-175 Descripción del atributo Nivel de riesgo

Información	Descripción
Nombre	<i>Análisis riesgo</i>
Concepto	<i>Riesgos del Proyecto</i>
Descripción	<i>Describe si es necesario definir explícitamente estrategias para realizar análisis de riesgo.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Estrategias explícitas, No estrategias explícitas</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos y de las primeras entrevistas con los usuarios y de su propio conocimiento de proceso del tipo de aplicación y del negocio.</i>
Confiabilidad de los datos de entrada	<i>El sistema experto verifica que sólo pueda ingresarse uno de los valores, ya que son excluyentes.</i>
Uso	<i>Permite determinar la aplicabilidad del modelo en espiral.</i>
Formato de los resultados de salida	<i>Texto. Actualiza el atributo Aplicabilidad Espiral.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-176 Descripción del atributo Análisis riesgo

Información	Descripción
Nombre	<i>Técnicas AR</i>
Concepto	<i>Riesgos del Proyecto</i>
Descripción	<i>Describe si se cuenta con los conocimientos para aplicar métodos y técnicas par evaluar el riesgo.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Se dispone, No se dispone</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos y de las primeras entrevistas con los usuarios y de su propio conocimiento de proceso del tipo de aplicación y del negocio.</i>
Confiabilidad de los datos de entrada	<i>El sistema experto verifica que sólo pueda ingresarse uno de los valores, ya que son excluyentes.</i>
Uso	<i>Permite determinar junto con otros atributos (Factor Riesgo y Habilidad Riesgo) si el factor de riesgo es evaluable.</i>
Formato de los resultados de salida	<i>Texto. Actualiza junto con otros atributos el atributo Riesgo.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-177 Descripción del atributo Técnicas AR

Información	Descripción
Nombre	<i>Identificación Alternativas</i>
Concepto	<i>Riesgos del Proyecto</i>
Descripción	<i>Describe si es posible identificar distintas alternativas para resolver los riesgos y si es posible volver a etapas anteriores en el desarrollo del sistema cuando se identifican las mejores alternativas.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Se identifican, No se identifican, Etapas anteriores</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 2</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos y de las primeras entrevistas con los usuarios y de su propio conocimiento de proceso del tipo de aplicación y del negocio.</i>
Confiabilidad de los datos de entrada	<i>No se consideran valores excluyentes.</i>
Uso	<i>Permite determinar si se cuenta con habilidad para la evaluación del riesgo la aplicabilidad del modelo de prototipo.</i>
Formato de los resultados de salida	<i>Texto. Actualiza el atributo Habilidad Riesgo y Aplicabilidad Espiral.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-178 Descripción del atributo Identificación Alternativas

Información	Descripción
Nombre	<i>Riesgos Alternativa</i>
Concepto	<i>Riesgos del Proyecto</i>
Descripción	<i>Describe si es posible identificar los riesgos asociados con cada una de las alternativas para el desarrollo del sistema y si es necesario volver a etapas anteriores en el desarrollo del sistema para la resolución de algún tópico nuevo de riesgo.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Se identifican, No se identifican, Etapas anteriores</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 2</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos y de las primeras entrevistas con los usuarios y de su propio conocimiento de proceso del tipo de aplicación y del negocio.</i>
Confiabilidad de los datos de entrada	<i>No se consideran valores excluyentes.</i>
Uso	<i>Permite determinar si se cuenta con habilidad para la evaluación del riesgo la aplicabilidad del modelo de prototipo.</i>
Formato de los resultados de salida	<i>Texto. Actualiza el atributo Habilidad Riesgo y Aplicabilidad Espiral.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-179 Descripción del atributo Riesgos Alternativa

Información	Descripción
Nombre	<i>Categorías riesgo</i>
Concepto	<i>Riesgos del Proyecto</i>
Descripción	<i>Describe si que tipo de riesgos se identifican en el desarrollo del sistema.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Técnicos, Otros riesgos</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos y de las primeras entrevistas con los usuarios y de su propio conocimiento de proceso del tipo de aplicación y del negocio.</i>
Confiabilidad de los datos de entrada	<i>No se consideran valores excluyentes.</i>
Uso	<i>Permite determinar si se existe factor de riesgo.</i>
Formato de los resultados de salida	<i>Texto. Actualiza el atributo Factor Riesgo.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-180 Descripción del atributo Categorías riesgo

Información	Descripción
Nombre	<i>Objetivo calidad</i>
Concepto	<i>Riesgos del Proyecto</i>
Descripción	<i>Describe si es necesario incorporar mecanismos explícitos para incorporar objetivos de calidad en el desarrollo del producto.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Mecanismos explícitos, No mecanismos explícitos</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos y de las primeras entrevistas con los usuarios y de su propio conocimiento de proceso del tipo de aplicación y del negocio.</i>
Confiabilidad de los datos de entrada	<i>El sistema experto verifica que sólo pueda ingresarse uno de los valores, ya que son excluyentes.</i>
Uso	<i>Permite determinar la aplicabilidad del modelo de prototipo.</i>
Formato de los resultados de salida	<i>Texto. Actualiza el atributo Aplicabilidad Espiral.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-181 Descripción del atributo Objetivo calidad

Información	Descripción
Nombre	<i>Terminación Proyectos</i>
Concepto	<i>Riesgos del Proyecto</i>
Descripción	<i>Describe si se estima la terminación prematura de proyectos no viables.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Prematura, En término</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Ingresada por el usuario</i>
Detalles acerca del método para obtener esta información	<i>Valor estimado por el usuario del sistema experto a partir del Informe de requerimientos y de las primeras entrevistas con los usuarios y de su propio conocimiento de proceso del tipo de aplicación y del negocio.</i>
Confiabilidad de los datos de entrada	<i>El sistema experto verifica que sólo pueda ingresarse uno de los valores, ya que son excluyentes.</i>
Uso	<i>Permite determinar la aplicabilidad del modelo de prototipo.</i>
Formato de los resultados de salida	<i>Texto. Actualiza el atributo Aplicabilidad Espiral.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-182 Descripción del atributo Terminación Proyectos

Información	Descripción
Nombre	<i>Riesgo Cascada</i>
Concepto	<i>Riesgos del Proyecto</i>
Descripción	<i>Describe el nivel de riesgo aceptable para un modelo de ciclo de vida en cascada.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Aceptable, No aceptable</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Obtenido a partir de Nivel de Riesgo</i>
Detalles acerca del método para obtener esta información	<i>De acuerdo con los valores ingresados por el usuario en Nivel de Riesgo se estima si es no aceptable para aplicar modelo en cascada</i>
Confiabilidad de los datos de entrada	—
Uso	<i>Permite determinar junto con otros atributos la aplicabilidad del ciclo de vida en Cascada en la Gestión de Proyectos.</i>
Formato de los resultados de salida	<i>Texto. Actualiza el atributo CV Propuesto Gestión</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-183 Descripción del atributo Riesgo Cascada

Información	Descripción
Nombre	<i>Riesgo Objetos</i>
Concepto	<i>Riesgos del Proyecto</i>
Descripción	<i>Describe el nivel de riesgo aceptable para un modelo de ciclo de vida en objetos.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Aceptable, No aceptable</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Obtenido a partir de Nivel de Riesgo</i>
Detalles acerca del método para obtener esta información	<i>De acuerdo con los valores ingresados por el usuario en Nivel de Riesgo se estima si es no aceptable para aplicar modelo en objetos</i>
Confiabilidad de los datos de entrada	—
Uso	<i>Permite determinar junto con otros atributos la aplicabilidad del ciclo de vida en Objetos en la Gestión de Proyectos.</i>
Formato de los resultados de salida	<i>Texto. Actualiza el atributo CV Propuesto Gestión</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-184 Descripción del atributo Riesgo Objetos

Información	Descripción
Nombre	<i>Habilidad Riesgo</i>
Concepto	<i>Riesgos del Proyecto</i>
Descripción	<i>Describe si existe habilidad para evaluar el nivel de riesgo para un modelo de ciclo de vida en espiral.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Existe, No existe</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Obtenido a partir de Identificación alternativas, Riesgos alternativa</i>
Detalles acerca del método para obtener esta información	<i>De acuerdo con los valores ingresados por el usuario en Identificación alternativas, Riesgos alternativa se estima si existe o no habilidad para evaluar el nivel de riesgo</i>
Confiabilidad de los datos de entrada	—
Uso	<i>Permite determinar junto con otros atributos la aplicabilidad del modelado en espiral y la evaluabilidad del riesgo.</i>
Formato de los resultados de salida	<i>Texto. Actualiza los atributos Aplicabilidad Espiral y Riesgo</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-185 Descripción del atributo Habilidad Riesgo

Información	Descripción
Nombre	<i>Factor Riesgo</i>
Concepto	<i>Riesgos del Proyecto</i>
Descripción	<i>Describe si existen factores de riesgo.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Existe, No existe</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Obtenido a partir de Viabilidad software, Riesgo Espiral, Categorías Riesgo.</i>
Detalles acerca del método para obtener esta información	<i>De acuerdo con los valores ingresados por el usuario Viabilidad software, Riesgo Espiral, Categorías Riesgo se estima si existe o no Factor de riesgo para aplicar modelo en espiral</i>
Confiabilidad de los datos de entrada	—
Uso	<i>Permite determinar junto con otros atributos la aplicabilidad del modelo espiral y la Evaluabilidad del Riesgo.</i>
Formato de los resultados de salida	<i>Texto. Junto con otros atributos actualiza los atributos Aplicabilidad Espiral y Riesgo</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-186 Descripción del atributo Factor Riesgo

Información	Descripción
Nombre	<i>Riesgo Espiral</i>
Concepto	<i>Riesgos del Proyecto</i>
Descripción	<i>Describe el nivel de riesgo aceptable para un modelo de ciclo de vida en espiral.</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Aceptable, No aceptable</i>
Nro. Valores por caso	<i>Mínimo 0, Máximo 1</i>
Fuente	<i>Obtenido a partir de Nivel de Riesgo</i>
Detalles acerca del método para obtener esta información	<i>De acuerdo con los valores ingresados por el usuario en Nivel de Riesgo se estima si es no aceptable para aplicar modelo en espiral</i>
Confiabilidad de los datos de entrada	—
Uso	<i>Permite determinar junto con otros atributos si existe Factor de riesgo.</i>
Formato de los resultados de salida	<i>Texto. Actualiza junto con otros atributos, el atributo Factor Riesgo</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-187 Descripción del atributo Riesgo Espiral

Información	Descripción
Nombre	Riesgo
Concepto	Riesgos del Proyecto
Descripción	Define si el riesgo es evaluable para un modelo de ciclo de vida en espiral.
Tipo Valor	Texto
Rango de valores	Evaluable, No evaluable
Nro. Valores por caso	Mínimo 0, Máximo 1
Fuente	Obtenido a partir de Factor Riesgo, Técnicas AR, Habilidad Riesgo
Detalles acerca del método para obtener esta información	De acuerdo con los valores ingresados por el usuario en Factor Riesgo, Técnicas AR, Habilidad Riesgo se estima si es no evaluable para aplicar modelo en espiral
Confiabilidad de los datos de entrada	—
Uso	Permite determinar junto con otros atributos la aplicabilidad del ciclo de vida en Espiral.
Formato de los resultados de salida	Texto. Actualiza el atributo CV Propuesto Gestión
Material de soporte	Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII

Tabla 6-188 Descripción del atributo Riesgo

Información	Descripción
Nombre	CV Propuesto Requisitos
Concepto	Área Requisitos
Descripción	Define el modelo ciclo de vida mas adecuado para el área Requisitos.
Tipo Valor	Texto
Rango de valores	Cascada, Objetos, Espiral, No hay propuesta
Nro. Valores por caso	Mínimo 1
Fuente	Obtenido a partir de Definición Requisitos, Definición Límites, Grado de Cumplimiento, Grado de Certidumbre
Detalles acerca del método para obtener esta información	De acuerdo con los de Definición Requisitos, Definición Límites, Grado de Cumplimiento, Grado de Certidumbre se estima el ciclo de vida mas adecuado para el área Requisitos
Confiabilidad de los datos de entrada	—
Uso	Permite determinar junto con otros atributos el ciclo de vida más adecuado para las características del proyecto analizado.
Formato de los resultados de salida	Texto. Actualiza el atributo CV Propuesto del Proyecto. Se edita resultado en pantalla.
Material de soporte	Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII

Tabla 6-189 Descripción del atributo CV Propuesto Requisitos

Información	Descripción
Nombre	<i>CV Propuesto Aplicación</i>
Concepto	<i>Área Aplicación</i>
Descripción	<i>Define el modelo ciclo de vida mas adecuado para el área Aplicación</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Cascada, Objetos, Espiral, No hay propuesta</i>
Nro. Valores por caso	<i>Mínimo 1</i>
Fuente	<i>Obtenido a partir de Componentes predominantes, Comportamiento predominante, Complejidad Subsistemas, Retroalimentación Fases Desarrollo, Progresión Fases Desarrollo, Opcionalidad, integración HW-SW, Modelado objetos, Modelado Prototipo.</i>
Detalles acerca del método para obtener esta información	<i>De acuerdo con de Componentes predominantes, Comportamiento predominante, Complejidad Subsistemas, Retroalimentación Fases Desarrollo, Progresión Fases Desarrollo, Opcionalidad, integración HW-SW, Modelado objetos, Modelado Prototipo se estima el ciclo de vida mas adecuado para el área Aplicación</i>
Confiabilidad de los datos de entrada	—
Uso	<i>Permite determinar junto con otros atributos el ciclo de vida más adecuado para las características del proyecto analizado.</i>
Formato de los resultados de salida	<i>Texto. Actualiza el atributo CV Propuesto del Proyecto. Se edita resultado en pantalla.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-190 Descripción del atributo CV Propuesto Aplicación

Información	Descripción
Nombre	<i>CV Propuesto Gestión</i>
Concepto	<i>Área Gestión Proyecto</i>
Descripción	<i>Define el modelo ciclo de vida mas adecuado para el área Gestión Proyecto</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Cascada, Objetos, Espiral, No hay propuesta</i>
Nro. Valores por caso	<i>Mínimo 1</i>
Fuente	<i>Obtenido a partir de Riesgo Cascada, Responsabilidad, Control Gestión, Formalidad, Factibilidad Prototipo, Conveniencia metodología, Entregas, Riesgo Objetos, Sistema OO, Aplicabilidad OO, Factibilidad Prototipo, Aplicabilidad Prototipo, aplicabilidad Espiral, Riesgo</i>
Detalles acerca del método para obtener esta información	<i>De acuerdo con los de Riesgo Cascada, Responsabilidad, Control Gestión, Formalidad, Factibilidad Prototipo, Conveniencia metodología, Entregas, Riesgo Objetos, Sistema OO, Aplicabilidad OO, Factibilidad Prototipo, Aplicabilidad Prototipo, aplicabilidad Espiral, Riesgo se estima el ciclo de vida mas adecuado para el área Gestión Proyecto</i>
Confiabilidad de los datos de entrada	—
Uso	<i>Permite determinar junto con otros atributos el ciclo de vida más adecuado para las características del proyecto analizado.</i>
Formato de los resultados de salida	<i>Texto. Actualiza el atributo CV Propuesto del Proyecto. Se edita resultado en pantalla.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-191 Descripción del atributo CV Propuesto Gestión

Información	Descripción
Nombre	<i>CV Propuesto del Proyecto</i>
Concepto	<i>CV Diagnóstico</i>
Descripción	<i>Define el modelo ciclo de vida mas adecuado para el Proyecto</i>
Tipo Valor	<i>Texto</i>
Rango de valores	<i>Cascada, Objetos, Espiral, No hay propuesta</i>
Nro. Valores por caso	<i>Mínimo 1</i>
Fuente	<i>Obtenido a partir de CV Propuesto Requisitos, CV Propuesto Aplicación, CV Propuesto Gestión</i>
Detalles acerca del método para obtener esta información	<i>De acuerdo con los de CV Propuesto Requisitos, CV Propuesto Aplicación, CV Propuesto Gestión se estima el ciclo de vida mas adecuado para el Proyecto</i>
Confiabilidad de los datos de entrada	<i>—</i>
Uso	<i>Permite determinar junto con otros atributos el ciclo de vida más adecuado para las características del proyecto analizado.</i>
Formato de los resultados de salida	<i>Texto. Actualiza el atributo CV Propuesto Proyecto y CV Propuesto por SE. Se edita resultado en pantalla.</i>
Material de soporte	<i>Sesiones: A.4.5 sesión IV, A.4.3 sesión VI, A.4.4 sesión VII, A.4.5 sesión VIII</i>

Tabla 6-192 Descripción del atributo CV Propuesto del Proyecto

6.2.5.2 COMPROBACION DE LOS CONOCIMIENTOS FACTICOS

Se han realizado junto con el experto revisiones exhaustivas de la definición de los atributos. Esta verificación ha permitido completar cuándo era necesario la tabla 6-3 de conceptos-atributos-valores y revisar las seudorreglas definidas anteriormente. La representación de los conocimientos fácticos se considera satisfactoria.

6.3 SINTESIS DE CONOCIMIENTOS

En los pasos precedentes se ha desarrollado el proceso de análisis de los conocimientos identificando, modelando y documentando los conocimientos estratégicos, tácticos y fácticos. A partir de ahora se presenta la síntesis de esos conocimientos en el Modelo Dinámico, también conocido como Modelo de Procesos y en el Modelo Estático. Por último, y como paso final de la fase de Conceptualización, ambos Modelos se integran en el Mapa de Conocimientos.

6.3.1 MODELO DE PROCESOS – MODELO DINAMICO

Para la construcción del modelo dinámico hay que tomar como punto de partida la identificación de los conocimientos estratégicos, definir una jerarquía entre las tareas y comprobar que no haya errores u olvidos. El experto participa

en este paso corroborando las metas, submetas y decisiones y los conceptos y atributos que se usan en cada estadio.

Las figuras 6-4 a 6-7 muestran la Jerarquía de Tareas (metas, submetas y procesos) para Seleccionar el Ciclo de Vida del proyecto.

Las figuras 6-8 a 6-10 muestran los conceptos y las relaciones interviniéntes en cada uno de los procedimientos (submetas).

Las Tablas 6-193 a 6-212 muestran la descripción de los procesos del Modelo Dinámico.

6.3.2 MODELO ESTÁTICO

El modelo estático está formado por los siguientes componentes que han sido documentados y actualizados durante el proceso de análisis de los conocimientos fácticos:

- Glosario de Términos: Tabla 6-1, páginas 152 a 155.
- Diccionario de conceptos: Tabla 6-2, páginas 156 a 158.
- Tabla de Concepto Atributo Valor: Tabla 6-3, páginas 159 a 165.
- Modelo Relacional de los Conceptos: Figura 6-2, página 166.

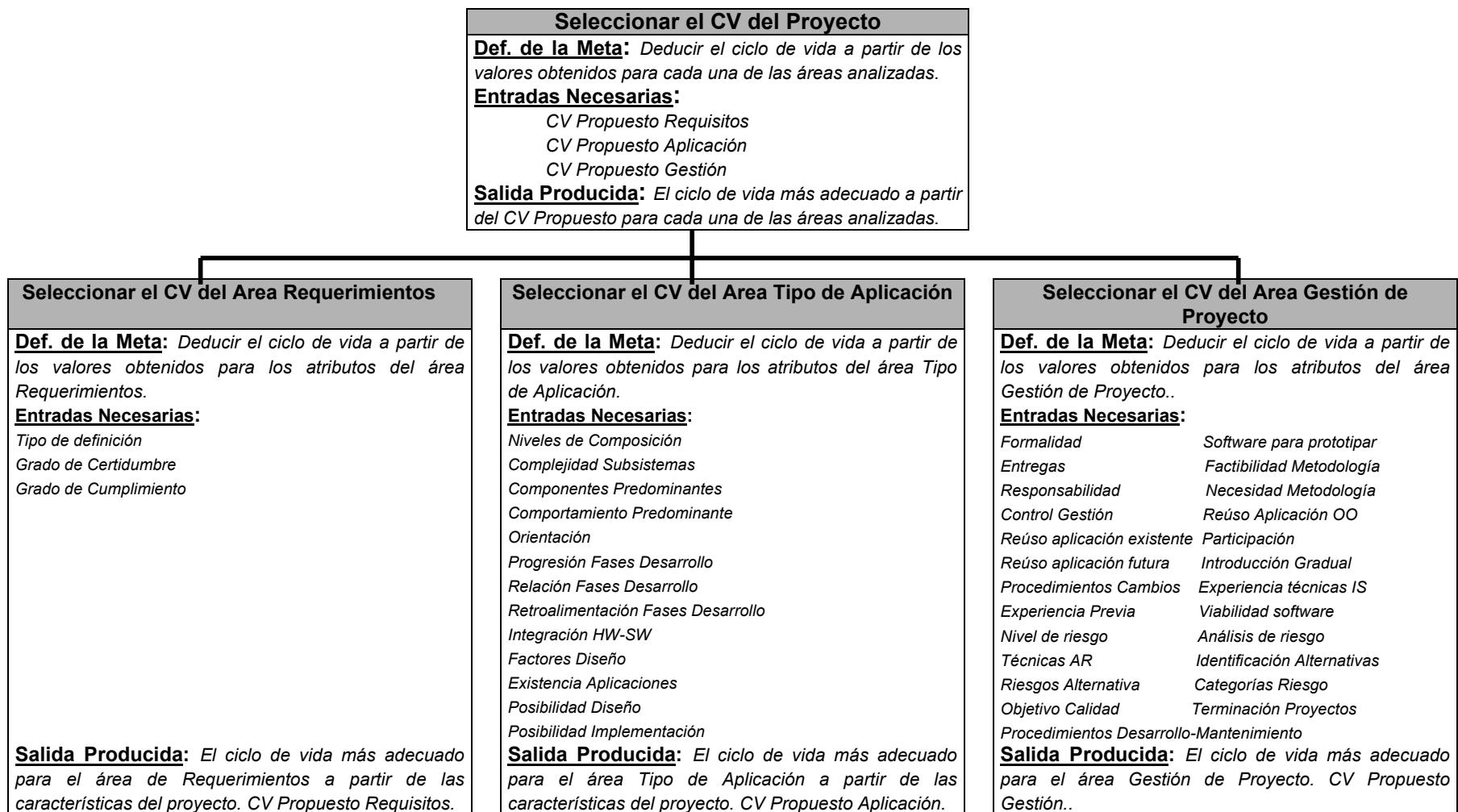


Figura 6-4 Jerarquía de Tareas de Seleccionar el Ciclo de Vida del Proyecto

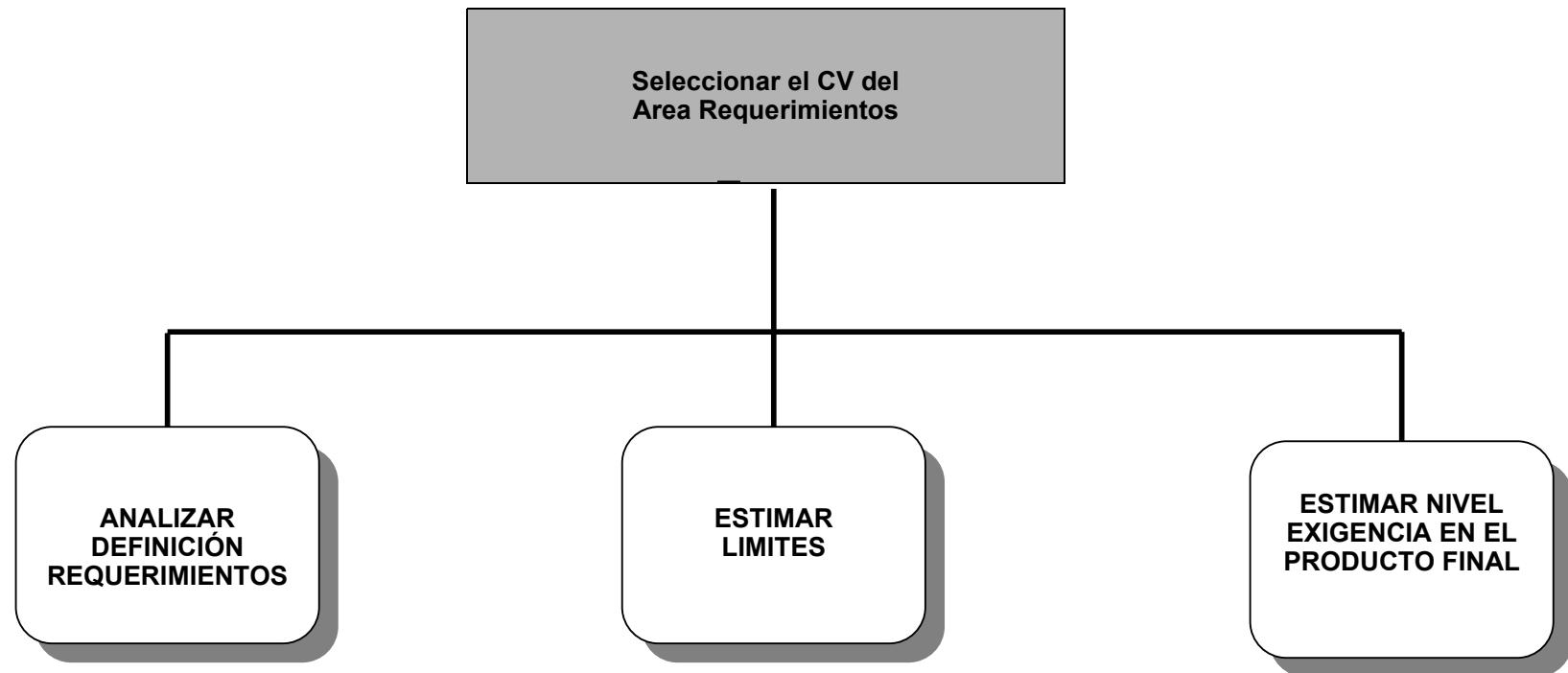


Figura 6-5 Jerarquía de Tareas de Seleccionar del Ciclo de Vida para el Área Especificación Requerimientos

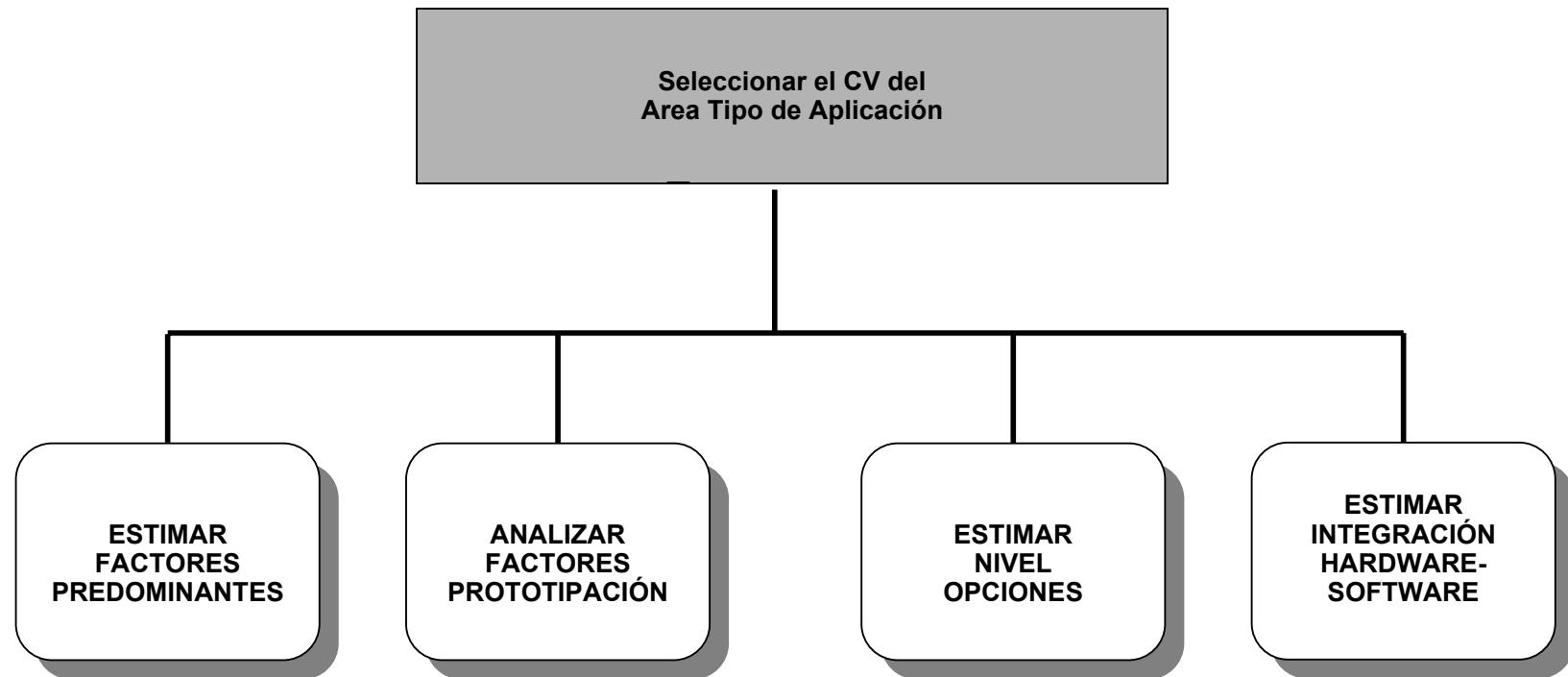


Figura 6-6 Jerarquía de Tareas de Seleccionar el Ciclo de Vida para el Área Tipo de Aplicación

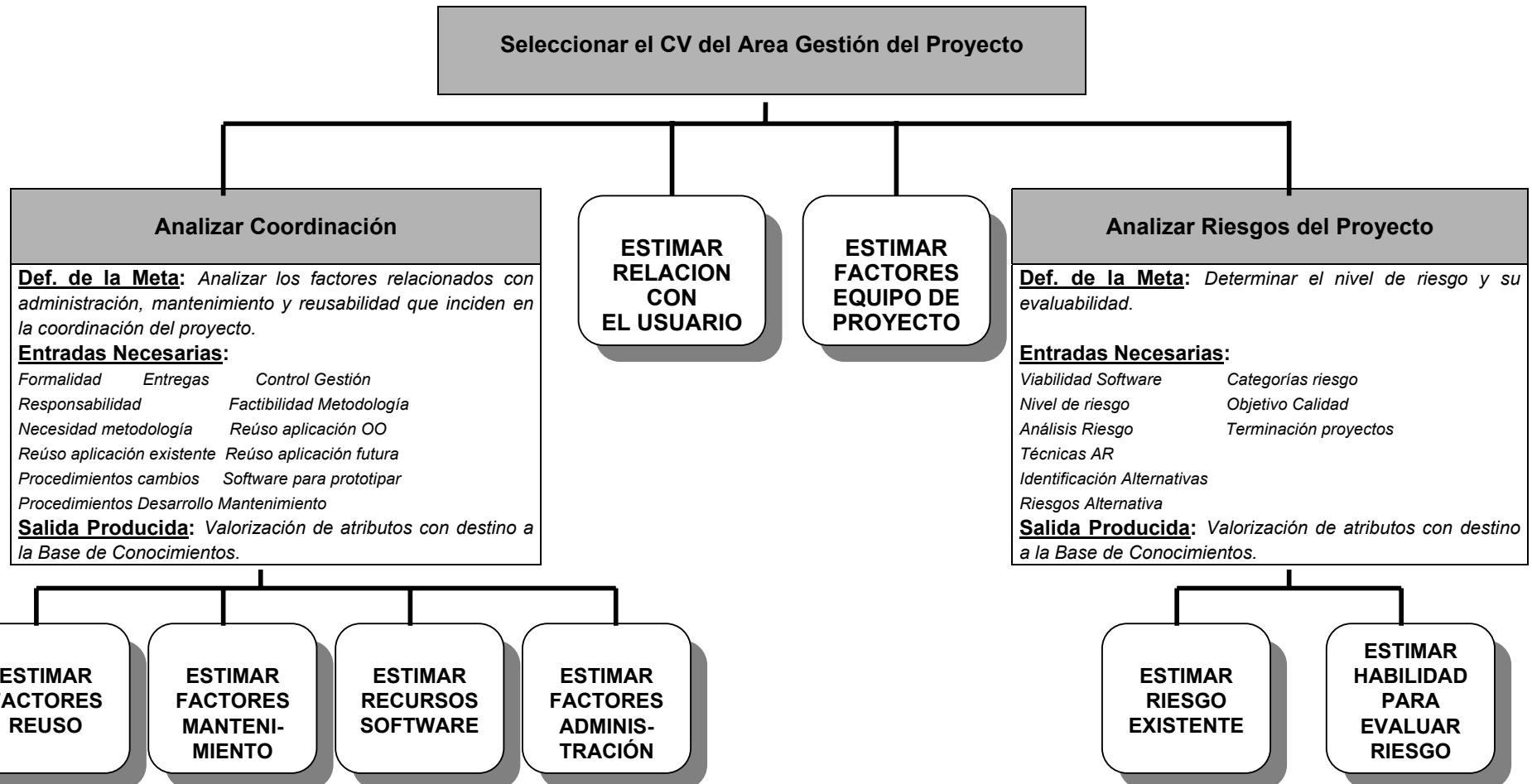


Figura 6-7 Jerarquía de Tareas de Seleccionar del Ciclo de Vida para el Área Gestión del Proyecto

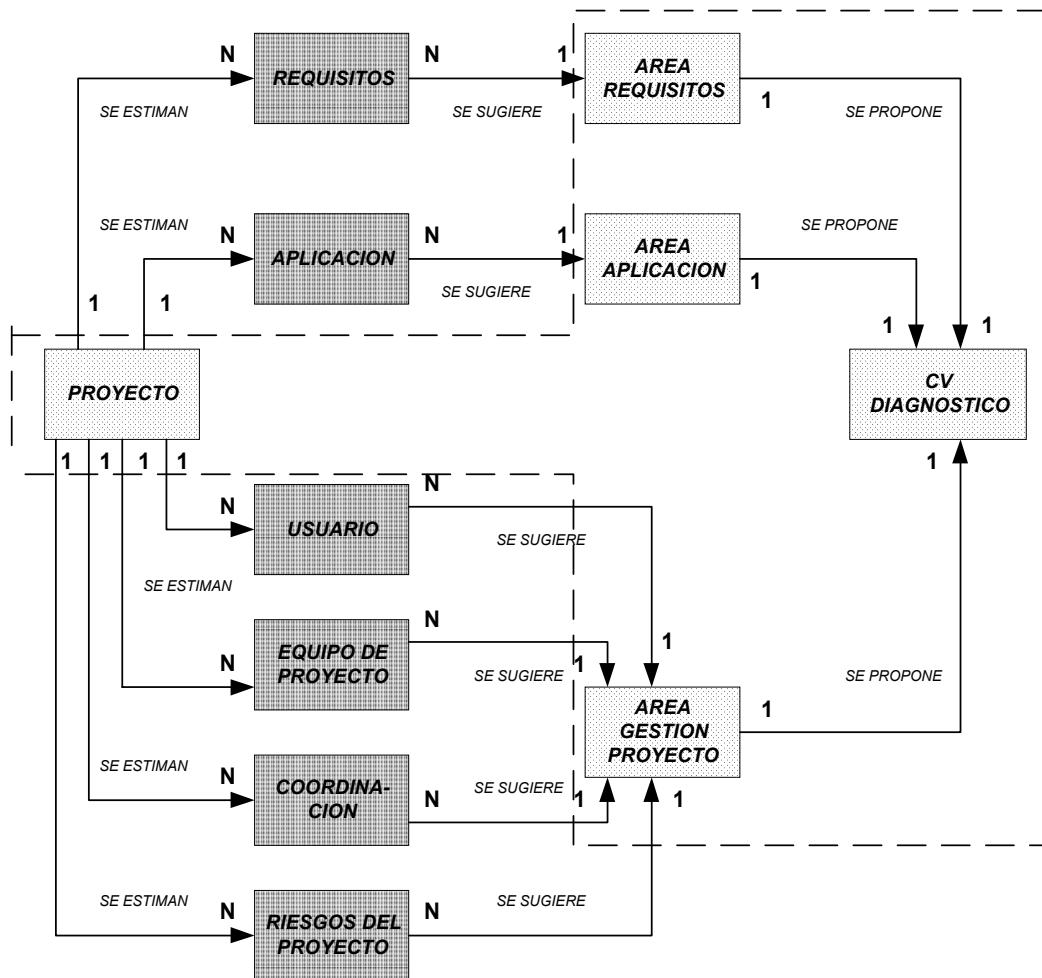


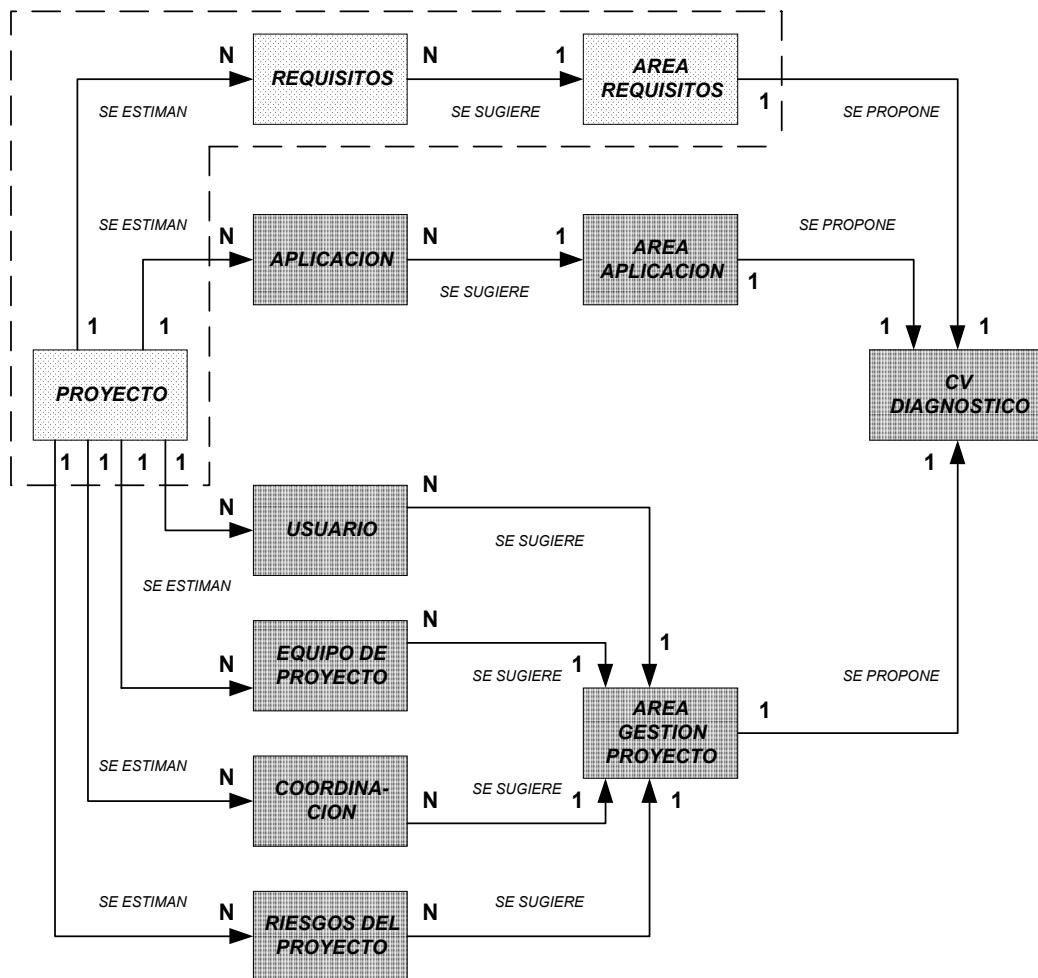
FIGURA 6-8 Conceptos Intervinientes en el Proceso
Seleccionar el CV del Proyecto

Analizar Definición Requerimientos	
Propósito:	Establecer el nivel de claridad con el que se encuentran definidos los requisitos y el grado de certidumbre en la definición de los mismos al inicio del proyecto.
Información necesaria:	Los valores asignados por el usuario del sistema a los atributos Tipo de Definición y Grado de certidumbre
Acciones:	<ol style="list-style-type: none"> 1. Solicitar al usuario los valores de Tipo de definición 2. Solicitar al usuario los valores Grado de certidumbre 3. Deducir Definición requisitos <ol style="list-style-type: none"> 3.1 Tipo de Definición formales y exhaustivos y uniformes = Definición requisitos es definición clara.

Tabla 6-193 Descripción del Proceso Analizar Definición Requerimientos

Estimar Límites	
Propósito:	Establecer el nivel de claridad con el que se encuentran definidos los límites al inicio del proyecto.
Información necesaria:	Los valores asignados por el usuario del sistema a los atributos Tipo de Definición
Acciones:	<ol style="list-style-type: none"> 1. Solicitar al usuario los valores de Tipo de definición 2. Deducir Definición Límites <ol style="list-style-type: none"> 3.1 Tipo de Definición formales y exhaustivos y uniformes = Definición límites es definición clara.

Tabla 6-194 Descripción del Proceso Estimar Límites

FIGURA 6-9 Conceptos Intervinientes en el Proceso
Seleccionar el CV del área Especificación Requerimientos

Estimar Nivel de Exigencia en el Producto Final	
Propósito:	<i>Establecer el grado de cumplimiento de los requerimientos en el producto final que exige el usuario.</i>
Información necesaria:	<i>Los valores asignados por el usuario del sistema a los atributos Grado de cumplimiento</i>
Acciones:	<p>1. Solicitar al usuario los valores Grado de certidumbre</p> <p>Tabla 6-195 Descripción del Proceso Estimar Nivel de Exigencia en el Producto Final</p>

Seleccionar el CV del Area Requerimientos	
Propósito:	<i>Deducir el ciclo de vida para el área Especificación de Requerimientos a partir de los valores obtenidos para los atributos del área requerimientos.</i>
Información necesaria:	<i>Los valores asignados por el usuario del sistema a los atributos Grado de Cumplimiento y Grado de Certidumbre.</i>
Acciones:	<p>1. Deducir si el CV del área es Objetos o Espiral</p> <p>1.1 Definición Requisitos es definición incierta + Definición Límites es definición incierta + Grado de certidumbre parcialmente definidos = CV Propuesto Requisitos es Objetos o Espiral</p> <p>1.2 Grado de Cumplimiento es usuario muy exigente = CV Propuesto Requisitos es Objetos o Espiral</p> <p>2. Deducir si el CV del área es Cascada</p> <p>2.1 Definición Requisitos es definición clara + Definición Límites es definición clara + Grado de certidumbre mayoritariamente definidos = CV Propuesto Requisitos es Cascada</p> <p>Tabla 6-196 Descripción del Proceso Seleccionar el CV del Area Requerimientos</p>

Estimar Factores Predominantes	
Propósito:	<i>Determinar las técnicas de modelado que mejor se adecuan a los factores predominantes de la aplicación a desarrollar.</i>
Información necesaria:	<i>Los valores asignados por el usuario del sistema a los atributos Componentes Predominantes, Comportamiento predominante y Orientación</i>
Acciones:	<p>1. Solicitar al usuario los valores de Componentes Predominantes + Comportamiento Predominante + Orientación</p> <p>2. Deducir modelo</p> <p>2.1 Componente predominante Algorítmico + Comportamiento Predominante batch + Resultado del procesamiento de Otros valores ingresados = CV Propuesto Aplicación Cascada</p> <p>2.2 Componente predominante Matemático o Gráfico = Modelado OO es conveniente</p> <p>2.3 Comportamiento predominante Dinámico o Interactivo o Tiempo Real = Modelado OO es conveniente</p> <p>2.4 Orientación Software de Base = Modelado OO es conveniente.</p> <p>2.5 Orientación Sistema Basado en Conocimiento = Modelado Prototipo es conveniente</p> <p>Tabla 6-197 Descripción del Proceso Estimar Factores Predominantes</p>

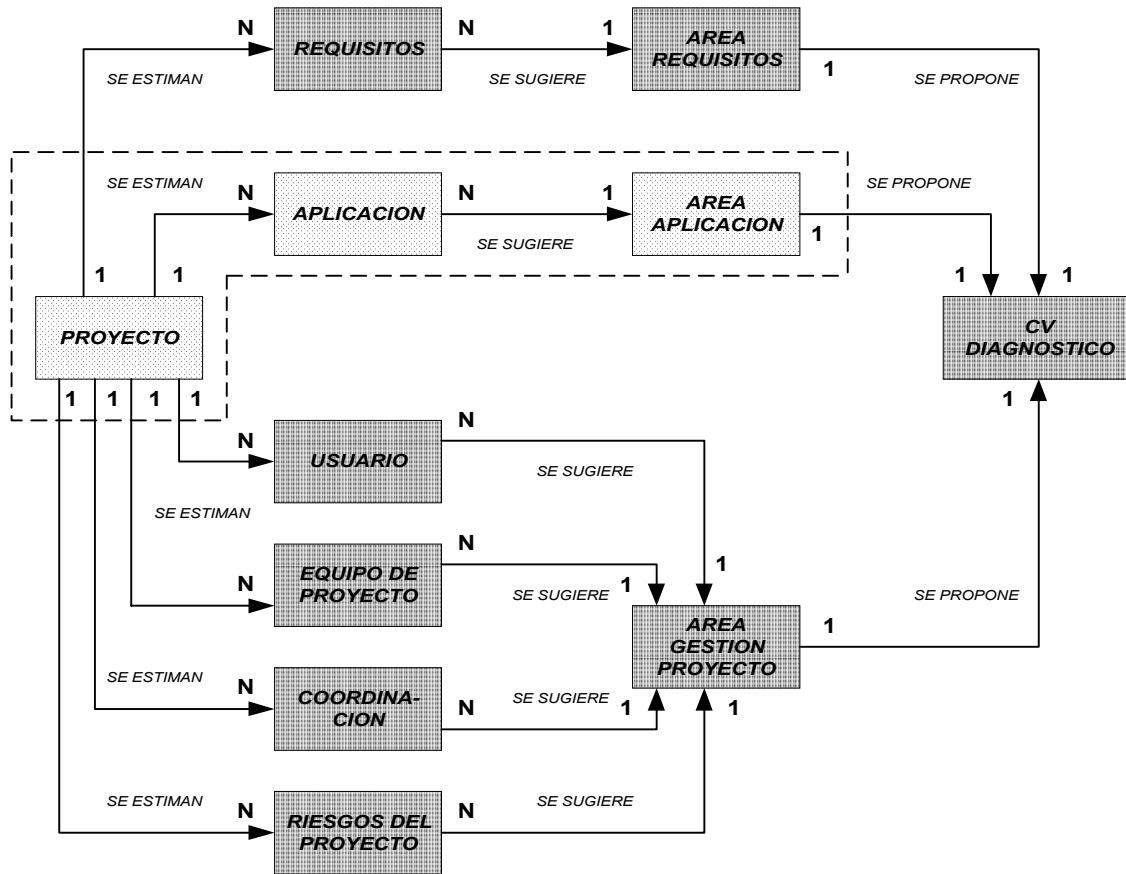


FIGURA 6-10 Conceptos Intervinientes en el Proceso
Seleccionar el CV del área Tipo de Aplicación

Estimar Nivel de Opciones	
Propósito:	Determinar si la aplicación a desarrollar es multiopcional desde el punto de vista del diseño y la implementación.
Información necesaria:	Los valores asignados por el usuario del sistema a los atributos Posibilidad Diseño, Posibilidad Implementación.
Acciones:	<ol style="list-style-type: none"> Solicitar al usuario los valores de los atributos necesarios Deducir grado de optionalidad de la aplicación <ol style="list-style-type: none"> Posibilidad Diseño es Varias Opciones = Optionalidad es Múltiple Posibilidad Implementación es Varias Opciones = Optionalidad es Múltiple Posibilidad Diseño es Pocas Opciones = Optionalidad es Simple Posibilidad Implementación es Pocas Opciones = Optionalidad es Simple

Tabla 6-198 Descripción del Proceso Estimar Nivel de Opciones

Analizar Factores Prototipación
Propósito: <i>Determinar si la técnicas de prototipación es conveniente para modelar la aplicación a desarrollar.</i>
Información necesaria: <i>Los valores asignados por el usuario del sistema a los atributos Niveles de Composición, Progresión Fases desarrollo, Relación Fases desarrollo, Factores Diseño, Existencia aplicaciones y Tipo de Modificación.</i>
Acciones: <ol style="list-style-type: none"> 1. Solicitar al usuario los valores de los atributos necesarios. 2. Deducir si el modelo de prototipos es conveniente <ol style="list-style-type: none"> 2.1 Niveles de Composición es pocos = Complejidad Subsistemas es baja 2.2 Niveles de Composición es varios = Complejidad Subsistemas es alta = Modelado Prototipo es conveniente 2.3 Relación Fases Desarrollo es Poca Dependencia o Independencia = Retroalimentación Fases desarrollo es baja 2.4 Relación Fases Desarrollo es Mucha Dependencia = Retroalimentación Fases desarrollo es alta = Modelado Prototipo es conveniente 2.5 Progresión Fases Desarrollo es No uniforme y Secuencial = Modelado Prototipo es conveniente 2.6 Factores Diseño es Problemas Arquitectura o Problemas eficacia = Modelado Prototipo es conveniente 2.7 Existencia Aplicaciones es innovador = Modelado Prototipo es conveniente 2.8 Tipo de Modificación es Redefinición o Extensión o Ampliación = Variabilidad procesos alta = Modelado Prototipo es conveniente

Tabla 6-199 Descripción del Proceso Analizar Factores Prototipación

Seleccionar el CV del Área Tipo de Aplicación
Propósito: <i>Deducir el ciclo de vida para el área Tipo de aplicación a partir de los valores obtenidos y deducidos para los atributos del área aplicación</i>
Información necesaria: <i>Los valores asignados por el usuario del sistema a los atributos Componentes predominantes y Comportamiento Predominante</i> <i>Los valores deducidos para los atributos Complejidad Subsistemas, Retroalimentación Fases Desarrollo, Modelado Prototipo, Modelado Objetos, Opcionalidad</i>
Acciones: <ol style="list-style-type: none"> 1. Deducir si el CV del área es Espiral <ol style="list-style-type: none"> 1.1 Modelado Objetos es conveniente + Modelado Prototipo es conveniente + Opcionalidad es Múltiple = CV Propuesto Aplicación es Espiral 1.2 Modelado Objetos es conveniente + Modelado Prototipo es conveniente + Integración HW-SW es fuertemente = CV Propuesto Aplicación es Espiral 2. Deducir si el CV del área es Objetos <ol style="list-style-type: none"> 2.1 Modelado Objetos es conveniente + Modelado Prototipo es conveniente = CV Propuesto Aplicación es Objetos 3. Deducir si el CV del área es Cascada <ol style="list-style-type: none"> 3.1 Complejidad Subsistemas es baja + Componente Predominante algorítmico + Progresión Fases Desarrollo Uniforme y Secuencial = CV Propuesto Aplicación es Cascada 3.2 Complejidad Subsistemas es baja + Comportamiento Predominante batch + Progresión Fases Desarrollo Uniforme y Secuencial = CV Propuesto Aplicación es Cascada 3.3 Retroalimentación Fases desarrollo baja + Componente Predominante algorítmico + Progresión Fases Desarrollo Uniforme y Secuencial = CV Propuesto Aplicación es Cascada 3.4 Retroalimentación Fases desarrollo baja + Comportamiento Predominante batch + Progresión Fases Desarrollo Uniforme y Secuencial = CV Propuesto Aplicación es Cascada

Tabla 6-200 Descripción del Proceso Seleccionar el CV del Área Tipo de Aplicación

Estimar Integración Hardware-Software
Propósito: <i>Obtener el nivel de integración entre el hardware y el software que requiere la aplicación a desarrollar.</i>
Información necesaria: <i>Los valores asignados por el usuario del sistema al atributo Integración HW-SW</i>
Acciones: <i>Solicitar al usuario los valores de Integración HW-SW</i>

Tabla 6-201 Descripción del Proceso Estimar Integración Hardware-Software

Estimar Factores Reúso
Propósito: <i>Determinar si existe una aplicación previa desarrollada en objetos y determinar la aplicabilidad del modelo de objetos.</i>
Información necesaria: <i>Los valores asignados por el usuario del sistema a los atributos Reúso Aplicación OO, Reúso aplicación existente, Reúso aplicación futura.</i>
Acciones: <ol style="list-style-type: none"> 1. Solicitar al usuario los valores de los atributos necesarios 2. Deducir aplicación previa desarrollada en objetos <ol style="list-style-type: none"> 2.1 Reúso Aplicación OO es Subsistema del existente o Ampliación del existente o Modificación del existente O Componentes del actual = Sistema OO es existe 2.2 Reúso Aplicación futura es muy necesario = Sistema OO es existe 3. Deducir si el modelo de objetos es aplicable <ol style="list-style-type: none"> 3.1 Reúso Aplicación existente es estrategias explícitas = Aplicabilidad OO es aplicable

Tabla 6-202 Descripción del Proceso Estimar Factores Reúso

Estimar Factores Mantenimiento
Propósito: <i>Determinar la aplicabilidad del modelo en espiral.</i>
Información necesaria: <i>Los valores asignados por el usuario del sistema a los atributos Procedimientos Cambios, Procedimientos Desarrollo- Mantenimiento.</i>
Acciones: <ol style="list-style-type: none"> 1. Solicitar al usuario los valores de los atributos necesarios 2. Deducir si el modelo en espiral es aplicable <ol style="list-style-type: none"> 2.1 Procedimientos Cambios es acuerdos confirmados = Aplicabilidad espiral es aplicable 2.2 Procedimientos Desarrollo-Mantenimiento es factible usar los mismos y es necesario usar los mismos = Procedimientos D-M es conveniente = Aplicabilidad espiral es aplicable

Tabla 6-203 Descripción del Proceso Estimar Factores Mantenimiento

Estimar Recursos Software
Propósito: <i>Determinar la factibilidad del modelo en prototipo.</i>
Información necesaria: <i>Los valores asignados por el usuario del sistema al atributo Software para prototipar.</i>
Acciones: <ol style="list-style-type: none"> 1. Solicitar al usuario los valores de los atributos necesarios 2. Deducir si el prototipo es factible <ol style="list-style-type: none"> 2.1 Software para prototipar es hay disponible o factible de adquirir = Factibilidad prototipo es factible

Tabla 6-204 Descripción del Proceso Estimar Recursos Software

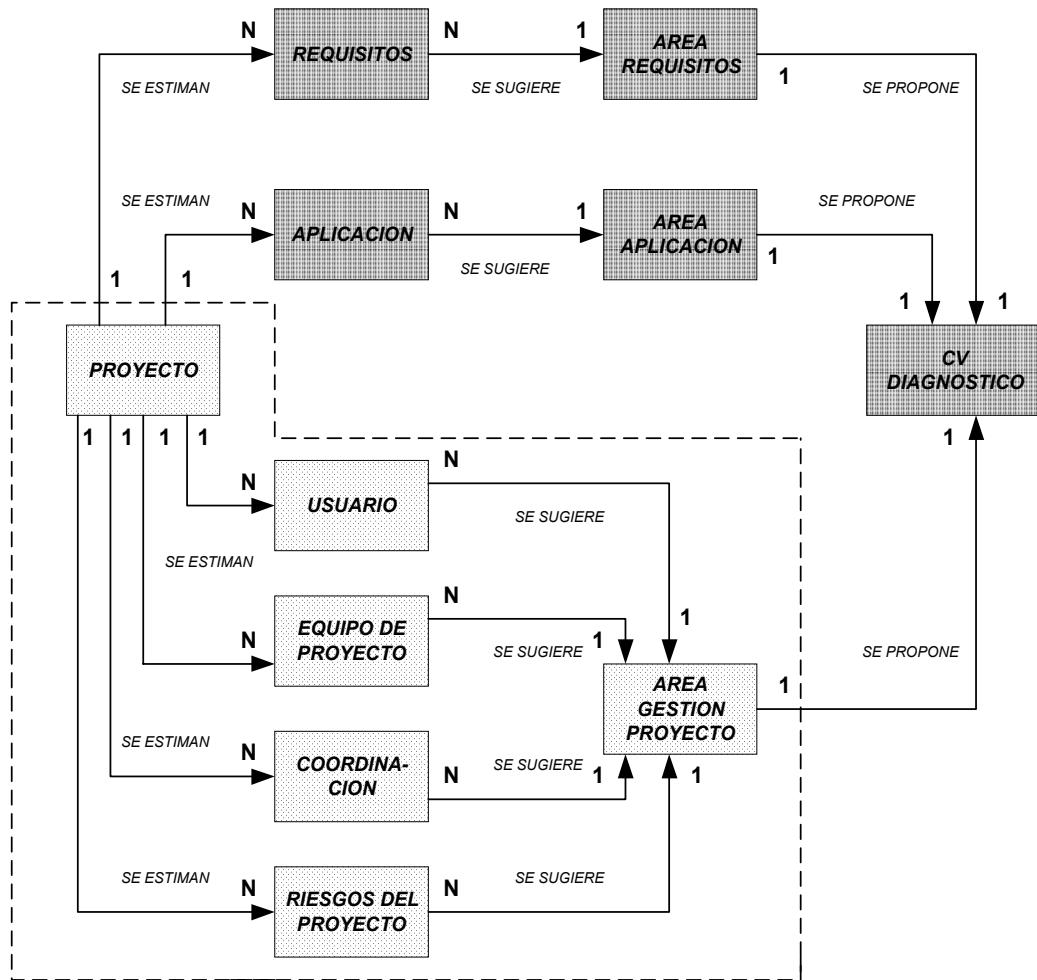


FIGURA 6-11 Conceptos Intervinientes en el Proceso
Seleccionar el CV del área Gestión de Proyecto

Estimar Factores Administración	
Propósito:	Determinar la existencia de entregas intermedias y el nivel de reconocimientos conveniente de la metodología a usar.
Información necesaria:	Los valores asignados por el usuario del sistema a los atributos formalidad, Entregas, Factibilidad Metodología, Necesidad Metodología.
Acciones:	<ol style="list-style-type: none"> 1. Solicitar al usuario los valores de los atributos necesarios 2. Deducir si existen entregas intermedias <ol style="list-style-type: none"> 2.1 Entregas es versión temprana o versión gradual o versión parcial = Entrega intermedia existe 3. Deducir conveniencia de usar metodología <ol style="list-style-type: none"> 3.1 Si Factibilidad metodología es ampliamente probada y Necesidad Metodología es ampliamente probada = Conveniencia metodología es ampliamente probada. 3.2 Si Factibilidad metodología es medianamente probada y Necesidad Metodología es medianamente probada = Conveniencia metodología es medianamente probada. 3.3 Si Factibilidad metodología es poco probada y Necesidad Metodología es poco probada = Conveniencia metodología es poco probada.

Tabla 6-205 Descripción del Proceso Estimar Factores Administración

Analizar Coordinación
<p>Propósito: <i>Analizar los factores que inciden en las decisiones de coordinación en el desarrollo de un proyecto.</i></p>
<p>Información necesaria: <i>Los valores asignados por el usuario del sistema a los atributos Formalidad, Entregas, Control Gestión, Responsabilidad, factibilidad Metodología, Necesidad Metodología, Reúso aplicación OO, Reúso aplicación existente, Reúso aplicación futura, Procedimientos cambios, Software para prototipar, Procedimientos Desarrollo-Mantenimiento..</i></p>
<p>Acciones:</p> <ol style="list-style-type: none"> 1. Solicitar al usuario los valores de los atributos necesarios. 2. Estimar Factores Reúso <ul style="list-style-type: none"> 2.1. Deducir aplicación previa desarrollada en objetos 2.2. Deducir si el modelo de objetos es aplicable 3. Estimar Factores mantenimiento <ul style="list-style-type: none"> 3.1 Deducir si el modelo en espiral es aplicable 4. Estimar Recursos software <ul style="list-style-type: none"> 4.1 Deducir si el prototipo es factible 5. Estimar factores Administración. <ul style="list-style-type: none"> 5.1 Deducir si existen entregas intermedias 5.2 Deducir conveniencia de usar metodología

Tabla 6-206 Descripción del Proceso Analizar Coordinación

Estimar Relación con Usuario
<p>Propósito: <i>Determinar el grado y tipo de relación que se espera del usuario para deducir la necesidad de usar un sistema de prototipación en el desarrollo de la aplicación.</i></p>
<p>Información necesaria: <i>Los valores asignados por el usuario del sistema a los atributos Participación e Introducción gradual.</i></p>
<p>Acciones:</p> <ol style="list-style-type: none"> 1. Solicitar al usuario los valores de los atributos necesarios 2. Deducir si el modelo por prototipo es aplicable <ul style="list-style-type: none"> 2.1 Participación es fuerte = Aplicabilidad Prototipo es aplicable 2.2 Introducción gradual es necesaria = Aplicabilidad Prototipo es aplicable

Tabla 6-207 Descripción del Proceso Estimar Relación con Usuario

Estimar Factores Equipo de Proyecto
<p>Propósito: <i>Determinar el grado y tipo de experiencia (en sistemas similares y en las técnicas de ingeniería de software a usar) con que cuenta el equipo afectado al desarrollo del sistema, para deducir la necesidad de usar un sistema de prototipación en el desarrollo de la aplicación.</i></p>
<p>Información necesaria: <i>Los valores asignados por el usuario del sistema a los atributos Experiencia técnicas IS, Experiencia previa.</i></p>
<p>Acciones:</p> <ol style="list-style-type: none"> 1. Solicitar al usuario los valores de los atributos necesarios 2. Deducir si el modelo por prototipo es aplicable <ul style="list-style-type: none"> 2.1 Experiencia técnicas IS es no hay = Aplicabilidad Prototipo es aplicable 2.2 Experiencia previa es no hay = Aplicabilidad Prototipo es aplicable

Tabla 6-208 Descripción del Proceso Estimar Factores Equipo de Proyecto

Estimar Riesgo Existente
Propósito: <i>Determinar la existencia del factor de riesgo y el nivel de aceptabilidad del riesgo para cada tipo de CV.</i>
Información necesaria: <i>Los valores asignados por el usuario del sistema a los atributos Viabilidad software, Nivel de riesgo, Categorías riesgo, Objetivo calidad, Terminación proyectos y valores existentes en la Base de conocimientos para Habilidad Riesgo</i>
Acciones: <ol style="list-style-type: none"> 1. Solicitar al usuario los valores de los atributos necesarios 2. Deducir existencia del factor de riesgo <ul style="list-style-type: none"> 2.1 Viabilidad software es hay dudas o Categorías riesgo es técnicos = Factor riesgo existe 3. Deducir el modelo aceptable de acuerdo al nivel del factor de riesgo. <ul style="list-style-type: none"> 3.1 Nivel de riesgo es alto o mediano = Riesgo espiral es aceptable 3.2 Nivel de riesgo es mediano o bajo o no hay = Riesgo objetos es aceptable 3.3 Nivel de riesgo es bajo o no hay = Riesgo cascada es aceptable 4. Deducir aplicabilidad del modelo en espiral. <ul style="list-style-type: none"> 4.1 Objetivo calidad es mecanismos explícitos y Factor riesgo existe = Aplicabilidad espiral es aplicable. 4.2 Terminación proyectos es prematura y Habilidad riesgo es existe = Aplicabilidad espiral es aplicable.
<i>Tabla 6-209 Descripción del Proceso Estimar Riesgo Existente</i>

Estimar Habilidad para Evaluar Riesgo
Propósito: <i>Determinar si se cuenta con la habilidad para la evaluación del riesgo y la aplicabilidad del modelo en espiral..</i>
Información necesaria: <i>Los valores asignados por el usuario del sistema a los atributos Análisis riesgo, Técnicas AR, Identificación alternativas, Riesgos Alternativa y valores existentes en la Base de conocimientos para Factor Riesgo</i>
Acciones: <ol style="list-style-type: none"> 1. Solicitar al usuario los valores de los atributos necesarios 2. Deducir habilidad para evaluar riesgo <ul style="list-style-type: none"> 2.1 Identificación Alternativas es se identifican y Riesgos alternativa es se identifican = Habilidad riesgo es existe 3. Deducir aplicabilidad del modelo en espiral. <ul style="list-style-type: none"> 3.1 Habilidad riesgo es existe y Identificación alternativas es etapas anteriores = Aplicabilidad espiral es aplicable 3.2 Habilidad riesgo es existe y Riesgos alternativa es etapas anteriores = Aplicabilidad espiral es aplicable 3.3 Factor riesgo es existe y Análisis riesgo es estrategias explícitas = Aplicabilidad espiral es aplicable 4. Deducir si el riesgo es evaluable para el modelo en espiral. <ul style="list-style-type: none"> 4.1 Factor Riesgo es existe y Habilidad Riesgo es existe y Técnicas AR es se dispone = Riesgo es evaluable.
<i>Tabla 6-210 Descripción del Proceso Estimar Habilidad para Evaluar Riesgo</i>

Analizar Riesgos del Proyecto
<p>Propósito: <i>Analizar los factores que inciden en las decisiones de coordinación en el desarrollo de un proyecto.</i></p>
<p>Información necesaria: <i>Los valores asignados por el usuario del sistema a los atributos Viabilidad software, Categorías riesgo, Nivel de riesgo, Objetivo Calidad, Análisis riesgo, Técnicas AR, Identificación Alternativas, Riesgos alternativa, Terminación proyectos. Valores existentes en la Base de conocimientos para Factor Riesgo, Riesgo Cascada, Riesgo objetos, Habilidad Riesgo, Factor riesgo, Riesgo espiral, Riesgo.</i></p>
<p>Acciones:</p> <ol style="list-style-type: none"> 1. Solicitar al usuario los valores de los atributos necesarios. 2. Estimar Riesgo existente <ol style="list-style-type: none"> 2.1 Deducir existencia del factor de riesgo 2.2 Deducir el modelo aceptable de acuerdo al nivel del factor de riesgo. 2.3 Deducir aplicabilidad del modelo en espiral. 3. Estimar Habilidad para evaluar riesgo <ol style="list-style-type: none"> 3.1 Deducir habilidad para evaluar riesgo 3.2 Deducir aplicabilidad del modelo en espiral. 3.3 Deducir si el riesgo es evaluable para el modelo en espiral.

Tabla 6-211 Descripción del Proceso Analizar Riesgos del Proyecto

Seleccionar el CV del Área Gestión del Proyecto
<p>Propósito: <i>Deducir el ciclo de vida para el área Gestión del Proyecto.</i></p>
<p>Información necesaria: <i>Los valores asignados por el usuario del sistema y deducidos para los atributos de los conceptos Coordinación, Usuario, Equipo de proyecto, Riesgos del proyecto.</i></p>
<p>Acciones:</p> <ol style="list-style-type: none"> 1. Deducir si el CV del área es Espiral <ol style="list-style-type: none"> 1.1 Aplicabilidad Prototipo es aplicable + Factibilidad Prototipo es factible + Riesgo es Evaluable + Aplicabilidad espiral es aplicable = CV Propuesto Gestión es Espiral 2. Deducir si el CV del área es Objetos <ol style="list-style-type: none"> 2.1 Riesgo Objetos aceptable + Sistema OO existe + Aplicabilidad OO aplicable = CV Propuesto Gestión Objetos 3. Deducir si el CV del área es Cascada <ol style="list-style-type: none"> 3.4 Factibilidad Prototipo no factible + Riesgo Cascada aceptable = CV Propuesto Gestión es Cascada 3.5 Riesgo Cascada aceptable + Entregas no versión temprana + Conveniencia Metodología ampliamente probada = CV Propuesto Gestión es Cascada 3.6 Riesgo Cascada aceptable + Formalidad muy formal + Entregas no versión temprana = CV Propuesto Gestión es Cascada 3.4 Conveniencia Metodología ampliamente probada + Responsabilidad Terceros + Control gestión muy ajustado = CV Propuesto Gestión es Cascada

Tabla 6-212 Descripción del Proceso Seleccionar el CV del Área Gestión del Proyecto

6.3.3 MAPA DE CONOCIMIENTOS

El Mapa de Conocimientos es un método para representar en dos dimensiones las conexiones que efectúa el cerebro cuando entiende hechos acerca de algo. Por lo tanto, representa el proceso de inferir valores de los

atributos. Los enlaces entre los atributos y los valores inferidos forman una parte importante de los conocimientos [Gómez, A. y otros 1997].

Las figuras comprendidas entre 6-12 y 6-19 describen los mapas de conocimiento que se han construido para la representación del problema. Tanto el experto como el grupo de expertos han identificado tres áreas que son las que se han tenido en cuenta para la construcción del Mapa, lo que ha facilitado la evaluación de estos subproblemas a resolver.

- Área Especificación de Requerimientos: figura 6-12
- Área Tipo de Aplicación: figuras 6-13 y 6-14
- Área Gestión de Proyecto: figuras 6-15, 6-16, 6-17, 6-18
- El Proyecto en su visión global: 6-19. En el caso de esta figura se ha identificado con línea punteada tanto las cajas, como las flechas que representan los atributos del concepto Proyecto, ya que si bien identifican a cada uno de los proyectos que se están analizando, el valor de los atributos no incide en el razonamiento del Sistema Experto.

6.4 COMPROBACION DE LA Conceptualización

Si bien se han realizado comprobaciones sobre los distintos modelos construidos en la fase de Conceptualización, se puso especial atención en la comprobación del Mapa de Conocimientos para verificar la consistencia entre los modelos. Los pasos seguidos en la comprobación fueron:

- Se verificó que todos los atributos de la periferia eran ingresados por el usuario o provenían de un archivo externo
- Se verificó que los atributos inferidos no fueran subjetivos y que estuvieran contenidos en la reglas
- Se verificó que los valores desconocidos de los atributos fueran un valor por omisión
- Se comprobó que los valores de los atributos sean usados. Los que no son usados en esta versión del prototipo se los dejó como parte del modelo conceptual a considerar en próximas versiones del prototipo
- Se completó y verificó que todos los atributos (periféricos o inferidos) se encuentren en la tabla de Concepto-Atributo-Valor

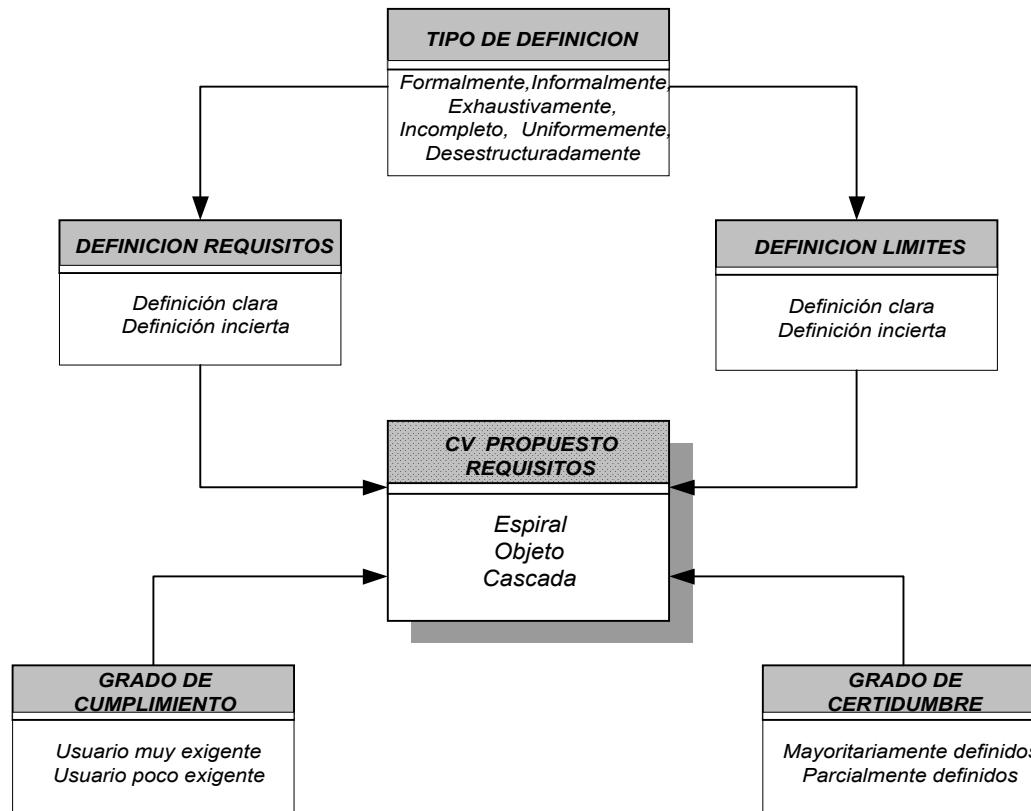


FIGURA 6-12 Mapa de Conocimientos del Área Especificación Requerimientos

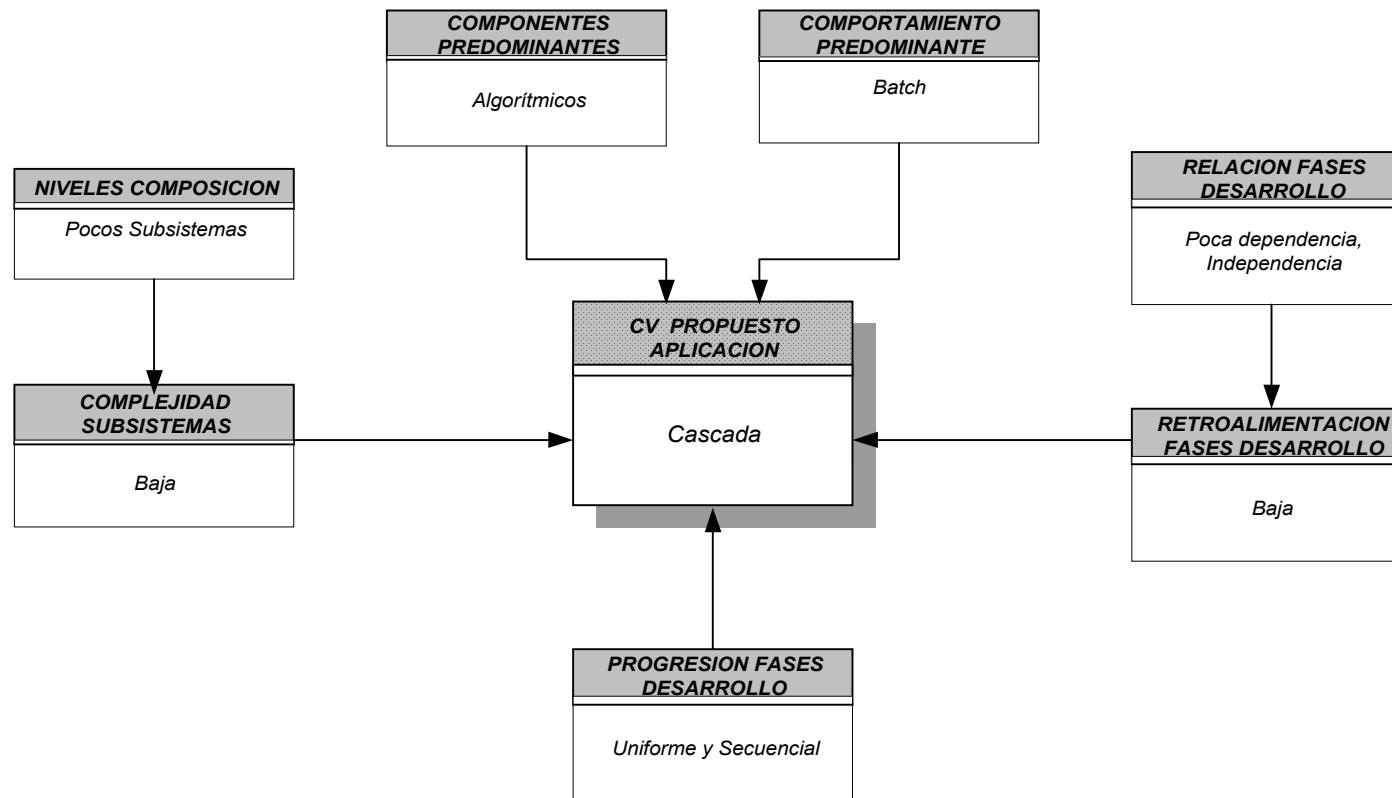


FIGURA 6-13 Mapa de Conocimientos del Área Tipo Aplicación - CV en Cascada

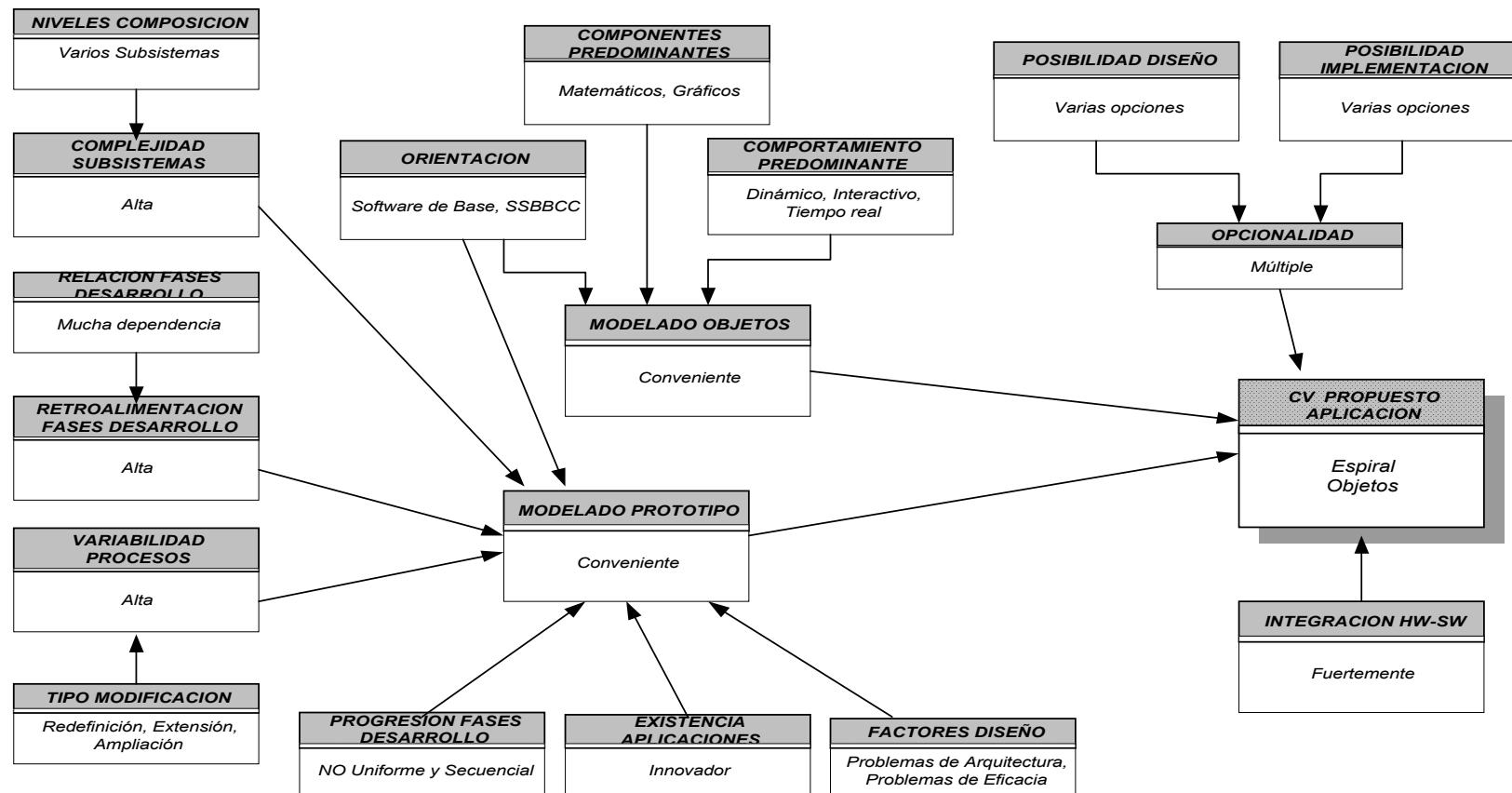


FIGURA 6-14 Mapa de Conocimientos del Área Tipo Aplicación - CV en Objetos y Espiral

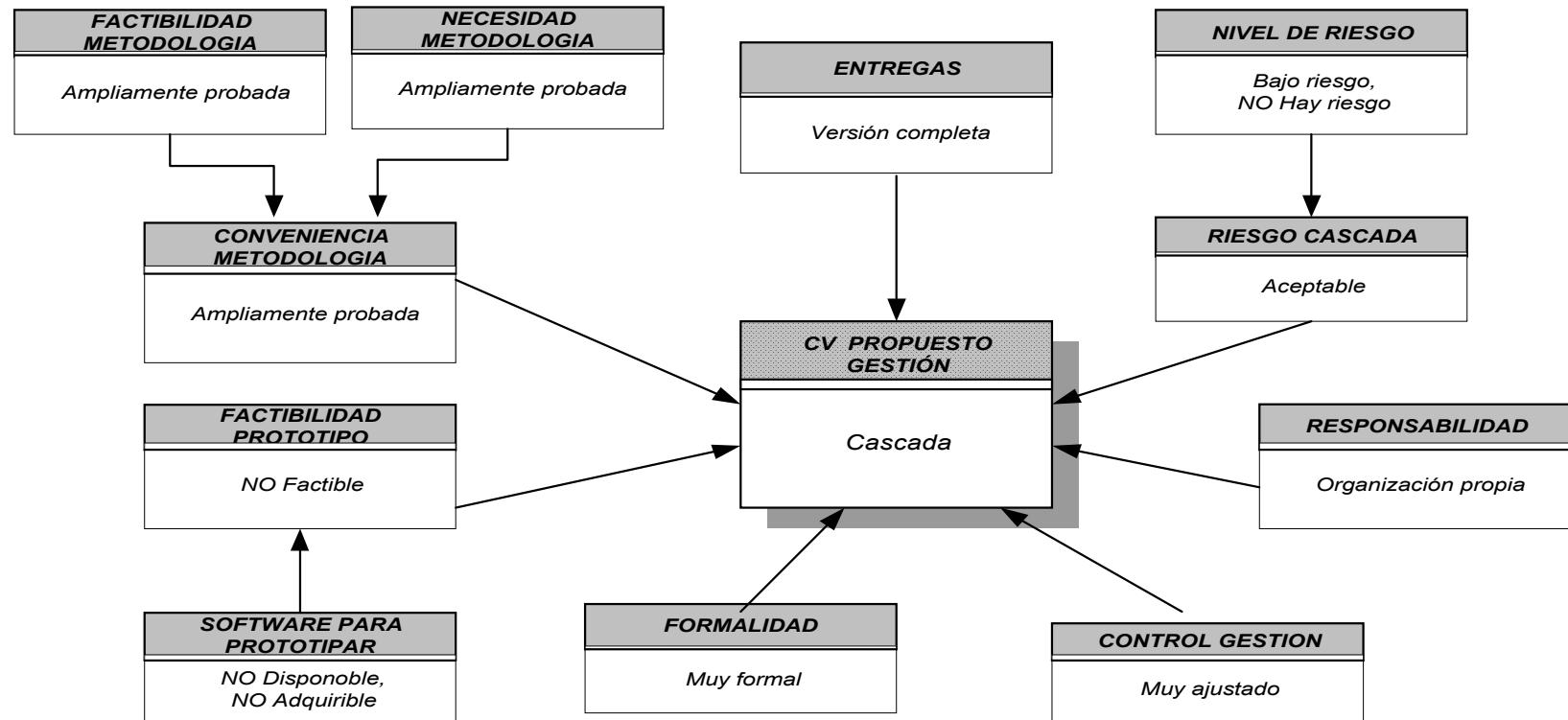


FIGURA 6-15 Mapa de Conocimientos del Área Gestión de Proyectos - CV en Cascada

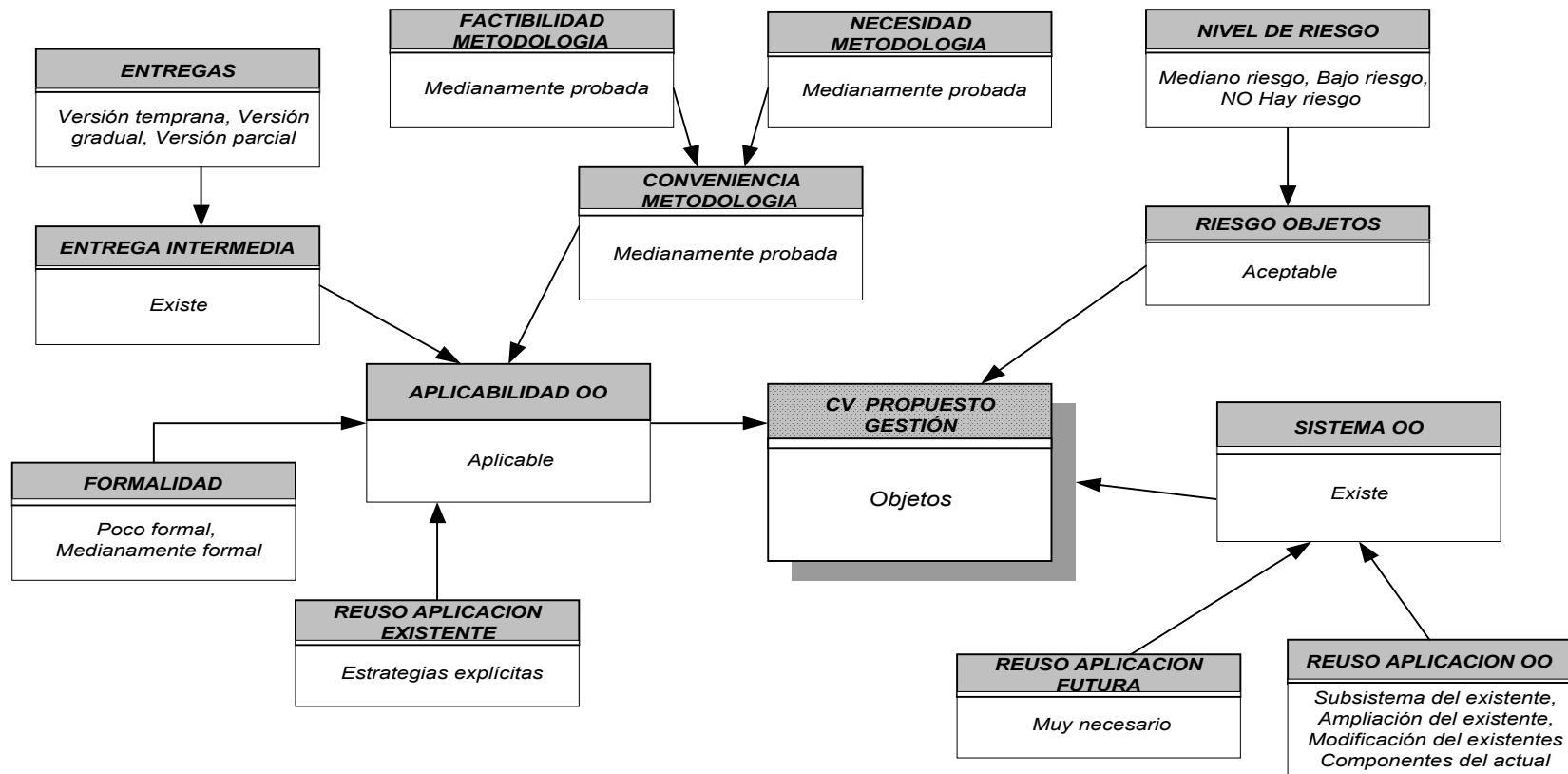


FIGURA 6-16 Mapa de Conocimientos del Área Gestión de Proyectos - CV en Objetos

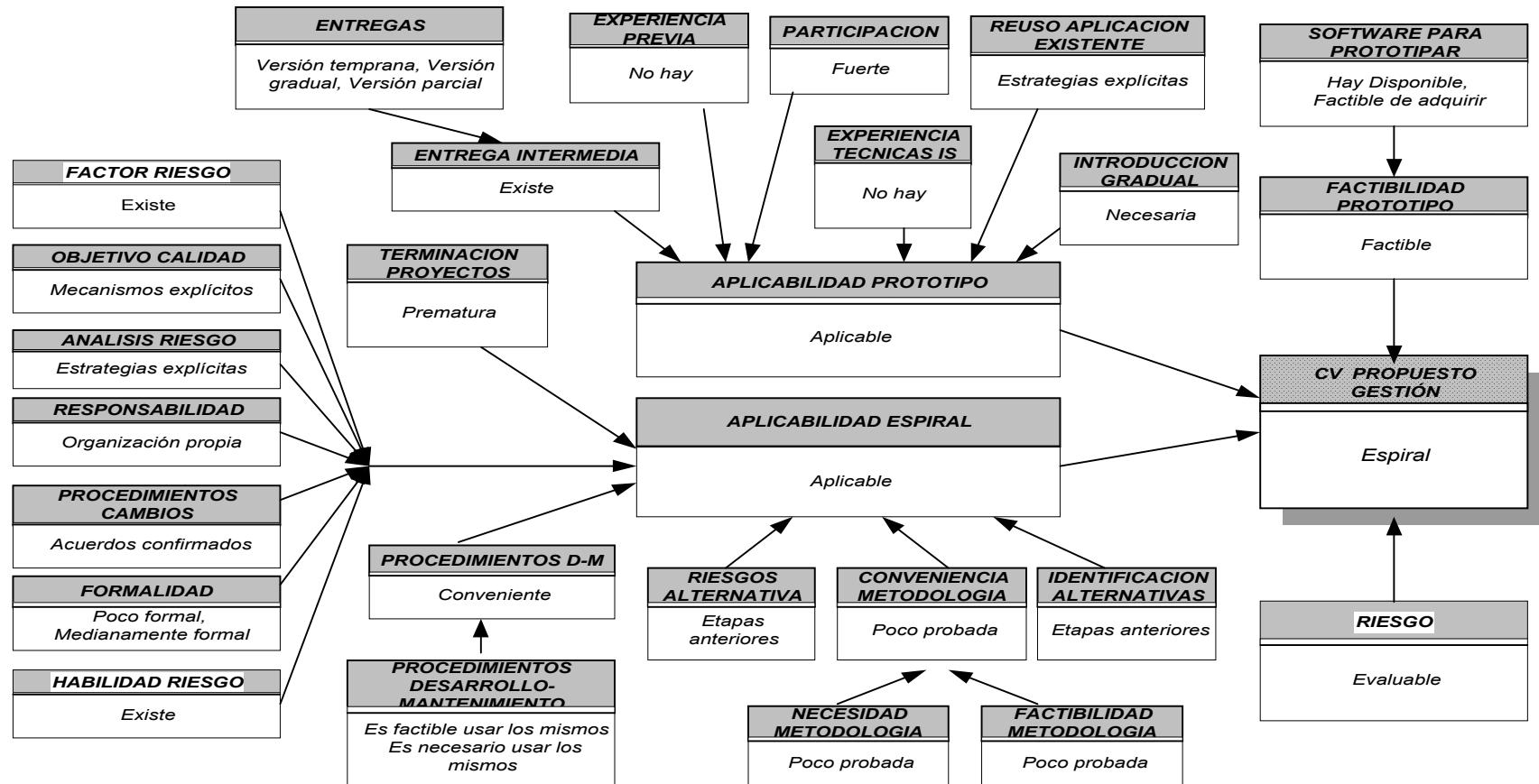


FIGURA 6-17 Mapa de Conocimientos del Área Gestión de Proyectos - CV en Espiral

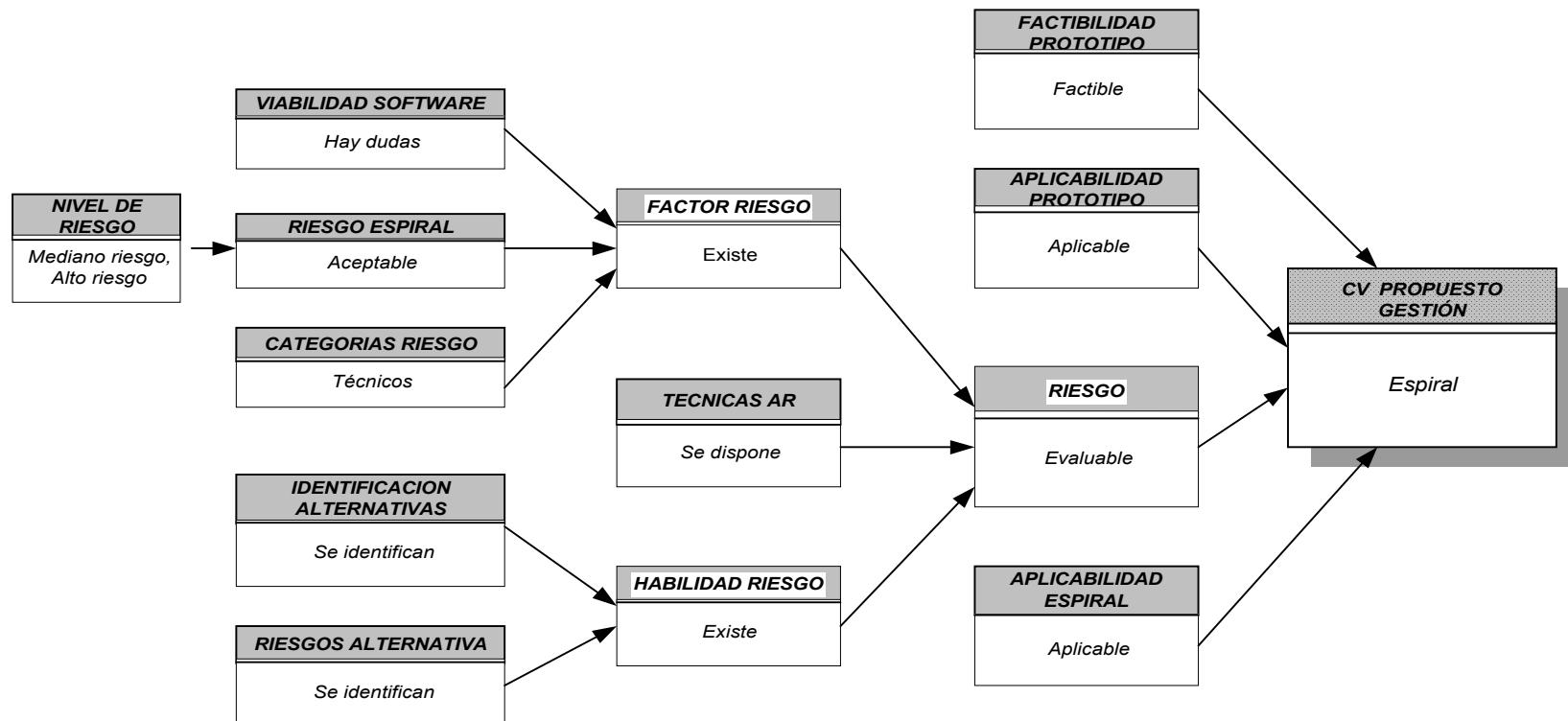


FIGURA 6-18 Continuación figura 6-17
Mapa de Conocimientos del Área Gestión de Proyectos - CV en Espiral

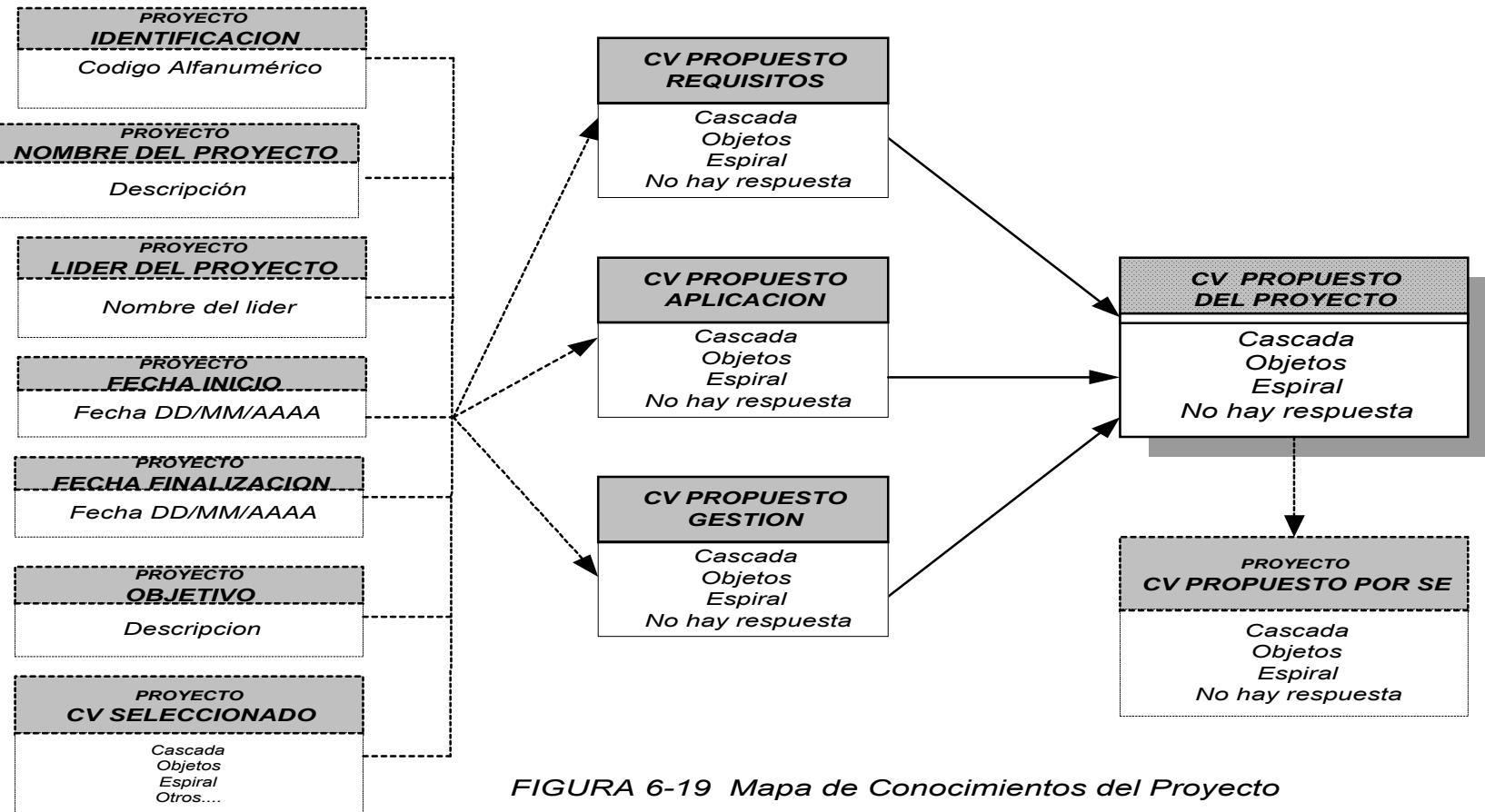


FIGURA 6-19 Mapa de Conocimientos del Proyecto

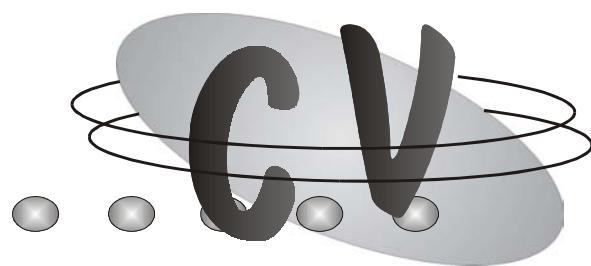
Los casos de prueba que se usaron para validar el Modelo Conceptual se detallan en el capítulo 9. Se usaron los casos 1.1; 1.2; 2; 3; 4; 6. El resultado de la comprobación fue el siguiente:

- Los casos, 1.1, 1.2, 2, 4 coincidieron perfectamente en su resolución por el experto y por el Mapa .
- Para el caso de prueba 3 se corrigió un error de tipo en las reglas, pero eso llevó a revisar todo el conjunto de reglas como verificación.
- Para el caso de prueba 6 se realizaron cambios en la modificación de los valores de algunos atributos.

El resultado de la comprobación fue altamente satisfactorio ya que el Modelo Conceptual reflejó el razonamiento del experto.

Capítulo 7

Formalización de Conocimientos



Los conocimientos que en la fase de Conceptualización fueron expresados en representaciones externas intermedias: árbol de descomposición funcional, seudorreglas y tablas de concepto-atributo-valor se representan en la fase de Formalización en representaciones compatibles con la tecnología computacional. En primer lugar se seleccionan los formalismos a utilizar. Luego se formalizan los conocimientos que serán posteriormente implementados en la herramienta seleccionada.

7.1 SELECCIÓN DE FORMALISMOS

Formalizar consiste en representar simbólicamente los conocimientos mediante alguno de los formalismos existentes, organizarlos de acuerdo con algún modelo de diseño y determinar los métodos de inferencia adecuados para manejar eficientemente y efectivamente dichos conocimientos [Gómez, A. y otros 1997]. Teniendo en cuenta los diversos tipos de formalismos existentes en la bibliografía de referencia [Gómez, A. y otros 1997], los formalismos de representación usados en la etapa de Conceptualización y que uno de los objetivos del presente trabajo es construir el sistema experto usando la herramienta Kappa los formalismos seleccionados son:

- Reglas de producción para las seudorreglas ya que su estructura es la misma
- Marcos para la tabla Concepto-atributo-valor.
- Procedimientos para los procesos a realizar

7.2 FORMALIZACION DE LOS CONOCIMIENTOS EN REGLAS DE PRODUCCION

Las seudorreglas han sido definidas en detalle en el capítulo de Conceptualización. Considerando que la formalización de las seudorreglas en reglas de producción es muy semejante a las ya detalladas en el capítulo anterior se ha realizado en un solo paso la formalización y la implementación de las reglas directamente en la herramienta Kappa-PC.

7.3 FORMALIZACION DE LOS CONOCIMIENTOS EN MARCOS

Cada concepto de la tabla Concepto-Atributo-Valor se formaliza en un marco clase, siguiendo la propuesta sugerida por Gómez, [Gómez, A. y otros 1997]. Las ranuras de esos marcos son los atributos de cada concepto. Los demonios de esos marcos precisan cuando y cómo se obtiene el valor de esas ranuras.

MC PROYECTO	Tipo Ranura	Min/ Max	Multiv.	Propiedad General	Valores Permitidos	Valor Omisión	Si Necesito	Si Modifico	Si Borro
<i>Ranura</i>									
(*) Identificación	Numérico	1/1	No	—	Entero > 0	—	PROC. Guardar	—	—
(*) Nombre proyecto	Conj. de caracteres	1/1	No	—	—	—	PROC. Guardar	—	—
(*) Líder del proyecto	Conj. de caracteres	1/1	No	—	—	—	PROC. Guardar	—	—
(*) Fecha inicio	Fecha	1/1	No	—	dd/mm/aa	—	PROC. Guardar	—	—
(*) Fecha finalización	Fecha	1/1	No	—	dd/mm/aa	—	PROC. Guardar	—	—
(*) Objetivo	Conj. de caracteres	1/n	Sí	—	—	—	PROC. Guardar	—	—
(*) CV seleccionado	Conj. de caracteres	1/1	No	—	—	—	PROC. Guardar	—	—
Requisitos	Marco	1/n	Sí	^MC Requisitos	—	—	—	—	—
Aplicación	Marco	1/n	Sí	^MC Aplicación	—	—	—	—	—
Gestión	Marco	1/n	Sí	^MC Gestión	—	—	—	—	—
CV Diagnóstico	Marco	1/n	Sí	^MC CV Diagnóstico	—	—	—	—	—

Tabla 7-1 Marco Clase Proyecto

MC GESTION	Tipo Ranura	Min/Max	Multiv.	Propiedad General	Valores Permitidos	Valor Omisión	Si Necesito	Si Modifico	Si Borro
<i>Ranura</i>									
Coordinación	Marco	1/n	Sí	^MC Coordinación	—	—	—	—	—
Usuario	Marco	1/n	Sí	^MC Usuario	—	—	—	—	—
Riesgos del proyecto	Marco	1/n	Sí	^MC Riesgos del proyecto	—	—	—	—	—
Equipo de proyecto	Marco	1/n	Sí	^MC Equipo de proyecto	—	—	—	—	—

Tabla 7-2 Marco Clase Gestión

MC REQUISITOS	<i>Tipo Ranura</i>	<i>Min/ Max</i>	<i>Multiv.</i>	<i>Propiedad General</i>	<i>Valores Permitidos</i>	<i>Valor Omisión</i>	<i>Si Necesito</i>	<i>Si Modifico</i>	<i>Si Borro</i>
(*)Tipo de Definición	Conj. de caracteres	1/n	Sí	—	Formalmente Informalmente Exhaustivamente Incompleto Uniformemente Desestructuradamente	—	PROC. <i>PreguntarTipoDefinición</i>	—	—
(*)Definición Requisitos	Conj. de caracteres	1/n	Sí	—	Definición Clara Definición Incierta Alta incertidumbre Baja incertidumbre	—	—	—	—
(*) Definición Límites	Conj. de caracteres	1/n	Sí	—	Definición Clara Definición Incierta Alta incertidumbre Baja incertidumbre	—	—	—	—
(*)Grado de certidumbre	Conj. de caracteres	1/1	No	—	Mayoritariamente definidos Parcialmente definidos	—	PROC. <i>ValoresExcluyentes</i>	—	—
(*)Grado de cumplimiento	Conj. de caracteres	1/1	No	—	Usuario muy exigente Usuario poco exigente	—	PROC. <i>ValoresExcluyentes</i>	—	—

Tabla 7-3 Marco Clase Requisitos

MC APLICACION	Tipo Ranura	Min/Max	Multiv.	Propiedad General	Valores Permitidos	Valor Omisión	Si Necesito	Si Modifico	Si Borro
<i>Ranura</i>									
(*) Niveles de Composición	Conj. de caracteres	1/1	No	—	Pocos Subsistemas Varios Subsistemas	—	PROC. ValoresExcluyentes	—	—
(*) Complejidad Subsistemas	Conj. de caracteres	1/1	No	—	Baja Alta	—	—	—	—
(*) Componentes Predominantes	Conj. de caracteres	1/n	Sí	—	Algorítmicos Matemáticos Gráficos	—	PROC. ValoresNoExcluyentes	—	—
(*) Comportamiento Predominante	Conj. de caracteres	1/n	Sí	—	Batch Dinámico Interactivo Tiempo Real	—	PROC. PreguntarComportamientoPredominante	—	—
(*) Orientación	Conj. de caracteres	1/1	No	—	Software de Base Sistema Basado en Conocimiento Otros	—	PROC. ValoresExcluyentes	—	—
(*) Progresión Fases desarrollo	Conj. de caracteres	1/1	No	—	Uniforme y Secuencial No Uniforme y Secuencial	—	PROC. ValoresExcluyentes	—	—
(*) Relación Fases desarrollo	Conj. de caracteres	1/1	No	—	Poca Dependencia Mucha Dependencia Independencia	—	PROC. ValoresExcluyentes	—	—
(*) Retroalimentación Fases desarrollo	Conj. de caracteres	1/1	No	—	Baja Alta	—	—	—	—
(*) Integración HW-SW	Conj. de caracteres	1/1	No	—	Fuertemente Medianamente Levemente	—	PROC. ValoresExcluyentes	—	—
(*) Factores diseño	Conj. de caracteres	1/n	Sí	—	Problemas de Arquitectura Problemas de eficacia	—	PROC. ValoresNoExcluyentes	—	—

Tabla 7-4a Marco Clase Aplicación

MC APLICACION	Tipo Ranura	Min/Max	Multiv.	Propiedad General	Valores Permitidos	Valor Omisión	Si Necesito	Si Modifico	Si Borro
<i>Ranura</i>									
(*) Existencia Aplicaciones	Conj. de caracteres	1/1	No	—	Innovador Conocido	—	PROC. ValoresExcluyentes	—	—
(*) Posibilidad Diseño	Conj. de caracteres	1/1	No	—	Pocas opciones Varias opciones	—	PROC. ValoresExcluyentes	—	—
(*) Posibilidad Implementación	Conj. de caracteres	1/1	No	—	Pocas opciones Varias opciones	—	PROC. ValoresExcluyentes	—	—
(*) Modelado Prototipo	Conj. de caracteres	1/1	No	—	Conveniente No conveniente	—	—	—	—
(*) Modelado Objetos	Conj. de caracteres	1/1	No	—	Conveniente No conveniente	—	—	—	—
(*) Variabilidad Procesos	Conj. de caracteres	1/1	No	—	Alta Baja	—	—	—	—
(*) Opcionalidad	Conj. de caracteres	1/1	No	—	Múltiple Simple	—	—	—	—
(*) Tipo modificación	Conj. de caracteres	1/n	Sí	—	Redefinición Extensión Ampliación Emisión gradual	—	PROC. ValoresNoExcluyentes	—	—

Tabla 7-4b Marco Clase Aplicación

MC COORDINACION Ranura	Tipo Ranura	Min/ Max	Multiv.	Propiedad General	Valores Permitidos	Valor Omisión	Si Necesito	Si Modifico	Si Borro
(*) Formalidad	Conj. de caracteres	1/1	No	—	Poco formal Medianamente formal Muy formal	—	PROC. ValoresExcluyentes	—	—
(*) Entregas	Conj. de caracteres	1/1	No	—	Versión temprana No versión temprana Versión gradual Versión parcial	—	PROC. Entregas	—	—
(*) Responsabilidad	Conj. de caracteres	1/1	No	—	Terceros Organización propia	—	PROC. ValoresExcluyentes	—	—
(*) Control Gestión	Conj. de caracteres	1/1	No	—	Muy ajustado Medianamente ajustado Poco ajustado	—	PROC. ValoresExcluyentes	—	—
(*) Software para prototipar	Conj. de caracteres	1/n	Sí	—	No disponible Hay disponible Factible de adquirir No adquirible	—	—	—	—
(*) Factibilidad metodología	Conj. de caracteres	1/1	No	—	Ampliamente probada Medianamente probada Poco probada	—	PROC. ValoresExcluyentes	—	—
(*) Necesidad Metodología	Conj. de caracteres	1/1	No	—	Ampliamente probada Medianamente probada Poco probada	—	PROC. ValoresExcluyentes	—	—
(*) Reúso aplicación existente	Conj. de caracteres	1/1	No	—	Estrategias explícitas No estrategias explícitas	—	PROC. ValoresExcluyentes	—	—

Tabla 7-5a Marco Clase Coordinación

MC COORDINACION Ranura	Tipo Ranura	Min/ Max	Multiv.	Propiedad General	Valores Permitidos	Valor Omisión	Si Necesito	Si Modifico	Si Borro
(*) Reúso aplicación OO	Conj. de caracteres	1/n	Sí	—	Subsistema del existente Ampliación del existente Modificación del existente Componentes del actual	—	PROC. ValoresNoExcluyentes	—	—
(*) Reúso aplicación futura	Conj. de caracteres	1/1	No	—	Muy necesario Medianamente necesario Poco necesario	—	PROC. ValoresExcluyentes	—	—
(*) Procedimientos Cambios	Conj. de caracteres	1/1	No	—	Acuerdos confirmados No Acuerdos confirmados	—	PROC. ValoresExcluyentes	—	—
(*) Procedimientos Desarrollo-Mantenimiento	Conj. de caracteres	1/n	Sí	—	Factible usar los mismos Factible usar diferentes Necesario usar los mismos Necesario usar diferentes	—	PROC. ValoresNoExcluyentes	—	—
(*) Conveniencia Metodología	Conj. de caracteres	1/1	No	—	Ampliamente probada Medianamente probada Poco probada	—	—	—	—
(*) Factibilidad Prototipo	Conj. de caracteres	1/1	No	—	Factible No factible	—	—	—	—
(*) Sistema OO	Conj. de caracteres	1/1	No	—	Existe No existe	—	—	—	—

Tabla 7-5b Marco Clase Coordinación

MC COORDINACION Ranura	Tipo Ranura	Min/ Max	Multiv.	Propiedad General	Valores Permitidos	Valor Omisión	Si Necesito	Si Modifico	Si Borro
(*) Entrega Intermedia	Conj. de caracteres	1/1	No	—	Existe No existe	—	—	—	—
(*) Aplicabilidad Espiral	Conj. de caracteres	1/1	No	—	Aplicable No aplicable	—	—	—	—
(*) Aplicabilidad Prototipo	Conj. de caracteres	1/1	No	—	Aplicable No aplicable	—	—	—	—
(*) Aplicabilidad OO	Conj. de caracteres	1/1	No	—	Aplicable No aplicable	—	—	—	—
(*) Procedimientos D-M	Conj. de caracteres	1/1	No	—	Conveniente No conveniente	—	PROC. ValorInconsistente	—	—

Tabla 7-5c Marco Clase Coordinación

MC USUARIO Ranura	Tipo Ranura	Min/ Max	Multiv.	Propiedad General	Valores Permitidos	Valor Omisión	Si Necesito	Si Modifico	Si Borro
(*) Participación	Conj. de caracteres	1/1	No	—	Fuerte Regular Poca	—	PROC. ValoresExcluyentes	—	—
(*) Introducción gradual	Conj. de caracteres	1/1	No	—	Necesaria No necesaria	—	—	—	—

Tabla 7-6 Marco Clase Usuario

MC RIESGOS DEL PROYECTO <i>Ranura</i>	Tipo Ranura	Min/ Max	Multiv.	Propiedad General	Valores Permitidos	Valor Omisión	Si Necesito	Si Modifico	Si Borro
(*) Viabilidad software	Conj. de caracteres	1/1	Sí	—	Hay certeza Hay dudas	—	PROC. ValoresExcluyentes	—	—
(*) Nivel de riesgo	Conj. de caracteres	1/1	No	—	Alto riesgo Mediano riesgo Bajo riesgo No hay riesgo	—	PROC. ValoresExcluyentes	—	—
(*) Análisis riesgo	Conj. de caracteres	1/1	No	—	Estrategias explícitas No estrategias explícitas	—	PROC. ValoresExcluyentes	—	—
(*) Técnicas AR	Conj. de caracteres	1/1	Sí	—	Se dispone No se dispone	—	PROC. ValoresExcluyentes	—	—
(*) Identificación Alternativas	Conj. de caracteres	1/1	No	—	Se identifican No se identifican Etapas anteriores	—	PROC. ValoresExcluyentes	—	—
(*) Riesgos Alternativa	Conj. de caracteres	1/1	No	—	Se identifican No se identifican Etapas anteriores	—	PROC. ValoresExcluyentes	—	—
(*) Categorías riesgo	Conj. de caracteres	1/n	No	—	Técnicos Otros riesgos	—	—	—	—
(*) Objetivo calidad	Conj. de caracteres	1/1	No	—	Mecanismos explícitos No mecanismos explícitos	—	PROC. ValoresExcluyentes	—	—
(*) Terminación proyectos	Conj. de caracteres	1/1	No	—	Prematura En término.	—	PROC. ValoresExcluyentes	—	—
(*) Riesgo Cascada	Conj. de caracteres	1/1	No	—	Aceptable No aceptable	—	—	—	—
(*) Riesgo Objetos	Conj. de caracteres	1/1	No	—	Aceptable No aceptable	—	—	—	—
(*) Habilidad Riesgo	Conj. de caracteres	1/1	No	—	Existe No existe	—	—	—	—

Tabla 7-7a Marco Clase Riesgos del Proyecto

MC RIESGOS DEL PROYECTO <i>Ranura</i>	Tipo Ranura	Min/ Max	Multiv.	Propiedad General	Valores Permitidos	Valor Omisión	Si Necesito	Si Modifico	Si Borro
(*) Factor Riesgo	Conj. de caracteres	1/1	No	—	Existe. No existe	—	—	—	—
(*) Riesgo Espiral	Conj. de caracteres	1/1	No	—	Aceptable No aceptable	—	—	—	—
(*) Riesgo	Conj. de caracteres	1/1	No	—	Evaluable No evaluable	—	—	—	—

Tabla 7-7b Marco Clase Riesgos del Proyecto

MC EQUIPO DE PROYECTO <i>Ranura</i>	Tipo Ranura	Min/ Max	Multiv.	Propiedad General	Valores Permitidos	Valor Omisión	Si Necesito	Si Modifico	Si Borro
(*) Experiencia técnicas IS	Conj. de caracteres	1/1	No	—	Hay No hay	—	PROC. ValoresExcluyentes	—	—
(*) Experiencia previa	Conj. de caracteres	1/1	No	—	Hay No hay	—	PROC. ValoresExcluyentes	—	—

Tabla 7-8 Marco Clase Equipo de proyecto

MC CV DIAGNOSTICO Ranura	Tipo Ranura	Min/ Max	Multiv.	Propiedad General	Valores Permitidos	Valor Omisión	Si Necesito	Si Modifico	Si Borro
(*) CV Propuesto Requisitos	Conj. de caracteres	1/n	No	—	Cascada Objetos Espiral No hay propuesta	—	—	—	—
(*) CV Propuesto Aplicación	Conj. de caracteres	1/n	No	—	Cascada Objetos Espiral No hay propuesta	—	—	—	—
(*) CV Propuesto Gestión	Conj. de caracteres	1/n	No	—	Cascada Objetos Espiral No hay propuesta	—	—	—	—
(*) CV Propuesto proyecto	Conj. de caracteres	1/1	No	—	Cascada Objetos Espiral No hay propuesta	—	—	—	—

Tabla 7-9 Marco Clase CV Diagnóstico

7.4 FORMALIZACION DE LOS CONOCIMIENTOS EN PROCEDIMIENTOS

A continuación se describen los procedimientos asociados a cada una de las propiedades representadas en los marcos que intervienen en la resolución del problema.

PROCEDIMIENTO *Proyecto:Guardar*

COMIENZO

```
{
Global:NOMBREBC = SelectFile( "Guardar Proyecto", "*.pro" );
If Not( Null?( Global:NOMBREBC ) )
Then {
  OpenWriteFile( Global:NOMBREBC );
  WriteLine( FormatValue( /*SISTEMA EXPERTO PARA LA SELECCION DE CICLOS DE
VIDA DE PROYECTOS DE SOFTWARE*/" ) );
  WriteLine( FormatValue( /*Copyright 2001 by Alberto Patron\n*/" ) );
  WriteLine( FormatValue( /*-----*/" ) );
  WriteLine( FormatValue( /*Proyecto: %s (%s)*/, Proyecto:Nombre,
Proyecto:ID ) );
  WriteLine( FormatValue( /*Fecha: %s*/, Date( ) ) );
  WriteLine( FormatValue( /*-----*/" ) );
  WriteInstance( Proyecto );
  ForAll [ x ]|Expresion ]
    WriteInstance( x );
  WriteLine( FormatValue( /*-----*/" ) );
  WriteLine( FormatValue( /*      Valor de las variables      */" ) );
  WriteLine( FormatValue( /*-----*/" ) );
  WriteLine( FormatValue( "Global:TA_Modif = %s;", Global:TA_Modif ) );
  WriteLine( FormatValue( "Global:TA_Concl = %s;", Global:TA_Concl ) );
  WriteLine( FormatValue( "Global:ER_Modif = %s;", Global:ER_Modif ) );
  WriteLine( FormatValue( "Global:ER_Concl = %s;", Global:ER_Concl ) );
  WriteLine( FormatValue( "Global:GP_Modif = %s;", Global:GP_Modif ) );
  WriteLine( FormatValue( "Global:GP_Concl = %s;", Global:GP_Concl ) );
  WriteLine( FormatValue( "Global:G_Concl = %s;", Global:G_Concl ) );
  WriteLine( FormatValue( "Global:Method = %s;", Global:Method ) );
  WriteLine( FormatValue( "Global:RE_Concl = %s;", Global:RE_Concl ) );
  WriteLine( FormatValue( "Global:recomendacion = %s;",
Global:recomendacion ) );
  CloseWriteFile( );
  PostMessage( "Proyecto guardado exitosamente" );
};

FIN
```

PROCEDIMIENTO ***Requisitos:PreguntarTipoDefinición***

```

COMIENZO
{
If LengthList(ER1:Valor)>0
Then
{
  If Member?(ER1:Valor,formalmente) And Member?(ER1:Valor,informalmente)
  Then
  {
    SetPostMessageTitle("Error en datos");
    PostMessage("Se seleccionaron dos valores contradictorios (formalmente,informalmente)");
  };

  If Member?(ER1:Valor,uniformemente) And Member?(ER1:Valor,desestructuradamente)
  Then
  {
    SetPostMessageTitle("Error en datos");
    PostMessage("Se seleccionaron dos valores contradictorios
(uniformemente,desestructuradamente)");
  };

  If Member?(ER1:Valor,exhaustivamente) And Member?(ER1:Valor,incompletos)
  Then
  {
    SetPostMessageTitle("Error en datos");
    PostMessage("Se seleccionaron dos valores contradictorios (exhaustivamente,incompletos)");
  };
};

FIN
}

```

PROCEDIMIENTO ***ValoresExcluyentes***

Este procedimiento esta preprogramado en Kappa-PC. Cuando se usa la opción de “SINGLE LIST BOX”, funcionalmente solo permite seleccionar un único valor, por lo tanto cumple con el procedimiento de ValoresExcluyentes.

PROCEDIMIENTO ***ValoresNoExcluyentes***

Este procedimiento esta preprogramado en Kappa-PC. Cuando se usa la opción de “MULTIPLE LIST BOX”, funcionalmente permite seleccionar varios valores, por lo tanto cumple con el procedimiento de ValoresNoExcluyentes.

PROCEDIMIENTO ***Aplicación:PreguntarComportamientoPredominante***

COMIENZO

```

{
If LengthList(ER1:Valor)>0
Then
{
  If Member?(TA51:Valor,batch) And Member?(TA5:Valor,interactivo)
  Then
  {
    SetPostMessageTitle("Error en datos");
    PostMessage("Se seleccionaron dos valores contradictorios (batch,interactivo)");
  };
  If Member?(TA51:Valor,batch) And Member?(TA5:Valor,tiempo_real)
  Then
  {
    SetPostMessageTitle("Error en datos");
    PostMessage("Se seleccionaron dos valores contradictorios (batch,tiempo_real)");
  };
  If Member?(TA51:Valor,batch) And Member?(TA5:Valor,dinámico)
  Then
  {
    SetPostMessageTitle("Error en datos");
    PostMessage("Se seleccionaron dos valores contradictorios (batch,dinámico)");
  };

  If Member?(TA51:Valor,ninguno) And Member?(TA5:Valor,interactivo)
  Then
  {
    SetPostMessageTitle("Error en datos");
    PostMessage("Se seleccionaron dos valores contradictorios (ninguno,interactivo)");
  };
  If Member?(TA51:Valor,ninguno) And Member?(TA5:Valor,tiempo_real)
  Then
  {
    SetPostMessageTitle("Error en datos");
    PostMessage("Se seleccionaron dos valores contradictorios (ninguno,tiempo_real)");
  };
  If Member?(TA51:Valor,ninguno) And Member?(TA5:Valor,dinámico)
  Then
  {
    SetPostMessageTitle("Error en datos");
    PostMessage("Se seleccionaron dos valores contradictorios (ninguno,dinámico)");
  };
};

};

FIN

```

PROCEDIMIENTO ***Coordinación:Entregas***

```

COMIENZO
{
If ( GP20:Valor #= versión_completa And GP18:Valor
    #= Si )
Then {
    SetPostMessageTitle( "Error en datos" );
    PostMessage( "Se seleccionaron dos valores contradictorios Pregunta 8= Si, Pregunta 6=
versión_completa" );
}
If ( GP20:Valor #= versión_parcial And GP18:Valor
    #= No )
Then {
    SetPostMessageTitle( "Error en datos" );
    PostMessage( "Se seleccionaron dos valores contradictorios Pregunta 8= No, Pregunta 6=
versión_parcial" );
}
If ( GP20:Valor #= versión_gradual And GP18:Valor
    #= No )
Then {
    SetPostMessageTitle( "Error en datos" );
    PostMessage( "Se seleccionaron dos valores contradictorios Pregunta 8= No, Pregunta 6=
versión_gradual" );
}
If ( GP20:Valor #= versión_temprana And GP18:Valor
    #= No )
Then {
    SetPostMessageTitle( "Error en datos" );
    PostMessage( "Se seleccionaron dos valores contradictorios Pregunta 8= No, Pregunta 6=
versión_temprana" );
}
};
FIN

```

PROCEDIMIENTO ***Coordinación:ValorInconsistente***

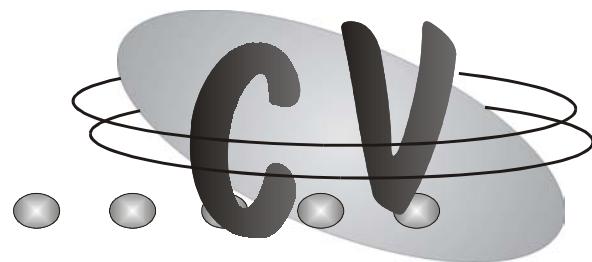
```

COMIENZO
{
If ( LengthList( GP26:Valor ) > 0 )
Then {
    If ( Member?( GP26:Valor, necesario )
        And Member?( GP26:Valor, indiferente ) )
    Then {
        SetPostMessageTitle( "Error en datos" );
        PostMessage( "Se seleccionaron dos valores contradictorios (necesario,indiferente)" );
    }
    If ( Member?( GP26:Valor, factible )
        And Member?( GP26:Valor, indiferente ) )
    Then {
        SetPostMessageTitle( "Error en datos" );
        PostMessage( "Se seleccionaron dos valores contradictorios (factible,indiferente)" );
    }
};
FIN

```

Capítulo 8

Implementación del Sistema



En este capítulo se describen en primer lugar las características de la herramienta que se usará para implementar el sistema experto. Luego usando un caso como ejemplo se presenta el sistema en sí, la interfaz y su operatoria general.

8.1 SELECCIÓN DE LA HERRAMIENTA

La herramienta seleccionada es Kappa PC desarrollada en 1990 por INTELLICORP Inc., que posee características altamente compatibles con la metodología que se ha aplicado en el desarrollo del sistema experto. Kappa brinda un entorno de desarrollo que permite prototipado rápido, lo que facilita un desarrollo basado en prototipado incremental que es lo que propone la metodología IDEAL. El ambiente de desarrollo es totalmente gráfico con una amplia gama de objetos que facilitan la operación de la aplicación. Tiene un entorno de programación orientada a objetos y facilidades para el desarrollo de las interfaces de usuario. Kappa tiene un lenguaje propio KAL con el que se escriben las funciones, los métodos y las reglas

Los formalismos de representación del conocimiento en Kappa son los objetos y las reglas. El motor de inferencias es de dos tipos:

- Encadenamiento hacia delante o encadenamiento conducido por hechos (forward chaining).
- Encadenamiento hacia atrás o encadenamiento conducido por objetivos (backward chaining).

8.1.1 LOS OBJETOS EN KAPPA

Los objetos permiten representar varias propiedades como la herencia, encapsulamiento de datos y código, jerarquías, herencia. Todas estas propiedades son compartidas por el sistema de marcos usado para formalizar los conocimientos. En Kappa los objetos también permiten representar los procedimientos representados en “si necesito”, “si modifco”, “si borro” propios del sistema de marcos.

El concepto de clase e instancias en los marcos y en los objetos es el mismo. Los objetos se componen de atributos llamados *slots* y a los procedimientos se los llama *métodos*.

Los *slots* tiene propiedades como la cardinalidad, el tipo (texto, numérico, booleano). Pueden ser:

- Heredados son derivados del objeto antecesor en la jerarquía. Se refieren generalmente a propiedades genéricas o comunes que comparten los objetos.
- Propios son los que aparecen solamente en un objeto y se refieren a propiedades específicas del objeto, o sea atributos propios del objeto.

Los *métodos* al igual que los *slots* pueden ser heredados o locales. Se activan cuando el valor del *slot* del cual dependen es accedido para ser consultado o modificado. Hay 4 tipos :

- If needed: este método se usa para inicializar el slot
- When accessed: se dispara cuando se pide el valor del slot
- Before change: se dispara en el momento previo a una modificación del valor del slot.
- After change: se dispara en el momento posterior a una modificación del valor del slot.

8.1.2 LAS REGLAS EN KAPPA

Las reglas representan el conocimiento causal del tipo SI-ENTONCES. Cada regla queda identificada por un nombre. La parte SI de la regla está constituida por un conjunto de premisas y la parte ENTONCES por un conjunto de conclusiones. En Kappa los hechos involucrados en premisas y conclusiones se refieren a valores de atributos de objetos. El razonamiento del sistema se basa en reglas y está gestionado por el motor de inferencia.

En el encadenamiento hacia delante la filosofía general consiste en introducir una serie de hechos que se consideran ciertos en un momento determinado y dejar que las reglas infieran los nuevos valores. Kappa podrá buscar todas las posibles consecuencias derivadas de los nuevos hechos o sólo algunas, dependiendo que la estrategia sea exhaustiva o selectiva.

En el encadenamiento hacia atrás la filosofía general consiste en establecer una serie de objetivos y dejar que las reglas deduzcan su veracidad o no, en función del estado actual del sistema. Kappa busca todas las reglas que puedan probar el objetivo.

8.2 IMPLEMENTACION EN LA HERRAMIENTA

Los pasos seguidos en la implementación del presente trabajo han sido:

- Declaración del modelo de marcos relacionados con las dos primeras áreas de análisis del sistema experto: Requisitos y Tipo de aplicación, codificando sus propiedades y métodos.
- Incorporación de las reglas de las dos primeras áreas Requisitos y Tipo de Aplicación. El sistema se corresponde con la estructura de razonamiento Encadenamiento hacia atrás (Backward Chaining) . Los objetivos que se fijan por orden de prioridad son: 1) Ciclo de Vida en Espiral, 2) Ciclo de vida Orientado a Objetos y 3) Ciclo de Vida en Cascada.
- Desarrollo de una primera interfaz de usuario, donde fue necesario prestar especial atención a la forma de preguntar acerca de las características del proyecto a analizar. El diseño se realizó sobre la base de recuadros, en cada recuadro se definió una pregunta con opciones de respuesta, ya sean del tipo SI / NO o con opciones predefinidas. Para ello se usaron las facilidades que brinda el Kappa en la definición de este tipo de diseño. Los detalles del diseño de interfaz de usuario se pueden apreciar adecuadamente a partir del ítem 8.3 donde se muestra el desarrollo completo de un caso ejemplo.
- Se completaron los marcos y las reglas correspondientes a la tercer área de análisis Gestión de proyecto.
- Se refinó la interfaz de usuario en la presentación de los resultados, y se prestó especial atención a la forma de agrupar las preguntas para facilitar al usuario el análisis de las características del proyecto a analizar.
- Se consideró especialmente que el usuario al usar el sistema en una primera aproximación pudiera desconocer algunos datos. En ese caso el sistema si bien le brinda el resultado obtenido a partir de la información existente, también le informa que preguntas han quedado sin responder. De esta forma el usuario puede completar la información faltante en las primeras entrevistas, luego incorporarlas al sistema y obtener un diagnóstico más preciso.
- Se ha realizado una sesión de pruebas con el grupo de expertos que participaron, realizándose así el ultimo refinamiento de la interfaz de usuario. Los expertos solicitaron conocer el ciclo de vida recomendado para cada área, además del ciclo de vida propuesto, esto le permite al usuario, dentro del ciclo de vida elegido, organizar el proyecto adaptándolo con mayor precisión a las características de cada área.
- Se ha desarrollado una evaluación exhaustiva del sistema con el experto principal.

8.2.1 ESTRUCTURA GENERAL DE LOS OBJETOS

La estructura general de los objetos del sistema es la siguiente:

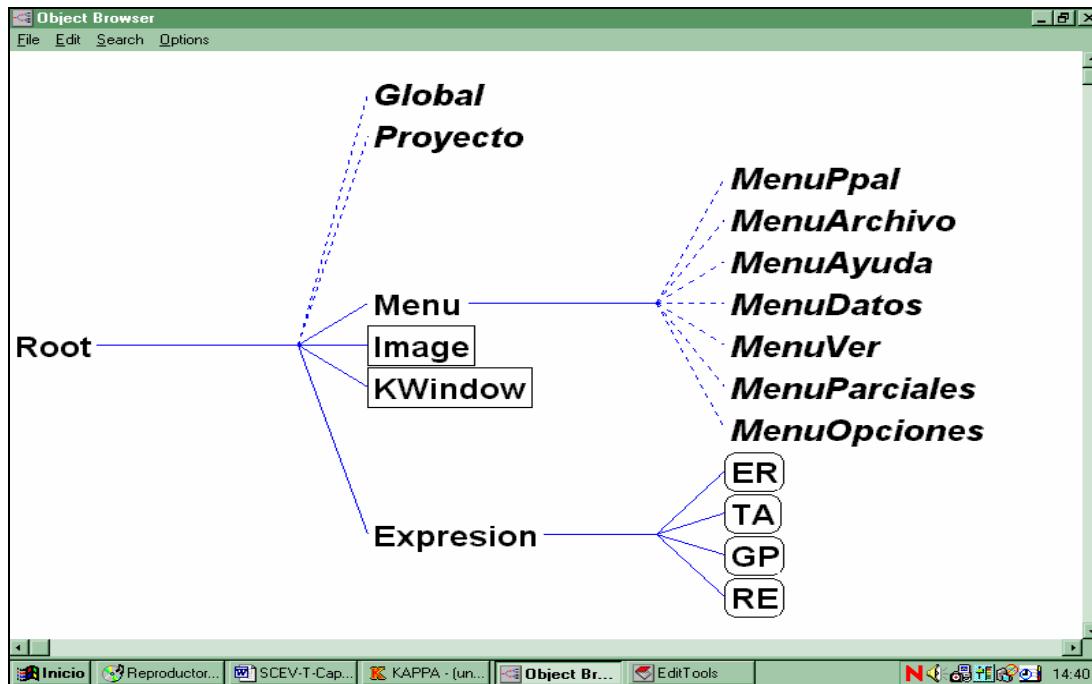


Figura 8-1: Estructura general de objetos del sistema

La estructura general de los objetos del área Especificación de Requerimientos es la siguiente:

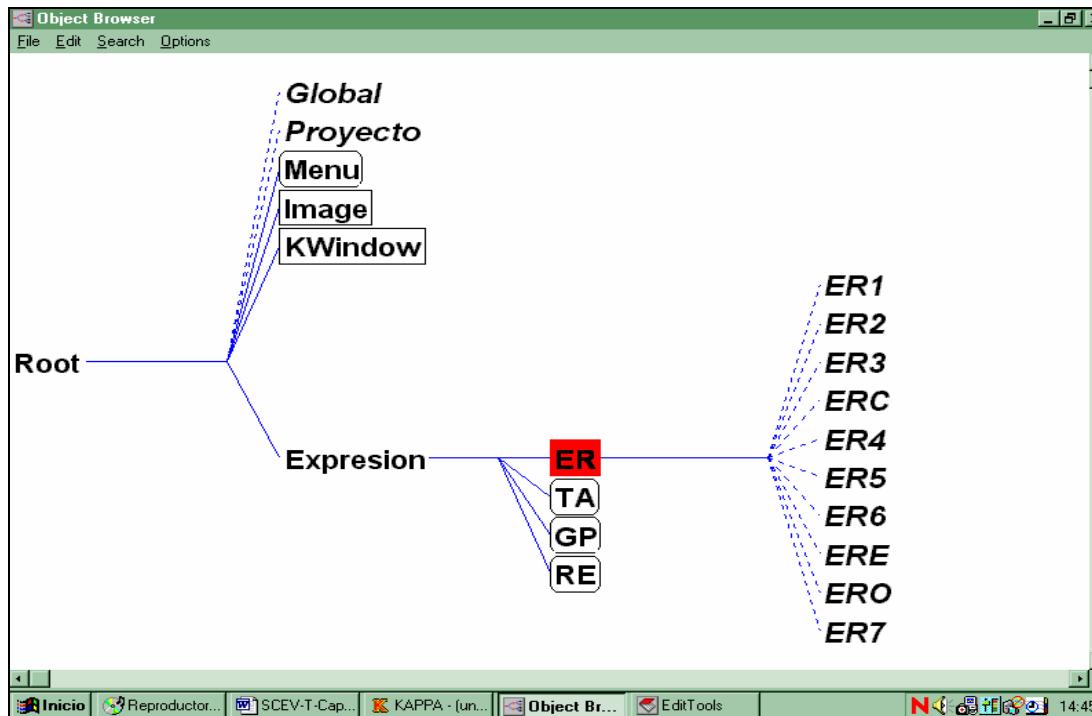


Figura 8-2: Estructura general de objetos del área Especificación Requerimientos

La estructura general de los objetos del área Tipo de Aplicación es la siguiente:

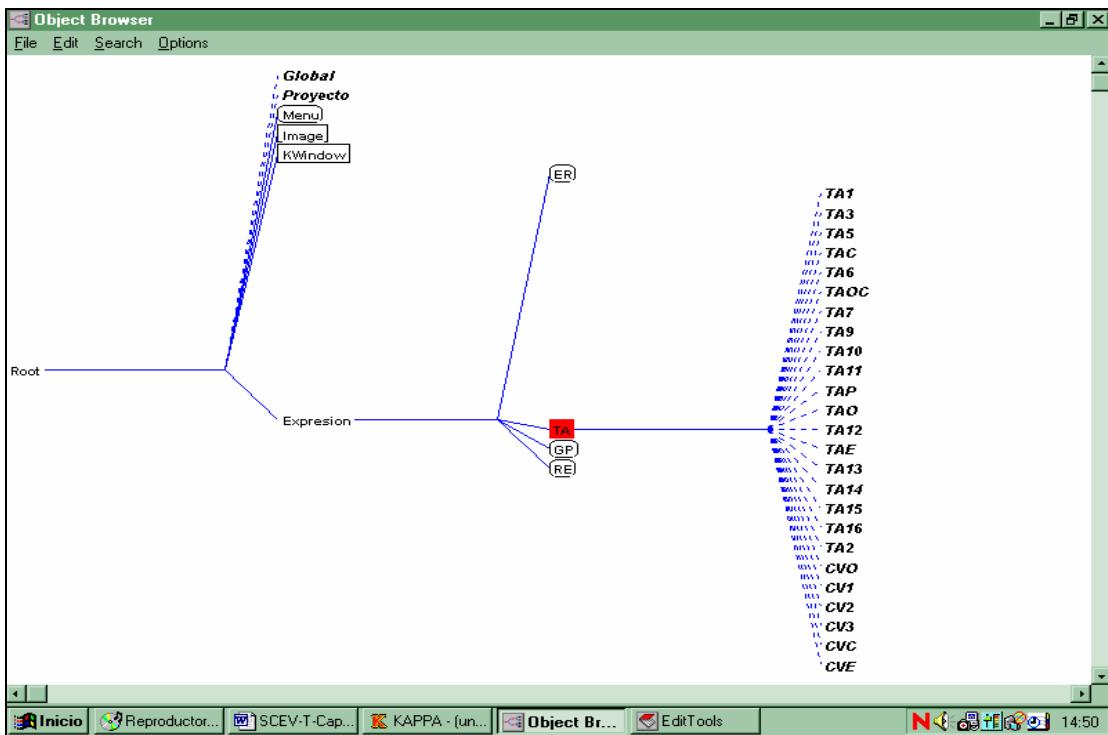


Figura 8-3: Estructura general de objetos del área Tipo de Aplicación

La estructura general de los objetos del área Gestión de Proyecto es la siguiente:

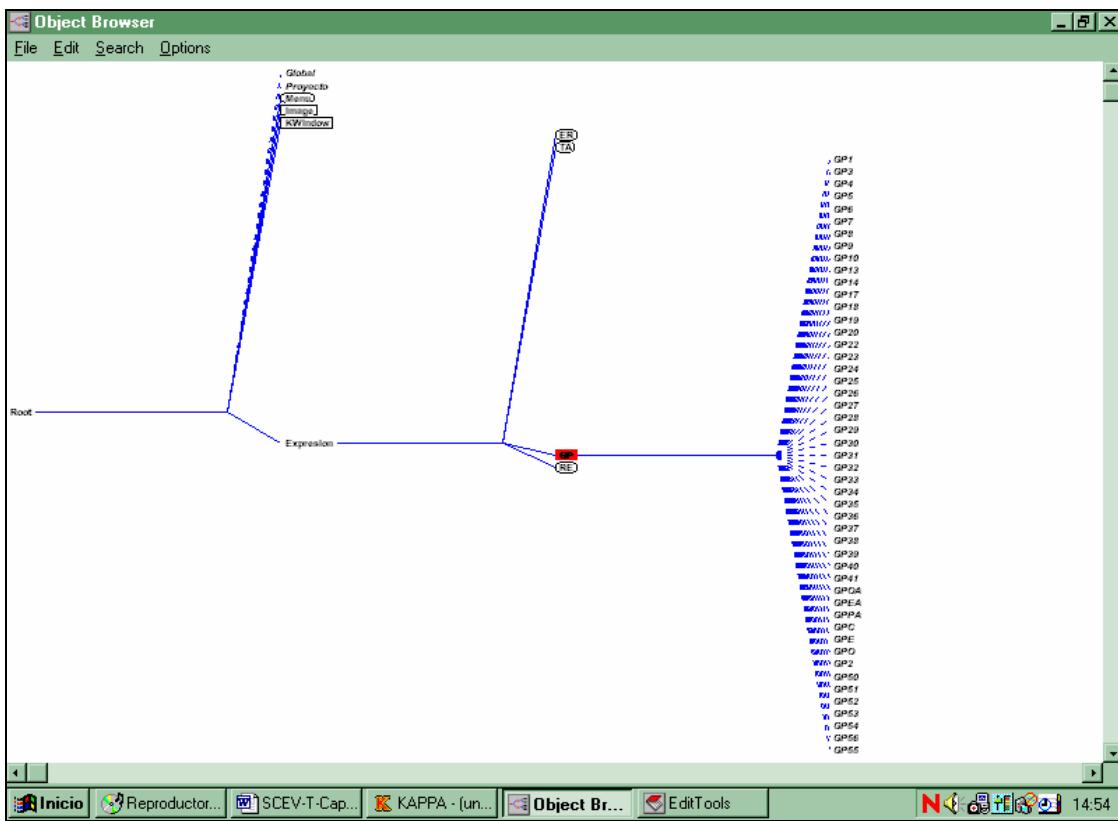


Figura 8-4: Estructura general de objetos del área Gestión de Proyecto

La estructura general de los objetos del área CV Diagnóstico:

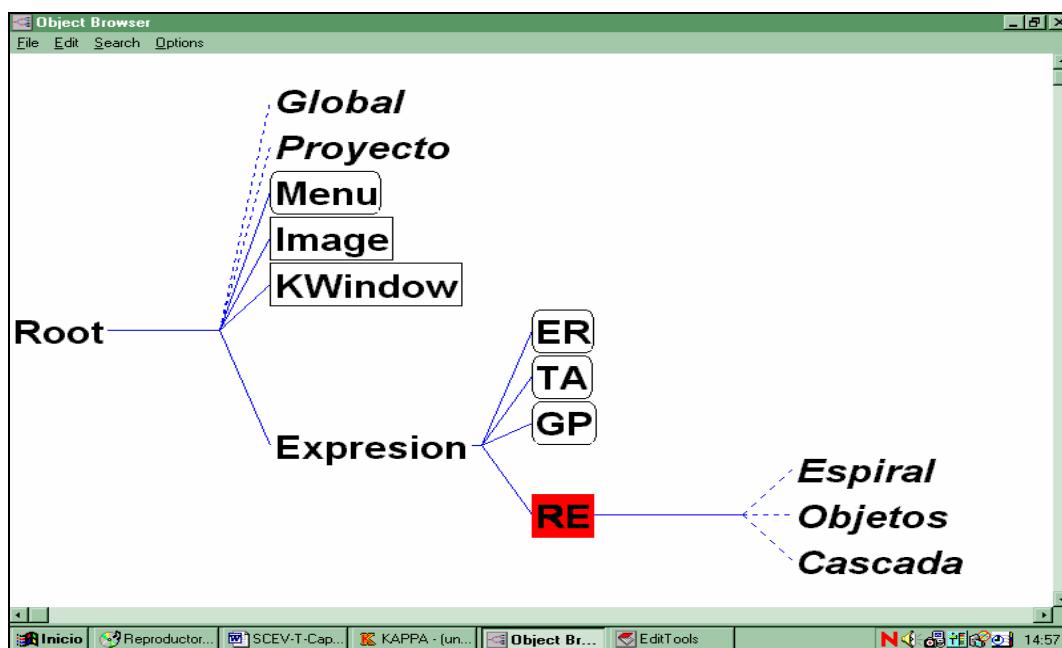


Figura 8-5: Estructura general de objetos del área CV Diagnóstico

8.3 DESARROLLO DE UN CASO EJEMPLO

A continuación se presenta un caso de ejemplo.

Cuando se ingresa al sistema se visualiza la pantalla de presentación que se observa en la figura 8-6.

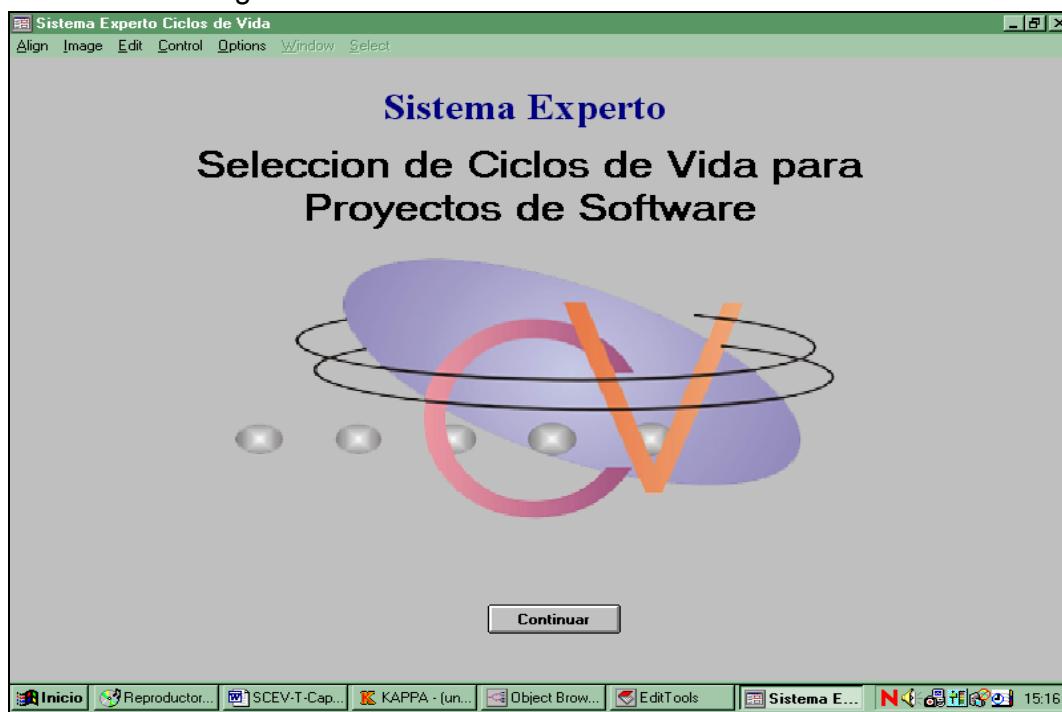


Figura 8-6: Ingreso al sistema

A continuación se visualiza la pantalla principal del sistema. Figura 8-7.

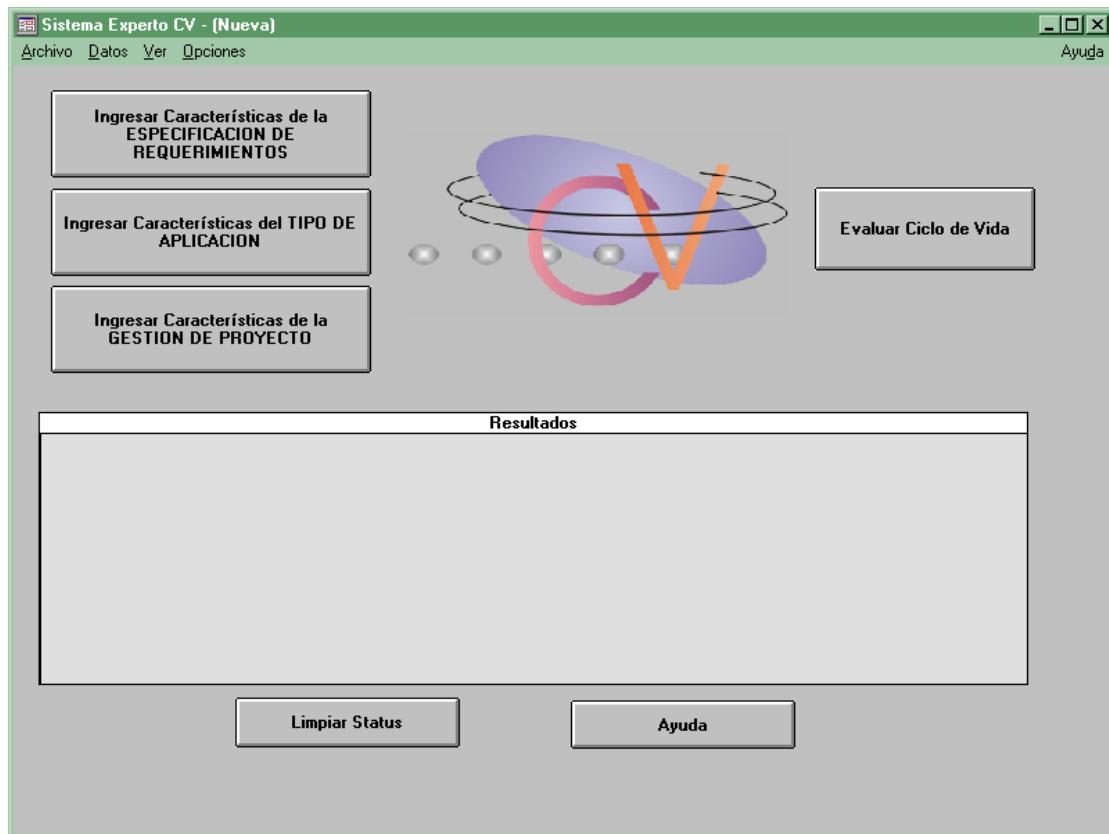


Figura 8-7: Pantalla Principal del Sistema

Usando el Menú desplegable de la barra en “Archivo” puede seleccionarse, (figura 8-8), algún proyecto ya ingresado (“Abrir Proyecto”), figura 8-9, para completar la información o comenzar con un proyecto nuevo (“Nuevo Proyecto. Si no hay indicación expresa el sistema asume que se trabajará con un proyecto nuevo.

Usando el menú desplegable de la barra en “Datos” puede ingresarse los datos que identifican al Proyecto , figura 8-10. Para los datos de las áreas puede ingresarse por cualquiera de las opciones indistintamente: Ingresar al área Especificación de Requerimientos, Ingresar al área Tipo de aplicación, o al área Gestión de Proyecto ya sea desde el menú desplegable en “Datos” o desde los botones de la pantalla Figura 8-11.

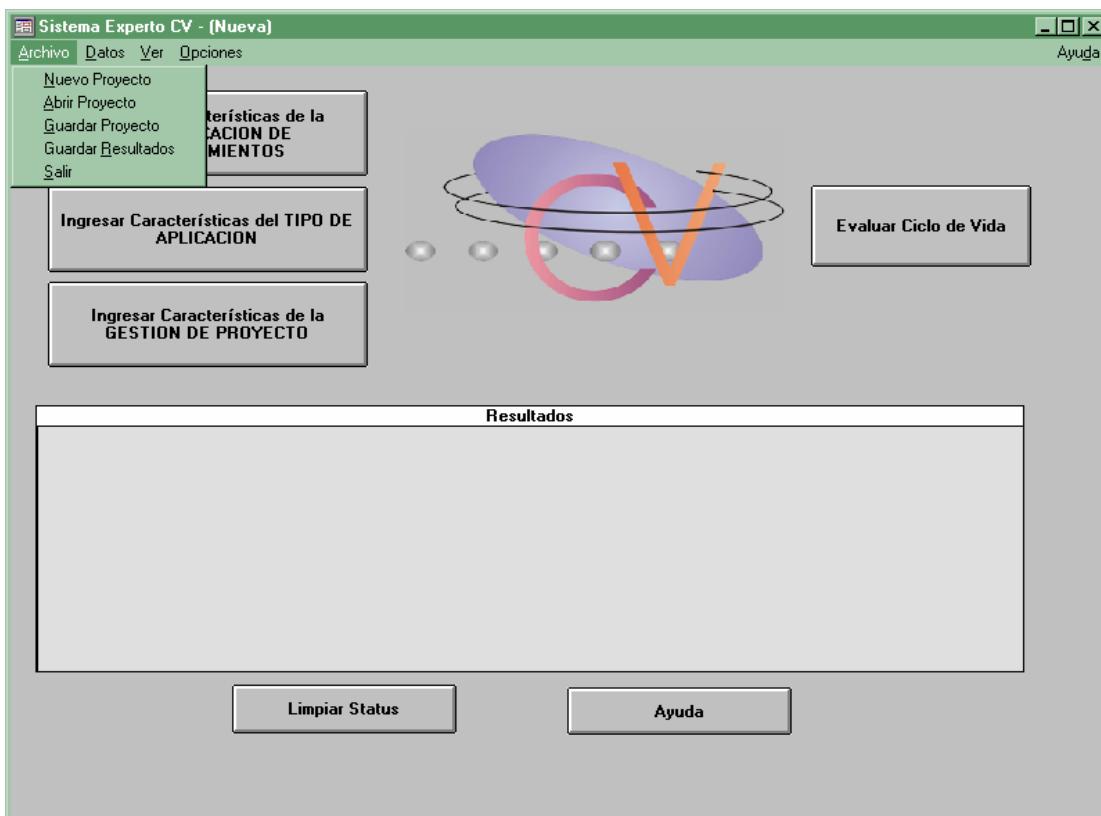


Figura 8-8: Selección de un proyecto

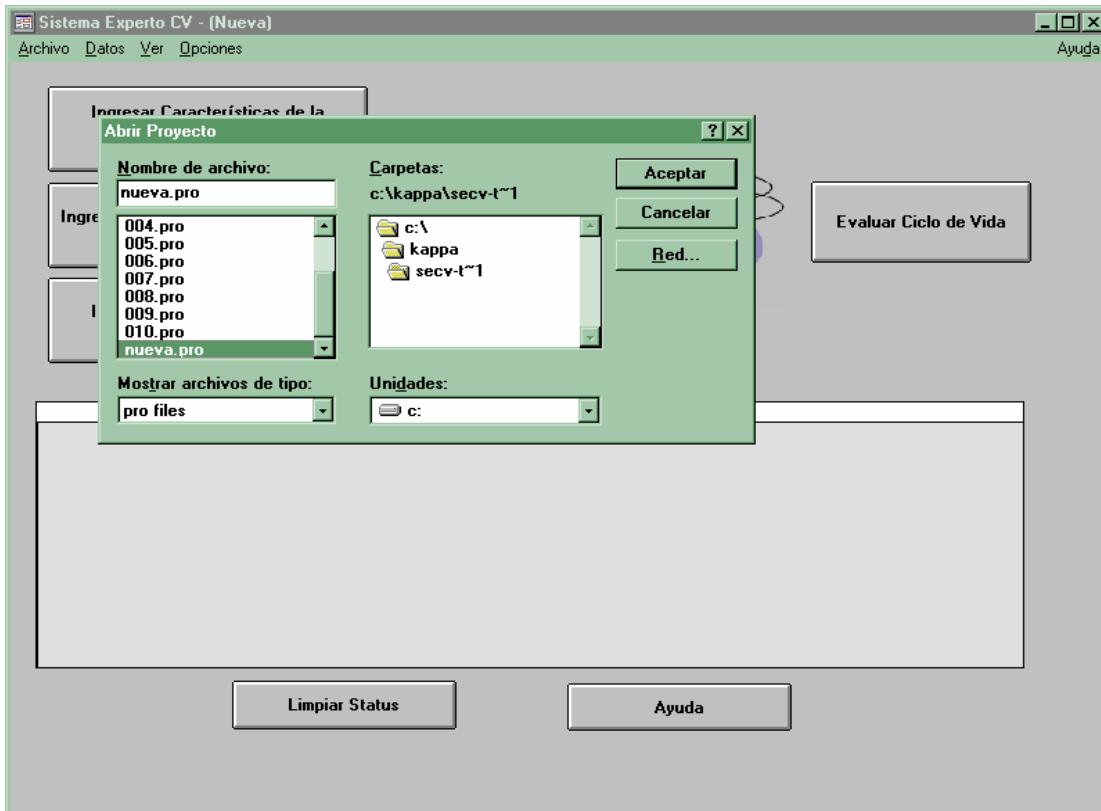


Figura 8-9: Abrir un Proyecto

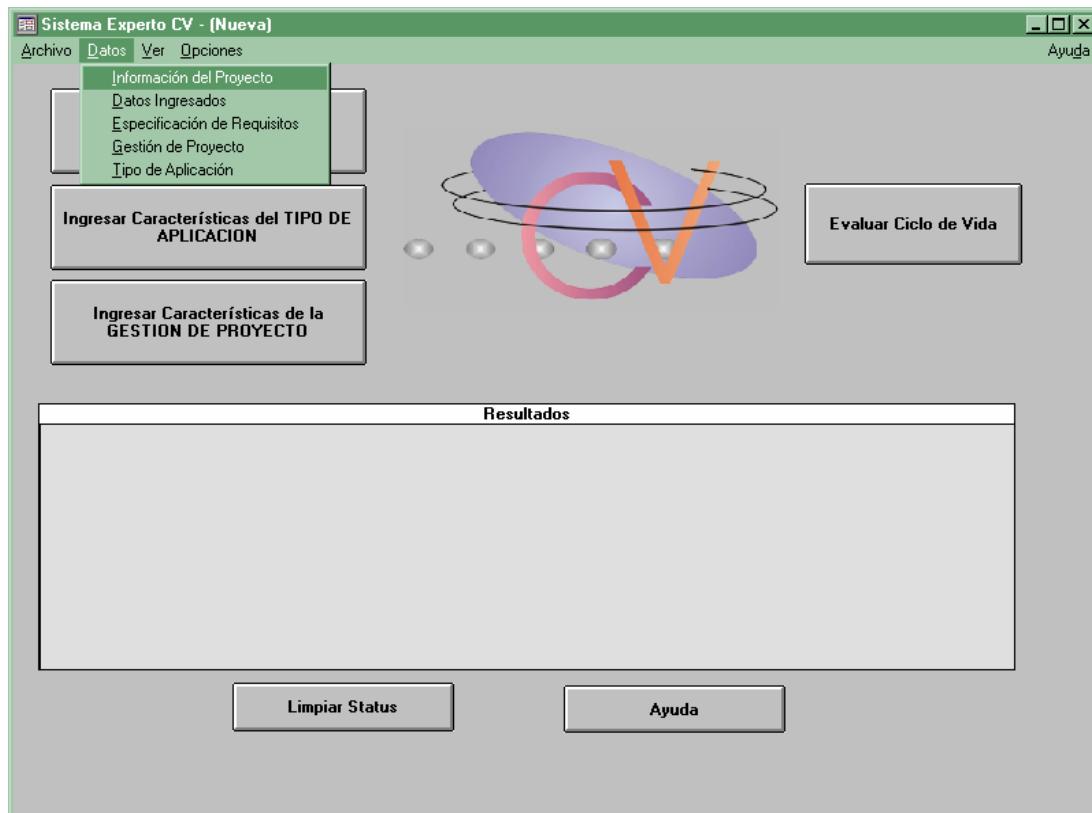


Figura 8-10: Identificación de un proyecto

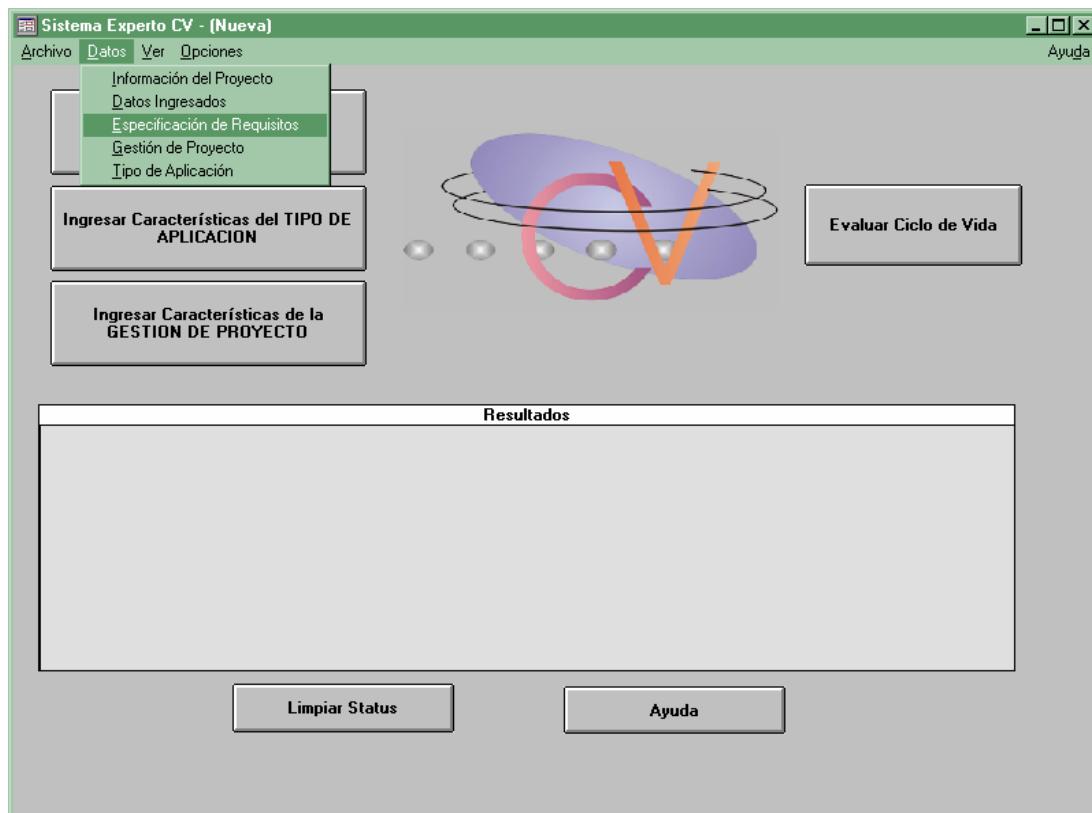


Figura 8-11: Áreas del Proyecto

Para el ejemplo se define un proyecto nuevo que se corresponde con el caso 2, del capítulo 9, y se seguirá el orden sugerido para un proyecto nuevo que es: Identificación del Proyecto (figura 8-12), Requerimientos, Tipo de Aplicación y Gestión de proyecto.

CASO EJEMPLO 2: Control de Stock

Proyecto: P002-Fabrica de ropa de hombre

Objetivo del proyecto: Desarrollar e implementar un sistema de control de stock para el depósito central y los negocios de venta al público.

Caso 2 Resultados del caso de prueba:	
Ciclo de vida esperado	Ciclo de vida sugerido por el sistema experto
CV Especificación Requerimientos Cascada	CV Especificación Requerimientos
CV Tipo de Aplicación Cascada	Tipo de Aplicación
CV Gestión de proyecto Objetos	Gestión de proyecto
CV del Proyecto Cascada	CV del Proyecto

Información del Proyecto

Align Image Edit Control Options Window Select

Información del Proyecto			
ID:	002	Nombre del Proyecto:	Control Stock
Líder del Proyecto:	Carlos Beltrami		
Objetivo	Desarrollar e implementar un sistema de stock para depósito central de una fabrica de ropa		
Fecha de Inicio:	19/02/1998	Fecha de Finalización:	30/06/1998
Ciclo de Vida Sugerido:		Ciclo de Vida seleccionado:	Cascada
Cerrar			

Inicio **Repr...** **Bibiana** **Tesis-bi** **KAPP...** **Objec...** **EditT...** **Siste...** **Info...** **SCEV...** **N...** **cert...** **pol...** 16:14

Figura 8-12: Ejemplo: Identificación del Proyecto

La figura 8-13 se corresponde con las características de Requerimientos. La interfaz de usuario esta diseñada para seleccionar las opciones que den respuesta a las siguientes preguntas (para el caso de ejemplo se han marcado en letra “negrita” la opción de respuesta seleccionada):

Especificación Requerimientos		
ID	Afirmación / Pregunta	Opciones
ER1	<i>Al comienzo del proyecto los requerimientos se encuentran definidos:</i>	Mayoritariamente
ER2	<i>El usuario ha explicitado los requisitos:</i>	Formalmente, Exhaustivamente, Uniformemente
ER3	<i>La exigencia del usuario en el cumplimiento de los requerimientos en el producto final es:</i>	Poco exigente

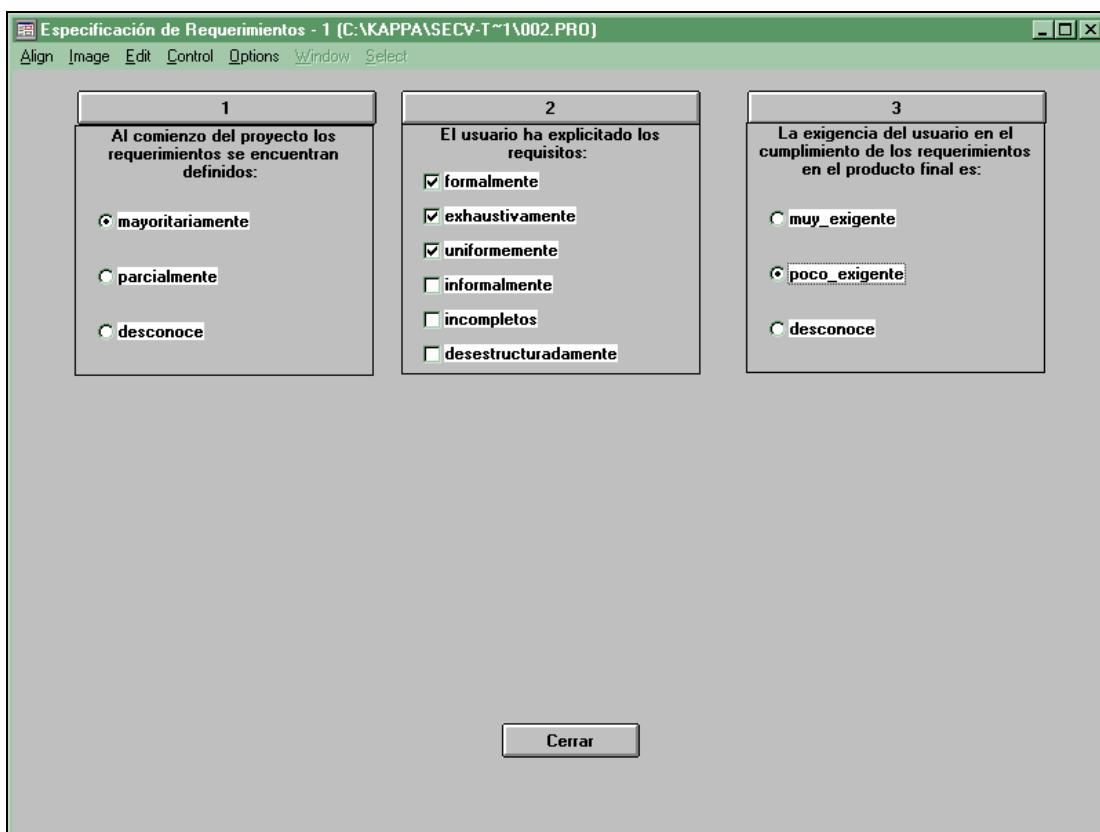


Figura 8-13: Ingresar cada una de las opciones de las Características de Especificación de Requerimientos

Cada uno de los cuadros de preguntas a completar tiene una opción de ayuda, es el recuadro superior donde figura el numero de cuadro, al presionar el botón (recuadro superior) el usuario obtiene información complementaria sobre las opciones a elegir. Se muestra a modo de ejemplo la información sobre el recuadro 2, figura 8-14.

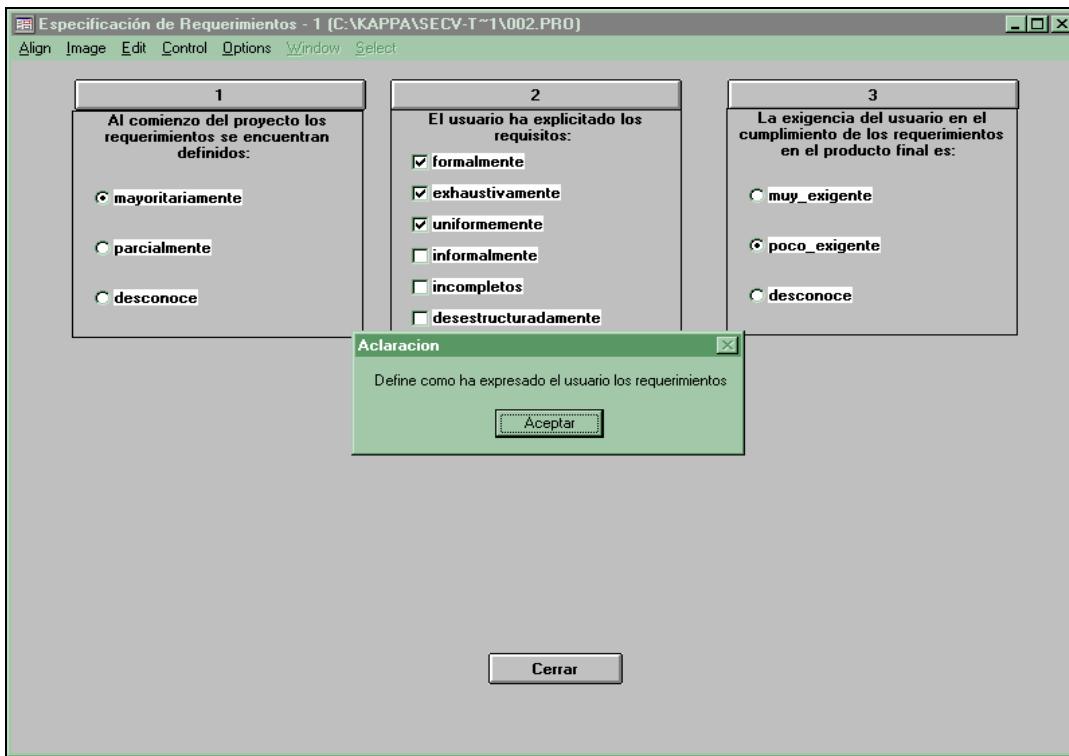


Figura 8-14: Especificación de Requerimientos - Consulta de ayuda

La figura 8-15 a 8-17 se corresponde con las características de Tipo de Aplicación. La interfaz de usuario esta diseñada para seleccionar las opciones que den respuesta a las siguientes preguntas (para el caso de ejemplo se han marcado en letra “negrita” la opción de respuesta seleccionada):

Tipo de Aplicación		
ID	Afirmación / Pregunta	Opciones
TA1	<i>El tipo de aplicación se orienta al desarrollo de :</i>	Otros
TA2	<i>Los componentes predominantes de la aplicación son:</i>	Algorítmico
TA3	<i>El comportamiento predominante en la aplicación es:</i>	Batch
TA4	<i>¿La progresión entre las fases del desarrollo es secuencial y uniforme?</i>	Si
TA5	<i>La relación entre las fases del proyecto es:</i>	Poca dependencia
TA6	<i>¿El tipo de aplicación requiere el desarrollo fuertemente integrado de hardware y software?</i>	No
TA7	<i>¿El sistema a diseñar es innovador?</i>	No
TA8	<i>Para lograr una mejor comprensión del sistema los niveles de descomposición en subsistemas son:</i>	Pocos
TA9	<i>Para lograr una mejor manipulación del sistema los niveles de descomposición en subsistemas son:</i>	Pocos
TA10	<i>Se requiere detectar en etapas tempranas del proyecto problemas de:</i>	Desconoce
TA11	<i>El Diseño del sistema a desarrollar puede resolverse por:</i>	Pocas opciones
TA12	<i>La Implementación del sistema a desarrollar puede resolverse por:</i>	Pocas opciones
TA13	<i>Las modificaciones explícitamente previstas son:</i>	Extensión

Tipo de Aplicación - 1 (C:\KAPPA\SECV-T~1\002.PRO)

Align Image Edit Control Options Window Select

1 El tipo de aplicación se orienta al desarrollo de: <input type="radio"/> software_de_base <input type="radio"/> SSBBCC <input checked="" type="radio"/> Otros	2 Los componentes predominantes de la aplicación son: <input type="checkbox"/> gráfico <input type="checkbox"/> matemático <input checked="" type="checkbox"/> algorítmico <input type="checkbox"/> otro	3 El comportamiento predominante de la aplicación es: <input checked="" type="checkbox"/> batch <input type="checkbox"/> interactivo <input type="checkbox"/> tiempo_real <input type="checkbox"/> dinámico <input type="checkbox"/> ninguno
4 ¿La progresión entre las fases del desarrollo es secuencial y uniforme?: <input checked="" type="radio"/> Si <input type="radio"/> No <input type="radio"/> Desconoce	5 La relación entre las fases de desarrollo del proyecto es: <input type="radio"/> mucha_dependencia <input checked="" type="radio"/> poca_dependencia <input type="radio"/> independencia	6 ¿El tipo de aplicación requiere el desarrollo fuertemente integrado de hardware y software?: <input type="radio"/> Si <input checked="" type="radio"/> No <input type="radio"/> Desconoce
Siguiente		

Figura 8-15: Características de Tipo de Aplicación- Pantalla 1

Tipo de Aplicación - 2 (C:\KAPPA\SECV-T~1\002.PRO)

Align Image Edit Control Options Window Select

7 ¿El sistema a diseñar es innovador?: <input type="radio"/> Si <input checked="" type="radio"/> No <input type="radio"/> Desconoce	8 Para lograr una mejor comprensión del sistema, los niveles de descomposición en subsistemas son: <input checked="" type="radio"/> pocos <input type="radio"/> muchos <input type="radio"/> desconoce	9 Para lograr una mejor manipulación del sistema, los niveles de descomposición en subsistemas son: <input checked="" type="radio"/> pocos <input type="radio"/> muchos <input type="radio"/> desconoce
10 Se requiere detectar en etapas tempranas del proyecto problemas de: <input type="checkbox"/> arquitectura <input type="checkbox"/> eficacia <input checked="" type="checkbox"/> desconoce	11 El diseño del Sistema a desarrollar puede resolverse por: <input checked="" type="radio"/> pocas_opciones <input type="radio"/> varias_opciones	12 La implementación del Sistema a desarrollar puede resolverse por: <input checked="" type="radio"/> pocas_opciones <input type="radio"/> varias_opciones
Anterior Siguiente		

Figura 8-16: Características de Tipo de Aplicación- Pantalla 2

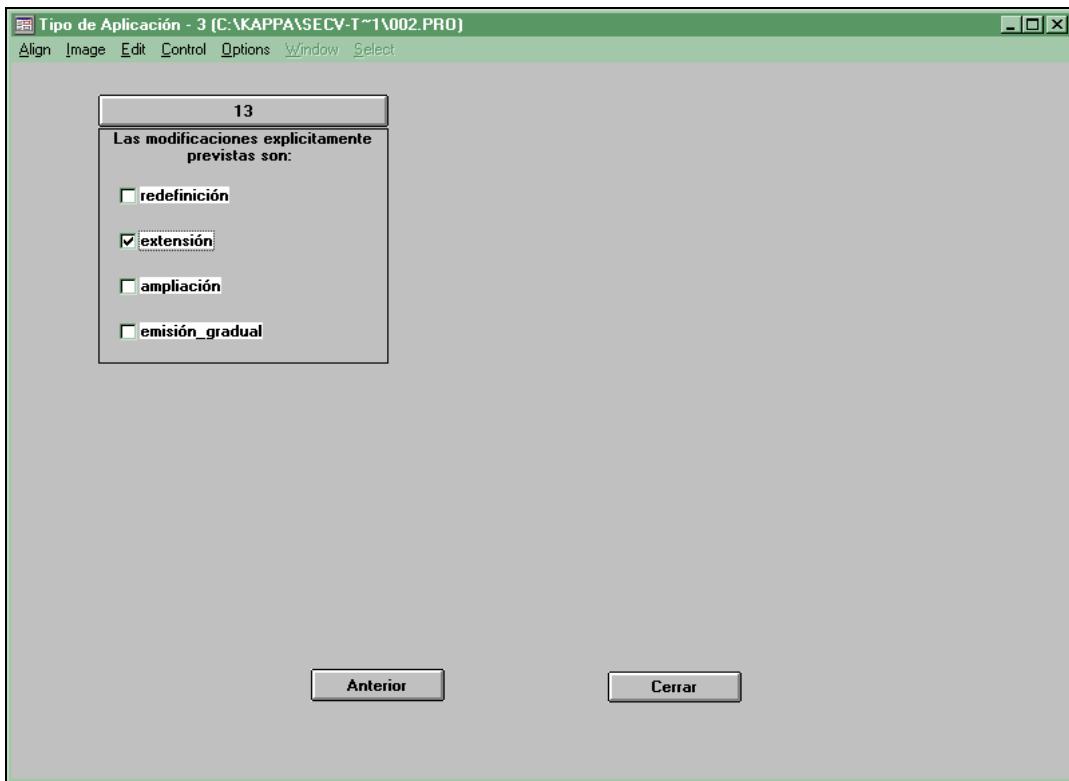


Figura 8-17: Características de Tipo de Aplicación- Pantalla 3

Las figuras 8-18 a 8-22 se corresponden con las características de la Gestión de Proyecto. La interfaz de usuario esta diseñada para seleccionar las opciones que den respuesta a las siguientes preguntas (para el caso de ejemplo se han marcado en letra “negrita” la opción de respuesta seleccionada):

Gestión de Proyecto		
ID	Afirmación / Pregunta	Opciones
GP1	<i>El desarrollo del proyecto es responsabilidad de:</i>	La organización
GP2	<i>¿El equipo de desarrollo tiene experiencia previa en el tipo de aplicación a desarrollar?</i>	Si
GP3	<i>Es factible usar metodologías de desarrollo:</i>	Medianamente probadas
GP4	<i>Es necesario usar metodologías de desarrollo:</i>	Medianamente probadas
GP5	<i>¿El equipo de desarrollo tiene experiencia con las técnicas de ingeniería del software que su usarán?</i>	Si
GP6	<i>El tipo de entrega que el usuario requiere de la aplicación es:</i>	Versión parcial
GP7	<i>El grado de participación del usuario en el desarrollo debe ser:</i>	Medio
GP8	<i>¿Es necesario facilitar la introducción gradual del sistema para los usuarios?</i>	Si
GP9	<i>Se requiere una gestión de proyecto:</i>	Poco formal
GP10	<i>El grado de control que se requiere de la gestión del proyecto es:</i>	Poco ajustado
GP11	<i>¿La gestión de proyecto prevé definir explícitamente estrategias para reusar software existente?</i>	No

GP12	¿Se prevé la necesidad de reutilizar los componentes del software a desarrollar en proyectos futuros?	Muy necesario
GP13	¿Existe un sistema previo desarrollado en objetos de forma tal que el proyecto actual es ..?	Sin relación
GP14	¿Existe la necesidad de reutilizar los componentes de un software ya existente desarrollado en objetos?	No
GP15	¿Se dispone de software para prototipar?	No
GP16	¿Es posible adquirir software para prototipar?	Desconoce
GP17	¿Existen dudas sobre la viabilidad del software?	No
GP18	¿Se estima la posibilidad de terminación prematura de proyectos no viables?	No
GP19	Se estima que el factor de riesgo en el desarrollo del sistema es:	Bajo
GP20	¿Se estiman riesgos técnicos en el desarrollo del sistema?	No
GP21	¿Se identifican los riesgos asociados con cada una de las alternativas?	Desconoce
GP22	¿Se cuenta con métodos y técnicas para evaluar el riesgo?	No
GP23	¿Se identifican las diferentes maneras de resolver los riesgos?	No
GP24	¿Es necesario definir explícitamente estrategias para realizar análisis de riesgo?	Desconoce
GP25	¿Es necesario volver a etapas anteriores en el desarrollo del sistema para la resolución de algún tópico nuevo de riesgo?	Desconoce
GP26	¿Es posible volver a etapas anteriores en el desarrollo del sistema cuando se identifican mejores alternativas?	Desconoce
GP27	¿Es necesario definir mecanismos explícitos para incorporar objetivos de calidad en el desarrollo del producto?	Si
GP28	El uso de los mismos procedimientos para desarrollo y mantenimiento del producto software está previsto como:	Indiferente
GP29	¿Se requiere de acuerdos confirmados para los cambios a efectuarse durante el desarrollo del proyecto?	Si

Gestión de Proyecto - 1 [C:\KAPPASECV-T~1\002.PRO]

Align Image Edit Control Options Window Select

1 El desarrollo del proyecto es responsabilidad de ...: <input checked="" type="radio"/> terceros <input checked="" type="radio"/> la_organización	2 ¿El equipo de desarrollo tiene experiencia previa en el tipo de aplicación a desarrollar?: <input checked="" type="radio"/> Si <input type="radio"/> No <input type="radio"/> Desconoce	3 Es factible usar metodologías de desarrollo ...: <input checked="" type="radio"/> ampliamente_probadas <input checked="" type="radio"/> medianamente_probadas <input type="radio"/> poco_probadas <input type="radio"/> Desconoce
4 Es necesario usar metodologías de desarrollo ...: <input checked="" type="radio"/> ampliamente_probadas <input checked="" type="radio"/> medianamente_probadas <input type="radio"/> poco_probadas <input type="radio"/> desconoce	5 ¿El equipo de desarrollo tiene experiencia con las técnicas de ingeniería del software que se usarán?: <input checked="" type="radio"/> Si <input type="radio"/> No <input type="radio"/> Desconoce	6 El tipo de entrega que el usuario requiere de la aplicación es ...: <input checked="" type="radio"/> versión_temprana <input checked="" type="radio"/> versión_gradual <input checked="" type="radio"/> versión_parcial <input type="radio"/> versión_completa <input type="radio"/> desconoce
Siguiente		

Figura 8-18: Características de Gestión de Proyecto - Pantalla 1

Gestión de Proyecto - 2 [C:\KAPPA\SECV-T^1\002.PRO]

Align Image Edit Control Options Window Select

<p>7 El grado de participación del usuario en el desarrollo debe ser ...:</p> <p><input type="radio"/> alto <input checked="" type="radio"/> medio <input type="radio"/> bajo</p>	<p>8 Es necesario facilitar la introducción gradual del sistema para los usuarios:</p> <p><input type="radio"/> Si <input type="radio"/> No <input type="radio"/> Desconoce</p>	<p>9 Se requiere una gestión de proyecto...:</p> <p><input type="radio"/> muy_formal <input type="radio"/> medianamente_formal <input checked="" type="radio"/> poco_formal <input type="radio"/> no_formal <input type="radio"/> desconoce</p>
<p>10 El grado de control que se requiere de la gestión del proyecto es:</p> <p><input type="radio"/> muy_ajustado <input type="radio"/> medianamente_ajustado <input checked="" type="radio"/> poco_ajustado</p>	<p>11 ¿La gestión del proyecto prevee definir explícitamente estrategias para reusar software ya existente?:</p> <p><input type="radio"/> Si <input checked="" type="radio"/> No <input type="radio"/> Desconoce</p>	<p>12 ¿Se prevee la necesidad de reutilizar los componentes del software a desarrollar en proyectos futuros?:</p> <p><input checked="" type="radio"/> muy_necesario <input type="radio"/> medianamente_necesario <input type="radio"/> poco_necesario <input type="radio"/> Desconoce</p>

Anterior **Siguiente**

Figura 8-19: Características de Gestión de Proyecto - Pantalla 2

Gestión de Proyecto - 3 [C:\KAPPA\SECV-T^1\002.PRO]

Align Image Edit Control Options Window Select

<p>13 ¿Existe un sistema previo desarrollado en objetos de forma tal que el proyecto actual es...?:</p> <p><input type="radio"/> un_subsistema <input type="radio"/> una_ampliación <input type="radio"/> una_modificación <input checked="" type="radio"/> sin_relación</p>	<p>14 Existe la necesidad de reutilizar los componentes de un software ya existente desarrollado en Objetos?:</p> <p><input type="radio"/> Si <input checked="" type="radio"/> No <input type="radio"/> Desconoce</p>	<p>15 ¿Se dispone de software para prototipar?:</p> <p><input type="radio"/> Si <input checked="" type="radio"/> No <input type="radio"/> Desconoce</p>
<p>16 ¿Es posible adquirir software para prototipar?:</p> <p><input checked="" type="radio"/> Si <input type="radio"/> No <input type="radio"/> Desconoce</p>	<p>17 ¿Existen dudas sobre la viabilidad del software?:</p> <p><input type="radio"/> Si <input checked="" type="radio"/> No <input type="radio"/> Desconoce</p>	<p>18 Se estima la posibilidad de la terminación prematura de proyectos no viables:</p> <p><input type="radio"/> Si <input checked="" type="radio"/> No <input type="radio"/> Desconoce</p>

Anterior **Siguiente**

Figura 8-20: Características de Gestión de Proyecto - Pantalla 3

Gestión de Proyecto - 4 (C:\KAPPA\SECV-T~1\002.PRO)

Align Image Edit Control Options Window Select

<p>19 Se estima que el factor de riesgo en el desarrollo del sistema es:</p> <p><input type="radio"/> alto <input checked="" type="radio"/> medio <input type="radio"/> bajo <input type="radio"/> sin_riesgo</p>	<p>20 ¿Se estiman riesgos técnicos en el desarrollo del sistema?:</p> <p><input type="radio"/> Si <input checked="" type="radio"/> No <input type="radio"/> Desconoce</p>	<p>21 ¿Se identifican los riesgos asociados con cada una de las alternativas?:</p> <p><input type="radio"/> Si <input type="radio"/> No <input checked="" type="radio"/> Desconoce</p>
<p>22 ¿Se cuenta con métodos y técnicas para evaluar el riesgo?:</p> <p><input type="radio"/> Si <input checked="" type="radio"/> No <input type="radio"/> Desconoce</p>	<p>23 ¿Se identifican las diferentes maneras de resolver los riesgos?:</p> <p><input type="radio"/> Si <input checked="" type="radio"/> No <input type="radio"/> Desconoce</p>	<p>24 ¿Es necesario definir explícitamente estrategias para realizar análisis de riesgo?:</p> <p><input type="radio"/> Si <input type="radio"/> No <input checked="" type="radio"/> Desconoce</p>

Anterior **Siguiente**

Figura 8-21: Características de Gestión de Proyecto - Pantalla 4

Gestión de Proyecto - 5 (C:\KAPPA\SECV-T~1\002.PRO)

Align Image Edit Control Options Window Select

<p>25 ¿Es necesario volver a etapas anteriores en el desarrollo del sistema para la resolución de algún tópico nuevo de riesgo?:</p> <p><input type="radio"/> Si <input checked="" type="radio"/> No <input type="radio"/> Desconoce</p>	<p>26 ¿Es posible volver a etapas anteriores en el desarrollo del sistema cuando se identifican mejores alternativas?:</p> <p><input type="radio"/> Si <input type="radio"/> No <input checked="" type="radio"/> Desconoce</p>	<p>27 ¿Es necesario definir mecanismos explícitos para incorporar objetivos de calidad en el desarrollo del producto?:</p> <p><input checked="" type="radio"/> Si <input type="radio"/> No <input type="radio"/> Desconoce</p>
<p>28 ¿El uso de los mismos procedimientos para desarrollo y mantenimiento del producto de software está previsto como?:</p> <p><input type="checkbox"/> necesario <input type="checkbox"/> factible <input checked="" type="checkbox"/> indiferente</p>	<p>29 ¿Se requiere de acuerdos confirmados para los cambios a efectuarse durante el desarrollo del proyecto?:</p> <p><input checked="" type="radio"/> Si <input type="radio"/> No <input type="radio"/> Desconoce</p>	

Anterior **Cerrar**

Figura 8-22: Características de Gestión de Proyecto - Pantalla 5

Cada vez que se completan las preguntas de una de las áreas, antes de pasar a otra el sistema verifica que se hayan completado todas las preguntas, caso contrario le informa al usuario cuales preguntas han quedado sin responder para que decida si desea continuar o si desea completar esa información. En la figura 8-23 se presenta un ejemplo del área gestión de Proyectos.

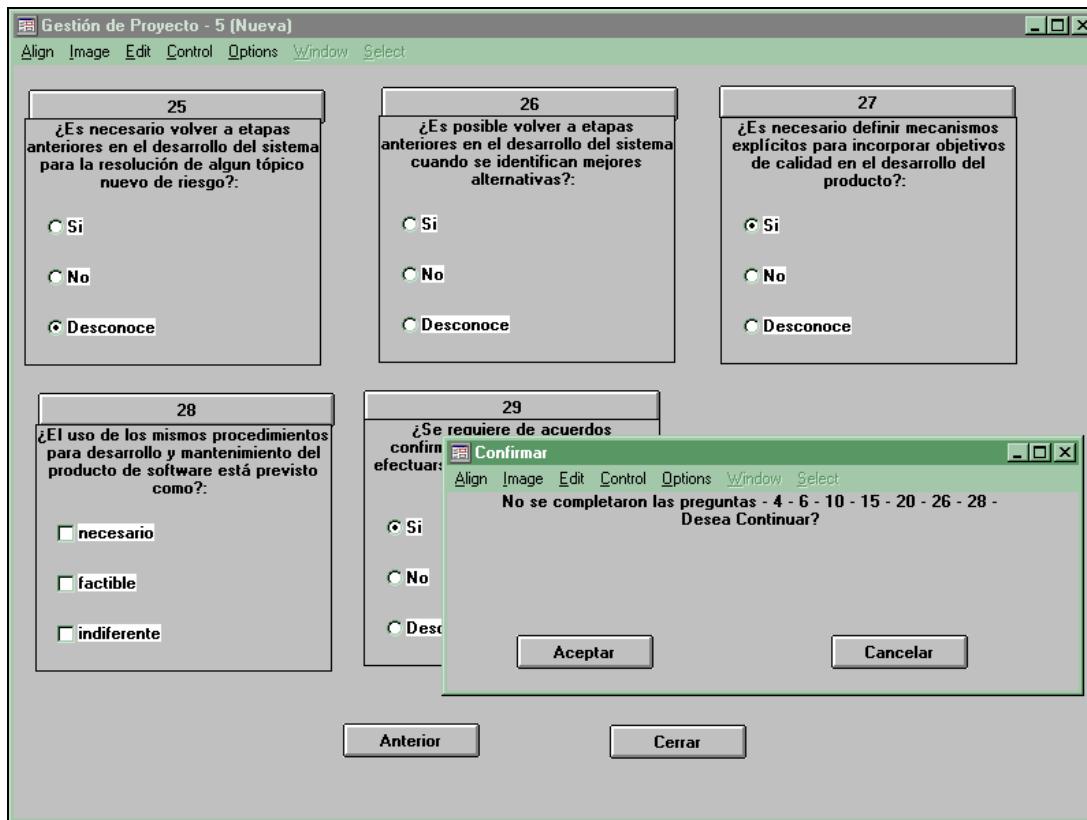


Figura 8-23: Verificación de preguntas respondidas del área Gestión de Proyectos

Una vez completadas cada una de las áreas el botón “Evaluar Ciclo de Vida” realiza el proceso de razonamiento y el sistema presenta los resultados finales, indicando el resultado propuesta para cada una de las áreas y el resultado final propuesto como el ciclo de vida mas adecuado para llevar adelante el desarrollo del proyecto, figura 8-24.

El sistema también detalla cada una de las reglas que se han aplicado en el proceso de razonamiento para ese caso en particular, figura 8-25.

El sistema permite guardar la información del proyecto, las reglas usadas en el razonamiento y los resultados obtenidos en un archivo “*.txt” para poder imprimir todos los datos del proyecto, figura 8-26.

El sistema actualiza automáticamente los datos de Información del Proyecto con el CV propuesto por el sistema experto, figura 8-27.

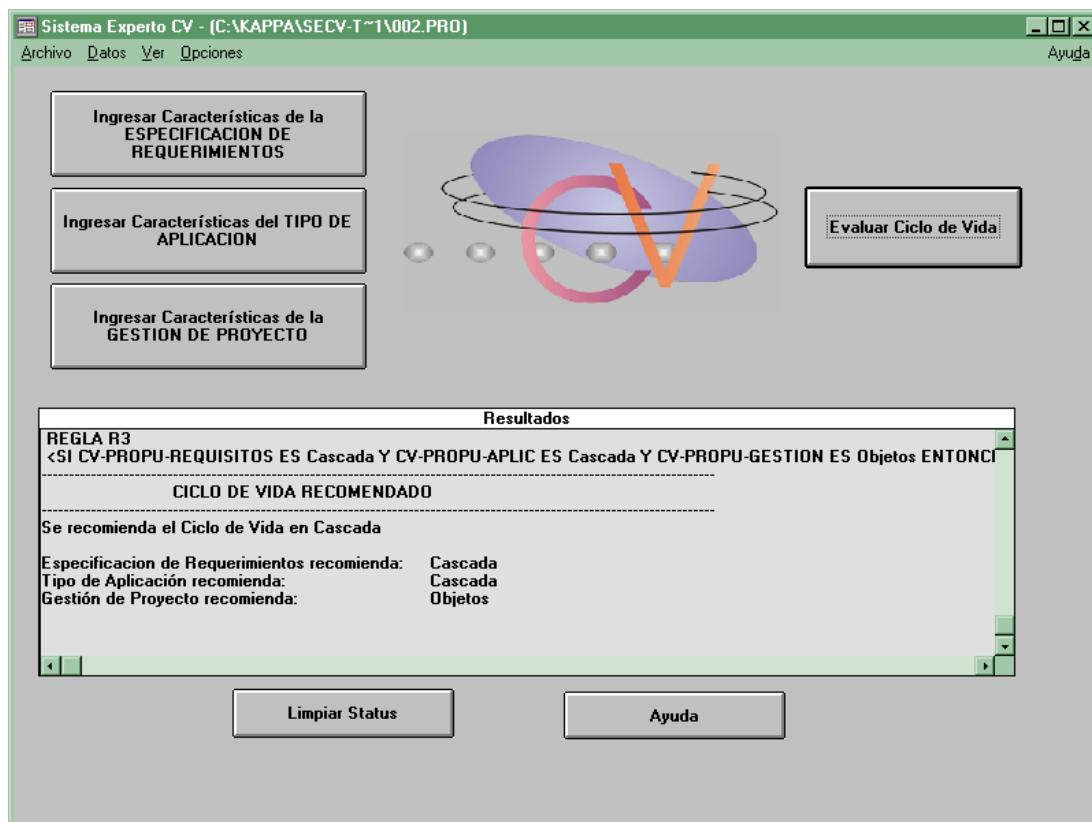


Figura 8-24: Recomendación por área y Recomendación del CV del Proyecto

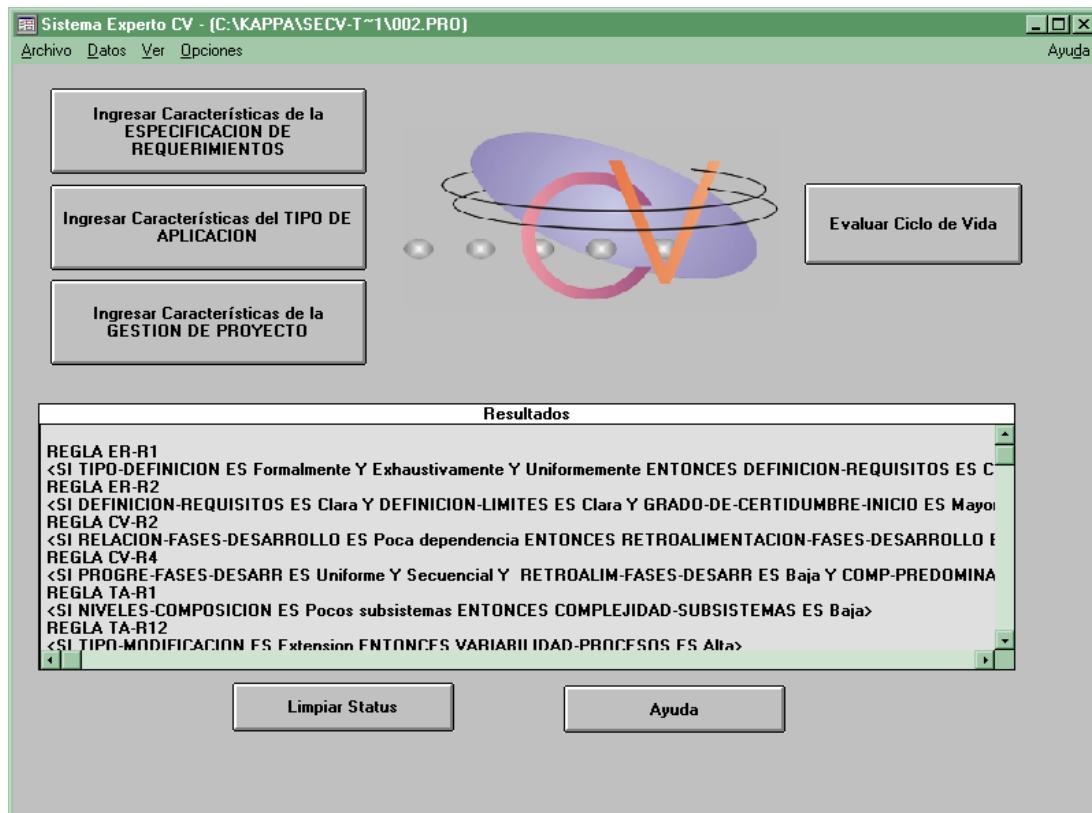


Figura 8-25: Detalle de Reglas aplicadas en el proceso de razonamiento

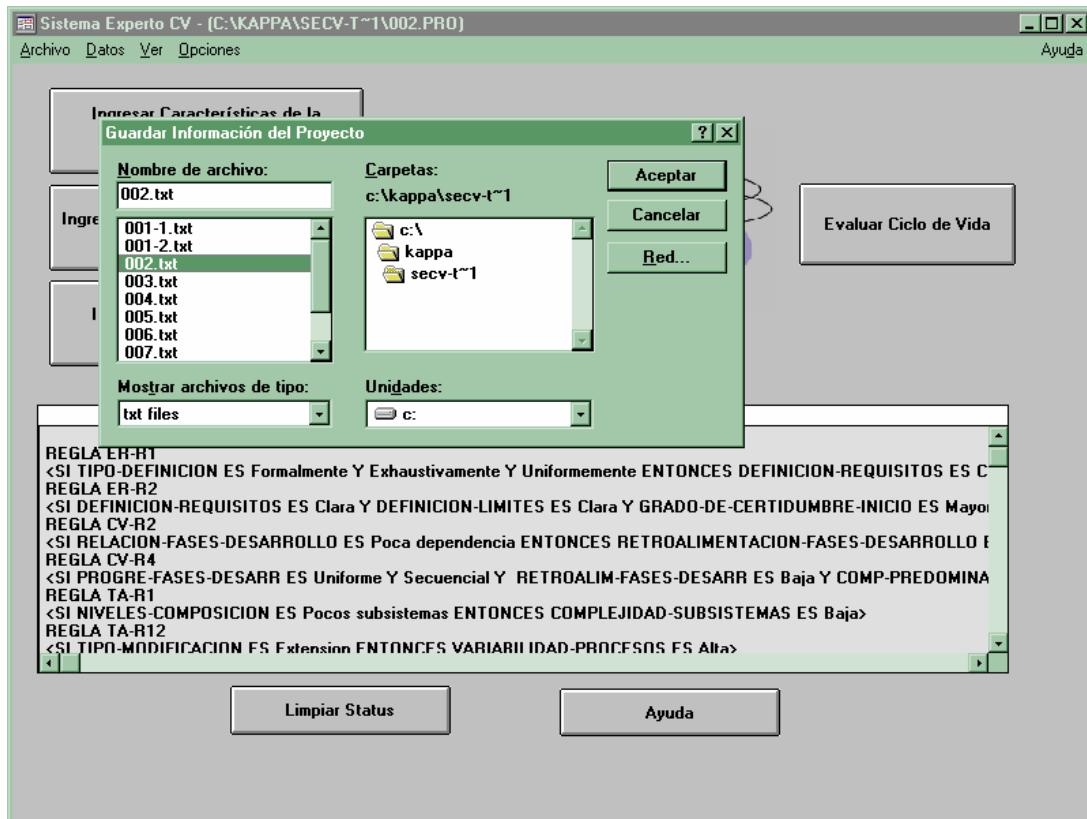


Figura 8-26: Guardar información y resultados del Proyecto

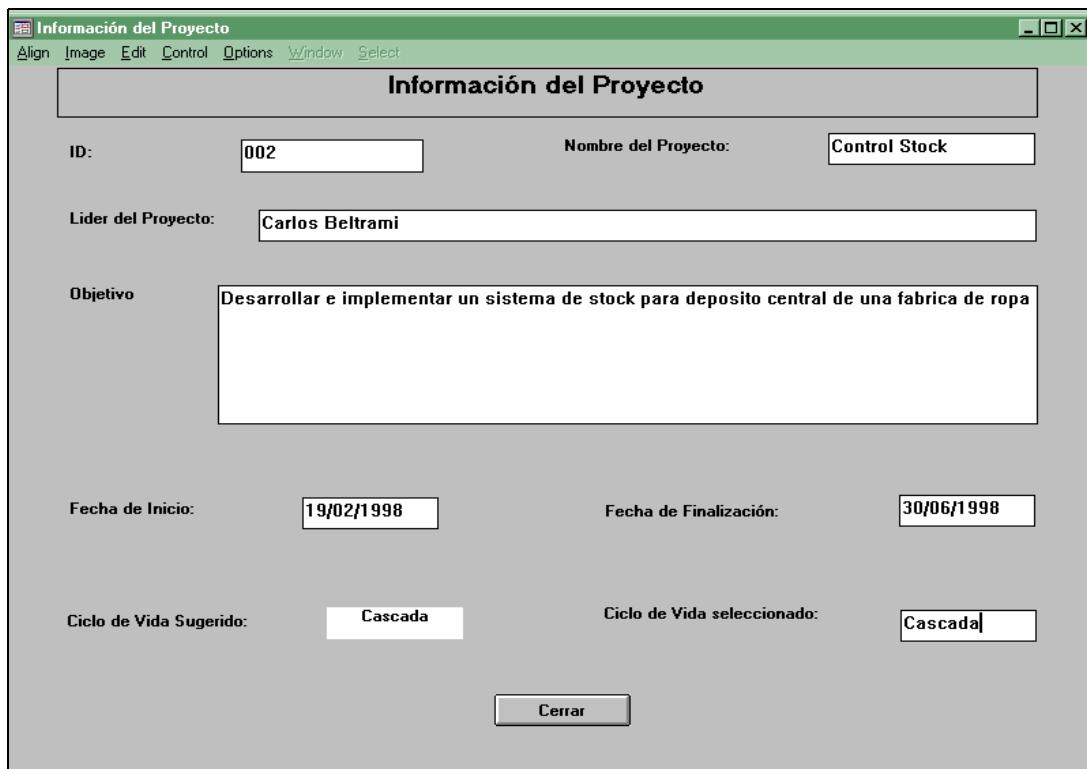


Figura 8-27: Actualización del CV propuesto por el Sistema Experto en los datos de Identificación del Proyecto

Se muestra a continuación la información guardada en el archivo “002.txt”, del caso de ejemplo.

**SISTEMA EXPERTO PARA LA SELECCION DE CICLOS DE VIDA DE PROYECTOS
DE SOFTWARE**

Copyright 2001 by Bibiana Rossi

Fecha: 21/07/2001

ID: 002

Proyecto: Control de stock

Líder de Proyecto: Lic. Carlos Beltrami

Objetivo: Desarrollar e Implementar un sistema. de control de stock para deposito central de una fabrica de ropa

Fecha de inicio: 19/02/1998

Fecha de finalización: 30/06/1998

Ciclo de Vida recomendado: Cascada

Ciclo de Vida seleccionado: Cascada

REGLA ER-R1

<SI TIPO-DEFINICION ES Formalmente Y Exhaustivamente Y Uniformemente
ENTONCES DEFINICION-REQUISITOS ES Clara Y DEFINICION-LIMITES ES Clara>

REGLA ER-R2

<SI DEFINICION-REQUISITOS ES Clara Y DEFINICION-LIMITES ES Clara Y
GRADO-DE-CERTIDUMBRE-INICIO ES Mayoritariamente
ENTONCES CV-PROPUESTO-REQUISITOS ES Cascada>

REGLA CV-R2

<SI RELACION-FASES-DESARROLLO ES Poca dependencia
ENTONCES RETROALIMENTACION-FASES-DESARROLLO ES Baja>

REGLA CV-R4

<SI PROGRE-FASES-DESARR ES Uniforme Y Secuencial Y
RETROALIM-FASES-DESARR ES Baja Y COMP-PREDOMINANTES ES Algorítmico
ENTONCES CV-PROP-APLICACION ES Cascada>

REGLA TA-R1

<SI NIVELES-COMPOSICION ES Pocos subsistemas
ENTONCES COMPLEJIDAD-SUBSISTEMAS ES Baja>

REGLA TA-R12

<SI TIPO-MODIFICACION ES Extensión
ENTONCES VARIABILIDAD-PROCESOS ES Alta>

REGLA TA-R16

<SI VARIABILIDAD-PROCESOS ES Alta

ENTONCES MODELADO-PROTOTIPO ES Conveniente>

REGLA GP-R2

<SI NIVEL-RIESGO ES Bajo

ENTONCES RIESGO-CASCADA ES Aceptable>

REGLA GP-R11

<SI REUSO-APLIC-FUTURA ES Muy necesario

ENTONCES SISTEMA-OO ES Existe>

REGLA GP-R12

<SI NIVEL-RIESGO ES Bajo

ENTONCES RIESGO-OO ES Aceptable>

REGLA GP-R17

<SI FORMALIDAD ES Poco Y CONVENIENCIA-METOD ES Medianamente probada

ENTONCES APPLICABILIDAD-OO ES Aplicable>

REGLA GP-R18

<SI APPLICABILIDAD-OO ES Aplicable Y SISTEMA-OO ES Existe Y RIESGO-OO ES Aceptable

ENTONCES CV-PROPU-GESTION ES Objetos>

REGLA GP-R22

<SI INTRODUCCION-GRADUAL ES Necesaria

ENTONCES APPLICABILIDAD-PROTOTIPO ES Aplicable>

REGLA GP-R28

<SI ENTREGAS ES Versión parcial

ENTONCES ENTREGA-INTERMEDIA ES Existe>

REGLA GP-R29

<SI ENTREGA-INTERMEDIA ES Existe

ENTONCES APPLICABILIDAD-OO ES Aplicable Y APPLICABILIDAD-PROTOTIPO ES Aplicable>

REGLA GP-R33

<SI PROC-CAMBIOS ES Acuerdos confirmados

ENTONCES APPLICABILIDAD-ESPIRAL ES Aplicable>

REGLA GP-R34

<SI RESPONSABILIDAD ES Organización propia

ENTONCES APPLICABILIDAD-ESPIRAL ES Aplicable>

REGLA GP-R37

<SI NECESIDAD-METOD ES Medianamente probada Y

FACTIB-METOD ES Medianamente probada

ENTONCES CONVENIENCIA-METOD ES Medianamente probada>

REGLA GP-R17

<SI FORMALIDAD ES Poco Y CONVENIENCIA-METOD ES Medianamente probada

ENTONCES APPLICABILIDAD-OO ES Aplicable>

REGLA GP-R18

<SI APPLICABILIDAD-OO ES Aplicable Y SISTEMA-OO ES Existe Y RIESGO-OO ES Aceptable

ENTONCES CV-PROPU-GESTION ES Objetos>
REGLA R3
<SI CV-PROPU-REQUISITOS ES Cascada Y CV-PROPU-APLIC ES Cascada Y
CV-PROPU-GESTION ES Objetos
ENTONCES CV-PROPU-PROYECTO ES Cascada>
REGLA ER-R2
<SI DEFINICION-REQUISITOS ES Clara Y DEFINICION-LIMITES ES Clara Y
GRADO-DE-CERTIDUMBRE-INICIO ES Mayoritariamente
ENTONCES CV-PROPUESTO-REQUISITOS ES Cascada>
REGLA R3
<SI CV-PROPU-REQUISITOS ES Cascada Y CV-PROPU-APLIC ES Cascada Y
CV-PROPU-GESTION ES Objetos
ENTONCES CV-PROPU-PROYECTO ES Cascada>
REGLA TA-R101
<SI PROGRESION-FASES-DESARR ES Uniforme Y Secuencial Y
RETROALIMENTACION-FASES-DESARR ES Baja Y
COMPORTAMIENTO-PREDOMINANTE ES Batch
ENTONCES CV-PROPU-APLIC ES Cascada>
REGLA R3
<SI CV-PROPU-REQUISITOS ES Cascada Y CV-PROPU-APLIC ES Cascada Y
CV-PROPU-GESTION ES Objetos
ENTONCES CV-PROPU-PROYECTO ES Cascada>
REGLA TA-R102
<SI PROGRESION-FASES-DESARR ES Uniforme Y Secuencial Y
COMPLEJIDAD-SUBSISTEMAS ES Baja Y
COMPONENTES-PREDOMINANTES ES Algorítmico
ENTONCES CV-PROPU-APLIC ES Cascada>
REGLA R3
<SI CV-PROPU-REQUISITOS ES Cascada Y CV-PROPU-APLIC ES Cascada Y
CV-PROPU-GESTION ES Objetos
ENTONCES CV-PROPU-PROYECTO ES Cascada>
REGLA TA-R103
<SI PROGRESION-FASES-DESARR ES Uniforme Y Secuencial Y
COMPLEJIDAD-SUBSISTEMAS ES Baja Y COMPORTAMIENTO-PREDOMINANTE ES Batch
ENTONCES CV-PROPU-APLIC ES Cascada>
REGLA R3
<SI CV-PROPU-REQUISITOS ES Cascada Y CV-PROPU-APLIC ES Cascada Y
CV-PROPU-GESTION ES Objetos
ENTONCES CV-PROPU-PROYECTO ES Cascada>

CICLO DE VIDA RECOMENDADO	
Se recomienda el Ciclo de Vida en Cascada	
Especificación de Requerimientos recomienda:	Cascada
Tipo de Aplicación recomienda:	Cascada
Gestión de Proyecto recomienda:	Objetos

El sistema ha respondido satisfactoriamente de acuerdo con la definición de los resultados esperados establecida al inicio del caso de prueba

Caso 2 Resultados del caso de prueba:	
Ciclo de vida esperado	Ciclo de vida sugerido por el sistema experto
CV Especificación Requerimientos Cascada	CV Especificación Requerimientos Cascada
CV Tipo de Aplicación Cascada	Tipo de Aplicación Cascada
CV Gestión de proyecto Objetos	Gestión de proyecto Objetos
CV del Proyecto Cascada	CV del Proyecto Cascada

Capítulo 9

Evaluación



El objetivo de la etapa de evaluación es garantizar la calidad del sistema experto. La calidad esta asociada con el funcionamiento correcto del sistema y que el sistema responda a las expectativas del usuario. La evaluación no es, una fase concreta de la ingeniería del conocimiento [Gómez, A. y otros 1997] sino un conjunto de actividades que ser realizan a lo largo de cada fase de desarrollo del sistema. Cada fase del proceso de desarrollo requiere una evaluación diferente aunque es conveniente utilizar los mismos casos de prueba a lo largo de todas las fases.

En este capítulo se presentan los casos de prueba con los que se ha evaluado el sistema. Se documenta la verificación y validación que se ha realizado y se mencionan los ítem que documentan la evaluación realizada a lo largo del desarrollo del sistema.

9.1 EVALUACIÓN DE LA ADQUISICIÓN DE CONOCIMIENTOS.

Para todas las sesiones cumplidas en la fase de adquisición de conocimientos se ha cumplido un ciclo de educación en el cual el ultimo de los pasos incluye la *Evaluación de la Sesión* (ver página 108). Las actividades en este paso consisten en evaluar los resultados obtenidos, la comprensión de los mismos por parte del ingeniero en conocimiento y la detección de información faltante para analizar en la siguiente sesión.

- **Entrevistas:** se encuentran las evaluaciones de las sesiones A.1, A.2, A.3 y A.4 (referencias en páginas 92 y 93).
- **Análisis de textos:** se encuentra la evaluación de la sesión B.1 (referencias en página 93).
- **Método Delphi:** se encuentran las evaluaciones de las sesiones C.1, C.2 y C.3 (referencias en página 93).
- **Técnica Nominal de Grupo:** se encuentra la evaluación de la sesión C.3 (referencias en pagina 93).
- **Emparrillado:** si bien la evaluación de la parrilla se concretó en una entrevista con el experto, el objetivo de dicha entrevista fue evaluar los resultados de la aplicación de la técnica por eso está asociada esa evaluación con esta técnica. Se encuentran referencias de la evaluación de las sesiones D.1 y A.4 (referencias en la página 93).

9.2 EVALUACIÓN DE LA CONCEPTUALIZACIÓN DE CONOCIMIENTOS.

Asegurar la calidad del modelo conceptual es una práctica de fundamental importancia. Usualmente el experto no entiende los formalismos de representación, de ahí la importancia de que el experto apruebe el modelo conceptual. [Sierra, A. 1996]. En este caso ha habido varias instancias de evaluación concretadas en entrevistas para la revisión parcial de cada uno de los modelos como así también para la Comprobación final de los distintos tipos de conocimientos:

- Comprobación de los conocimientos estratégicos: Se encuentran referencias de la evaluación en la página 180. En las revisiones parciales a través de entrevistas se evaluó el modelo de descomposición funcional.
- Comprobación de los conocimientos tácticos: Se encuentran referencias de la evaluación en la página 215. En las revisiones parciales a través de entrevistas se revisaron detalladamente las seudorreglas.
- Comprobación de los conocimientos fácticos: Se encuentran referencias de la evaluación en la página 215. En las revisiones parciales se revisaron: el Glosario de Términos, el Diccionario de Conceptos, la Tabla de Concepto Atributo Valor, la Definición de los Atributos
- Comprobación del Modelo dinámico y del Modelo estático: una vez terminados fueron revisados por el experto principal.
- Comprobación del Mapa de Conocimientos y de la fase de Conceptualización: el experto seleccionó los casos de prueba 1-1. 1-2, 2, 3, 4 y 6. Se encuentran referencias de la comprobación en la página 268.

9.3 EVALUACIÓN DE LA FORMALIZACIÓN DE CONOCIMIENTOS.

En este caso en particular, el experto entiende de formalismos computacionales lo que fortaleció la evaluación de la formalización respecto de la conceptualización. El experto seleccionó los mismos casos de prueba usados en la comprobación del Mapa de Conocimiento: 1-1. 1-2, 2, 3, 4 y 6

La evaluación se ha realizado teniendo en cuenta los criterios propuestos por Gómez [Gómez, A. y otros 1997] y ha consistido en:

- Marcos: se ha revisado si coincide con el modelo conceptual, y se analizaron Redundancias, Incompletud e Inconsistencia.
- Reglas: se han revisado analizando Redundancia, Identidad, Subsunción o reglas embebidas, Condiciones SI innecesarias, Callejones sin salida, Redundancia indirecta (reglas circulares), Reglas ausentes, Reglas inalcanzables, Inconsistencia, Incompatibilidades.

9.4 EVALUACIÓN DE LA IMPLEMENTACIÓN DE CONOCIMIENTOS.

Se ha revisado la definición de objetos y reglas en la herramienta.

Se ha aplicado el juego de ensayo probando cada uno de los casos propuestos por los expertos, en el sistema. Los casos presentados son casos reales que fueron liderados por los expertos consultados. Los proyectos ya fueron cumplidos. Se detallan en los apartados siguientes cada uno de los casos, detallando dos cuadros:

- Resultados del caso de prueba, donde se presentan los resultados esperados con el caso de prueba y los resultados obtenidos por el sistema experto.
- Valores del caso de prueba, donde se especifican los valores particulares de ese proyecto para cada área de análisis.

9.4.1 CASO EJEMPLO 1.1: Facturación Telefónica

Proyecto: 001-1 Telefónica

Objetivo del proyecto: Desarrollar e implementar un sistema de facturación para el tráfico telefónico.

Caso 1.1 Resultados del caso de prueba:	
Ciclo de vida esperado	Ciclo de vida sugerido por el sistema experto
CV Especificación Requerimientos Sin respuesta	CV Especificación Requerimientos - ninguno -
CV Tipo de Aplicación 1- Espiral 2- Objetos	Tipo de Aplicación Espiral Objetos
CV Gestión de proyecto Sin respuesta	Gestión de proyecto - ninguno -
CV del Proyecto No se pudo sacar ninguna conclusión de los datos	CV del Proyecto No se pudo sacar ninguna conclusión de los datos

El sistema experto, ha respondido satisfactoriamente de acuerdo con la definición de los casos de prueba.

Caso 1.1 Valores del caso de prueba		
Especificación Requerimientos		
ID	Afirmación / Pregunta	Opciones
ER1	Al comienzo del proyecto los requerimientos se encuentran definidos:	Parcialmente
ER2	El usuario ha explicitado los requisitos:	Formalmente Incompletos Desestructuradamente
ER3	La exigencia del usuario en el cumplimiento de los requerimientos en el producto final es:	Poco exigente
Tipo de Aplicación		
ID	Afirmación / Pregunta	Opciones
TA1	El tipo de aplicación se orienta al desarrollo de :	Otros
TA2	Los componentes predominantes de la aplicación son:	Matemático Algorítmico
TA3	El comportamiento predominante en la aplicación es:	Batch
TA4	¿La progresión entre las fases del desarrollo es secuencial y uniforme?	No
TA5	La relación entre las fases del proyecto es:	Poca dependencia
TA6	¿El tipo de aplicación requiere el desarrollo fuertemente integrado de hardware y software?	Si
TA7	¿El sistema a diseñar es innovador?	No
TA8	Para lograr una mejor comprensión del sistema los niveles de descomposición en subsistemas son:	Pocos
TA9	Para lograr una mejor manipulación del sistema los niveles de descomposición en subsistemas son:	Pocos
TA10	Se requiere detectar en etapas tempranas del proyecto problemas de:	Desconoce
TA11	El Diseño del sistema a desarrollar puede resolverse por:	Varias opciones
TA12	La Implementación del sistema a desarrollar puede resolverse por:	Varias opciones
TA13	Las modificaciones explícitamente previstas son:	Extensión Ampliación
Gestión de Proyecto		
ID	Afirmación / Pregunta	Opciones
GP1	El desarrollo del proyecto es responsabilidad de:	Terceros
GP2	¿El equipo de desarrollo tiene experiencia previa en el tipo de aplicación a desarrollar?	Si
GP3	Es factible usar metodologías de desarrollo:	Medianamente probadas
GP4	Es necesario usar metodologías de desarrollo:	Medianamente probadas
GP5	¿El equipo de desarrollo tiene experiencia con las técnicas de ingeniería del software que su usarán?	Si
GP6	El tipo de entrega que el usuario requiere de la aplicación es:	Versión parcial
GP7	El grado de participación del usuario en el desarrollo debe ser:	Medio
GP8	¿Es necesario facilitar la introducción gradual del sistema para los usuarios?	Si
GP9	Se requiere una gestión de proyecto:	Medianamente formal
GP10	El grado de control que se requiere de la gestión del proyecto es:	Medianamente ajustado

GP11	<i>¿La gestión de proyecto prevé definir explícitamente estrategias para reusar software existente?</i>	No
GP12	<i>¿Se prevé la necesidad de reutilizar los componentes del software a desarrollar en proyectos futuros?</i>	Medianamente necesario
GP13	<i>¿Existe un sistema previo desarrollado en objetos de forma tal que el proyecto actual es ..?</i>	Sin relación
GP14	<i>¿Existe la necesidad de reutilizar los componentes de un software ya existente desarrollado en objetos?</i>	No
GP15	<i>¿Se dispone de software para prototipar?</i>	No
GP16	<i>¿Es posible adquirir software para prototipar?</i>	Desconoce
GP17	<i>¿Existen dudas sobre la viabilidad del software?</i>	No
GP18	<i>¿Se estima la posibilidad de terminación prematura de proyectos no viables?</i>	No
GP19	<i>Se estima que el factor de riesgo en el desarrollo del sistema es:</i>	Medio
GP20	<i>¿Se estiman riesgos técnicos en el desarrollo del sistema?</i>	Si
GP21	<i>¿Se identifican los riesgos asociados con cada una de las alternativas?</i>	Desconoce
GP22	<i>¿Se cuenta con métodos y técnicas para evaluar el riesgo?</i>	Desconoce
GP23	<i>¿Se identifican las diferentes maneras de resolver los riesgos?</i>	Desconoce
GP24	<i>¿Es necesario definir explícitamente estrategias para realizar análisis de riesgo?</i>	No
GP25	<i>¿Es necesario volver a etapas anteriores en el desarrollo del sistema para la resolución de algún tópico nuevo de riesgo?</i>	Desconoce
GP26	<i>¿Es posible volver a etapas anteriores en el desarrollo del sistema cuando se identifican mejores alternativas?</i>	Desconoce
GP27	<i>¿Es necesario definir mecanismos explícitos para incorporar objetivos de calidad en el desarrollo del producto?</i>	No
GP28	<i>El uso de los mismos procedimientos para desarrollo y mantenimiento del producto software está previsto como:</i>	Indiferente
GP29	<i>¿Se requiere de acuerdos confirmados para los cambios a efectuarse durante el desarrollo del proyecto?</i>	Si

SISTEMA EXPERTO PARA LA SELECCION DE CICLOS DE VIDA DE PROYECTOS DE SOFTWARE
 Copyright 2001 by Bibiana Rossi

Fecha: 19/05/2001
 ID: 001-1
 Proyecto: Telefónica
 Líder de Proyecto: Lic. Laura Lucchini
 Objetivo: Desarrollar e implementar un sistema de facturación para el tráfico telefónico
 Fecha de inicio: 10/03/1998
 Fecha de finalización: 10/12/1998
 Ciclo de Vida recomendado: Ninguno
 Ciclo de Vida seleccionado: Ninguno

REGLA CV-R2
 <SI RELACION-FASES-DESARROLLO ES Poca dependencia
 ENTONCES RETROALIMENTACION-FASES-DESARROLLO ES Baja>

REGLA CV-R5

<SI PROGRE-FASES-DESARR ES No uniforme Y Secuencial

ENTONCES MODELADO-PROTOTIPO ES Conveniente>

REGLA TA-R1

<SI NIVELES-COMPOSICION ES Pocos subsistemas

ENTONCES COMPLEJIDAD-SUBSISTEMAS ES Baja>

REGLA TA-R4

<SI COMPONENTES-PREDOMINANTES ES Matematicos

ENTONCES MODELADO-OBJETOS ES Conveniente>

REGLA TA-R12

<SI TIPO-MODIFICACION ES Extension

ENTONCES VARIABILIDAD-PROCESOS ES Alta>

REGLA TA-R13

<SI TIPO-MODIFICACION ES Ampliacion

ENTONCES VARIABILIDAD-PROCESOS ES Alta>

REGLA TA-R16

<SI VARIABILIDAD-PROCESOS ES Alta

ENTONCES MODELADO-PROTOTIPO ES Conveniente>

REGLA TA-R20

<SI MODELADO-OBJETOS ES Conveniente Y MODELADO-PROTOTIPO ES Conveniente

ENTONCES CV-PROPU-APLICACION ES Objetos>

REGLA TA-R21

<SI MODELADO-OBJETOS ES Conveniente Y MODELADO-PROTOTIPO ES Conveniente Y

INTREGRACION-HW-SW ES Fuertemente

ENTONCES CV-PROPU-APLICACION ES Espiral>

REGLA TA-R22

<SI POSIBILIDAD-DISEÑO ES Varias opciones

ENTONCES OPCIONALIDAD ES Múltiple>

REGLA TA-R23

<SI POSIBILIDAD-IMPLEMENTACION ES Varias opciones

ENTONCES OPCIONALIDAD ES Múltiple>

REGLA TA-R24

<SI MODELADO-OBJETOS ES Conveniente Y MODELADO-PROTOTIPO ES Conveniente Y

OPCIONALIDAD ES Múltiple

ENTONCES CV-PROPU-APLICACION ES Espiral>

REGLA GP-R13

<SI NIVEL-RIESGO ES Mediano

ENTONCES RIESGO-OO ES Aceptable>

REGLA GP-R15

<SI FORMALIDAD ES Medianamente Y CONVENIENCIA-METOD ES Medianamente probada

ENTONCES APLICABILIDAD-OO ES Aplicable>
 REGLA GP-R22
 <SI INTRODUCCION-GRADUAL ES Necesaria
 ENTONCES APLICABILIDAD-PROTOTIPO ES Aplicable>
 REGLA GP-R28
 <SI ENTREGAS ES Versión parcial
 ENTONCES ENTREGA-INTERMEDIA ES Existe>
 REGLA GP-R29
 <SI ENTREGA-INTERMEDIA ES Existe
 ENTONCES APLICABILIDAD-OO ES Aplicable Y APLICABILIDAD-PROTOTIPO ES Aplicable>
 REGLA GP-R33
 <SI PROC-CAMBIOS ES Acuerdos confirmados
 ENTONCES APLICABILIDAD-ESPIRAL ES Aplicable>
 REGLA GP-R36
 <SI CATEGORIA-RIESGO ES Técnico
 ENTONCES FACTOR-RIESGO ES Existe>
 REGLA GP-R37
 <SI NECESIDAD-METOD ES Medianamente probada Y
 FACTIB-METOD ES Medianamente probada
 ENTONCES CONVENIENCIA-METOD ES Medianamente probada>
 REGLA GP-R15
 <SI FORMALIDAD ES Medianamente Y CONVENIENCIA-METOD ES Medianamente probada
 ENTONCES APLICABILIDAD-OO ES Aplicable>
 REGLA GP-R50
 <SI NIVEL-RIESGO ES Mediano
 ENTONCES RIESGO-ESPIRAL ES Aceptable>
 REGLA GP-R52
 <SI RIESGO-ESPIRAL ES Aceptable
 ENTONCES FACTOR-RIESGO ES Existe>

CICLO DE VIDA RECOMENDADO

No se pudo sacar ninguna conclusión de los datos

Especificación de Requerimientos recomienda:	- ninguno -
Tipo de Aplicación recomienda:	Espiral Objetos
Gestión de Proyecto recomienda:	- ninguno -

9.4.2 CASO EJEMPLO 1.2: Facturación Telefónica

Proyecto: 001-2 Telefónica

Objetivo del proyecto: Desarrollar e implementar un sistema de facturación para el tráfico telefónico.

Caso 1.2 Resultados del caso de prueba:	
Ciclo de vida esperado	Ciclo de vida sugerido por el sistema experto
CV Especificación Requerimientos 1- Espiral 2- Objetos	CV Especificación Requerimientos Espiral Objetos
CV Tipo de Aplicación 1- Espiral 2- Objetos	Tipo de Aplicación Espiral Objetos
CV Gestión de proyecto Espiral	Gestión de proyecto Espiral
CV del Proyecto Espiral	CV del Proyecto Se recomienda el Ciclo de Vida en Espiral

El sistema experto, ha respondido satisfactoriamente de acuerdo con la definición de los casos de prueba.

Caso 1.2 Valores del caso de prueba		
Especificación Requerimientos		
ID	Afirmación / Pregunta	Opciones
ER1	Al comienzo del proyecto los requerimientos se encuentran definidos:	Parcialmente
ER2	El usuario ha explicitado los requisitos:	Informalmente Incompletos Desestructuradamente
ER3	La exigencia del usuario en el cumplimiento de los requerimientos en el producto final es:	Poco exigente

Tipo de Aplicación		
ID	Afirmación / Pregunta	Opciones
TA1	El tipo de aplicación se orienta al desarrollo de :	Otros
TA2	Los componentes predominantes de la aplicación son:	Matemático - Algorítmico
TA3	El comportamiento predominante en la aplicación es:	Batch
TA4	¿La progresión entre las fases del desarrollo es secuencial y uniforme?	No
TA5	La relación entre las fases del proyecto es:	Poca dependencia
TA6	¿El tipo de aplicación requiere el desarrollo fuertemente integrado de hardware y software?	Si
TA7	¿El sistema a diseñar es innovador?	No

TA8	Para lograr una mejor comprensión del sistema los niveles de descomposición en subsistemas son:	Pocos
TA9	Para lograr una mejor manipulación del sistema los niveles de descomposición en subsistemas son:	Pocos
TA10	Se requiere detectar en etapas tempranas del proyecto problemas de:	Desconoce
TA11	El Diseño del sistema a desarrollar puede resolverse por:	Varias opciones
TA12	La Implementación del sistema a desarrollar puede resolverse por:	Varias opciones
TA13	Las modificaciones explícitamente previstas son:	Extensión - Ampliación
Gestión de Proyecto		
ID	Afirmación / Pregunta	Opciones
GP1	El desarrollo del proyecto es responsabilidad de:	Terceros
GP2	¿El equipo de desarrollo tiene experiencia previa en el tipo de aplicación a desarrollar?	No
GP3	Es factible usar metodologías de desarrollo:	Medianamente probadas
GP4	Es necesario usar metodologías de desarrollo:	Medianamente probadas
GP5	¿El equipo de desarrollo tiene experiencia con las técnicas de ingeniería del software que su usarán?	Si
GP6	El tipo de entrega que el usuario requiere de la aplicación es:	Versión gradual
GP7	El grado de participación del usuario en el desarrollo debe ser:	Alto
GP8	¿Es necesario facilitar la introducción gradual del sistema para los usuarios?	Si
GP9	Se requiere una gestión de proyecto:	Medianamente formal
GP10	El grado de control que se requiere de la gestión del proyecto es:	Medianamente ajustado
GP11	¿La gestión de proyecto prevé definir explícitamente estrategias para reusar software existente?	Desconoce
GP12	¿Se prevé la necesidad de reutilizar los componentes del software a desarrollar en proyectos futuros?	Poco necesario
GP13	¿Existe un sistema previo desarrollado en objetos de forma tal que el proyecto actual es ..?	Sin relación
GP14	¿Existe la necesidad de reutilizar los componentes de un software ya existente desarrollado en objetos?	No
GP15	¿Se dispone de software para prototipar?	Desconoce
GP16	¿Es posible adquirir software para prototipar?	Si
GP17	¿Existen dudas sobre la viabilidad del software?	No
GP18	¿Se estima la posibilidad de terminación prematura de proyectos no viables?	Desconoce
GP19	Se estima que el factor de riesgo en el desarrollo del sistema es:	Medio
GP20	¿Se estiman riesgos técnicos en el desarrollo del sistema?	Si
GP21	¿Se identifican los riesgos asociados con cada una de las alternativas?	Si
GP22	¿Se cuenta con métodos y técnicas para evaluar el riesgo?	Si
GP23	¿Se identifican las diferentes maneras de resolver los riesgos?	Si
GP24	¿Es necesario definir explícitamente estrategias para realizar análisis de riesgo?	Si
GP25	¿Es necesario volver a etapas anteriores en el desarrollo del sistema para la resolución de algún tópico nuevo de riesgo?	Desconoce
GP26	¿Es posible volver a etapas anteriores en el desarrollo del sistema cuando se identifican mejores alternativas?	Desconoce
GP27	¿Es necesario definir mecanismos explícitos para incorporar objetivos de calidad en el desarrollo del producto?	Desconoce
GP28	El uso de los mismos procedimientos para desarrollo y mantenimiento del producto software está previsto como:	Necesario - Factible
GP29	¿Se requiere de acuerdos confirmados para los cambios a efectuarse durante el desarrollo del proyecto?	Si

SISTEMA EXPERTO PARA LA SELECCION DE CICLOS DE VIDA DE PROYECTOS DE SOFTWARE

Copyright 2001 by Bibiana Rossi

Fecha: 19/05/2001

ID: 001-2

Proyecto: Telefónica

Líder de Proyecto: Lic. Laura Lucchini

Objetivo: Desarrollar e implementar un sistema de facturación para el tráfico telefónico

Fecha de inicio: 10/03/1998

Fecha de finalización: 10/12/1998

Ciclo de Vida recomendado: Espiral

Ciclo de Vida seleccionado: Espiral

REGLA ER-R3

<SI TIPO-DEFINICION ES Informalmente Y Incompleto Y Desestructuradamente
ENTONCES DEFINICION-REQUISITOS ES Incierta Y DEFINICION-LIMITES ES Incierta>

REGLA ER-R4

<SI DEFINICION-REQUISITOS ES Incierta Y DEFINICION-LIMITES ES Incierta Y
GRADO-DE-CERTIDUMBRE-INICIO ES Parcialmente
ENTONCES CV-PROPUESTO-REQUISITOS ES Espiral Y Objetos>

REGLA CV-R2

<SI RELACION-FASES-DESARROLLO ES Poca dependencia
ENTONCES RETROALIMENTACION-FASES-DESARROLLO ES Baja>

REGLA CV-R5

<SI PROGRE-FASES-DESARR ES No uniforme Y Secuencial
ENTONCES MODELADO-PROTOTIPO ES Conveniente>

REGLA TA-R1

<SI NIVELES-COMPOSICION ES Pocos subsistemas
ENTONCES COMPLEJIDAD-SUBSISTEMAS ES Baja>

REGLA TA-R4

<SI COMPONENTES-PREDOMINANTES ES Matemáticos
ENTONCES MODELADO-OBJETOS ES Conveniente>

REGLA TA-R12

<SI TIPO-MODIFICACION ES Extensión
ENTONCES VARIABILIDAD-PROCESOS ES Alta>

REGLA TA-R13

<SI TIPO-MODIFICACION ES Ampliación
ENTONCES VARIABILIDAD-PROCESOS ES Alta>

REGLA TA-R16

<SI VARIABILIDAD-PROCESOS ES Alta
 ENTONCES MODELADO-PROTOTIPO ES Conveniente>

REGLA TA-R20

<SI MODELADO-OBJETOS ES Conveniente Y MODELADO-PROTOTIPO ES Conveniente
 ENTONCES CV-PROPU-APLICACION ES Objetos>

REGLA TA-R21

<SI MODELADO-OBJETOS ES Conveniente Y MODELADO-PROTOTIPO ES Conveniente Y
 INTREGRACION-HW-SW ES Fuertemente
 ENTONCES CV-PROPU-APLICACION ES Espiral>

REGLA TA-R22

<SI POSIBILIDAD-DISEÑO ES Varias opciones
 ENTONCES OPCIONALIDAD ES Múltiple>

REGLA TA-R23

<SI POSIBILIDAD-IMPLEMENTACION ES Varias opciones
 ENTONCES OPCIONALIDAD ES Múltiple>

REGLA TA-R24

<SI MODELADO-OBJETOS ES Conveniente Y MODELADO-PROTOTIPO ES Conveniente Y
 OPCIONALIDAD ES Múltiple
 ENTONCES CV-PROPU-APLICACION ES Espiral>

REGLA GP-R13

<SI NIVEL-RIESGO ES Mediano
 ENTONCES RIESGO-OO ES Aceptable>

REGLA GP-R15

<SI FORMALIDAD ES Medianamente Y CONVENIENCIA-METOD ES Medianamente probada
 ENTONCES APPLICABILIDAD-OO ES Aplicable>

REGLA GP-R20

<SI SOFT-PROTOTIPO ES Factible
 ENTONCES FACTIBILIDAD-PROTOTIPO ES Factible>

REGLA GP-R22

<SI INTRODUCCION-GRADUAL ES Necesaria
 ENTONCES APPLICABILIDAD-PROTOTIPO ES Aplicable>

REGLA GP-R23

<SI PARTICIPACION-USUARIO ES Fuerte
 ENTONCES APPLICABILIDAD-PROTOTIPO ES Aplicable>

REGLA GP-R25

<SI ENTREGAS ES Versión gradual
 ENTONCES ENTREGA-INTERMEDIA ES Existe>

REGLA GP-R27

<SI EXPERIENCIA-PREVIA ES No hay
 ENTONCES APPLICABILIDAD-PROTOTIPO ES Aplicable>

REGLA GP-R29
 <SI ENTREGA-INTERMEDIA ES Existe
 ENTONCES APPLICABILIDAD-OO ES Aplicable Y APPLICABILIDAD-PROTOTIPO ES Aplicable>

REGLA GP-R31
 <SI PROC-DES-MANTE ES Necesario Y Factible
 ENTONCES PROCEDIMIENTOS-DM ES Conveniente>

REGLA GP-R32
 <SI PROCEDIMIENTOS-DM ES Conveniente
 ENTONCES APPLICABILIDAD-ESPIRAL ES Aplicable>

REGLA GP-R33
 <SI PROC-CAMBIOS ES Acuerdos confirmados
 ENTONCES APPLICABILIDAD-ESPIRAL ES Aplicable>

REGLA GP-R36
 <SI CATEGORIA-RIESGO ES Tecnico
 ENTONCES FACTOR-RIESGO ES Existe>

REGLA GP-R37
 <SI NECESIDAD-METOD ES Medianamente probada Y
 FACTIB-METOD ES Medianamente probada
 ENTONCES CONVENIENCIA-METOD ES Medianamente probada>

REGLA GP-R15
 <SI FORMALIDAD ES Medianamente Y CONVENIENCIA-METOD ES Medianamente probada
 ENTONCES APPLICABILIDAD-OO ES Aplicable>

REGLA GP-R38
 <SI RIESGO-ALTERNATIVAS ES Se identifican Y
 IDENTIFICACION-ALTERNATIVAS ES Se identifican
 ENTONCES HABILIDAD-RIESGO ES Existe>

REGLA GP-R43
 <SI FACTOR-RIESGO ES Existe Y ANALISIS-RIESGO ES Estrategias explicitas
 ENTONCES APPLICABILIDAD-ESPIRAL ES Aplicable>

REGLA GP-R45
 <SI APPLICAB-PROTOTIPO ES Aplicable Y FACTIB-PROTOTIPO ES Factible Y
 RIESGO ES Evaluable Y APPLICAB-ESPIRAL ES Aplicable
 ENTONCES CV-PROPU-GESTION ES Espiral>

REGLA R2
 <SI CV-PROPU-REQUISITOS ES Espiral Y CV-PROPU-APLIC ES Espiral Y
 CV-PROPU-GESTION ES Espiral
 ENTONCES CV-PROPU-PROYECTO ES Espiral>

REGLA ER-R4
 <SI DEFINICION-REQUISITOS ES Incierta Y DEFINICION-LIMITES ES Incierta Y
 GRADO-DE-CERTIDUMBRE-INICIO ES Parcialmente

ENTONCES CV-PROPUESTO-REQUISITOS ES Espiral Y Objetos>

REGLA R15

<SI CV-PROPU-REQUISITOS ES Objetos Y CV-PROPU-APLIC ES Objetos Y
CV-PROPU-GESTION ES Espiral

ENTONCES CV-PROPU-PROYECTO ES Objetos>

REGLA R2

<SI CV-PROPU-REQUISITOS ES Espiral Y CV-PROPU-APLIC ES Espiral Y
CV-PROPU-GESTION ES Espiral

ENTONCES CV-PROPU-PROYECTO ES Espiral>

REGLA R24

<SI CV-PROPU-REQUISITOS ES Espiral Y CV-PROPU-APLIC ES Objetos Y
CV-PROPU-GESTION ES Espiral

ENTONCES CV-PROPU-PROYECTO ES Espiral>

REGLA R26

<SI CV-PROPU-REQUISITOS ES Objetos Y CV-PROPU-APLIC ES Espiral Y
CV-PROPU-GESTION ES Espiral

ENTONCES CV-PROPU-PROYECTO ES Espiral>

REGLA GP-R50

<SI NIVEL-RIESGO ES Mediano

ENTONCES RIESGO-ESPIRAL ES Aceptable>

REGLA GP-R52

<SI RIESGO-ESPIRAL ES Aceptable

ENTONCES FACTOR-RIESGO ES Existe>

REGLA GP-R43

<SI FACTOR-RIESGO ES Existe Y ANALISIS-RIESGO ES Estrategias explicitas

ENTONCES APPLICABILIDAD-ESPIRAL ES Aplicable>

REGLA GP-R45

<SI APPLICAB-PROTOTIPO ES Aplicable Y FACTIB-PROTOTIPO ES Factible Y
RIESGO ES Evaluable Y APPLICAB-ESPIRAL ES Aplicable

ENTONCES CV-PROPU-GESTION ES Espiral>

REGLA R2

<SI CV-PROPU-REQUISITOS ES Espiral Y CV-PROPU-APLIC ES Espiral Y
CV-PROPU-GESTION ES Espiral

ENTONCES CV-PROPU-PROYECTO ES Espiral>

REGLA R15

<SI CV-PROPU-REQUISITOS ES Objetos Y CV-PROPU-APLIC ES Objetos Y
CV-PROPU-GESTION ES Espiral

ENTONCES CV-PROPU-PROYECTO ES Objetos>

REGLA R24

<SI CV-PROPU-REQUISITOS ES Espiral Y CV-PROPU-APLIC ES Objetos Y

CV-PROPU-GESTION ES Espiral
 ENTONCES CV-PROPU-PROYECTO ES Espiral>

REGLA R26
 <SI CV-PROPU-REQUISITOS ES Objetos Y CV-PROPU-APLIC ES Espiral Y
 CV-PROPU-GESTION ES Espiral
 ENTONCES CV-PROPU-PROYECTO ES Espiral>

REGLA GP-R56
 <SI FACTOR-RIESGO ES Existe Y HABILIDAD-RIESGO ES Existe Y
 TECNICAS-AR ES Se dispone
 ENTONCES RIESGO ES Evaluable>

REGLA GP-R45
 <SI APPLICAB-PROTOTIPO ES Aplicable Y FACTIB-PROTOTIPO ES Factible Y
 RIESGO ES Evaluable Y APPLICAB-ESPIRAL ES Aplicable
 ENTONCES CV-PROPU-GESTION ES Espiral>

REGLA R2
 <SI CV-PROPU-REQUISITOS ES Espiral Y CV-PROPU-APLIC ES Espiral Y
 CV-PROPU-GESTION ES Espiral
 ENTONCES CV-PROPU-PROYECTO ES Espiral>

REGLA R15
 <SI CV-PROPU-REQUISITOS ES Objetos Y CV-PROPU-APLIC ES Objetos Y
 CV-PROPU-GESTION ES Espiral
 ENTONCES CV-PROPU-PROYECTO ES Objetos>

REGLA R24
 <SI CV-PROPU-REQUISITOS ES Espiral Y CV-PROPU-APLIC ES Objetos Y
 CV-PROPU-GESTION ES Espiral
 ENTONCES CV-PROPU-PROYECTO ES Espiral>

REGLA R26
 <SI CV-PROPU-REQUISITOS ES Objetos Y CV-PROPU-APLIC ES Espiral Y
 CV-PROPU-GESTION ES Espiral
 ENTONCES CV-PROPU-PROYECTO ES Espiral>

CICLO DE VIDA RECOMENDADO

Se recomienda el Ciclo de Vida en Espiral

Especificación de Requerimientos recomienda:	Espiral Objetos
Tipo de Aplicación recomienda:	Espiral Objetos
Gestión de Proyecto recomienda:	Espiral

9.4.3 CASO EJEMPLO 2: Control de Stock

Proyecto: 002 Fabrica de ropa de hombre

Objetivo del proyecto: Desarrollar e implementar un sistema de control de stock para el depósito central y los negocios de venta al público.

Caso 2 Resultados del caso de prueba:	
Ciclo de vida esperado	Ciclo de vida sugerido por el sistema experto
CV Especificación Requerimientos Cascada	CV Especificación Requerimientos Cascada
CV Tipo de Aplicación Cascada	Tipo de Aplicación Cascada
CV Gestión de proyecto Objetos	Gestión de proyecto Objetos
CV del Proyecto Cascada	CV del Proyecto Se recomienda el Ciclo de Vida en Cascada

El sistema experto, ha respondido satisfactoriamente de acuerdo con la definición de los casos de prueba.

Caso 2 Valores del caso de prueba		
Especificación Requerimientos		
ID	Afirmación / Pregunta	Opciones
ER1	<i>Al comienzo del proyecto los requerimientos se encuentran definidos:</i>	Mayoritariamente
ER2	<i>El usuario ha explicitado los requisitos:</i>	Formalmente, Exhaustivamente, Uniformemente
ER3	<i>La exigencia del usuario en el cumplimiento de los requerimientos en el producto final es:</i>	Poco exigente
Tipo de Aplicación		
ID	Afirmación / Pregunta	Opciones
TA1	<i>El tipo de aplicación se orienta al desarrollo de :</i>	Otros
TA2	<i>Los componentes predominantes de la aplicación son:</i>	Algorítmico
TA3	<i>El comportamiento predominante en la aplicación es:</i>	Batch
TA4	<i>¿La progresión entre las fases del desarrollo es secuencial y uniforme?</i>	Si
TA5	<i>La relación entre las fases del proyecto es:</i>	Poca dependencia
TA6	<i>¿El tipo de aplicación requiere el desarrollo fuertemente integrado de hardware y software?</i>	No
TA7	<i>¿El sistema a diseñar es innovador?</i>	No
TA8	<i>Para lograr una mejor comprensión del sistema los niveles de descomposición en subsistemas son:</i>	Pocos
TA9	<i>Para lograr una mejor manipulación del sistema los niveles de descomposición en subsistemas son:</i>	Pocos

TA10	<i>Se requiere detectar en etapas tempranas del proyecto problemas de:</i>	Desconoce
TA11	<i>El Diseño del sistema a desarrollar puede resolverse por:</i>	Pocas opciones
TA12	<i>La Implementación del sistema a desarrollar puede resolverse por:</i>	Pocas opciones
TA13	<i>Las modificaciones explícitamente previstas son:</i>	Extensión

Gestión de Proyecto

ID	Afirmación / Pregunta	Opciones
GP1	<i>El desarrollo del proyecto es responsabilidad de:</i>	La organización
GP2	<i>¿El equipo de desarrollo tiene experiencia previa en el tipo de aplicación a desarrollar?</i>	Si
GP3	<i>Es factible usar metodologías de desarrollo:</i>	Medianamente probadas
GP4	<i>Es necesario usar metodologías de desarrollo:</i>	Medianamente probadas
GP5	<i>¿El equipo de desarrollo tiene experiencia con las técnicas de ingeniería del software que su usarán?</i>	Si
GP6	<i>El tipo de entrega que el usuario requiere de la aplicación es:</i>	Versión parcial
GP7	<i>El grado de participación del usuario en el desarrollo debe ser:</i>	Medio
GP8	<i>¿Es necesario facilitar la introducción gradual del sistema para los usuarios?</i>	Si
GP9	<i>Se requiere una gestión de proyecto:</i>	Poco formal
GP10	<i>El grado de control que se requiere de la gestión del proyecto es:</i>	Poco ajustado
GP11	<i>¿La gestión de proyecto prevé definir explícitamente estrategias para reusar software existente?</i>	No
GP12	<i>¿Se prevé la necesidad de reutilizar los componentes del software a desarrollar en proyectos futuros?</i>	Muy necesario
GP13	<i>¿Existe un sistema previo desarrollado en objetos de forma tal que el proyecto actual es ..?</i>	Sin relación
GP14	<i>¿Existe la necesidad de reutilizar los componentes de un software ya existente desarrollado en objetos?</i>	No
GP15	<i>¿Se dispone de software para prototipar?</i>	No
GP16	<i>¿Es posible adquirir software para prototipar?</i>	Desconoce
GP17	<i>¿Existen dudas sobre la viabilidad del software?</i>	No
GP18	<i>¿Se estima la posibilidad de terminación prematura de proyectos no viables?</i>	No
GP19	<i>Se estima que el factor de riesgo en el desarrollo del sistema es:</i>	Bajo
GP20	<i>¿Se estiman riesgos técnicos en el desarrollo del sistema?</i>	No
GP21	<i>¿Se identifican los riesgos asociados con cada una de las alternativas?</i>	Desconoce
GP22	<i>¿Se cuenta con métodos y técnicas para evaluar el riesgo?</i>	No
GP23	<i>¿Se identifican las diferentes maneras de resolver los riesgos?</i>	No
GP24	<i>¿Es necesario definir explícitamente estrategias para realizar análisis de riesgo?</i>	Desconoce
GP25	<i>¿Es necesario volver a etapas anteriores en el desarrollo del sistema para la resolución de algún tópico nuevo de riesgo?</i>	Desconoce
GP26	<i>¿Es posible volver a etapas anteriores en el desarrollo del sistema cuando se identifican mejores alternativas?</i>	Desconoce
GP27	<i>¿Es necesario definir mecanismos explícitos para incorporar objetivos de calidad en el desarrollo del producto?</i>	Si
GP28	<i>El uso de los mismos procedimientos para desarrollo y mantenimiento del producto software está previsto como:</i>	Indiferente
GP29	<i>¿Se requiere de acuerdos confirmados para los cambios a efectuarse durante el desarrollo del proyecto?</i>	Si

SISTEMA EXPERTO PARA LA SELECCION DE CICLOS DE VIDA DE PROYECTOS DE SOFTWARE

Copyright 2001 by Bibiana Rossi

Fecha: 19/05/2001

ID: 002

Proyecto: Control de stock

Lider de Proyecto: Lic. Carlos Beltrami

Objetivo: Desarrollar e Implementar un Sist. de control de stock para deposito central y negocios.

Fecha de inicio: 06/08/1999

Fecha de finalización: 22/03/2000

Ciclo de Vida recomendado: Cascada

Ciclo de Vida seleccionado: Cascada

REGLA ER-R1

<SI TIPO-DEFINICION ES Formalmente Y Exhaustivamente Y Uniformemente
ENTONCES DEFINICION-REQUISITOS ES Clara Y DEFINICION-LIMITES ES Clara>

REGLA ER-R2

<SI DEFINICION-REQUISITOS ES Clara Y DEFINICION-LIMITES ES Clara Y
GRADO-DE-CERTIDUMBRE-INICIO ES Mayoritariamente
ENTONCES CV-PROPUESTO-REQUISITOS ES Cascada>

REGLA CV-R2

<SI RELACION-FASES-DESARROLLO ES Poca dependencia
ENTONCES RETROALIMENTACION-FASES-DESARROLLO ES Baja>

REGLA CV-R4

<SI PROGRE-FASES-DESARR ES Uniforme Y Secuencial Y
RETROALIM-FASES-DESARR ES Baja Y COMP-PREDOMINANTES ES Algorítmico
ENTONCES CV-PROP-APLICACION ES Cascada>

REGLA TA-R1

<SI NIVELES-COMPOSICION ES Pocos subsistemas
ENTONCES COMPLEJIDAD-SUBSISTEMAS ES Baja>

REGLA TA-R12

<SI TIPO-MODIFICACION ES Extensión
ENTONCES VARIABILIDAD-PROCESOS ES Alta>

REGLA TA-R16

<SI VARIABILIDAD-PROCESOS ES Alta
ENTONCES MODELADO-PROTOTIPO ES Conveniente>

REGLA GP-R2

<SI NIVEL-RIESGO ES Bajo
ENTONCES RIESGO-CASCADA ES Aceptable>

REGLA GP-R11

<SI REUSO-APLIC-FUTURA ES Muy necesario
 ENTONCES SISTEMA-OO ES Existe>

REGLA GP-R12

<SI NIVEL-RIESGO ES Bajo
 ENTONCES RIESGO-OO ES Aceptable>

REGLA GP-R17

<SI FORMALIDAD ES Poco Y CONVENIENCIA-METOD ES Medianamente probada
 ENTONCES APLICABILIDAD-OO ES Aplicable>

REGLA GP-R18

<SI APLICABILIDAD-OO ES Aplicable Y SISTEMA-OO ES Existe Y RIESGO-OO ES Aceptable
 ENTONCES CV-PROPU-GESTION ES Objetos>

REGLA GP-R22

<SI INTRODUCCION-GRADUAL ES Necesaria
 ENTONCES APLICABILIDAD-PROTOTIPO ES Aplicable>

REGLA GP-R28

<SI ENTREGAS ES Version parcial
 ENTONCES ENTREGA-INTERMEDIA ES Existe>

REGLA GP-R29

<SI ENTREGA-INTERMEDIA ES Existe
 ENTONCES APLICABILIDAD-OO ES Aplicable Y APLICABILIDAD-PROTOTIPO ES Aplicable>

REGLA GP-R33

<SI PROC-CAMBIOS ES Acuerdos confirmados
 ENTONCES APLICABILIDAD-ESPIRAL ES Aplicable>

REGLA GP-R34

<SI RESPONSABILIDAD ES Organización propia
 ENTONCES APLICABILIDAD-ESPIRAL ES Aplicable>

REGLA GP-R37

<SI NECESIDAD-METOD ES Medianamente probada Y
 FACTIB-METOD ES Medianamente probada
 ENTONCES CONVENIENCIA-METOD ES Medianamente probada>

REGLA GP-R17

<SI FORMALIDAD ES Poco Y CONVENIENCIA-METOD ES Medianamente probada
 ENTONCES APLICABILIDAD-OO ES Aplicable>

REGLA GP-R18

<SI APLICABILIDAD-OO ES Aplicable Y SISTEMA-OO ES Existe Y RIESGO-OO ES Aceptable
 ENTONCES CV-PROPU-GESTION ES Objetos>

REGLA R3

<SI CV-PROPU-REQUISITOS ES Cascada Y CV-PROPU-APLIC ES Cascada Y
 CV-PROPU-GESTION ES Objetos
 ENTONCES CV-PROPU-PROYECTO ES Cascada>

REGLA ER-R2

<SI DEFINICION-REQUISITOS ES Clara Y DEFINICION-LIMITES ES Clara Y
GRADO-DE-CERTIDUMBRE-INICIO ES Mayoritariamente
ENTONCES CV-PROPUESTO-REQUISITOS ES Cascada>

REGLA R3

<SI CV-PROPU-REQUISITOS ES Cascada Y CV-PROPU-APLIC ES Cascada Y
CV-PROPU-GESTION ES Objetos
ENTONCES CV-PROPU-PROYECTO ES Cascada>

REGLA TA-R101

<SI PROGRESION-FASES-DESARR ES Uniforme Y Secuencial Y
RETROALIMENTACION-FASES-DESARR ES Baja Y
COMPORTAMIENTO-PREDOMINANTE ES Batch
ENTONCES CV-PROPU-APLIC ES Cascada>

REGLA R3

<SI CV-PROPU-REQUISITOS ES Cascada Y CV-PROPU-APLIC ES Cascada Y
CV-PROPU-GESTION ES Objetos
ENTONCES CV-PROPU-PROYECTO ES Cascada>

REGLA TA-R102

<SI PROGRESION-FASES-DESARR ES Uniforme Y Secuencial Y
COMPLEJIDAD-SUBSISTEMAS ES Baja Y
COMPONENTES-PREDOMINANTES ES Algorítmico
ENTONCES CV-PROPU-APLIC ES Cascada>

REGLA R3

<SI CV-PROPU-REQUISITOS ES Cascada Y CV-PROPU-APLIC ES Cascada Y
CV-PROPU-GESTION ES Objetos
ENTONCES CV-PROPU-PROYECTO ES Cascada>

REGLA TA-R103

<SI PROGRESION-FASES-DESARR ES Uniforme Y Secuencial Y
COMPLEJIDAD-SUBSISTEMAS ES Baja Y COMPORTAMIENTO-PREDOMINANTE ES Batch
ENTONCES CV-PROPU-APLIC ES Cascada>

REGLA R3

<SI CV-PROPU-REQUISITOS ES Cascada Y CV-PROPU-APLIC ES Cascada Y
CV-PROPU-GESTION ES Objetos
ENTONCES CV-PROPU-PROYECTO ES Cascada>

CICLO DE VIDA RECOMENDADO

Se recomienda el Ciclo de Vida en Cascada

Especificación de Requerimientos recomienda:	Cascada
Tipo de Aplicación recomienda:	Cascada
Gestión de Proyecto recomienda:	Objetos

9.4.4 CASO EJEMPLO 3: Control de Morosos

Proyecto: 003 Municipal

Objetivo del proyecto: Desarrollar e implementar un sistema para el control de morosos de la tasa municipal de alumbrado, barrido y limpieza.

Caso 3 Resultados del caso de prueba:	
Ciclo de vida esperado	Ciclo de vida sugerido por el sistema experto
CV Especificación Requerimientos 1- Espiral 2- Objetos 3- Cascada	CV Especificación Requerimientos Espiral Objetos Cascada
CV Tipo de Aplicación Cascada	Tipo de Aplicación Cascada
CV Gestión de proyecto Cascada	Gestión de proyecto Cascada
CV del Proyecto Cascada	CV del Proyecto Se recomienda el Ciclo de Vida en Cascada

El sistema experto, ha respondido satisfactoriamente de acuerdo con la definición de los casos de prueba.

Caso 3 Valores del caso de prueba		
Especificación Requerimientos		
ID	Afirmación / Pregunta	Opciones
ER1	Al comienzo del proyecto los requerimientos se encuentran definidos:	Mayoritariamente
ER2	El usuario ha explicitado los requisitos:	Formalmente, Exhaustivamente, Uniformemente
ER3	La exigencia del usuario en el cumplimiento de los requerimientos en el producto final es:	Muy exigente,
Tipo de Aplicación		
ID	Afirmación / Pregunta	Opciones
TA1	El tipo de aplicación se orienta al desarrollo de :	Otros
TA2	Los componentes predominantes de la aplicación son:	Algorítmico
TA3	El comportamiento predominante en la aplicación es:	Batch
TA4	¿La progresión entre las fases del desarrollo es secuencial y uniforme?	Si
TA5	La relación entre las fases del proyecto es:	Poca dependencia
TA6	¿El tipo de aplicación requiere el desarrollo fuertemente integrado de hardware y software?	No
TA7	¿El sistema a diseñar es innovador?	No
TA8	Para lograr una mejor comprensión del sistema los niveles de descomposición en subsistemas son:	Pocos

TA9	<i>Para lograr una mejor manipulación del sistema los niveles de descomposición en subsistemas son:</i>	Pocos
TA10	<i>Se requiere detectar en etapas tempranas del proyecto problemas de:</i>	Desconoce
TA11	<i>El Diseño del sistema a desarrollar puede resolverse por:</i>	Pocas opciones
TA12	<i>La Implementación del sistema a desarrollar puede resolverse por:</i>	Pocas opciones
TA13	<i>Las modificaciones explícitamente previstas son:</i>	Redefinición

Gestión de Proyecto

ID	Afirmación / Pregunta	Opciones
GP1	<i>El desarrollo del proyecto es responsabilidad de:</i>	La organización
GP2	<i>¿El equipo de desarrollo tiene experiencia previa en el tipo de aplicación a desarrollar?</i>	Si
GP3	<i>Es factible usar metodologías de desarrollo:</i>	Ampliamente probadas
GP4	<i>Es necesario usar metodologías de desarrollo:</i>	Ampliamente probadas
GP5	<i>¿El equipo de desarrollo tiene experiencia con las técnicas de ingeniería del software que su usarán?</i>	Si
GP6	<i>El tipo de entrega que el usuario requiere de la aplicación es:</i>	Versión completa
GP7	<i>El grado de participación del usuario en el desarrollo debe ser:</i>	Medio
GP8	<i>¿Es necesario facilitar la introducción gradual del sistema para los usuarios?</i>	Si
GP9	<i>Se requiere una gestión de proyecto:</i>	Poco formal
GP10	<i>El grado de control que se requiere de la gestión del proyecto es:</i>	Medianamente ajustado
GP11	<i>¿La gestión de proyecto prevé definir explícitamente estrategias para reusar software existente?</i>	No
GP12	<i>¿Se prevé la necesidad de reutilizar los componentes del software a desarrollar en proyectos futuros?</i>	Poco necesario
GP13	<i>¿Existe un sistema previo desarrollado en objetos de forma tal que el proyecto actual es ..?</i>	Sin relación
GP14	<i>¿Existe la necesidad de reutilizar los componentes de un software ya existente desarrollado en objetos?</i>	No
GP15	<i>¿Se dispone de software para prototipar?</i>	No
GP16	<i>¿Es posible adquirir software para prototipar?</i>	No
GP17	<i>¿Existen dudas sobre la viabilidad del software?</i>	No
GP18	<i>¿Se estima la posibilidad de terminación prematura de proyectos no viables?</i>	No
GP19	<i>Se estima que el factor de riesgo en el desarrollo del sistema es:</i>	Sin riesgo
GP20	<i>¿Se estiman riesgos técnicos en el desarrollo del sistema?</i>	No
GP21	<i>¿Se identifican los riesgos asociados con cada una de las alternativas?</i>	No
GP22	<i>¿Se cuenta con métodos y técnicas para evaluar el riesgo?</i>	No
GP23	<i>¿Se identifican las diferentes maneras de resolver los riesgos?</i>	No
GP24	<i>¿Es necesario definir explícitamente estrategias para realizar análisis de riesgo?</i>	No
GP25	<i>¿Es necesario volver a etapas anteriores en el desarrollo del sistema para la resolución de algún tópico nuevo de riesgo?</i>	No
GP26	<i>¿Es posible volver a etapas anteriores en el desarrollo del sistema cuando se identifican mejores alternativas?</i>	No
GP27	<i>¿Es necesario definir mecanismos explícitos para incorporar objetivos de calidad en el desarrollo del producto?</i>	No
GP28	<i>El uso de los mismos procedimientos para desarrollo y mantenimiento del producto software está previsto como:</i>	Indiferente
GP29	<i>¿Se requiere de acuerdos confirmados para los cambios a efectuarse durante el desarrollo del proyecto?</i>	Si

SISTEMA EXPERTO PARA LA SELECCION DE CICLOS DE VIDA DE PROYECTOS DE SOFTWARE

Copyright 2001 by Bibiana Rossi

Fecha: 19/05/2001

ID: 003

Proyecto: Control de Morosos

Líder de Proyecto: Lic. Enrique Fernández

Objetivo: Desarrollar un sistema para control de morosos de la Tasa Municipal de ABL

Fecha de inicio: 9/02/1999

Fecha de finalización: 12/06/1999

Ciclo de Vida recomendado: Cascada

Ciclo de Vida seleccionado: Cascada

REGLA ER-R1

<SI TIPO-DEFINICION ES Formalmente Y Exhaustivamente Y Uniformemente
ENTONCES DEFINICION-REQUISITOS ES Clara Y DEFINICION-LIMITES ES Clara>

REGLA ER-R2

<SI DEFINICION-REQUISITOS ES Clara Y DEFINICION-LIMITES ES Clara Y
GRADO-DE-CERTIDUMBRE-INICIO ES Mayoritariamente
ENTONCES CV-PROPUESTO-REQUISITOS ES Cascada>

REGLA ER-R5

<SI GRADO-CUMPLIMIENTO-PRODUCTO-FINAL ES Usuario muy exigente
ENTONCES CV-PROPUESTO-REQUISITOS ES Espiral Y Objetos>

REGLA CV-R2

<SI RELACION-FASES-DESARROLLO ES Poca dependencia
ENTONCES RETROALIMENTACION-FASES-DESARROLLO ES Baja>

REGLA CV-R4

<SI PROGRE-FASES-DESARR ES Uniforme Y Secuencial Y
RETROALIM-FASES-DESARR ES Baja Y COMP-PREDOMINANTES ES Algorítmico
ENTONCES CV-PROP-APLICACION ES Cascada>

REGLA TA-R1

<SI NIVELES-COMPOSICION ES Pocos subsistemas
ENTONCES COMPLEJIDAD-SUBSISTEMAS ES Baja>

REGLA TA-R11

<SI TIPO-MODIFICACION ES Redefinición
ENTONCES VARIABILIDAD-PROCESOS ES Alta>

REGLA TA-R16

<SI VARIABILIDAD-PROCESOS ES Alta
ENTONCES MODELADO-PROTOTIPO ES Conveniente>

REGLA GP-R1

<SI NECESIDAD-METOD ES Ampliamente probada Y
FACTIBILIDAD-METOD ES Ampliamente probada
ENTONCES CONVENIENCIA-METOD ES Ampliamente probada>

REGLA GP-R3

<SI SOFT-PROTOTIPO ES No disponible Y No adquirible
ENTONCES FACTIBILIDAD-PROTOTIPO ES No factible>

REGLA GP-R4

<SI NIVEL-RIESGO ES No hay
ENTONCES RIESGO-CASCADA ES Aceptable>

REGLA GP-R5

<SI CONVENIENCIA-METOD ES Ampliamente probada Y
ENTREGAS ES Versión completa Y RIESGO-CASCADA ES Aceptable
ENTONCES CV-PROPU-GESTION ES Cascada>

REGLA GP-R16

<SI FACTIB-PROTOTIPO ES No factible Y RIESGO-CASCADA ES Aceptable
ENTONCES CV-PROPU-GESTION ES Cascada>

REGLA GP-R19

<SI NIVEL-RIESGO ES No hay
ENTONCES RIESGO-OO ES Aceptable>

REGLA GP-R22

<SI INTRODUCCION-GRADUAL ES Necesaria
ENTONCES APPLICABILIDAD-PROTOTIPO ES Aplicable>

REGLA GP-R33

<SI PROC-CAMBIOS ES Acuerdos confirmados
ENTONCES APPLICABILIDAD-ESPIRAL ES Aplicable>

REGLA GP-R34

<SI RESPONSABILIDAD ES Organización propia
ENTONCES APPLICABILIDAD-ESPIRAL ES Aplicable>

REGLA R0

<SI CV-PROPU-REQUISITOS ES Cascada Y CV-PROPU-APLIC ES Cascada Y
CV-PROPU-GESTION ES Cascada
ENTONCES CV-PROPU-PROYECTO ES Cascada>

REGLA ER-R2

<SI DEFINICION-REQUISITOS ES Clara Y DEFINICION-LIMITES ES Clara Y
GRADO-DE-CERTIDUMBRE-INICIO ES Mayoritariamente
ENTONCES CV-PROPUESTO-REQUISITOS ES Cascada>

REGLA R0

<SI CV-PROPU-REQUISITOS ES Cascada Y CV-PROPU-APLIC ES Cascada Y
CV-PROPU-GESTION ES Cascada
ENTONCES CV-PROPU-PROYECTO ES Cascada>

REGLA R6

<SI CV-PROPU-REQUISITOS ES Objetos Y CV-PROPU-APLIC ES Cascada Y
CV-PROPU-GESTION ES Cascada
ENTONCES CV-PROPU-PROYECTO ES Cascada>

REGLA R10

<SI CV-PROPU-REQUISITOS ES Espiral Y CV-PROPU-APLIC ES Cascada Y
CV-PROPU-GESTION ES Cascada
ENTONCES CV-PROPU-PROYECTO ES Cascada>

REGLA TA-R101

<SI PROGRESION-FASES-DESARR ES Uniforme Y Secuencial Y
RETROALIMENTACION-FASES-DESARR ES Baja Y
COMPORTAMIENTO-PREDOMINANTE ES Batch
ENTONCES CV-PROPU-APLIC ES Cascada>

REGLA R0

<SI CV-PROPU-REQUISITOS ES Cascada Y CV-PROPU-APLIC ES Cascada Y
CV-PROPU-GESTION ES Cascada
ENTONCES CV-PROPU-PROYECTO ES Cascada>

REGLA R6

<SI CV-PROPU-REQUISITOS ES Objetos Y CV-PROPU-APLIC ES Cascada Y

CV-PROPU-GESTION ES Cascada
 ENTONCES CV-PROPU-PROYECTO ES Cascada>
 REGLA R10
 <SI CV-PROPU-REQUISITOS ES Espiral Y CV-PROPU-APLIC ES Cascada Y
 CV-PROPU-GESTION ES Cascada
 ENTONCES CV-PROPU-PROYECTO ES Cascada>
 REGLA TA-R102
 <SI PROGRESION-FASES-DESARR ES Uniforme Y Secuencial Y
 COMPLEJIDAD-SUBSISTEMAS ES Baja Y
 COMPONENTES-PREDOMINANTES ES Algorítmico
 ENTONCES CV-PROPU-APLIC ES Cascada>
 REGLA R0
 <SI CV-PROPU-REQUISITOS ES Cascada Y CV-PROPU-APLIC ES Cascada Y
 CV-PROPU-GESTION ES Cascada
 ENTONCES CV-PROPU-PROYECTO ES Cascada>
 REGLA R6
 <SI CV-PROPU-REQUISITOS ES Objetos Y CV-PROPU-APLIC ES Cascada Y
 CV-PROPU-GESTION ES Cascada
 ENTONCES CV-PROPU-PROYECTO ES Cascada>
 REGLA R10
 <SI CV-PROPU-REQUISITOS ES Espiral Y CV-PROPU-APLIC ES Cascada Y
 CV-PROPU-GESTION ES Cascada
 ENTONCES CV-PROPU-PROYECTO ES Cascada>
 REGLA TA-R103
 <SI PROGRESION-FASES-DESARR ES Uniforme Y Secuencial Y
 COMPLEJIDAD-SUBSISTEMAS ES Baja Y COMPORTAMIENTO-PREDOMINANTE ES Batch
 ENTONCES CV-PROPU-APLIC ES Cascada>
 REGLA R0
 <SI CV-PROPU-REQUISITOS ES Cascada Y CV-PROPU-APLIC ES Cascada Y
 CV-PROPU-GESTION ES Cascada
 ENTONCES CV-PROPU-PROYECTO ES Cascada>
 REGLA R6
 <SI CV-PROPU-REQUISITOS ES Objetos Y CV-PROPU-APLIC ES Cascada Y
 CV-PROPU-GESTION ES Cascada
 ENTONCES CV-PROPU-PROYECTO ES Cascada>
 REGLA R10
 <SI CV-PROPU-REQUISITOS ES Espiral Y CV-PROPU-APLIC ES Cascada Y
 CV-PROPU-GESTION ES Cascada
 ENTONCES CV-PROPU-PROYECTO ES Cascada>

CICLO DE VIDA RECOMENDADO

Se recomienda el Ciclo de Vida en Cascada

Especificación de Requerimientos recomienda:	Espiral Objetos Cascada
Tipo de Aplicación recomienda:	Cascada
Gestión de Proyecto recomienda:	Cascada

9.4.5 CASO EJEMPLO 4: Técnica Emparrillado

Proyecto: 004 Emparrillado.

Objetivo del proyecto: Desarrollar e implementar una aplicación para el calculo automático de la técnica de emparrillado. El sistema debe dibujar los gráficos resultantes del calculo.

Caso 4 Resultados del caso de prueba:	
Ciclo de vida esperado	Ciclo de vida sugerido por el sistema experto
CV Especificación Requerimientos 1- Espiral 2- Objetos	CV Especificación Requerimientos Espiral Objetos
CV Tipo de Aplicación Objetos	Tipo de Aplicación Objetos
CV Gestión de proyecto Objetos	Gestión de proyecto Objetos
CV del Proyecto Objetos	CV del Proyecto Se recomienda el Ciclo de Vida de Objetos

El sistema experto, ha respondido satisfactoriamente de acuerdo con la definición de los casos de prueba.

Caso 4 Resultados del caso de prueba:		
Especificación Requerimientos		
ID	Afirmación / Pregunta	Opciones
ER1	<i>Al comienzo del proyecto los requerimientos se encuentran definidos:</i>	No
ER2	<i>El usuario ha explicitado los requisitos:</i>	Informalmente, Incompletos Desestructuradamente
ER3	<i>La exigencia del usuario en el cumplimiento de los requerimientos en el producto final es:</i>	Muy exigente,
Tipo de Aplicación		
ID	Afirmación / Pregunta	Opciones
TA1	<i>El tipo de aplicación se orienta al desarrollo de :</i>	Software de base
TA2	<i>Los componentes predominantes de la aplicación son:</i>	Gráfico – Matemático
TA3	<i>El comportamiento predominante en la aplicación es:</i>	Interactivo – Dinámico
TA4	<i>¿La progresión entre las fases del desarrollo es secuencial y uniforme?</i>	No

TA5	<i>La relación entre las fases del proyecto es:</i>	Mucha dependencia
TA6	<i>¿El tipo de aplicación requiere el desarrollo fuertemente integrado de hardware y software?</i>	No
TA7	<i>¿El sistema a diseñar es innovador?</i>	Si
TA8	<i>Para lograr una mejor comprensión del sistema los niveles de descomposición en subsistemas son:</i>	Pocos
TA9	<i>Para lograr una mejor manipulación del sistema los niveles de descomposición en subsistemas son:</i>	Pocos
TA10	<i>Se requiere detectar en etapas tempranas del proyecto problemas de:</i>	Eficacia
TA11	<i>El Diseño del sistema a desarrollar puede resolverse por:</i>	Pocas opciones
TA12	<i>La Implementación del sistema a desarrollar puede resolverse por:</i>	Pocas opciones
TA13	<i>Las modificaciones explícitamente previstas son:</i>	Redefinición - Ampliación

Gestión de Proyecto

ID	Afirmación / Pregunta	Opciones
GP1	<i>El desarrollo del proyecto es responsabilidad de:</i>	La organización
GP2	<i>¿El equipo de desarrollo tiene experiencia previa en el tipo de aplicación a desarrollar?</i>	No
GP3	<i>Es factible usar metodologías de desarrollo:</i>	Medianamente probadas
GP4	<i>Es necesario usar metodologías de desarrollo:</i>	Medianamente probadas
GP5	<i>¿El equipo de desarrollo tiene experiencia con las técnicas de ingeniería del software que su usarán?</i>	No
GP6	<i>El tipo de entrega que el usuario requiere de la aplicación es:</i>	Versión temprana
GP7	<i>El grado de participación del usuario en el desarrollo debe ser:</i>	Medio
GP8	<i>¿Es necesario facilitar la introducción gradual del sistema para los usuarios?</i>	No
GP9	<i>Se requiere una gestión de proyecto:</i>	Poco formal
GP10	<i>El grado de control que se requiere de la gestión del proyecto es:</i>	Poco ajustado
GP11	<i>¿La gestión de proyecto prevé definir explícitamente estrategias para reusar software existente?</i>	Si
GP12	<i>¿Se prevé la necesidad de reutilizar los componentes del software a desarrollar en proyectos futuros?</i>	Muy necesario
GP13	<i>¿Existe un sistema previo desarrollado en objetos de forma tal que el proyecto actual es ...?</i>	Sin relación
GP14	<i>¿Existe la necesidad de reutilizar los componentes de un software ya existente desarrollado en objetos?</i>	Desconoce
GP15	<i>¿Se dispone de software para prototipar?</i>	Si
GP16	<i>¿Es posible adquirir software para prototipar?</i>	Si
GP17	<i>¿Existen dudas sobre la viabilidad del software?</i>	No
GP18	<i>¿Se estima la posibilidad de terminación prematura de proyectos no viables?</i>	No
GP19	<i>Se estima que el factor de riesgo en el desarrollo del sistema es:</i>	Bajo
GP20	<i>¿Se estiman riesgos técnicos en el desarrollo del sistema?</i>	No
GP21	<i>¿Se identifican los riesgos asociados con cada una de las alternativas?</i>	No
GP22	<i>¿Se cuenta con métodos y técnicas para evaluar el riesgo?</i>	Desconoce
GP23	<i>¿Se identifican las diferentes maneras de resolver los riesgos?</i>	Desconoce

GP24	<i>¿Es necesario definir explícitamente estrategias para realizar análisis de riesgo?</i>	No
GP25	<i>¿Es necesario volver a etapas anteriores en el desarrollo del sistema para la resolución de algún tópico nuevo de riesgo?</i>	No
GP26	<i>¿Es posible volver a etapas anteriores en el desarrollo del sistema cuando se identifican mejores alternativas?</i>	Desconoce
GP27	<i>¿Es necesario definir mecanismos explícitos para incorporar objetivos de calidad en el desarrollo del producto?</i>	Si
GP28	<i>El uso de los mismos procedimientos para desarrollo y mantenimiento del producto software está previsto como:</i>	Necesario – Factible
GP29	<i>¿Se requiere de acuerdos confirmados para los cambios a efectuarse durante el desarrollo del proyecto?</i>	No

SISTEMA EXPERTO PARA LA SELECCION DE CICLOS DE VIDA DE PROYECTOS DE SOFTWARE
 Copyright 2001 by Bibiana Rossi

Fecha: 19/05/2001
 ID: 004
 Proyecto: Técnica Emparrillado
 Líder de Proyecto: Lic. Enrique Fernández
 Objetivo: Desarrollar aplicación para el calculo automático y gráficos de la técnica de emparrillado.
 Fecha de inicio: 22/02/2000
 Fecha de finalización: 15/09/2000
 Ciclo de Vida recomendado: Objetos
 Ciclo de Vida seleccionado: Objetos

REGLA ER-R3
 <SI TIPO-DEFINICION ES Informalmente Y Incompleto Y Desestructuradamente
 ENTONCES DEFINICION-REQUISITOS ES Incierta Y DEFINICION-LIMITES ES Incierta>

REGLA ER-R4
 <SI DEFINICION-REQUISITOS ES Incierta Y DEFINICION-LIMITES ES Incierta Y
 GRADO-DE-CERTIDUMBRE-INICIO ES Parcialmente
 ENTONCES CV-PROPUESTO-REQUISITOS ES Espiral Y Objetos>

REGLA ER-R5
 <SI GRADO-CUMPLIMIENTO-PRODUCTO-FINAL ES Usuario muy exigente
 ENTONCES CV-PROPUESTO-REQUISITOS ES Espiral Y Objetos>

REGLA CV-R3
 <SI RELACION-FASES-DESARROLLO ES Mucha dependencia
 ENTONCES RETROALIMENTACION-FASES-DESARROLLO ES Alta>

REGLA CV-R5
 <SI PROGRE-FASES-DESARR ES No uniforme Y Secuencial
 ENTONCES MODELADO-PROTOTIPO ES Conveniente>

REGLA TA-R1

<SI NIVELES-COMPOSICION ES Pocos subsistemas
 ENTONCES COMPLEJIDAD-SUBSISTEMAS ES Baja>

REGLA TA-R4

<SI COMPONENTES-PREDOMINANTES ES Matemáticos
 ENTONCES MODELADO-OBJETOS ES Conveniente>

REGLA TA-R5

<SI COMPONENTES-PREDOMINANTES ES Gráficos
 ENTONCES MODELADO-OBJETOS ES Conveniente>

REGLA TA-R7

<SI COMPORTAMIENTO-PREDOMINANTE ES Interactivo
 ENTONCES MODELADO-OBJETOS ES Conveniente>

REGLA TA-R9

<SI ORIENTACION ES Software de base
 ENTONCES MODELADO OBJETOS ES Conveniente>

REGLA TA-R11

<SI TIPO-MODIFICACION ES Redefinición
 ENTONCES VARIABILIDAD-PROCESOS ES Alta>

REGLA TA-R13

<SI TIPO-MODIFICACION ES Ampliación
 ENTONCES VARIABILIDAD-PROCESOS ES Alta>

REGLA TA-R15

<SI FACTORES-DISEÑO ES Problemas de eficacia
 ENTONCES MODELADO-PROTOTIPO ES Conveniente>

REGLA TA-R16

<SI VARIABILIDAD-PROCESOS ES Alta
 ENTONCES MODELADO-PROTOTIPO ES Conveniente>

REGLA TA-R17

<SI EXISTENCIA-APLICACIONES ES Innovador
 ENTONCES MODELADO PROTOTIPO ES Conveniente>

REGLA TA-R20

<SI MODELADO-OBJETOS ES Conveniente Y MODELADO-PROTOTIPO ES Conveniente
 ENTONCES CV-PROPU-APLICACION ES Objetos>

REGLA GP-R2

<SI NIVEL-RIESGO ES Bajo
 ENTONCES RIESGO-CASCADA ES Aceptable>

REGLA GP-R11

<SI REUSO-APLIC-FUTURA ES Muy necesario
 ENTONCES SISTEMA-OO ES Existe>

REGLA GP-R12

<SI NIVEL-RIESGO ES Bajo

ENTONCES RIESGO-OO ES Aceptable>

REGLA GP-R17

<SI FORMALIDAD ES Poco Y CONVENIENCIA-METOD ES Medianamente probada
 ENTONCES APLICABILIDAD-OO ES Aplicable>

REGLA GP-R18

<SI APLICABILIDAD-OO ES Aplicable Y SISTEMA-OO ES Existe Y RIESGO-OO ES Aceptable
 ENTONCES CV-PROPU-GESTION ES Objetos>

REGLA GP-R20

<SI SOFT-PROTOTIPO ES Factible
 ENTONCES FACTIBILIDAD-PROTOTIPO ES Factible>

REGLA GP-R21

<SI SOFT-PROTOTIPO ES Disponible
 ENTONCES FACTIBILIDAD-PROTOTIPO ES Factible>

REGLA GP-R24

<SI ENTREGAS ES Versión temprana
 ENTONCES ENTREGA-INTERMEDIA ES Existe>

REGLA GP-R26

<SI REUSO-APLIC-EXISTENTE ES Estrategias explicitas
 ENTONCES APLICABILIDAD-OO ES Aplicable>

REGLA GP-R18

<SI APLICABILIDAD-OO ES Aplicable Y SISTEMA-OO ES Existe Y RIESGO-OO ES Aceptable
 ENTONCES CV-PROPU-GESTION ES Objetos>

REGLA GP-R27

<SI EXPERIENCIA-PREVIA ES No hay
 ENTONCES APLICABILIDAD-PROTOTIPO ES Aplicable>

REGLA GP-R29

<SI ENTREGA-INTERMEDIA ES Existe
 ENTONCES APLICABILIDAD-OO ES Aplicable Y APLICABILIDAD-PROTOTIPO ES Aplicable>

REGLA GP-R30

<SI EXPERIENCIA-TECNICAS-IS ES No hay
 ENTONCES APLICABILIDAD-PROTOTIPO ES Aplicable>

REGLA GP-R31

<SI PROC-DES-MANTE ES Necesario Y Factible
 ENTONCES PROCEDIMIENTOS-DM ES Conveniente>

REGLA GP-R32

<SI PROCEDIMIENTOS-DM ES Conveniente
 ENTONCES APLICABILIDAD-ESPIRAL ES Aplicable>

REGLA GP-R34

<SI RESPONSABILIDAD ES Organización propia
 ENTONCES APLICABILIDAD-ESPIRAL ES Aplicable>

REGLA GP-R37

<SI NECESIDAD-METOD ES Medianamente probada Y
 FACTIB-METOD ES Medianamente probada
 ENTONCES CONVENIENCIA-METOD ES Medianamente probada>

REGLA GP-R17

<SI FORMALIDAD ES Poco Y CONVENIENCIA-METOD ES Medianamente probada
 ENTONCES APPLICABILIDAD-OO ES Aplicable>

REGLA GP-R18

<SI APPLICABILIDAD-OO ES Aplicable Y SISTEMA-OO ES Existe Y RIESGO-OO ES Aceptable
 ENTONCES CV-PROPU-GESTION ES Objetos>

REGLA R1

<SI CV-PROPU-REQUISITOS ES Objetos Y CV-PROPU-APLIC ES Objetos Y
 CV-PROPU-GESTION ES Objetos
 ENTONCES CV-PROPU-PROYECTO ES Objetos>

REGLA ER-R4

<SI DEFINICION-REQUISITOS ES Incierta Y DEFINICION-LIMITES ES Incierta Y
 GRADO-DE-CERTIDUMBRE-INICIO ES Parcialmente
 ENTONCES CV-PROPUESTO-REQUISITOS ES Espiral Y Objetos>

REGLA R1

<SI CV-PROPU-REQUISITOS ES Objetos Y CV-PROPU-APLIC ES Objetos Y
 CV-PROPU-GESTION ES Objetos
 ENTONCES CV-PROPU-PROYECTO ES Objetos>

REGLA R18

<SI CV-PROPU-REQUISITOS ES Espiral Y CV-PROPU-APLIC ES Objetos Y
 CV-PROPU-GESTION ES Objetos
 ENTONCES CV-PROPU-PROYECTO ES Objetos>

CICLO DE VIDA RECOMENDADO

Se recomienda el Ciclo de Vida de Objetos

Especificación de Requerimientos recomienda: Espiral Objetos

Tipo de Aplicación recomienda: Objetos

Gestión de Proyecto recomienda: Objetos

9.4.6 CASO EJEMPLO 5: Inscripción Universitaria

Proyecto: 005 Inscripción Sistemas

Objetivo del proyecto: Desarrollar e implementar un sistema que permita la inscripción en las materias a cursar para alumnos de una carrera universitaria.

Caso 5 Resultados del caso de prueba:	
Ciclo de vida esperado	Ciclo de vida sugerido por el sistema experto
CV Especificación Requerimientos 1-Espiral 2- Objetos	CV Especificación Requerimientos Espiral Objetos
CV Tipo de Aplicación 1-Espiral 2- Objetos	Tipo de Aplicación Espiral Objetos
CV Gestión de proyecto Espiral	Gestión de proyecto Espiral
CV del Proyecto Espiral	CV del Proyecto Se recomienda el Ciclo de Vida en Espiral

El sistema experto, ha respondido satisfactoriamente de acuerdo con la definición de los casos de prueba.

Caso 5 Valores del caso de prueba		
Especificación Requerimientos		
ID	Afirmación / Pregunta	Opciones
ER1	Al comienzo del proyecto los requerimientos se encuentran definidos:	Mayoritariamente
ER2	El usuario ha explicitado los requisitos:	Informalmente, Incompletos Desestructuradamente
ER3	La exigencia del usuario en el cumplimiento de los requerimientos en el producto final es:	Muy exigente
Tipo de Aplicación		
ID	Afirmación / Pregunta	Opciones
TA1	El tipo de aplicación se orienta al desarrollo de :	Otros
TA2	Los componentes predominantes de la aplicación son:	Algorítmico
TA3	El comportamiento predominante en la aplicación es:	Interactivo
TA4	¿La progresión entre las fases del desarrollo es secuencial y uniforme?	No
TA5	La relación entre las fases del proyecto es:	Mucha dependencia
TA6	¿El tipo de aplicación requiere el desarrollo fuertemente integrado de hardware y software?	Si

TA7	<i>¿El sistema a diseñar es innovador?</i>	No
TA8	<i>Para lograr una mejor comprensión del sistema los niveles de descomposición en subsistemas son:</i>	Muchos
TA9	<i>Para lograr una mejor manipulación del sistema los niveles de descomposición en subsistemas son:</i>	Muchos
TA10	<i>Se requiere detectar en etapas tempranas del proyecto problemas de:</i>	Arquitectura – Eficacia
TA11	<i>El Diseño del sistema a desarrollar puede resolverse por:</i>	Varias opciones
TA12	<i>La Implementación del sistema a desarrollar puede resolverse por:</i>	Varias opciones
TA13	<i>Las modificaciones explícitamente previstas son:</i>	Redefinición

Gestión de Proyecto

ID	Afirmación / Pregunta	Opciones
GP1	<i>El desarrollo del proyecto es responsabilidad de:</i>	La organización
GP2	<i>¿El equipo de desarrollo tiene experiencia previa en el tipo de aplicación a desarrollar?</i>	No
GP3	<i>Es factible usar metodologías de desarrollo:</i>	Medianamente probadas
GP4	<i>Es necesario usar metodologías de desarrollo:</i>	Medianamente probadas
GP5	<i>¿El equipo de desarrollo tiene experiencia con las técnicas de ingeniería del software que su usarán?</i>	No
GP6	<i>El tipo de entrega que el usuario requiere de la aplicación es:</i>	Versión completa
GP7	<i>El grado de participación del usuario en el desarrollo debe ser:</i>	Alto
GP8	<i>¿Es necesario facilitar la introducción gradual del sistema para los usuarios?</i>	Si
GP9	<i>Se requiere una gestión de proyecto:</i>	Medianamente formal
GP10	<i>El grado de control que se requiere de la gestión del proyecto es:</i>	Medianamente ajustado
GP11	<i>¿La gestión de proyecto prevé definir explícitamente estrategias para reusar software existente?</i>	No
GP12	<i>¿Se prevé la necesidad de reutilizar los componentes del software a desarrollar en proyectos futuros?</i>	Muy necesario
GP13	<i>¿Existe un sistema previo desarrollado en objetos de forma tal que el proyecto actual es ..?</i>	Sin relación
GP14	<i>¿Existe la necesidad de reutilizar los componentes de un software ya existente desarrollado en objetos?</i>	No
GP15	<i>¿Se dispone de software para prototipar?</i>	No
GP16	<i>¿Es posible adquirir software para prototipar?</i>	Si
GP17	<i>¿Existen dudas sobre la viabilidad del software?</i>	No
GP18	<i>¿Se estima la posibilidad de terminación prematura de proyectos no viables?</i>	Desconoce
GP19	<i>Se estima que el factor de riesgo en el desarrollo del sistema es:</i>	Alto
GP20	<i>¿Se estiman riesgos técnicos en el desarrollo del sistema?</i>	Si
GP21	<i>¿Se identifican los riesgos asociados con cada una de las alternativas?</i>	Si
GP22	<i>¿Se cuenta con métodos y técnicas para evaluar el riesgo?</i>	Si
GP23	<i>¿Se identifican las diferentes maneras de resolver los riesgos?</i>	Si
GP24	<i>¿Es necesario definir explícitamente estrategias para realizar análisis de riesgo?</i>	Si
GP25	<i>¿Es necesario volver a etapas anteriores en el desarrollo del sistema para la resolución de algún tópico nuevo de riesgo?</i>	Desconoce

GP26	<i>¿Es posible volver a etapas anteriores en el desarrollo del sistema cuando se identifican mejores alternativas?</i>	Si
GP27	<i>¿Es necesario definir mecanismos explícitos para incorporar objetivos de calidad en el desarrollo del producto?</i>	Si
GP28	<i>El uso de los mismos procedimientos para desarrollo y mantenimiento del producto software está previsto como:</i>	Necesario – Factible
GP29	<i>¿Se requiere de acuerdos confirmados para los cambios a efectuarse durante el desarrollo del proyecto?</i>	Si

<p>SISTEMA EXPERTO PARA LA SELECCION DE CICLOS DE VIDA DE PROYECTOS DE SOFTWARE</p> <p>Copyright 2001 by Bibiana Rossi</p> <hr/> <p>Fecha: 19/05/2001</p> <p>ID: 005</p> <p>Proyecto: Inscripción Universitaria</p> <p>Líder de Proyecto: Ing. María Florencia Pollo Cattaneo</p> <p>Objetivo: Desarrollar un sistema que permite la inscripción en materias para alumnos universitarios.</p> <p>Fecha de inicio: 01/08/1998</p> <p>Fecha de finalización: 15/02/2000</p> <p>Ciclo de Vida recomendado: Espiral</p> <p>Ciclo de Vida seleccionado: Espiral</p> <hr/> <p>REGLA ER-R3</p> <p><SI TIPO-DEFINICION ES Informalmente Y Incompleto Y Desestructuradamente ENTONCES DEFINICION-REQUISITOS ES Incierta Y DEFINICION-LIMITES ES Incierta></p> <p>REGLA ER-R5</p> <p><SI GRADO-CUMPLIMIENTO-PRODUCTO-FINAL ES Usuario muy exigente ENTONCES CV-PROPUESTO-REQUISITOS ES Espiral Y Objetos></p> <p>REGLA CV-R3</p> <p><SI RELACION-FASES-DESARROLLO ES Mucha dependencia ENTONCES RETROALIMENTACION-FASES-DESARROLLO ES Alta></p> <p>REGLA CV-R5</p> <p><SI PROGRE-FASES-DESARR ES No uniforme Y Secuencial ENTONCES MODELADO-PROTOTIPO ES Convenientе></p> <p>REGLA TA-R7</p> <p><SI COMPORTAMIENTO-PREDOMINANTE ES Interactivo ENTONCES MODELAO-OBJETOS ES Convenientе></p> <p>REGLA TA-R11</p> <p><SI TIPO-MODIFICACION ES Redefinición ENTONCES VARIABILIDAD-PROCESOS ES Alta></p> <p>REGLA TA-R14</p>	
--	--

<SI FACTORES-DISEÑO ES Problemas de arquitectura
ENTONCES MODELADO-PROTOTIPO ES Conveniente>

REGLA TA-R15

<SI FACTORES-DISEÑO ES Problemas de eficacia
ENTONCES MODELADO-PROTOTIPO ES Conveniente>

REGLA TA-R16

<SI VARIABILIDAD-PROCESOS ES Alta
ENTONCES MODELADO-PROTOTIPO ES Conveniente>

REGLA TA-R20

<SI MODELADO-OBJETOS ES Conveniente Y MODELADO-PROTOTIPO ES Conveniente
ENTONCES CV-PROPU-APLICACION ES Objetos>

REGLA TA-R21

<SI MODELADO-OBJETOS ES Conveniente Y MODELADO-PROTOTIPO ES Conveniente Y
INTREGRACION-HW-SW ES Fuertemente
ENTONCES CV-PROPU-APLICACION ES Espiral>

REGLA TA-R22

<SI POSIBILIDAD-DISEÑO ES Varias opciones
ENTONCES OPCIONALIDAD ES Múltiple>

REGLA TA-R23

<SI POSIBILIDAD-IMPLEMENTACION ES Varias opciones
ENTONCES OPCIONALIDAD ES Múltiple>

REGLA TA-R24

<SI MODELADO-OBJETOS ES Conveniente Y MODELADO-PROTOTIPO ES Conveniente Y
OPCIONALIDAD ES Múltiple
ENTONCES CV-PROPU-APLICACION ES Espiral>

REGLA GP-R11

<SI REUSO-APLIC-FUTURA ES Muy necesario
ENTONCES SISTEMA-OO ES Existe>

REGLA GP-R15

<SI FORMALIDAD ES Medianamente Y CONVENIENCIA-METOD ES Medianamente probada
ENTONCES APPLICABILIDAD-OO ES Aplicable>

REGLA GP-R20

<SI SOFT-PROTOTIPO ES Factible
ENTONCES FACTIBILIDAD-PROTOTIPO ES Factible>

REGLA GP-R22

<SI INTRODUCCION-GRADUAL ES Necesaria
ENTONCES APPLICABILIDAD-PROTOTIPO ES Aplicable>

REGLA GP-R23

<SI PARTICIPACION-USUARIO ES Fuerte
ENTONCES APPLICABILIDAD-PROTOTIPO ES Aplicable>

REGLA GP-R27

<SI EXPERIENCIA-PREVIA ES No hay

ENTONCES APLICABILIDAD-PROTOTIPO ES Aplicable>

REGLA GP-R30

<SI EXPERIENCIA-TECNICAS-IS ES No hay

ENTONCES APLICABILIDAD-PROTOTIPO ES Aplicable>

REGLA GP-R31

<SI PROC-DES-MANTE ES Necesario Y Factible

ENTONCES PROCEDIMIENTOS-DM ES Conveniente>

REGLA GP-R32

<SI PROCEDIMIENTOS-DM ES Conveniente

ENTONCES APLICABILIDAD-ESPIRAL ES Aplicable>

REGLA GP-R33

<SI PROC-CAMBIOS ES Acuerdos confirmados

ENTONCES APLICABILIDAD-ESPIRAL ES Aplicable>

REGLA GP-R34

<SI RESPONSABILIDAD ES Organización propia

ENTONCES APLICABILIDAD-ESPIRAL ES Aplicable>

REGLA GP-R36

<SI CATEGORIA-RIESGO ES Técnico

ENTONCES FACTOR-RIESGO ES Existe>

REGLA GP-R37

<SI NECESIDAD-METOD ES Medianamente probada Y

FACTIB-METOD ES Medianamente probada

ENTONCES CONVENIENCIA-METOD ES Medianamente probada>

REGLA GP-R15

<SI FORMALIDAD ES Medianamente Y CONVENIENCIA-METOD ES Medianamente probada

ENTONCES APLICABILIDAD-OO ES Aplicable>

REGLA GP-R38

<SI RIESGO-ALTERNATIVAS ES Se identifican Y

IDENTIFICACION-ALTERNATIVAS ES Se identifican

ENTONCES HABILIDAD-RIESGO ES Existe>

REGLA GP-R39

<SI HABILIDAD-RIESGO ES Existe Y IDENTIFICACION-ALTERNATIVAS ES Etapas anteriores

ENTONCES APLICABILIDAD-ESPIRAL ES Aplicable>

REGLA GP-R40

<FORMULACION EXTERNA>

REGLA GP-R43

<SI FACTOR-RIESGO ES Existe Y ANALISIS-RIESGO ES Estrategias explicitas

ENTONCES APLICABILIDAD-ESPIRAL ES Aplicable>

REGLA GP-R44

<SI FACTOR-RIESGO ES Existe Y OBJETIVO-CALIDAD ES Mecanismos explícitos
ENTONCES APLICABILIDAD-ESPIRAL ES Aplicable>

REGLA GP-R45

<SI APICAB-PROTOTIPO ES Aplicable Y FACTIB-PROTOTIPO ES Factible Y
RIESGO ES Evaluable Y APICAB-ESPIRAL ES Aplicable
ENTONCES CV-PROPU-GESTION ES Espiral>

REGLA R2

<SI CV-PROPU-REQUISITOS ES Espiral Y CV-PROPU-APLIC ES Espiral Y
CV-PROPU-GESTION ES Espiral

ENTONCES CV-PROPU-PROYECTO ES Espiral>

REGLA R2

<SI CV-PROPU-REQUISITOS ES Espiral Y CV-PROPU-APLIC ES Espiral Y
CV-PROPU-GESTION ES Espiral

ENTONCES CV-PROPU-PROYECTO ES Espiral>

REGLA R15

<SI CV-PROPU-REQUISITOS ES Objetos Y CV-PROPU-APLIC ES Objetos Y
CV-PROPU-GESTION ES Espiral

ENTONCES CV-PROPU-PROYECTO ES Objetos>

REGLA R24

<SI CV-PROPU-REQUISITOS ES Espiral Y CV-PROPU-APLIC ES Objetos Y
CV-PROPU-GESTION ES Espiral

ENTONCES CV-PROPU-PROYECTO ES Espiral>

REGLA R26

<SI CV-PROPU-REQUISITOS ES Objetos Y CV-PROPU-APLIC ES Espiral Y
CV-PROPU-GESTION ES Espiral

ENTONCES CV-PROPU-PROYECTO ES Espiral>

REGLA GP-R52

<SI RIESGO-ESPIRAL ES Aceptable

ENTONCES FACTOR-RIESGO ES Existe>

REGLA GP-R43

<SI FACTOR-RIESGO ES Existe Y ANALISIS-RIESGO ES Estrategias explicitas
ENTONCES APLICABILIDAD-ESPIRAL ES Aplicable>

REGLA GP-R45

<SI APICAB-PROTOTIPO ES Aplicable Y FACTIB-PROTOTIPO ES Factible Y
RIESGO ES Evaluable Y APICAB-ESPIRAL ES Aplicable
ENTONCES CV-PROPU-GESTION ES Espiral>

REGLA R2

<SI CV-PROPU-REQUISITOS ES Espiral Y CV-PROPU-APLIC ES Espiral Y
CV-PROPU-GESTION ES Espiral

ENTONCES CV-PROPU-PROYECTO ES Espiral>

REGLA R15

<SI CV-PROPU-REQUISITOS ES Objetos Y CV-PROPU-APLIC ES Objetos Y
CV-PROPU-GESTION ES Espiral

ENTONCES CV-PROPU-PROYECTO ES Objetos>

REGLA R24

<SI CV-PROPU-REQUISITOS ES Espiral Y CV-PROPU-APLIC ES Objetos Y
CV-PROPU-GESTION ES Espiral

ENTONCES CV-PROPU-PROYECTO ES Espiral>

REGLA R26

<SI CV-PROPU-REQUISITOS ES Objetos Y CV-PROPU-APLIC ES Espiral Y
CV-PROPU-GESTION ES Espiral

ENTONCES CV-PROPU-PROYECTO ES Espiral>

REGLA GP-R44

<SI FACTOR-RIESGO ES Existe Y OBJETIVO-CALIDAD ES Mecanismos explícitos
ENTONCES APPLICABILIDAD-ESPIRAL ES Aplicable>

REGLA GP-R45

<SI APPLICAB-PROTOTIPO ES Aplicable Y FACTIB-PROTOTIPO ES Factible Y
RIESGO ES Evaluable Y APPLICAB-ESPIRAL ES Aplicable

ENTONCES CV-PROPU-GESTION ES Espiral>

REGLA R2

<SI CV-PROPU-REQUISITOS ES Espiral Y CV-PROPU-APLIC ES Espiral Y
CV-PROPU-GESTION ES Espiral

ENTONCES CV-PROPU-PROYECTO ES Espiral>

REGLA R15

<SI CV-PROPU-REQUISITOS ES Objetos Y CV-PROPU-APLIC ES Objetos Y
CV-PROPU-GESTION ES Espiral

ENTONCES CV-PROPU-PROYECTO ES Objetos>

REGLA R24

<SI CV-PROPU-REQUISITOS ES Espiral Y CV-PROPU-APLIC ES Objetos Y
CV-PROPU-GESTION ES Espiral

ENTONCES CV-PROPU-PROYECTO ES Espiral>

REGLA R26

<SI CV-PROPU-REQUISITOS ES Objetos Y CV-PROPU-APLIC ES Espiral Y
CV-PROPU-GESTION ES Espiral

ENTONCES CV-PROPU-PROYECTO ES Espiral>

REGLA GP-R51

<SI NIVEL-RIESGO ES Alto
ENTONCES RIESGO-ESPIRAL ES Aceptable>

REGLA GP-R52

<SI RIESGO-ESPIRAL ES Aceptable
 ENTONCES FACTOR-RIESGO ES Existe>

REGLA GP-R43

<SI FACTOR-RIESGO ES Existe Y ANALISIS-RIESGO ES Estrategias explicitas
 ENTONCES APPLICABILIDAD-ESPIRAL ES Aplicable>

REGLA GP-R45

<SI APPLICAB-PROTOTIPO ES Aplicable Y FACTIB-PROTOTIPO ES Factible Y
 RIESGO ES Evaluable Y APPLICAB-ESPIRAL ES Aplicable
 ENTONCES CV-PROPU-GESTION ES Espiral>

REGLA R2

<SI CV-PROPU-REQUISITOS ES Espiral Y CV-PROPU-APLIC ES Espiral Y
 CV-PROPU-GESTION ES Espiral
 ENTONCES CV-PROPU-PROYECTO ES Espiral>

REGLA R15

<SI CV-PROPU-REQUISITOS ES Objetos Y CV-PROPU-APLIC ES Objetos Y
 CV-PROPU-GESTION ES Espiral
 ENTONCES CV-PROPU-PROYECTO ES Objetos>

REGLA R24

<SI CV-PROPU-REQUISITOS ES Espiral Y CV-PROPU-APLIC ES Objetos Y
 CV-PROPU-GESTION ES Espiral
 ENTONCES CV-PROPU-PROYECTO ES Espiral>

REGLA R26

<SI CV-PROPU-REQUISITOS ES Objetos Y CV-PROPU-APLIC ES Espiral Y
 CV-PROPU-GESTION ES Espiral
 ENTONCES CV-PROPU-PROYECTO ES Espiral>

REGLA GP-R44

<SI FACTOR-RIESGO ES Existe Y OBJETIVO-CALIDAD ES Mecanismos explícitos
 ENTONCES APPLICABILIDAD-ESPIRAL ES Aplicable>

REGLA GP-R45

<SI APPLICAB-PROTOTIPO ES Aplicable Y FACTIB-PROTOTIPO ES Factible Y
 RIESGO ES Evaluable Y APPLICAB-ESPIRAL ES Aplicable
 ENTONCES CV-PROPU-GESTION ES Espiral>

REGLA R2

<SI CV-PROPU-REQUISITOS ES Espiral Y CV-PROPU-APLIC ES Espiral Y
 CV-PROPU-GESTION ES Espiral
 ENTONCES CV-PROPU-PROYECTO ES Espiral>

REGLA R15

<SI CV-PROPU-REQUISITOS ES Objetos Y CV-PROPU-APLIC ES Objetos Y
 CV-PROPU-GESTION ES Espiral
 ENTONCES CV-PROPU-PROYECTO ES Objetos>

REGLA R24

<SI CV-PROPU-REQUISITOS ES Espiral Y CV-PROPU-APLIC ES Objetos Y
CV-PROPU-GESTION ES Espiral
ENTONCES CV-PROPU-PROYECTO ES Espiral>

REGLA R26

<SI CV-PROPU-REQUISITOS ES Objetos Y CV-PROPU-APLIC ES Espiral Y
CV-PROPU-GESTION ES Espiral
ENTONCES CV-PROPU-PROYECTO ES Espiral>

REGLA GP-R56

<SI FACTOR-RIESGO ES Existe Y HABILIDAD-RIESGO ES Existe Y
TECNICAS-AR ES Se dispone
ENTONCES RIESGO ES Evaluable>

REGLA GP-R45

<SI APPLICAB-PROTOTIPO ES Aplicable Y FACTIB-PROTOTIPO ES Factible Y
RIESGO ES Evaluable Y APPLICAB-ESPIRAL ES Aplicable
ENTONCES CV-PROPU-GESTION ES Espiral>

REGLA R2

<SI CV-PROPU-REQUISITOS ES Espiral Y CV-PROPU-APLIC ES Espiral Y
CV-PROPU-GESTION ES Espiral
ENTONCES CV-PROPU-PROYECTO ES Espiral>

REGLA R15

<SI CV-PROPU-REQUISITOS ES Objetos Y CV-PROPU-APLIC ES Objetos Y
CV-PROPU-GESTION ES Espiral
ENTONCES CV-PROPU-PROYECTO ES Objetos>

REGLA R24

<SI CV-PROPU-REQUISITOS ES Espiral Y CV-PROPU-APLIC ES Objetos Y
CV-PROPU-GESTION ES Espiral
ENTONCES CV-PROPU-PROYECTO ES Espiral>

REGLA R26

<SI CV-PROPU-REQUISITOS ES Objetos Y CV-PROPU-APLIC ES Espiral Y
CV-PROPU-GESTION ES Espiral
ENTONCES CV-PROPU-PROYECTO ES Espiral>

CICLO DE VIDA RECOMENDADO

Se recomienda el Ciclo de Vida en Espiral

Especificación de Requerimientos recomienda: Espiral Objetos

Tipo de Aplicación recomienda: Espiral Objetos

Gestión de Proyecto recomienda: Espiral

9.4.7 CASO EJEMPLO 6: Organizar Comisiones

Proyecto: 006-Comisiones Inscripción

Objetivo del proyecto: Desarrollar e implementar un sistema que organice las comisiones para el dictado de clases, informando: materia-profesor-alumnos de cada comisión.

Caso 6 Resultados del caso de prueba:	
Ciclo de vida esperado	Ciclo de vida sugerido por el sistema experto
CV Especificación Requerimientos 1- Espiral 2- Objetos 3- Cascada	CV Especificación Requerimientos Espiral Objetos Cascada
CV Tipo de Aplicación Cascada	Tipo de Aplicación Cascada
CV Gestión de proyecto Cascada	Gestión de proyecto Cascada
CV del Proyecto Cascada	CV del Proyecto Se recomienda el Ciclo de Vida en Cascada

El sistema experto, ha respondido satisfactoriamente de acuerdo con la definición de los casos de prueba.

Caso 6 Valores del caso de prueba		
Especificación Requerimientos		
ID	Afirmación / Pregunta	Opciones
ER1	Al comienzo del proyecto los requerimientos se encuentran definidos:	Mayoritariamente
ER2	El usuario ha explicitado los requisitos:	Formalmente, Exhaustivamente, Uniformemente
ER3	La exigencia del usuario en el cumplimiento de los requerimientos en el producto final es:	Muy exigente
Tipo de Aplicación		
ID	Afirmación / Pregunta	Opciones
TA1	El tipo de aplicación se orienta al desarrollo de :	Otros
TA2	Los componentes predominantes de la aplicación son:	Algorítmico
TA3	El comportamiento predominante en la aplicación es:	Batch
TA4	¿La progresión entre las fases del desarrollo es secuencial y uniforme?	Si
TA5	La relación entre las fases del proyecto es:	Poca dependencia,
TA6	¿El tipo de aplicación requiere el desarrollo fuertemente integrado de hardware y software?	No

TA7	<i>¿El sistema a diseñar es innovador?</i>	No
TA8	<i>Para lograr una mejor comprensión del sistema los niveles de descomposición en subsistemas son:</i>	Pocos
TA9	<i>Para lograr una mejor manipulación del sistema los niveles de descomposición en subsistemas son:</i>	Pocos
TA10	<i>Se requiere detectar en etapas tempranas del proyecto problemas de:</i>	Desconoce
TA11	<i>El Diseño del sistema a desarrollar puede resolverse por:</i>	Pocas opciones
TA12	<i>La Implementación del sistema a desarrollar puede resolverse por:</i>	Pocas opciones
TA13	<i>Las modificaciones explícitamente previstas son:</i>	Redefinición

Gestión de Proyecto

ID	Afirmación / Pregunta	Opciones
GP1	<i>El desarrollo del proyecto es responsabilidad de:</i>	La organización
GP2	<i>¿El equipo de desarrollo tiene experiencia previa en el tipo de aplicación a desarrollar?</i>	Si
GP3	<i>Es factible usar metodologías de desarrollo:</i>	Ampliamente probadas
GP4	<i>Es necesario usar metodologías de desarrollo:</i>	Ampliamente probadas
GP5	<i>¿El equipo de desarrollo tiene experiencia con las técnicas de ingeniería del software que su usarán?</i>	Si
GP6	<i>El tipo de entrega que el usuario requiere de la aplicación es:</i>	Versión completa
GP7	<i>El grado de participación del usuario en el desarrollo debe ser:</i>	Bajo
GP8	<i>¿Es necesario facilitar la introducción gradual del sistema para los usuarios?</i>	Si
GP9	<i>Se requiere una gestión de proyecto:</i>	Muy formal
GP10	<i>El grado de control que se requiere de la gestión del proyecto es:</i>	Muy ajustado
GP11	<i>¿La gestión de proyecto prevé definir explícitamente estrategias para reusar software existente?</i>	No
GP12	<i>¿Se prevé la necesidad de reutilizar los componentes del software a desarrollar en proyectos futuros?</i>	Poco necesario
GP13	<i>¿Existe un sistema previo desarrollado en objetos de forma tal que el proyecto actual es ..?</i>	Sin relación
GP14	<i>¿Existe la necesidad de reutilizar los componentes de un software ya existente desarrollado en objetos?</i>	No
GP15	<i>¿Se dispone de software para prototipar?</i>	No
GP16	<i>¿Es posible adquirir software para prototipar?</i>	No
GP17	<i>¿Existen dudas sobre la viabilidad del software?</i>	No
GP18	<i>¿Se estima la posibilidad de terminación prematura de proyectos no viables?</i>	Desconoce
GP19	<i>Se estima que el factor de riesgo en el desarrollo del sistema es:</i>	Bajo
GP20	<i>¿Se estiman riesgos técnicos en el desarrollo del sistema?</i>	No
GP21	<i>¿Se identifican los riesgos asociados con cada una de las alternativas?</i>	Desconoce
GP22	<i>¿Se cuenta con métodos y técnicas para evaluar el riesgo?</i>	Desconoce
GP23	<i>¿Se identifican las diferentes maneras de resolver los riesgos?</i>	Desconoce
GP24	<i>¿Es necesario definir explícitamente estrategias para realizar análisis de riesgo?</i>	Desconoce
GP25	<i>¿Es necesario volver a etapas anteriores en el desarrollo del sistema para la resolución de algún tópico nuevo de riesgo?</i>	Desconoce

GP26	<i>¿Es posible volver a etapas anteriores en el desarrollo del sistema cuando se identifican mejores alternativas?</i>	No
GP27	<i>¿Es necesario definir mecanismos explícitos para incorporar objetivos de calidad en el desarrollo del producto?</i>	No
GP28	<i>El uso de los mismos procedimientos para desarrollo y mantenimiento del producto software está previsto como:</i>	Necesario – Factible
GP29	<i>¿Se requiere de acuerdos confirmados para los cambios a efectuarse durante el desarrollo del proyecto?</i>	Sí

SISTEMA EXPERTO PARA LA SELECCION DE CICLOS DE VIDA DE PROYECTOS DE SOFTWARE
 Copyright 2001 by Bibiana Rossi

Fecha: 20/05/2001
 ID: 006
 Proyecto: Organizar Comisiones
 Líder de Proyecto: Lic. Carlos Leone
 Objetivo: Desarrollar un sistema que organice las comisiones para el dictado de clases, informando: materia-profesor-alumnos de cada comisión
 Fecha de inicio: 1/05/2000
 Fecha de finalización: 1/08/2000
 Ciclo de Vida recomendado: Cascada
 Ciclo de Vida seleccionado: Cascada

REGLA ER-R1
 <SI TIPO-DEFINICION ES Formalmente Y Exhaustivamente Y Uniformemente
 ENTONCES DEFINICION-REQUISITOS ES Clara Y DEFINICION-LIMITES ES Clara>

REGLA ER-R2
 <SI DEFINICION-REQUISITOS ES Clara Y DEFINICION-LIMITES ES Clara Y
 GRADO-DE-CERTIDUMBRE-INICIO ES Mayoritariamente
 ENTONCES CV-PROPUESTO-REQUISITOS ES Cascada>

REGLA ER-R5
 <SI GRADO-CUMPLIMIENTO-PRODUCTO-FINAL ES Usuario muy exigente
 ENTONCES CV-PROPUESTO-REQUISITOS ES Espiral Y Objetos>

REGLA CV-R2
 <SI RELACION-FASES-DESARROLLO ES Poca dependencia
 ENTONCES RETROALIMENTACION-FASES-DESARROLLO ES Baja>

REGLA CV-R4
 <SI PROGRE-FASES-DESARR ES Uniforme Y Secuencial Y
 RETROALIM-FASES-DESARR ES Baja Y COMP-PREDOMINANTES ES Algorítmico
 ENTONCES CV-PROP-APLICACION ES Cascada>

REGLA TA-R1
 <SI NIVELES-COMPOSICION ES Pocos subsistemas

ENTONCES COMPLEJIDAD-SUBSISTEMAS ES Baja>

REGLA TA-R11

<SI TIPO-MODIFICACION ES Redefinición

ENTONCES VARIABILIDAD-PROCESOS ES Alta>

REGLA TA-R16

<SI VARIABILIDAD-PROCESOS ES Alta

ENTONCES MODELADO-PROTOTIPO ES Conveniente>

REGLA GP-R2

<SI NIVEL-RIESGO ES Bajo

ENTONCES RIESGO-CASCADA ES Aceptable>

REGLA GP-R1

<SI NECESIDAD-METOD ES Ampliamente probada Y

FACTIBILIDAD-METOD ES Ampliamente probada

ENTONCES CONVENIENCIA-METOD ES Ampliamente probada>

REGLA GP-R3

<SI SOFT-PROTOTIPO ES No disponible Y No adquirible

ENTONCES FACTIBILIDAD-PROTOTIPO ES No factible>

REGLA GP-R5

<SI CONVENIENCIA-METOD ES Ampliamente probada Y ENTREGAS ES Versión completa Y

RIESGO-CASCADA ES Aceptable

ENTONCES CV-PROPU-GESTION ES Cascada>

REGLA GP-R12

<SI NIVEL-RIESGO ES Bajo

ENTONCES RIESGO-OO ES Aceptable>

REGLA GP-R16

<SI FACTIB-PROTOTIPO ES No factible Y RIESGO-CASCADA ES Aceptable

ENTONCES CV-PROPU-GESTION ES Cascada>

REGLA GP-R22

<SI INTRODUCCION-GRADUAL ES Necesaria

ENTONCES APLICABILIDAD-PROTOTIPO ES Aplicable>

REGLA GP-R31

<SI PROC-DES-MANTE ES Necesario Y Factible

ENTONCES PROCEDIMIENTOS-DM ES Conveniente>

REGLA GP-R32

<SI PROCEDIMIENTOS-DM ES Conveniente

ENTONCES APLICABILIDAD-ESPIRAL ES Aplicable>

REGLA GP-R33

<SI PROC-CAMBIOS ES Acuerdos confirmados

ENTONCES APLICABILIDAD-ESPIRAL ES Aplicable>

REGLA GP-R34

<SI RESPONSABILIDAD ES Organización propia
ENTONCES APPLICABILIDAD-ESPIRAL ES Aplicable>

REGLA R0

<SI CV-PROPU-REQUISITOS ES Cascada Y CV-PROPU-APLIC ES Cascada Y
CV-PROPU-GESTION ES Cascada
ENTONCES CV-PROPU-PROYECTO ES Cascada>

REGLA ER-R2

<SI DEFINICION-REQUISITOS ES Clara Y DEFINICION-LIMITES ES Clara Y
GRADO-DE-CERTIDUMBRE-INICIO ES Mayoritariamente
ENTONCES CV-PROPUESTO-REQUISITOS ES Cascada>

REGLA R0

<SI CV-PROPU-REQUISITOS ES Cascada Y CV-PROPU-APLIC ES Cascada Y
CV-PROPU-GESTION ES Cascada
ENTONCES CV-PROPU-PROYECTO ES Cascada>

REGLA R6

<SI CV-PROPU-REQUISITOS ES Objetos Y CV-PROPU-APLIC ES Cascada Y
CV-PROPU-GESTION ES Cascada
ENTONCES CV-PROPU-PROYECTO ES Cascada>

REGLA R10

<SI CV-PROPU-REQUISITOS ES Espiral Y CV-PROPU-APLIC ES Cascada Y
CV-PROPU-GESTION ES Cascada
ENTONCES CV-PROPU-PROYECTO ES Cascada>

REGLA TA-R101

<SI PROGRESION-FASES-DESARR ES Uniforme Y Secuencial Y
RETROALIMENTACION-FASES-DESARR ES Baja Y
COMPORTAMIENTO-PREDOMINANTE ES Batch
ENTONCES CV-PROPU-APLIC ES Cascada>

REGLA R0

<SI CV-PROPU-REQUISITOS ES Cascada Y CV-PROPU-APLIC ES Cascada Y
CV-PROPU-GESTION ES Cascada
ENTONCES CV-PROPU-PROYECTO ES Cascada>

REGLA R6

<SI CV-PROPU-REQUISITOS ES Objetos Y CV-PROPU-APLIC ES Cascada Y
CV-PROPU-GESTION ES Cascada
ENTONCES CV-PROPU-PROYECTO ES Cascada>

REGLA R10

<SI CV-PROPU-REQUISITOS ES Espiral Y CV-PROPU-APLIC ES Cascada Y
CV-PROPU-GESTION ES Cascada
ENTONCES CV-PROPU-PROYECTO ES Cascada>

REGLA TA-R102

<SI PROGRESION-FASES-DESARR ES Uniforme Y Secuencial Y
 COMPLEJIDAD-SUBSISTEMAS ES Baja Y
 COMPONENTES-PREDOMINANTES ES Algorítmico
 ENTONCES CV-PROPU-APLIC ES Cascada>

REGLA R0

<SI CV-PROPU-REQUISITOS ES Cascada Y CV-PROPU-APLIC ES Cascada Y
 CV-PROPU-GESTION ES Cascada
 ENTONCES CV-PROPU-PROYECTO ES Cascada>

REGLA R6

<SI CV-PROPU-REQUISITOS ES Objetos Y CV-PROPU-APLIC ES Cascada Y
 CV-PROPU-GESTION ES Cascada
 ENTONCES CV-PROPU-PROYECTO ES Cascada>

REGLA R10

<SI CV-PROPU-REQUISITOS ES Espiral Y CV-PROPU-APLIC ES Cascada Y
 CV-PROPU-GESTION ES Cascada
 ENTONCES CV-PROPU-PROYECTO ES Cascada>

REGLA TA-R103

<SI PROGRESION-FASES-DESARR ES Uniforme Y Secuencial Y
 COMPLEJIDAD-SUBSISTEMAS ES Baja Y COMPORTAMIENTO-PREDOMINANTE ES Batch
 ENTONCES CV-PROPU-APLIC ES Cascada>

REGLA R0

<SI CV-PROPU-REQUISITOS ES Cascada Y CV-PROPU-APLIC ES Cascada Y
 CV-PROPU-GESTION ES Cascada
 ENTONCES CV-PROPU-PROYECTO ES Cascada>

REGLA R6

<SI CV-PROPU-REQUISITOS ES Objetos Y CV-PROPU-APLIC ES Cascada Y
 CV-PROPU-GESTION ES Cascada
 ENTONCES CV-PROPU-PROYECTO ES Cascada>

REGLA R10

<SI CV-PROPU-REQUISITOS ES Espiral Y CV-PROPU-APLIC ES Cascada Y
 CV-PROPU-GESTION ES Cascada
 ENTONCES CV-PROPU-PROYECTO ES Cascada>

CICLO DE VIDA RECOMENDADO

Se recomienda el Ciclo de Vida en Cascada

Especificación de Requerimientos recomienda:	Espiral Objetos Cascada
Tipo de Aplicación recomienda:	Cascada
Gestión de Proyecto recomienda:	Cascada

9.4.8 CASO EJEMPLO 7: Administración Consorcio

Proyecto: 007 Consorcio

Objetivo del proyecto: Desarrollar e implementar un sistema de facturación mensual para consorcios de edificios de departamento.

Caso 7 Resultados del caso de prueba:	
Ciclo de vida esperado	Ciclo de vida sugerido por el sistema experto
CV Especificación Requerimientos Cascada	CV Especificación Requerimientos Cascada
CV Tipo de Aplicación Cascada	Tipo de Aplicación Cascada
CV Gestión de proyecto Cascada	Gestión de proyecto Cascada
CV del Proyecto Cascada	CV del Proyecto Se recomienda el Ciclo de Vida en Cascada

El sistema experto, ha respondido satisfactoriamente de acuerdo con la definición de los casos de prueba.

Caso 7 Valores del caso de prueba:		
Especificación Requerimientos		
ID	Afirmación / Pregunta	Opciones
ER1	Al comienzo del proyecto los requerimientos se encuentran definidos:	Mayoritariamente
ER2	El usuario ha explicitado los requisitos:	Formalmente, Exhaustivamente, Uniformemente
ER3	La exigencia del usuario en el cumplimiento de los requerimientos en el producto final es:	Poco exigente,

Tipo de Aplicación		
ID	Afirmación / Pregunta	Opciones
TA1	El tipo de aplicación se orienta al desarrollo de :	Otros
TA2	Los componentes predominantes de la aplicación son:	Algorítmico
TA3	El comportamiento predominante en la aplicación es:	Batch
TA4	¿La progresión entre las fases del desarrollo es secuencial y uniforme?	Si
TA5	La relación entre las fases del proyecto es:	Poca dependencia,
TA6	¿El tipo de aplicación requiere el desarrollo fuertemente integrado de hardware y software?	No

TA7	<i>¿El sistema a diseñar es innovador?</i>	No
TA8	<i>Para lograr una mejor comprensión del sistema los niveles de descomposición en subsistemas son:</i>	Pocos
TA9	<i>Para lograr una mejor manipulación del sistema los niveles de descomposición en subsistemas son:</i>	Pocos
TA10	<i>Se requiere detectar en etapas tempranas del proyecto problemas de:</i>	Desconoce
TA11	<i>El Diseño del sistema a desarrollar puede resolverse por:</i>	Pocas opciones
TA12	<i>La Implementación del sistema a desarrollar puede resolverse por:</i>	Pocas opciones
TA13	<i>Las modificaciones explícitamente previstas son:</i>	Ampliación

Gestión de Proyecto

ID	Afirmación / Pregunta	Opciones
GP1	<i>El desarrollo del proyecto es responsabilidad de:</i>	Terceros
GP2	<i>¿El equipo de desarrollo tiene experiencia previa en el tipo de aplicación a desarrollar?</i>	Si
GP3	<i>Es factible usar metodologías de desarrollo:</i>	Desconoce
GP4	<i>Es necesario usar metodologías de desarrollo:</i>	Desconoce
GP5	<i>¿El equipo de desarrollo tiene experiencia con las técnicas de ingeniería del software que su usarán?</i>	Desconoce
GP6	<i>El tipo de entrega que el usuario requiere de la aplicación es:</i>	Versión completa
GP7	<i>El grado de participación del usuario en el desarrollo debe ser:</i>	Bajo
GP8	<i>¿Es necesario facilitar la introducción gradual del sistema para los usuarios?</i>	Desconoce
GP9	<i>Se requiere una gestión de proyecto:</i>	Desconoce
GP10	<i>El grado de control que se requiere de la gestión del proyecto es:</i>	Poco ajustado
GP11	<i>¿La gestión de proyecto prevé definir explícitamente estrategias para reusar software existente?</i>	No
GP12	<i>¿Se prevé la necesidad de reutilizar los componentes del software a desarrollar en proyectos futuros?</i>	Medianamente necesario
GP13	<i>¿Existe un sistema previo desarrollado en objetos de forma tal que el proyecto actual es ...?</i>	Sin relación
GP14	<i>¿Existe la necesidad de reutilizar los componentes de un software ya existente desarrollado en objetos?</i>	No
GP15	<i>¿Se dispone de software para prototipar?</i>	No
GP16	<i>¿Es posible adquirir software para prototipar?</i>	No
GP17	<i>¿Existen dudas sobre la viabilidad del software?</i>	No
GP18	<i>¿Se estima la posibilidad de terminación prematura de proyectos no viables?</i>	No
GP19	<i>Se estima que el factor de riesgo en el desarrollo del sistema es:</i>	Sin riesgo
GP20	<i>¿Se estiman riesgos técnicos en el desarrollo del sistema?</i>	No
GP21	<i>¿Se identifican los riesgos asociados con cada una de las alternativas?</i>	Desconoce
GP22	<i>¿Se cuenta con métodos y técnicas para evaluar el riesgo?</i>	Desconoce
GP23	<i>¿Se identifican las diferentes maneras de resolver los riesgos?</i>	Desconoce
GP24	<i>¿Es necesario definir explícitamente estrategias para realizar análisis de riesgo?</i>	No
GP25	<i>¿Es necesario volver a etapas anteriores en el desarrollo del sistema para la resolución de algún tópico nuevo de riesgo?</i>	Desconoce
GP26	<i>¿Es posible volver a etapas anteriores en el desarrollo del sistema cuando se identifican mejores alternativas?</i>	Desconoce

GP27	<i>¿Es necesario definir mecanismos explícitos para incorporar objetivos de calidad en el desarrollo del producto?</i>	No
GP28	<i>El uso de los mismos procedimientos para desarrollo y mantenimiento del producto software está previsto como:</i>	Indiferente
GP29	<i>¿Se requiere de acuerdos confirmados para los cambios a efectuarse durante el desarrollo del proyecto?</i>	Desconoce

SISTEMA EXPERTO PARA LA SELECCION DE CICLOS DE VIDA DE PROYECTOS DE SOFTWARE

Copyright 2001 by Bibiana Rossi

Fecha: 20/05/2001

ID: 007

Proyecto: Consorcio

Líder de Proyecto: Lic. Carlos Beltrami

Objetivo: Desarrollar e Implementar un sistema de facturación mensual para consorcios de edificios de departamentos.

Fecha de inicio: 12/05/1998

Fecha de finalización: 20/10/1988

Ciclo de Vida recomendado: Cascada

Ciclo de Vida seleccionado: Cascada

REGLA ER-R1

<SI TIPO-DEFINICION ES Formalmente Y Exhaustivamente Y Uniformemente
ENTONCES DEFINICION-REQUISITOS ES Clara Y DEFINICION-LIMITES ES Clara>

REGLA ER-R2

<SI DEFINICION-REQUISITOS ES Clara Y DEFINICION-LIMITES ES Clara Y
GRADO-DE-CERTIDUMBRE-INICIO ES Mayoritariamente
ENTONCES CV-PROPUESTO-REQUISITOS ES Cascada>

REGLA CV-R2

<SI RELACION-FASES-DESARROLLO ES Poca dependencia
ENTONCES RETROALIMENTACION-FASES-DESARROLLO ES Baja>

REGLA CV-R4

<SI PROGRE-FASES-DESARR ES Uniforme Y Secuencial Y
RETROALIM-FASES-DESARR ES Baja Y COMP-PREDOMINANTES ES Algorítmico
ENTONCES CV-PROP-APLICACION ES Cascada>

REGLA TA-R1

<SI NIVELES-COMPOSICION ES Pocos subsistemas
ENTONCES COMPLEJIDAD-SUBSISTEMAS ES Baja>

REGLA TA-R13

<SI TIPO-MODIFICACION ES Ampliación
ENTONCES VARIABILIDAD-PROCESOS ES Alta>

REGLA TA-R16

<SI VARIABILIDAD-PROCESOS ES Alta
ENTONCES MODELADO-PROTOTIPO ES Conveniente>

REGLA GP-R3

<SI SOFT-PROTOTIPO ES No disponible Y No adquirible
ENTONCES FACTIBILIDAD-PROTOTIPO ES No factible>

REGLA GP-R4

<SI NIVEL-RIESGO ES No hay
ENTONCES RIESGO-CASCADA ES Aceptable>

REGLA GP-R16

<SI FACTIB-PROTOTIPO ES No factible Y RIESGO-CASCADA ES Aceptable
ENTONCES CV-PROPU-GESTION ES Cascada>

REGLA GP-R19

<SI NIVEL-RIESGO ES No hay
ENTONCES RIESGO-OO ES Aceptable>

REGLA ER-R2

<SI DEFINICION-REQUISITOS ES Clara Y DEFINICION-LIMITES ES Clara Y
GRADO-DE-CERTIDUMBRE-INICIO ES Mayoritariamente
ENTONCES CV-PROPUESTO-REQUISITOS ES Cascada>

REGLA R0

<SI CV-PROPU-REQUISITOS ES Cascada Y CV-PROPU-APLIC ES Cascada Y
CV-PROPU-GESTION ES Cascada
ENTONCES CV-PROPU-PROYECTO ES Cascada>

REGLA TA-R101

<SI PROGRESION-FASES-DESARR ES Uniforme Y Secuencial Y
RETROALIMENTACION-FASES-DESARR ES Baja Y
COMPORTAMIENTO-PREDOMINANTE ES Batch
ENTONCES CV-PROPU-APLIC ES Cascada>

REGLA R0

<SI CV-PROPU-REQUISITOS ES Cascada Y CV-PROPU-APLIC ES Cascada Y
CV-PROPU-GESTION ES Cascada
ENTONCES CV-PROPU-PROYECTO ES Cascada>

REGLA TA-R102

<SI PROGRESION-FASES-DESARR ES Uniforme Y Secuencial Y
COMPLEJIDAD-SUBSISTEMAS ES Baja Y
COMPONENTES-PREDOMINANTES ES Algorítmico
ENTONCES CV-PROPU-APLIC ES Cascada>

REGLA R0

<SI CV-PROPU-REQUISITOS ES Cascada Y CV-PROPU-APLIC ES Cascada Y
CV-PROPU-GESTION ES Cascada
ENTONCES CV-PROPU-PROYECTO ES Cascada>

REGLA TA-R103

<SI PROGRESION-FASES-DESARR ES Uniforme Y Secuencial Y
COMPLEJIDAD-SUBSISTEMAS ES Baja Y COMPORTAMIENTO-PREDOMINANTE ES Batch
ENTONCES CV-PROPU-APLIC ES Cascada>

REGLA R0

<SI CV-PROPU-REQUISITOS ES Cascada Y CV-PROPU-APLIC ES Cascada Y
CV-PROPU-GESTION ES Cascada
ENTONCES CV-PROPU-PROYECTO ES Cascada>

CICLO DE VIDA RECOMENDADO

Se recomienda el Ciclo de Vida en Cascada

Especificación de Requerimientos recomienda: Cascada

Tipo de Aplicación recomienda: Cascada

Gestión de Proyecto recomienda: Cascada

9.4.9 CASO EJEMPLO 8: Registración Contable

Proyecto: 008 Contable

Objetivo del proyecto: Desarrollar e implementar un sistema de registro de asientos contables para una empresa distribuidora de productos farmacológicos.

Caso 8 Resultados del caso de prueba:	
Ciclo de vida esperado	Ciclo de vida sugerido por el sistema experto
CV Especificación Requerimientos Sin respuesta	CV Especificación Requerimientos - ninguno -
CV Tipo de Aplicación Objetos	Tipo de Aplicación Objetos Cascada
CV Gestión de proyecto Cascada	Gestión de proyecto Cascada
CV del Proyecto Sin Respuesta	CV del Proyecto No se pudo sacar ninguna conclusión de los datos

El sistema experto, ha respondido satisfactoriamente de acuerdo con la definición de los casos de prueba.

Caso 8 Valores del caso de prueba		
Especificación Requerimientos		
ID	Afirmación / Pregunta	Opciones
ER1	Al comienzo del proyecto los requerimientos se encuentran definidos:	Mayoritariamente
ER2	El usuario ha explicitado los requisitos:	Exhaustivamente, Informalmente Desestructuradamente
ER3	La exigencia del usuario en el cumplimiento de los requerimientos en el producto final es:	Poco exigente,
Tipo de Aplicación		
ID	Afirmación / Pregunta	Opciones
TA1	El tipo de aplicación se orienta al desarrollo de :	Otros
TA2	Los componentes predominantes de la aplicación son:	Algorítmico
TA3	El comportamiento predominante en la aplicación es:	Interactivo
TA4	¿La progresión entre las fases del desarrollo es secuencial y uniforme?	Si
TA5	La relación entre las fases del proyecto es:	Mucha dependencia,
TA6	¿El tipo de aplicación requiere el desarrollo fuertemente integrado de hardware y software?	No
TA7	¿El sistema a diseñar es innovador?	No

TA8	Para lograr una mejor comprensión del sistema los niveles de descomposición en subsistemas son:	Pocos
TA9	Para lograr una mejor manipulación del sistema los niveles de descomposición en subsistemas son:	Pocos
TA10	Se requiere detectar en etapas tempranas del proyecto problemas de:	Desconoce
TA11	El Diseño del sistema a desarrollar puede resolverse por:	Pocas opciones
TA12	La Implementación del sistema a desarrollar puede resolverse por:	Pocas opciones
TA13	Las modificaciones explicitamente previstas son:	Extensión - Ampliación

Gestión de Proyecto

ID	Afirmación / Pregunta	Opciones
GP1	El desarrollo del proyecto es responsabilidad de:	Terceros
GP2	¿El equipo de desarrollo tiene experiencia previa en el tipo de aplicación a desarrollar?	Si
GP3	Es factible usar metodologías de desarrollo:	Medianamente probadas
GP4	Es necesario usar metodologías de desarrollo:	Desconoce
GP5	¿El equipo de desarrollo tiene experiencia con las técnicas de ingeniería del software que su usarán?	Desconoce
GP6	El tipo de entrega que el usuario requiere de la aplicación es:	Versión parcial
GP7	El grado de participación del usuario en el desarrollo debe ser:	Medio
GP8	¿Es necesario facilitar la introducción gradual del sistema para los usuarios?	Desconoce
GP9	Se requiere una gestión de proyecto:	Medianamente formal
GP10	El grado de control que se requiere de la gestión del proyecto es:	Medianamente ajustado
GP11	¿La gestión de proyecto prevé definir explícitamente estrategias para reusar software existente?	Desconoce
GP12	¿Se prevé la necesidad de reutilizar los componentes del software a desarrollar en proyectos futuros?	Poco necesario
GP13	¿Existe un sistema previo desarrollado en objetos de forma tal que el proyecto actual es ..?	Sin relación
GP14	¿Existe la necesidad de reutilizar los componentes de un software ya existente desarrollado en objetos?	No
GP15	¿Se dispone de software para prototipar?	No
GP16	¿Es posible adquirir software para prototipar?	No
GP17	¿Existen dudas sobre la viabilidad del software?	No
GP18	¿Se estima la posibilidad de terminación prematura de proyectos no viables?	No
GP19	Se estima que el factor de riesgo en el desarrollo del sistema es:	Bajo
GP20	¿Se estiman riesgos técnicos en el desarrollo del sistema?	No
GP21	¿Se identifican los riesgos asociados con cada una de las alternativas?	Desconoce
GP22	¿Se cuenta con métodos y técnicas para evaluar el riesgo?	Desconoce
GP23	¿Se identifican las diferentes maneras de resolver los riesgos?	Desconoce
GP24	¿Es necesario definir explícitamente estrategias para realizar análisis de riesgo?	Desconoce
GP25	¿Es necesario volver a etapas anteriores en el desarrollo del sistema para la resolución de algún tópico nuevo de riesgo?	Desconoce
GP26	¿Es posible volver a etapas anteriores en el desarrollo del sistema cuando se identifican mejores alternativas?	Desconoce
GP27	¿Es necesario definir mecanismos explícitos para incorporar objetivos de calidad en el desarrollo del producto?	No
GP28	El uso de los mismos procedimientos para desarrollo y mantenimiento del producto software está previsto como:	Indiferente
GP29	¿Se requiere de acuerdos confirmados para los cambios a efectuarse durante el desarrollo del proyecto?	Si

SISTEMA EXPERTO PARA LA SELECCION DE CICLOS DE VIDA DE PROYECTOS DE SOFTWARE

Copyright 2001 by Bibiana Rossi

Fecha: 20/05/2001

ID: 008

Proyecto: Registración Contable

Líder de Proyecto: Lic. Carlos Beltrami

Objetivo: Desarrollar e Implementar un sistema de registración de asientos contables para una empresa distribuidora de productos farmacológicos.

Fecha de inicio: 10/05/1998

Fecha de finalización: 30/11/1998

Ciclo de Vida recomendado: Cascada

Ciclo de Vida seleccionado: Cascada

REGLA CV-R3

<SI RELACION-FASES-DESARROLLO ES Mucha dependencia
ENTONCES RETROALIMENTACION-FASES-DESARROLLO ES Alta>

REGLA TA-R1

<SI NIVELES-COMPOSICION ES Pocos subsistemas
ENTONCES COMPLEJIDAD-SUBSISTEMAS ES Baja>

REGLA TA-R7

<SI COMPORTAMIENTO-PREDOMINANTE ES Interactivo
ENTONCES MODELADO-OBJETOS ES Conveniente>

REGLA TA-R12

<SI TIPO-MODIFICACION ES Extensión
ENTONCES VARIABILIDAD-PROCESOS ES Alta>

REGLA TA-R13

<SI TIPO-MODIFICACION ES Ampliación
ENTONCES VARIABILIDAD-PROCESOS ES Alta>

REGLA TA-R16

<SI VARIABILIDAD-PROCESOS ES Alta
ENTONCES MODELADO-PROTOTIPO ES Conveniente>

REGLA TA-R20

<SI MODELADO-OBJETOS ES Conveniente Y MODELADO-PROTOTIPO ES Conveniente
ENTONCES CV-PROPU-APLICACION ES Objetos>

REGLA GP-R2

<SI NIVEL-RIESGO ES Bajo
ENTONCES RIESGO-CASCADA ES Aceptable>

REGLA GP-R3

<SI SOFT-PROTOTIPO ES No disponible Y No adquirible
ENTONCES FACTIBILIDAD-PROTOTIPO ES No factible>

REGLA GP-R12

<SI NIVEL-RIESGO ES Bajo
ENTONCES RIESGO-OO ES Aceptable>

REGLA GP-R16

<SI FACTIB-PROTOTIPO ES No factible Y RIESGO-CASCADA ES Aceptable
ENTONCES CV-PROPU-GESTION ES Cascada>

REGLA GP-R28

<SI ENTREGAS ES Versión parcial

ENTONCES ENTREGA-INTERMEDIA ES Existe> REGLA GP-R29 <SI ENTREGA-INTERMEDIA ES Existe ENTONCES APLICABILIDAD-OO ES Aplicable Y APLICABILIDAD-PROTOTIPO ES Aplicable> REGLA GP-R33 <SI PROC-CAMBIOS ES Acuerdos confirmados ENTONCES APLICABILIDAD-ESPIRAL ES Aplicable> REGLA TA-R102 <SI PROGRESION-FASES-DESARR ES Uniforme Y Secuencial Y COMPLEJIDAD-SUBSISTEMAS ES Baja Y COMPONENTES-PREDOMINANTES ES Algorítmico ENTONCES CV-PROPU-APLIC ES Cascada>
CICLO DE VIDA RECOMENDADO
No se pudo sacar ninguna conclusión de los datos
Especificación de Requerimientos recomienda: Tipo de Aplicación recomienda: Gestión de Proyecto recomienda:

9.4.10 CASO EJEMPLO 9: Portal Empleos

Proyecto: 009 Empleos

Objetivo del proyecto: Desarrollar e implementar un sistema de acceso masivo por Internet para oferta y búsqueda de empleos.

Caso 9 Resultados del caso de prueba:	
Ciclo de vida esperado	Ciclo de vida sugerido por el sistema experto
CV Especificación Requerimientos 1- Espiral 2- Objetos	CV Especificación Requerimientos Espiral Objetos
CV Tipo de Aplicación Objetos	Tipo de Aplicación Objetos
CV Gestión de proyecto Objetos	Gestión de proyecto Objetos
CV del Proyecto Objetos	CV del Proyecto Se recomienda el Ciclo de Vida de Objetos

El sistema experto, ha respondido satisfactoriamente de acuerdo con la definición de los casos de prueba.

Caso 9 Valores del caso de prueba		
Especificación Requerimientos		
ID	Afirmación / Pregunta	Opciones
ER1	Al comienzo del proyecto los requerimientos se encuentran definidos:	Parcialmente
ER2	El usuario ha explicitado los requisitos:	Informalmente Incompletos Desestructuradamente
ER3	La exigencia del usuario en el cumplimiento de los requerimientos en el producto final es:	Muy exigente,
Tipo de Aplicación		
ID	Afirmación / Pregunta	Opciones
TA1	El tipo de aplicación se orienta al desarrollo de :	Otros
TA2	Los componentes predominantes de la aplicación son:	Grafico - Otro
TA3	El comportamiento predominante en la aplicación es:	Interactivo - Tiempo Real
TA4	¿La progresión entre las fases del desarrollo es secuencial y uniforme?	No
TA5	La relación entre las fases del proyecto es:	Mucha dependencia,
TA6	¿El tipo de aplicación requiere el desarrollo fuertemente integrado de hardware y software?	No
TA7	¿El sistema a diseñar es innovador?	No
TA8	Para lograr una mejor comprensión del sistema los niveles de descomposición en subsistemas son:	Muchos
TA9	Para lograr una mejor manipulación del sistema los niveles de descomposición en subsistemas son:	Muchos
TA10	Se requiere detectar en etapas tempranas del proyecto problemas de:	Arquitectura – Eficacia
TA11	El Diseño del sistema a desarrollar puede resolverse por:	Pocas opciones
TA12	La Implementación del sistema a desarrollar puede resolverse por:	Pocas opciones
TA13	Las modificaciones explícitamente previstas son:	Emisión gradual
Gestión de Proyecto		
ID	Afirmación / Pregunta	Opciones
GP1	El desarrollo del proyecto es responsabilidad de:	Terceros
GP2	¿El equipo de desarrollo tiene experiencia previa en el tipo de aplicación a desarrollar?	No
GP3	Es factible usar metodologías de desarrollo:	Medianamente probadas
GP4	Es necesario usar metodologías de desarrollo:	Medianamente probadas
GP5	¿El equipo de desarrollo tiene experiencia con las técnicas de ingeniería del software que su usarán?	Si
GP6	El tipo de entrega que el usuario requiere de la aplicación es:	Versión parcial
GP7	El grado de participación del usuario en el desarrollo debe ser:	Medio
GP8	¿Es necesario facilitar la introducción gradual del sistema para los usuarios?	No
GP9	Se requiere una gestión de proyecto:	Medianamente formal
GP10	El grado de control que se requiere de la gestión del proyecto es:	Muy ajustado
GP11	¿La gestión de proyecto prevé definir explícitamente estrategias para reusar software existente?	Si

GP12	<i>¿Se prevé la necesidad de reutilizar los componentes del software a desarrollar en proyectos futuros?</i>	Muy necesario
GP13	<i>¿Existe un sistema previo desarrollado en objetos de forma tal que el proyecto actual es ..?</i>	Sin relación
GP14	<i>¿Existe la necesidad de reutilizar los componentes de un software ya existente desarrollado en objetos?</i>	No
GP15	<i>¿Se dispone de software para prototipar?</i>	Desconoce
GP16	<i>¿Es posible adquirir software para prototipar?</i>	Desconoce
GP17	<i>¿Existen dudas sobre la viabilidad del software?</i>	No
GP18	<i>¿Se estima la posibilidad de terminación prematura de proyectos no viables?</i>	No
GP19	<i>Se estima que el factor de riesgo en el desarrollo del sistema es:</i>	Medio
GP20	<i>¿Se estiman riesgos técnicos en el desarrollo del sistema?</i>	Si
GP21	<i>¿Se identifican los riesgos asociados con cada una de las alternativas?</i>	Si
GP22	<i>¿Se cuenta con métodos y técnicas para evaluar el riesgo?</i>	Si
GP23	<i>¿Se identifican las diferentes maneras de resolver los riesgos?</i>	Si
GP24	<i>¿Es necesario definir explícitamente estrategias para realizar análisis de riesgo?</i>	Si
GP25	<i>¿Es necesario volver a etapas anteriores en el desarrollo del sistema para la resolución de algún tópico nuevo de riesgo?</i>	Desconoce
GP26	<i>¿Es posible volver a etapas anteriores en el desarrollo del sistema cuando se identifican mejores alternativas?</i>	Desconoce
GP27	<i>¿Es necesario definir mecanismos explícitos para incorporar objetivos de calidad en el desarrollo del producto?</i>	Si
GP28	<i>El uso de los mismos procedimientos para desarrollo y mantenimiento del producto software está previsto como:</i>	Necesario
GP29	<i>¿Se requiere de acuerdos confirmados para los cambios a efectuarse durante el desarrollo del proyecto?</i>	Si

SISTEMA EXPERTO PARA LA SELECCION DE CICLOS DE VIDA DE PROYECTOS DE SOFTWARE
 Copyright 2001 by Bibiana Rossi

Fecha: 21/07/2001
 ID: 009
 Proyecto: Portal Empleos
 Líder de Proyecto: Ing. Mariano Wechsler
 Objetivo: Desarrollar e Implementar un sistema de acceso masivo por Internet para oferta y búsqueda de empleos.
 Fecha de inicio: 10/07/1999
 Fecha de finalización: 20/05/2000
 Ciclo de Vida recomendado: Objetos
 Ciclo de Vida seleccionado: Objetos

REGLA ER-R3
 <SI TIPO-DEFINICION ES Informalmente Y Incompleto Y Desestructuradamente
 ENTONCES DEFINICION-REQUISITOS ES Incierta Y DEFINICION-LIMITES ES Incierta>

REGLA ER-R4

<SI DEFINICION-REQUISITOS ES Incierta Y DEFINICION-LIMITES ES Incierta Y
GRADO-DE-CERTIDUMBRE-INICIO ES Parcialmente
ENTONCES CV-PROPUESTO-REQUISITOS ES Espiral Y Objetos>

REGLA ER-R5

<SI GRADO-CUMPLIMIENTO-PRODUCTO-FINAL ES Usuario muy exigente
ENTONCES CV-PROPUESTO-REQUISITOS ES Espiral Y Objetos>

REGLA CV-R3

<SI RELACION-FASES-DESARROLLO ES Mucha dependencia
ENTONCES RETROALIMENTACION-FASES-DESARROLLO ES Alta>

REGLA CV-R5

<SI PROGRE-FASES-DESARR ES No uniforme Y Secuencial
ENTONCES MODELADO-PROTOTIPO ES Conveniente>

REGLA TA-R5

<SI COMPONENTES-PREDOMINANTES ES Gráficos
ENTONCES MODELADO-OBJETOS ES Conveniente>

REGLA TA-R7

<SI COMPORTAMIENTO-PREDOMINANTE ES Interactivo
ENTONCES MODELADO-OBJETOS ES Conveniente>

REGLA TA-R8

<SI COMPORTAMIENTO-PREDOMINANTE ES Tiempo real
ENTONCES MODELADO OBJETOS ES Conveniente>

REGLA TA-R14

<SI FACTORES-DISEÑO ES Problemas de arquitectura
ENTONCES MODELADO-PROTOTIPO ES Conveniente>

REGLA TA-R15

<SI FACTORES-DISEÑO ES Problemas de eficacia
ENTONCES MODELADO-PROTOTIPO ES Conveniente>

REGLA TA-R20

<SI MODELADO-OBJETOS ES Conveniente Y MODELADO-PROTOTIPO ES Conveniente
ENTONCES CV-PROPU-APLICACION ES Objetos>

REGLA GP-R11

<SI REUSO-APLIC-FUTURA ES Muy necesario ENTONCES SISTEMA-OO ES Existe>

REGLA GP-R13

<SI NIVEL-RIESGO ES Mediano ENTONCES RIESGO-OO ES Aceptable>

REGLA GP-R15

<SI FORMALIDAD ES Medianamente Y CONVENIENCIA-METOD ES Medianamente probada
ENTONCES APPLICABILIDAD-OO ES Aplicable>

REGLA GP-R18

<SI APPLICABILIDAD-OO ES Aplicable Y SISTEMA-OO ES Existe Y RIESGO-OO ES Aceptable

ENTONCES CV-PROPU-GESTION ES Objetos>

REGLA GP-R26

<SI REUSO-APLIC-EXISTENTE ES Estrategias explicitas

ENTONCES APLICABILIDAD-OO ES Aplicable>

REGLA GP-R18

<SI APLICABILIDAD-OO ES Aplicable Y SISTEMA-OO ES Existe Y RIESGO-OO ES Aceptable

ENTONCES CV-PROPU-GESTION ES Objetos>

REGLA GP-R27

<SI EXPERIENCIA-PREVIA ES No hay

ENTONCES APLICABILIDAD-PROTOTIPO ES Aplicable>

REGLA GP-R28

<SI ENTREGAS ES Versión parcial

ENTONCES ENTREGA-INTERMEDIA ES Existe>

REGLA GP-R29

<SI ENTREGA-INTERMEDIA ES Existe

ENTONCES APLICABILIDAD-OO ES Aplicable Y APLICABILIDAD-PROTOTIPO ES Aplicable>

REGLA GP-R33

<SI PROC-CAMBIOS ES Acuerdos confirmados

ENTONCES APLICABILIDAD-ESPIRAL ES Aplicable>

REGLA GP-R36

<SI CATEGORIA-RIESGO ES Técnico

ENTONCES FACTOR-RIESGO ES Existe>

REGLA GP-R37

<SI NECESIDAD-METOD ES Medianamente probada Y

FACTIB-METOD ES Medianamente probada

ENTONCES CONVENIENCIA-METOD ES Medianamente probada>

REGLA GP-R15

<SI FORMALIDAD ES Medianamente Y CONVENIENCIA-METOD ES Medianamente probada

ENTONCES APLICABILIDAD-OO ES Aplicable>

REGLA GP-R18

<SI APLICABILIDAD-OO ES Aplicable Y SISTEMA-OO ES Existe Y RIESGO-OO ES Aceptable

ENTONCES CV-PROPU-GESTION ES Objetos>

REGLA GP-R38

<SI RIESGO-ALTERNATIVAS ES Se identifican Y

IDENTIFICACION-ALTERNATIVAS ES Se identifican

ENTONCES HABILIDAD-RIESGO ES Existe>

REGLA GP-R43

<SI FACTOR-RIESGO ES Existe Y ANALISIS-RIESGO ES Estrategias explicitas

ENTONCES APLICABILIDAD-ESPIRAL ES Aplicable>

REGLA GP-R44

<SI FACTOR-RIESGO ES Existe Y OBJETIVO-CALIDAD ES Mecanismos explícitos
ENTONCES APPLICABILIDAD-ESPIRAL ES Aplicable>

REGLA R1

<SI CV-PROPU-REQUISITOS ES Objetos Y CV-PROPU-APLIC ES Objetos Y
CV-PROPU-GESTION ES Objetos
ENTONCES CV-PROPU-PROYECTO ES Objetos>

REGLA R18

<SI CV-PROPU-REQUISITOS ES Espiral Y CV-PROPU-APLIC ES Objetos Y
CV-PROPU-GESTION ES Objetos
ENTONCES CV-PROPU-PROYECTO ES Objetos>

REGLA ER-R4

<SI DEFINICION-REQUISITOS ES Incierta Y DEFINICION-LIMITES ES Incierta Y
GRADO-DE-CERTIDUMBRE-INICIO ES Parcialmente
ENTONCES CV-PROPU-REQUISITOS ES Espiral Y Objetos>

REGLA R1

<SI CV-PROPU-REQUISITOS ES Objetos Y CV-PROPU-APLIC ES Objetos Y
CV-PROPU-GESTION ES Objetos
ENTONCES CV-PROPU-PROYECTO ES Objetos>

REGLA R18

<SI CV-PROPU-REQUISITOS ES Espiral Y CV-PROPU-APLIC ES Objetos Y
CV-PROPU-GESTION ES Objetos
ENTONCES CV-PROPU-PROYECTO ES Objetos>

REGLA GP-R50

<SI NIVEL-RIESGO ES Mediano
ENTONCES RIESGO-ESPIRAL ES Aceptable>

REGLA GP-R52

<SI RIESGO-ESPIRAL ES Aceptable
ENTONCES FACTOR-RIESGO ES Existe>

REGLA GP-R43

<SI FACTOR-RIESGO ES Existe Y ANALISIS-RIESGO ES Estrategias explicitas
ENTONCES APPLICABILIDAD-ESPIRAL ES Aplicable>

REGLA GP-R44

<SI FACTOR-RIESGO ES Existe Y OBJETIVO-CALIDAD ES Mecanismos explícitos
ENTONCES APPLICABILIDAD-ESPIRAL ES Aplicable>

REGLA GP-R56

<SI FACTOR-RIESGO ES Existe Y HABILIDAD-RIESGO ES Existe Y
TECNICAS-AR ES Se dispone
ENTONCES RIESGO ES Evaluable>

CICLO DE VIDA RECOMENDADO	
Se recomienda el Ciclo de Vida de Objetos	
Especificación de Requerimientos recomienda:	Espiral Objetos
Tipo de Aplicación recomienda:	Objetos
Gestión de Proyecto recomienda:	Objetos

9.4.11 CASO EJEMPLO 10: Pedidos Drogería

Proyecto: 010 Drogería

Objetivo del proyecto: Desarrollar e implementar un sistema de toma de pedidos en línea para una distribuidora de productos farmacéuticos, con el armado automático del pedido.

Caso 10 Resultados del caso de prueba:	
Ciclo de vida esperado	Ciclo de vida sugerido por el sistema experto
CV Especificación Requerimientos 1- Espiral 2- Objetos	CV Especificación Requerimientos Espiral Objetos
CV Tipo de Aplicación 1- Espiral 2- Objetos	Tipo de Aplicación Espiral Objetos
CV Gestión de proyecto Espiral	Gestión de proyecto Espiral
CV del Proyecto Espiral	CV del Proyecto Se recomienda el Ciclo de Vida en Espiral

El sistema experto, ha respondido satisfactoriamente de acuerdo con la definición de los casos de prueba.

Caso 10 Valores del caso de prueba		
Especificación Requerimientos		
ID	Afirmación / Pregunta	Opciones
ER1	<i>Al comienzo del proyecto los requerimientos se encuentran definidos:</i>	<i>Parcialmente</i>
ER2	<i>El usuario ha explicitado los requisitos:</i>	<i>Informalmente Incompletos</i>

		Desestructuradamente
ER3	<i>La exigencia del usuario en el cumplimiento de los requerimientos en el producto final es:</i>	Muy exigente,
Tipo de Aplicación		
ID	Afirmación / Pregunta	Opciones
TA1	<i>El tipo de aplicación se orienta al desarrollo de :</i>	Otros
TA2	<i>Los componentes predominantes de la aplicación son:</i>	Matemático Algorítmico - Otro
TA3	<i>El comportamiento predominante en la aplicación es:</i>	Interactivo
TA4	<i>¿La progresión entre las fases del desarrollo es secuencial y uniforme?</i>	No
TA5	<i>La relación entre las fases del proyecto es:</i>	Mucha dependencia,
TA6	<i>¿El tipo de aplicación requiere el desarrollo fuertemente integrado de hardware y software?</i>	Si
TA7	<i>¿El sistema a diseñar es innovador?</i>	Si
TA8	<i>Para lograr una mejor comprensión del sistema los niveles de descomposición en subsistemas son:</i>	Muchos
TA9	<i>Para lograr una mejor manipulación del sistema los niveles de descomposición en subsistemas son:</i>	Muchos
TA10	<i>Se requiere detectar en etapas tempranas del proyecto problemas de:</i>	Arquitectura – Eficacia
TA11	<i>El Diseño del sistema a desarrollar puede resolverse por:</i>	Varias opciones
TA12	<i>La Implementación del sistema a desarrollar puede resolverse por:</i>	Varias opciones
TA13	<i>Las modificaciones explícitamente previstas son:</i>	Redefinición – Extensión Ampliación
Gestión de Proyecto		
ID	Afirmación / Pregunta	Opciones
GP1	<i>El desarrollo del proyecto es responsabilidad de:</i>	La organización
GP2	<i>¿El equipo de desarrollo tiene experiencia previa en el tipo de aplicación a desarrollar?</i>	No
GP3	<i>Es factible usar metodologías de desarrollo:</i>	Poco probadas
GP4	<i>Es necesario usar metodologías de desarrollo:</i>	Poco probadas
GP5	<i>¿El equipo de desarrollo tiene experiencia con las técnicas de ingeniería del software que su usarán?</i>	No
GP6	<i>El tipo de entrega que el usuario requiere de la aplicación es:</i>	Versión temprana
GP7	<i>El grado de participación del usuario en el desarrollo debe ser:</i>	Alto
GP8	<i>¿Es necesario facilitar la introducción gradual del sistema para los usuarios?</i>	Si
GP9	<i>Se requiere una gestión de proyecto:</i>	Medianamente formal
GP10	<i>El grado de control que se requiere de la gestión del proyecto es:</i>	Medianamente ajustado
GP11	<i>¿La gestión de proyecto prevé definir explícitamente estrategias para reusar software existente?</i>	No
GP12	<i>¿Se prevé la necesidad de reutilizar los componentes del software a desarrollar en proyectos futuros?</i>	Medianamente necesario
GP13	<i>¿Existe un sistema previo desarrollado en objetos de forma tal que el proyecto actual es ...?</i>	Sin relación
GP14	<i>¿Existe la necesidad de reutilizar los componentes de un software ya existente desarrollado en objetos?</i>	No
GP15	<i>¿Se dispone de software para prototipar?</i>	No

GP16	<i>¿Es posible adquirir software para prototipar?</i>	Si
GP17	<i>¿Existen dudas sobre la viabilidad del software?</i>	No
GP18	<i>¿Se estima la posibilidad de terminación prematura de proyectos no viables?</i>	Si
GP19	<i>Se estima que el factor de riesgo en el desarrollo del sistema es:</i>	Alto
GP20	<i>¿Se estiman riesgos técnicos en el desarrollo del sistema?</i>	Si
GP21	<i>¿Se identifican los riesgos asociados con cada una de las alternativas?</i>	Si
GP22	<i>¿Se cuenta con métodos y técnicas para evaluar el riesgo?</i>	Si
GP23	<i>¿Se identifican las diferentes maneras de resolver los riesgos?</i>	Si
GP24	<i>¿Es necesario definir explícitamente estrategias para realizar análisis de riesgo?</i>	Si
GP25	<i>¿Es necesario volver a etapas anteriores en el desarrollo del sistema para la resolución de algún tópico nuevo de riesgo?</i>	Si
GP26	<i>¿Es posible volver a etapas anteriores en el desarrollo del sistema cuando se identifican mejores alternativas?</i>	Si
GP27	<i>¿Es necesario definir mecanismos explícitos para incorporar objetivos de calidad en el desarrollo del producto?</i>	Si
GP28	<i>El uso de los mismos procedimientos para desarrollo y mantenimiento del producto software está previsto como:</i>	Necesario – Factible
GP29	<i>¿Se requiere de acuerdos confirmados para los cambios a efectuarse durante el desarrollo del proyecto?</i>	Si

SISTEMA EXPERTO PARA LA SELECCION DE CICLOS DE VIDA DE PROYECTOS DE SOFTWARE
 Copyright 2001 by Bibiana Rossi

Fecha: 20/05/2001
 ID: 010
 Proyecto: Pedidos Droguería
 Líder de Proyecto: Lic. Carlos Leone
 Objetivo: Desarrollar e Implementar un sistema de toma de pedidos en línea para una distribuidora de productos farmacéuticos, con el armado automático del pedido.
 Fecha de inicio: 14/03/1998
 Fecha de finalización: 30/05/1999
 Ciclo de Vida recomendado: Espiral
 Ciclo de Vida seleccionado: Espiral

REGLA ER-R3
 <SI TIPO-DEFINICION ES Informalmente Y Incompleto Y Desestructuradamente
 ENTONCES DEFINICION-REQUISITOS ES Incierta Y DEFINICION-LIMITES ES Incierta>
 REGLA ER-R4
 <SI DEFINICION-REQUISITOS ES Incierta Y DEFINICION-LIMITES ES Incierta Y
 GRADO-DE-CERTIDUMBRE-INICIO ES Parcialmente
 ENTONCES CV-PROPUESTO-REQUISITOS ES Espiral Y Objetos>
 REGLA ER-R5

<SI GRADO-CUMPLIMIENTO-PRODUCTO-FINAL ES Usuario muy exigente
ENTONCES CV-PROPUESTO-REQUISITOS ES Espiral Y Objetos>

REGLA CV-R3

<SI RELACION-FASES-DESARROLLO ES Mucha dependencia
ENTONCES RETROALIMENTACION-FASES-DESARROLLO ES Alta>

REGLA CV-R5

<SI PROGRE-FASES-DESARR ES No uniforme Y Secuencial
ENTONCES MODELADO-PROTOTIPO ES Conveniente>

REGLA TA-R4

<SI COMPONENTES-PREDOMINANTES ES Matemáticos
ENTONCES MODELADO-OBJETOS ES Conveniente>

REGLA TA-R7

<SI COMPORTAMIENTO-PREDOMINANTE ES Interactivo
ENTONCES MODELAO-OBJETOS ES Conveniente>

REGLA TA-R12

<SI TIPO-MODIFICACION ES Extensión
ENTONCES VARIABILIDAD-PROCESOS ES Alta>

REGLA TA-R11

<SI TIPO-MODIFICACION ES Redefinición
ENTONCES VARIABILIDAD-PROCESOS ES Alta>

REGLA TA-R13

<SI TIPO-MODIFICACION ES Ampliación
ENTONCES VARIABILIDAD-PROCESOS ES Alta>

REGLA TA-R14

<SI FACTORES-DISEÑO ES Problemas de arquitectura
ENTONCES MODELADO-PROTOTIPO ES Conveniente>

REGLA TA-R15

<SI FACTORES-DISEÑO ES Problemas de eficacia
ENTONCES MODELADO-PROTOTIPO ES Conveniente>

REGLA TA-R16

<SI VARIABILIDAD-PROCESOS ES Alta
ENTONCES MODELADO-PROTOTIPO ES Conveniente>

REGLA TA-R17

<SI EXISTENCIA-APLICACIONES ES Innovador
ENTONCES MODELADO PROTOTIPO ES Conveniente>

REGLA TA-R20

<SI MODELADO-OBJETOS ES Conveniente Y MODELADO-PROTOTIPO ES Conveniente
ENTONCES CV-PROPU-APLICACION ES Objetos>

REGLA TA-R21

<SI MODELADO-OBJETOS ES Conveniente Y MODELADO-PROTOTIPO ES Conveniente Y

INTREGRACION-HW-SW ES Fuertemente
 ENTONCES CV-PROPU-APLICACION ES Espiral>
 REGLA TA-R22
 <SI POSIBILIDAD-DISEÑO ES Varias opciones
 ENTONCES OPCIONALIDAD ES Múltiple>
 REGLA TA-R23
 <SI POSIBILIDAD-IMPLEMENTACION ES Varias opciones
 ENTONCES OPCIONALIDAD ES Múltiple>
 REGLA TA-R24
 <SI MODELADO-OBJETOS ES Conveniente Y MODELADO-PROTOTIPO ES Conveniente Y
 OPCIONALIDAD ES Múltiple
 ENTONCES CV-PROPU-APLICACION ES Espiral>
 REGLA GP-R20
 <SI SOFT-PROTOTIPO ES Factible
 ENTONCES FACTIBILIDAD-PROTOTIPO ES Factible>
 REGLA GP-R22
 <SI INTRODUCCION-GRADUAL ES Necesaria
 ENTONCES APLICABILIDAD-PROTOTIPO ES Aplicable>
 REGLA GP-R23
 <SI PARTICIPACION-USUARIO ES Fuerte
 ENTONCES APLICABILIDAD-PROTOTIPO ES Aplicable>
 REGLA GP-R24
 <SI ENTREGAS ES Versión temprana
 ENTONCES ENTREGA-INTERMEDIA ES Existe>
 REGLA GP-R27
 <SI EXPERIENCIA-PREVIA ES No hay
 ENTONCES APLICABILIDAD-PROTOTIPO ES Aplicable>
 REGLA GP-R29
 <SI ENTREGA-INTERMEDIA ES Existe
 ENTONCES APLICABILIDAD-OO ES Aplicable Y APLICABILIDAD-PROTOTIPO ES Aplicable>
 REGLA GP-R30
 <SI EXPERIENCIA-TECNICAS-IS ES No hay
 ENTONCES APLICABILIDAD-PROTOTIPO ES Aplicable>
 REGLA GP-R31
 <SI PROC-DES-MANTE ES Necesario Y Factible
 ENTONCES PROCEDIMIENTOS-DM ES Conveniente>
 REGLA GP-R32
 <SI PROCEDIMIENTOS-DM ES Conveniente
 ENTONCES APLICABILIDAD-ESPIRAL ES Aplicable>
 REGLA GP-R33

<SI PROC-CAMBIOS ES Acuerdos confirmados
ENTONCES APPLICABILIDAD-ESPIRAL ES Aplicable>

REGLA GP-R34

<SI RESPONSABILIDAD ES Organización propia
ENTONCES APPLICABILIDAD-ESPIRAL ES Aplicable>

REGLA GP-R36

<SI CATEGORIA-RIESGO ES Técnico
ENTONCES FACTOR-RIESGO ES Existe>

REGLA GP-R38

<SI RIESGO-ALTERNATIVAS ES Se identifican Y
IDENTIFICACION-ALTERNATIVAS ES Se identifican
ENTONCES HABILIDAD-RIESGO ES Existe>

REGLA GP-R39

<SI HABILIDAD-RIESGO ES Existe Y
IDENTIFICACION-ALTERNATIVAS ES Etapas anteriores
ENTONCES APPLICABILIDAD-ESPIRAL ES Aplicable>

REGLA GP-R40

<FORMULACION EXTERNA>

REGLA GP-R41

<SI HABILIDAD-RIESGO ES Existe Y RIESGOS-ALTERNATIVA ES Etapas anteriores
ENTONCES APPLICABILIDAD-ESPIRAL ES Aplicable>

REGLA GP-R42

<SI HABILIDAD-RIESGO ES Existe Y TERMINACION-PROYECTOS ES Prematura
ENTONCES APPLICABILIDAD-ESPIRAL ES Aplicable>

REGLA GP-R43

<SI FACTOR-RIESGO ES Existe Y ANALISIS-RIESGO ES Estrategias explicitas
ENTONCES APPLICABILIDAD-ESPIRAL ES Aplicable>

REGLA GP-R44

<SI FACTOR-RIESGO ES Existe Y OBJETIVO-CALIDAD ES Mecanismos explícitos
ENTONCES APPLICABILIDAD-ESPIRAL ES Aplicable>

REGLA GP-R51

<SI NIVEL-RIESGO ES Alto
ENTONCES RIESGO-ESPIRAL ES Aceptable>

REGLA GP-R52

<SI RIESGO-ESPIRAL ES Aceptable
ENTONCES FACTOR-RIESGO ES Existe>

REGLA GP-R43

<SI FACTOR-RIESGO ES Existe Y ANALISIS-RIESGO ES Estrategias explicitas
ENTONCES APPLICABILIDAD-ESPIRAL ES Aplicable>

REGLA GP-R44

<SI FACTOR-RIESGO ES Existe Y OBJETIVO-CALIDAD ES Mecanismos explícitos
 ENTONCES APLICABILIDAD-ESPIRAL ES Aplicable>

REGLA GP-R53

<SI NECESIDAD-METOD ES Poco probada Y FACTIB-METOD ES Poco probada
 ENTONCES CONVENIENCIA-METOD ES Poco probada>

REGLA GP-R54

<SI FORMALIDAD ES Medianamente Y CONVENIENCIA-METOD ES Poco probada
 ENTONCES APLICABILIDAD-ESPIRAL ES Aplicable>

REGLA GP-R56

<SI FACTOR-RIESGO ES Existe Y HABILIDAD-RIESGO ES Existe Y
 TECNICAS-AR ES Se dispone
 ENTONCES RIESGO ES Evaluable>

REGLA GP-R45

<SI APICAB-PROTOTIPO ES Aplicable Y FACTIB-PROTOTIPO ES Factible Y
 RIESGO ES Evaluable Y APICAB-ESPIRAL ES Aplicable
 ENTONCES CV-PROPU-GESTION ES Espiral>

REGLA R2

<SI CV-PROPU-REQUISITOS ES Espiral Y CV-PROPU-APLIC ES Espiral Y
 CV-PROPU-GESTION ES Espiral
 ENTONCES CV-PROPU-PROYECTO ES Espiral>

REGLA R15

<SI CV-PROPU-REQUISITOS ES Objetos Y CV-PROPU-APLIC ES Objetos Y
 CV-PROPU-GESTION ES Espiral
 ENTONCES CV-PROPU-PROYECTO ES Objetos>

REGLA R2

<SI CV-PROPU-REQUISITOS ES Espiral Y CV-PROPU-APLIC ES Espiral Y
 CV-PROPU-GESTION ES Espiral
 ENTONCES CV-PROPU-PROYECTO ES Espiral>

REGLA ER-R4

<SI DEFINICION-REQUISITOS ES Incierta Y DEFINICION-LIMITES ES Incierta Y
 GRADO-DE-CERTIDUMBRE-INICIO ES Parcialmente
 ENTONCES CV-PROPUESTO-REQUISITOS ES Espiral Y Objetos>

REGLA R15

<SI CV-PROPU-REQUISITOS ES Objetos Y CV-PROPU-APLIC ES Objetos Y
 CV-PROPU-GESTION ES Espiral
 ENTONCES CV-PROPU-PROYECTO ES Objetos>

REGLA R2

<SI CV-PROPU-REQUISITOS ES Espiral Y CV-PROPU-APLIC ES Espiral Y
 CV-PROPU-GESTION ES Espiral
 ENTONCES CV-PROPU-PROYECTO ES Espiral>

REGLA R24

<SI CV-PROPU-REQUISITOS ES Espiral Y CV-PROPU-APLIC ES Objetos Y

CV-PROPU-GESTION ES Espiral

ENTONCES CV-PROPU-PROYECTO ES Espiral>

REGLA R26

<SI CV-PROPU-REQUISITOS ES Objetos Y CV-PROPU-APLIC ES Espiral Y

CV-PROPU-GESTION ES Espiral ENTONCES CV-PROPU-PROYECTO ES Espiral>

REGLA R24

<SI CV-PROPU-REQUISITOS ES Espiral Y CV-PROPU-APLIC ES Objetos Y

CV-PROPU-GESTION ES Espiral

ENTONCES CV-PROPU-PROYECTO ES Espiral>

REGLA R26

<SI CV-PROPU-REQUISITOS ES Objetos Y CV-PROPU-APLIC ES Espiral Y

CV-PROPU-GESTION ES Espiral

ENTONCES CV-PROPU-PROYECTO ES Espiral>

CICLO DE VIDA RECOMENDADO

Se recomienda el Ciclo de Vida en Espiral

Especificación de Requerimientos recomienda: Espiral Objetos

Tipo de Aplicación recomienda: Espiral Objetos

Gestión de Proyecto recomienda: Espiral

9.5 RESULTADOS DE LA EVALUACIÓN DEL SISTEMA EXPERTO

El sistema implementado se probó con cuatro grupos de usuarios: los expertos que participaron en el desarrollo del sistema; profesionales que se dedican al desarrollo de proyectos de software; docentes que enseñan el tema en cátedras universitarias; y alumnos que usaran el sistema para la aplicación práctica de los conceptos de ciclo de vida. Se solicitó a cada grupo que analizara el sistema desde los siguientes aspectos:

- Corrección:** existencia de elementos redundantes, incompletos o inconsistentes.
- Validez:** la capacidad del sistema de tratar los casos y el cumplimiento de los requisitos.
- Usabilidad:** la amigabilidad del sistema para ser operado, el grado de facilidad de aprendizaje para operar el sistema.
- Utilidad:** este aspecto tiene que ver con los beneficios que aporta el sistema para la organización o institución que lo aplicará respecto de no hacerlo.

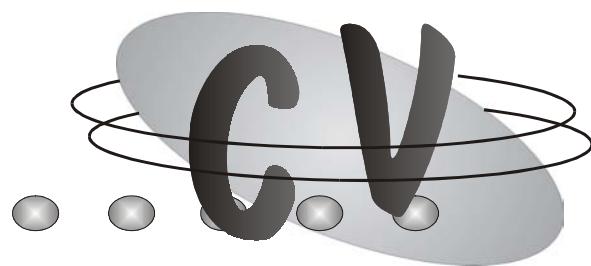
1. Expertos que participaron en el desarrollo del sistema:
 - Corrección:** no encontraron elementos redundantes, incompletos o inconsistentes.
 - Validez:** permitieron corregir algunos errores de codificación que afectaban al razonamiento del sistema. Particularmente los casos 4 y 5.
 - Usabilidad:** se redefinió alguna funcionalidad en el diseño de pantallas previsto originalmente
 - Utilidad:** participaron del proyecto porque estaban interesados desde el aspecto profesional para contrastar sus apreciaciones del proyecto en la selección del ciclo de vida, lo que se consideró como una mejora en la calidad y tiempos de trabajo en el desarrollo de sistemas informáticos. En su faceta docente para mejorar la explicación del tema ciclos de vida con la posibilidad de una aplicación práctica concreta, lo que consideran una mejora en la calidad de las clases y por ende para la imagen de la institución educativa.
2. Profesionales que se dedican al desarrollo de proyectos de software:
 - Corrección:** no encontraron elementos redundantes, incompletos o inconsistentes.
 - Validez:** permitieron corregir algunos errores de codificación que afectaban al razonamiento del sistema. Particularmente los casos 5 y 9.
 - Usabilidad:** se redefinió alguna funcionalidad en el diseño de pantallas previsto originalmente
 - Utilidad:** opinaron que el sistema les facilitaba recordar aspectos a analizar del proyecto, que en algunos casos se les pasaban por alto, y contrastar sus apreciaciones desde su propia experiencia con la propuesta del sistema para la selección del ciclo de vida. Se lo consideró como una mejora de recursos en tiempos, personal entrenado y calidad de resultados por la toma más adecuada de decisiones tempranas.
3. Docentes que enseñan el tema en cátedras universitarias:
 - Corrección:** no encontraron elementos redundantes, incompletos o inconsistentes.
 - Validez:** permitieron corregir algunos errores de codificación que afectaban al razonamiento del sistema. Particularmente los casos 4, 5 y 9.
 - Usabilidad:** permitió redefinir con mayor precisión, algunas de las preguntas de las áreas a evaluar.
 - Utilidad:** encontraron que el sistema facilitaba y lograba encontrar mayor especificidad para el tema ciclos de vida y desarrollo de proyectos. Permite ejercitación práctica concreta, pudiendo aprovechar mejor los casos reales de trabajo en la definición de casos de práctica. Consideraron una mejora significativa en la comprensión del tema y en la preparación de clases más interesantes para los alumnos.

4. Alumnos que usaran el sistema para la aplicación práctica de los conceptos de ciclo de vida:
 - Corrección:** no encontraron elementos redundantes, incompletos o inconsistentes.
 - Validez:** no encontraron errores que afectaran al razonamiento del sistema.
 - Usabilidad:** permitió redefinir con mayor precisión, algunas de las preguntas y opciones de las áreas a evaluar.
 - Utilidad:** consideraron una mejora significativa la posibilidad de realizar práctica para la mejor la comprensión del tema y para que las clases fueran más motivadoras e interesantes. Consideraron de mejor calidad la preparación del tema ya que la teoría les resulta en general mas aburrida y tenían que estudiar de memoria en lugar de comprender el tema. Además les permitió relacionar y comprender mejor la utilidad de una metodología de trabajo en la actividad profesional. También valoraron mejor el nivel del docente. Algunos pudieron armar casos de práctica en relación a alguna experiencia laboral previa que tenían y contrastar lo que el sistema proponía con lo que habían hecho.

El resultado de la evaluación se consideró altamente satisfactorio, por el experto principal y por el Ingeniero en conocimiento.

Capítulo 10

Conclusiones y Futuras Líneas de Investigación



El desarrollo de la tesis de magíster ha permitido profundizar los temas y las técnicas estudiadas durante el curso de Magíster. Este capítulo considera tres aspectos importantes en relación al trabajo realizado y a su continuidad. En la primera parte de Conclusiones se detallarán los aportes originales que el trabajo de tesis ha generado en el dominio de conocimiento de la Selección del Modelo de Ciclo de Vida más adecuado. En la segunda parte se mencionan aspectos y líneas de investigación que se pueden considerar para ampliar el presente desarrollo. En la tercera parte se mencionan temas de investigación complementarios.

10.1. CONCLUSIONES DEL TRABAJO

Constituyen aportes originales del presente trabajo:

- Una primer estructuración del conocimiento asociado al proceso de selección de un modelo de Ciclo de Vida, en tres áreas de análisis, Definición de los Requerimientos, Tipo de Aplicación y Gestión del Proyecto.
- La identificación y síntesis de las variables de análisis asociadas a cada una de las áreas definidas que definen la información requerida para proponer el Modelo del Ciclo.
- El desarrollo de un sistema decisorio sobre la base de los escenarios identificados.
- La modularidad del sistema de escenarios que permite la extensión de la selección a otros modelos de ciclos de vida.
- La modularidad de identificación de áreas que permite la extensión, agregado o subdivisión de otras áreas de análisis.
- La modularidad de identificación de las variables que permite la extensión de valores posibles para cada una de ellas.
- Aplicación y adaptación para este caso en particular de la técnica Método Delphi para adquirir información del grupo de expertos.
- Desarrollo de una herramienta de software que asista al Ingeniero de Software en el procesos de:
 - ✓ Analizar las variables y las opciones para cada una de ellas que identifican un proyecto de desarrollo de sistemas.

- ✓ Seleccionar el modelo de ciclo de vida más adecuado para cada área de análisis.
 - ✓ Seleccionar el modelo de ciclo de vida más adecuado para el proyecto.
 - ✓ Explicar el modelo de razonamiento aplicado para la selección sugerida. No se usa la opción del Kappa PC, ya que no resultaba fácil de comprender por usuarios no entrenados, por lo tanto se programó en la herramienta que junto con la solución sugerida se identificaran todas las reglas que fueron usadas en el razonamiento para obtener ese resultado.
 - ✓ Facilitar la congruencia en el proceso de decisión del modelo de ciclo de vida más adecuado, entre distintos proyectos de desarrollo de sistemas de una misma organización y entre distintos equipos de desarrollo de una misma organización.
- Desarrollo de un modelo de estructuración del conocimiento y de una herramienta de software que asista a los docentes en la explicación y práctica del tema en las cátedras universitarias.

10.2. LINEAS DE INVESTIGACIÓN Y DESARROLLO

A partir de los resultados obtenidos pueden plantearse las siguientes líneas de trabajo:

- Extender la lista de modelos de ciclo de vida seleccionables con otros modelos de Ciclo de vida: Prototipos, Proceso Unificado, Emisión Gradual, cualquier otro modelo que se deseé incluir.
- Extender, o redefinir los escenarios ya identificados incorporando variables para determinar las características del proyecto considerando técnicas como COCOMO, Puntos de Función, Marcos de Sistemas propuestos por Jackson, [Jackson, M. 1995], [Jackson, M. a 1999], [Jackson, M. b 1999].
- Completar el modelo de estructuración del conocimiento que define los criterios de selección de ciclo de vida más adecuado para un proyecto software dado.

- En el actual modelo desarrollado se han relacionado en el razonamiento, las variables de cada una de las áreas entre sí. Puede extenderse el razonamiento del modelo incorporando la incidencia de las relaciones, de las variables de las áreas entre el conjunto de áreas definidas.
- Determinar el aporte que la Ingeniería del Conocimiento puede realizar al proceso de automatizar la selección del modelo de ciclo de vida de software, considerando especialmente que el modelo debe ser evolutivo y, por lo tanto, debe permitir una actualización continua, que es una propiedad característica de los sistemas basados en conocimiento.

10.3. LINEAS DE INVESTIGACIÓN COMPLEMENTARIAS

A partir de dificultades que se presentaron al resolver el trabajo de tesis, se han considerado algunos temas complementarios como Líneas de Investigación Y Desarrollo para trabajos de Final de materias, de Especialidad o Tesis de Grado.

- Software que facilite el cálculo de la Viabilidad del proyecto, cuando se aplica el Test de Viabilidad propuesto en la bibliografía de referencia [Gómez, A. y otros 1997].
- Software para cálculo de las matrices y gráficos de la técnica de Emparrillado, con opción para usar distintas fórmulas de distancia, rangos de evaluación, metodologías y lenguajes de programación. Las experiencias que se han realizado son:
 - ✓ Describe en detalle el proceso de cálculo de la parrilla y algunas dificultades encontradas durante el desarrollo del trabajo, que están pendientes de ser completadas por versiones posteriores. El software sólo permite usar la fórmula de distancia lineal, calcula las matrices, permitiendo usar rangos de valoración entre 1 y 50 y realiza los gráficos. En este trabajo se han presentado varios casos resueltos por la herramienta. Se ha utilizado una metodología orientada a los datos y está programado en Visual Basic.
Estado: Tesis de Grado presentada. Licenciatura en Sistemas de Información. Universidad Nacional de Luján. 2000. Tesista: Lic. Enrique Fernández, Directora: Lic. Bibiana Rossi.

- ✓ El software permite usar la fórmula de distancia lineal, calcula las matrices permitiendo usar rangos de valoración entre 1 y 999 y realiza los gráficos. Se ha desarrollado siguiendo la metodología Orientada a objetos Proceso Unificado y Notación UML, se utilizo para la documentación la herramienta CASE Rational Rose y se programó en Microsoft Java.

Estado: Trabajo final presentado. Materia Programación Orientada a Objetos. Licenciatura en Sistemas de Información. Universidad Nacional de Luján. 2001. Alumno A.S. Marcelo Castro, Directora: Lic. Bibiana Rossi.

- ✓ El software permite usar distintas fórmulas de distancia, calcula las matrices permitiendo usar rangos de valoración entre 1 y 999 y realiza los gráficos. Se ha desarrollado siguiendo la metodología Orientada a Objetos Proceso Unificado y Notación UML. Aún no se ha decidido el lenguaje de programación a usar.

Estado: Tesis de Grado en curso. presentada. Universidad de Buenos Aires, Facultad de Ingeniería. 2001. Tesista: Sr. Emiliano Castañeda, Directora: Lic. Bibiana Rossi.

Capítulo 11

Bibliografía



En este capítulo se detalla la bibliografía usada en el presente trabajo y las abreviaturas.

11. 1 REFERENCIAS BIBLIOGRAFICAS

Böehm, B.W., “*Software Engineering Economics*”. Prentice-Hall, Englewood Cliffs. Nueva Jersey, 1981.

Böehm, B.W., “*A Spiral Model of Software Development and Enhancement*”. Software Engineering Project Management, IEEE Computer Society, pp 128-142, 1987.

Booch, G., “*Análisis y Diseño Orientado a Objetos con Aplicaciones*”. Addison-Wesley, Wilmington, Delaware, 1996.

Booch, G., Rumbaugh, J. y Jacobson, I.: *El Lenguaje Unificado de Modelado*. Addison Wesley, 1999.

Burch, J.G. y Grudnitski, G., “*Diseño de Sistemas de Información*”. Editorial Limusa, Noriega editores, Megabyte. 1994.

Charette, R.N., “*Building Bridges over Intelligent Rivers*”. American Programmer, vol. 5, N° 7, pp 2-9, 1992.

Chen, P.S., “*The Entity-Relationship Model: Toward a Unifying View of Data*”. TODS, 1. 9-36.1976.

Gómez, A., Juristo, N., Montes, C., Pazos, J. “*Ingeniería del Conocimiento*”. Centro de Estudios Ramón Areces. S.A., Madrid, 1997

Hawryszkiewycz, I.T., “*Introducción al Análisis y Diseño de Sistemas con ejemplos prácticos*”. Anaya, Madrid, 1990.

IEEE, “*Standard for Developing Software Life Cycle Processes*”. IEEE Std|. 1074-1991, Nueva York, IEEE Computer Society, 1991.

Ierache, J.S., “*Métricas en sistemas orientados a objetos*”. Cuaderno de reportes técnicos. CAPIS, Instituto Tecnológico de Buenos Aires, Argentina, 1999.

ISO, "ISO/IEC 12701-1 Software life-cycle process", 1994.

Jackson, M., "Software Requirement & Specification". ACM Press, Addison-Wesley, 1995.

Jackson, M. a, "Problem Analysis Using Small Problem Frames", South African Computer Journal 22; Special Issue on WOFACS'98, pp 47-60, 1999.

Jackson, M. b, "Problem Analysis and Structure", Keynote Talk at ITG/SEV Symposium. Zürich, 29 september 1999.

Jacobson, I. et al., "Object Oriented Software Engineering. A Use Case Driven Approach". ACM Press, Addison-Wesley, Workingham 1992.

Jacobson, I., Booch, G., Rumbaugh, J., "The Unified Software Development Process". Addison-Wesley, 1999.

Juristo Juzgado, N. a, "Introducción a la Ingeniería de software". Máster en Ingeniería del Software e Ingeniería del Conocimiento, unidad 1, Madrid, 1996.

Juristo Juzgado, N. b, "Proceso de Construcción de Software y Ciclos de vida". Máster en Ingeniería del Software e Ingeniería del Conocimiento, unidad 2, Madrid, 1996.

Macro, A.. "Software Engineering concepts and management". Prentice-Hall, Nueva York, 1990.

Martin, J. y Odell, J., "Análisis y Diseño Orientado a Objetos". Prentice-Hall. 1994.

Paulk, M. et al., "Capability Maturity Model for Software", Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 1993.

Piattini, M.G. y Daryanani, S.N., Elementos y Herramientas en el Desarrollo de Sistemas de Información. Ra-ma 1995.

Piattini, M.G., Calvo-Manzano, J., Cervera, J. y Fernández, L., "Análisis y Diseño detallado de Aplicaciones Informáticas de Gestión". Rama, Madrid, 1996.

Pressman, R.S., “*Ingeniería del Software Un Enfoque Práctico*”, Mc Graw Hill, Madrid, 1997

Rincón, A., Plágaro, J., “*Diccionario conceptual de Informática y Comunicaciones*”. Paraninfo, Madrid, 1998

Rossi, B., Britos, P., García Martínez, R., “*Modelado de Objetos*”. Revista N° 21 del Instituto Tecnológico de Buenos Aires, Argentina, 1998.

Royce, W.W., “*Managing the development of large Software Systems: concepts and Techniques*”. Proceedings, Wescon, 1970.

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F. y Lorensen, W., “*Modelado y Diseño Orientado a Objetos*”. Prentice Hall, España. 1996.

Whitten Jeffrey L., Bentley Lonnie D. y Barlow Victor M.: *Análisis y Diseño de Sistemas de Información, 3era. edición*. Editorial IRWIN. Madrid, 1996.

11.2 ABREVIATURAS

CASE: Computer Aided Software Engineering

CV: Ciclo de Vida

DFD: Diagrama de Flujo de Datos

HW: Hardware

IC: Ingeniero del Conocimiento

INCO: Ingeniería del Conocimiento

IS: Ingeniería del Software

OO : Orientación a Objetos

OOSE: Object Oriented Software Engineering

SE: Sistema Experto

SEI: Software Engineering Institute

SW: Software

UML: Unified Modelling Language – Lenguaje Unificado de Modelado

Capítulo 12

Anexos



En este capítulo se detallan los anexos. En primer lugar se presentan los anexos del capítulo 3, Definición del problema: el control de configuración. En segundo lugar se presentan los anexos del capítulo 5, Adquisición de Conocimientos: las sesiones de entrevista realizadas al experto y las sesiones de extracción de conocimientos realizadas sobre el análisis de la bibliografía del dominio de aplicación del presente trabajo.

12.1 ANEXOS DEL CAPITULO 3: CONTROL DE CONFIGURACIÓN

Las actividades de Gestión de Configuración para este proyecto son:

- ◆ Identificación de la Configuración
- ◆ Control de la Configuración
- ◆ Generación de Informes de Estado

◆ **Identificación de la Configuración:**

Nombre elegido para la aplicación: SECV

Objetivo de la Aplicación: Asistir al líder de un proyecto de desarrollo de Software en la selección del modelo de ciclo de vida mas apropiado según las características particulares de ese proyecto.

Ciclo de vida del Software: Modelo Troncocónico

Fases del Ciclo de vida: se enumeran las fases para este proyecto y se determinan los productos que se espera obtener al final de cada una de ellas.

1. Identificación de la tarea

- Documento que describe el objetivo, alcance, ámbito, y metodología a aplicar en el desarrollo del SECV.
- Plan de actividades.
- Estudio de Viabilidad del proyecto.

Estos productos servirán de base para la redacción del capítulo 3 y 4 del presente trabajo de tesis.

2. Desarrollo de prototipos

- Trascripción de las Entrevistas con el Experto
- Tablas de Ventajas / Desventajas de los Modelos de Ciclos de vida
- Cuestionario Método Delphi sesión I
- Resultados Método Delphi, sesión I
- Cuestionario Método Delphi sesión II
- Resultados Método Delphi, sesión II
- Cuestionario Método Delphi sesión III

- Resultados Método Delphi, sesión III
- Glosario de términos
- Tabla de Concepto-Atributo-Valor
- Diagrama Relacional de Conceptos
- Determinación y Resultados del Emparrillado
- Diccionario de Conceptos
- Modelo de conocimientos estratégicos
- Modelo de conocimientos tácticos – seudorreglas
- Modelo de conocimientos fácticos
- Casos de prueba
- Modelo Dinámico
- Mapa de Conocimientos
- Definición de Marcos
- Definición de Procedimientos
- Definición de Reglas de Producción
- Codificación de Marcos, Procedimientos y Reglas en Kappa PC

Estos productos servirán de base para la redacción del capítulo 5, 6 y 7 del presente trabajo de tesis.

3. Ejecución de la Construcción del Sistema Integrado

- Casos de prueba presentados por el experto
- Sistema funcionando en Kappa PC
- Informe final del resultado de la prueba del sistema
- Manual de usuario del sistema

Estos productos servirán de base para la redacción del capítulo 8 y 9 del presente trabajo de tesis.

4. Actuación para conseguir el mantenimiento perfectivo

- Nuevas versiones de documentos ya presentados en etapas anteriores con actualizaciones.
- Nuevas versiones de la aplicación

5. Lograr una adecuada transferencia tecnológica

- Material de capacitación en el uso del sistema, casos de ejemplo

6. Preparación de la Tesis de Magíster

- Documento de tesis para evaluar
- Correcciones del documento de tesis
- Documento final de la tesis
- Versión definitiva de la aplicación a los efectos del trabajo de tesis.

➤ **Líneas Bases establecidas:**

A los efectos del presente trabajo se ha realizado una simplificación tanto de las líneas base como de los elementos de configuración definidos. En este caso el criterio adoptado es, que dado que es una sola persona, el tesista, quien realiza la documentación del proyecto y la programación se ha definido una sola línea base para todo el proceso de Desarrollo y Programación y una línea base para el Mantenimiento Perfectivo.

1. Línea Base Desarrollo, Codificación y Evaluación: comprende las fases

- Identificación de la tarea,
- Desarrollo de prototipos,
- Ejecución de la Construcción del Sistema Integrado,
- Lograr una adecuada transferencia tecnológica y
- Preparación de la Tesis de Magíster

Los Elementos de configuración que comprende son todos los productos obtenidos en las fases mencionadas.

2. Línea Base Mantenimiento Perfectivo: comprende la fase de

- Actuación para conseguir el mantenimiento perfectivo

Los Elementos de configuración que comprende son todos los productos obtenidos en la fase mencionada.

➤ **Nomenclatura de los elementos del Producto:**

Todos los nombres de los elementos que componen el Producto, se encuentran sujetos a la nomenclatura que se detalla a continuación:

Caracteres 1-2: Valor fijo “CV”, que identifica al Producto Sistema Experto para la selección del Ciclo de Vida

Caracteres 3: Identifica a la línea base. Sus valores posibles son

- DCE: Desarrollo, Codificación y Evaluación
- MP: Mantenimiento Perfectivo

Caracteres 4-7: Identificación numérica del elemento

Carácter 8: Identificación de la versión del elemento

◆ **Control de Configuración:**

Se implementa el siguiente mecanismo para el control de cambios:

➤ **Generación de una solicitud de cambio**

Ante el requerimiento de un cambio funcional o un reporte de error, se completa la correspondiente solicitud.

➤ **Ingreso de la solicitud a la Base de Datos de cambios**

Una vez recibida la solicitud de cambio, se la ingresa en la Base de Datos de cambios.

➤ **Análisis de la solicitud de cambio**

Cada solicitud de cambio debe ser analizada por el Ingeniero en Conocimiento, conjuntamente con el Comité de Control de Cambios, y decidir si se rechaza o se acepta el cambio. La decisión tomada por el Comité queda registrada en la Base de Datos de Cambios.

➤ **Evaluación de la solicitud de cambio**

Si se decide la aceptación de la solicitud de cambio, el Ingeniero en Conocimiento, debe realizar la evaluación técnica de la misma, emitiendo un informe en donde se exprese el esfuerzo requerido para satisfacer el pedido, las repercusiones que dicho cambio genera en otros elementos y el costo estimado. La evaluación realizada, queda registrada en la Base de Datos de Cambios.

➤ **Generación de la orden de cambio**

El informe generado durante la evaluación de la solicitud de cambio, se somete al análisis del Comité de Control de Cambios, el cual le asigna la prioridad y los recursos necesarios. Se emite una Orden de Cambio

➤ **Realización del cambio**

Se realiza el cambio. Seguimiento y control de la modificación

➤ **Prueba e implementación del cambio**

Se certifica que el cambio funciona correctamente y se procede a su implementación, a través de la modificación de manuales y documentos que deban reflejar el cambio.

♦ **Generación de Informes de Estado:**

➤ **Registro de solicitudes de cambio**

<i>Consultora BIROSSI</i>	<i>Nro. Solicitud:.....</i>
Solicitud de cambios	
Producto : SECV Fecha: dd/mm/aaaa Responsable del Pedido: Area / Sector / Empresa : Recibido por: Fecha: dd/mm/aaaa Analizado por: Fecha: dd/mm/aaaa ACEPTADO / RECHAZADO	
Descripción del cambio solicitado <hr/> <hr/> <hr/>	
Solución Propuesta <hr/> <hr/> <hr/>	
Elementos del Producto afectados por el cambio <hr/> <hr/> <hr/>	
Documentación anexa <hr/> <hr/> <hr/>	
Hoja : 1	

Consultora BIROSSI

Nro. Solicitud:.....

Solicitud de cambios***Estimación del cambio******Solución Propuesta por el equipo técnico***

Tiempo evaluado en horas / hombre por perfil:

Perfil	Estimación en Horas
<i>Ing. Conocimiento</i>	
<i>Programador</i>	
.....	
.....	

Costo del cambio:

Perfil	Estimación en Horas	Costo x hora	Costo total
<i>Ing. Conocimiento</i>			
<i>Programador</i>			
.....			
.....			
TOTAL			

Cronograma de desarrollo e implementación:

Plan de Pruebas:

Hoja : 2

➤ **Orden de cambio**

Consultora BIROSSI	Nro. Solicitud:
Orden de cambio	
Producto : SECV	Fecha: dd/mm/aaaa
Responsable del Pedido:	
Area / Sector / Empresa :	
Recibido por:	Fecha: dd/mm/aaaa
Analizado por:	Fecha: dd/mm/aaaa
Descripción del cambio Solicitado <hr/> <hr/> <hr/> <hr/> <hr/>	
Solución a implementar <hr/> <hr/> <hr/> <hr/> <hr/>	
Elementos del Producto afectados por el cambio <hr/> <hr/> <hr/> <hr/> <hr/>	
Hoja : 1	

Consultora BIROSSI

Nro. Solicitud:.....

Orden de cambio**Documentación anexa**

Estimación del cambio**Tiempo evaluado en horas / hombre por perfil:**

Perfil	Estimación en Horas	Nombre Funcionario
Ing. Conocimiento		
Programador		
.....		
.....		

Responsable del cambio:**Comentarios:**

Fecha de entrega:**Hoja : 2**

➤ **Registro en la Base de Datos de Cambios**

Consultora BIROSSI

Registro de Cambios

Producto	: SECV
Solicitud Número	:
Responsable del Pedido	:
Area / Sector / Empresa	:
Recibido por	:
Fecha de recepción	: dd/mm/aaaa
Descripción breve	: _____

Analizado por	:
Fecha del análisis	: dd/mm/aaaa
Decisión	: (Aprobación / Rechazo)
Comentarios	: _____

Prioridad asignada	:
Estado actual del pedido	:

Fecha de terminación	: dd/mm/aaaa
Funcionarios involucrados	: _____

Hoja : 1

➤ **Registro de Instalaciones**

Consultora BIROSSI

Registro de Instalaciones

Producto : SECV
Solicitud Número :
Descripción breve : _____

Responsable :

Fecha de terminación : dd/mm/aaaa

Lugar de instalación :

Fecha de instalación : dd/mm/aaaa

Versión instalada :

Hoja : 1

➤ **Informe de Estado de los cambios**

Consultora BIROSSI

Fecha: dd/mm/aaaa

Informe de Estado de los Cambios**Producto: SECV****Desde Fecha: dd/mm/aaaa****Hasta Fecha: dd/mm/aaaa**

Solicitud Número	Fecha Recep.	Descripción	Prioridad	Estado Actual
.....	dd/mm/aaaa
.....	dd/mm/aaaa
.....	dd/mm/aaaa
.....	dd/mm/aaaa
.....	dd/mm/aaaa
.....	dd/mm/aaaa
.....	dd/mm/aaaa
.....	dd/mm/aaaa
.....	dd/mm/aaaa
.....	dd/mm/aaaa
.....	dd/mm/aaaa
.....	dd/mm/aaaa

Hoja : 1

➤ **Informe de Instalaciones**

Consultora BIROSSI

Fecha: dd/mm/aaaa

Informe de Instalaciones

Producto: SECV

Desde Fecha: dd/mm/aaaa

Hasta Fecha: dd/mm/aaaa

Hoja : 1

12.2 ANEXOS DEL CAPITULO 5: SESIONES DE ADQUISICIÓN DE CONOCIMIENTOS

- El grupo de sesiones identificadas con A.x se corresponde con las entrevistas realizadas al experto.
- El grupo de sesiones identificadas con B.x se corresponde con las sesiones de extracción de conocimiento, particularmente del análisis de la bibliografía.

A.1 SESIÓN I

A.1.1 Preparación de la Sesión I

- *Información a tratar:* Primera aproximación a la tarea y a su problemática. Establecer ámbito, alcances y objetivo del sistema experto a desarrollar.
- *Amplitud y Profundidad:* Establecer el ámbito general de desarrollo de la tarea sin detallar ningún caso específico.
- *Técnica utilizada:* Entrevista no estructurada.
- *Preparación de Preguntas:*
 - ¿En qué momento del desarrollo de un sistema de información se selecciona el ciclo de vida del proyecto?
 - ¿Quién lleva a cabo la tarea?
 - ¿Qué dificultades tiene la tarea?
 - ¿En qué aspecto serviría de apoyo el sistema experto?

A.1.2 Realización de la Sesión I

La entrevista se realiza en la oficina del experto, habiendo acordado previamente la hora de inicio y fin de la entrevista. Se le explica al experto el objetivo de la entrevista y el tipo de preguntas que se van a realizar y que se tomará nota de sus respuestas. El entrevistado se muestra muy dispuesto. Se desarrolla la sesión de la entrevista.

A.1.3 Trascipción de la Sesión I

Entrevista realizada el 20 de octubre de 1999

Experto: Dr. Gregorio Perichinsky

Ingeniero del Conocimiento: Bibiana Rossi

Lugar: oficina del experto

Tiempo: 14 a 15.30 hs.

Objetivos: Establecer alcances y objetivos del proyecto

IC. *¿Cuáles son las primeras actividades a llevar a cabo, en términos generales en el desarrollo de un proyecto informático?*

E. Si se trabaja con criterios metodológicos las primeras fases son la Definición de los Requerimientos y la Planificación o calendarización del proyecto y el Análisis del sistema actual y/o futuro.

IC. ¿En qué fase se incluye la actividad de la selección del ciclo de vida más adecuado para el proyecto?

E. Esto puede variar según el criterio del líder a cargo del proyecto, puede ser un tópico en la especificación de los requerimientos, o puede incluirlo como una de las primeras actividades del Análisis del Sistema. Lo que es cierto es que no es conveniente iniciar el análisis del sistema sin haber definido el modelo de ciclo de vida a utilizar. Yo diría que como momento más tardío debe ser la primer actividad de la fase de análisis. En mi caso particular suelo hacerlo antes de calendarizar el proyecto ya que me facilita considerablemente el armado del plan.

IC. ¿Existe algún tipo de dificultad en la actividad de selección del ciclo de vida?

E. Unas cuantas, y podría decir que hay dificultades propias de la tarea de selección y dificultades que son particulares cuando es necesario enseñar este tema a los estudiantes de informática en la universidad.

IC. ¿En qué consisten las dificultades propias de la tarea de selección?

E. Intentaré puntualizar las más conflictivas:

- Existen diversos modelos de ciclo de vida entre los que realizar la selección: cascada, prototipado de usar y tirar, incremental, emisión gradual, mejora iterativa, ensamblaje de componentes, espiral, prototipado operativo, prototipado rápido, etc. No existe un modelo de ciclo de vida que funcione para cualquier proyecto. Desde la presentación del ciclo de vida en cascada hasta el presente se han presentado un promedio de 30 ciclos de vida posibles, que yo conozco, para llevar adelante un proyecto. Esto implica conocerlos, reconocer cuales son los aplicables y tenerlos presentes en el momento de la selección.
- Es necesario tener en cuenta un con amplio conjunto de variables que representan las características particulares del proyecto. El análisis de estas variables se complica, ya que se determinan con relación a las ventajas y desventajas de aplicación de los distintos ciclos de vida y en relación con el proyecto en particular. Este proceso de que variables considerar es un proceso fuertemente relacionado con la experiencia personal del líder de proyecto. Un líder de proyecto que se inicia carece de experiencia suficiente para conocer todos los ciclos de vida posibles y mucho más aun cuáles son los aspectos del proyecto que debe considerar. Y la única manera de adquirir esa experiencia es haciendo selecciones, con la cascada de errores que implica en el desarrollo de un sistema las equivocaciones iniciales.
- La actividad se realiza al inicio del proyecto. Suponiendo que el líder de proyecto tenga la suficiente experiencia para conocer al menos 10 ciclos de vida y que pueda definir las variables a considerar, es probable que no pueda cuantificarlas apropiadamente porque al inicio del proyecto muchas de ellas son solo estimaciones. Con el avance del proyecto se puede recién tener mas precisión respecto de esas características.

IC. ¿En qué consisten las dificultades al enseñar el tema a los estudiantes?

E. En la bibliografía aparecen explicados con bastante claridad los ciclos de vida, pero en cuanto a cuales son las características del proyecto a tener en cuenta, la bibliografía es ambigua. Para explicarme con mayor claridad: en la bibliografía se menciona que es

necesario tener en cuenta las políticas de la empresa, la cultura de la organización, la disponibilidad para correr riesgos, la volatilidad y comprensión de los requisitos, el dominio de aplicación, la complejidad del proyecto, etcétera. Es cierto que hay que tener en cuenta todo esto pero así explicado es muy ambiguo, ¿cómo se tiene en cuenta la política de la empresa?, o ¿qué se mide para saber la complejidad del proyecto? En este tema el conocimiento no está sistemáticamente organizado y no se especifica cómo debe realizarse el proceso de selección. Por esto, la tarea es muy difícil de explicar para los alumnos y para un profesional sin experiencia.

IC. ¿Quiénes son los encargados de seleccionar el ciclo de vida?

E. El líder de proyecto o quienquiera que cumpla esa función en el desarrollo de un proyecto informático.

IC. ¿Cuál puede ser entonces el apoyo que preste el sistema experto a desarrollar?

E. Entiendo que el mayor apoyo está en la especificación del conjunto de variables que representan las características particulares del proyecto de desarrollo, que es necesario considerar para seleccionar el ciclo de vida.

IC. ¿Quiénes utilizarían en su tarea el sistema experto a desarrollar?

E. Los que tengan la responsabilidad de liderar el desarrollo de sistemas informáticos, o sea los líderes de proyecto y los docentes de las cátedras de análisis y diseño de sistemas, por supuesto los alumnos que cursen esas materias para su práctica.

A.1.4 Análisis de la Sesión I

- **Conocimientos extraídos:**

Los conocimientos extraídos de la sesión I se encuentran reflejados en el capítulo III, Definición del problema y en el capítulo IV, Estudio de Viabilidad.

- **Ubicación de la tarea:**

- Las primeras fases son la Definición de los Requerimientos y la Planificación o calendarización del proyecto y el Análisis del sistema actual y/o futuro.
- No es conveniente iniciar el análisis del sistema sin haber definido el modelo de ciclo de vida a utilizar. Yo diría que como momento más tardío debe ser la primer actividad de la fase de análisis. En mi caso particular suelo hacerlo antes de calendarizar el proyecto ya que me facilita considerablemente el armado del plan.

- **Dificultades:**

- Existen diversos modelos de ciclo de vida entre los que realizar la selección: cascada, prototipado de usar y tirar, incremental, emisión gradual, mejora iterativa, ensamblaje de componentes, espiral, prototipado operativo, prototipado rápido, etc. No existe un modelo de ciclo de vida que funcione para cualquier proyecto.

- Es necesario tener en cuenta un con amplio conjunto de variables que representan las características particulares del proyecto. El análisis de estas variables se complica, ya que se determinan con relación a las ventajas y desventajas de aplicación de los distintos ciclos de vida y en relación con el proyecto en particular. Este proceso de qué variables considerar es un proceso fuertemente relacionado con la experiencia personal del líder de proyecto.
 - La actividad se realiza al inicio del proyecto, es probable que no sea posible cuantificarlas apropiadamente porque al inicio del proyecto muchas de ellas son sólo estimaciones. Con el avance del proyecto se puede recién tener mas precisión respecto de esas características.
 - En la bibliografía se menciona que es necesario tener en cuenta las políticas de la empresa, la cultura de la organización, la disponibilidad para correr riesgos, la volatilidad y comprensión de los requisitos, el dominio de aplicación, la complejidad del proyecto, etcétera. En este tema el conocimiento no está sistemáticamente organizado y no se especifica cómo debe realizarse el proceso de selección.
- *Apoyo del SE a desarrollar:*
 - Entiendo que el mayor apoyo está en la especificación del conjunto de variables que representan las características particulares del proyecto de desarrollo, que es necesario considerar para seleccionar el ciclo de vida.
 - *Responsable de la tarea:*
 - Los que tengan la responsabilidad de liderar el desarrollo de sistemas informáticos, o sea los líderes de proyecto y los docentes de las cátedras de análisis y diseño de sistemas, por supuesto los alumnos que cursen esas materias para su práctica.
 - *Conocimientos a educir en próximas sesiones:*
 - ¿Es posible establecer una jerarquía de importancia entre los ciclos de vida?
 - ¿Cuál puede ser un buen punto de partida para identificar las variables que definen las características particulares de un proyecto de desarrollo?
 - ¿Qué material bibliográfico recomienda para el proceso de extracción de conocimientos?

A.1.5 Evaluación de la Sesión I

- *¿Se han logrado los objetivos?*

Si se han cumplido los objetivos ya que se obtuvo la información prevista, sobre los objetivos, se determinó en gran medida los alcances y ámbito del

proyecto, como así también se obtuvo información para el estudio de viabilidad.

- *¿Es necesario volver sobre lo mismo?*

Sí es necesario para:

- Definir el alcance del prototipo respecto de los ciclos de vida que incluirá.
- Obtener orientación respecto del material bibliográfico más conveniente para el proceso de extracción de conocimientos.
- Analizar cómo es más conveniente encarar la dificultad que presenta la falta de sistematización en el conocimiento respecto de las variables que identifican las características de un proyecto.

- *Número y tipo de sesiones para cubrir el área*

No es posible establecer en este momento el número de sesiones necesarias, hasta no tener mayor precisión sobre los alcances del sistema.

A.2 SESIÓN II

A.2.1 Preparación de la Sesión II

- *Información a tratar:* Alcance del sistema experto a desarrollar. Características que intervienen en el estudio de viabilidad del proyecto.
- *Amplitud y Profundidad:* Precisar los alcances del sistema, analizar el grado de las dificultades, precisar la información para completar el estudio de viabilidad.
- *Técnica utilizada:* Entrevista estructurada.
- *Preparación de Preguntas:*

¿Es posible establecer una jerarquía de importancia entre los ciclos de vida?

¿Cuál puede ser un buen punto de partida para identificar las variables que definen las características particulares de un proyecto de desarrollo?

¿Qué material bibliográfico recomienda para el proceso de extracción de conocimientos?

De los proyectos en los que ha participado, ¿existe documentación que pueda ser utilizada para probar y evaluar el sistema?

¿Para seleccionar el CV es necesario utilizar el sentido común?

¿Para seleccionar el CV se requiere un alto nivel de abstracción?

¿Para realizar la selección del CV qué pasos intermedios es necesario cumplir?

¿Es conveniente justificar el CV que se ha seleccionado?

¿El sistema debe buscar la solución óptima?

A.2.2 Realización de la Sesión II

La entrevista se realiza en la oficina del experto, habiendo acordado previamente la hora de inicio y fin de la entrevista. Se le explica al experto el objetivo de la entrevista, las preguntas que se van a realizar y que se tomará nota de sus respuestas. Se desarrolla la sesión de la entrevista.

A.2.3 Transcripción de la Sesión II

Entrevista realizada el 26 de octubre de 1999

Experto: Dr. Gregorio Perichinsky

Ingeniero del Conocimiento: Bibiana Rossi

Lugar: oficina del experto

Tiempo: 17 a 18.30 hs.

Objetivos: a) Completar alcances y objetivos del proyecto
b) Identificación de la viabilidad

IC. *Usted ha mencionado que existe un promedio de 30 ciclos de vida, quizás algunos más. ¿Es posible establecer una jerarquía de importancia?*

E. Si es muy posible hacerlo ya que en varios casos muchos de ellos son similares en su esencia básica y se diferencian en alguna fase o en algún detalle. Si tuviera que agruparlos genéricamente de alguna forma, se puede decir que existen los modelos con filosofía de cascada, los modelos con filosofía de prototipo, los modelos con filosofía de orientación a objetos y quizás algunos modelos menos usados como el modelo en espiral.

IC. *¿Cuál le parece que puede ser un buen punto de partida para identificar las variables que representan las características del proyecto que son necesarias conocer para seleccionar el ciclo de vida?*

E. Entiendo que un buen comienzo y casi diría imprescindible es partir de una análisis de la bibliografía existente para después consensuar con la opinión y experiencia de otros profesionales.

IC. *¿A qué otros profesionales consultaría?*

E. A los docentes de las cátedras de análisis y diseño, que trabajan en la empresa como líderes de proyecto y además tienen la dificultad de enseñar el tema y a algunos profesionales reconocidos con quien he trabajado en algunos proyectos de envergadura.

IC. *¿Es posible contactarlos para invitarlos a participar y ver si es posible comprometerlos para algunas entrevistas? Estoy pensando particularmente en entrevistas de tipo grupal con ellos.*

E. Por supuesto, yo me pondré en contacto para avisarles del proyecto y cuando acordemos les aviso de la entrevista.

IC. *¿Qué material bibliográfico recomienda para el proceso de extracción de conocimientos?*

E. El material que recomiendo es el siguiente:

- A Spiral Model of Software Development and Enhancement Software Engineering Project Management, de Boehm,
- Análisis y Diseño Orientado a Objetos con Aplicaciones de Booch,
- Estándares del IEEE y de ISO.
- Análisis y Diseño Orientado a Objetos de Martin y Odell.
- Elementos y Herramientas en el Desarrollo de Sistemas de Información de Piattini
- Análisis y Diseño detallado de Aplicaciones Informáticas de Gestión de Piattini
- Ingeniería del Software Un Enfoque Práctico de Pressman
- Modelado y Diseño Orientado a Objetos de Rumbaugh

IC. *De los proyectos en los que ha participado, ¿existe documentación que pueda ser utilizada para probar y evaluar el sistema?*

E. Si es posible utilizar la documentación existente como así también los prácticos ya preparados por los docentes de las cátedras de análisis y diseño de sistemas.

IC. *¿Para seleccionar el CV es necesario utilizar el sentido común?*

E. Sí por sentido común se entiende experiencia, la experiencia si es necesaria. En realidad se requiere de experiencia previa tanto para reconocer qué variables es necesario analizar como así también para estimar aquellas que puedan ser relativamente inciertas al inicio de un proyecto, pero que es necesario tenerlas en cuenta para la selección del ciclo de vida.

IC. *¿Para seleccionar el CV se requiere un alto nivel de abstracción?*

E. No mucho, es necesario tener presentes las características a evaluar en términos generales, y precisar su valor. Es posible que sea necesario reconocer alguna característica particular de algún proyecto que no se haya evaluado previamente. O incorporar a la evaluación presente alguna característica que haya resultado importante en proyectos previos. Justamente por eso es importante la experiencia.

IC. *¿Para realizar la selección del CV qué pasos intermedios es necesario cumplir?*

- a) Tener claros los requerimientos y el objetivo del proyecto
- b) Conocer la lista de variables que caracterizan el problema
- c) Analizar si es necesario agregar alguna variable nueva
- d) Asignarle un valor a las variables de acuerdo al proyecto a desarrollar
- e) Analizar cual es el ciclo de vida que responde mas adecuadamente a las características del proyecto.

IC. *¿Es conveniente justificar el CV que se ha seleccionado?*

E. Es muy necesario no solo para justificar la selección del proyecto presente en desarrollo sino también para contar con mecanismos que permitan a posteriori la revisión de los criterios de selección, sobre todo en aquellos casos donde la elección no resultó acertada.

IC. *¿El sistema debe buscar la solución óptima?*

E. No existe una solución optima, siempre es necesario una solución de compromiso. El solo hecho de tener sistematizada el análisis de las características del proyecto ya optimiza considerablemente la forma actual en la que se realiza la selección.

A.2.4 Análisis de la Sesión II

- *Conocimientos extraídos:*

Los conocimientos extraídos de la sesión II se encuentran reflejados en el capítulo III, Definición del problema y en el capítulo IV, Estudio de Viabilidad.

- *Conocimientos a educir en próximas sesiones:*

- Revisar con el experto cada una de las dimensiones de plausibilidad, justificación, adecuación y éxito del Test de Viabilidad. El experto conoce ampliamente de informática así que es factible explicarle brevemente el Test y repasar cada una de las características conjuntamente con él.
- ¿Cuál es la forma más adecuada para el análisis de la bibliografía?

A.2.5 Evaluación de la Sesión II

- *¿Se han logrado los objetivos?*

Si se han cumplido, se determinó los alcances y ámbito del proyecto, como así también se obtuvo información para el estudio de viabilidad.

- *¿Es necesario volver sobre lo mismo?*

Sí es necesario para:

- Revisar definitivamente las dimensiones del Test de Viabilidad.
- Obtener orientación precisa de cómo realizar el proceso de extracción de conocimientos.

- *Número y tipo de sesiones para cubrir el área*

Es necesario establecer dos tipos de sesiones. Una serie de sesiones con el experto y otro conjunto de sesiones con el grupo de expertos a quien desea dar participación el experto principal.

En cuanto a la serie de sesiones con el experto se estima una sesión para completar la información para la viabilidad del proyecto y otra sesión para preparar las entrevistas con el grupo.

En cuanto al grupo de expertos es necesario organizar la adquisición de conocimientos aplicando las técnicas adecuadas para un equipo de expertos. Despues de la segunda sesión prevista con el experto principal para preparar la adquisición de conocimientos con el grupo de expertos, será posible estimar el número de sesiones necesarias.

A.3 SESION III

A.3.1 Preparación de la Sesión III

- *Información a tratar:* Dimensiones del Test de Viabilidad. Extracción de conocimientos. Adquisición de conocimientos con el grupo de expertos.

- *Amplitud y Profundidad:* Necesarias para completar el Test de Viabilidad.
- *Técnica utilizada:* Entrevista estructurada para Test de Viabilidad y abierta para proceso de extracción de conocimientos y proceso de adquisición de conocimientos con el grupo de expertos.
- *Preparación de Preguntas:* Proponer la técnica de Método Delphi.
Revisar cada una de las características de las tablas de las dimensiones del Test de Viabilidad.
¿Al analizar la bibliografía cuáles son los conceptos o términos que es necesario identificar?

A.3.2 Realización de la Sesión III

La entrevista se realiza en la oficina del experto, habiendo acordado previamente la hora de inicio y fin de la entrevista. Se le explica al experto los temas a tratar. Se inicia la sesión con una breve explicación del Test de Viabilidad.

A.3.3 Resumen de la Sesión III

<i>Entrevista realizada el 31 de octubre de 1999</i>
Experto: Dr. Gregorio Perichinsky
Ingeniero del Conocimiento: Bibiana Rossi
Lugar: oficina del experto
Tiempo: 18 a 20 hs.
Objetivos:
a) Identificación de la viabilidad
b) Identificar los términos para realizar un análisis estructural de textos
c) Proponer el método Delphi para adquisición de conocimientos con grupo de expertos.

IC. Breve explicación del Test de Viabilidad. Revisión de cada una de las características de las dimensiones de plausibilidad, justificación, adecuación y éxito.

E. Revisó y ratificó los valores de las características del Test de Viabilidad.

IC. Al hacer la lectura y el análisis de la bibliografía recomendada qué conceptos o términos es necesario identificar?

E. Las ventajas y desventajas de los ciclos de vida y los problemas o casos de ejemplo que presentan.

IC. Breve explicación del Método Delphi. ¿Colaboraría en el diseño de los cuestionarios y en el análisis de los mismos?

E. Por supuesto. Me parece una excelente idea consensuar la opinión de los expertos, es una forma clara y ordenada de hacerlo. Los primeros cuestionarios pueden basarse en el análisis que resulte en los textos de las ventajas y desventajas de cada ciclo de vida.

A.3.4 Análisis de la Sesión III

- *Conocimientos extraídos:*

Los conocimientos extraídos de la sesión III se encuentran reflejados en el en el capítulo IV, Estudio de Viabilidad.

- *Actividades para continuar la adquisición de conocimientos:*

- Preparar las primeras cuestionarios para las sesiones de adquisición de conocimiento con el Método Delphi.

A.3.5 Evaluación de la Sesión III

- *¿Se han logrado los objetivos?*

Si se han cumplido, se determinaron las características del Test de Viabilidad.

Se definió el proceso para extracción de conocimientos de la bibliografía y el experto se comprometió a colaborar en las sesiones de adquisición de conocimientos con el grupo de expertos.

- *¿Es necesario volver sobre lo mismo?*

No parece necesario.

- *Número y tipo de sesiones para cubrir el área*

Se estima

- 1 sesión con el experto principal para la preparación de los cuestionarios.
- 3 sesiones con el grupo de expertos.
- 3 sesiones con el experto principal para realizar el análisis y tabulación de los resultados.
- 3 sesiones con el experto principal para completar la fase inicial de adquisición de los conocimientos.

B.1 EXTRACCION DE CONOCIMIENTOS

B.1.1 Preparación del Análisis de textos

- *Información a tratar:* Tipos de Ciclos de vida. Ventajas y desventajas de los ciclos de vida. Casos de aplicación.
- *Técnica utilizada:* Análisis estructural de textos.
- *Términos a buscar:*
 - Conceptos: ciclo de vida, fases, etapas
 - Es una característica de ...;
 - Es una ventaja de ..., es una desventaja de
 - Es un caso de aplicación de...

B.1.2 Realización de las sesiones de extracción de conocimientos

El IC realizó el análisis de la bibliografía disponible tomando como base los textos mencionados por el experto en la sesión II.

El análisis de textos se realizó en 25 sesiones de 3 horas aproximadamente durante un mes.

Se documentó el análisis de textos en una matriz de cuatro columnas con el siguiente diseño.

Título del grupo de CV			
Modelo	Ventajas	Desventajas	Casos de Ejemplo
Aaaa	➤ Ventaja 1 ➤ Ventaja 2 ➤ ...	➤ Desventaja 1 ➤ Desventaja 2 ➤ ...	Descripción
Bbbb	➤ Ventaja 1 ➤ Ventaja 2 ➤ ...	➤ Desventaja 1 ➤ Desventaja 2 ➤ ...	Descripción
Cccc	➤ Ventaja 1 ➤ Ventaja 2 ➤ ...	➤ Desventaja 1 ➤ Desventaja 2 ➤ ...	Descripción

El experto principal revisó la documentación en 4 momentos:

- Al finalizar el primer análisis de textos
- Al finalizar los 4 primeros textos
- Al finalizar los 8 primeros textos
- Al dar por terminada la extracción de conocimientos

B.1.3 Documentación del análisis realizado

Se adjunta a continuación de la presente documentación en el ítem B 1.6, el documento de trabajo: Modelos de Ciclos de Vida, en el que se sintetizó el análisis de textos.

B.1.4 Análisis de la extracción de conocimientos

- *Conocimientos extraídos:*

Ventajas, desventajas y características de los modelos de ciclos de vida de acuerdo con la bibliografía revisada.

No se han detectado en la bibliografía casos de ejemplo.

- *Conocimientos a educir en próximas sesiones:*

Identificar con mayor precisión las características particulares de los ciclos de vida.

B.1.5 Evaluación de las sesiones de extracción de conocimientos

- *¿Se han logrado los objetivos?*

Sí se han cumplido, se realizó un análisis de la bibliografía más relevante recomendada por el experto.

Se ha precisado el alcance del prototipo:

- Se convino con el experto que para la realización de este primer prototipo no es necesario continuar con el análisis bibliográfico.
- Se ha convenido en virtud de la revisión realizada por el experto en trabajar sobre los ciclos de vida en Cascada, Espiral y Orientado a objetos para la realización de este primer prototipo.
- Debe considerarse en el diseño la posibilidad de incorporar más conocimiento al sistema experto una vez que se haya desarrollado y esté en uso, tanto sea por la incorporación de mas bibliografía estudiada como así también por la incorporación de más modelos de ciclos de vida.

- *¿Es necesario volver sobre lo mismo?*

Es necesario refinar la información obtenida y consensuarla con el grupo de expertos, siguiendo el procedimiento propuesto en el método Delphi.

- *Número y tipo de sesiones para cubrir el área*

Se estima:

- 1 sesión con el experto principal para la preparación de los cuestionarios.
- 3 sesiones con el grupo de expertos.
- 3 sesiones con el experto principal para realizar el análisis y tabulación de los resultados.
- 3 sesiones con el experto principal para completar la fase inicial de adquisición de los conocimientos.

B.1.6 : Resultado del Análisis de textos

MODELOS DE CICLOS DE VIDA

Prototipado Rápido

Es un enfoque que intenta remediar algunos de las falencias del ciclo de vida (1)

Prototipo Evolutivo

El propósito del Prototipo Evolutivo es permitir al sistema evolucionar a medida que es usado.

El sistema es construido gradualmente.

Al principio se hace el suficiente desarrollo que permita al usuario llevar a cabo una o más de las tareas en forma completa (1)

Prototipo de usar y tirar

Es usado con el propósito de obtener especificaciones y clarificaciones (1)

Una vez que ha servido su propósito es descartado (1)

Prototipado Incremental

Aquí el sistema es construido en forma incremental, una sección por vez (1)

Combina elementos del modelo lineal (cascada) con la filosofía interactiva de los prototipos (5)

Cada secuencia lineal produce un incremento del software. (5)

El primer incremento a menudo es un producto esencial (núcleo).

El plan afronta la modificación del producto central a fin de cumplir las necesidades del cliente y la entrega de las funciones y características adicionales.

Este proceso se repite hasta que el producto está completo (5)

Se centra en la entrega de un producto operacional en cada incremento (5)

El modelo incremental corrige la necesidad de una secuencia no lineal de pasos de desarrollo. En el modelo incremental se va creando el sistema software añadiendo componentes funcionales al sistema llamado incremento. En cada paso sucesivo, se actualiza el sistema con nuevas funcionalidades o requisitos, es decir, cada versión o refinamiento parte de una versión previa y le añade nuevas funciones.

El sistema software ya no se ve como una única entidad monolítica con una fecha fija de entrega, sino como una integración de resultados sucesivos obtenidos después de cada iteración.

Cascada (Waterfall) (2) (5)

Sugiere un enfoque sistemático, secuencial del desarrollo del software. El numero de fases o etapas que se proponen en este ciclo de vida suele variar, aunque suelen ser: análisis de requisitos del sistema, análisis de requisitos del software, diseño preliminar, diseño detallado, codificación , pruebas, explotación y mantenimiento.

Tiene las siguientes actividades (5) : Ingeniería y modelado de Sistemas / Información, Análisis de los requisitos del software, Diseño, Generación del código, Pruebas, Mantenimiento (5). Cada fase empieza cuando se ha terminado la fase anterior. Para pasar de una fase a otra es necesario conseguir todos los objetivos de la etapa previa.

Modelo en espiral (2) (5)

Se centra en un gráfico en espiral en el cual

- la dimensión radial representa el costo acumulado incurrido en lograr los pasos hasta la fecha
- la dimensión angular representa el progreso hecho en cada ciclo de la espiral
- cada ciclo comienza con la identificación de:
 - los objetivos de la porción del producto que se esta elaborando
 - las restricciones impuestas sobre la aplicación de alternativas (2)
- el siguiente paso es evaluar las alternativas con respecto a los objetivos y las restricciones (2)
- el siguiente paso involucra la formulación de una estrategia para resolver las fuentes de riesgo (2)
- el siguiente paso esta determinado por los riesgos remanentes. Si la performance o los riesgos de interfaces de usuario dominan el desarrollo del programa entonces el siguiente paso podría ser desarrollo evolutivo. (2)
- cada ciclo es completado con una revisión involucrando a la gente a la que le concierne el producto (2)

Según Pressman el modelo tiene las siguientes regiones de tareas: (5) comunicación con el cliente, planificación, análisis de riesgos, Ingeniería, construcción y adaptación, evaluación del cliente

Metodología OMT (Object Modeling Tool) (6)

Consta de varias fases:

- *El análisis* se dedica a la comprensión y modelado de la aplicación y del dominio en el cual funciona
- La arquitectura global del sistema se determina durante el *Diseño del Sistema*. Utilizando el modelo de objetos como guía, se organiza el sistema en subsistemas
- Durante la fase de *Diseño de Objetos*, se elaboran los modelos de análisis, se refinan, y después se optimizan para producir un diseño práctico

Una aproximación orientada a objetos:

- traslada gran parte del esfuerzo del desarrollo del software hacia la fase de análisis del ciclo de vida
- centra su atención en las estructuras de datos, y no en las funciones que haya que efectuar

El propósito del análisis orientado a objetos es modelar el sistema del mundo real para que sea posible entenderlo

El diseño del sistema es la estrategia de alto nivel para resolver el problema y construir una solución

Durante el diseño del sistema, se decide la estructura y el estilo global

Durante el diseño de objetos, se ejecuta la estrategia seleccionada durante el diseño del sistema, y se rellena los detalles

Los desarrollos experimentados son capaces de combinar varios pasos, o bien pueden llevar a cabo varios pasos en paralelo para ciertas partes del proyecto

Modelos CV OO Booch (7)

Es iterativo en el sentido de que conlleva el refinamiento sucesivo de una arquitectura orientada a objetos, por la cual se aplica la experiencia y resultados de cada versión a la siguiente iteración del análisis y el diseño. El proceso es incremental en el sentido de que cada paso por un ciclo análisis/diseño/evolución lleva a refinar gradualmente las decisiones estratégicas y tácticas, convergiendo en última instancia hacia una solución que se encuentra con los requerimientos reales del usuario final y que además es simple, fiable y adaptable.

Utiliza el diseño global circular que enfatiza el desarrollo incremental e iterativo de un sistema mediante el refinamiento de vistas lógicas y físicas diferentes, aunque consistentes del sistema como un todo. El diseño global circular es el fundamento del proceso del diseño orientado a objetos.

Microproyecto: esta compuesto por las siguientes actividades:

- identificar las clases y objetos a un nivel de abstracción dado
- identificar las semánticas de estas clases y objetos
- identificar las relaciones entre estas clases y objetos
- especificar el interfaz y después la implementación de estas clases y objetos

Macroproceso

sirve como marco de referencia para controlar al microproceso. Incluye prácticas como gestión de configuraciones, control de calidad, recorridos de código y documentación. Incluye las siguientes actividades.

- establecer requisitos centrales para el software (conceptualización)
- desarrollar un modelo del comportamiento deseado del sistema (análisis)
- crear una arquitectura para la implementación (diseño)
- transformar la implementación mediante refinamiento sucesivo (evolución)
- gestionar la evolución posventa o postentrega (mantenimiento)

Es de importancia capital tener un liderazgo fuerte en el proyecto que gestione y dirija activamente las actividades del mismo

Two Leg Model (2)

Contiene procesos de abstracción separados hasta que una especificación formal es conseguida, seguido por un conjunto de pasos formales deductivos de rectificación para proceder a lo largo del diseño y la codificación (2)

Análisis Estructurado/Diseño Estructurado (6)

Es útil para aquellos problemas en los que las funciones sean más importantes y complejas que los datos

Modelo Convencional (4)

Esta compuesto por las siguientes fases

- fase de requerimientos
- fase de diseño
- fase de implementación
- llover el diseño al código

Separa los requerimientos del comportamiento interno. Separa los mecanismos de alto y bajo nivel. Se mezclan los mecanismos de intra módulos con decisiones de implementación de lenguaje

Mejora Iterativa (3)

Incluye el uso de un diseño modular descendente, un diseño cuidadoso antes de la codificación, componentes modulares bien estructurados y un mínimo número de implementadores.

Cada paso puede hacer uso de refinamiento. Representa una forma práctica de aplicar refinamiento paso a paso.

El primer paso consiste de una implementación inicial simple del esqueleto del subproblema de un proyecto. Se crea una lista de control de proyecto que contiene todas las tareas que necesitan completarse para conseguir la implementación final deseada.

En los pasos restantes se mejora iterativamente hasta que la implementación final es conseguida. Cada etapa iterativa consiste en seleccionar y quitar la siguiente tarea de la lista, diseñando la implementación para la tarea seleccionada.(fase de diseño). codificarla y debugear la implementación de la tarea (implementación), realizar el análisis de la implementación parcial existente a esta altura de iteración y poner al día la lista de control de proyecto. El proceso es iterado hasta la lista de control de proyecto esté vacía.

La mejora iterativa es un algoritmo heurístico que comienza con la implementación de un subproblema y prosigue con la modificación iterativa de la implementación existente basada en un conjunto de guías informales para conseguir la implementación completa deseada. Esta técnica involucra el desarrollo de un producto de software a través de una secuencia de pasos sucesivos de diseño y implementación empezando con una “predicción” inicial y una implementación de un esqueleto del subproblema.

Enfoque operacional (4)

- ✓ incluye los requerimientos del comportamiento interno en la especificación operacional
- ✓ separa la orientación a problema de la especificación operacional de las consideraciones de implementación
- ✓ todos los mecanismos funcionales son explicitados en la especificación funcional
- ✓ separa los mecanismos de su realización en términos de lenguaje de implementación

PROTOTIPOS			
Modelo	Ventajas	Desventajas	Ejemplo
Prototipo Evolutivo	<ul style="list-style-type: none"> ➤ No está limitado a proyectos chicos también puede manejar proyectos grandes (1) ➤ Sirve para la formular los requerimientos de un sistema de Software cuando los requerimientos del usuario son vagos, incompletos o inestables (1) ➤ Puede servir como herramienta para experimentar con nuevas e innovadoras ideas de diseño (1) ➤ Puede servir como factor de seguridad en desarrollos con alto factor de riesgo (1) ➤ Puede servir como forma de reaccionar ante potenciales cambios organizacionales (1) ➤ Puede servir como forma de promover al cliente a participar del proceso de desarrollo (1) ➤ Facilita un ambiente de enseñanza para usuarios finales potenciales durante el desarrollo (1) ➤ Puede facilitar la introducción gradual de un sistema de computación en una organización (1) ➤ Es usado cuando hay un gran nivel de incertidumbre (1) ➤ Es usado cuando hay varias opciones de diseño e implementación (1) ➤ Es usado cuando hay dificultades en formular las especificaciones (1) ➤ Es usado cuando no hay experiencia previa en el desarrollo con una técnica específica (1) ➤ Cuando se necesita un método para producir el sistema en forma gradual (1) 		

Modelo	Ventajas	Desventajas	Ejemplo
Prototipo de usar y tirar	<ul style="list-style-type: none"> ➤ Sirve para formular los requerimientos de un sistema de Software cuando los requerimientos del usuario son vagos, incompletos o inestables 	<ul style="list-style-type: none"> ➤ Se puede caer en el error de prolongar en el desarrollo final, decisiones ineficientes utilizadas en el prototipo 	
Prototipado Incremental	<ul style="list-style-type: none"> ➤ Los incrementos se pueden planear para gestionar riesgos técnicos ➤ Se ajusta a entornos de alta incertidumbre, por no tener la necesidad de poseer un conjunto exhaustivo de requisitos, (especificaciones, diseños, etc.), al comenzar el sistema, ya que cada refinamiento amplía los requisitos y las especificaciones derivadas de la fase anterior. ➤ Es particularmente útil cuando la dotación de personal no está disponible para una implementación completa en cuanto a la fecha límite de gestión que se ha establecido (5) 	<ul style="list-style-type: none"> ➤ Falta de planificación a largo plazo ➤ Tentación de caer en el modelo de codificar y corregir ➤ Existe el problema de determinar si los requisitos propuestos son válidos, se detectan tarde. ➤ Puede encomendar demasiados recursos a una solución errónea ➤ Es difícil para más de una persona trabajar en un único prototipo ➤ Aunque permite el cambio continuo de requisitos existe el problema de si los requisitos propuestos son válidos. ➤ Los errores en los requisitos se detectan tarde y su corrección resulta tan costosa como en el modelo en cascada. 	
Prototipado Rápido		<ul style="list-style-type: none"> ➤ Descuida el análisis apropiado ➤ Recae en la intuición de los desarrolladores ➤ Lleva a decisiones de diseño prematuras ➤ La adquisición de conocimientos y el análisis se vuelven manejados por la implementación ➤ Se hace difícil de mantener ya que favorece la falta de documentación del análisis y diseño. 	

CASCADA			
Modelo	Ventajas	Desventajas	Ejemplo
Cascada	<ul style="list-style-type: none"> ➤ Ayuda a prevenir que se sobrepasen las fechas de entrega y los costos esperados. ➤ Al final de cada fase el personal técnico y los usuarios tienen la oportunidad de revisar el progreso del proyecto. ➤ Da facilidades a los gestores para controlar el progreso de los sistemas. ➤ Reconocimiento de ciclos de realimentación entre etapas (2) ➤ Es mejor que un enfoque hecho al azar (5) 	<ul style="list-style-type: none"> ➤ Enfoque secuencial para construir un producto en el cual la iteración no es evidente. ➤ Se tarda mucho tiempo en pasar por todo el ciclo, dado que hasta que no se finalice una fase no se pasa a la siguiente. ➤ El enfoque top-down necesita ser matizado con un paso de reciclado (vuelta para atrás) para cubrir temas como riesgo y reuso de módulos de software (2) ➤ No maneja adecuadamente aspectos concernientes al desarrollo de familias de programas y de organización de software para permitir cambios (2) ➤ Asume progresión relativamente uniforme en los pasos de elaboración (2) ➤ El énfasis en elaborados documentos como criterio de finalización de las fases de requerimientos y diseño no funciona bien para muchas clases de software, particularmente las aplicaciones interactivos. ➤ No contempla la clase de desarrollo evolutivo que presenta el prototipado rápido y los lenguajes de cuarta generación. (2) ➤ Los proyectos reales rara vez siguen el modelo secuencial que propone el modelo (5). ➤ No contempla los posibles modos de desarrollo de software futuros, asociados con las 	

Modelo	Ventajas	Desventajas	Ejemplo
		<p>capacidades de la programación automática, transformación de programas y asistentes de software basados en el conocimiento. (2)</p> <ul style="list-style-type: none"> ➤ A menudo es difícil que el cliente explice todos los requisitos (5) ➤ Una versión de trabajo del programa no estará disponible hasta que el proyecto esté muy avanzado (5) ➤ No refleja el proceso real de desarrollo de software. Los proyectos reales raramente siguen este flujo secuencial, puesto que siempre hay iteraciones. Aunque en este modelo la iteración esté permitida en etapas contiguas (MACRO,1990),en la vida real la iteración abarca mas de una etapa. Un caso típico es la redefinición de los requisitos cuando se está codificando la aplicación. ➤ Acentúa el fracaso de la industria del software con el usuario final. En este caso, el usuario debe tener paciencia, ya que el sistema en funcionamiento no estará disponible hasta las fases finales del proyecto. ➤ Asume una especificación de requisitos perfecta. ➤ Detección de errores tardía ➤ Agravado para KBS (incompleto y con especificaciones cambiantes) ➤ Administradores y usuarios tienen poca idea de como quedará el sistema cuando se especifican los requerimientos 	

ESPIRAL			
Modelo	Ventajas	Desventajas	Ejemplo
Modelo en espiral	<ul style="list-style-type: none"> ➤ En el desarrollo interno existe una gran flexibilidad y libertad para ajustarse a los acuerdos etapa por etapa, para aplazar acuerdos de opciones específicas, para establecer miniespirales para resolver caminos críticos, para ajustar niveles de esfuerzo, o para acomodar prácticas como prototipado desarrollo evolutivo y uso de métodos de diseño ajustado al costo. ➤ Alienta el desarrollo de especificaciones que no son necesariamente uniformes, exhaustivas o formales, al diferir la elaboración detallada de los elementos de software de bajo riesgo, y evita roturas innecesarias en sus diseños, hasta que los elementos de alto riesgo del diseño sean establecidos. ➤ La revisión de los principales objetivos sirve para asegurar que todas las partes involucradas están de acuerdo respecto al método de trabajo para la siguiente fase. ➤ Existe un reconocimiento explícito de las diferentes alternativas para alcanzar los objetivos de un proyecto. La identificación de riesgos asociados con cada una de las alternativas y las diferentes maneras de resolverlos son el centro del modelo. Con los métodos tradicionales, es habitual dejar las partes, más difíciles para el final y empezar con las más fáciles y de menor riesgo, obteniendo así la ilusión de un gran avance. 	<ul style="list-style-type: none"> ➤ Trabaja bien en los desarrollos internos, pero necesita un ajuste posterior para adaptarlo a la subcontratación de software. ➤ Necesidad de expertos en evaluación de riesgos para identificar y manejar las fuentes de riesgos de un proyecto. ➤ Requiere una considerable habilidad para la evaluación del riesgo y depende de ella para el éxito (5) ➤ Deposita una gran cantidad de confianza en la habilidad de los desarrolladores de software para identificar y manejar las fuentes de riesgo del proyecto. (5) ➤ En general, los pasos del proceso necesitan una elaboración adicional para asegurar que todos los participantes de un desarrollo de software estén operando en un contexto consistente. (5) ➤ Presenta un riesgo de decisiones prematuras de diseño ➤ Puede demostrar dificultad para ejecutar especificaciones con una adecuada performance ➤ En el desarrollo de software bajo contrato no existe esta flexibilidad y libertad, por lo que es necesario mucho tiempo para definir los contratos, ya que los entregables no están previamente definidos de forma clara. 	

Modelo	Ventajas	Desventajas	Ejemplo
	<ul style="list-style-type: none"> ➤ Contesta la pregunta “cuánto es suficiente?” para cada una de las fuentes de la actividad del proyecto y para el manejo de los recursos. ➤ La división de los proyectos en ciclos, cada uno con un acuerdo al final de cada ciclo, implica que existe un acuerdo para los cambios que hay que realizar o para terminar el proyecto, en función de lo que se ha aprendido desde el inicio del proyecto. ➤ El modelo se adapta a cualquier tipo de actividad, incluidas algunas que no existen en otros métodos (por ejemplo, consulta de asesores expertos o investigadores ajenos) que son muy útiles para la consecución de los objetivos de un proyecto. ➤ Puede aplicarse a lo largo de la vida del proyecto (5) ➤ Acomoda una mezcla apropiada de enfoques orientados a especificaciones, orientados a prototipos, orientados a simulaciones, orientados a transformación automática, y otros ➤ Es aplicable tanto a esfuerzos de desarrollo como de mejora (enhacement). (2) ➤ Incorpora prototipado como una opción de reducción ➤ corte innecesario en su diseño y como una opción de reducción de riesgo en cualquier etapa del desarrollo Promueve el desarrollo de especificaciones que no son necesariamente uniformes, exhaustivas o formales en las que posterga la elaboración detallada de elementos de software de bajo riesgo, evitando un corte innecesario en su diseño (2) 	<ul style="list-style-type: none"> ➤ Todavía no encaja en el mundo de la adquisición de software por contrato. ➤ A no ser que se realice una inspección por expertos, en este tipo de proyecto se tendrá la ilusión de progresar por un período, y sin embargo, se encuentra dirigido directamente hacia el desastre. ➤ Las personas pueden encontrarlo difícil para dirigir transformaciones 	

Modelo	Ventajas	Desventajas	Ejemplo
	<ul style="list-style-type: none"> ➤ Mejora la estimación y reduce el costo de corregir ➤ Permite vueltas atrás en etapas tempranas del espiral al identificar alternativas más atractivas o al necesitar la resolución de un nuevo riesgo. (2) ➤ Su rango de opciones y enfoque orientado a riesgos permite acomodar las mejores características de los modelos de software existentes evitando sus dificultades (2) ➤ Permite explícitamente estrategias para desarrollar familias de programas y para reusar software existente (2) ➤ Permite la evolución del ciclo de vida, crecimiento y cambios en el producto de software. (2) ➤ Provee mecanismos para incorporar objetivos de calidad del software en el proceso de desarrollo del producto de software. (2) ➤ Se focaliza en eliminar errores y alternativas poco atractivas en forma temprana (2) ➤ Permite iteraciones, vueltas atrás y terminación prematura de proyectos no viables (2) ➤ Puede soportar y ser soportado por ambientes de desarrollo de software avanzados (2) ➤ No implica procedimientos separados para desarrollo y mejoramiento de software. (2) ➤ Provee un marco viable para el desarrollo de sistemas de software o hardware integrados. (2) ➤ Evita forzar procedimientos del desarrollo de software en los paradigmas de desarrollo de hardware 		

ORIENTACION A OBJETOS			
Modelo	Ventajas	Desventajas	Ejemplo
Modelo para Desarrollo de Sistemas Orientados a Objetos	<ul style="list-style-type: none"> ➤ La tecnología de objetos pretende acelerar el desarrollo de sistemas de una manera iterativa e incremental. ➤ La ventaja principal que permiten fijar hitos más frecuentemente, realizando entregas de sistemas que son operativos cada dos o tres meses para recibir retroalimentación del cliente lo antes posible e ir adaptando la aplicación según cambien las necesidades y se refinen los requisitos 	<ul style="list-style-type: none"> ➤ Otro aspecto importante en la tecnología de objetos es la de "generalizar" los componentes para que sean reutilizables, lo que incrementa los costos de desarrollo entre un 10 y 50%, por lo que resulta imprescindible un desarrollo que optimice esta inversión. ➤ El inconveniente que presentan es la dificultad de gestionar de manera formal los proyectos que siguen estos ciclos de vida aunque este problema se puede paliar diferenciando el "micro" del "macroproceso". 	
Object Modeling Tool (OMT) (6)	<ul style="list-style-type: none"> ➤ Produce un diseño limpio y fácil de comprender, que resulta más fácil de probar, mantener y extender ➤ El diseño resultante es más adaptable, los cambios futuros serán mucho más sencillos ➤ Da al proceso de desarrollo una base más estable, y permite utilizar un único concepto unificador de software a lo largo del proceso: el concepto de objeto, de tal forma que la información registrada durante el análisis no se pierde ni se transforma cuando se produce el diseño y la implementación ➤ La organización de un sistema en torno de objetos da al desarrollo una estabilidad mayor que las orientadas a funciones. ➤ Es más flexible al cambio y más extensible. ➤ Cambios de funcionalidad se admiten con facilidad en el diseño orientado a objetos 	<ul style="list-style-type: none"> ➤ La distinción entre análisis y diseño podrá parecer a veces arbitraria y confusa 	

Ejemplo	Ventajas	Desventajas	Ejemplo
	<ul style="list-style-type: none"> ➤ No hay discontinuidad en los modelos. La misma notación hace más sencillo repetir los pasos de desarrollo con grados de detalle cada vez más finos. Cada iteración añade o clarifica características. Hay menos oportunidades para incongruencias y errores ➤ El paradigma del mundo real formado por objetos y relaciones proporciona el contexto para comprender el comportamiento dinámico y funcional ➤ Los sistemas son más fáciles de entender, esto hace que el diseño sea más intuitivo y simplifica la seguibilidad entre los requisitos y el código de software. El diseño resulta más coherente para personas que no fueran parte del equipo original del diseño ➤ Se incrementa la reutilizabilidad de los componentes de un proyecto para el siguiente ➤ Integra mejor las bases de datos con el código ➤ La descomposición de software OO modela más de cerca la percepción que de la realidad tiene la persona. Por tanto, no es sorprendente que el software desarrollado resulte más comprensible, extensible y mantenible. 		

Modelo	Ventajas	Desventajas	Ejemplo
Modelos CV OO Booch	<ul style="list-style-type: none"> ➤ La tecnología de objetos pretende acelerar el desarrollo de sistemas de una manera iterativa e incremental. ➤ Permite fijar hitos más frecuentemente, realizando entregas de sistemas que son operativos, para recibir retroalimentación del cliente lo antes posible e ir adaptando la aplicación según cambian las necesidades y se refinan los requisitos. ➤ El enfoque evolutivo que el macroproceso adopta para el desarrollo significa que hay oportunidades para identificar problemas en momentos tempranos del ciclo de vida y responder a estos riesgos antes de que comprometan el éxito del proyecto (7) ➤ El proceso de desarrollo OO ayuda explícitamente a desarrollar factores de calibración (7) ➤ La gestión de proyectos OO, en estado estables, provoca una reducción en la cantidad total de recursos que se necesitan y un desplazamiento en el ritmo de su despliegue respecto a métodos más tradicionales. ➤ Los costos del ciclo de vida son con frecuencia menores que los de un enfoque tradicional porque el producto resultante tiende a ser de mucha mejor calidad y por eso es mucho mas flexible ante el cambio 	<ul style="list-style-type: none"> ➤ La tecnología de objetos pretende <i>generalizar</i> los componentes para que sean reutilizables, lo que incrementa los costos de desarrollo entre un 10 y un 50%, por lo que resulta imprescindible un desarrollo que optimice esta inversión. ➤ Dificultad de gestionar de manera formal los proyectos que siguen estos ciclos de vida aunque, como se ha señalado, este problema se puede paliar diferenciando el <i>micro</i> del <i>macroproceso</i> ➤ En algunos casos tiene un alto costo de puesta en marcha (7) ➤ Tiene problemas de eficacia en ciertos casos particulares (7) ➤ El microproceso del desarrollo orientado a objetos es inestable de forma innata y requiere una dirección activa para forzar su conclusión (7) 	

Modelo	Ventajas	Desventajas	Ejemplo
	<ul style="list-style-type: none"> ➤ Alienta la reutilización de componentes de software. ➤ La integración incremental tiende a reducir el riesgo de desarrollo porque acelera el descubrimiento de problemas de arquitectura y eficacia en etapas tempranas del desarrollo (7) ➤ El macroproceso funciona muy bien para asegurar la calidad porque permite una recolección de datos continua sobre la tasa de descubrimiento de errores (7) ➤ La integración sucede incrementalmente a lo largo de todo el ciclo de vida en vez de ocurrir en un evento explosivo (7) ➤ Un desarrollo orientado a objetos explota la potencia expresiva de todo los lenguajes de programación orientados a objetos (7) ➤ Lleva a sistemas más flexibles al cambio (7) ➤ Reduce el riesgo de desarrollo (7) ➤ Resulta atractivo al funcionamiento de la mente humana (7) ➤ El enfoque de desarrollo incremental es extremadamente apropiado para el paradigma orientado a objetos (7) 		

OTROS			
Modelo	Ventajas	Desventajas	Ejemplo
Parnas information hiding approach	➤ Maneja adecuadamente aspectos concernientes al desarrollo de familias de programas y a la organización de software para acomodar cambios (2)	➤ No ha sido completamente elaborado para ver como cubre aspectos tales como prototipado y reusabilidad (2)	
Two Leg Model	➤ Contiene procesos de abstracción separados hasta que se obtiene una especificación formal (2)	➤ Tiene retos en acomodar reusabilidad de software, familias de programas y compromisos (trade-offs) de diseño físico – lógico (2)	
Automatic Paradigm	➤ Incorpora al ciclo de vida un marco conceptual para incorporar capacidades de programación automática, transformación de programas y asistentes basados en conocimiento (2)	➤ Tiene problemas en escalar a sistemas muy grandes (2) ➤ Tiene problemas en acomodar familias de programas (2) ➤ Tiene problemas en manejar opciones entre nuevas y antiguas capacidades (2)	
Análisis /Diseño Estucurado (6)		➤ Tienen unos límites del sistema bien definidos, así que puede resultar difícil extender un diseño SA/SD hasta unos nuevos límites	
Modelo Convencional (4)	➤ Los requerimientos en prosa pueden ser leídos y aprobados directamente por los clientes (4) ➤ El prototipado es posible por iteración del ciclo de desarrollo entero ➤ Provee de puntos útiles de chequeo (milestones) (4)	➤ Los requerimientos informales son notorios por su ambigüedad (4) ➤ El testeo y la prueba de correctitud son muy difíciles (4) ➤ Es difícil que soporte automatización completa (4) ➤ Los requerimientos de caja negra son difíciles de especificar (4) ➤ La descomposición top-down es difícil y riesgosa	

Modelo	Ventajas	Desventajas	Ejemplo
Mejora Iterativa	<ul style="list-style-type: none"> ➤ Facilita el logro de objetivos de desarrollo de software para muchos proyectos (3) ➤ Provee una forma práctica de usar un enfoque top-down step-wise (3) 	<ul style="list-style-type: none"> ➤ Generalmente no es fácil de aplicar cuando el proyecto es de tamaño considerable(3) ➤ Requiere que el problema y su solución estén bien comprendidos (3) 	
Modelo Operacional (4)	<ul style="list-style-type: none"> ➤ Los requerimientos formales pueden ser procesados por la maquina (4) ➤ Está disponible prototipado rápido automáticamente (4) ➤ El testing o verificación es evitado derivando la implementación de la especificación usando solo transformaciones y mapeos que han sido probados (4) ➤ Las especificaciones ejecutables proveen resultados tempranos y (4) ➤ Maneja objetos formales, lo que implica que se pueden desarrollar templete y herramientas (4) ➤ Los usuarios tienen un modelo del sistema para interiorizarse y evaluarlo ➤ La implementación transformacional garantiza la corrección ➤ La transformación y realización son altamente automáticas. ➤ Direcciona directamente el conflicto estructural entre la eficiencia y mantenimiento sencillo. 	<ul style="list-style-type: none"> ➤ Los requerimientos formales son inaccesibles para los usuarios finales y otras personas no técnicas (4) ➤ Presenta peligro de decisiones de diseño prematuras (4) ➤ Puede resultar difícil ejecutar especificaciones con performance adecuada ➤ Es una tecnología nueva y subdesarrollada (4) ➤ La implementación transformacional es nueva y no desarrollada tecnológicamente ➤ Las personas pueden encontrarlo difícil para dirigir transformaciones 	

Bibliografía Analizada

1. Experiencie with evolutionary Prototyping in a Large Software Project. Sharam Hekmatpour
2. A spiral model of software development and enhacement. Barry Boehm
3. Iterative Enhacement: A Practical Technique for Software Development. Victor R. Basili and Albert J. Turner
4. The operational versus the conventional approach to software development. Pamela Zave
5. Ingeniería del Software – Un enfoque práctico. 4ta Ed. – Roger S. Pressman. Madrid. 1998
6. Modelado y diseño orientado a objetos OMT- Rumbaugh
7. Análisis y Diseño Orientado a Objetos con Aplicaciones - Booch
8. Estándares del IEEE 1074-1991
9. Estándares ISO 12701-1
10. Análisis y Diseño Orientado a Objetos - Martin y Odell
11. Elementos y Herramientas en el desarrollo de sistemas de información- Piattini
12. Análisis y diseño detallado de aplicaciones informáticas de Gestión- Piattini