APLICACIÓN DE LA METODOLOGÍA QFD EN EL PROCESO DE INGENIERÍA DE REQUISITOS

CAROLINA ARROYAVE
ALEJANDRA MAYA
CARLOS OROZCO

Trabajo de Grado para optar por el título de
Ingeniero de Sistemas

Asesor
RAFAEL DAVID RINCÓN BERMÚDEZ
Magíster en Matemáticas Aplicadas
Magíster en Sistemas de Calidad

UNIVERSIDAD EAFIT
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS
INGENIERÍA DE SISTEMAS
2007

Nota de Aceptación

_____

_____

_____

_____

Presidente del Jurado

_____

Jurado

_____

Jurado

Medellín, 16 de Mayo del año 2007

A nuestras familias, profesores y amigos por el apoyo incondicional en todo momento y su valioso aporte en nuestra formación académica.

AGRADECIMIENTOS

Debemos manifestar que no hubiésemos logrado sacar adelante nuestro trabajo de grado sin contar con una serie de personas a las cuales deseamos agradecer por su ayuda, colaboración y apoyo.

Primero, debemos dar gracias a Dios, por estar con nosotros en cada paso que hemos dado, por iluminar nuestra mente y por haber puesto en el camino a aquellas personas que han sido nuestro soporte y compañía durante toda nuestra carrera.

A nuestro asesor de proyecto Rafael David Rincón, por creer siempre que lograríamos sacar nuestro proyecto exitosamente, esto nos mantuvo motivados todo este tiempo. Así mismo, por su colaboración y buena disposición para escuchar nuestras propuestas, dudas y comentarios; con sus sugerencias y correcciones oportunas pudimos elaborar un muy buen trabajo.

Al personal administrativo del Colegio Anglo Español, por abrirnos sus puertas y brindarnos la oportunidad de probar y aplicar nuestra propuesta, proporcionándonos una retroalimentación muy importante para el objetivo de este trabajo. Esperamos que los resultados obtenidos puedan llevarse a la realidad en muy poco tiempo.

A nuestras familias, porque desde el inicio de nuestra carrera nos han ofrecido su cariño y comprensión, fueron un soporte incondicional en todo momento. A Aleja, Carlos y Camilo por su entendimiento y ánimos para seguir adelante, fue vital para poder llegar a este punto.

A los amigos y demás personas que de una u otra forma tuvieron confianza en nosotros y en el desarrollo de nuestro trabajo y siempre nos manifestaron su apoyo hasta la consolidación de este triunfo.

TABLA DE CONTENIDO

INTRODUCCIÓN 1

CAPITULO 1: PROPUESTA DE INVESTIGACIÓN 5

1.1 ANTECEDENTES 5

1.1.1 PROBLEMAS EN LA INGENIERÍA DE SOFTWARE 5

1.1.2 BENEFICIOS DE LA METODOLOGÍA QFD 11

1.2 PLANTEAMIENTO DEL PROBLEMA 13

1.3 OBJETIVOS 14

1.3.1 OBJETIVO GENERAL 14

1.3.2 OBJETIVOS ESPECÍFICOS 14

1.4 ALCANCE 15

1.5 HIPÓTESIS 15

1.6 PREGUNTA DE INVESTIGACIÓN 16

1.7 METODOLOGÍA DE LA INVESTIGACIÓN 16

CAPITULO 2: MARCO TEÓRICO 19

2.1 DESPLIEGUE DE LA FUNCIÓN DE CALIDAD: QFD 19

2.1.1 HISTORIA DE LA METODOLOGÍA QFD 19

2.1.2 ¿QUÉ ES QFD? 23

2.1.3 VENTAJAS DE QFD 26

2.1.4 LA CASA DE LA CALIDAD 28

2.1.5 SQFD 36

2.1.6 OTRAS HERRAMIENTAS DE QFD 41

2.2 MODELO DE KANO 46
2.2.1 TIPOS DE REQUISITOS DE LOS CLIENTES 47
2.2.2 CUESTIONARIOS 49
2.2.3 ANÁLISIS BÁSICO DE LOS DATOS 51
2.2.4 ANÁLISIS DETALLADO DE LOS DATOS 53
2.2.5 CICLO DINÁMICO DEL PRODUCTO 57

2.3 CAPABILITY MATURITY MODEL INTEGRATION (CMMI) 59
2.3.1 ORÍGENES Y PRINCIPIOS DEL CAPABILITY MATURITY MODEL (CMM) 59
2.3.2 ¿QUÉ ES CMMI? 61
2.3.3 COMPONENTES DE CMMI 63
2.3.4 ÁREAS DE PROCESO DE INGENIERÍA 64
2.3.5 ÁREA DE PROCESO: DESARROLLO DE REQUISITOS (RD) 68
2.3.6 ÁREA DE PROCESO: ADMINISTRACIÓN DE REQUISITOS (REQM) 73

CAPITULO 3: PROPUESTA METODOLÓGICA 78

3.1 PROCESO DE INGENIERÍA DE REQUISITOS 80
3.1.1 MODELO DE PROCESO DE INGENIERÍA DE REQUISITOS 82
3.1.2 FLUJO DE ACTIVIDADES 93

3.2 MODELADO DEL NEGOCIO 95
3.2.1 Tarea 1: Identificar los procesos del negocio y el problema 97
3.2.2 Tarea 2: Identificación de los roles del entorno 98
3.2.3 Tarea 3: Especificar las reglas de negocio 98
3.2.4 Tarea 4: Creación de un glosario de términos 99

3.3 ELICITACIÓN DE REQUISITOS 100
3.3.1 Tarea 1: Identificación de Participantes (Stakeholders) 101
3.3.2 Tarea 2: Preparar y realizar las sesiones de elicitación 106
3.3.3 Tarea 3: Identificar requisitos de información 120
3.3.4 Tarea 4: Elaborar el Documento de Requisitos del Sistema 124
3.3.5 Tarea 5: Negociar Requisitos 130

3.4 ANÁLISIS DE REQUISITOS 133
3.4.1 Tarea 1: Clasificación de los requisitos 134
3.4.2 Tarea 2: Identificación de conflictos en los requisitos 137
3.4.3 Tarea 3: Elaborar matrices de relación 140
3.4.4 Tarea 4: Elaborar los cuestionarios de Kano 146
3.4.5 Tarea 5: Ejecutar el ciclo Análisis – Elicitación 148
3.4.6 Tarea 6: Tabulación de los resultados de los cuestionarios de Kano 149
3.4.7 Tarea 7: Construir la Casa de la Calidad 150
3.4.8 Tarea 8: Elaborar el Documento de Análisis de Requisitos 168

3.5 VALIDACIÓN DE REQUISITOS 171
3.5.1 Tarea 1: Construcción de prototipos 175
3.5.2 Tarea 2: Preparar y realizar las sesiones de walkthrough 177
3.5.3 Tarea 3: Validar los requisitos de información y funcionales 180
3.5.4 Tarea 4: Validar los requisitos no funcionales 181
3.5.5 Tarea 5: Cerrar la versión del documento de especificación de requisitos 182

3.6 GESTIÓN DE REQUISITOS 183
3.6.1 Gestión de cambios 185

CAPITULO 4: CASO PRÁCTICO 191

4.1 CONTEXTO DE LA ORGANIZACIÓN 191
4.1.1 HISTORIA 191
4.1.2 PROGRAMAS 192

4.2 DOCUMENTO DE REQUISITOS DEL SISTEMA 193
4.2.1 PROCESOS DE NEGOCIO 193
4.2.2 REGLAS DE NEGOCIO 196
4.2.3 ROLES Y RESPONSABILIDADES 197
4.2.4 DESCRIPCIÓN DEL SISTEMA ACTUAL 199
4.2.5 PARTICIPANTES Y RESPONSABILIDADES 201
4.2.6 CATÁLOGO DE NECESIDADES 202
4.2.7 GLOSARIO DE TÉRMINOS 215

4.3 DOCUMENTO DE ANÁLISIS DE REQUISITOS 217

4.3.1 REQUISITOS DEL SISTEMA CLASIFICADOS 217

4.3.2 MATRICES DE RELACIÓN 221

4.3.3 RESULTADOS DEL MODELO DE KANO 226

4.3.4 CASA DE LA CALIDAD 229

4.3.5 CONFLICTOS 243

4.3.6 APÉNDICES 245

TRABAJO FUTURO 246

CONCLUSIONES 248

BIBLIOGRAFÍA 254

ANEXOS

ÍNDICE DE FIGURAS

Figura 1. Costo Relativo de corregir un error.....6
Figura 2. Costo relativo para corregir un defecto en el ciclo de vida del software...7
Figura 3. La ingeniería de requisitos como un proceso de comunicación.....11
Figura 4. Etapas de la metodología de investigación.....16
Figura 5. Traducción de la palabra hinshitsu kino tenkai.....23
Figura 6. Qué(s).....29
Figura 7. Cómo(s).....30
Figura 8. Qué(s) vs. Cómo(s).....30
Figura 9. Conversión de Cómo(s) en Qué(s).....31
Figura 10. Relaciones entre los Qué(s) y los Cómo(s).....32
Figura 11. Tipos de Relaciones.....32
Figura 12. Ejemplo de relaciones entre los Qué(s) y los Cómo(s).....33
Figura 13. Ejemplo de una Casa de la Calidad.....35
Figura 14. Ejemplo matriz Actor x Caso de Uso.....40
Figura 15. Ejemplo matriz Calidad Requerida x Caso de Uso.....40
Figura 16. Ejemplo matriz Casos de Uso x Objetos.....41
Figura 17. Ejemplo de un diagrama de afinidad.....43
Figura 18. Ejemplo de un diagrama de árbol.....45
Figura 19. Tipos de Requisitos del cliente.....48
Figura 20. Ejemplo de tabulación de datos en el modelo de Kano.....52
Figura 21. Concentración de respuestas en forma gráfica.....54
Figura 22. Tipos de Requisitos del cliente.....57
Figura 23. Ciclo de Shewhart.....60
Figura 24. Representaciones de CMMI.....62
Figura 25. Áreas de Proceso de Ingeniería.....65
Figura 26. Estructura para identificar el cumplimiento de una meta o práctica.....79
Figura 27. La ingeniería de requisitos en el ciclo de vida del software.....81
Figura 28. Dimensiones de la ingeniería de requisitos.....82

Figura 29. Modelo de proceso propuesto por Amador Durán 83
Figura 30. Arquitectura general de RUP 85
Figura 31. Modelo de proceso, adaptado de [Durán, 2000] 87
Figura 32. Flujo de actividades y productos de trabajo propuestos 94
Figura 33. Modelado del Negocio 96
Figura 34. Proceso Usual de captura de requisitos 102
Figura 35. Disposición para construir un diagrama de afinidad 109
Figura 36. Formato de las tarjetas para los requisitos 112
Figura 37. Generación de un diagrama de afinidad 112
Figura 38. Ilustración de agrupación de ideas por afinidad 113
Figura 39. Esquema de un diagrama de afinidad 114
Figura 40. Ejemplo de un diagrama de afinidad 114
Figura 41. Relación entre objetivos y requisitos-C 115
Figura 42. Plantilla para requisitos de información 121
Figura 43. Portada de Documento de Requisitos 126
Figura 44. Lista de cambios 127
Figura 45. Tabla de contenido del documento de requisitos 130
Figura 46. Plantilla para definir conflictos 138
Figura 47. Notación para los pesos asociados de las relaciones 140
Figura 48. Ejemplo de un diagrama de contexto 142
Figura 49. Ejemplo de una matriz de relación Objetivo x Requisito-C 144
Figura 50. Ejemplo de una función CRUD que no debe ser eliminada 147
Figura 51. Ejemplo de diagrama de casos de uso 157
Figura 52. Qué(s) vs. Cómo(s) 158
Figura 53. Tipos de Relaciones 160
Figura 54. Estructura de la Matriz de Relaciones de la Casa de la Calidad 161
Figura 55. Estructura de la matriz de correlación en la Casa de la Calidad 163
Figura 56. Casa de la Calidad con todos sus componentes 166
Figura 57. Portada del documento de análisis del sistema 168
Figura 58. Estructura del documento de análisis de requisitos 170

Figura 59. Proceso de control de cambios 188
Figura 60. Diagrama de árbol - Sistematizar los libros 204
Figura 61. Diagrama de árbol - Calificaciones del Programa Regular 205
Figura 62. Diagrama de árbol - Calificaciones del Programa Tutorial 206
Figura 63. Diagrama de árbol - Listas después del proceso de matrícula 207
Figura 64. Diagrama de árbol - Informes y reportes 207
Figura 65. Diagrama de árbol - Aspectos generales del sistema 208
Figura 66. Casos de uso de la funcionalidad Calificaciones 229
Figura 67. Casos de uso de la funcionalidad Calificaciones 230
Figura 68. Casos de uso de la funcionalidad Calificaciones 231
Figura 69. Casos de uso de la funcionalidad matrículas 232
Figura 70. Cuestionario de atribución de importancia 245
Figura 71. Las cuatro fases de QFD 246
Figura 72. Fases del QFD aplicadas a software 247

ÍNDICE DE TABLAS

Tabla 1. Estadísticas de resultados de proyectos de software en su estado final ... 8
Tabla 2. Principales diferencias entre los dos tipos de investigación 15
Tabla 3. Ejemplo de los Qué(s) para el diseño de una página Web 29
Tabla 4. Terminología: QFD vs. OO 39
Tabla 5. Ejemplo de cuestionario Kano 50
Tabla 6. Concentración de respuestas 51
Tabla 7. Clasificación de requisitos 52
Tabla 8. Concentración de respuestas 54
Tabla 9. Ejemplo de una tabla de concentración de respuestas 56
Tabla 10. Diferentes combinaciones de requisitos 58
Tabla 11. Deseos y aportes de los Stakeholders 105
Tabla 12. Nomenclatura recomendada por la IEEE 137
Tabla 13. Ejemplo de una matriz de relación Actores x Requisitos-C 142
Tabla 14. Ejemplo de Requisitos no Funcionales x Requisitos Funcionales 146
Tabla 15. Concentración de respuestas 148
Tabla 16. Tabulación de resultados para el Modelo de Kano 150
Tabla 17. Estructura del componente Qué(s) de la Casa de la Calidad 152
Tabla 18. Estructura del componente de prioridad de la Casa de la Calidad 153
Tabla 19. Plantilla para la especificación de los casos de us 155
Tabla 20. Estructura del componente Cómo(s) de la Casa de la Calidad 158
Tabla 21. Correspondencia entre las calificaciones cuantitativas y cualitativas .. 196
Tabla 22. Prioridades asignadas a los Requisitos Funcionales 227
Tabla 23. Prioridades asignadas a los Requisitos No Funcionales 228

INTRODUCCIÓN

La necesidad de aplicar los principios de otras ingenierías al desarrollo de software está plenamente justificada por el alto grado de fracasos en estos proyectos. Desde principios de los 90's el Standish Group ha realizado estudios sobre la situación del desarrollo de software y sus principales problemas en Estados Unidos, mostrando que, casi un tercio de los proyectos de desarrollo de software se cancelan durante su ejecución y que la gran mayoría presenta graves desviaciones respectos a plazos y presupuestos iniciales. Muchas de las causas de fracaso de estos proyectos estaban directamente relacionadas con la pobre identificación y gestión de los requisitos que debían cumplir las aplicaciones desarrolladas para satisfacer las necesidades de los usuarios.

Resultados como los anteriores, dejan claro que la Ingeniería de Software requiere de métodos, herramientas y prácticas que le permitan alcanzar un nivel de madurez que responda a las exigencias actuales del mercado y de las tecnologías de información.

El motivo principal de este trabajo se centra en proponer una metodología formal para aquellos equipos de desarrollo de sistemas software que están conscientes de que la satisfacción de los clientes es uno de los aspectos más relevantes para el éxito de una empresa y que por ello deben buscar alternativas que permitan desarrollar productos que suplan sus necesidades y cumplan con las especificaciones negociadas desde el principio.

La adaptación de la metodología QFD (*Quality Function Deployment*) a la ingeniería de requisitos se hará de una forma innovadora, sin perder la esencia de dicha metodología. Adicionalmente, a través de la propuesta realizada, se pretende enfrentar y prevenir las causas de fracaso que se han identificado a lo largo de los años, como son la insuficiente administración de requisitos,

comunicación ambigua e imprecisa, inconsistencias no detectadas, validaciones tardías de los requisitos y ejecución de cambios sin control.

Esta propuesta ha sido desarrollada por un grupo de estudiantes universitarios de la carrera Ingeniería de Sistemas de la Universidad EAFIT, quienes han tenido siempre un amplio interés por la Ingeniería de Software y sus diferentes disciplinas, donde la preocupación principal está orientada al mejoramiento del proceso de software, pues se tiene la conciencia que actividades propias de esta área, como la gestión de requisitos de clientes y usuarios, son algunas de las consideradas más críticas en el desarrollo y la producción del software. También se promueve la utilización de técnicas muy conocidas pero que fueron adaptadas hacia el logro real del objetivo para el cual fueron diseñadas.

Adicionalmente, al ser desplegado este proyecto en un contexto universitario, se busca contribuir con la motivación de otros estudiantes para que continúen con el trabajo futuro que se deja planteado y desarrollen propuestas creativas que se puedan aplicar en algunos años en las empresas del país.

Por ser una propuesta metodológica compuesta por una serie de pasos sencillos, pero que requieren de una organización y secuencia, se hace necesario que sea validada con algún caso real, que permita constatar posibles falencias para mejorarlas e identificar los beneficios que aportará al proceso de la ingeniería de requisitos. Por esto, como valor agregado, la propuesta será aplicada a un problema real donde se darán a conocer los resultados obtenidos, los inconvenientes enfrentados y las fortalezas identificadas en la metodología.

Esta tesis se encuentra dividida de la siguiente forma:

El **Capítulo 1** está dedicado a la introducción de la tesis, explicando los antecedentes del problema que se plantea y el alcance del mismo, así como también se deja plasmada la pregunta de investigación y la metodología que se seguirá para cumplir los objetivos. Adicionalmente, se exponen algunos beneficios

que proporciona la metodología QFD, de una manera muy general, dado que esto será ampliado a lo largo del trabajo.

El **Capítulo 2** se compone de tres grandes temas que constituyen la base teórica de la investigación y posterior propuesta. El primero de ellos expone ampliamente en qué consiste la metodología QFD y sus ventajas, la forma en la cual se construye tradicionalmente la Casa de la Calidad, la relación que tiene con la Ingeniería de Software y algunas herramientas útiles que facilitan la construcción de dicha casa.

El segundo tema se refiere al modelo de Kano, en el cual se establecen lineamientos para identificar aquellos requisitos realmente necesarios e importantes para darle base al proyecto. Así mismo, se definen los cuestionarios a través de los cuales se pretenden priorizar los requisitos con ayuda del cliente y, posteriormente, la forma como debe analizarse la información obtenida.

El tercer tema, trata el modelo CMMI y las áreas de proceso relacionadas con requisitos, pues se pretende cumplir con algunas prácticas y metas que este modelo propone, para darle a la metodología un nivel de calidad que le dé mayor valor a la misma.

El **Capítulo 3** constituye finalmente el centro del trabajo de investigación, pues en 6 grandes etapas se procura darle una identificación y un seguimiento eficaz a los requisitos. Cada una de estas etapas se encuentra dividida en tareas que se deben ejecutar una a una para garantizar el cumplimiento del objetivo que cada una de ellas intenta conseguir.

En el **Capítulo 4** se encuentra el planteamiento de un problema real al cual se le aplicará cada uno de los pasos de la metodología, explicados a lo largo del capítulo 3, teniendo en cuenta el cumplimiento de cada una de los aspectos que se proponen y que sea consistente con las prácticas del modelo CMMI.

Finalmente, se hace referencia al **trabajo futuro** que se dejará planteado, con el fin de que otras personas se motiven a darle continuidad a la metodología y pueda convertirse en todo un modelo de Ingeniería de Software para ser aplicado en grandes proyectos y que éstos logren ser exitosos. Luego se describen las **conclusiones** derivadas del desarrollo del trabajo, en las cuales se incluye el análisis de los resultados obtenidos al utilizar la metodología propuesta en un caso real, identificando así sus posibles ventajas y falencias.

CAPITULO 1: PROPUESTA DE INVESTIGACIÓN

1.1 ANTECEDENTES

1.1.1 PROBLEMAS EN LA INGENIERÍA DE SOFTWARE

La principal medida del éxito de un sistema de software es el grado en el cual dicho sistema alcanza el propósito para el cual fue planeado. La ingeniería de requisitos es la rama de la ingeniería de software encargada de producir ese propósito, por medio de la identificación de los stakeholders y sus necesidades, y de documentar esta información de tal forma que sea fácil de analizar, comunicar, y posteriormente implementar. Sin embargo, existe un gran número de dificultades inherentes a este proceso. Una de ellas es que los stakeholders (patrocinadores, usuarios y desarrolladores) pueden ser numerosos y estar distribuidos en diferentes localidades; adicionalmente, sus metas pueden variar o estar en conflicto, pues la perspectiva del problema puede cambiar, dependiendo del ambiente en el cual trabajan y de las tareas que llevan a cabo. También es posible que las metas no sean explícitas o que sean muy difíciles de expresar, e inevitablemente la satisfacción de éstas puede verse restringida por una gran variedad de factores fuera de control.

Debido a las dificultades mencionadas anteriormente, la recolección y la administración efectiva de los requisitos ha sido ampliamente reconocida por un gran número de expertos de la industria de software como una actividad crítica para la liberación exitosa de los sistemas de software. Adicionalmente, la Ingeniería de Requisitos es vital para el éxito en la planeación, la estimación y el control de cambios en un proyecto de software, pues ayuda a los equipos del proyecto a prevenir sobrecostos innecesarios.

Una investigación realizada por el Standish Goup [Standish Group, 2004], demostró que “el contrato”, entre clientes, usuarios finales y organizaciones

desarrolladoras de software en el cual se define qué es lo que se va a producir, es una fuente primaria de los riesgos del proyecto y de los defectos en el software. Una investigación adicional [Martin, 1984] demostró que el 82% del esfuerzo necesario para corregir un producto de software está enfocado en corregir defectos en los requisitos, el 13% es dedicado a arreglar errores de diseño, y sólo el 1% es dedicado a arreglar errores en el código.

Varios estudios, expuestos en [Mead et al., 2005], revelaron que la corrección de los defectos ocasionados durante la fase de ingeniería de requisitos puede costar hasta 200 veces más cuando se corrigen en fases posteriores. En la Figura 1 se muestra el costo relativo de corregir un error en una determinada fase. Como puede verse, la relación de corregir un error durante la fase de requisitos vs. la fase de mantenimiento es de 200:1.

Figura 1. *Costo Relativo de corregir un error*
Fuente: [Mead et al., 2005]

Además, se estimó que un error encontrado después de la puesta en producción de un producto software o en el proceso final de certificación, toma aproximadamente 8 horas para ser arreglado, mientras que un error encontrado en la fase de requisitos toma, en promedio, 15 minutos. En la Figura 2 (tomada de [Borland, 2005]) se puede apreciar que el esfuerzo relativo para arreglar un defecto aumenta cuando éste se deja propagar a través del ciclo de vida del software.

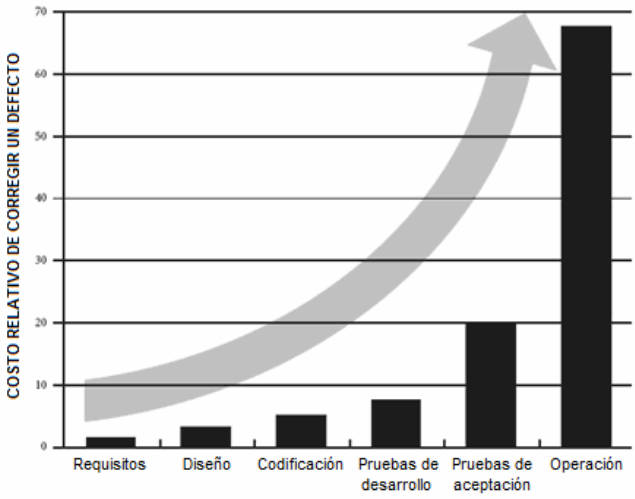


Figura 2. *Costo relativo para corregir un defecto en el ciclo de vida del software*
Fuente: [Borland, 2005]

Los siguientes datos, tomados de [Standish Group, 2004], muestran los problemas vividos actualmente en el campo de la Ingeniería de Software.

1. Aproximadamente dos terceras partes de los proyectos de software son considerados como fracasos, más de la mitad exceden el presupuesto, y el 84% sufren de demoras en la entrega. Sólo en Estados Unidos, los errores de software costaron a la economía casi US$ 60 billones al año, y en el pacífico de Asia el despilfarro en proyectos de software está estimado entre US$15 y US$20 billones.

2. En la Tabla 1 se pueden apreciar las estadísticas del estado final de los proyecto de software en una década.

   - **Éxito:** El proyecto es terminado a tiempo y dentro del presupuesto, con todas las características y funciones especificadas originalmente.

   - **Cuestionado:** El proyecto es terminado y es operacional, pero se sobrepasó el presupuesto, se entregó tarde y con menos características y funciones de las especificadas inicialmente.

- **Fracaso:** El proyecto es cancelado antes de ser terminado, o nunca es implementado.

	2004	2000	1994
Éxito	34%	28%	16%
Cuestionado	51%	49%	53%
Fracaso	15%	23%	31%

Tabla 1. *Estadísticas de resultados de proyectos de software en su estado final*
Fuente: [Standish Group, 2004]

3. El 71% de los proyectos de software que fracasan lo hacen por no tener un buen manejo de los requisitos.

4. Se ha estimado que el 85% de los defectos en el desarrollo de software se originan en los requisitos. Una vez que los defectos son embebidos en los requisitos, se aumenta el grado de dificultad para removerlos, además son muy difíciles de encontrar por medio de pruebas.

5. Los errores más comunes durante el análisis y la elicitación de requisitos son: suposiciones incorrectas (49%), omisión de requisitos (29%), inconsistencia en los requisitos (13%) y ambigüedades (5%).

Adicionalmente, Amador Durán, considerado uno de los más grandes exponentes de la ingeniería de requisitos, expone en su tesis doctoral [Durán, 2000] las siguientes ideas:

“En mayor parte, los problemas de desarrollo de software a nivel mundial están relacionados con la ingeniería de requisitos, y dentro de ésta, con las técnicas hasta ahora utilizadas para la elicitación de requisitos aspectuales eficientes.

Aunque se disponga de excelentes lenguajes de especificación de requisitos e incluso, aunque se consiga que los clientes y usuarios validen una determinada especificación, si no se han elicitado los requisitos correctos, todo el trabajo de

desarrollo terminará con un producto técnicamente correcto pero inútil, ya que no satisfacerá las necesidades que dieron origen a su desarrollo. “

Los problemas a los que se enfrenta la elicitación de requisitos son múltiples. En [Raghavan *et al.,* 1994] se establecen cinco grandes categorías de problemas dentro de la elicitación: problemas de *articulación*, de *comunicación*, de *limitaciones cognitivas*, de *conducta humana* y *técnicos*.

Problemas de articulación: Están relacionados con la expresión de sus necesidades por parte de clientes y usuarios y la comprensión de dichas necesidades por parte de los desarrolladores. Por ejemplo, algunos desarrolladores no escuchan apropiadamente a los clientes y usuarios, bien porque creen haber entendido sus necesidades rápidamente, bien porque se dedican a pensar inmediatamente sobre aspectos de implementación y no se ponen en el lugar de clientes y usuarios.

Problemas de comunicación: Los clientes, usuarios y desarrolladores tienen culturas y vocabularios muy diferentes, con la posibilidad de que los mismos términos tengan significados distintos en los distintos vocabularios, o que su significado se vea enormemente afectado por el contexto. Además, no sólo la cultura y el vocabulario son distintos, las preocupaciones sobre el sistema a desarrollar también suelen serlo. Mientras los clientes y usuarios suelen preocuparse por aspectos de alto nivel como facilidad de uso o fiabilidad, los desarrolladores suelen preocuparse por aspectos de bajo nivel como utilización de recursos, algoritmos, etc.

Problemas de limitaciones cognitivas: El ingeniero de requisitos debe tener un conocimiento adecuado del dominio del problema y no hacer suposiciones sobre ello, al igual que los clientes y usuarios no deben hacer suposiciones sobre aspectos tecnológicos. Muchas veces las ideas preconcebidas sobre una posible solución afectan a la forma en que se establece la definición del problema.

Problemas de conducta humana: Puede haber conflictos y ambigüedades en los roles que cada persona debe jugar en el proceso de elicitación. Por ejemplo, algunos clientes y usuarios piensan que los desarrolladores les harán todas las preguntas necesarias sobre el dominio del problema, y los desarrolladores piensan que los clientes y usuarios les proporcionarán toda la información necesaria sin necesidad de preguntar por su parte, con lo que pueden quedar aspectos sin tratar.

Otro problema de este tipo se presenta por la suposición o temor del usuario a que el sistema a desarrollar cambie su forma de trabajar o incluso ponga en peligro su puesto de trabajo, lo cual puede provocar que algunos usuarios retengan información o incluso saboteen el desarrollo, por ejemplo proporcionando información falsa.

Problemas técnicos: El software tiene que resolver problemas cada vez más complejos, por lo que sus requisitos son también cada vez más complejos y contemplan detalles cada vez más específicos del dominio del problema. Debido a que el hardware y el software cambian rápidamente, se pueden hacer asequibles requisitos que antes eran inabordables por su complejidad o por su coste.

En [Durán, 2000] se exponen dos conclusiones muy importantes sobre los problemas en la Ingeniería de Requisitos:

- Los problemas de la elicitación de requisitos no pueden resolverse de una forma puramente tecnológica, porque el contexto social es mucho más crucial que en las fases de programación, especificación o diseño.

- La ingeniería de requisitos es básicamente un problema de descubrimiento y comunicación de las necesidades de clientes y usuarios, por lo que la propiedad más importante de una especificación es que sirva como canal de comunicación entre los participantes en el proceso de ingeniería de requisitos (ver figura 3).

Figura 3. *La ingeniería de requisitos como un proceso de comunicación*

Fuente: [Durán, 2000]

1.1.2 BENEFICIOS DE LA METODOLOGÍA QFD

En [Hernández, 2003] se describe el QFD[1] (Quality Function Deployment) como "una metodología muy poderosa en el diseño y revisión de nuevos productos, que permite cumplir con los requisitos de los clientes mediante el despliegue de éstos a través de las diferentes fases del desarrollo de nuevos productos."

Esta metodología fue introducida en Japón en el año de 1960 por el profesor Yoji Akao, y su primera implementación fue en Mitsubishi Heavy Industries Kobe Shipyard en 1972. Para finales de los años 70's casi todas las empresas manufactureras en Japón estaban utilizando QFD.

"Esta metodología ha ganado gran relevancia en diferentes sectores de la industria, pues se ha encontrado en ella una estrategia para lograr satisfacer las necesidades del cliente. Esto se debe en gran medida a que su enfoque es "escuchemos la voz del cliente", lo cual le permite a la empresa ser más proactiva a los problemas de calidad, en vez de ser reactiva a ellos a través de la espera de las quejas del cliente." [Hernández, 2003].

[1] La metodología QFD será explicada con mayor detenimiento en el Capítulo 2.

"QFD trae un número de beneficios a las organizaciones que intentan incrementar su competitividad mejorando continuamente calidad y productividad. El proceso tiene los beneficios de ser orientado al cliente, eficiente en tiempo, orientado al trabajo en equipo y hacia la documentación. Estos beneficios se explican en los siguientes párrafos.

- **Orientado al cliente:** QFD requiere la recolección del input y retroalimentación del cliente. Esta información se traduce en un conjunto de requisitos específicos del cliente.

- **Eficiente en tiempo:** QFD puede reducir el tiempo de desarrollo porque se centra en requisitos específicos del cliente y claramente identificados. Debido a esto, no se desperdicia tiempo en desarrollar características que tienen poco o nulo valor para el cliente

- **Orientado al trabajo en equipo:** QFD es un enfoque orientado al trabajo en equipo. Todas las decisiones están basadas en el consenso e incluyen discusión a fondo y tormenta de ideas. Puesto que todas las acciones que deben tomarse se identifican como parte del proceso, los individuos ven dónde encajan en la escena completa, promoviendo de esta manera el trabajo en equipo.

- **Orientado a la documentación:** QFD fuerza el aspecto de la documentación. Uno de los productos del proceso QFD es un documento amplio y completo que reúne todos los datos pertinentes acerca de todos los procesos. Este documento cambia constantemente al conocer nueva información y descartar la obsoleta. Tener información actualizada sobre los requisitos del cliente y sobre los procesos internos es particularmente útil cuando ocurre un trastorno." [Goetsch y Davis, 1994].

En [Menks, Ahmed y Fu, 2000] se listan las siguientes ventajas[2] de usar QFD en el proceso de software (SQFD):

- Se involucran los clientes y usuarios en el proceso.
- Enfoque en las necesidades del cliente
- Constructor de equipos
- Mejora de la calidad de los productos o servicios
- Disminución del tiempo en el ciclo de desarrollo
- Disminución de costos
- Mayor productividad
- Reducción de cambios en la etapa de diseño
- Bueno para la comunicación, toma de decisiones y planeamiento
- Permite concentrar gran cantidad de información en un espacio pequeño

1.2 PLANTEAMIENTO DEL PROBLEMA

Con las evidencias expuestas hasta el momento, se pretende mostrar que, además de que los problemas relacionados con los requisitos son persistentes, difíciles de corregir y costosos, dichos problemas son los principales causantes de los fracasos en los proyectos de software, pues sus efectos sobre un proyecto pueden ser, entre otros, exceder el presupuesto, no cumplir con el cronograma, reducciones significativas en el alcance, liberación de aplicaciones de muy baja calidad, el producto entregado no tiene el uso esperado y, en algunos casos, se puede llegar a la cancelación del proyecto.

Ante esta situación, se hace apremiante desarrollar una propuesta para la utilización de la metodología QFD en el proceso de ingeniería de requisitos, con el fin de aprovechar las ventajas de dicha metodología en la elaboración de “un documento de requisitos de calidad”, el cual se tome como base para el desarrollo

[2] Las ventajas de usar QFD son explicadas con mayor detenimiento en la sección 2.1.3

de un sistema software que satisfaga, a cabalidad, todas las necesidades de los clientes y usuarios, y de esta forma se eviten los problemas enunciados anteriormente.

1.3 OBJETIVOS

1.3.1 OBJETIVO GENERAL

Elaborar una propuesta para la utilización de la metodología QFD (Quality Function Deployment) en el proceso de Ingeniería de Requisitos, con el fin de asegurar que las necesidades de los usuarios han sido entendidas y que el diseño corresponde con ellas.

1.3.2 OBJETIVOS ESPECÍFICOS

- Presentar un contexto del problema donde se justifique la necesidad de utilizar la metodología QFD en el proceso de Ingeniería de Requisitos.
- Definir un modelo de proceso para la utilización de la metodología QFD en la Ingeniería de Requisitos.
- Identificar el cumplimiento de las prácticas específicas del modelo CMMI en la metodología propuesta.
- Aplicar la propuesta metodológica en un caso práctico.

1.4 ALCANCE

El trabajo de investigación que se desarrollará se limita a la estructuración de un modelo conceptual que permita crear un marco de referencia para ser utilizado en el proceso de Ingeniería de Requisitos en cualquier organización de software. Adicionalmente, la metodología intentará resolver algunos de los problemas mencionados en la sección 1.1.1.

El tipo de investigación que se desarrollará será de tipo cualitativo. En [Cook, 1997] se menciona que existen grandes diferencias entre una investigación cualitativa y una cuantitativa. La tabla 2 muestra las principales diferencias identificadas entre los dos tipos de investigación antes mencionados.

Investigación Cualitativa	Investigación Cuantitativa
Fundamentado en la realidad, descriptivo e inductivo	Orientado a la comprobación, confirmación e hipotético deductivo.
Válido: datos “reales”, “ricos” y “profundos”	Fiable: datos “sólidos”
Subjetivo	Objetivo
Basado en la observación, percepción	Basado en la medición

Tabla 2. *Principales diferencias entre los dos tipos de investigación (cualitativa y cuantitativa)*
Fuente: [Cook, 1997]

1.5 HIPÓTESIS

La aplicación de la metodología QFD en el proceso de Ingeniería de Requisitos permite desarrollar un producto software que satisface totalmente las necesidades de los clientes y usuarios.

1.6 PREGUNTA DE INVESTIGACIÓN

¿Existirá alguna manera de mejorar el proceso de Ingeniería de Requisitos aplicando la metodología QFD, de tal forma que se pueda desarrollar un producto software que satisfaga totalmente las necesidades de los clientes y usuarios?

1.7 METODOLOGÍA DE LA INVESTIGACIÓN

La metodología a utilizar en este proyecto constará de siete etapas fundamentales, las cuales se muestran en la Figura 4; a continuación, se realiza una breve descripción de cada etapa.

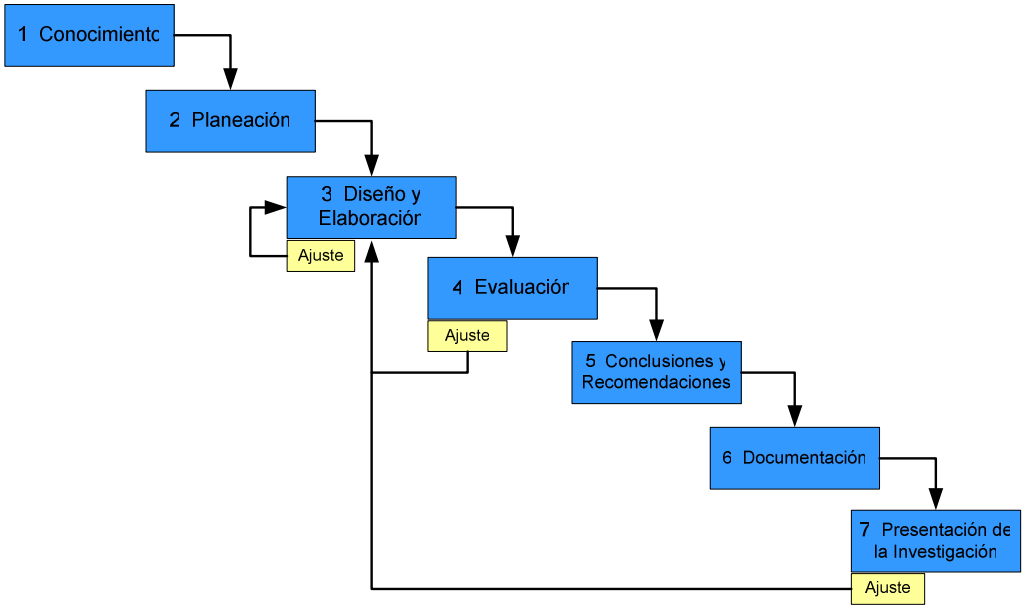


Figura 4. *Etapas de la metodología de investigación*

Fuente: Elaboración propia

1. **Conocimiento:** En esta etapa del proyecto, los participantes responsables estarán a cargo de realizar una exhaustiva búsqueda y preparación sobre todo lo relacionado con el tema de QFD y su posible aplicación en la elicitación de los requisitos para productos software, utilizando servicios de Internet y fuentes

de información, tanto primarias como secundarias. También se estudiarán las diferentes metodologías de elicitación utilizadas hasta el momento para poder conocer sus ventajas y desventajas.

2. **Planeación:** Esta etapa constará de la realización del plan del proyecto, una estimación de tiempos y de recursos, además de un cronograma más detallado para el proyecto. Se buscará lograr en este punto la definición de hitos en cada etapa para poder realizar evaluaciones en cada una de ellas. Así mismo, se realizará un estudio del proyecto para observar su viabilidad e identificar las posibles limitaciones.

3. **Diseño y Elaboración:** Después de conocer bien el tema a tratar, se elaborará un diseño de la metodología a desarrollar, con el fin de documentar todo lo que se va realizar durante el proyecto, de una forma clara y concisa. En esta etapa también se mirarán y utilizarán las ventajas de las metodologías de elicitación de requisitos utilizadas hasta el momento, para lograr consolidarlas en la elaboración de la nueva metodología utilizando QFD. En este punto se encuentra una subetapa de ajuste, la cual se utilizará si es necesario cambiar la planeación en medio del diseño o la elaboración; se espera no llegar a este punto, ya que esto podría causar reproceso y utilización excesiva de recursos.

4. **Evaluación:** Una vez se tenga elaborada la propuesta metodológica, se buscará realizar una evaluación objetiva del método desarrollado, al compararlo con los más utilizados en la actualidad. Con los resultados obtenidos, se sacarán las conclusiones pertinentes y si es del caso, se realizarán los ajustes necesarios, hasta lograr que la metodología pueda ser utilizada exitosamente por una organización de software.

5. **Conclusiones y Recomendaciones:** En esta etapa se presentarán los resultados obtenidos durante el proceso de investigación y se harán algunas recomendaciones. Como el alcance de esta investigación es limitado, a partir

de los resultados obtenidos se podrán generar líneas de investigación que la complementen.

6. **Documentación:** Todo la investigación se documentará con el fin de mantener evidencia del trabajo realizado para permitir la reproducibilidad de la misma y para respaldar los resultados obtenidos en ella.

7. **Presentación de la Investigación:** Concluida la investigación, se presentará al asesor del proyecto de grado, para su aprobación definitiva.

CAPITULO 2: MARCO TEÓRICO

2.1 DESPLIEGUE DE LA FUNCIÓN DE CALIDAD: QFD

2.1.1 HISTORIA DE LA METODOLOGÍA QFD

El QFD fue introducido en Japón a finales de los años 60 por los profesores Shigeru Mizuno y Yoji Akao, durante una época en la cual las industrias japonesas habían quebrado después de la Segunda Guerra Mundial. De modo que el desarrollo de productos basado en la copia y la imitación tuvo que cambiar por el desarrollo de productos basado en la originalidad.

Después de la Segunda Guerra Mundial, el Control Estadístico de Calidad (SQC, Statical Quality Control) fue introducido en Japón y llegó a ser la actividad central de calidad, principalmente en el área de manufactura. Luego esto fue integrado con técnicas del Dr. Juran, quien durante su visita en 1954 a Japón, enfatizó en la importancia de hacer control de calidad a la parte de administración del negocio y se basaron en las enseñanzas del Dr. Kaoru Ishikawa, quien dirigió el movimiento de la Campaña del Control de Calidad, convenciendo a la gerencia superior de las empresas sobre la importancia que tenía cada empleado que hacía parte de la empresa. Esta evolución fue soportada también por la publicación del Dr. Feigenbaum en 1961 sobre el Control Total de la Calidad[3] (TQC, Total Quality Control); de esta forma el SQC fue transformado en TQC en Japón durante el período comprendido entre 1960 y 1965.

[3] TQC es un sistema para optimizar la producción, basado en ideas desarrolladas por las industrias japonesas de los años 50. El sistema empezó con el concepto de círculos de calidad en grupos de 10 a 20 trabajadores, a quienes se les dio la responsabilidad de la calidad de los productos que ellos produjeron. Esto gradualmente se involucró en varias técnicas que envolvieron trabajadores y gerentes para maximizar productividad y calidad, incluyendo la supervisión cercana del personal y de la excelencia en el servicio al cliente.

Fue durante este tiempo que se presentó el concepto y la metodología del QFD. La industria japonesa de automóviles estaba en medio de un rápido crecimiento a través del desarrollo de nuevos productos y al mismo tiempo se presentaron dos situaciones que fueron las semillas a través de las cuales fue concebido el QFD:

- Las personas empezaron a reconocer la importancia del diseño de la calidad, pero cómo podría ser hecha, no se encontraba en los libros en esos días.
- Las compañías estaban listas para usar cuadros y gráficos para el control de calidad de los procesos, pero estos eran producidos en el sitio de manufactura después de que los nuevos productos estaban fabricados.

El propósito de los Profesores Mizuno y Akao era desarrollar un método de aseguramiento de la calidad que plasmara la satisfacción del cliente en un producto antes de que fuera manufacturado, ya que otros métodos de control de calidad estaban enfocados en arreglar un problema durante o después de que el producto fuera manufacturado.

La primera aplicación formal que consolidó el concepto de la calidad del diseño, fue en el año 1972 en los Astilleros de Kobe de Mitsubishi Heavy Industries (Japón), ayudados por Yoji Akao, Shigeru Mizuno y Yasushi Furukawa, quienes desarrollaron una estructura denominada **matriz de la calidad**, que sistematizaba la relación entre las necesidades de los clientes y las características de calidad inherentes a los productos. Esta matriz constituye actualmente el núcleo del QFD.

Todas las ideas y desarrollos propuestos fueron integrados y configurados en el despliegue de la calidad (QD, Quality Deployment). QD se define como una metodología que convierte las demandas de los usuarios en características de calidad, determina la calidad del diseño del producto terminado y sistemáticamente despliega esta calidad en los procesos y sus relaciones. [Akao, 1997].

Hubo otro elemento que se mezcló a la estructura del QFD desde el valor de la ingeniería, el cual mostró una manera de definir las funciones de un producto. Fue Katsuyoshi Ishihara quien expandió esto, pensando alrededor de las funciones de los procesos de negocio. El despliegue de estas funciones llegaron a conectarse con lo que después se denominó QFD. El Dr. Mizuno definió QFD como un despliegue paso a paso de unas funciones u operaciones de trabajo que incorporan calidad en detalle a través de la sistematización.

Hacia 1975 la Sociedad Japonesa de Control de la Calidad (JSQC)[4] constituyó un comité que se encargó de estudiar el QFD y formular su metodología. Este trabajo se realizó durante 13 años, al cabo de los cuales publicaron un estudio sobre las aplicaciones del QFD en 80 empresas japonesas, en las cuales se utilizaba la metodología para algunos objetivos como los siguientes:

- Establecer la calidad del diseño y la calidad planificada
- Realizar el benchmarking[5] de productos de la competencia
- Desarrollar productos que posicionaran una compañía por encima de su competencia
- Recolectar y analizar información sobre la calidad en el mercado
- Reducir el número de problemas iniciales de calidad
- Reducir el número de cambios en el diseño
- Reducir el tiempo de desarrollo de los productos o servicios y los costos asociados a éste
- Aumentar la participación del mercado
- Comunicar a procesos posteriores información relacionada con la calidad

[4] Fundada en 1970 con el objetivo de fomentar la investigación en tecnologías de administración de la calidad y promover su uso.

[5] Técnica gerencial basada en la comparación, en la cual se busca introducir las mejores prácticas de las empresas de la competencia que son más exitosas o de aquellas empresas reconocidas como líderes de la industria en otra organización, sin tener que pasar por los ensayos de prueba y error por los cuales ya pasó el líder.

Entre los años 1977 y 1984, Toyota adoptó la metodología logrando la reducción de costos asociados con el desarrollo del producto en un 61% y la disminución de tiempos en el ciclo de desarrollo en un 33%; adicionalmente, mejoró la calidad del producto en un 33%.

En el año 1983, el profesor Yoji Akao introdujo el QFD en Norte América a través de un pequeño artículo publicado en la revista Quality Progress. Este método se expandió y fue incorporado por compañías como Digital Equipment Corporation, Ford Motor Company, Hewlett-Packard, y en organizaciones de servicio como St. Clair Hospital, en Pittsburgh.

En el año 1984 comenzó a aplicarse en el desarrollo de software, introduciendo así el concepto de Despliegue de la función de la calidad en Software (SQFD), el cual representa la transferencia de la tecnología del despliegue de función de la calidad de un producto de manufactura a un desarrollo de software. Esta transferencia ha sido acompañada por modificaciones especiales (adición, cambios y/o eliminación de algunas matrices) al proceso del despliegue de función de la calidad para la implementación exitosa en el desarrollo de software. SQFD utiliza un conjunto de matrices que facilitan la captura de los requisitos del cliente y su conversión a requisitos del producto.

El QFD evolucionó a la misma vez con una idea del marketing: *"el diseño debe reflejar los gustos y deseos de los clientes más que el potencial tecnológico o las preferencias de los ingenieros de diseño"*. En esta idea, conocida como dualidad *product-out - market-in,* se pone en el centro de la escena al cliente y se obliga a las empresas a mejorar la comunicación y la planificación entre áreas funcionales como marketing e ingeniería; pues normalmente, estas áreas suelen trabajar aparte, concentrándose en los temas y expectativas de trabajo propios de cada una y perdiendo el objetivo principal de su trabajo: el cliente.

Durante los últimos años ha aumentado la gama de aplicaciones del QFD y su refinamiento metodológico. Se ha utilizado en el planeamiento estratégico, tanto

en operaciones de manufactura como de servicios, en empresas grandes y pequeñas; se ha aplicado a la comprensión de fenómenos organizacionales y a la mejora de servicios en el sector público y en la educación.

2.1.2 ¿QUÉ ES QFD?

Quality Function Deployment o *Despliegue de función de la calidad*, es una traducción literal de las palabras japonesas *hinshitsu kino tenkai*, que originalmente fueron traducidas como *Quality Function Evolution*.

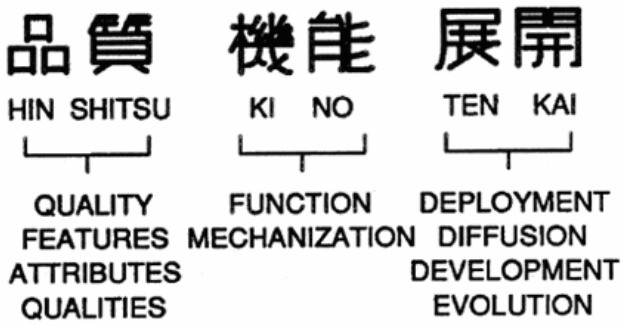


Figura 5. *Traducción de la palabra hinshitsu kino tenkai*
Fuente: [Karlsson, 1996]

Después, en el primer seminario de QFD en Estados Unidos, el patrocinador Masaaki Imai creía que el término *evolution* sugería impropiamente el significado de *cambio*, entonces *hinshitsu tenkai* se tradujo como Quality Deployment (despliegue de la calidad). Sin embargo, el uso de la palabra *function* en QFD ha sido mal interpretado, pues se refiere a un análisis funcional de las fases del proceso del negocio para mejorar la calidad del mismo proceso de desarrollo del producto y no se refiere a la función del producto en su contexto.

Entender las necesidades del consumidor y transformar esas necesidades en acciones es algo imprescindible en el mundo actual; QFD es un método de diseño

de productos y servicios que recoge la voz del cliente y le permite a una organización priorizar las necesidades del mismo, traduciéndolas en características de diseño y operación y encontrando respuestas innovadoras a esas necesidades.

El QFD permite transmitir a través de los procesos organizaciones las características o atributos de calidad que el cliente demanda, de esta forma se mejora la efectividad de los procesos para que puedan contribuir al aseguramiento de dichas características, permitiendo a las organizaciones sobrepasar las expectativas del cliente.

A lo largo del tiempo, este tema ha sido tratado por diferentes autores y especialistas en el tema, quienes han construido sus propias definiciones, que les permiten visualizar el tema desde diferentes perspectivas:

- Glen Mazur, en su manual de curso Comprehensive Quality Function Deployment Overview, define el QFD como un sistema de calidad que se focaliza en brindar valor a través de la búsqueda de necesidades del cliente tanto explícitas como implícitas, traducir esas necesidades en acciones o diseños y desplegar esto a través de la organización.

- Marvin E. González define el concepto como una metodología que traduce la voz del cliente en parámetros de diseño para que estos puedan desplegarse de forma horizontal dentro de los departamentos de planeación, ingeniería, manufactura, ensamble y servicio.

- El Doctor Georg Herzwurm lo define como un método desarrollado en Japón para proveer una forma de comunicación sistemática entre los clientes y desarrolladores, pero más informal y completa que solo las especificaciones de los requisitos. Esto permite focalizarse más en reconocer y satisfacer las necesidades de los clientes.

- La Asociación Latinoamericana de QFD define el Despliegue de Función de la Calidad como un sistema que busca focalizar el diseño de los productos y servicios en dar respuesta a las necesidades de los clientes. Eso significa alinear lo que el cliente requiere con lo que la organización produce.

Independientemente de que el QFD sea considerado como una herramienta, un método, un instrumento, una técnica o un sistema, los objetivos principales son cubrir 3 problemas mayores identificados en los métodos tradicionales existentes: Indiferencia con la voz del cliente, pérdida de la información, y diferentes individuos con diferentes funciones que trabajan en diversos requisitos.

Al igual que un sistema de calidad que implementa elementos de sistemas de pensamiento con elementos de Psicología y Epistemología, QFD provee un sistema de procesos comprensivo de desarrollo para entender las necesidades del cliente, entender cómo los clientes o usuarios finales llegan a estar interesados, eligen y están satisfechos; analizar cómo se entienden las necesidades del cliente, decidir cuáles características incluir, determinar cuál nivel de funcionamiento entregar y ligar de forma inteligente las necesidades del cliente con el diseño, desarrollo, ingeniería y manufactura.

QFD ayuda a las organizaciones a identificar las necesidades explícitas e implícitas, traducir estas necesidades en acciones y diseños y focalizar varias funciones de negocio hacia la realización de metas comunes.

La metodología puede ser usada para productos tangibles y servicios, incluyendo productos manufacturados, industrias de servicio, productos de software, proyectos de tecnología de información, desarrollo de procesos de negocio, gobierno, cuidado de la salud, iniciativas ambientales, entre otras aplicaciones.

Identifica los Qué(s), define los Cómo(s) y por medio de evaluación y análisis, sugiere métodos para la solución de un problema. De esta forma permite responder a tres preguntas básicas:

- ¿Cuáles son las características que el cliente desea?
- ¿Qué funciones debe tener el producto y qué funciones se deben usar para proveer el producto o servicio?
- Basados en los recursos disponibles, ¿Cómo se puede proveer lo que el cliente quiere?

De esta forma, QFD identifica los requisitos del consumidor y establece una metodología para asegurar que esos requisitos tengan una influencia positiva en el diseño del producto o en el desarrollo del proceso.

2.1.3 VENTAJAS DE QFD

QFD aporta un gran número de beneficios a las organizaciones que trabajan por mejorar continuamente su calidad, productividad y por ende, su competitividad.

2.1.3.1 Ventajas orientadas al Cliente y al producto

- Asegura la consistencia entre los requisitos del cliente y las características medibles del producto, y entre la planeación y el proceso de producción. Esto se refleja en clientes más satisfechos con los productos o servicios y en la disminución del número de manifestaciones de no conformidad.
- Traduce los requisitos del producto en requisitos (técnicos) significativos en cada etapa del proceso de desarrollo y producción.
- Identifica fortalezas y debilidades comparativas de los productos respecto a la competencia, lo cual le permite a la empresa compararse y concluir sobre la forma en que está satisfaciendo las necesidades de los clientes

- Ofrece un método estructurado para conectarse con todo el conocimiento en el desarrollo de nuevos productos en cualquier organización y facilita su administración y control.

2.1.3.2 Eficiente en consumo de recursos

- Logra que las cosas se muevan más rápido porque la planeación toma lugar en una etapa temprana, así se minimizan las interpretaciones equivocadas de prioridades y objetivos y se reducen costos de puesta en marcha.
- QFD se centra en requisitos del cliente, específicos y claramente identificados; debido a esto no se desperdicia tiempo en desarrollar características que tienen poco valor para el cliente.
- Reduce el número de cambios de ingeniería.
- Reduce el tiempo de ciclo en el diseño.

2.1.3.3 Ventajas orientadas al Trabajo en equipo

- QFD es un enfoque orientado al trabajo en equipo. Todas las decisiones están basadas en el consenso e incluyen discusión a fondo y tormenta de ideas. Puesto que todas las acciones que deben tomarse se identifican como parte del proceso, los individuos ven dónde encajan en la escena completa, promoviendo de esta manera el trabajo en equipo.
- Informa y convence a todos aquellos responsables de varias etapas del proceso de la relación entre la calidad de los resultados de cada fase (del proceso del QFD) y la calidad del producto final.
- Reúne personas de varias disciplinas y facilita la formación de equipos capaces de encontrar los requisitos del cliente.

2.1.3.4 Ventajas orientadas hacia la documentación

- QFD motiva hacia la documentación del conocimiento de mercadotecnia, diseño, ingeniería y manufactura del producto de una manera consistente y objetiva.

- Refuerza el interés en documentar los requisitos del cliente y los procesos internos, lo cual es particularmente útil cuando ocurre un trastorno. Esta documentación debe mantenerse actualizada con el fin de descartar información obsoleta.

2.1.4 LA CASA DE LA CALIDAD

La Casa de la Calidad es una matriz que traduce los requisitos del cliente en características técnicas necesarias que permitan satisfacerlas, plasmadas en el desarrollo de un nuevo producto. Básicamente es el centro y motor que conduce el proceso entero de QFD. De acuerdo con Hauser y Clausing, “es un tipo de mapa conceptual que provee los medios para el planeación y la comunicación interfuncional” [Tapke].

Hay muchas formas diferentes de la Casa de la Calidad, pero su habilidad para ser adaptada a los requisitos de un problema particular hace de ella un sistema robusto y confiable para usar. Su forma general se compone de los requisitos del cliente, requisitos técnicos, una matriz de interrelaciones, una matriz de correlación técnica y unas prioridades técnicas. A continuación se explicará cada uno de estos componentes.

2.1.4.1 Qué(s)

Los pasos iniciales en la formación de la Casa de la Calidad incluyen determinar, clarificar y especificar las necesidades de los clientes, por esto la Casa de la

Calidad comienza con el cliente, para establecer una identificación plena de lo que quiere; para esto es útil la pregunta: *"¿Qué quiere el cliente?".*

Los deseos definen la calidad de un producto y son las expresiones que los clientes utilizan para describir los productos y sus características deseables; esto es lo que se conoce comúnmente como la *Voz del cliente*. Estos deseos se organizan en una lista de Requisitos del Cliente de una forma muy general (Primer nivel), y más adelante se irán disgregando en necesidades más específicas.

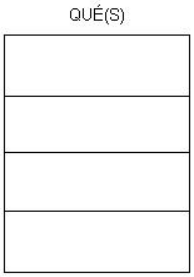


Figura 6. *Qué(s)*

Fuente: Elaboración Propia

La voz del cliente se debe representar generalmente en los propios términos del cliente para no correr peligro de mal interpretar lo que ellos desean. Tomando como ejemplo, un caso para el mejoramiento del diseño de una página Web, las necesidades que el equipo investigador pudo identificar, fueron:

Primarias	Secundarias	Terciarias
Apariencia	Gráficas	Se deben usar más gráficas para hacer las páginas más interesantes
	Texto	Los textos son fáciles de leer
	Diseño de la página	El diseño de la página es uniforme y estandarizado
Contenido	Información	La información es fácil de encontrar
	Links	Mayor número de links a otros sitios y correo electrónico. Buena integración de los links dentro del texto.
Otros	Velocidad	Las páginas son rápidas para cargar

Tabla 3. *Ejemplo de los Qué(s) para el diseño de una página Web*
Fuente: [Tan, Xie y Chia, 1997]

2.1.4.2 Cómo(s)

Los Qué(s) son trasladados en los correspondientes cómo(s). Estos cómo(s) son usualmente conocidos como *requisitos técnicos del producto*. Representan los medios por los cuales una compañía responde a lo que el cliente desea y necesita. Cada requisito técnico puede afectar una o más voces del cliente y deben ser algo que trabaje por satisfacer la voz del cliente y, además, deben ser medibles.

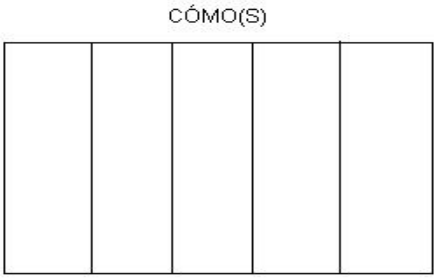


Figura 7. *Cómo(s)*
Fuente: Elaboración Propia

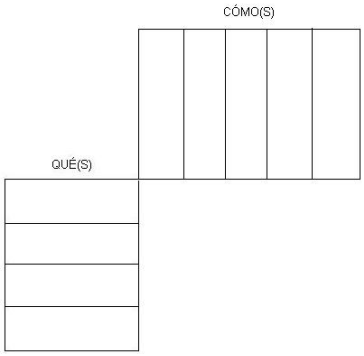


Figura 8. *Qué(s) vs. Cómo(s)*
Fuente: Elaboración Propia

Este proceso de identificar Qué(s) y Cómo(s) se repite tantas veces como sea necesario, donde los Cómo(s) se van convirtiendo en nuevos Qué(s) (ver figura 9). Esto se realiza con el fin de depurar los Cómo(s) hasta un nivel ejecutable, asegurando que los objetivos no se pierdan.

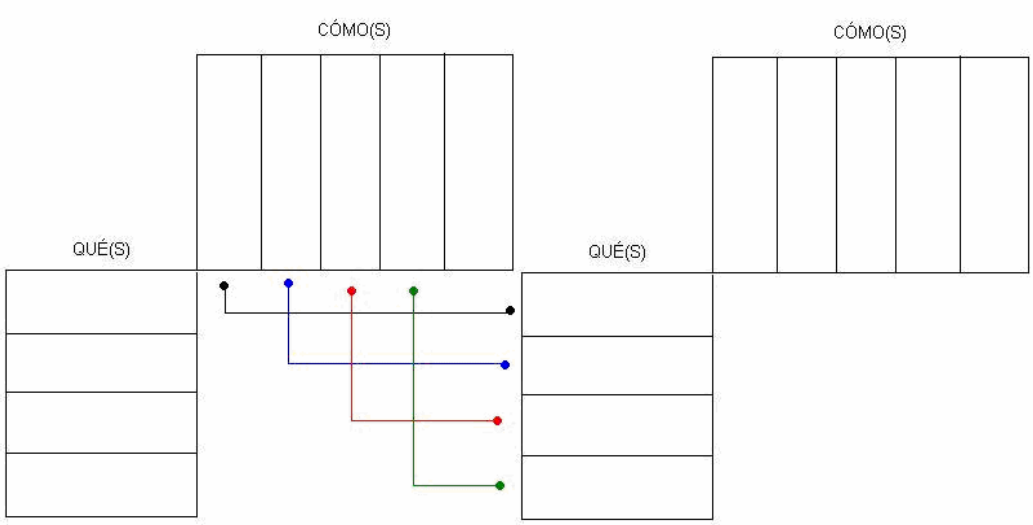


Figura 9. *Conversión de Cómo(s) en Qué(s)*

Fuente: Elaboración Propia

Para identificar los Cómo(s) es útil realizar sesiones de lluvias de ideas con personas que sepan acerca del problema que se desea solucionar. Continuando con el ejemplo del mejoramiento del diseño de una página Web, los requisitos técnicos relacionados con la voz del cliente serían:

- Uso de gráficas
- Incremento de la información en las páginas
- Formato del texto
- Ortografía y gramática
- Diseño estándar de páginas
- Links integrados al texto
- Proveer información que pueda descargarse a través de los links
- Proveer links que permitan regresar a la página principal
- Organización de páginas (índice, contenidos, grupos)
- Links adecuados y actualizados
- Disminución del tamaño de las páginas para incrementar la velocidad de carga
- Velocidad del computador y comunicación

2.1.4.3 Relaciones

Luego de establecer los Qué(s) y los Cómo(s), la construcción de la Casa de la Calidad continúa con el establecimiento de las relaciones entre la voz del cliente y los requisitos técnicos.

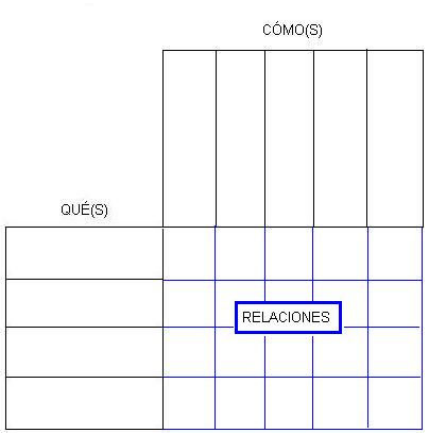


Figura 10. *Relaciones entre los Qué(s) y los Cómo(s)*
Fuente: Elaboración Propia

La matriz relacional

Para construir esta matriz es necesario establecer si las relaciones existen entre cada Qué y cada Cómo. Estas relaciones pueden ser categorizadas como **fuertes, medianas y débiles.** Para representar esta categorización se podrían utilizar, por ejemplo, los símbolos mostrados en la figura 11.

Figura 11. *Tipos de Relaciones*
Fuente: Elaboración Propia

La asignación y clasificación de las relaciones deben ser realizadas de forma cuidadosa y verificada un número considerable de veces. Normalmente son

identificadas por un equipo compuesto por personas de diferentes campos con el fin de que compartan su experiencia en la construcción de la Casa de la Calidad. En la figura 12 se muestra la Casa de la Calidad para el ejemplo del diseño de la página Web con la matriz de relaciones.

TERCIARIO	Velocidad de los computadores y comunicaciones	Incremento de la información en las páginas	Formato del texto	Ortografía y gramática	Diseño estándar de páginas	Links integrados al texto	Proveer información que pueda descargarse	Proveer links que permitan regresar a la página principal	Organización de páginas (índice, contenido y grupos)	Links adecuados y actualizados	Disminución del tamaño de las páginas	Velocidad de los computadores y comunicaciones	**Número de Fila**	**Peso de la fila**
Usar más gráficas para hacer la página web más interesante	◎				□								1	6
Texto fácil de leer		□	◎	○									2	3
El diseño de la página es uniforme y entendible			□		◎			□					3	1
Suficiente información provista por la página web		◎											4	2
Información fácil de ubicar					□				◎				5	4
Mejores links. Por ejemplo, que permitan conectarse a otros sitios, para enviar correo							○	○		◎			6	8
Buena integración de los links en el texto, en vez de links "Click aquí"						◎	○						7	5
Rápida velocidad de carga de las páginas	○	○									○	◎	8	7
Número de columna	1	2	3	4	5	6	7	8	9	10	11	12		
Peso de la columna														
Importancia														

Figura 12. *Ejemplo de relaciones entre los Qué(s) y los Cómo(s)*
Fuente: [Tan, Xie y Chia, 1997]

La matriz de correlación: El techo de la casa

Esta matriz ayuda a los ingenieros de software o los ingenieros de diseño a especificar varias características de ingeniería que han sido mejoradas. El techo contiene la información más crítica pues en él se refleja la correlación entre los requisitos técnicos; es decir, permite conocer el efecto que un incremento o mejora de un requisito técnico tiene sobre los demás. Las correlaciones se representan

por “O” para indicar que existe una correlación positiva y “X” para indicar que existe una correlación negativa.

Una correlación positiva entre dos características técnicas, significa que el incremento o disminución de una de ellas influye de la misma manera en la otra característica. En la figura 13 puede verse que al incrementar el uso de formatos de texto en las páginas Web, se incrementa también su diseño estándar, por esta razón existe una correlación positiva (O) entre estas dos características.

Por el contrario, una correlación negativa simboliza que el incremento en una característica técnica, disminuye directamente la presencia o desarrollo de la otra característica. Por ejemplo, una correlación negativa puede estar entre incrementar texto e información en las páginas y la velocidad de las páginas y comunicaciones, como se muestra en la figura 13.

2.1.4.4 El Cuánto

Para terminar de estructurar la Casa de la Calidad, se construye una ponderación para determinar una correlación específica entre un requisito del cliente y una característica técnica. Esta ponderación se muestra en la última fila de la matriz de la calidad.

Este valor se calcula multiplicando la prioridad dada por los clientes a cada requisito por una asignación que se hace, correspondiente a los símbolos que representan el grado de correlación y sumando el resultado de todas las multiplicaciones realizadas sobre esa columna.

Un valor de ponderación total alto sugiere la necesidad de dirigir las actividades de diseño o desarrollo tecnológico hacia este resultado.

En la figura 13 se muestra la Casa de la Calidad terminada para el ejemplo del diseño de la página Web.

Co-relaciones
O Positiva
X Negativa

Relaciones
◎ Fuerte
○ Moderada
□ Débil

PRIMARIA	SECUNDARIA	TERCIARIA	Velocidad de los computadores y comunicaciones	Incremento de la información en las páginas	Formato del texto	Ortografía y gramática	Diseño estándar de páginas	Links integrados al texto	Proveer información que pueda descargarse a través de los links	Proveer links que permitan regresar a la página principal	Organización de páginas (índice, contenido y grupos)	Links adecuados y actualizados	Disminución del tamaño de las páginas para incrementar la velocidad	Velocidad de los computadores y comunicaciones	**Número de Fila**	**Peso de la fila**
Apariencia	Gráficas	Usar más gráficas para hacer la página web más interesante	◎				□								1	6
	Texto	Texto fácil de leer		□	◎	○									2	3
	Diseño de Página	El diseño de la página es uniforme y entendible			□		◎			□					3	1
Contenido	Información	Suficiente información provista por la página web		◎											4	2
		Información fácil de ubicar					□				◎				5	4
	Links	Mejores links. Por ejemplo, que permitan conectarse a otros sitios, para enviar correo							○	○		◎			6	8
		Buena integración de los links en el texto, en vez de links "Click aquí"						◎	○						7	5
Otros	Velocidad	Rápida velocidad de carga de las páginas	○	○									○	◎	8	7
		Número de columna	1	2	3	4	5	6	7	8	9	10	11	12		
		Peso de la columna	51	34	16	9	23	25	39	25	20	40	21	35		
		Importancia	1	5	11	12	8	6	3	6	10	2	9	4		

Figura 13. *Ejemplo de una Casa de la Calidad*

Fuente: [Tan, Xie y Chia, 1997]

Aunque existen otros componentes de la Casa de la Calidad, tales como la matriz de conflictos en los Qué(s), el análisis de la competencia y la evaluación de los requisitos técnicos, estos componentes no fueron explicados, puesto que no serán contemplados dentro de la propuesta metodológica.

2.1.5 SOFTWARE QUALITY FUNCTION DEPLOYMENT (SQFD)

El éxito de QFD en manufactura condujo a un interés creciente por la aplicación de técnicas orientadas hacia la calidad en el desarrollo de software. La calidad en el software se está convirtiendo en una cuestión cada vez más clave del desarrollo de software, así como en manufactura lo es el desarrollo de un producto.

SQFD al igual que QFD, se enfoca en mejorar la calidad del desarrollo de los procesos de software. Implementa técnicas de mejora de la calidad durante la fase de elicitación de requisitos del proceso de especificación de requisitos de software.

Es una técnica anticipada a la elicitación de requisitos, pues precede cualquier ciclo vital del desarrollo de software. Una meta fundamental de SQFD es traducir las necesidades del cliente en productos de óptima funcionalidad. SQFD ha desarrollado algunas características únicas para resolver esta meta. Por ejemplo, la Casa de la Calidad en QFD requiere que todos los participantes estén en la misma localización física al mismo tiempo. Por el contrario, en el desarrollo de software, los clientes y los desarrolladores pueden estar separados.

Compañías como AT&T y Laboratorios Bell han usado SQFD para identificar características técnicas y rasgos en varios proyectos de software, encontrando ciertos beneficios acerca de estos proyectos, tales como:

- Mejor atención a las perspectivas del cliente
- Mejor comunicación entre los departamentos

- Facilidad para el análisis de la información
- La estructura de la información mostraba lo que se había decidido y el por qué.

Otras empresas han reportado el uso de QFD en diferentes proyectos; en algunos se conocen resultados y en otros han quedado en curso porque se han encontrado ciertas dificultades para aplicarlos [Hagg, 1992]. Por ejemplo, IBM reportó en 1991 un caso de estudio del uso de QFD aplicado a un cajero automático. En este reporte, las matrices se mostraban correlacionando las necesidades del cliente con las características del producto, modos de fallo de las necesidades del cliente y modos de fallo de las necesidades del producto. En Junio de 1991, IBM anunció que el entrenamiento con SQFD había sido completado por 2800 ingenieros de software.

Texas Instuments reportó que el QFD fue utilizado para identificar y priorizar los requisitos del cliente para actividades y productos a ser liberados en 1991. De esta forma, QFD fue usado para aumentar requisitos de alto nivel para los productos y servicios requeridos que apoyaron las mejoras de proceso de la ingeniería de software. Sin embargo, el reporte no menciona nada sobre si SQFD había sido utilizado como una entrada en el ciclo de desarrollo del sistema.

2.1.5.1 Características y beneficios principales de SQFD:

- Productividad creciente del analista y el programador
- Pocos cambios en el diseño
- Reducción del número de errores que pasan de una fase a otra
- Sistemas de software de calidad que satisfacen los requisitos del cliente
- Sistemas de software que requieren menos mantenimiento, permitiendo que parte del presupuesto del proyecto sea aplicado a otro proyecto
- Mejora la comunicación entre los departamentos, implicando clientes, vendedores, directores, desarrolladores, arquitectos del sistema, ingenieros, analistas de negocio, personas encargadas de pruebas, y otros participantes.

Las comunicaciones están sistematizadas y bien documentadas de una manera concisa para la trazabilidad y la integridad de los procesos, tanto de diseño como de desarrollo.

2.1.5.2 Herramientas y Técnicas de SQFD

- *Co-Authoring (En el mismo lugar – En momento diferente):* Técnica popular con un grupo de desarrolladores de programas, manuales y especificaciones. Divide el proyecto en diferentes procedimientos, capítulos y sesiones y los asigna a diferentes personas.
- *Sistema de Mensajes (Diferentes lugares – Momento diferente):* Permite a los participantes hacer un intercambio asíncrono de información. La información se intercambia a través de Internet.
- *Sitio de reunión (Mismo lugar – Mismo momento):* Se utilizan las técnicas de lluvia de ideas y entrevistas.
- *Conferencia (Diferentes lugares – Mismo momento):* Permite a las personas interactuar en tiempo real a través de conferencias en tiempo real y teleconferencias.

2.1.5.3 QFD y el Paradigma Orientado a Objetos (OO)

En 1995, Walter M. Lamia del Instituto de Ingeniería de Software (SEI), realizó una publicación en la cual describía muchas técnicas para integrar QFD con el diseño de software orientado a objetos.

Terminología

La terminología de QFD y OO difieren, sin embargo la literatura existente ha encontrado que los conceptos que se manejan en las dos áreas son virtualmente idénticos.

ORIENTADO A OBJETOS	QFD
Actores	Clientes
Casos de Uso	Función
Énfasis en modelar el usuario del sistema	Voz del cliente

Tabla 4. *Terminología: QFD vs. OO*

Fuente: [Lamia, 1995]

Énfasis

El paradigma orientado a objetos se centra en la modelación de los usuarios del sistema. Estos modelos son usados hacia el diseño del sistema. Este concepto de hacer énfasis en el usuario es paralelo a la doctrina del QFD de la *"voz del cliente"* en la cual comprender y desplegar la voz del cliente a través del diseño del producto y su entrega es clave en el proceso QFD.

Modelos de Casos de Uso

Los casos de uso pueden ser una herramienta para identificar y comunicar descripciones funcionales, ya que es la técnica que más se acerca para solucionar la falta de herramientas para elaborar un modelo funcional.

Matrices QFD-OO

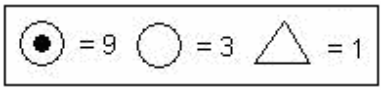
Existe una serie de matrices QFD que pueden ser relacionadas con el análisis orientado a objetos. A continuación se mostrarán algunas de ellas, la cuales más adelante serán retomadas, aplicándolas a un caso real.

Actor x Caso de Uso

- Muestra las funciones y operaciones en las que cada actor participa.
- Los casos de uso representan una función que un sistema debería realizar por cada actor.
- Relaciona las diferentes necesidades con varios actores.

- A través del establecimiento de una función de peso, se calculan los pesos para cada caso de uso.

  Ejemplo:



Actores	Buscar un libro	Encontrar un libro en un estante	Revisar un libro	**Peso de Fila**
Bibliotecario	○	◉	◉	**21**
Estudiante	◉		◉	**18**
Peso	**12**	**9**	**18**	

Figura 14. *Ejemplo matriz Actor x Caso de Uso*

Fuente: [Lamia, 1995]

Calidad requerida x Casos de uso

Esta matriz es muy importante, ya que provee la voz del cliente. Es útil en los casos donde el uso humano es una parte importante de la elicitación del sistema.

Calidad Demandada	Buscar un libro	Revisar un libro
Debe ser rápido	◉	
Debe ser a prueba de errores		◉

Figura 15. *Ejemplo matriz Calidad Requerida x Caso de Uso*

Fuente: [Lamia, 1995]

Casos de uso X objetos

- Muestra cuáles objetos participan en cada caso de uso
- Particularmente útil para recolectar información para ser usada en la definición completa de modelos de estado y modelos de proceso

Caso de Uso	Bibliotecario	Libro	Terminal
Buscar un libro		⊙	⊙
Revisar un libro	⊙		⊙

Figura 16. *Ejemplo matriz Casos de Uso x Objetos*

Fuente: [Lamia, 1995]

2.1.6 OTRAS HERRAMIENTAS DE QFD

2.1.6.1 Diagramas de Afinidad

"Herramienta administrativa que sirve para organizar grandes listados de ideas en grupos naturales, de acuerdo con criterios establecidos por un equipo de trabajo" [Michael Trick, 1996].

Su origen se debe al antropólogo japonés Kawakita Jiro, por esto el diagrama de afinidad también se conoce como KJ. Es una herramienta que permite organizar la información recopilada en sesiones de Lluvias de Ideas. Su objetivo es reunir hechos, opiniones e ideas que se encuentran desorganizadas, analizando los datos por afinidad mutua.

Casos en los que se utiliza:

- El problema o situación que se está analizando es muy complejo o de difícil comprensión y aparentemente se encuentra desorganizado.
- Se requiere de la participación y soporte de todo el equipo de trabajo.
- Se quiere determinar los temas claves de un gran número de ideas y problemas

Ventajas:

- Permite generar ideas creativas que luego se organizan y relacionan, ayudando a comprender el problema
- Ayuda a abordar un problema de manera directa
- Promueve la comunicación entre los integrantes de un equipo de trabajo
- Permite construir ideas entre los miembros de un equipo, basados en ideas generadas de forma individual
- Permite la identificación de relaciones no convencionales entre ideas

Desventajas:

- Dificultad para organizar las ideas que a veces son inciertas o no se comprenden
- Dificultad para evolucionar y dejar a un lado las ideas existentes y adaptarse a las nuevas formas de pensar
- Debe promoverse el trabajo en equipo con el fin de que haya una colaboración y entendimiento mutuo
- El conjunto de datos a organizar puede llegar a ser muy amplio

La esencia de este diagrama consiste en agrupar las diferentes ideas, opiniones, temas, expresiones, en función de la relación que tienen entre sí. Un ejemplo de este diagrama se muestra en la figura 17.

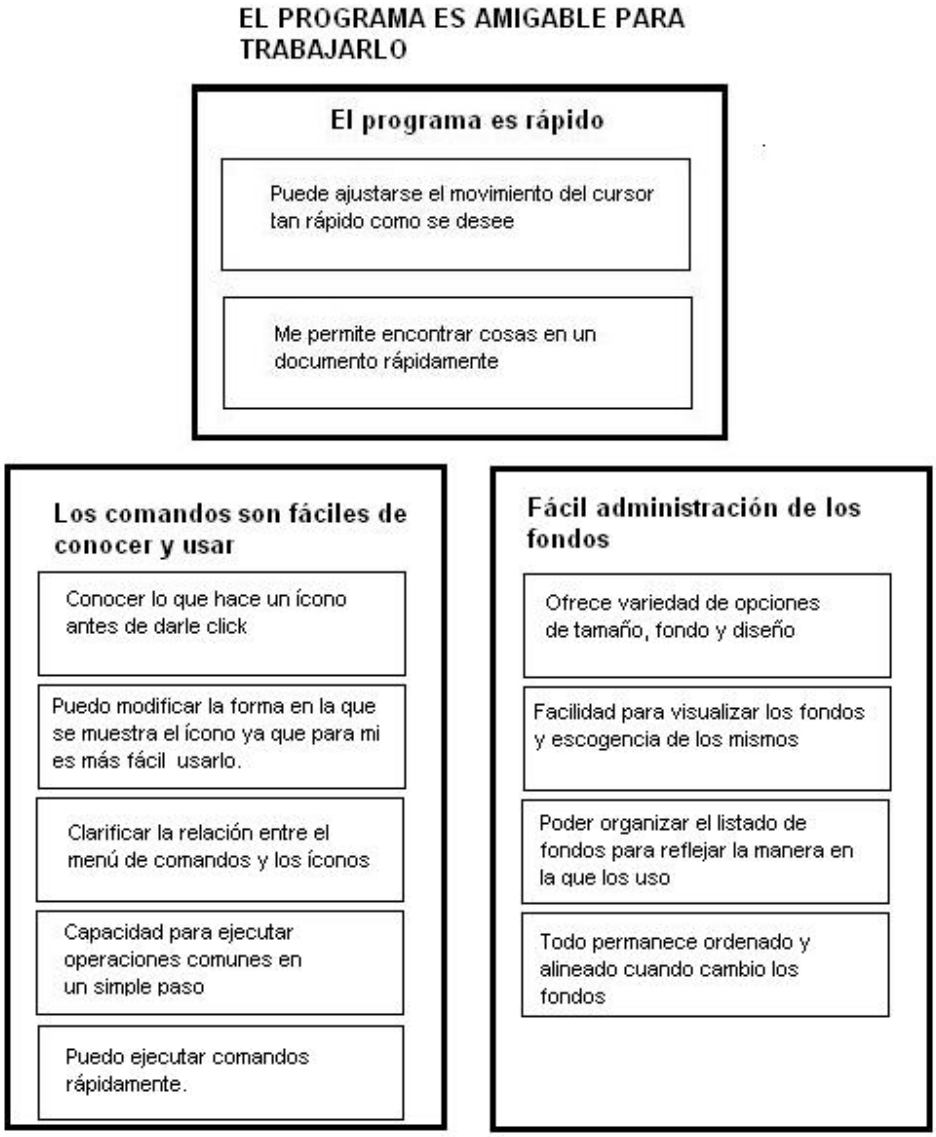


Figura 17. *Ejemplo de un diagrama de afinidad*

Fuente: [Harty, 2001]

2.1.6.2 DIAGRAMAS DE ÁRBOL:

Herramienta de análisis que permite establecer relaciones causales entre las diferentes necesidades del cliente que se lograron extraer, asimismo permite detectar necesidades del cliente que están implícitas o no consideradas. Luego

estas necesidades se comparan y priorizan para identificar aquellas que se encuentran en niveles iguales.

Se representa a través de una estructura en la que se parte de la meta general (el tronco) y se continúa con la identificación de niveles de acción más precisos (las ramas). Las ramas del primer nivel constituyen medios para alcanzar la meta, pero a su vez estos medios también son metas y objetivos intermedios para alcanzar la meta, que a su vez son metas y objetivos intermedios que se alcanzarán con un nuevo nivel. De esta forma se va repitiendo la estructura hasta lograr un grado de concreción sobre los métodos a emplear.

Por medio de técnicas como la lluvia de ideas se extraen las necesidades de la voz del cliente (verbalizaciones) y por medio de diagrama de afinidad, se agrupan por niveles, por ejemplo Necesidades de nivel 1 (necesidades generales), Necesidades nivel 2, 3, …n (necesidades más específicas). Lo recomendado es tener hasta 3 niveles de necesidades.

De acuerdo con [Mazur, 2000], habrá aproximadamente de 5 a 10 necesidades de nivel 1 que se subdividirán en 20 a 40 necesidades de nivel 2, 40 a 80 necesidades de nivel 3, etc.

En la figura 18 es posible visualizar un diagrama de árbol a través del cual se analizaron las actividades necesarias para desarrollar un producto capaz de obtener éxito en el mercado.

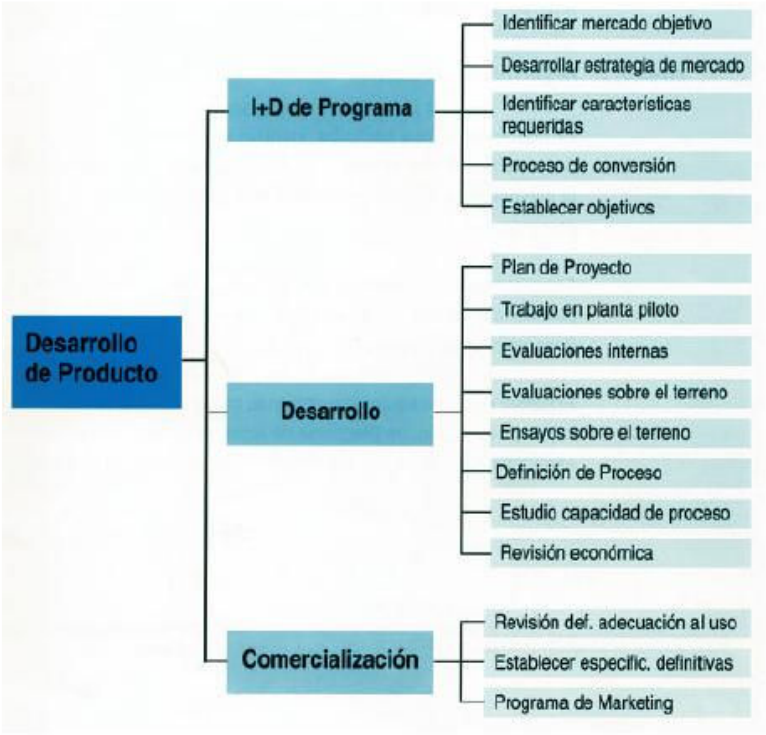


Figura 18. *Ejemplo de un diagrama de árbol*

Fuente: http://www.fundibeq.org/metodologias/herramientas/diagrama_de_arbol.pdf

Ventajas:

- Estimula a los integrantes del equipo a ampliar su modo de pensar para crear soluciones
- Mantiene al equipo vinculado a las metas y submetas generales de una tarea
- Permite descomponer una meta en fases u objetivos concretos

2.1.6.3 MATRICES Y TABLAS:

Las matrices de relaciones permiten analizar la relación existente entre las necesidades del cliente y los parámetros de diseño. Son usadas para documentar las percepciones del equipo sobre las interrelaciones.

2.2 MODELO DE KANO

La identificación e interpretación de la "**voz del cliente"** es el primer paso en el proceso de gestión de la calidad. La buena interpretación de estas necesidades permite generar un producto de alta calidad, que satisfaga completamente al cliente. Sólo después de haberse identificado las necesidades o exigencias, latentes o explícitas del cliente, podrá realizarse la traducción a **"funciones"** y de éstas a la definición de características técnicas, que serán la base para hacer operacional la definición del producto o servicio a desarrollar.

El proceso de **"escucha"** de las necesidades del cliente presenta un problema fundamental: la dificultad de lograr una adecuada comunicación entre el usuario final del producto o servicio y la persona encargada de interpretar estos requisitos y convertirlos en tangibles. El modelo de Kano (1984) es una herramienta eficaz en este sentido debido a que permite realizar una clasificación singular de los requisitos del cliente hacia el producto y su posterior caracterización en el diseño industrial.

Hacia fines de la década de 1970, Noriaki Kano, académico japonés de la Universidad de Tokio, amplió el concepto de calidad utilizado hasta entonces, que juzgaba a la calidad de los productos sobre una sola escala, de "bueno" a "malo".

Siguiendo en parte la tradición de las teorías de Herzberg sobre "motivación e higiene" [Valdés, 2007], Kano utilizó dos dimensiones para evaluar la calidad: (1) el grado de rendimiento de un producto y (2) el grado de satisfacción del cliente que lo utiliza, ampliando así el concepto de calidad utilizado hasta entonces.

Trabajando sobre este plano cartesiano de funcionalidad-satisfacción, Kano definió tres tipos de calidad: calidad obligatoria, calidad unidimensional (o de rendimiento) y calidad atractiva, exponiendo además 3 supuestos para justificar su teoría, los cuales son:

1. Las ideas “invisibles” de la calidad pueden hacerse visibles.

2. Para algunos requisitos del cliente, la satisfacción es proporcional a la funcionalidad del producto.

3. Algunos requisitos del cliente no son unidimensionales, existen también elementos “obligatorios” y “atractivos”.

La creencia de que la satisfacción de los clientes era proporcional a la funcionalidad de las diversas características del producto, lo que se ha llamado “unidimensionalidad”, estuvo presente durante mucho tiempo, aunque esta proporcionalidad no siempre se manifiesta, y en algunos casos el cliente está menos satisfecho cuando el producto es poco funcional, pero no está más satisfecho si la funcionalidad supera cierto nivel (requisitos “obligatorios”). En otros casos, el cliente no está insatisfecho por la falta de funcionalidad, pero si esta se incrementa su satisfacción aumenta (requisitos “atractivos”).

Kano ideó un cuestionario que clasifica las características de un producto para facilitar su diseño y orientar la estrategia de mercadotecnia.

2.2.1 TIPOS DE REQUISITOS DE LOS CLIENTES

El método mide la relación entre la funcionalidad de los productos y la satisfacción que esta funcionalidad les brinda a los clientes. La funcionalidad es una medida del grado en que un producto cumple con sus propósitos utilitarios en una cierta dimensión.

El modelo de Kano establece, para cada requisito del cliente, la relación entre satisfacción y funcionalidad y permite discriminar entre los requisitos clasificándolos en “atractivos” (delighters), “obligatorios” (basics) y “lineales” (performance).

La figura 19 ayuda a comprender mejor esta agrupación. En ella se han dibujado tres tipos ideales de atributos, en función de la relación entre funcionalidad y satisfacción.

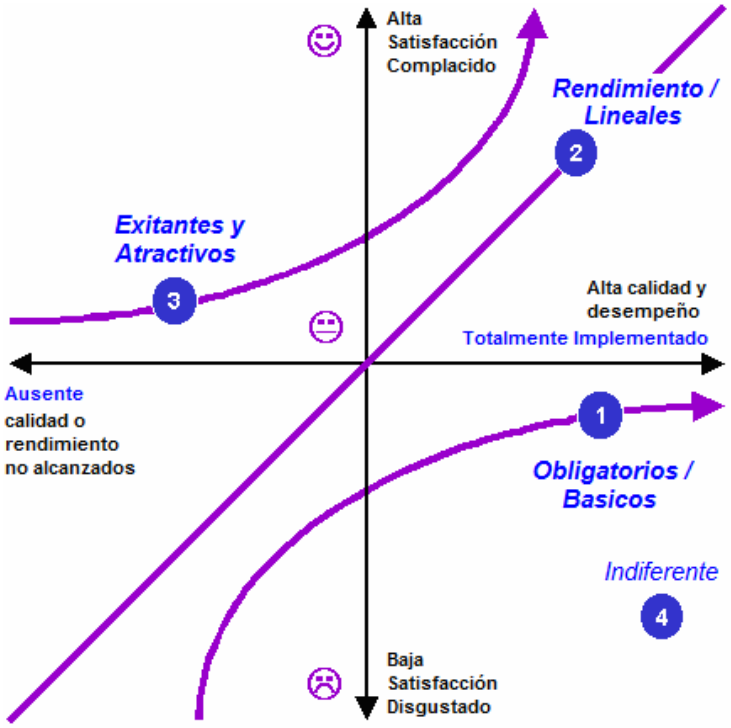


Figura 19. *Tipos de Requisitos del cliente*

Fuente: [NPD-NET, 2005]

Los requisitos atractivos son aquellos que, por debajo de cierto umbral de funcionalidad, mantienen un nivel de satisfacción relativamente bajo y constante, pero que, una vez superado ese umbral, producen un aumento significativo de la satisfacción. Por ejemplo, en el caso del software, la satisfacción que produce la recomendación de un experto en algún medio es atractiva, puesto que a muchos clientes potenciales no les preocupa que pocas personas recomienden el producto, pero, a partir de cierto momento, cuando el número de recomendaciones aumenta y ellas se hacen sentir en el mercado, también se incrementa la satisfacción derivada de utilizar el producto.

Los requisitos unidimensionales se caracterizan porque la satisfacción que producen aumenta de modo aproximadamente proporcional al nivel de funcionalidad. Responden a la percepción tradicional de la relación entre funcionalidad y satisfacción: a mayor funcionalidad, se observa una mayor satisfacción, de aquí el nombre alternativo de satisfactores para estos requisitos. Se debe entender que el concepto de funcional es mucho más amplio, no es solo en características que realicen alguna función, por ejemplo para el caso de un producto químico, más funcionalidad podría significar “más preocupación por el medio ambiente” y deriva su sentido de las crecientes exigencias y requisitos de los clientes en este tema.

Por ultimo, los requisitos obligatorios son aquellos que, hacia las gamas bajas de funcionalidad, aumentan la satisfacción en relación directa con la funcionalidad pero que, superado cierto umbral, dejan de producir un incremento importante en la satisfacción.

2.2.2 CUESTIONARIOS

Noriaki Kano ideó un conjunto de preguntas que deben ser realizadas a los participantes para cumplir con la clasificación propuesta por él. Estos cuestionarios asociados con una tabla de evaluación, permiten realizar la especificación de los requisitos de forma adecuada.

Cada uno de los cuestionarios siempre cuenta con un número par de cuestionamientos relacionados con los requisitos del cliente. La razón por la cual las preguntas vienen en pares es porque, para cada requisito, se realiza una pregunta funcional y otra disfuncional. La forma de realizar la consulta sería así:

- ¿Cuál sería su sentimiento si la característica X está presente en el producto? (requisitos funcionales)

- ¿Cuál sería su sentimiento si la característica X NO está presente en el producto? (requisitos disfuncionales)

Para cada sección, el cliente responde entre 5 únicas opciones, como se muestra en la Tabla 5, por tanto cada una de las preguntas debe contar con este grupo de posibles respuestas.

Si el rendimiento de combustible es bueno, ¿cómo te sientes?	1. Me agrada. 2. Es de esperarse 3. Neutral 4. Lo acepto 5. Me desagrada
Si el rendimiento de combustible NO es bueno, ¿cómo te sientes?	1. Me agrada. 2. Es de esperarse 3. Neutral 4. Lo acepto 5. Me desagrada

Tabla 5. *Ejemplo de cuestionario Kano*
Fuente: [León, 2005]

2.2.2.1 Cuestionario De Atribución De Importancia

Ligado al cuestionario principal de Kano, se deben realizar también una serie de preguntas de atribución de importancia, con el fin de conocer si cada una de las características del sistema es en verdad importante para los participantes. Para ello se utiliza la segunda parte de la encuesta, el cuestionario de atribución de importancia (Tabla 6), que también se utiliza para las representaciones alternas de requisitos.

Cuestionario de Atribución de Importancia En cada pregunta, marque con un círculo el número de la escala que mejor refleje su opinión									
	Para nada Importante		Algo Importante		Importante		Muy Importante		Extremo Importante
	◄								►
Requerimiento 1	1	2	3	4	5	6	7	8	9
Requerimiento 2	1	2	3	4	5	6	7	8	9
.....	1	2	3	4	5	6	7	8	9
Requerimiento X	1	2	3	4	5	6	7	8	9

Tabla 6. *Concentración de respuestas*

Fuente: [León, 2005]

2.2.3 ANÁLISIS BÁSICO DE LOS DATOS

Con base en las respuestas obtenidas en los cuestionarios de Kano, se debe buscar en la tabla 7 la correspondencia con las respuestas a los dos tipos de pregunta (Disfuncional y Funcional), para cada uno de los requisitos, con el fin de clasificar las características del producto en una de estas 6 categorías:

A: Atractivo
O: Obligatorio
U: Unidimensional
I: Indiferencia
Inv.: Respuesta inversa
D: Respuesta dudosa

Requerimientos Funcionales	Requerimientos disfuncionales				
	1	2	3	4	5
1	D	A	A	A	U
2	Inv.	I	I	I	O
3	Inv.	I	I	I	O
4	Inv.	I	I	I	O
5	Inv.	Inv.	Inv.	Inv.	D

Tabla 7. *Clasificación de requisitos*

Fuente: [León, 2005]

El procedimiento puede ser ilustrado con la figura 20.

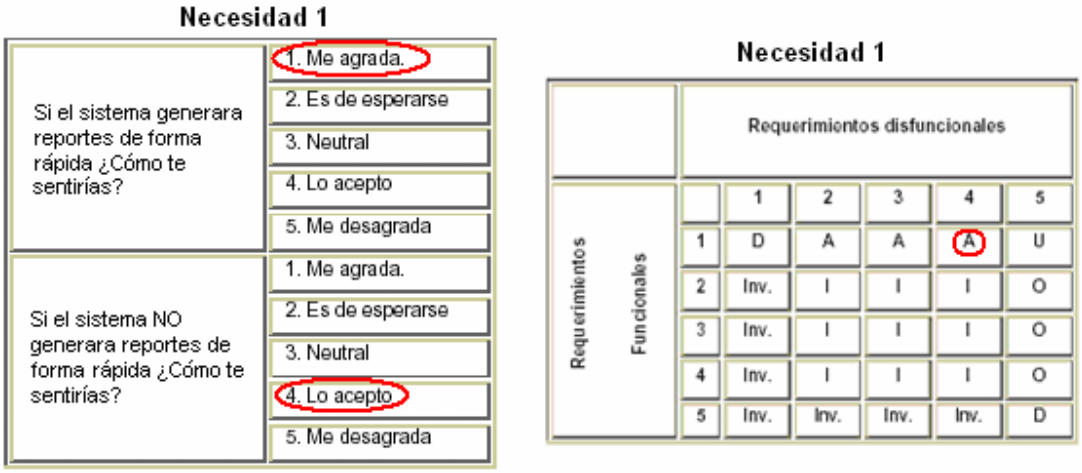


Figura 20. *Ejemplo de tabulación de datos en el modelo de Kano*

Fuente: Elaboración propia

El usuario o cliente contestó para la primera necesidad, en la sección funcional, la opción "*Me agrada*" y para la sección disfuncional "*Lo Acepto*"; tomando la correspondencia con la tabla 7 para estas dos respuestas, se puede clasificar esta necesidad, para este usuario, como ***atractiva***.

Anteriormente se realizó la descripción de las tres primeras categorías enunciadas, pero se puede observar que existe un total de 6 clasificaciones para los requisitos. Los tres faltantes se describen así:

Un cliente puede ser ***Indiferente*** a una característica de calidad, esto indica que una mayor o menor funcionalidad respecto a esta característica no se refleja en un aumento o disminución de la satisfacción del cliente.

Una ***respuesta inversa*** indica que la interpretación de criterios funcionales y disfuncionales del diseñador es la inversa a la percepción del cliente (lo que la pregunta supone como funcional es percibido como no funcional por quien responde). Por último, cuando existe una contradicción en las respuestas a las preguntas, se clasifica en el último grupo, el de ***respuesta dudosa*** (ante un par de preguntas complementarias no es razonable contestar "me desagrada" a la pregunta funcional y "me desagrada" a la disfuncional).

2.2.4 ANÁLISIS DETALLADO DE LOS DATOS

El mapa de las respuestas

El análisis detallado comienza con una observación minuciosa del mapa de las respuestas, es decir, se crea una tabla de concentración de respuestas correspondiente a cada una de las preguntas del cuestionario (se muestra un ejemplo en la tabla 8), en donde el objetivo es observar la dispersión de las respuestas. Otros autores (Matzler y Hinterhuber, 1998) prefieren elaborar un gráfico en dos dimensiones con estos datos, en donde también se pueden identificar patrones de agrupaciones de respuestas (Figura 21).

Pregunta X	Requerimientos disfuncionales					
Requerimientos Funcionales		1	2	3	4	5
	1			..	.....	
	2				..	
	3			....		
	4	.....	....	..		
	5	.....				

Tabla 8. *Concentración de respuestas*

Fuente: [León, 2005]

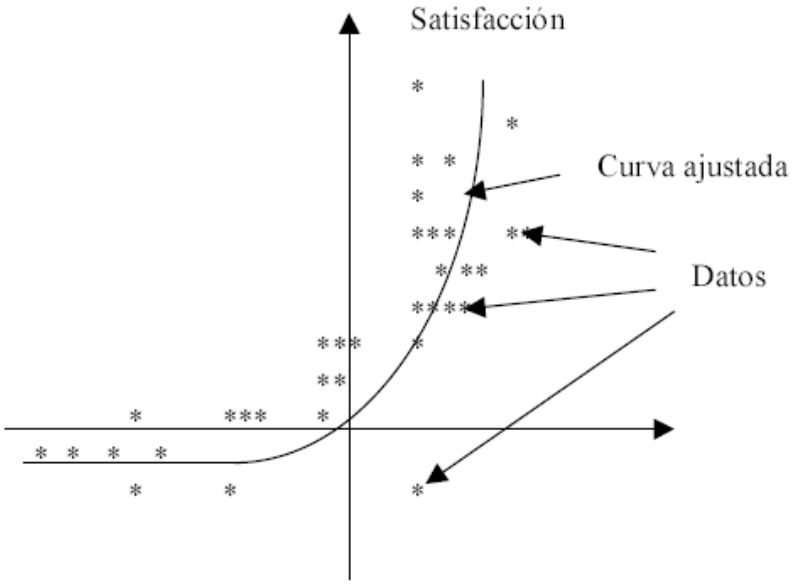


Figura 21. *Concentración de respuestas en forma gráfica*

Fuente: [León, 2005]

A continuación se confeccionan índices que sintetizan la información del análisis básico y tablas que ordenan los atributos según su tipo: primero los obligatorios, seguidos de los unidimensionales, los atractivos, los indiferentes y, finalmente, los inversos.

Una Prueba Estadística

El cuestionario de Kano se utiliza generalmente en contextos de desarrollo de productos donde tienen primacía los enfoques cualitativos. Sin embargo, existe una prueba estadística que permite evaluar la significatividad de la clasificación de Kano. Esta prueba consiste en calcular el valor de la diferencia absoluta de las dos frecuencias más votadas de las alternativas (Atractivo, Obligatorio, Unidimensional, Indiferencia, Inversa, y Dudosa) y compararlo con el estadístico:

$$Q = \sqrt{(a+b)(2n-a-b)/2n}$$

Donde:

a, b: las frecuencias de las dos observaciones más frecuentes

n: número total de respuestas.

Fórmula 1

Este valor Q se compara con el de la diferencia absoluta Abs (*a*-*b*), y si la diferencia absoluta es menor, esto indica que no hay una diferencia significativa entre las dos clasificaciones más frecuentes de cada pregunta, por lo que debe investigarse más a fondo, para descubrir la presencia de segmentos de mercado identificables o problemas en la formulación de la pregunta.

Una representación alterna

Se ha propuesto una interpretación alterna de la clasificación de los requisitos, basada en el incremento de la satisfacción (indicado en la fórmula 2 como Mejor) o bien el decremento de la misma (indicado en la fórmula 3 como Peor) debida a la inclusión o ausencia de una necesidad como característica del producto. Estas fórmulas se obtienen de la percepción de ser Mejores que la competencia al satisfacer requisitos tipo A (Atractivos) y U (Unidimensionales), o bien de la de ser Peores que la competencia al no satisfacer requisitos tipo U (Unidimensionales) y

O (Obligatorios). En el denominador de ambas fórmulas aparece una sumatoria de las percepciones de atributo A (Atractivos), O (Obligatorios), U (Unidimensionales) e I (Indiferencia). Nótese que se han suprimido de esta sumatoria las percepciones de Inv. (Respuesta inversa) y D (Respuesta dudosa), por su propio carácter confuso.

$$Mejor = \frac{A+U}{A+U+O+I}$$

Fórmula 2

$$Peor = \frac{U+O}{A+U+O+I}$$

Fórmula 3

Para simplificar los cálculos, es conveniente elaborar una tabla de concentración de resultados (tabla 9), en la cual se agrupen los resultados obtenidos para cada uno de los requisitos en torno a la clasificación obtenida. Con base en las fórmulas 2 y 3, se obtienen las columnas C1 y C2 de la tabla 9, al multiplicar los valores obtenidos de Mejor y Peor por la Importancia promedio (la octava columna de la tabla 9), la cual es a su vez obtenida a partir del promedio de la evaluación de importancia del requisito. Estos valores C1 y C2 se grafican en una escala bidimensional de Mejor y Peor, en donde se puede identificar la clasificación del tipo de requisito (figura 22).

Requerimiento	A	O	U	I	Inv	D	Imp.	Mejor	Peor	C1	C2
1	53	20	35	6	0	0	0.55	0.77	0.48	0.42	0.26
2	40	17	33	24	0	0	0.91	0.68	0.44	0.62	0.40

Tabla 9. *Ejemplo de una tabla de concentración de respuestas*

Fuente: [León, 2005]

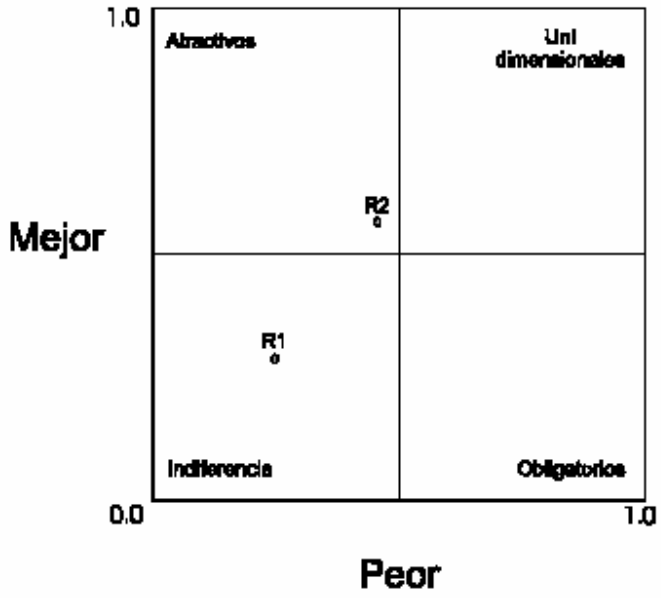


Figura 22. *Tipos de Requisitos del cliente*

Fuente: [León, 2005]

2.2.5 CICLO DINÁMICO DEL PRODUCTO

En general, los clientes no compran solamente un producto básico, sino un bien más complejo, integrado por atributos genéricos fundamentales y otros atributos complementarios. A medida que los mercados maduran, los clientes modifican sus expectativas y exigencias con respecto a los productos. Así, lo que hoy se considera un producto aumentado, mañana puede ser un producto básico o un producto esperado y las empresas deben trabajar continuamente para diferenciar sus productos con nuevos atributos; por ejemplo, para el caso de un analgésico, el producto básico puede considerarse el comprimido que contiene el principio activo y los excipientes; el producto esperado serían los comprimidos correctamente presentados en su envase, con un prospecto claro; y por último, el producto ampliado sería la suma del producto esperado más algunos servicios como una línea directa de consulta para el paciente.

Tipo de producto	Atributos requeridos en la clasificación de Kano
Producto básico	Atributos obligatorios
Producto esperado	Atributos obligatorios + Atributos unidimensionales
Producto ampliado	Atributos obligatorios + Atributos unidimensionales + Atributos atractivos incorporados
Producto potencial	Atributos obligatorios + Atributos unidimensionales + Atributos atractivos incorporados + Atributos atractivos aún no incorporados

Tabla 10. *Diferentes combinaciones de requisitos*

Fuente: [Yacuzzy y Martín, 2002]

El modelo de Kano busca fundamentalmente discriminar entre requisitos atractivos, obligatorios y unidimensionales, poniéndolos en correspondencia con la clasificación de la figura 22. Esta discriminación tiene valor a la hora de tomar decisiones estratégicas. En efecto, en el momento de asignar los recursos, es recomendable concentrarlos en el diseño y la implantación de aquellos requisitos que harán al producto o servicio más atractivo. Dado que los requisitos de los clientes son generalmente numerosos, es importante lograr una mezcla adecuada de atributos: se deben incorporar todos los requisitos obligatorios (que conforman el producto básico), una cantidad razonable de requisitos unidimensionales (equivalentes a tener un producto esperado que sea competitivo) y algunos requisitos atractivos (para ofrecer un producto ampliado que se destaque respecto a los de la competencia). Sobre esta base se desarrolla un producto ampliado y una estrategia de mercadeo que combine los requisitos del siguiente modo:

1. Deben incluirse todos los requisitos obligatorios y unidimensionales (producto esperado)

2. Deben incluirse algunos requisitos atractivos (producto ampliado).

3. Otros requisitos atractivos deben dejarse para versiones futuras del producto (producto potencial). No todos los requisitos atractivos deberían incluirse en la

primera versión del producto. Independientemente de la capacidad de desarrollo del laboratorio, siempre es posible algún tipo de competencia que imite algunas características innovadoras del producto. Por lo tanto, es conveniente dejar para versiones sucesivas algunos requisitos atractivos.

4. Los requisitos indiferentes deben omitirse o bien estudiarse con más profundidad para futuras versiones del producto.

2.3 CAPABILITY MATURITY MODEL INTEGRATION (CMMI)

2.3.1 ORÍGENES Y PRINCIPIOS DEL CAPABILITY MATURITY MODEL (CMM)

El éxito de Administración Total de la Calidad ó TQM (Total Quality Management), originó a finales de la década de los 80´s, que el proceso de madurez de software adoptara los principios de administración de la calidad del producto, que ya tenía cerca de 70 años de existencia

Walter Shewart definió los principios de control estadístico de la calidad, los cuales fueron ampliamente demostrados en trabajos posteriores hechos por: Edwards Demming, Phil Crosby, Watts Humprey y por el Software Engineering Institute (SEI).

La premisa de estos pioneros de la calidad dice que: El proceso real de mejora debe seguir una secuencia de pasos, el proceso debe ser primero visible, después repetible y por último medible.

Shewhart mostró que cualquier proceso racional envuelve cuatro pasos:

1. Pensar qué necesita ser realizado: analizar o especificar los requisitos o necesidades, conocido de otra forma como “qué” se tiene que hacer.

2. Pensar cómo hacerlo: diseñar o planear las formas en las cuales se puede alcanzar lo que tiene que ser hecho; dicho de otra forma, determinar “cómo” hacer lo que se tiene que hacer.

3. Hacerlo.

4. Analizar las lecciones aprendidas de las etapas realizadas: evaluar y verificar áreas de oportunidad; dicho de otra manera, “ver” cómo se ha trabajado.

Este modelo conforma un ciclo continuo mediante el cual se van haciendo mejoras a un proceso, con el conocimiento que se obtiene al término de cada iteración del ciclo. Este conocimiento se crea principalmente en la etapa tres del ciclo, en la cual son evaluadas las actividades que se realizan para satisfacer una necesidad previamente establecida, en la etapa inicial del proceso.

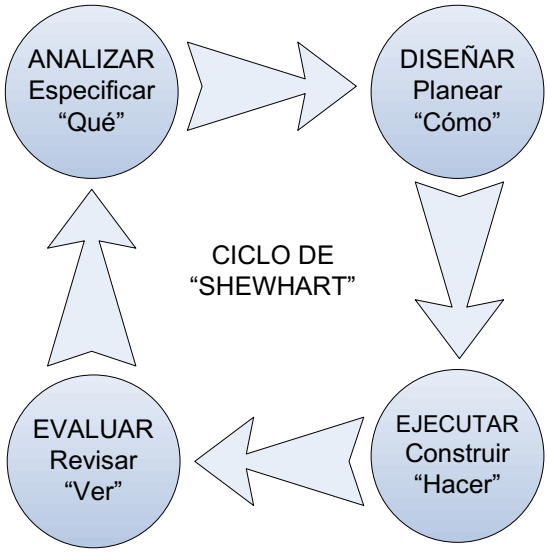


Figura 23. *Ciclo de Shewhart*
Fuente: [Barrios, 2000]

Estos trabajos realizados en el área de calidad despertaron el interés de las organizaciones software por involucrarse en los procesos de mejora de sus

procesos y productos. A partir de esto se generaron iniciativas públicas que se enfocaron en el estudio del proceso de mejora del software. Algunas instituciones que publicaron iniciativas fueron: Las Fuerzas Armadas de Alemania Federal, el Ministerio de Defensa del Reino Unido y El departamento de Defensa de Estados Unidos de América. Esta última Institución creó el Instituto de Ingeniería de Software (SEI), en el cual, bajo el liderazgo de Watts Humphrey, se desarrolló en 1987 el Modelo de Madurez del software: “Capability Maturity Model” (CMM).

El modelo CMM, sustituido posteriormente por CMMI, ofrecía un marco de referencia que podía ser usado para evaluar y mejorar los procesos usados en una organización. En dichos procesos estaba incluido el proceso de desarrollo de software.

2.3.2 ¿QUÉ ES CMMI?

CMMI (Capability Maturity Model Integration) es una evolución de los modelos de CMM, los cuales brindan un conjunto refinado de prácticas, basadas en años de experiencia, que permiten medir la madurez del proceso de ingeniería de software. Usando el modelo CMMI se alienta ampliamente la mejora y el enfoque integrado, tanto de los sistemas como de la ingeniería de software.

La utilización del modelo CMMI mejora la predicción de: los costos críticos, desarrollo, funcionamiento, programación y satisfacción del usuario. CMMI es designado de tal manera que puede ser adaptado a las metas de la organización, a diferencia de los modelos de mejora de procesos anteriores. La flexibilidad del CMMI brinda la oportunidad para enfocar el proceso de mejora hacia las áreas más importantes o más grandes de la organización.

Los modelos CMMI están organizados en dos representaciones, continua y por niveles. Estos tipos de representación ofrecen dos enfoques alternativos para la

mejora del proceso. En la figura 24 se muestra una comparación entre los dos tipos de representación.

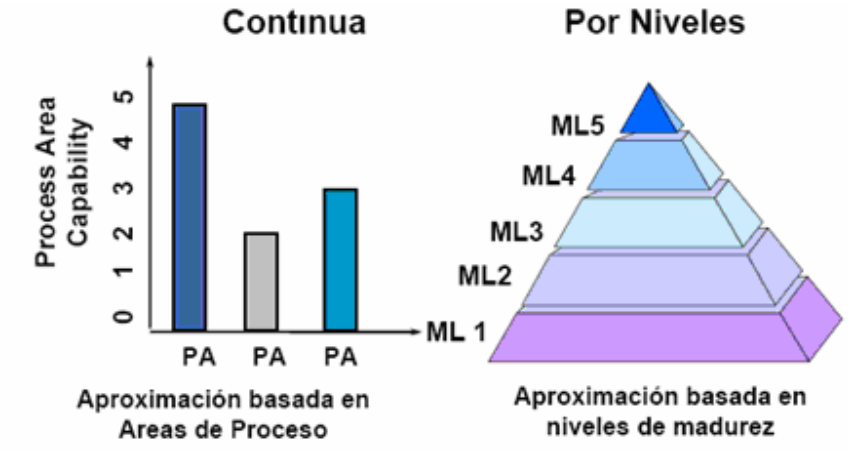


Figura 24. *Representaciones de CMMI*

Fuente: [Mayol, 2003]

Representación por niveles

La representación por niveles está basada en la madurez organizacional y las capacidades combinadas de un conjunto de procesos relacionados. La mejora organizacional se mide por niveles de madurez. Esta representación tiene un orden recomendado para enfocar el proceso de mejora, empezando con prácticas de administración básica, que van progresando a lo largo de una trayectoria probada.

Representación Continua

La representación continua está basada en las capacidades de los procesos, es decir, el rango de los resultados esperados que pueden ser alcanzados siguiendo el proceso. La mejora de procesos está medida en niveles de capacidad que se relacionan con el alcance de metas específicas y genéricas en cada área de proceso.

La representación continua provee a las organizaciones la flexibilidad de elegir a cuáles procesos será enfocada la mejora, así como también permite definir cuánto se mejorará cada proceso. Esta representación permite seleccionar el orden de realizar la mejora de los procesos que mejor cumpla las metas del negocio y que tenga menos riesgos.

Esta representación tiene las siguientes ventajas:

- Provee una flexibilidad máxima para enfocarse en áreas específicas, de acuerdo con las metas y objetivos del negocio.
- Es evolutivo, es decir, el proceso de mejora se va dando en forma gradual, sin realizar cambios radicales al proceso.
- Ofrece libertad de cómo realizar las prácticas definidas, ya que por ser un marco de referencia, únicamente indica qué es lo que se tiene que hacer y no cómo hacerlo.
- Brinda una estructura familiar para aquellos que buscan una transición de la comunidad de Ingeniería de Sistemas.

2.3.3 COMPONENTES DE CMMI

Áreas de Proceso

Agrupación de prácticas relacionadas en una determinada área que, cuando se ejecutan colectivamente, permiten cumplir con las metas consideradas importantes para realizar mejoras significativas en esta área.

Las prácticas son acciones a ser ejecutadas para alcanzar las metas de las Áreas de Proceso.

Metas y Prácticas.

El modelo CMMI define metas y prácticas para cada área de proceso de forma individual, de acuerdo con los niveles de capacidad.

Meta: Es un establecimiento de alto nivel que pretende ser alcanzado mediante la implementación efectiva de un grupo de prácticas.

Práctica: Es una descripción de una acción que es necesaria para cumplir con los elementos principales de un área de proceso.

2.3.4 ÁREAS DE PROCESO DE INGENIERÍA

A continuación se realizará una descripción de las áreas de proceso relacionadas con la ingeniería de requisitos. Toda la información contenida en esta sección fue traducida del documento CMMI, DEV. V1.2 [SEI, 2006].

Las áreas de proceso de Ingeniería cubren las actividades de desarrollo y el mantenimiento, las cuales se comparten a través de todas las disciplinas de ingeniería.

Las áreas de proceso de Ingeniería también integran los procesos asociados con las diferentes disciplinas de ingeniería en un único proceso de desarrollo de producto, soportando así una estrategia de mejoramiento del proceso orientado al producto.

Las áreas de proceso de Ingeniería aplican al desarrollo de cualquier producto o servicio en el dominio el desarrollo (por ejemplo, productos de software, productos de hardware, servicios o procesos).

Las áreas de proceso de Ingeniería son las siguientes:

- Desarrollo de requisitos
- Administración de Requisitos
- Solución Técnica
- Integración del Producto
- Verificación
- Validación

La figura 25 provee una vista de las interacciones entre las seis áreas de proceso de Ingeniería.

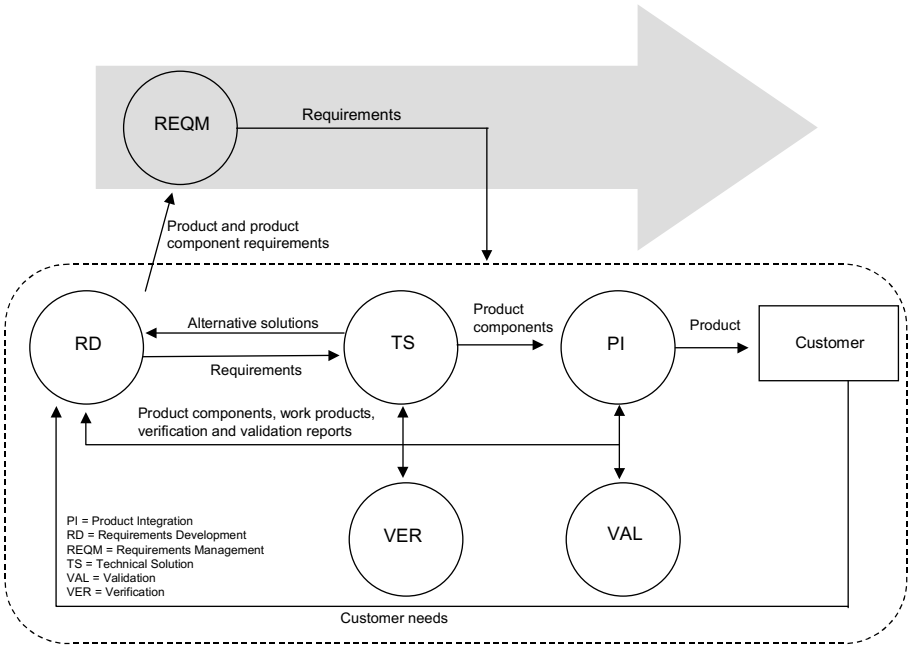


Figura 25. *Áreas de Proceso de Ingeniería*

Fuente: [SEI, 2006]

El área de proceso Desarrollo de Requisitos (RD) identifica las necesidades del cliente y traduce esas necesidades en requisitos de producto. El conjunto de requisitos de producto es analizado para producir una solución conceptual de alto nivel. Ese conjunto de requisitos es asignado posteriormente para establecer un

conjunto inicial de requisitos de componente de producto. Otros requisitos que ayudan a definir el producto son derivados y asignados a componentes de producto.

Ese conjunto de requisitos de productos y componente de producto definen claramente el desempeño del producto, las características de diseño, requisitos de verificación, entre otros, en términos que el desarrollador conoce y usa.

El área de proceso Desarrollo de Requisitos suministra los requisitos al área de proceso Solución Técnica, donde los requisitos son convertidos en la arquitectura del producto, el diseño de componente de producto y el componente de producto en sí (por ejemplo, codificación y fabricación). Los requisitos también son suministrados al área de proceso Integración del Producto, donde se combinan los componentes del producto y se verifican las interfaces, para asegurar que cumplan los requisitos de interfaz suministrados por el Desarrollo de Requisitos.

El área de proceso Administración de Requisitos mantiene los requisitos. Ésta describe actividades para obtener y controlar los cambios en los requisitos, asegurando que otros planes y datos relevantes se mantienen actualizados. Esta área de proceso provee trazabilidad de los requisitos del cliente al producto y del producto al componente de producto.

La Administración de Requisitos asegura que los cambios en los requisitos son reflejados en los planes de proyecto, actividades y productos de trabajo. Este ciclo de cambio debe afectar todos las áreas de proceso de Ingeniería; por consiguiente, la administración de requisitos es una secuencia de eventos dinámica generalmente recursiva. El área de proceso Administración de Requisitos es fundamental para un proceso de diseño de ingeniería disciplinado y controlado.

El área de proceso Solución Técnica desarrolla paquetes de datos técnicos para componentes de producto que serán usados por el área de proceso Integración

del Producto. Soluciones alternativas son examinadas bajo criterios establecidos, con el fin de seleccionar el diseño más óptimo. Esos criterios pueden ser significantemente diferentes entre productos, dependiendo del tipo de producto, el entorno operacional, los requisitos de desempeño, los requisitos de soporte, el costo y el cronograma de liberación.

El área de proceso Solución Técnica depende de las prácticas específicas en el área de proceso Verificación, para realizar la verificación del diseño y revisiones detalladas durante el diseño y antes de la construcción final.

El área de proceso Verificación asegura que los productos de trabajo seleccionados cumplen con los requisitos especificados. Esta área de proceso selecciona los métodos de verificación que serán usados para verificar los productos de trabajo contra los requisitos especificados. La verificación es por lo general un proceso incremental, empezando con la verificación de los componentes de producto y, usualmente, concluyendo con la verificación de productos totalmente ensamblados.

La verificación también dirige revisiones detalladas, las cuales son un método demostrado para remover defectos y suministrar un valioso entendimiento sobre los productos de trabajo y los componentes de producto a ser desarrollados y mantenidos.

El área de proceso Validación evalúa incrementalmente los productos contra las necesidades de los clientes. La validación deberá ser ejecutada en un ambiente operacional o en un ambiente operacional simulado. La coordinación con el cliente durante la validación es un elemente importante en esta área de proceso.

El área de proceso Validación incluye la validación de productos, componentes de producto, productos de trabajo intermedios seleccionados y procesos. Estos elementos validados, generalmente requieren reverificación y revalidación. Los

problemas descubiertos durante la validación son solucionados usualmente en el área de proceso Desarrollo de Requisitos o en Solución Técnica.

El área de proceso Integración del Producto contiene las prácticas específicas asociadas con generar la mejor secuencia de integración posible, integrar los componentes de producto y liberar el producto al cliente.

La Integración del Producto usa las prácticas específicas de las áreas de procesos Validación y Verificación en la implementación del proceso de integración del producto.

2.3.5 ÁREA DE PROCESO: DESARROLLO DE REQUISITOS (RD)

Es un área de proceso de ingeniería de nivel de madurez 3. El propósito del Desarrollo de Requisitos (RD – Requirements Development) es producir y analizar los requisitos del cliente, del producto y de componente de producto.

Esta área de proceso describe tres tipos de requisitos: requisitos de cliente, requisitos de producto y requisitos de componente de producto. Estos requisitos dirigen las necesidades de los stakeholders más relevante, incluyendo aquellos correspondientes a varias fases del ciclo de vida (por ejemplo, criterios de aceptación de pruebas) y atributos del producto (por ejemplo, seguridad, fiabilidad, mantenibilidad). Los requisitos también dirigen las restricciones causadas por la selección de soluciones de diseño.

El desarrollo de los requisitos incluye las siguientes actividades:

- Elicitación, análisis, validación y comunicación de las necesidades de los clientes, expectativas y restricciones para obtener los requisitos de los clientes, las cuales constituyen un entendimiento de lo que satisfacerá a los stakeholders

- Recolección y coordinación de las necesidades de los stakeholders
- Desarrollo del ciclo de vida de los requisitos del producto
- Establecimiento de los requisitos de los clientes
- Establecimiento de los requisitos del producto inicial y del componente del producto, consistente con los requisitos de los clientes.

PRÁCTICAS ESPECÍFICAS POR CADA META ESPECÍFICA

META 1: Desarrollar requisitos del cliente

Las necesidades, expectativas, restricciones e interfaces son recolectadas y traducidas en requisitos del cliente.

Las necesidades de los stakeholders son la base para determinar los requisitos del cliente. Las necesidades, expectativas, restricciones, interfaces y los conceptos operacionales y de producto son analizados, armonizados, refinados y elaborados para su traducción en un conjunto de requisitos de cliente.

Esto se realiza de forma cíclica con algún representante de los stakeholders, con el fin de facilitar la interacción y resolver los conflictos que se puedan presentar.

- **Práctica 1.1: Elicitar las necesidades**

  Elicitar las necesidades, expectativas, restricciones e interfaces de los stakeholders para cada fase del ciclo de vida del producto.

  Elicitar va más allá de la recolección e identificación de requisitos adicionales que no habían sido expresados explícitamente por los clientes. Esos requisitos adicionales deben dirigir las diferentes actividades en el ciclo de vida del producto y además medir su impacto sobre éste.

- **Práctica 1.2: Desarrollar los requisitos del cliente**

  Transformar las necesidades, expectativas, restricciones e interfaces de los stakeholder en requisitos de cliente.

  Los diferentes aportes de los stakeholders deben ser consolidados, la información faltante debe ser encontrada y los conflictos deben ser resueltos, cuando se está documentando el conjunto de requisitos de usuario. Estos pueden contener algunas necesidades, expectativas y restricciones sujetas a validación y verificación.

META 2: Desarrollar requisitos de producto

Los requisitos de cliente son refinados y elaborados para desarrollar los requisitos de producto y de componente de producto.

Los requisitos de producto son analizados en conjunto con el desarrollo del concepto operacional para derivar un conjunto de requisitos más precisos llamados “requisitos de producto y de componente de producto”, los cuales están relacionados con el ciclo de vida de cada producto.

- **Práctica 2.1: Establecer requisitos de producto y de componente**

  Establecer y mantener los requisitos de producto y componente de producto que están basados en los requisitos del cliente.

  Los requisitos del cliente deben ser expresados en los términos de éste y no pueden tener descripciones técnicas. Los requisitos de producto son la expresión de los requisitos del cliente en términos técnicos, los cuales sirven posteriormente para decisiones de diseño. Un ejemplo de esta traducción se puede encontrar en *la Casa de la Calidad*, la cual mapea los deseos y necesidades del cliente en parámetros técnicos.

- **Práctica 2.2: Asignar requisitos de componente de producto**

  Asignar los requisitos para cada componente de producto.

  Los requisitos de componente de producto de la solución definida incluye la asignación del desempeño del producto, restricciones de diseño y forma y función para cumplir con los requisitos y facilitar la producción.

- **Práctica 2.3: Identificar requisitos de interfaz**

  Identificar requisitos de interfaz

  En esta práctica específica se identifican las interfaces entre las funciones y se definen los requisitos de interfaz entre productos o componentes de producto, identificados en la arquitectura. Estos requisitos son controlados como parte de la integración entre el producto y el componente de producto y son una parte integral en la definición de la arquitectura.

META 3: Analizar y validar los requisitos

Los requisitos son analizados y validados, y se desarrolla una definición de la funcionalidad requerida.

Las prácticas específicas de esta meta soportan el desarrollo de los requisitos en las dos metas descritas anteriormente (desarrollar requisitos de cliente y desarrollar requisitos de producto).

Las prácticas específicas asociadas en esta meta específica cubren el análisis y validación de los requisitos con respecto al ambiente deseado por el usuario.

El objetivo de los diferentes análisis es determinar los requisitos que satisfacen las necesidades, expectativas y restricciones de los stakeholders. En paralelo a esta

actividad se deben determinar los parámetros que serán usados para evaluar la efectividad del producto.

Los requisitos son validados con el fin de incrementar la probabilidad de que el producto resultante se desempeñará de la forma deseada en el ambiente del usuario.

- **Práctica 3.1: Establecer conceptos y escenarios operacionales**

  Establecer y mantener conceptos operacionales y sus escenarios asociados.

  Un escenario es una secuencia de eventos que pueden ocurrir al usar el producto, que se utiliza para hacer explícitas las necesidades de los stakeholders. En contraste, un concepto operacional depende tanto de la solución como del escenario, estos conceptos operacionales son refinados como decisiones de solución y con base en estos se desarrollan requisitos de bajo nivel más detallados.

- **Práctica 3.2: Establecer una definición de funcionalidad requerida**

  Establecer y mantener una definición de la funcionalidad requerida.

  La definición de funcionalidad es la descripción de lo que el producto debe hacer. Puede incluir acciones, secuencia, entradas, salidas, o cualquier otra información que especifique la forma en la cual será usado el producto.

- **Práctica 3.3: Analizar los requisitos**

  Analizar los requisitos para asegurar que son necesarios y suficientes.

  Se analizan los requisitos para un nivel de jerarquía del producto a la luz de los conceptos operacionales y los escenarios, con el fin de determinar si son necesarios y suficientes para cumplir los objetivos de niveles más altos en la

jerarquía. Estos a su vez proveen la base para un análisis más detallado en niveles más bajos.

- **Práctica 3.4: Analizar los requisitos para alcanzar el balance**

  Analizar los requisitos para balancear las necesidades con las restricciones de los stakeholders.

  Las necesidades y restricciones de los stakeholders pueden dirigir los costos, el cronograma, la funcionalidad, la reusabilidad de componentes, la matenibilidad o el riesgo.

- **Práctica 3.5: Validar los requisitos**

  Validar los requisitos para asegurar que el producto resultante se desempeñe satisfactoriamente en el ambiente de usuario.

  La validación de los requisitos es realizada durante el desarrollo con los usuarios finales, para ganar confianza en que los requisitos son capaces de guiar un desarrollo que lleve a una validación exitosa. Esta actividad debería ser integrada con actividades de la gestión del riesgo.

2.3.6 ÁREA DE PROCESO: ADMINISTRACIÓN DE REQUISITOS (REQM)

Es un área de proceso de nivel de madurez 2. El propósito de la Administración de Requisitos es gestionar los requisitos del proyecto, de productos y componentes de producto e identificar inconsistencias entre dichos requisitos, los planes de proyecto y los productos de trabajo.

Los procesos de la administración de requisitos gestionan todos los requisitos recibidos o generados por el proyecto, incluyendo los requisitos técnicos y no técnicos, así como también aquellos requisitos impuestos en el proyecto por la

organización. En particular, si el área de proceso Desarrollo de Requisitos (RD) se encuentra implementada, sus procesos generarán un producto y componentes de producto, los cuales también deberán se administrados por los procesos de Administración de Requisitos. Cuando las áreas de proceso Administración de Requisitos, Desarrollo de Requisitos y Solución Técnica son implementadas, sus procesos asociados deben estar muy ligados y deben ser ejecutados concurrentemente.

El proyecto toma pasos apropiados para asegurar que el conjunto de requisitos acordado es administrado para soportar la planeación y ejecución de las necesidades del proyecto. Cuando un proyecto recibe los requisitos de un proveedor de requisitos aprobado, éstos son revisados con el proveedor para resolver problemas y prevenir interpretaciones equivocadas, antes de que los requisitos sean incorporados dentro del plan del proyecto. Una vez que el proveedor y el receptor de los requisitos llegan a un acuerdo, se genera un compromiso de todos los participantes del proyecto. El proyecto administra los cambios en los requisitos a medida que éstos evolucionan y se identifican inconsistencias entre los planes, productos de trabajo y requisitos.

Parte de la administración de requisitos es documentar los cambios en los requisitos y mantener una trazabilidad bidireccional entre las fuentes de los requisitos y todos los productos y componente de producto.

PRÁCTICAS Y ESPECÍFICAS POR CADA META ESPECÍFICA

META 1: Administrar los requisitos

Los requisitos son administrados y las inconsistencias con los planes de proyecto y los productos de trabajo son identificadas.

El proyecto mantiene un conjunto actual y aprobado de requisitos sobre todo el ciclo de vida del proyecto, haciendo lo siguiente:

- Administrando todos los cambios en los requisitos
- Manteniendo las relaciones entre los requisitos, el plan de proyecto y los productos de trabajo
- Identificando inconsistencias entre los requisitos, el plan de proyecto y los productos de trabajo
- Tomando acciones correctivas

- **Práctica 1.1: Obtener un entendimiento de los requisitos**

  Desarrollar un entendimiento del significado de los requisitos con ayuda de los proveedores de éstos.

  A medida que el proyecto madura será necesario que todas las actividades o disciplinas reciban los requisitos derivados. Para evitar requisitos inconsistentes, se establecen criterios para designar canales apropiados, o fuentes oficiales, de los cuales se reciban los requisitos. Las actividades de recepción conducen al análisis de los requisitos con el proveedor, con el fin de asegurar el entendimiento del significado de dicho requisitos. El resultado de ese análisis y diálogo es un acuerdo del conjunto de requisitos.

- **Práctica 1.2: Obtener un compromiso con los requisitos**

  Obtener compromiso de los participantes del proyecto con los requisitos.

  Mientras la práctica específica anterior trata sobre alcanzar un entendimiento de los requisitos del proveedor, esta práctica específica trata sobre el acuerdo y compromiso entre aquellos que tienen que llevar a cabo las actividades necesarias para implementar los requisitos.

Los requisitos evolucionan a través del proyecto, especialmente como fue descrito en la práctica específica del área de proceso Desarrollo de Requisitos. A medida que los requisitos evolucionan, esta práctica específica asegura que los participantes del proyecto se comprometen con los requisitos aprobados y los cambios resultantes en los planes de proyecto, actividades y productos de trabajo.

- **Práctica 1.3: Administrar los cambios en los requisitos**

  Manejar los cambios en los requisitos a medida que ellos evolucionan durante el proyecto.

  Durante el proyecto, los requisitos cambian por una gran variedad de razones. A medida que las necesidades cambian y el trabajo avanza, requisitos adicionales se derivan y es necesario realizar cambios al conjunto de requisitos existente. Es esencial administrar eficiente y efectivamente esas adiciones y cambios. Para analizar efectivamente el impacto de los cambios, es necesario que la fuente de cada requisito sea conocida y la razón fundamental para cualquier cambio esté documentada. El gerente de proyecto debe, sin embargo, seguir medidas apropiadas para la evaluación de la volatilidad de los requisitos para juzgar si es necesario establecer nuevos controles.

- **Práctica 1.4: Mantener una trazabilidad bidireccional de los requisitos**

  Mantener la trazabilidad bidireccional ente los requisitos y los productos de trabajo.

  El propósito de esta práctica específica es mantener la trazabilidad bidireccional de los requisitos por cada nivel de descomposición del producto. Cuando los requisitos son bien administrados, la trazabilidad puede ser establecida desde la fuente del requisito hasta los requisitos de más bajo nivel

y viceversa. Cada trazabilidad bidireccional ayuda a determinar que todas las fuentes de requisitos han sido completamente direccionadas y que todos los requisitos de bajo nivel pueden ser trazados a una fuente válida.

La trazabilidad también puede cubrir las relaciones a otras entidades como productos de trabajo intermedio y finales, cambios en documentación de diseño y planes de prueba. La trazabilidad puede cubrir tanto relaciones horizontales como verticales. La trazabilidad es particularmente útil en la evaluación del impacto de los cambios en los requisitos en las actividades del proyecto y los productos de trabajo.

- **Práctica 1.5: Identificar inconsistencias entre los productos de trabajo y los requisitos**

  Identificar inconsistencias entre los planes de proyecto, los productos de trabajo y los requisitos.

  Esta práctica específica encuentra inconsistencias entre los requisitos, los planes de proyecto y los productos de trabajo e inicia la acción correctiva para repararlos.

CAPITULO 3: PROPUESTA METODOLÓGICA

En este capítulo se desarrollará la metodología propuesta, la cual constituye el objetivo real de este trabajo.

Se inicio con el análisis de cuatro modelos de proceso, con el fin de identificar cuál de ellos proporcionaría una base sólida para la propuesta metodológica. Luego de realizar una comparación de los beneficios y desventajas que ofrecía cada uno de ellos, se concluyó que, el modelo que más se adaptaba a la metodología, era el propuesto por Amador Durán en su tesis *"Un Entorno Metodológico de Ingeniería de Requisitos para Sistemas de Información",* que consta de 3 actividades básicas, cuyas características permitirían desplegar paso a paso la propuesta desarrollada. A este modelo se le realizaron algunas adecuaciones consideradas necesarias para el propósito de la metodología.

Luego, en la sección 3.2, se propone realizar un *modelado del negocio* con el fin de obtener un conocimiento muy claro de cuál es el problema que se desea resolver y todas aquellas características de la organización que se hace necesario conocer para llevar a cabo el proyecto.

Finalmente, en las secciones 3.3, 3.4, 3,5 y 3.6, se definen cuatro grandes actividades: *Elicitación, Análisis, Validación y Gestión de Requisitos*, las cuales constituyen el punto central del proyecto de grado. Cada una de estas etapas se divide en tareas específicas dentro de las cuales surgen algunos artefactos o productos de trabajo necesarios para el desarrollo de otras tareas.

Dentro de la etapa de Análisis se encuentra una de las tareas principales, ésta se refiere a la construcción de la Casa de la Calidad a través de una serie de pasos donde se consolidan muchos de los resultados obtenidos en las tareas desarrolladas previamente. Es en este punto donde se realiza uno de los aportes más valiosos dentro de toda la propuesta metodológica, pues se plantea de una

forma innovadora la manera de correlacionar los Cómo(s) en el techo de la Casa de la Calidad.

Por último, el lector podrá percibir que a lo largo de cada etapa se estará asegurando el cumplimiento de varias prácticas específicas y, junto con ellas, dos metas de las áreas de proceso Desarrollo de Requisitos (RD) y Administración de Requisitos (REQM) del modelo CMMI. De esta forma, se asegura que cualquier organización que aplique la metodología estará llevando a cabo tareas que tienen implícitas las ventajas que proporciona dicho modelo, garantizando de alguna forma que el producto o servicio que se va a entregar al cliente cumplirá con sus expectativas.

Para resaltar que se está dando cumplimiento a una práctica o una meta específica, se propuso el manejo de ciertas convenciones, tal como se muestra en la figura 26.

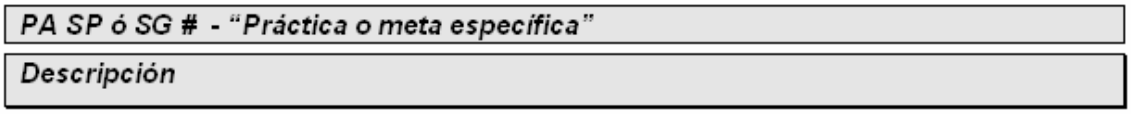


Figura 26. *Estructura propuesta para identificar el cumplimiento de una meta o práctica*
Fuente: Elaboración propia

- **PA:** Indica el área de proceso a la cual pertenece la meta o práctica a la cual se le está dando cumplimiento.
- **SP o SG:** Indica si a lo que se le está dando cumplimiento es a una práctica específica o a una meta específica del área de proceso enunciada en el campo anterior.
- **#:** Identificador de la práctica o meta específica definida por el modelo CMMI.
- **Práctica o meta específica:** En esta parte se describe la práctica o la meta a la cual se está haciendo referencia

- **Descripción:** En este campo se determina finalmente, el por qué se está dando cumplimiento a la meta o práctica a través de los diferentes puntos de la propuesta.

3.1 PROCESO DE INGENIERÍA DE REQUISITOS

La ingeniería de requisitos, hasta el momento, sigue siendo una disciplina inmadura, pues no cuenta con estándares mundialmente aceptados que permitan recopilar efectivamente las necesidades de los clientes y usuarios. “De hecho, no existe una definición universalmente aceptada.” [Durán, 2000]

A continuación se exponen algunas definiciones presentadas por diferentes autores:

Para P. Hsia, A. Davis, y D. Kung, la ingeniería de requisitos se refiere a “Todas las actividades relacionadas con: **(a)** identificación y documentación de las necesidades de clientes y usuarios. **(b)** creación de un documento que describe la conducta externa y las restricciones asociadas [de un sistema] que satisfará dichas necesidades. **(c)** análisis y validación del documento de requisitos para asegurar consistencia, compleción y viabilidad. **(d)** Evolución de las necesidades.” [Durán, 2000].

La IEEE define la ingeniería de requisitos como “**(a)** el proceso de estudiar las necesidades del usuario para llegar a una definición de requisitos de sistema, hardware o software. **(b)** El proceso de estudiar y refinar los requisitos de sistema, hardware o software.” [Durán, 2000].

Según el Departamento de Defensa de los Estados Unidos, la ingeniería de requisitos involucra todas las actividades del ciclo de vida del software dedicadas a la identificación, el análisis, la documentación, la validación de los requisitos y todos los procesos que soportan estas actividades.

Una definición muy simple, pero completa de la ingeniería de requisitos, es expuesta por Amador Duran en su tesis doctoral, en la cual se considera la ingeniería de requisitos como “un proceso de descubrimiento y comunicación de las necesidades de clientes y usuarios y la gestión de los cambios en dichas necesidades.” [Durán, 2000]

En ese mismo documento es presentada la figura 27, en la cual se muestra la situación de la ingeniería de requisitos dentro del ciclo de vida del desarrollo de software.

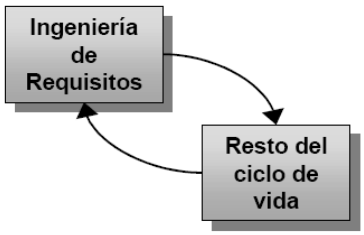


Figura 27. *La ingeniería de requisitos en el ciclo de vida del desarrollo de software*
Fuente: [Durán, 2000]

La visión de Pohl sobre la ingeniería de requisitos es “un proceso de construcción de una especificación de requisitos en el que se avanza desde especificaciones iniciales, que no poseen las propiedades oportunas, hasta especificaciones finales completas, formales y acordadas entre todos los participantes” [Durán, 2000].

En la figura 28 se muestran las dimensiones del proceso de ingeniería de requisitos, según la visión de Pohl. Como puede verse, los factores que influyen en el proceso son: Compleción, Formalidad y Acuerdo.

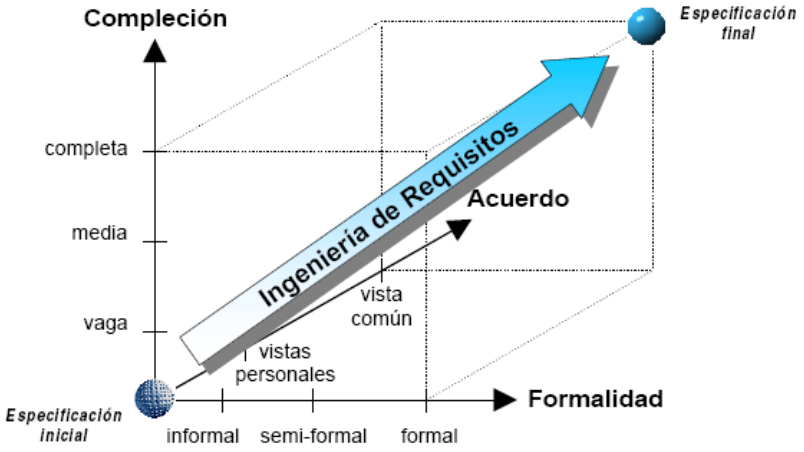


Figura 28. *Dimensiones de la ingeniería de requisitos*

Fuente: [Durán, 2000]

La *compleción* se refiere al grado de completitud del conocimiento sobre el sistema que se desea desarrollar, es decir, el llegar a conocer la totalidad de los requisitos que debe satisfacer el sistema.

El *formalismo* está ligado a la representación del conocimiento sobre los requisitos.

Dado que la ingeniería de requisitos es un proceso en el que participan personas con diferentes puntos de vista, es necesario llegar a un punto de *acuerdo*, normalmente mediante algún tipo de negociación.

3.1.1 MODELO DE PROCESO DE INGENIERÍA DE REQUISITOS

Después de evaluar los siguientes modelos de proceso de ingeniería de requisitos:

- Modelo propuesto por Amador Durán
- Modelo de Pohl
- Modelo SWEBOK

- Modelo Espiral,

se concluyó que el modelo de Amador Durán, mostrado en la figura 29, es el que mejor se adapta a la propuesta metodológica, pues presenta algunas ventajas bastante significativas con respecto a los demás modelos evaluados.

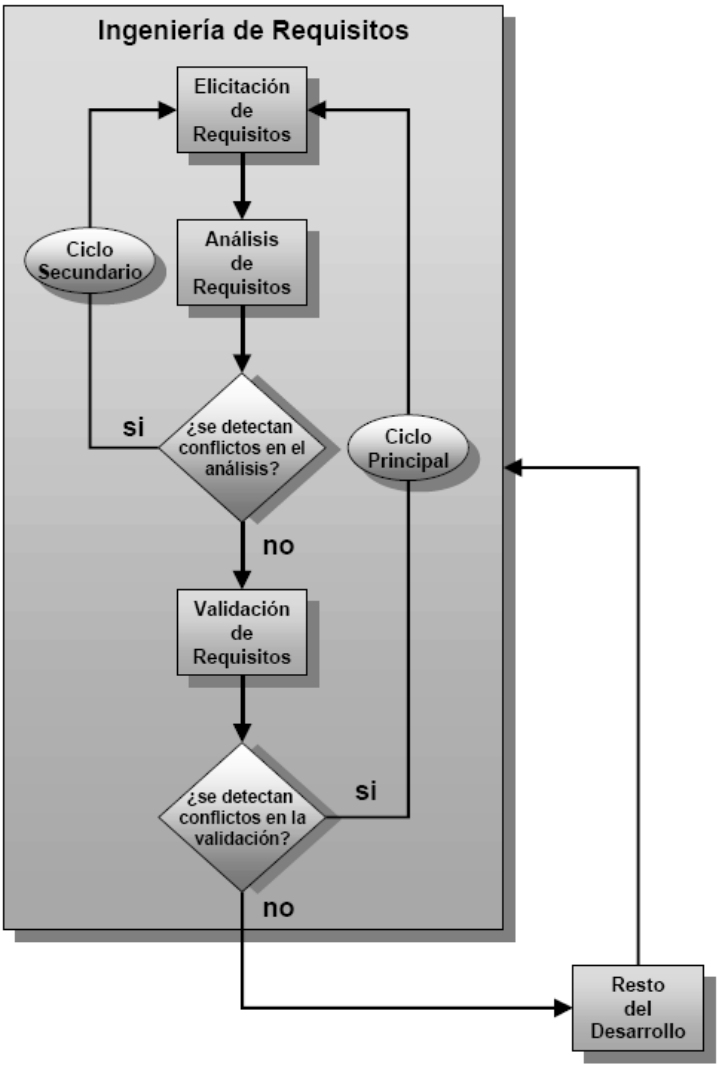


Figura 29. *Modelo de proceso de ingeniería de requisitos propuesto por Amador Durán*
Fuente: [Durán, 2000]

En esta sección se describe detalladamente cada una de las actividades que constituyen el modelo de proceso elegido, así como sus ventajas frente los modelos mencionados anteriormente.

3.1.1.1 Descripción del modelo de proceso

El modelo de proceso de ingeniería de requisitos propuesto por Amador Durán consta de tres actividades principales, *elicitación*, *análisis* y *validación*, y su principal característica es la iteratividad, puesto que consta de tres ciclos de iteración.

Cabe aclarar que “el proceso de elicitar requisitos, analizarlos y validarlos es iterativo por naturaleza, dado que es prácticamente imposible obtener todos los requisitos sin tener que volver atrás en algún momento del proceso.” [Durán, 2000]

A este modelo de proceso se le hicieron cuatro modificaciones, con el fin de adaptarlo mejor a la propuesta metodológica.

La primera adaptación fue agregar la actividad de “Modelado del negocio” antes del proceso de ingeniería de requisitos, pues se considera que enfrentarse a un proceso de elicitación de requisitos y posterior desarrollo de un sistema, sin conocer las principales características y la terminología propia del negocio, puede provocar que el producto final no sea el esperado por los clientes y usuarios.

Aunque en la propuesta de Amador se hace referencia a esta actividad en la primera tarea de la elicitación de requisitos, se considera que el modelado del negocio no hace parte de esta actividad y por esta razón se decidió realizarla en una actividad independiente, tanto de la elicitación, como del proceso de ingeniería de requisitos en sí, tal y como lo propone la metodología RUP (Rational Unified Process) (ver figura 30).

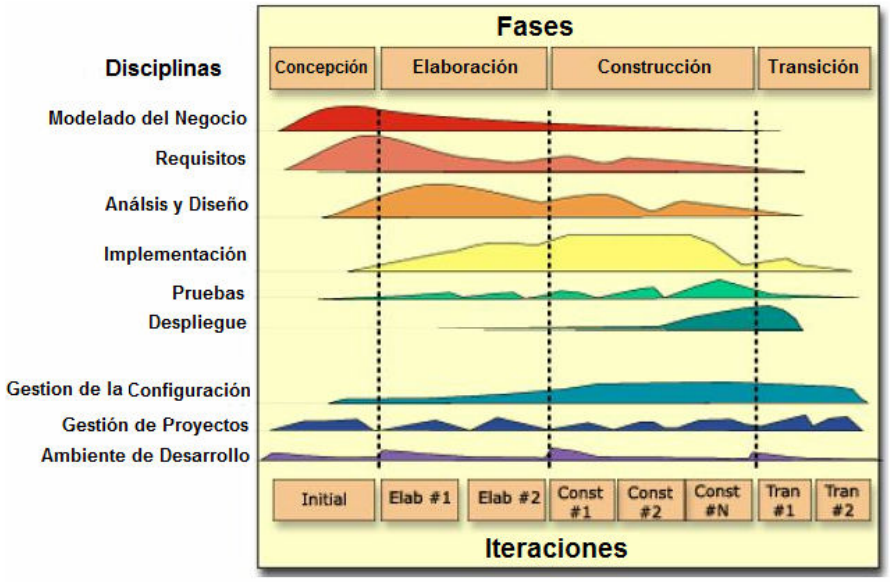


Figura 30. *Arquitectura general de RUP*

Fuente: http://dn.codegear.com/article/33319

La segunda adaptación tiene que ver con el ciclo de iteración elicitación – análisis (ciclo secundario en la figura 29). Para este caso, la ejecución de este ciclo (en su primera iteración) no estará condicionada a la existencia de conflictos en los requisitos, pues, aunque el ingeniero de requisitos no haya detectado inconsistencias, será necesario volver a la etapa de elicitación, con el fin de priorizar las necesidades por medio de cuestionarios de Kano y determinar con los clientes y usuarios la fuerza de las relaciones en las matrices de relación.

El tercer cambio realizado al modelo de proceso propuesto por Amador fue la inclusión de una segunda condición para entrar en el ciclo principal (elicitación – análisis – validación). Esta condición revela la posibilidad de que durante la actividad de validación se encuentren requisitos que estaban ocultos hasta el momento y que por esta razón sea necesario volver a realizar sesiones de elicitación. Cabe aclarar que Amador hace referencia a la aparición de nuevos requisitos en la definición de este ciclo, sin embargo, no manifiesta este hecho en su diagrama de proceso. Se considera importante hacer énfasis en esta condición pues en la propuesta metodológica se presentan los prototipos como una

herramienta muy eficaz en las actividades de validación de requisitos, sin embargo es reconocido por un gran número de autores, entre ellos Amador, como un medio igualmente efectivo para elicitar nuevos requisitos; incluso, en varios trabajos los prototipos se presentan como herramientas de elicitación.

La introducción de una nueva actividad (Gestión de Requisitos) es la cuarta y última adaptación realizada desde el modelo propuesto en [Durán, 2000]. La Gestión de Requisitos es una actividad transversal a todo el proceso de Ingeniería de Requisitos, que abarca todo el ciclo de vida del producto y consiste en gestionar los cambios realizados a los requisitos, para asegurar su consistencia. Esta actividad será explicada con mayor detenimiento en la sección 3.6.

En la figura 31 se muestra el modelo de proceso de ingeniería de requisitos después de las adaptaciones realizadas al modelo propuesto por Amador Durán en [Durán, 2000].

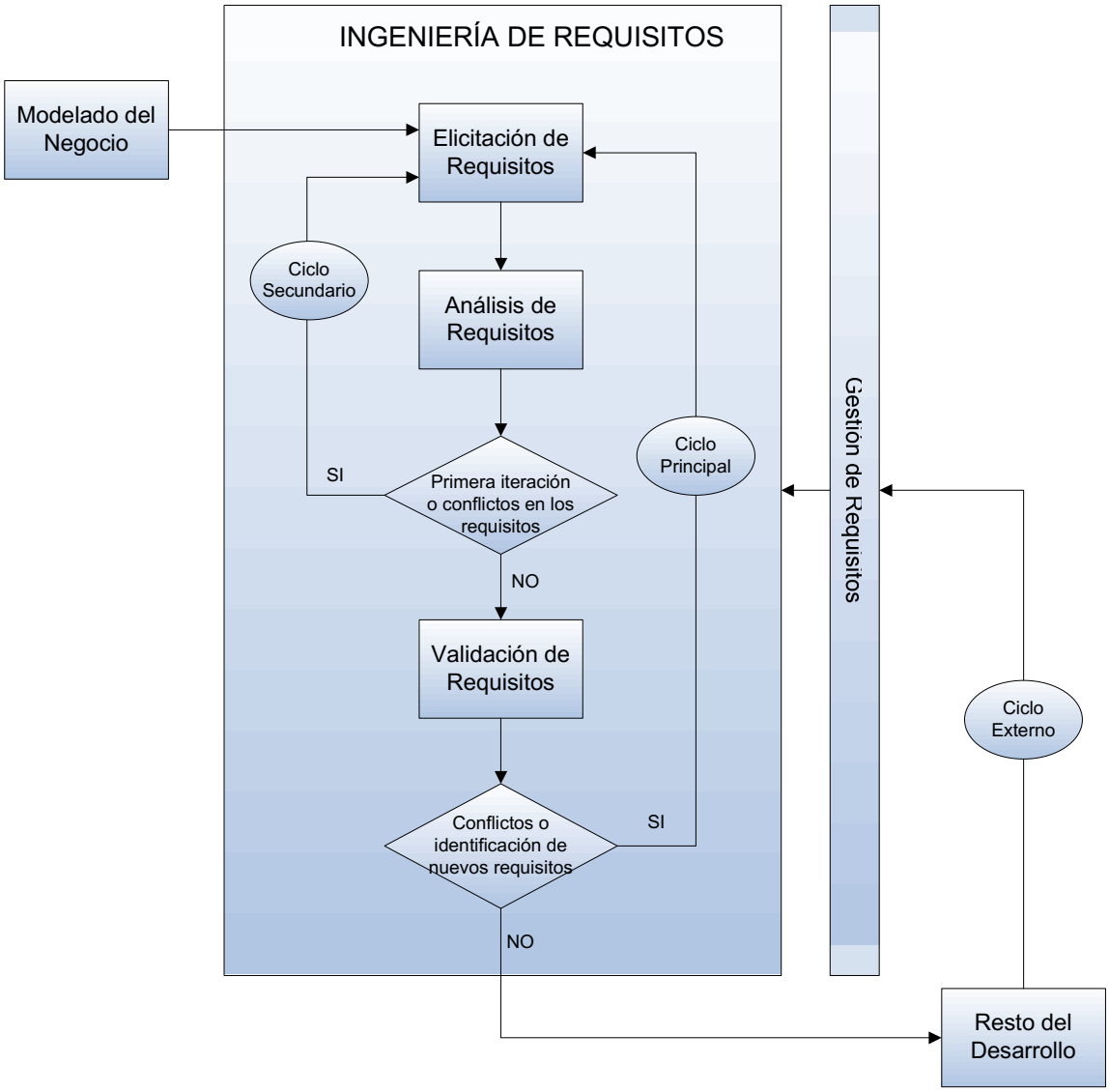


Figura 31. *Modelo de proceso de ingeniería de requisitos, adaptado de [Durán, 2000]*

Fuente: Elaboración propia

A continuación se describirá cada uno de los ciclos que componen el modelo y se definirán a grosso modo los objetivos de cada una de las actividades principales de la ingeniería de requisitos: *elicitación, análisis y validación*. Estas actividades serán descritas con mayor detenimiento en las secciones 3.3, 3.4 y 3.5, respectivamente.

3.1.1.2 Ciclos de iteración

En el modelo se definen tres posibles ciclos de iteración, dos internos al proceso y uno externo, como puede verse en la figura 31:

- *Ciclo elicitación – análisis – validación:* Este es el ciclo principal e indica la posibilidad de que durante el proceso de validación de los requisitos por parte de los clientes y usuarios, aparezcan conflictos o nuevos requisitos que hasta entonces estaban ocultos. En esas circunstancias, es necesario resolver dichos conflictos y consensuar los nuevos requisitos mediante nuevas reuniones de elicitación/negociación, repitiendo a continuación las actividades de análisis y validación.

- *Ciclo elicitación – análisis:* Es un ciclo secundario e indica la posibilidad de que durante la realización del análisis de los requisitos elicitados se descubran conflictos o deficiencias en dichos requisitos, lo que puede provocar la necesidad de nuevas reuniones de elicitación/negociación y el posterior análisis de sus resultados. Como se dijo anteriormente, en la primera iteración de este ciclo será necesario volver a la etapa de elicitación, aunque no existan conflictos en los requisitos elicitados, con el fin de priorizar las necesidades expresadas por los clientes y usuarios y determinar la fuerza de las relaciones entre necesidades, objetivos y actores, por medio de cuestionarios de Kano y matrices de relación, respectivamente, elaborados durante la actividad de análisis de requisitos.

- *Ciclo ingeniería de requisitos – resto del desarrollo:* Este ciclo externo, muestra la posibilidad de que durante el resto del desarrollo sea necesario volver a alguna de las actividades de ingeniería de requisitos, posiblemente porque se detecte la necesidad de renegociar algunos requisitos de difícil implementación, porque aparezcan nuevos requisitos durante el desarrollo, etc.

3.1.1.3 Actividades del proceso

Como se dijo anteriormente, las tres actividades que componen el modelo de procesos de ingeniería de requisitos propuesto por Amador Durán son: elicitación, análisis y validación.

- ***Elicitación de requisitos***

  Para Amador, esta actividad es la más importante de la ingeniería de requisitos, pues en ella se mantiene la interacción más importante entre clientes y usuarios e ingenieros de requisitos. Los principales objetivos que se deben intentar alcanzar en esta actividad son:

  - *Conocer el dominio del problema:* Para que los ingenieros de requisitos puedan entenderse con los clientes y usuarios y sean capaces de transmitir dicho conocimiento al resto del equipo de desarrollo, es fundamental que conozcan el dominio del problema.

  - *Descubrir las necesidades reales de clientes y usuarios:* Además de aquellas necesidades explícitamente manifestadas por los clientes y usuarios, es muy importante descubrir aquellas necesidades que la mayor parte de las veces se asumen y toman por implícitas.

  - *Consensuar los requisitos entre los propios clientes y usuarios:* Cuando existen distintos grupos de clientes y usuarios y cada uno presente sus necesidades, usualmente se generan contradicciones entre ellas, por lo que se hace necesario negociar entre los participantes hasta obtener una visión común de los requisitos.

- ***Análisis de Requisitos***

  En esta actividad se integran y analizan los requisitos elicitados, esto suele provocar la identificación de requisitos que faltan, inconsistencias y conflictos

entre los requisitos. A continuación se describen los objetivos principales de esta actividad.

- *Detectar conflictos en los requisitos-C*[6]: Los requisitos–C suelen ser información proveniente de distintas fuentes y normalmente presentan contradicciones o ambigüedades debido a su naturaleza informal. Durante el análisis de los requisitos–C, normalmente mediante la construcción de un modelo, es habitual que al incrementar la precisión con la que es necesario expresar los requisitos aparezcan dichas contradicciones o ambigüedades, que deberán resolverse en nuevas reuniones de elicitación mediante algún mecanismo de negociación.

- *Profundizar en el conocimiento del dominio del problema:* Por mucho que el ingeniero de requisitos aprenda sobre el dominio del problema, siempre quedarán aspectos desconocidos. Construir un modelo suele conllevar un incremento en el grado de conocimiento del problema que puede facilitar el proceso de construir un producto útil para clientes y usuarios.

- *Establecer las bases para el diseño:* Normalmente, los modelos obtenidos durante la realización del análisis de los requisitos–C suelen constituir las bases para las actividades de diseño.

- ***Validación de Requisitos***

  El objetivo primordial de esta actividad es confirmar que los requisitos–C corresponden realmente a las necesidades de clientes y usuarios, de forma que se evite la situación en la que el producto final, que puede ser técnicamente correcto, no es satisfactorio.

[6] En la tesis doctoral de Amador durán [Durán, 2000], los requisitos-C se refieren a los requisitos desde el punto de vista de los clientes y usuarios. Este tipo de requisitos serán explicados posteriormente en la sección 3.1.1.3 “Productos del Proceso”

La validación de los requisitos debe ser realizada principalmente por clientes y usuarios, aunque los ingenieros de requisitos pueden ayudar en el proceso mediante las explicaciones oportunas o mediante el uso de prototipos.

Las actividades de validación, sobre todo con la utilización de prototipos, conllevan generalmente a la elicitación de nuevos requisitos pues, a medida que el nuevo sistema se va perfilando, suelen ir apareciendo nuevas necesidades que hasta entonces estaban ocultas.

La cuarta actividad contenida en el modelo de proceso es la **gestión de requisitos**, la cual se encarga de administrar la obtención incremental de los requisitos y los inevitables cambios a los que están sujetos.

El objetivo principal de esta actividad es asegurar la consistencia ente los requisitos y el sistema construido (o en construcción). Por esta razón, la gestión de requisitos es esencial no sólo durante las etapas de planificación y análisis sino durante todo el proceso de desarrollo.

Dado que la gestión de cambios es una de las actividades más importantes dentro de la gestión de requisitos es clave definir y utilizar técnicas, procedimientos y herramientas para controlar los cambios que se hagan a la especificación de requisitos, siendo el concepto de trazabilidad de gran importancia para lograr estos propósitos.

3.1.1.4 Productos del Proceso

Los tres productos principales que se incluyen en el modelo de procesos elegido, son los siguientes:

- ***Requisitos–C:*** Corresponden a requisitos desde el punto de vista del cliente y son el resultado principal de la actividad de elicitación. Estos requisitos deben expresarse de forma que todos los participantes en el proceso de ingeniería de requisitos sean capaces de entenderlos, especialmente los clientes y usuarios.

Para conseguir ese objetivo, en [Duran, 2000], se propone utilizar principalmente casos de uso expresados en lenguaje natural.

En el presente trabajo de investigación, el uso de los términos requisitos-C y necesidades tienen el mismo significado, pues se refieren a la "Voz del Cliente", sin ser interpretada por los desarrolladores.

- ***Requisitos–D:*** Son los requisitos desde el punto de vista del desarrollador, que junto con el prototipo y los posibles conflictos, constituyen el resultado principal de la actividad de análisis. La forma de expresar estos requisitos suele consistir en la elaboración de un modelo del sistema a desarrollar basado en los requisitos–C.

- ***Prototipo:*** La construcción de un prototipo del sistema a desarrollar puede facilitar enormemente tanto la validación de los requisitos por parte de los clientes y usuarios como la elicitación de nuevos requisitos. Los prototipos suelen ser de dos tipos:

  - De usar y tirar: Se utilizan principalmente para elicitar y validar requisitos relacionados con la interfaz de usuario

  - Evolutivos: Se centran más en los requisitos funcionales.

- ***Conflictos:*** Es importante registrar los conflictos que vayan surgiendo para poder acometer su resolución de forma organizada, involucrando a los participantes en el proceso de negociación.

3.1.1.5 Ventajas del modelo de proceso elegido

- En ninguno de los otros modelos se contempla el posible ciclo con el resto del desarrollo. La identificación de este ciclo representa una ventaja muy significativa del modelo de proceso elegido frente a los demás modelos, pues durante las etapas posteriores del desarrollo, la probabilidad de que cambien

algunos requisitos o aparezcan nuevos es muy alta y, por tanto, debe existir una actividad dentro del proceso de ingeniería de requisitos que contemple esta posibilidad.

- En el modelo propuesto por Amador, a diferencia de los demás modelos, se describen claramente los productos del proceso que se supone, se deben obtener con la realización de cada una de las actividades de la ingeniería de requisitos.

- Aunque en todos los modelos de proceso evaluados se mencionan los prototipos, ya sea para realizar las actividades de elicitación de requisitos, o bien para hacer la validación de los mismos, Amador hace mucho más énfasis en el uso de éstos durante el proceso de ingeniería de requisitos. Se considera que esta propuesta es muy acertada, pues la construcción de un prototipo del sistema, no sólo facilita la labor del ingeniero de requisitos para realizar las actividades de validación y elicitación, sino que también permite al cliente tener una idea más clara del sistema a desarrollar e identificar nuevos requisitos o conflictos que hayan permanecido ocultos hasta el momento.

- Los requisitos de información no suelen incluirse como un grupo separado en las metodologías actuales, sin embargo, en la propuesta de Amador Durán, este tipo de requisitos son identificados claramente.

3.1.2 FLUJO DE ACTIVIDADES

En la figura 32 se muestra el flujo de actividades de todo el proceso de ingeniería de requisitos.

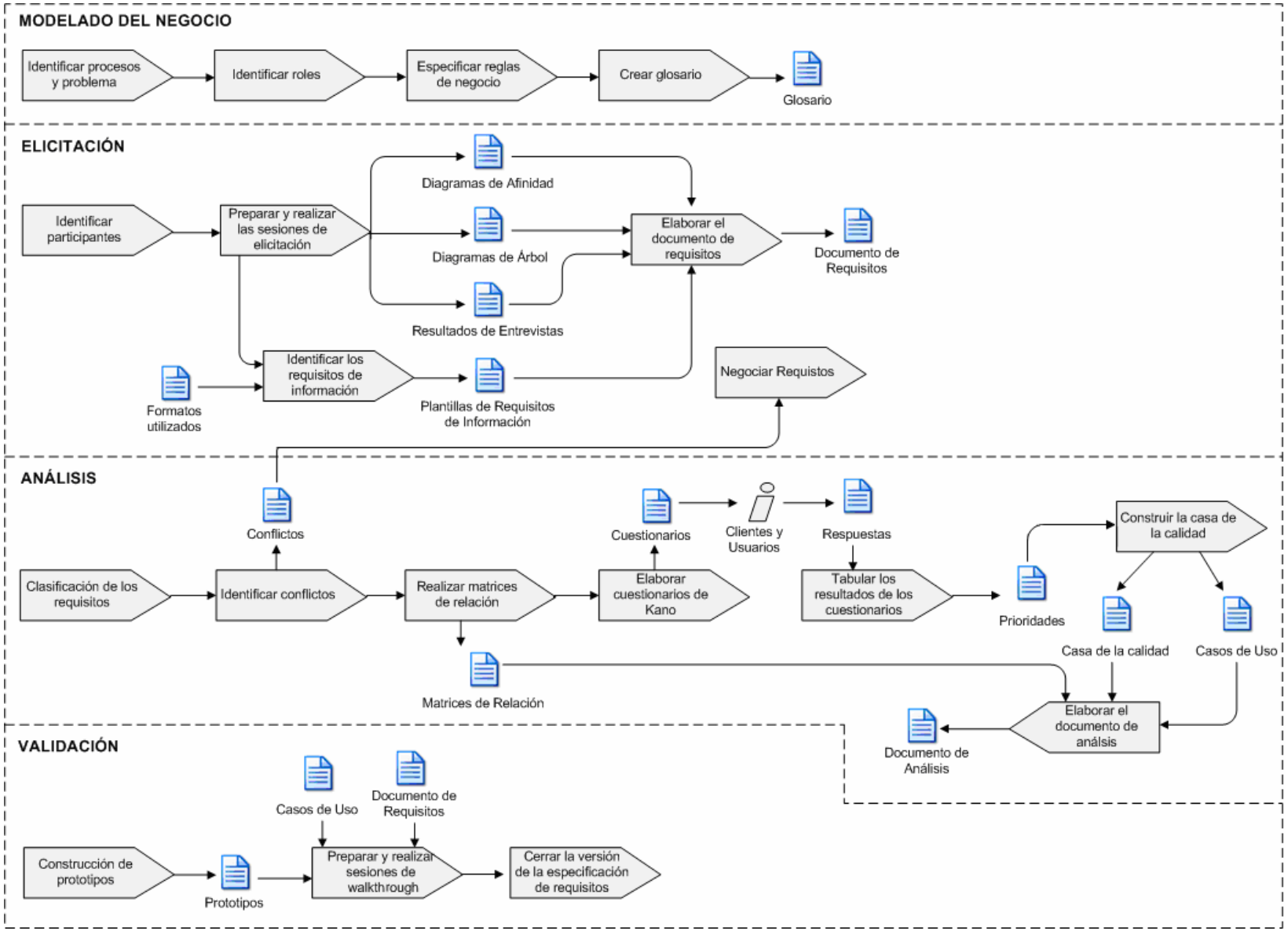


Figura 32. *Flujo de actividades y productos de trabajo propuestos*

Fuente: Elaboración Propia

3.2 MODELADO DEL NEGOCIO

Enfrentarse a un proceso de elicitación de requisitos y posterior desarrollo de un sistema sin conocer las principales características y la terminología propia del negocio, puede provocar que el producto final no sea el esperado por clientes y usuarios; es por esto que antes de realizar reuniones con ellos para identificar y comprender sus necesidades reales, es fundamental conocer el dominio del problema y evaluar el contexto organizacional y operacional del negocio.

El modelo de negocio describirá cómo obtener la visión de los objetivos de la organización y basado en esta visión, se definirán los procesos, los roles, y las responsabilidades de la organización.

Generalmente una empresa se encuentra organizada por medio de un conjunto de *procesos de negocio*; cada uno de estos procesos se caracteriza porque tiene asociados una colección de *datos* que son producidos y manipulados mediante un conjunto de *tareas*, en las cuales ciertos *agentes* (por ejemplo, trabajadores o departamentos) participan, de acuerdo con un *flujo de trabajo* determinado. Además, estos procesos se hallan sujetos a un conjunto de *reglas de negocio*, que determinan la estructura de la información y las políticas de la empresa. La finalidad del modelado del negocio es describir cada proceso de éste, especificando sus datos, actividades (o tareas), roles (o agentes) y reglas de negocio. [Ortín et al., 2001]

Según [López, 2005] el principal objetivo del modelo de negocios es:

- Entender la estructura y la dinámica de la organización en la cual el sistema trabajará.
- Entender los problemas en la organización e identificar las mejoras potenciales.

- Asegurar que el cliente, el usuario final y el desarrollador tienen claro el objetivo de la organización.
- Manejar los requisitos necesarios para soportar los objetivos de la organización.

Para facilitar la comprensión del negocio y para ubicarlo en el contexto del problema a resolver, se visualiza el modelo de negocio y se proporciona un lenguaje gráfico que ilustra la interacción entre los participantes de este modelo, los cuales son: los usuarios del modelo, el rol que desempeñan, los procesos y los objetos o artículos producidos por la organización.

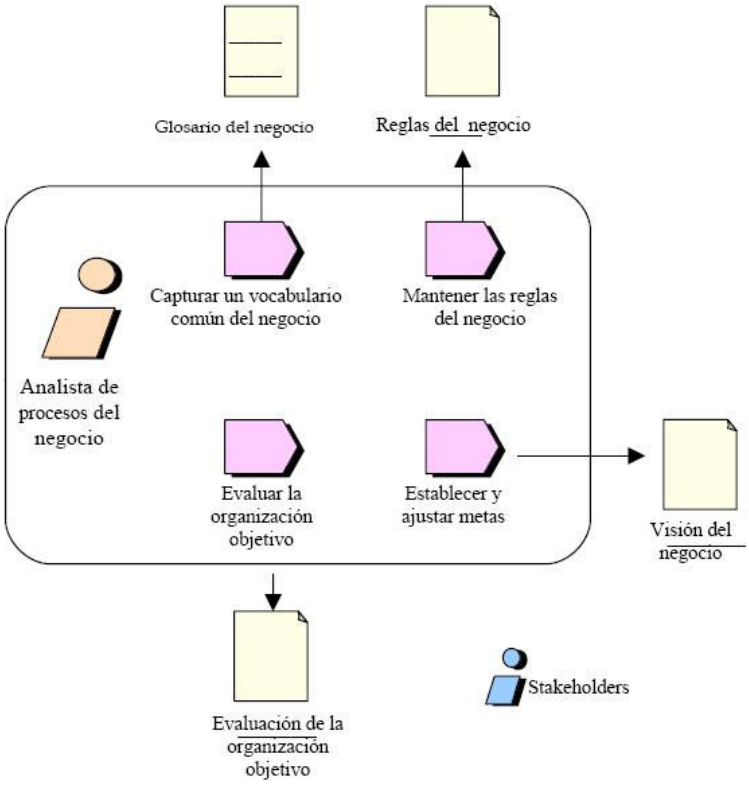


Figura 33. *Modelado del Negocio*
Fuente: [Contreras y Laciar, 2004]

El modelado del negocio está divido en las siguientes tareas:

1. Identificar los procesos del negocio y el problema
2. Identificación de los roles del entorno
3. Especificar las reglas de negocio
4. Creación de un glosario de términos

A continuación se describe detalladamente cada una de estas tareas.

3.2.1 Tarea 1: Identificar los procesos del negocio y el problema

El primer paso del modelado del negocio consiste en capturar los procesos de negocio de la organización bajo estudio, puesto que a través de estos procesos es posible definir los límites del modelado posterior.

En primer lugar se deben identificar los objetivos estratégicos de la empresa. Se debe tener en cuenta que estos objetivos son complejos y poseen un nivel de abstracción muy alto; cada uno de ellos puede ser descompuesto en un conjunto de sub-objetivos más concretos, que deberán cumplirse para conseguir el objetivo estratégico original. A su vez, estos sub-objetivos pueden ser descompuestos en otros, de manera que se defina una jerarquía de objetivos. Según el estudio realizado por [Ortín et al., 2001], dos o tres niveles de descomposición son suficientes.

Para los objetivos del último nivel se define un proceso de negocio cuyo propósito será dar soporte a dicho objetivo.

Con el fin de comprender las necesidades y difundir confianza entre los participantes del proyecto, el ingeniero de requisitos debe conocer muy bien el dominio del problema para evitar que se presenten malentendidos. Por esto es necesario basarse en modelos del sistema actual, documentación de la

organización, resultados de cuestionarios exploratorios, información proveniente de expertos, entre otros, para tener una clara identificación de cuál es el problema, a quiénes afecta, cuál es su impacto y qué aspectos podría generar una solución a este problema.

3.2.2 Tarea 2: Identificación de los roles del entorno

Esta tarea consiste en encontrar los agentes involucrados en la realización de los procesos de negocio; cada uno de ellos desempeña cierto papel o juega un rol cuando colabora con otros para llevar a cabo las actividades del negocio. En este punto es muy importante identificar con cuáles individuos se debe trabajar para recolectar información, considerando su conocimiento del negocio, habilidades de comunicación y disponibilidad.

3.2.3 Tarea 3: Especificar las reglas de negocio

En una organización, tanto los procesos como los datos que estos manejan están restringidos por las reglas del negocio. Estas reglas aseguran que la actividad de la empresa se lleva a cabo de acuerdo con restricciones impuestas desde el entorno (leyes o normas) o desde dentro de la propia organización. [Ortín et al., 2001]

Las reglas de negocio pueden dividirse en:

- **Reglas de Restricción:** Especifican políticas o condiciones que restringen la estructura y comportamiento de los objetos y procesos. Estas reglas pueden ser divididas en:

- *Reglas de estímulo-respuesta:* Restringen el comportamiento y especifican las condiciones que deben cumplirse para activar una operación.
- *Reglas de restricción de operación:* Especifican condiciones que deben cumplirse antes y después de ejecutarse una operación.
- *Reglas estructurales*: Especifican restricciones sobre los tipos de objetos y las asociaciones.

- **Reglas de Derivación:** Especifican políticas y condiciones para inferir o calcular hechos (información) a partir de otros hechos existentes en el negocio.
- **Reglas de Existencia:** Establecen cuándo puede existir un determinado objeto.

3.2.4 Tarea 4: Creación de un glosario de términos

Una de las mayores responsabilidades de los ingenieros de requisitos es evitar la ambigüedad de términos y conceptos relacionados con el negocio. Para esto es necesario crear un glosario donde se capturen las definiciones de los términos del dominio del problema, acrónimos, abreviaturas, etc, de tal modo que en cualquier etapa posterior no se tengan dudas sobre las expresiones utilizadas en los artefactos producidos durante el flujo de trabajo.

En este glosario se deben incluir las reglas del negocio (de restricción, de derivación y de existencia) identificadas en el paso anterior.

3.3 ELICITACIÓN DE REQUISITOS

El Instituto de Ingeniería de Software (SEI[7]) define la elicitación de requisitos como el proceso de identificar necesidades y conciliar las disparidades entre las comunidades involucradas para el propósito de definir y refinar los requisitos para satisfacer las restricciones de esas comunidades.

Para Pohl, el objetivo de la elicitación es hacer explícito el conocimiento oculto sobre las necesidades de clientes y usuarios y el sistema a desarrollar, de forma que todos los participantes en el problema sean capaces de entenderlo.

En [Durán, 2000] se considera la elicitación como "la actividad de la ingeniería de requisitos en la que se estudia el dominio del problema y se interactúa con los clientes y usuarios para obtener y registrar información, y si es necesario, negociar los requisitos que deberá satisfacer el sistema a desarrollar, desde el punto de vista de clientes y usuarios."

La elicitación es reconocida por muchos autores como una de las actividades más críticas en el proceso de desarrollo de software, dado que una pobre ejecución de esta actividad puede ocasionar el fracaso del proyecto. Adicionalmente, por ser una actividad en la que se involucran personas con diferentes niveles de conocimiento, expectativas del sistema, dificultades para expresar sus necesidades, entre otras, se dificulta la obtención, por parte del ingeniero de requisitos, del conjunto de necesidades reales que se deberán implementar en el producto software.[8]

Para realizar la elicitación de requisitos se propone realizar las siguientes tareas:

1. Identificación de Participantes

[7] http://www.sei.cmu.edu
[8] Los problemas más representativos en la elicitación de requisitos se encuentran explicados con mayor detalle en la sección 1.1 "Problemas en la Ingeniería de Software".

2. Preparar y realizar las sesiones de elicitación

3. Identificar requisitos de información

4. Elaborar el Documento de Requisitos del Sistema

La tarea 5 "*Negociación de Requisitos*" es una actividad que se realiza de forma transversal a todo el proceso de elicitación y por ello no deber ser tomado como un paso siguiente a la tarea 4. La negociación es un proceso constante en la elicitación y va implícito en cada de sus actividades. Además, esta tarea se debe ejecutar siempre que se deseen resolver los conflictos identificados durante las actividades de análisis y validación de requisitos.

A continuación se describe detalladamente cada una de estas tareas.

3.3.1 Tarea 1: Identificación de Participantes (Stakeholders)

La captura de requisitos es, principalmente, un proceso de comunicación. No se limita a la interacción unidireccional entre un usuario y un analista de requisitos, sino que involucra a muchas personas y una transferencia bidireccional de información. Por consiguiente, uno de los mayores problemas que se tiene en la captura y gestión de requisitos es, en sí, la comunicación; o más bien dicho, la falta de comunicación y los defectos que pueda presentar ésta.

Muchos son los stakeholders que deben participar en un proceso de ingeniería de requisitos, y la exclusión de alguno, la diferencia en puntos de vista de entre ellos, las ambigüedades y las malas interpretaciones, pueden causar desde retrasos en el proyecto, correcciones, hasta su fracaso. Por esta razón, se debe tener muy en cuenta cuáles son los participantes relevantes y el número de éstos, que deben estar presentes para tener una mayor posibilidad de obtener un conjunto de buenos requisitos.

3.3.1.1 Exclusión de Stakeholders

Antes se pensaba que para obtener la comunicación necesaria a la hora de capturar requisitos era suficiente tener como actores durante el proceso a quien documenta la especificación de requisitos y al que los brinda (Generalmente eran el jefe de proyecto y el cliente), y se excluía al resto de los stakeholders, como por ejemplo los usuarios, quienes son los que efectivamente van a utilizar el sistema.

Estos stakeholders son excluidos, generalmente, con la intención de bajar costos o de apresurar el proceso de desarrollo del sistema, generando como resultado una captura de requisitos en poco tiempo, pero obteniendo también una especificación relativamente pobre y la mayoría de las veces incompleta. Este supuesto ahorro de costos y tiempo conseguido a través de la exclusión de participantes provocará, en el largo plazo, un aumento de costos por correcciones incorporadas por los stakeholders que no fueron incluidos en la elicitación, como por ejemplo los operadores del sistema y quienes lo desarrollan.

Basándonos en la exclusión de participantes para realizar la captura de requisitos, se logra observar que este proceso se torna ineficiente al involucrar 3 sub-procesos dentro de los cuales los principales protagonistas son el jefe de proyecto y el cliente. Esta situación se ilustra en la figura 34.

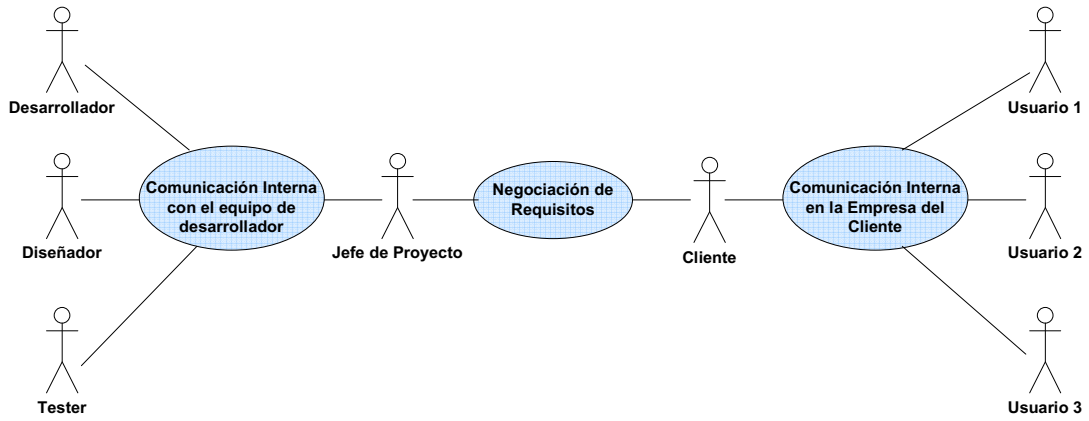


Figura 34. *Proceso Usual de captura de requisitos*

Fuente: [Rojas, 2006]

El cliente captura los requisitos de los usuarios del sistema u otros stakeholders dentro de la empresa. Por otro lado, el jefe de proyecto recibe las correcciones y comentarios sobre el sistema por parte de desarrolladores y diseñadores de software. Después de estos procesos, tanto el jefe del proyecto como el cliente, realizan un proceso mental (o escrito) interno en el cual "filtran" requisitos con base en sus puntos de vista, proceso en el cual se modifican, eliminan o agregan requisitos, para luego hacer la negociación de estos en un tercer proceso entre ellos. Algunas metas y motivaciones de los demás participantes del proyecto son dejadas a un lado, provocando que en la etapa de pruebas de aceptación, esta negligencia provoque correcciones en el sistema, y por consiguiente, retrasos en las entregas y problemas en el presupuesto y plazos.

3.3.1.2 Principales Stakeholders a ser involucrados

Los stakeholders del sistema pueden ser muy variados, igual que sus motivaciones para el éxito y sus respectivos aportes para el proyecto, aunque naturalmente, cada uno de los aportes entregados por los participantes deben ser tomados en cuenta, con el fin de no generar insatisfacciones o reproceso. A continuación se define cada uno de los principales participantes del proceso de captura de requisitos [Rojas, 2006]:

- **Cliente**: El principal contacto con la empresa, y quien finalmente pagará por el proyecto de software que se desarrolla. Usualmente (pero no necesariamente) él es quien decide los requisitos y contacta al equipo de desarrollo para que éste genere el sistema.

  Hay muchos tipos de clientes, puede que incluso el usuario sea un cliente. Sin embargo, generalmente se usa la palabra "cliente" para denominar al contacto principal con la empresa para la cual se desarrollará el proyecto.

- **Usuario**: La persona que interactúa directamente con el sistema desarrollado. A menudo, no es la misma persona que el cliente. Los usuarios pueden ser de 2 tipos:
  - **Usuario Líder:** Son los que comprenden el ambiente del sistema o el dominio del problema en donde será empleado el software desarrollado. Ellos proporcionan al equipo técnico los detalles y requisitos de las interfaces del sistema.
  - **Usuario Final:** Personas que usarán el sistema desarrollado, quienes utilizarán las interfaces y los manuales de usuario.
- **Jefe de Proyecto**: Líder del equipo de desarrollo. Su misión es planificar, motivar, organizar y controlar a los profesionales que realizan el trabajo de software.
- **Ingeniero de Requisitos**: Es el encargado de capturar, analizar, documentar y validar las necesidades de los demás stakeholders. Juega un rol central en la recolección y diseminación de la información acerca del producto.
- **Desarrollador**: Es quien diseña, implementa y mantiene el software del sistema.
- **Tester**: Es quien se asegura que todo el sistema funcione apropiadamente y de acuerdo con lo planeado. Generalmente basa su trabajo en estándares de calidad de software (CMMI, ISO, etc.)

Como ya se dijo, las motivaciones de cada stakeholder son variadas, al igual que el aporte de cada uno de ellos. Ambos factores se muestran en la tabla 11.

Stakeholder	Motivación	Aporte
Cliente	Introducir el cambio (que el sistema a desarrollar traerá consigo) con el máximo beneficio	Estrategias de negocio y de sistemas de información, tendencias de la industria
Usuario (Operador)	Introducir el cambio con la menor cantidad de interrupciones posible	Procesos de negocio, procedimientos operacionales
Jefe de Proyecto	Completar el proceso de desarrollo exitosamente con los recursos disponibles	Gestión de proyecto, procesos de desarrollo y entrega de software
Analista de requisitos	Especificar requisitos a tiempo y dentro del presupuesto	Métodos y herramientas de ingeniería de requisitos
Desarrollador	Producir un sistema técnicamente excelente, usar las tecnologías más recientes	Tecnologías más recientes, métodos de diseño, ambientes y lenguajes de programación
Tester	Asegurar la conformidad del producto con los estándares de producto y procesos	Procesos, estándares y métodos de software

Tabla 11. *Deseos y aportes de los Stakeholders*

Fuente: [Rojas, 2006]

Se debe tener claro los problemas que pueden ocurrir si no se cuenta con los stakeholders principales en el proceso de adquisición de requisitos. Por tanto, se propone utilizar los participantes expuestos para obtener unas necesidades más concretas y efectivas y con esto generar requisitos menos volátiles.

En el momento de seleccionar los participantes, se debe conocer que la cantidad de éstos también puede generar problemas, puesto que al aumentar el número de personas reunidas para realizar la captura, se puede perder de vista el objetivo y puede ser más difícil llevar el orden de las reuniones; por tanto, el ingeniero de requisitos debe tenerlo en cuenta al momento de organizar las actividades de elicitación.

La propuesta, como mínimo, muestra a los stakeholders más relevantes que deben ser parte de las actividades de elicitación; sin embargo, la participación y el número de dichos stakeholders es adaptable, según el proyecto que se vaya a ejecutar.

3.3.2 Tarea 2: Preparar y realizar las sesiones de elicitación

Esta tarea puede considerarse como una de las más críticas dentro de la ingeniería de requisitos, pues en ella se presenta la interacción más representativa entre el equipo desarrollador, los clientes y los usuarios. Uno de los retos más importantes de la elicitación es garantizar que los requisitos del sistema sean consistentes con las necesidades de la organización y con las futuras necesidades de los usuarios. Las reuniones y actividades de elicitación permiten conocer y comprender dichas necesidades y resolver todos los conflictos que puedan resultar, con el fin de obtener información de importancia para construir un primer bosquejo de los requisitos-C e identificar posibles cambios en ellos.

Esta tarea requiere de gran cuidado, pues generalmente el equipo de desarrollo no conoce a profundidad la forma como se mueve el negocio y los detalles de las operaciones que allí se ejecutan; así mismo, los clientes y usuarios no están conscientes de qué es lo que realmente necesita saber el equipo de desarrollo para llevar a cabo su labor. Por lo anterior, se podría afirmar que de estas sesiones y de la información que se pueda extraer de ellas depende en gran medida el éxito del proyecto.

RD SP 1.1 - Elicitar las Necesidades
Esta tarea asegura el cumplimiento de la primera práctica del área de proceso RD del modelo CMMI "Elicitar las necesidades", a través de las diferentes actividades con las cuales se propone capturar las expectativas y restricciones sobre las cuales se sentarán las bases para llevar a cabo el proyecto.

Entre las técnicas más utilizadas para hacer la elicitación de requisitos, se encuentran:

- Entrevistas y cuestionarios
- Joint Application Development (JAD)

- Brainstorming
- Workshops
- Casos de Uso

Para llevar a cabo esta tarea se propone utilizar tres técnicas: diagramas de afinidad, diagramas de árbol y entrevistas. Las dos primeras son herramientas de la metodología QFD y la razón principal por la cual fueron seleccionadas para llevar a cabo el proceso de elicitación de requisitos, es que permiten identificar, recolectar y organizar la información de una forma rápida y eficiente. Adicionalmente, estas técnicas facilitan la construcción de la Casa de la Calidad, pues agrupan la información según la estructura de la casa.

Cabe resaltar que para la elaboración de los diagramas de afinidad y de árbol es necesario realizar sesiones de lluvia de ideas o brainstorming, las cuales permiten la captura de, aproximadamente, el 90% de la información relevante.

A través de estas técnicas se podrán afinar los objetivos, identificar potenciales vacíos en la percepción del problema por parte de los clientes y usuarios, clarificar aspectos puntuales o confrontar posibles incongruencias, y mantener a todo el equipo vinculado a las metas y sub-metas generales de una tarea.

Generalmente, las sesiones de grupo no permiten profundizar en algunos aspectos del sistema que son conocidos con mayor detalle por ciertos participantes. En estos casos se recomienda la realización de entrevistas para completar la información suministrada en las sesiones grupales.

Dado que una parte importante dentro de la aplicación de QFD es la recolección de las necesidades del usuario, se podrían utilizar diferentes técnicas, enmarcadas en el campo de la creatividad, para el desarrollo de productos y servicios que tratan los tres niveles de información: Lo que se *dice*, lo que se *piensa* y lo que se *hace.* Entre estas técnicas se encuentran:

- Visualización: Maquetas, personas, collages, storyboards y escenarios.
- Contexmapping: Cuadernos de análisis cultural (Probes), observación con el método de cartas de IDEO (www.ideo.com).
- Creative Problem Solving

3.3.2.1 Elaborar diagramas de afinidad

Conocer las categorías y conceptos de la comunidad de clientes y usuarios es fundamental para poder entender las necesidades de dicha comunidad y su forma de expresarlas. Gracias a las tareas indicadas en la sección 3.2 que se llevaron a cabo para conocer a profundidad el negocio y el alcance del problema a resolver, el ingeniero de requisitos tendrá mayores bases para preparar la sesión y podrá generar confianza en los clientes y usuarios al reflejar un entendimiento de sus problemas.

La sesión para construir un diagrama de afinidad debe estar liderada por el ingeniero de requisitos, quien deberá ser una persona creativa, que genere confianza en el grupo y no se ponga nerviosa; debe evitar que los participantes monopolicen la reunión y fomentar que las personas más tímidas y cohibidas participen; debe evitar los silencios, resumiendo ideas obtenidas al final de cada aspecto tratado y volviéndolas a exponer, así mismo deberá moderar la participación de los miembros y controlar el cumplimiento de las reglas.

Pasos:

1. **Informar a los stakeholders sobre la sesión:** El ingeniero de requisitos deberá comunicar a todos los stakeholders sobre la actividad que se realizará, la fecha y la hora exactas de la reunión, y deberá expresarles la importancia de su presencia para lograr el éxito del proyecto. El número ideal de personas que deben participar en la sesión está considerado entre 4 y 10. Los participantes para esta actividad de elicitación de requisitos son normalmente clientes,

usuarios, ingenieros de requisitos, desarrolladores y, si es necesario, algún experto en temas relevantes para el proyecto.

2. **Preparar el ambiente:** En cuanto a las condiciones físicas, se debe disponer de un lugar espacioso y cómodo y de un tablero o un papelógrafo para exponer las ideas que surjan. Los participantes deberán colocar las sillas en círculo alrededor de una mesa.

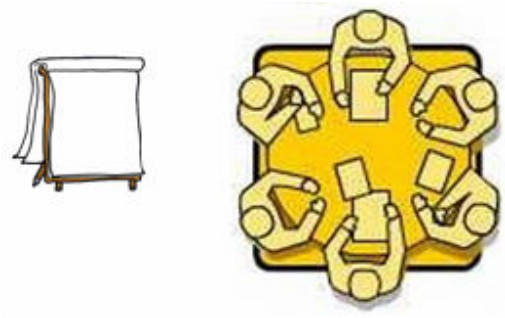


Figura 35. *Disposición de los participantes para construir un diagrama de afinidad*
Fuente: Elaboración propia

Se debe preparar todo lo necesario, desde documentos, papel y bolígrafo, hasta agua, bebida y comida, ya que dependiendo de la situación a analizar, estas sesiones se pueden volver algo extensas.

En cuanto a las condiciones humanas, las personas que componen el grupo deben estar motivadas para solucionar el problema, y con un ambiente que propicie la participación de todos. Deberán sentirse confiados y con la sensación de que pueden hablar sin que se produzcan críticas.

Todas las ideas en principio deberán tener el mismo valor, pues cualquiera de ellas puede ser la clave para dar inicio a la solución. Cabe anotar que durante la celebración no deberán asistir espectadores.

3. **Plantear el tema o situación:** El moderador deberá comenzar explicando el tema acerca del cual se va a hablar, el por qué del mismo y cuáles son los problemas que se han identificado en las tareas anteriores. Luego se fijan los objetivos de la reunión y los tiempos disponibles para tratar cada tema.

   Una vez terminada la sesión, es necesario pedir a los participantes que no abandonen el tema, y que anoten todas las ideas que les surjan; esto es importante ya que el alejamiento de la tensión de la reunión facilita asociaciones con otros aspectos. Las mejores ideas se podrían producir en este período para tratarlas en una siguiente sesión, bien sea para construir diagramas de afinidad o para aplicarlas en otras técnicas.

4. **Generar lluvia de ideas:** Cabe anotar que hasta este punto aun no se tienen identificadas las necesidades concretas de los clientes y usuarios, ni cuáles son los objetivos que se desean lograr a través del sistema. Este es uno de los pasos esenciales dentro de la técnica del diagrama de afinidad, pues permitirá generar una gran variedad de necesidades y objetivos, que servirán como base para definir formalmente los requisitos y cumplir así con el objetivo de la elicitación.

   Fases:

   ### 4.1 Generación de ideas para obtener requisitos funcionales

   La primera parte de este paso puede constituirse para generar ideas alrededor de los requisitos funcionales y la determinación del ámbito del sistema; esto se refiere a qué aspectos serán responsabilidad del sistema y qué aspectos se gestionarán manualmente o por otro procedimiento.

   Para llevar a cabo lo anterior, el ingeniero de requisitos puede plantear alguna pregunta que sirva como punto de partida para ir generando ideas. Se pueden tener en cuenta preguntas como: ¿Qué debe hacer el sistema con la

información que se maneja para alcanzar los objetivos del negocio?, ¿Un usuario debe ser informado sobre ciertos sucesos en el sistema?, ¿Se pueden cumplir todas las necesidades con los requisitos identificados?, ¿Qué información se debe crear, modificar en el sistema?, ¿Qué información se debe extraer del sistema?

Los participantes deben aportan libremente nuevas ideas de forma aleatoria hasta que el moderador decida parar, bien sea porque no se están generando suficientes ideas alrededor del tema y este se debe postergar, o porque el número de ideas ha sido suficiente para pasar a la siguiente fase.

4.2 Generación de ideas para obtener requisitos no funcionales

La segunda parte se constituye alrededor de la identificación de los requisitos no funcionales. Estos requisitos están relacionados con aspectos técnicos o legales enmarcados dentro de características como la **facilidad de uso** (tiempo de capacitación, disponibilidad de ayudas, tips y documentación), la **confiabilidad** (disponibilidad, exactitud, promedio de fallos, tiempo promedio de buen funcionamiento), el **rendimiento** (tiempo de respuesta, uso de recursos, capacidad de procesamiento) y el **soporte** (lenguajes de programación, bases de datos, herramientas, estándares de programación, sistema operativo, servidor de aplicaciones).

5. **Transferir los datos a notas Post-it o en tarjetas:** Cada idea se registra en hojitas Post it y se ubica en forma dispersa en un tablero; las ideas deben estar en letras grandes y oscuras para que sean visibles desde una distancia de un metro. [Mazur, 2001] menciona que la afinidad se establece "como si se estuvieran acomodando animales en un parque zoológico".

   Se deben usar de 4 a 7 palabras, como mínimo un sustantivo y un verbo, y se debe evitar el uso de palabras aisladas y expresiones complicadas. Normalmente un diagrama de afinidad se compone de 40 a 60 elementos.

Para los requisitos funcionales se debe tener en cuenta anotar debajo de la idea propuesta el rol de la persona que estará encargada de interactuar con el sistema para llevar a cabo la acción, tal como lo muestra la figura 36.

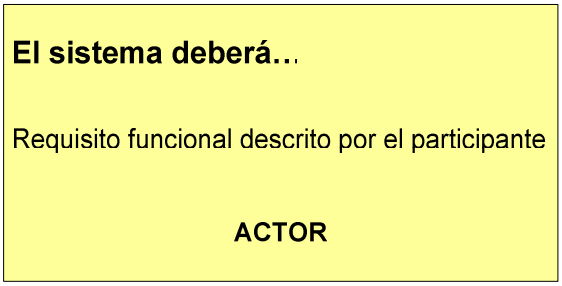


Figura 36. *Formato de las tarjetas para los requisitos*
Fuente: Elaboración propia

6. **Seleccionar un moderador que reúna las ideas en grupos similares:** Los miembros del equipo deben leer muy bien todas las ideas y reunirlas en grupos que congreguen ideas con *afinidad mutua.*

   Luego de que se hayan agrupados dos o tres tarjetas, éstas deben ser etiquetadas en una nueva tarjeta y se debe verificar que estén agrupadas de forma apropiada. El título del grupo debe ser corto pero muy significativo.

   Aquellas ideas que no encajen en el grupo deben ser removidas y llevadas a un grupo de *no clasificadas, aisladas* o *misceláneas.*

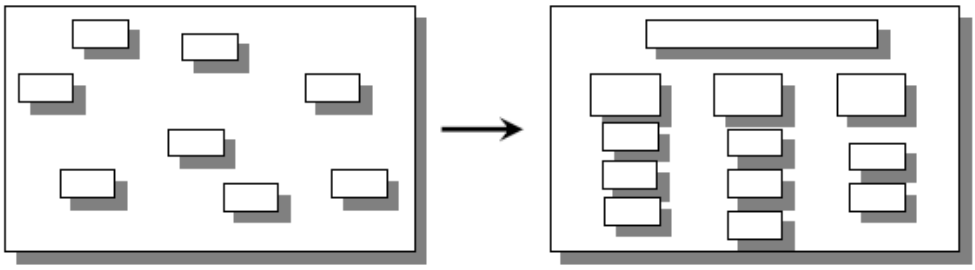


Figura 37. *Generación de un diagrama de afinidad*
Fuente: [Min. Fomento, 2005]

En caso de que otra persona del grupo decida mover una idea hacia un grupo diferente al considerado por otro participante, se pueden crear duplicados con el fin de analizar la idea desde varios puntos de vista.

Para los grupos de ideas muy grandes puede aplicarse de nuevo el proceso y dividirlo en subgrupos, tantas veces como sea necesario.

En la figura 37 se ilustra cómo se podría organizar un grupo de necesidades según su afinidad.

Figura 38. *Ilustración de agrupación de ideas por afinidad*
Fuente: Elaboración propia

7. **Dibujar el diagrama de afinidad:** El siguiente paso es dibujar un diagrama de los grupos formados en el paso anterior. Para este propósito, los grupos de tarjetas recolectados son colocados de forma tal que muestren su mutua relación. Estas tarjetas se pegan en una hoja y se usan símbolos para indicar su mutua relación.

   Para ello se escribe la formulación del problema en la parte superior del diagrama. Posteriormente se situarán los encabezamientos sobre sus respectivos grupos de ideas. Se pueden dibujar líneas conectando encabezados primarios y secundarios, grupos y subgrupos indicando la

relación existente. El resultado final se asemejará bastante a un diagrama organizacional.

Un esquema de cómo dibujar el diagrama de afinidad se puede ver en la figura 39.

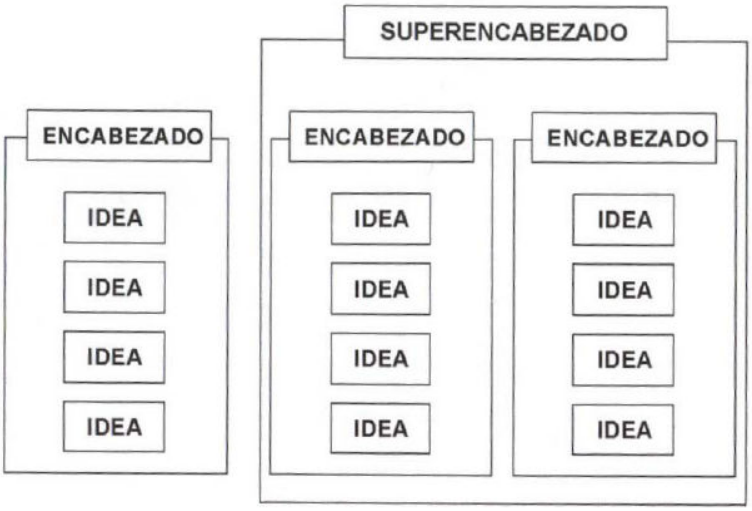


Figura 39. *Esquema de un diagrama de afinidad*
Fuente: [Min. Fomento, 2005]

Adicionalmente, se debe incluir el actor que realiza cada una de los requisitos-C. El diagrama de afinidad que se muestra en la figura 40 fue realizado a partir de las ideas mostradas en la figura 38.

Figura 40. *Ejemplo de un diagrama de afinidad*

Fuente: Elaboración propia

El ingeniero de requisitos deberá pedirle a los participantes que describan las principales características de los actores identificados.

8. **Discusión:** Por último se presenta el contenido del diagrama y se explica el contenido de cada tarjeta dentro del grupo al cual pertenece. El grupo puede discutir el contenido del diagrama definiendo nuevas ideas, utilizando expresiones como “Esto es”, “Creo”, “Parece”.

   Los productos de esta tarea son los objetivos del sistema representados en los títulos de cada agrupación y los requisitos representados en las ideas que conforman los grupos, las cuales han sido identificadas y revisadas en las tareas anteriores.

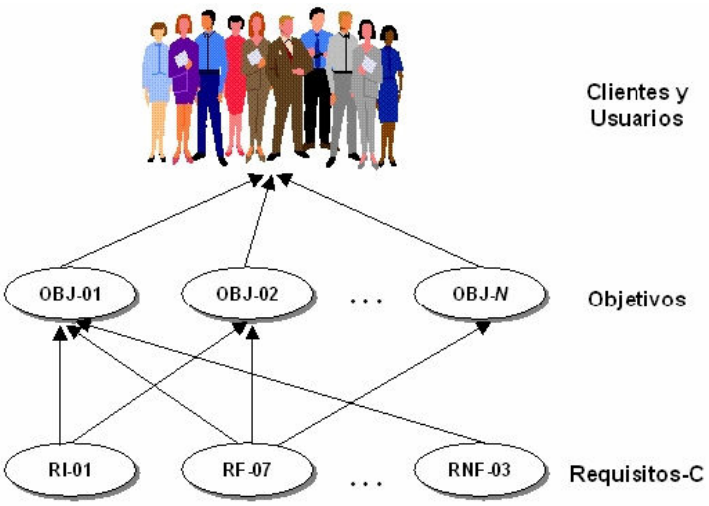


Figura 41. *Relación entre objetivos y requisitos-C*

Fuente: [Durán, 2000]

3.3.2.2 Elaborar diagrama de árbol

Los árboles, aunque son generalmente utilizados como una herramienta de análisis, en la presente propuesta serán utilizados para encontrar requisitos

implícitos en las necesidades que se lograron extraer con la elaboración de los diagramas de afinidad. Además, permiten realizar comparaciones y priorizaciones de las necesidades para determinar su grado de importancia en el sistema a desarrollar.

Los árboles de jerarquía ilustran la estructura de las interrelaciones entre los grupos de necesidades identificados en los diagramas de afinidad.

Por medio de esta herramienta se pueden identificar restricciones del sistema y requisitos de información que no hayan sido determinados hasta el momento.

Pasos:

1. **Establecer el objetivo:** Antes de iniciar esta actividad de elicitación se debe expresar a los participantes de la reunión que el objetivo de esta actividad es profundizar un poco más en cada una de las necesidades identificadas en la actividad anterior.

2. **Estructurar el árbol:** La necesidad más general (objetivo) se ubica al lado izquierdo de la página para desprender de ahí las necesidades específicas correspondientes a este objetivo.

   Cada nivel del árbol debe generarse a través de preguntas para cada una de las necesidades de los niveles superiores; por ejemplo, se podrían utilizar preguntas tales como: ¿Qué información está relacionada con esta necesidad? ¿Cómo cumplir esta necesidad? Para responder a estas preguntas, es útil continuar con la técnica de lluvia de ideas.

   Si se desea comprobar la lógica del diagrama, se puede plantear la pregunta de izquierda a derecha ¿cómo cumplir la necesidad? y de derecha a izquierda ¿para qué se necesita hacer esto?

3. **Identificar necesidades implícitas:** Si es necesario, se añaden más necesidades para completar el árbol, validando la lógica explicada en el punto anterior. Normalmente son necesidades que están implícitas y que su identificación requiere de un análisis profundo.

   Cabe aclarar, que las nuevas necesidades también deben ser ingresadas en los diagramas de afinidad con el fin de asegurar la consistencia de la información.

4. **Priorizar necesidades:** Luego se debe pedir al cliente y a los usuarios que asignen una prioridad a las necesidades, con el fin de identificar sobre cuáles de ellas se debe centrar el grupo de trabajo, estableciendo su importancia y su urgencia, de tal forma que si se va a hacer un desarrollo iterativo e incremental, se tengan los criterios suficientes para saber cuáles requisitos deben implementarse en cada versión y, de esta manera ir depurando aquellos que realmente no son necesarios o no encajan con la esencia del proyecto. La idea básica es ir obteniendo los requisitos como un refinamiento de los objetivos, de forma que la existencia de un requisito esté siempre justificada como una necesidad para alcanzar uno o más objetivos.

Es muy probable que durante las sesiones de lluvia de ideas realizadas para la elaboración de los diagramas de afinidad y de árbol, el ingeniero de requisitos identifique algunas ideas que deben ser analizadas con mayor profundidad o porque vea la necesidad de aclarar aspectos que considere importantes con algunos de los participantes. En estos casos se deberá recurrir a las entrevistas como medio de elicitación / negociación de requisitos.

Así mismo, al ser una de las formas de comunicación más natural entre las personas, será posible que puedan expresar más fácilmente cuáles son los servicios o capacidades que esperan de un sistema en cuanto a las características mencionadas en el paso 4.2 "*Generación de ideas para obtener requisitos no funcionales",* de la técnica Diagramas de Afinidad; de esta forma será posible

clarificar cuáles serán los requisitos no funcionales esperados por los clientes y usuarios.

3.3.2.3 Entrevistas

El objetivo de las entrevistas es ampliar la información contenida en los diagramas de afinidad, por esto se recomienda llevar como material para la entrevista los diagramas obtenidos en las sesiones de lluvia de ideas.

Las preguntas pueden ser enfocadas a distintos elementos del sistema, como por ejemplo usuario, proceso, producto, entre otros. Estas preguntas deben ser de alto nivel y realizadas al comienzo del proceso, para así lograr la recaudación de aspectos globales del problema. El éxito del uso de esta técnica dependerá de la experiencia y sensibilidad del entrevistador, ya que además de preparar estratégicamente el curso de la entrevista, debe ser capaz de interpretar la significancia de las respuestas. [Bahamonde y Rossel, 2003]

Pasos:

1. **Seleccionar a las personas a quienes se va a entrevistar:** No es recomendable realizar un gran número de entrevistas, por lo cual se deben seleccionar aquellos participantes con los cuales se desea ampliar algún punto específico que no haya quedado claro durante las sesiones anteriores o porque son participantes que pueden ofrecer nuevas perspectivas y visiones en cuanto a requisitos nuevos y/o existentes.

2. **Planificar la entrevista:** Es importante tener muy claramente definido cuál será el objetivo de la entrevista con cada persona, esto permitirá ahorrar tiempo y centrarse realmente en los puntos que sea necesario aclarar para evitar tocar temas que puedan ser irrelevantes.

Así mismo, se debe establecer la hora, la fecha, el lugar y la duración de la entrevista. Se recomienda que el lugar esté libre de interrupciones y donde el entrevistado se sienta cómodo.

3. **Realización de las entrevistas:** Inicialmente el entrevistador debe romper el hielo explicándole al entrevistado de la manera más natural lo que se pretende con esta actividad, cómo se utilizará la información obtenida allí, etc.

   El entrevistado es quien debe tener el control sobre la entrevista, es decir, su aporte durante la reunión debe ser aproximadamente del 80%, mientras que el entrevistador será quien se dedique a realizar preguntas y descubrir opiniones o experiencias que se tengan con respecto al proyecto.

   Las preguntas que se realicen durante las entrevistas se conocen como *Libres de Contexto,* es decir, no pueden responderse con un *"sí"* o un *"no"*, deben ser formuladas de modo que ayuden a entender la perspectiva del participante, no estar predispuestas con sugerencias acerca de la solución y no deben contener opiniones propias. Estas preguntas deben realizarse al inicio de la entrevista para luego pasar a preguntas más concretas.

   Ejemplos de preguntas que no son libres de contexto pueden ser: ¿Necesita usted una pantalla más amplia o no?, "Tengo una pregunta dividida en tres partes…", ¿Esto se puede realizar de esta manera?, ¿No es más fácil si nos conectamos a…?, ¿No considera que esto es muy complejo?, ¿Por qué se debe hacer de esa forma y no de esta?

   Para capturar los requisitos no funcionales se deben realizar preguntas que permitan identificar los factores de calidad esperados por el cliente y usuario. Ejemplos de este tipo de preguntas pueden ser: ¿Cuáles son sus expectativas con respecto a la forma de usar el sistema?, ¿Cuál es el rendimiento deseado?

4. **Conclusión:** Al culminar la entrevista se debe retomar todo lo que se ha logrado obtener de información con el fin de confirmar que no ha habido algún tipo de confusión. Se debe dejar abierta la posibilidad de volver a contactar al entrevistado para aclarar dudas que surjan al estudiar la información o al contrastarla con otros entrevistados.

5. **Análisis de la entrevista:** Luego de terminar la entrevista se deben pasar en limpio las notas, reorganizar la información y compararla con la de otras entrevistas para identificar en qué difieren o en qué se asemejan. Cabe la posibilidad de enviarle esta información al entrevistado para que la revise y corrija cualquier inconsistencia que encuentre y así tener una información más confiable para construir los requisitos.

3.3.3 Tarea 3: Identificar requisitos de información

En esta tarea en concreto se deben identificar o revisar los requisitos de almacenamiento de información que deberá satisfacer el sistema. Normalmente estos requisitos son la respuesta a la pregunta *¿qué información, relevante para los objetivos de su negocio, deberá almacenar el sistema?*

Cabe aclarar que esta tarea no es secuencial, es una tarea que se va cumpliendo implícitamente durante las tareas anteriores de elicitación, donde se van capturando ciertas necesidades que tienen que ver con los datos que el sistema deberá manejar.

Dado que en las sesiones de lluvia de ideas realizadas es difícil capturar este tipo de requisitos en su totalidad, se recomienda realizar entrevistas a las personas que mayor conocimiento tengan sobre los datos e información vital para la operación del negocio, apoyándose en formatos digitales o físicos que se estén

utilizando en ese momento para llevar a cabo los registros o en la información almacenada en otros sistemas.

Para expresar este tipo de requisitos, se puede usar la plantilla propuesta por Amador Durán en [Durán, 2000] ilustrada en la figura 42.

RI–<*id*>	<*nombre descriptivo*>
Versión	<*nº de la versión actual*> (<*fecha de la versión actual*>)
Autores	• <*autor de la versión actual*> (<*organización del autor*>) …
Fuentes	• <*fuente de la versión actual*> (<*organización de la fuente*>) …
Requisitos asociados	• Rx–y <*nombre del requisito*> …
Descripción	El sistema deberá almacenar la información correspondiente a <*concepto relevante*>. En concreto:
Datos específicos	• <*datos específicos sobre el concepto relevante*> • …
Importancia	<*importancia del requisito*>
Urgencia	<*urgencia del requisito*>
Estado	<*estado del requisito*>
Estabilidad	<*estabilidad del requisito*>
Comentarios	<*comentarios adicionales sobre el requisito*>

Figura 42. *Plantilla para requisitos de información*

Fuente: [Durán, 2000]

A continuación se explicará el significado que le da Amador Durán [Durán, 2000] a cada campo de la plantilla:

1. **Identificador y nombre descriptivo:** Siguiendo las recomendaciones de la IEEE, cada requisito se debe identificar por un código único y un nombre descriptivo. Por simplicidad este código puede especificarse como *RI*.

2. **Versión:** Para poder gestionar distintas versiones, este campo contiene el número y la fecha de la versión actual del objetivo.

3. **Autores, Fuentes:** Estos campos contienen el nombre y la organización de los autores (normalmente desarrolladores) y de las fuentes (clientes o usuarios),

de la versión actual del objetivo, de forma que la trazabilidad pueda llegar hasta las personas que propusieron la necesidad del requisito.

4. **Descripción**: Para los requisitos de almacenamiento de información este campo usa un patrón lingüístico (*"El sistema deberá almacenar la información correspondiente a"*) que se debe completar con el concepto relevante sobre qué información debe almacenar el sistema.

5. **Datos específicos**: Este campo contiene una lista de los datos específicos asociados al concepto relevante, de los que pueden indicarse todos aquellos aspectos que se consideren oportunos (descripción, restricciones, ejemplos, etc.).

6. **Requisitos asociados**: En este campo se indican otros requisitos que estén asociados por algún motivo con el requisito que se está describiendo.

7. **Importancia:** Este campo indica la importancia del cumplimiento del objetivo para los clientes y usuarios. Se puede asignar un valor numérico o alguna expresión enumerada como *vital*, *importante* o *quedaría bien*; en el caso de que no se haya establecido aún la importancia, se puede indicar que está *por determinar.*

8. **Urgencia:** Este campo indica la urgencia del cumplimiento del objetivo para los clientes y usuarios en el supuesto caso de un desarrollo incremental. Como en el caso anterior, se puede asignar un valor numérico o una expresión enumerada como *inmediatamente*, *hay presión*, *puede esperar* o *por determinar.*

9. **Estado:** Este campo indica el estado del objetivo desde el punto de vista de su desarrollo. El objetivo puede estar *en construcción,* si se está elaborando; *pendiente de negociación,* si tiene algún conflicto a ser solucionado; *pendiente de validación,* si no tiene ningún conflicto pendiente y está a la espera de

validación o, por último, puede estar *validado,* si ha sido validado por clientes y usuarios.

10. **Estabilidad:** Este campo indica la estabilidad del objetivo, es decir una estimación de la probabilidad que pueda sufrir cambios en el futuro. Esta estabilidad puede indicarse mediante un valor numérico o mediante una expresión clasificada como *alta*, *media* o *baja* o *por definir,* en el caso de que aún no se haya determinado. La información sobre la estabilidad, bien a nivel de objetivos como en este caso, bien a nivel de requisitos, ayuda a los diseñadores a diseñar software que prevea de antemano la necesidad de posibles cambios futuros en aquellos aspectos relacionados con los elementos identificados como inestables durante la fase de ingeniería de requisitos, favoreciendo así el mantenimiento y la evolución del software.

11. **Comentarios:** Cualquier otra información sobre el objetivo que no encaje en los campos anteriores puede recogerse en este apartado.

Como resultado de esta actividad, se obtiene un documento con los diagramas de afinidad, un listado con los actores del sistema y las plantillas con los requisitos de información.

RD SP 1.2 - Desarrollar los requisitos del cliente
La utilización de las técnicas ejecutadas anteriormente permitirá darle cumplimiento a la práctica dos de la primera meta del área de proceso RD "Desarrollar los requisitos del cliente", pues a través de ellas será posible consolidar la información obtenida en cada uno de los pasos ejecutados; así mismo, por medio de la posterior fase de negociación, se pretenderá resolver los conflictos que resulten luego de ejecutar dichas técnicas.

RD SG 1 - Desarrollar requisitos del cliente
Con el cumplimiento de las dos prácticas específicas "Elicitar las necesidades" y "Desarrollar los requisitos del cliente" se asegura el cumplimiento de la primera meta del área de proceso Desarrollo de Requisitos.

3.3.4 Tarea 4: Elaborar el Documento de Requisitos del Sistema

El documento de los requisitos del software es una declaración escrita de lo que el software hará, pero no de cómo lo hará.

Lo que hará el software es percibido directamente por los usuarios (personas u otros sistemas de software). Cuando un usuario realiza cierta acción, el software responde de una manera particular y cuando un sistema externo presenta una petición, consigue una respuesta particular. La comprensión de estas acciones se captura en el documento de requisitos.

El documento de requisitos debe servir como un acuerdo o contrato con los clientes y puede ser usado por los desarrolladores y testers para implementar la aplicación y verificar que ésta cumpla con dicho acuerdo. Si realmente este documento se trata como un contrato de desarrollo de software, todas las partes estarán encaminadas hacia expectativas comunes sobre la aplicación, algo muy necesario para colaborar con el éxito del proyecto.

Algunos beneficios adicionales que este documento proporciona, están expuestos en [Berezin, 1999]:

- Los clientes podrán saber desde el inicio si sus necesidades serán resueltas.
- Los desarrolladores podrán estimar el tiempo que deben emplear para crear la aplicación.
- Las personas de aseguramiento de la calidad tendrán una base para probar la aplicación.

Para escribir un documento de requisitos es necesario tener en cuenta algunas recomendaciones que permiten que sea más legible, propuestas de nuevo por [Berezin, 1999]:

- Usar sentencias cortas, párrafos cortos, listas enumeradas o cualquier cosa que permita a los lectores darse una idea de lo que trata el documento.

- Escribir en voz activa cuando sea posible. Por ejemplo, *"El sistema pasará el conjunto de datos a la aplicación"* es mucho más claro que *"Los datos serán pasados a la aplicación".*

- Se debe estar seguro de que las secciones del documento estén definidas en un orden lógico.

- Se pueden ilustrar ciertos puntos con diagramas y tablas con el fin de hacerlos más claros.

- Evitar utilizar el lenguaje informático y cuando sea necesario definir los términos que se necesiten usar.

- Usar herramientas de revisión ortográfica.

- Para aplicaciones grandes o complejas, es conveniente dividir el documento en dos o más partes. Por ejemplo, si la aplicación interactúa con sistemas legados, la integración de los requisitos se puede realizar en un documento separado porque estos tienden a ser muy largos, muy técnicos y tienen una relación muy débil con el resto de los requisitos. Por otra parte, si la aplicación requiere características de seguridad y auditoría, estos requisitos pueden ponerse en documentos separados como un documento de interfaces de aplicaciones externas.

3.3.4.1 Partes del Documento de Requisitos

A continuación se hará una descripción detallada de las partes que conformarán el documento de requisitos, basados en algunas de las indicaciones dadas por Amador Durán en [Durán, 2000].

- **Portada:** Las partes que conforman la portada se pueden visualizar en la figura 43.

Figura 43. *Portada de Documento de Requisitos*
Fuente: [Durán, 2000]

- **Nombre del proyecto:** En esta parte se especifica el nombre del proyecto al cual pertenece el documento de requisitos.

- **Versión:** La versión se compone de dos dígitos, el primero indica el número de la entrega formal que se está haciendo al cliente, la cual se debe incrementar cada que haya una nueva. El segundo número indica cambios dentro de la misma versión aún no entregada, y se debe incrementar cada vez que se publica una versión con cambios respecto a la última que se publicó y que no se vaya a entregar formalmente todavía.

- **Fecha:** Fecha de la publicación de la versión.

- **Equipo de desarrollo:** Nombre de la empresa o equipo de desarrollo.

- **Cliente:** Nombre del cliente

- **Lista de cambios:** El documento debe incluir una lista de cambios en la que se especifiquen, para cada versión del documento, los cambios producidos en el mismo con un formato similar al que puede verse en la figura 44.

Núm.	Fecha	Descripción	Autores
0	$fecha_0$	Versión x.y	$autor_0$
1	$fecha_1$	$descripción\ cambio_1$	$autor_1$
⋮	⋮	⋮	⋮
n	$fecha_n$	$descripción\ cambio_n$	$autor_n$

Figura 44. *Lista de cambios*
Fuente: [Durán, 2000]

Para cada cambio realizado se debe incluir la fecha, una descripción y los autores.

- **Índice:** Incluir una tabla de contenido con el fin de darle claridad a la estructura del documento y permitirle al lector encontrar más fácilmente la información necesaria.

- **Lista de figuras y tablas:** Incluir dos índices en los cuales se muestren los listados de figuras y las tablas que estén en el documento.

- **Introducción:** Esta parte presenta a grandes rasgos, un esquema de la aplicación. Debe contener una descripción breve de las principales características del sistema software que se va a desarrollar, la situación actual que genera la necesidad del nuevo desarrollo, la problemática que se está tratando, y cualquier otra consideración que sitúe al posible lector en el contexto oportuno para comprender el resto del documento.

- **Procesos de negocio:** En este punto se describe el proceso del negocio y cómo la aplicación será usada en este contexto.

Cuando algunos procesos de negocio sean susceptibles de cambios con la introducción del nuevo sistema, es importante describir el proceso del negocio utilizando los sistemas existentes y describir el proceso futuro del negocio usando el sistema que se va a desarrollar.

- **Reglas de Negocio:** En esta sección se expresa la colección de políticas y restricciones del negocio, identificadas en la tarea 3 del modelado del negocio.
- **Roles del Entorno:** Se deben listar y describir cada uno de los agentes involucrados en la realización de los procesos de negocio.

Una vez especificados los objetivos, procesos, roles, políticas y restricciones del negocio, estos se deben mostrar a los expertos del negocio para que los lean y revisen si realmente el equipo ha logrado comprender las necesidades actuales del negocio.

- **Descripción de la situación actual:** En esta sección se deberá incluir una descripción detallada de la forma en la cual se realizan los procesos a sistematizar, haciendo énfasis en los problemas y dificultades que se presentan normalmente en la operación diaria.

En caso de que exista algún sistema y éste tenga alguna relación con el proyecto, se puede realizar una descripción de aquellos aspectos que se considere importante resaltar.

Se puede incluir un diagrama mostrando las interfaces entre el sistema a desarrollar y otros existentes.

- **Participantes y responsabilidad:** Esta sección debe contener una lista con todos los participantes en el proyecto, tanto desarrolladores como clientes y usuarios. Por cada participante se deberá indicar su nombre, el papel que desempeña en el proyecto, la organización a la que pertenece y cualquier otra información adicional que se considere oportuna.

- **Catálogo de Necesidades:** Esta sección se divide en las siguientes sub-secciones en las que se describen las necesidades identificadas durante el proceso de elicitación.

  - **Requisitos de almacenamiento de información:** En esta parte se deben incluir los requisitos de almacenamiento de información que se hayan identificado en la tarea 3 de la sección 3.3 *"Elicitación de Requisitos"*, utilizando la plantilla para requisitos de almacenamiento de información que se muestra en la figura 42.

  - **Diagramas de afinidad:** En esta sección se deberán incluir los diferentes diagramas de afinidad realizados en la tarea 2 de la sección 3.3 "*Elicitación de Requisitos*".

  - **Diagramas de Árbol:** En esta sección se deberán incluir los diferentes diagramas de árbol realizados en la tarea 2 de la sección 3.3 "*Elicitación de Requisitos*".

  - **Definición de los actores:** Se establece el listado de los actores que se hayan identificado durante la elaboración de los diagramas de afinidad.

- **Glosario de términos:** En esta parte se incluye el glosario de términos definido en la tarea 4 de la sección 3.2 *"Modelado del Negocio"*, preferiblemente en orden alfabético.

- **Apéndices:** Los apéndices son usados para capturar cualquier información que no encaje naturalmente en otro lugar del documento de requisitos, pero que se considera importante y oportuno incluir en el mismo. Se debe tener en cuenta identificar con letras mayúsculas ordenadas alfabéticamente (A, B, C).

  Algunos ejemplos de la información que puede incluirse en esta parte son:

  - Resultados de entrevistas

- Ejemplos de problemas a ser resueltos por la aplicación
- Información que es útil para una audiencia específica, por ejemplo, ciertas especificaciones que son de mayor relevancia para los desarrolladores que para los usuarios.
- Información para ampliar alguno de los puntos expuestos durante el documento.

En la figura 45 se puede observar un esquema del documento de requisitos:

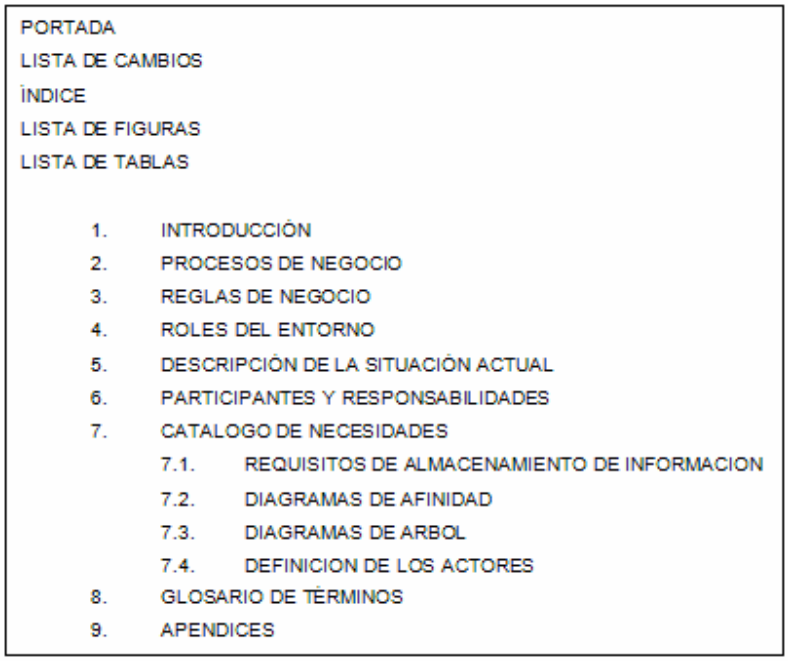


Figura 45. *Tabla de contenido del documento de requisitos*
Fuente: Elaboración propia

3.3.5 Tarea 5: Negociar Requisitos

"El desarrollo del proceso software se basa en una constante negociación entre el cliente y el desarrollador en busca del beneficio mutuo y constante, o sea, el cliente quiere ***"ganar"*** *y el desarrollador también quiere* ***"ganar"****, por lo que el*

centro de la negociación entre ambos adquiere una especial relevancia en la fase de los requisitos del sistema." [Granollers, 2004]

En todo proceso de ingeniería de requisitos intervienen distintos individuos y algunas veces existen intereses que se cruzan, por lo cual es importante estar abiertos al cambio y realizar una evaluación de los requisitos antes de definir si son factibles para el cliente, con el fin de estar seguros que brindarán un resultado completo, correcto y sin ambigüedades.

La negociación de requisitos, también denominada resolución de conflictos, se ocupa de resolver los problemas que puedan surgir en los requisitos por diversos motivos; por ejemplo, que haya peticiones por parte de clientes y usuarios que sean incompatibles o que no se disponga de los recursos necesarios para la realización de ciertos aspectos del sistema.

Para resolver estos conflictos, es necesario consultar con todos los participantes afectados y registrar tanto el conflicto como las decisiones tomadas y la persona que las tomó. Así mismo, es necesario añadir que los autores de estas decisiones asumen que los conflictos pueden aparecer no sólo durante el análisis, sino también durante la etapa de validación.

Los conflictos que se presenten no se deben resolver de forma autoritaria**,** sino mediante un proceso de negociación; por ello, para el cumplimiento de esta tarea se debe realizar un razonamiento sobre los requisitos obtenidos, con el fin de detectar posibles inconsistencias o conflictos. Esta tarea es propia del ingeniero de requisitos, aunque los usuarios pueden participar por si es necesario aclarar alguno y se deba entrar a negociar; de esta forma se incrementa la comunicación entre el equipo de desarrollo con los clientes y usuarios, asegurando en parte, el éxito del proyecto.

Algunos cuestionamientos que puede plantear el ingeniero de requisitos están propuestos en [Herrero, 2005]:

- ¿El requisito es necesario?
- ¿Tiene un nivel de detalle apropiado?
- ¿Está bien delimitado y sin ambigüedad?
- ¿Se solapa con otros requisitos?
- ¿Se puede probar el requisito una vez implementado?
- ¿Se dispone de presupuesto para ese requisito?

Pohl propone tener en cuenta 4 factores durante el proceso de negociación [Durán, 2000]:

- Hacer explícitos los problemas y evitar los conflictos emocionales entre los participantes, de forma tal que quede claro qué es lo que se negocia y que dicha negociación no se vea afectada por motivos personales.

- Hacer explícitos para cada conflicto las alternativas, las argumentaciones y las razones subyacentes que los provocan, de forma que la negociación pueda basarse en las raíces del conflicto.

- Asegurarse que se toman las decisiones correctas, de forma que la mayoría de los participantes estén de acuerdo con los resultados de la negociación y no se sientan desplazados del proceso.

- Asegurarse de involucrar a las personas adecuadas en el momento adecuado, para evitar tener que volver a replantear las negociaciones porque alguno de los participantes afectados no participó en las negociaciones oportunas.

Por último y muy importante, se deben tener en cuenta ciertos aspectos para que esta tarea sea exitosa y para que los requisitos sean comunicados de manera efectiva, las cuales están propuestas en [Monografías, Requisitos]:

- Documentar todos los requisitos a un nivel de detalle apropiado.
- Mostrar todos los requisitos a los involucrados en el sistema.

- Analizar el impacto que causen los cambios en los requisitos antes de aceptarlos.
- Establecer las relaciones entre requisitos que indiquen dependencias.
- Negociar con flexibilidad para que exista un beneficio mutuo.
- Enfocarse en intereses y no en posiciones.

3.4 ANÁLISIS DE REQUISITOS

Esta es la etapa de la ingeniería de requisitos en la cual el ingeniero de requisitos y los demás miembros del equipo de desarrollo se encargan de realizar un completo y detallado entendimiento de las necesidades elicitadas en la etapa previa.

El enfoque de la metodología permite que el cliente y el usuario también estén presentes en este momento, obteniendo mayor calidad y entendimiento de la información elicitada y generando satisfacción en el cliente. Además, esta es la etapa en la cual el cliente y los usuarios muestran las prioridades de las necesidades del sistema, con el fin de conocer en cuáles de ellas se debe centrar el grupo de trabajo.

En el proceso de análisis se encontrarán habitualmente gran cantidad de problemas en los Requisitos-C, áreas no especificadas, requisitos contradictorios, y afirmaciones (aparentemente) vagas e irrelevantes. Por consiguiente, al momento de volver donde el cliente y los usuarios para obtener la priorización de las necesidades, se debe buscar también mejorar la calidad de estos requisitos; sabiendo lo que se quiere conseguir, qué aspectos de los requisitos obtenidos inicialmente interesa aclarar, y su razón.

Los diagramas de afinidad, los diagramas de árbol y el resultado de las entrevistas se convierten en las entradas del proceso de análisis. En este momento, el

ingeniero de requisitos necesita convertir este gran volumen de información en modelos formales que puedan ser interpretados por las demás actividades del ciclo de vida del software. Los modelos formales que se propone utilizar son las matrices de relación y la Casa de la Calidad.

La actividad de análisis de los requisitos está divida en las siguientes tareas:

1. Clasificación de los requisitos
2. Identificación de conflictos en los requisitos
3. Realizar matrices de relación
4. Elaborar los cuestionarios de Kano
5. Ejecutar el ciclo Análisis – Elicitación
6. Aplicar el modelo de Kano
7. Construir la Casa de la Calidad
8. Elaborar el Documento de Análisis de Requisitos

A continuación se describe detalladamente cada una de estas tareas.

3.4.1 Tarea 1: Clasificación de los requisitos

Como primera tarea, la clasificación de los requisitos permitirá al equipo de trabajo y en especial al ingeniero de requisitos, conocer de forma clara los diferentes tipos de necesidades que han especificado los participantes.

Al identificar los diferentes grupos en los que se dividen las necesidades, se puede observar la estructura básica del sistema, revelando su organización interna, la

forma en la cual se va a comportar y la manera como se debe desempeñar en un ambiente específico. Estos tres aspectos mencionados se pueden vislumbrar clasificando las necesidades, respectivamente, en tres categorías:

- **Necesidades de información:** Este tipo de requisitos describen qué información debe almacenar el sistema para satisfacer las necesidades de clientes y usuarios. Identifican los conceptos relevantes sobre los que se debe almacenar información y los datos específicos que son de interés. La información de requisitos está condensada en las plantillas realizadas en la tarea 3 de la sección 3.3 *"Elicitación de Requisitos".*

- **Necesidades Funcionales**: Los Requisitos funcionales especifican acciones que el sistema debe ser capaz de realizar, sin tomar en consideración ningún tipo de restricción física. Los requisitos funcionales especifican el comportamiento de entrada y salida del sistema y surgen de la razón fundamental de la existencia del producto.

- **Necesidades No funcionales:** Este tipo de requisitos no forma parte de la funcionalidad principal de la aplicación, sino que son especificaciones del *cómo* debe funcionar la aplicación. Son difíciles de identificar pero deben ser capturadas y clasificadas para brindar una mayor satisfacción al cliente.

  Entre este tipo de requisitos se encuentran, entre otros, el entorno de desarrollo o ejecución (sistema operativo, servidores en los que correrá, lenguajes, etc), las restricciones que se aplicarán y las prestaciones (tiempo de respuesta mínimo, alta disponibilidad, etc.). A continuación se describen diferentes tipos de requisitos no funcionales, aunque pueden variar según el autor.

  - **Rendimiento**: Especificación de los requisitos relacionados con la carga que se espera, tenga que soportar el sistema. Por ejemplo, el número de

terminales, el número esperado de usuarios simultáneamente conectados, número de transacciones por segundo que deberá soportar el sistema, etc.

Todos estos requisitos deben ser medibles. Por ejemplo, indicando “el 95% de las transacciones deben realizarse en menos de 1 segundo”, en lugar de “los operadores no deben esperar a que se complete la transacción”.

- **Usabilidad:** Especificación de los elementos relacionados con la capacidad del producto de software para ser comprendido, aprendido, utilizado y atractivo para el usuario.

- **Seguridad**: Especificación de elementos que protegerán al software de accesos, usos y sabotajes maliciosos, así como de modificaciones o destrucciones maliciosas o accidentales. Los requisitos pueden especificar:

  - Empleo de técnicas criptográficas
  - Registro de ficheros con “logs” de actividad
  - Asignación de determinadas funcionalidades a determinados módulos
  - Restricciones de comunicación entre determinados módulos
  - Comprobaciones de integridad de información crítica

- **Portabilidad**: Especificación de atributos que debe presentar el software para facilitar su traslado a otras plataformas u entornos. Pueden incluirse:

  - Porcentaje de componentes dependientes del servidor
  - Porcentaje de código dependiente del servidor
  - Uso de un determinado lenguaje por su portabilidad
  - Uso de un determinado compilador o plataforma de desarrollo
  - Uso de un determinado sistema operativo

- **Fiabilidad:** Especificación de los factores de fiabilidad necesaria del sistema. Esto se expresa generalmente como el tiempo entre los incidentes permisibles, o el total de incidentes permitidos.

- **Disponibilidad**: Especificación de los factores de disponibilidad final exigidos al sistema. Normalmente expresados en porcentaje (%) de tiempo en los que el software tiene que mostrar disponibilidad.

Se propone utilizar identificadores únicos para cada uno de los requisitos encontrados con el fin de facilitar su gestión y trazabilidad. Siguiendo las recomendaciones de la IEEE, se propone utilizar la nomenclatura expuesta en la tabla 12.

Tipo de Necesidad	Nomenclatura
Objetivo	OBJ-XXX
Requisito Funcional	RF-XXX
Requisito No Funcional	RNF-XXX
Requisitos de Información	RI-XXX

Tabla 12. *Nomenclatura recomendada por la IEEE*

Fuente: Elaboración propia

3.4.2 Tarea 2: Identificación de conflictos en los requisitos

La identificación de conflictos es la tarea que consiste en buscar las necesidades que al ser implementadas reducen o inhiben el desarrollo de otra(s) necesidades(s).

El objetivo de esta tarea es verificar la consistencia del conjunto de necesidades elicitadas. Dicho conjunto es consistente internamente sí y sólo sí no existen conflictos entre las necesidades que contiene [Durán, 2000].

Los conflictos entre requisitos pueden ser de los siguientes tipos:

- **Conflictos de conducta:** Dos o más necesidades especifican conductas distintas del sistema para las mismas condiciones y el mismo estímulo externo.

- **Conflictos de términos:** Se utilizan términos distintos para referirse al mismo concepto.
- **Conflictos de característica:** Dos o más necesidades especifican aspectos contradictorios para la misma característica del sistema.
- **Conflictos temporales:** Dos o más necesidades exigen características temporales contradictorias al sistema.

Se debe tener en cuenta que la consistencia radica también en la necesidad de un mismo nivel de detalle y de un mismo estilo de redacción y de presentación de las necesidades. Aunque para este caso, la definición formal propuesta cumple con dicha característica a cabalidad.

Evidentemente, si las necesidades son contradictorias, es imposible que pueda existir un sistema que satisfaga todas a la vez, por tanto el ingeniero de requisitos debe estar muy alerta de este tipo de conflictos en las necesidades.

Para la documentación de los conflictos se propone utilizar la plantilla de definición de conflictos sugerida por Amador, en [Durán, 2000], la cual se muestra en la figura 46:

CFL–<*id*>	<*nombre descriptivo*>
Versión	<*nº de la versión actual*> (<*fecha de la versión actual*>)
Autores	• <*autor de la versión actual*> (<*organización del autor*>) …
Fuentes	• <*fuente de la versión actual*> (<*organización de la fuente*>) …
Objs./Reqs. en conflicto	• OBJ-*x* / Rx-y <*nombre del objetivo o requisito en conflicto*> …
Descripción	<*descripción del conflicto*>
Alternativas	• <*descripción alternativa de solución*> (<*autores alternativa*>) …
Solución	<*descripción de la solución adoptada (si se ha acordado)*>
Importancia	<*importancia de la resolución del conflicto*>
Urgencia	<*urgencia de la resolución del conflicto*>
Estado	<*estado del resolución del conflicto*>
Comentarios	<*comentarios adicionales sobre el conflicto*>

Figura 46. *Plantilla para definir conflictos*

Fuente: [Durán, 2000]

A continuación se describen cada uno de los campos de la plantilla:

- **Identificador y nombre descriptivo**: al igual que el resto de la información correspondiente a los requisitos–C, cada conflicto debe poderse identificar de forma única y tener un nombre descriptivo. El prefijo propuesto para lograr una rápida identificación es *CFL*.

- **Versión, Autores, Fuentes**: estos campos tienen el mismo significado que en las plantillas para objetivos y requisitos, aunque referidos al conflicto. En este caso especial, las fuentes son los stakeholders que deben participar en las posibles negociaciones necesarias para su resolución.

- **Objetivos y requisitos en conflicto**: este campo debe contener una lista con los objetivos y/o requisitos afectados por el conflicto.

- **Descripción**: este campo debe contener la descripción del conflicto.

- **Alternativas**: este campo debe contener una lista con las posibles alternativas de solución que se hayan identificado para solucionar el conflicto, así como los autores de dichas alternativas.

- **Solución**: este campo debe contener la descripción de la solución negociada del conflicto, una vez que se haya acordado.

- **Importancia, Urgencia**: estos campos indican, respectivamente, la importancia y la urgencia de la resolución del conflicto.

- **Estado**: este campo indica el estado de resolución del conflicto, que podrá estar *en negociación* o bien, *resuelto*.

- **Comentarios**: este campo tiene el mismo significado que en las plantillas descritas previamente.

3.4.3 Tarea 3: Elaborar matrices de relación

Las matrices de relación propuestas son un conjunto de artefactos que brindarán al ingeniero de requisitos un primer acercamiento a la arquitectura del sistema, mostrándole, desde diferentes puntos de vista, el conjunto de relaciones del sistema y la fuerza de esta relación. De este modo se podrá vislumbrar más fácilmente las necesidades, los actores y los objetivos que son más relevantes.

Con el desarrollo de esta segunda tarea también se podrá obtener un grupo de matrices de trazabilidad, las cuales expondrán los posibles objetos que deben ser modificados al realizar algún tipo de cambio.

La fuerza de la relación se podrá determinar con los pesos asociados a cada una de ellas:

Figura 47. *Notación para los pesos asociados de las relaciones*
Fuente: Elaboración propia

Esto permitirá ver las matrices como un objeto de dos dimensiones, mostrando no sólo la relación entre los objetos, sino también la profundidad de ésta. Cabe aclarar que el peso de la relación debe ser asignado por los clientes y usuarios y no por el ingeniero de requisitos, pues este último puede tener una apreciación totalmente errada de la importancia de cada una de las necesidades.

La propuesta metodológica plantea un conjunto de matrices que se deben realizar como mínimo, aunque no son camisa de fuerza para el ingeniero de requisitos, quien puede definir nuevas matrices o eliminar algunas, si lo considera necesario.

En esta tarea el ingeniero de requisitos deberá elaborar las matrices con la información adquirida hasta el momento y deberá marcar las relaciones sin asignarles ningún peso, dado que, como se dijo anteriormente, la importancia debe ser indicada por los clientes y usuarios.

La asignación de los pesos se realizará posteriormente durante la tarea 5 “Ejecutar el ciclo Análisis – Elicitación”. Sin embargo, a continuación se describe la importancia de cada una de las matrices propuestas cuando ya tienen los pesos de las relaciones asignados.

3.4.3.1 Matriz: Actor x Requisitos-C Funcionales

En esta matriz están representadas las funciones y operaciones en la que cada uno de los actores participa. Se muestra el conjunto de necesidades (Requisitos-C) escrito de forma coherente, concisa y clara, con el fin de crear una representación de todas las funcionalidades que el sistema debe proporcionar para cada actor.

Como se dijo anteriormente, este tipo de matrices permite al ingeniero de requisitos visualizar la fuerza de la relación. Por esta razón, se considera que esta representación es más completa que un diagrama de relación de contexto, puesto que en este último, al sólo poder ver la relación del actor con la funcionalidad, no se podría tomar alguna decisión de priorización de alguna de ellas.

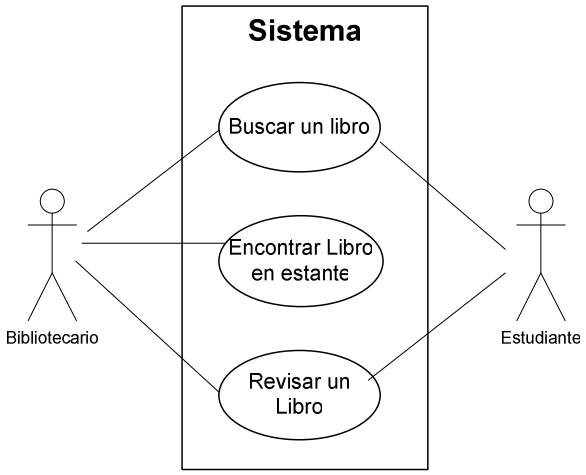


Figura 48. *Ejemplo de un diagrama de contexto*

Fuente: Elaboración propia

En la figura 48 se ve que el actor bibliotecario cuenta con muchas relaciones funcionales, pero no se puede conocer cuál de ellas se podría aplazar, en caso de que fuera necesario. Por el contrario, asignar un peso a la relación en las matrices, permite conocer cuál de las funcionalidades en cierto momento podría ser aplazada. Además, las matrices proporcionan una calificación objetiva por fila y columna, facilitando identificar cuál conjunto o funcionalidad es la más relevante y cuál actor es el más representativo.

Actores	Buscar un libro	Encontrar un libro en un estante	Revisar un libro	**Peso de Fila**
Bibliotecario	○	◉	◉	21
Estudiante	◉		◉	18
Peso	**12**	**9**	**18**	

Tabla 13. *Ejemplo de una matriz de relación Actores x Requisitos-C*

Fuente: Elaboración propia

En este ejemplo se puede ver que el actor "*bibliotecario"* es el más relevante, puesto que cuenta con un número mayor de peso (21) y por tanto una relación funcional muy fuerte. Además, ilustra que la funcionalidad "*revisar un libro*" es la

más utilizada por los actores del sistema y que la funcionalidad "*Encontrar un libro en un estante*" podría ser aplazada en algún momento, dado que no es de mucho impacto. Partiendo de lo mencionado anteriormente, el ingeniero de requisitos y el equipo de desarrollo deben saber que los actores y las funcionalidades más relevantes deben estar muy bien definidos y los cambios realizados a estos generarán un mayor impacto en el sistema.

3.4.3.2 Matriz: Requisitos-C Funcionales x Objetivos

Los objetivos y sus necesidades asociadas también deben ser observados según su peso de relación, puesto que cada una de las necesidades elicitadas puede contar con una fuerza de relación diferente dentro del objetivo en el cual se encuentra y esto permitiría conocer cuáles son las necesidades más relevantes a desarrollar y a cuáles se les podría dar espera.

En el diagrama de afinidad y en el diagrama de árbol desarrollados en el proceso de elicitación, se pueden observar las relaciones de las necesidades dentro de los objetivos, pero de forma similar a la relación en el diagrama contextual, no se puede discernir sobre cuál de las necesidades es la más o la menos relevante, por tal razón se cuenta con esta matriz para mostrar, de una forma más adecuada, la relación interna de las necesidades en el objetivo.

En la figura 49 se puede observar este tipo de relación:

Objetivos	Ingresar Alumno	Modificar Alumno	Ingresar Usuario	Consultar Alumno	Peso de Fila
Gestionar Alumnos	◉	◉		○	21
Administrar Usuarios	○		◉		12
Peso	12	9	9	3	

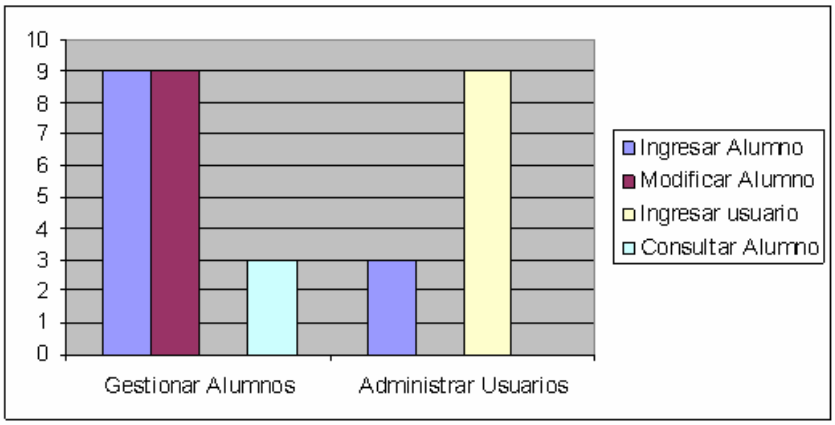


Figura 49. *Ejemplo de una matriz de relación Objetivo x Requisito-C*
Fuente: Elaboración propia

Las necesidades "*Ingresar Alumno*" y "*Modificar Alumno*" son de mucho peso dentro del objetivo "*Gestionar Alumno*", por tanto se deben tener muy bien definidas y en cuenta para los cambios, a diferencia de la necesidad "*Consultar Alumno*", que para este caso, no es tan relevante.

3.4.3.3 Matriz: Requisitos-C Funcionales x Requisitos de información

La matriz de Requisito-C Funcional x Requisito de información muestra la relación que existe entre la necesidad elicitada y la información asociada con dicha necesidad. Además, en esta matriz se podrá ver también cuáles necesidades utilizan la información de las otras, asegurando así la trazabilidad de la información y por tanto el control que se le debe adicionar en el desarrollo.

La información es un insumo muy importante en las empresas y por tanto su manejo dentro del sistema debe ser de mucho cuidado. Partiendo de esta

premisa, la utilización de la matriz de relación Requisito-C x Requisitos de Información, le permitirá conocer al grupo de desarrollo cómo se deben manejar los datos dentro del sistema y la forma como se conectan con las diferentes funcionalidades.

Sobra especificar que esta matriz cuenta también con relaciones de peso y por tanto habrán algunas más relevantes que otras.

3.4.3.4 Matriz: Requisitos-C No Funcionales x Requisitos-C Funcionales

Como es conocido, los requisitos no funcionales o de calidad deben ser tenidos en cuenta para el desarrollo de cualquier sistema de software. Dado que en la propuesta metodológica también fueron elicitados un conjunto de necesidades que no van necesariamente ligadas a la funcionalidad, sino que especificaban factores de calidad con los cuales debe contar el sistema, serán adicionadas a las matrices estas necesidades con el fin de obtener una mejor visión de dichos factores y su relación con cada una de las funcionalidades del sistema. Esto permitirá al equipo de desarrollo tenerlos en cuenta en todo momento, para cumplir con ellos y, así satisfacer al cliente.

Existen necesidades no funcionales que son transversales a todo el sistema, como la utilización de recursos, pero también existen algunas que son específicas para ciertas necesidades funcionales y que no afectan de forma directa a otras, por tanto esta matriz brindará la facilidad de no generar desarrollos innecesarios en módulos que en verdad no necesiten esta característica no funcional.

Para poder ilustrar este tipo de relaciones, se puede observar la tabla 14:

Requisito-C No funcional	Ingresar Alumno	Realizar cierre de cartera	Generar reportes de pagos	Peso de Fila
Debe ser rápido	△	◉	◉	19
No debe consumir muchos recursos	◉	◉	◉	27
Peso	**10**	**18**	**18**	

Tabla 14. *Ejemplo de un matriz de relación Requisitos no Funcionales x Requisitos Funcionales*
Fuente: Elaboración propia

3.4.4 Tarea 4: Elaborar los cuestionarios de Kano

Con el desarrollo de esta tarea se busca realizar el conjunto completo de cuestionarios de Kano para poder priorizar cada uno de las necesidades elicitadas previamente.

3.4.4.1 Cuestionario de Atribución de Importancia

Como fue explicado en la sección 2.2.2 "*Cuestionarios*", Noriaki Kano propuso con su método un conjunto de preguntas para realizarle a los clientes y usuarios, con el fin de poder clasificar y priorizar las necesidades elicitadas. En la propuesta metodológica, no serán utilizados todos los cuestionarios brindados en el modelo, puesto que no es necesario clasificar las necesidades en atractivas, obligatorias y unidimensionales. Por esta razón, sólo se realizará el cuestionario de atribución de importancia, con el fin de tomar la verdadera prioridad que le dan los clientes a las necesidades.

Se propone analizar las necesidades que se puedan convertir en actividades de inserción, modificación, eliminación y consulta (CRUD), con el fin de identificar si deben ser, o no, ingresados en el cuestionario, puesto que dichas funcionalidades son de obvia inclusión en el software y por ende, podrían aumentar el tiempo necesario para responder el cuestionario. Sin embargo, se debe considerar que existen funciones CRUD que tienen alto grado de complejidad y que por esta

razón deben ser ingresadas en el cuestionario. Se deja a criterio del ingeniero de requisitos la inclusión de necesidades de tipo crear, modificar, eliminar y consultar en cuestionario.

Figura 50. *Ejemplo de una función CRUD que no debe ser eliminada*
Fuente: Elaboración propia

En la figura 50 se puede observar cómo la necesidad "*matricular un alumno en el Colegio*", aun siendo un requisito que se puede convertir en una función de inserción de un alumno en la base de datos, no debe ser eliminada por el ingeniero de requisitos, dado que tiene mucho impacto en otras necesidades.

Cabe resaltar que los requisitos funcionales que sean eliminados del cuestionario de Kano no deben ser ingresados en los Qué(s), al momento de construir la Casa de la Calidad, dado que en ella sólo deben ser incluidos los requisitos más relevantes para la satisfacción de los clientes y usuarios; además su adición aumentaría el tamaño y la complejidad de la casa, sin ningún sentido.

Para obtener la priorización, se propone utilizar un cuestionario con la estructura mostrada en la tabla 15, en el cual se les solicita a los clientes y usuarios que asignen un valor de importancia para cada uno de los requisitos.

Cuestionario de Atribución de Importancia En cada pregunta, marque con un círculo el número de la escala que mejor refleje su opinión									
	Para nada Importante		Algo Importante		Importante		Muy Importante		Extremo Importante
	←								→
Requerimiento 1	1	2	3	4	5	6	7	8	9
Requerimiento 2	1	2	3	4	5	6	7	8	9
.....	1	2	3	4	5	6	7	8	9
Requerimiento X	1	2	3	4	5	6	7	8	9

Tabla 15. *Concentración de respuestas*

Fuente: [León, 2005]

Después de realizadas el conjunto de preguntas, queda como siguiente tarea entregárselas a el cliente y los usuarios con el fin de obtener las respuestas correspondientes y, de este modo, continuar con el análisis de los datos.

3.4.5 Tarea 5: Ejecutar el ciclo Análisis – Elicitación

El ingeniero de requisitos, después de terminar el diseño y completar el contenido de los cuestionarios de Kano, debe realizar una nueva visita a los clientes y usuarios con el fin de cumplir tres objetivos específicos:

- Resolver los conflictos encontrados hasta el momento.
- Entregar a los clientes y usuarios los cuestionarios de Kano desarrollados en la tarea anterior para que estos los diligencien.
- Completar las matrices de relación elaboradas en la tarea 3 con ayuda de los clientes y usuarios, quienes deben determinar la fuerza de las relaciones entre las necesidades, los objetivos y los actores.

Este ciclo es una forma de retroalimentar el análisis desarrollado hasta el momento, pues con la información obtenida durante las sesiones de elicitación / negociación de requisitos (explicadas en la sección 3.3.5), el ingeniero de requisitos puede obtener un mayor entendimiento de lo que realmente necesitan los clientes y usuarios.

Al finalizar esta actividad se deberán obtener los siguientes resultados:

- Los conflictos identificados en la tarea 2 deben quedar resueltos completamente.
- Las matrices elaboradas en la tarea 3 deberán tener la fuerza asociada con cada una de las relaciones identificadas.
- Los cuestionarios de Kano deben estar completamente diligenciados, pues con esta información se obtendrá la prioridad asignada por los clientes y usuarios a cada requisito.

3.4.6 Tarea 6: Tabulación de los resultados obtenidos en los cuestionarios de Kano

Una vez se reciban los cuestionarios diligenciados, se deberá calcular, para cada uno de los requisitos, el promedio de todos los valores asignados por los clientes y usuarios en la escala de atribución de importancia.

Se propone agrupar los resultados en una tabla con la estructura que se muestra a continuación.

Requisito	Prioridad
Requisito 1	X_1
Requisito 2	X_2
. . .	. . .
Requisito n	X_n

Tabla 16. *Tabulación de resultados para el Modelo de Kano*
Fuente: Elaboración propia

La prioridad asignada a cada uno de los requisitos será utilizada posteriormente en la construcción de la Casa de la Calidad.

RD SP 3.3 - Analizar los requisitos
A través de la aplicación de los cuestionarios de Kano y el análisis de los requisitos por medio de las matrices propuestas, el equipo estará cumpliendo con la práctica "Analizar los requisitos" de la tercera meta del área de proceso RD en el modelo CMMI, pudiendo identificar cuáles son los requisitos realmente importantes, necesarios y suficientes para darle cumplimiento al proyecto.

3.4.7 Tarea 7: Construir la Casa de la Calidad

Como es sabido, la Casa de la Calidad es una matriz que traduce los requisitos del cliente en características técnicas necesarias que permitan satisfacerlas, plasmadas en el desarrollo de un nuevo producto. Para el caso particular de la metodología, esta traducción se realizará directamente de las necesidades elicitadas, después de un análisis previo, a casos de uso.

La principal motivación para convertir las necesidades funcionales a casos de uso, es que el modelado orientado a objetos es una de las metodologías más utilizadas en todo el mundo para realizar el análisis y diseño de productos de software. Esto podría generar que la propuesta metodológica tuviera una mayor acogida en la comunidad de software, puesto que los casos de uso son una herramienta con la

cual la gran mayoría de los clientes y usuarios ya se encuentran familiarizados y, adicionalmente, las organizaciones utilizan para dirigir todo su proceso de desarrollo.

En primera instancia se consideró ingresar el conjunto completo de necesidades en los Qué(s) de la Casa de la Calidad, dado que se obtendría una trazabilidad importante de todas las necesidades elicitadas. Sin embargo, una de las preocupaciones al momento de construir la casa es, precisamente, su tamaño, puesto que el grado de complejidad aumenta según el número de entradas (Qué) de la casa. Esto se debe a que para cada entrada debe existir por lo menos una característica técnica (Cómo) que lo satisfaga.

Observando la problemática enunciada anteriormente, se decidió reducir el número de necesidades que serían incluidas en la Casa de la Calidad. Los requisitos de información fueron los primeros en ser descartados, pues se asume que con las plantillas elaboradas en la tarea 3 de la elicitación de requisitos y las matrices de relación creadas en la tarea 3 del análisis, se tiene información suficiente para ser desarrollados posteriormente y mantener su trazabilidad.

En segunda instancia, se redujo el conjunto de entradas de la casa al momento de elaborar los cuestionarios de Kano, donde se enunciaron los criterios para identificar el grado de importancia de las necesidades relacionadas con funcionalidades CRUD y eliminar aquellas que no tuvieran mucha relevancia en la elaboración de los cuestionarios.

Después de realizar la tabulación de las respuestas obtenidas de los clientes y usuarios, se propone excluir los requisitos funcionales con prioridad menor que 4 pues se puede asumir que para los clientes y usuarios no son tan importantes.

Por el contrario, todos los requisitos no funcionales deberán ser incluidos en la Casa de la Calidad, dado que éstos representan factores de calidad del producto

software, los cuales deben ser implementados para lograr una mayor satisfacción de los clientes y usuarios.

A continuación se explican detalladamente los pasos necesarios para construir la Casa de la Calidad.

3.4.7.1 Paso 1: Qué(s) – La voz del cliente

Las entradas de este componente de la casa se deberán organizar de forma similar a la estructura del diagrama de afinidad desarrollado en la tarea 3 del proceso de elicitación, en el cual un objetivo general se desglosa en necesidades específicas, generando dos niveles de profundidad. Además, los Qué(s) se dividirán en dos secciones: las necesidades funcionales y las no funcionales.

Se propone agrupar los requisitos no funcionales según su característica, es decir, cada requisito debe ser catalogado en uno de los siguientes grupos: rendimiento, seguridad, fiabilidad, disponibilidad, mantenibilidad, portabilidad y usabilidad.

La estructura de este componente se ilustra en la tabla 17:

QUÉ(S)	
Requisitos-C Funcionales	
Objetivo 1	Necesidad 1.1
	Necesidad 1.2
Objetivo 2	Necesidad 2.1
	Necesidad 2.2
	Necesidad 2.3
Requisitos-C No Funcionales	
Usabilidad	Necesidad 1
	Necesidad 2
Seguridad	Necesidad 3

Tabla 17. *Estructura del componente Qué(s) de la Casa de la Calidad*

Fuente: Elaboración propia

Es muy importante tener en cuenta que la información de las necesidades debe ser escrita tal y como fueron expresadas por los clientes y usuarios durante las sesiones de elicitación y que está contenida en los diagramas de afinidad; es decir, no se debe intentar traducir *"La voz del cliente"* a atributos técnicos, pues en esta traducción se podría perder información muy importante para la satisfacción del cliente.

3.4.7.2 Paso 2: Prioridades – Importancia para los clientes

A cada una de las necesidades incluidas en el paso anterior se les debe asignar una prioridad, la cual debe estar en una escala de 4 a 9, donde 4 representa el nivel de prioridad más bajo. Esta escala no toma el intervalo de 1 a 4, dado que los requisitos con prioridad menor a 4 fueron excluidos de la construcción de la casa, como se dijo anteriormente.

El nivel de importancia de las necesidades se adquiere del resultado obtenido en la tarea 6 "*Tabulación de los resultados obtenidos en los cuestionarios de Kano*". Estos resultados se pasan directamente a una columna que se encuentra contigua al componente Qué, como se muestra a continuación.

QUÉ(S)		Prioridad
Requisitos-C Funcionales		
Objetivo 1	Necesidad 1.1	X_1
	Necesidad 1.2	X_2
Objetivo 2	Necesidad 2.1	X_3
	Necesidad 2.2	X_4
	Necesidad 2.3	X_5
Requisitos-C No Funcionales		
Usabilidad	Necesidad 1	X_6
	Necesidad 2	X_7
Seguridad	Necesidad 3	X_8

Tabla 18. *Estructura del componente de prioridad de la Casa de la Calidad*

Fuente: Elaboración propia

3.4.7.3 Paso 3: Cómo(s) – Requisitos Técnicos

El desarrollo de los Cómo(s), está ligado, en gran parte, al método de definición de casos de uso propuesta por Ivar Jacobson, un importante contribuyente al desarrollo de los modelos UML y el Proceso Unificado de desarrollo de software (RUP- Rational Unified Process).

Los casos de uso son requisitos funcionales que describen de una manera detallada el comportamiento del sistema con los distintos actores que interactúan con él. No definen todos los requisitos (por ej. los tipos de datos, interfaces externas, niveles de rendimiento esperado, etc.), pero representan el hilo conductor que vincula a todos los requisitos posibles (actuales y futuros) de una aplicación.

Al momento de comenzar el desarrollo del componente Cómo(s) de la Casa de la Calidad, el ingeniero de requisitos tiene como primera tarea identificar los posibles casos de uso en los que se puede convertir cada una de las necesidades elicitadas. Dado que las necesidades todavía se encuentran expresadas de forma general y en lenguaje natural, representando explícitamente “la voz del cliente”, el ingeniero de requisitos debe realizar una abstracción mayor de las necesidades contenidas en los Qué(s) con ayuda del resto de la información elicitada durante las entrevistas y las sesiones de lluvia de ideas, con el fin de obtener una especificación formal de los casos de uso que vayan surgiendo de los Requisitos-C.

Cada uno de los casos de uso identificados (Cómo) debe satisfacer y tener relación con uno o varios Qué(s); si el ingeniero de requisitos encuentra un caso de uso que no cumpla con alguna de estas dos condiciones, debe ser eliminado, puesto que no tendría ninguna utilidad en el sistema. Lo anterior permitirá tener siempre un conjunto de casos de uso refinados y concretos del sistema.

Para cada caso de uso se debe realizar una especificación formal, en la cual se describe de forma general “la secuencia de acciones que el sistema deberá llevar a cabo y que produce un resultado observable para un actor” [Jacobson et al., 2000]. En la tabla 19 se describen los componentes de la especificación de los casos de uso:

CU-<número>	<Nombre descriptivo>
Autor:	<Autor>
Fecha:	<Fecha de realización>
Descripción: Breve descripción de cómo se debe comportar el sistema en este caso de uso y su evento de activación	
Actores: Actores que utilizan este caso de uso (pueden ser tomados de las matrices de relaciones).	
Precondiciones: Se expresan en lenguaje natural las condiciones necesarias para que se pueda realizar el caso de uso.	
Flujo Normal: Este campo contiene la secuencia normal de interacciones del caso de uso. En cada paso, un actor o el sistema realiza una o más acciones, o se realiza otro caso de uso. Se asume que, después de realizar el último paso, el caso de uso termina.	
Flujo Alternativo: Describen caminos alternativos que pueden ser ejecutados por el actor o el sistema, generalmente están dados por una condición de excepción.	
Poscondiciones: En este campo se expresan en lenguaje natural las condiciones que se deben cumplir después de la terminación normal del caso de uso.	

Tabla 19. *Plantilla para la especificación de los casos de us*
Fuente: [Wikipedia, Casos de Uso]

Como se puede apreciar en la plantilla anterior, el primer campo representa el identificador del caso de uso, el cual está compuesto por las letras CU, que son la abreviación de las palabras Caso de Uso, y un número único que lo diferencia de los demás. Este identificador debe ser incluido en la definición del Cómo, con el fin de tener una trazabilidad entre los Cómo(s) y los casos de usos especificados.

Aunque algunos requisitos funcionales fueron excluidos para la construcción de la Casa de la Calidad con el fin de reducir su complejidad, se propone darles el mismo trato que a aquellos requisitos que sí fueron incluidos. Es decir, se deben traducir las necesidades a casos de uso y, posteriormente, realizar su especificación. La principal motivación para describir completamente el funcionamiento del sistema por medio de casos de uso, es poder realizar de una manera más óptima la validación de los requisitos con el uso de prototipos y walkthroughs.

El valor verdadero de un caso de uso, según [Wikipedia, Casos de Uso], reposa en dos áreas:

- La descripción escrita del comportamiento del sistema al afrontar una tarea de negocio o un requisito de negocio.
- La posición o contexto del caso de uso entre otros casos de uso, manifestada por medio de los diagramas. Éste es un mecanismo que promueve una imagen fácil del comportamiento del sistema, un entendimiento común entre el cliente/propietario/usuario y el equipo de desarrollo.

Teniendo en cuenta estas premisas sobre los casos de uso, se propone que, además de realizar una especificación, se elabore un diagrama de casos de uso con el fin de mostrar cómo se relacionan los casos de uso entre sí y con los actores.

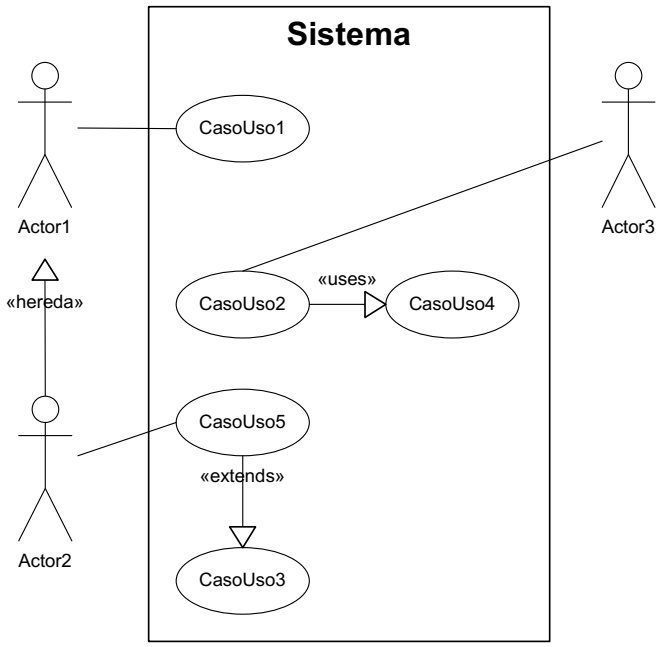


Figura 51. *Ejemplo de diagrama de casos de uso*
Fuente: Elaboración propia

Para mayor información sobre casos de uso y el lenguaje unificado de modelado (UML), se puede consultar [Jacobson et al., 2000].

Como se dijo anteriormente, los casos de uso representan únicamente los requisitos funcionales. Por esta razón, para las necesidades no funcionales, el ingeniero de requisitos, con la colaboración del equipo desarrollador, deberá identificar diferentes estrategias a través de las cuales se puedan satisfacer dichas necesidades. En los *Cómo* se deberá incluir una frase concreta y clara que resuma cada estrategia y de esta forma poder relacionar las necesidades no funcionales con una o más especificaciones técnicas que la satisfagan.

De forma similar a los Qué(s), se propone que la estructura del componente Cómo(s) de la Casa de la Calidad, se divida en dos secciones diferentes: una para los casos de uso y otra en la que se describan las estrategias necesarias para cumplir los requisitos no funcionales.

La estructura de este componente se ilustra en la tabla 20:

CÓMO(S)	**Casos de Uso**	CU-001 Caso de uso 1	CU-002 Caso de uso 2	CU-003 Caso de uso 3	**Requisitos No Funcionales**	Estrategia 1	Estrategia 2	Estrategia 3

Tabla 20. *Estructura del componente Cómo(s) de la Casa de la Calidad*

Fuente: Elaboración propia

Hasta el momento se han construido dos componentes de la Casa de la Calidad, como puede apreciarse en la figura 52.

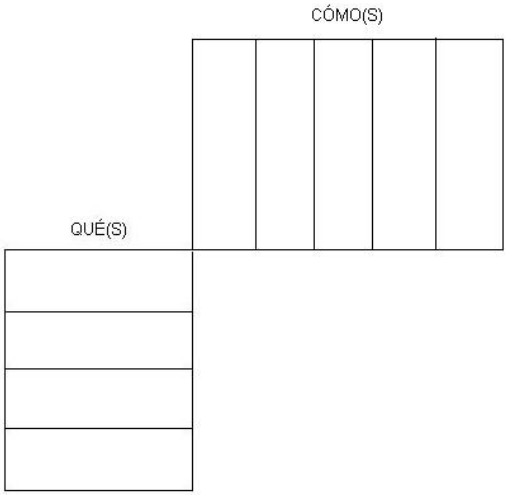


Figura 52. *Qué(s) vs. Cómo(s)*

Fuente: Elaboración propia

RD SP 2.1 - Establecer requisitos de producto y de componente
En la Casa de la Calidad el componente Qué(s) permite expresar las necesidades en los términos de los clientes y usuarios, sin utilizar expresiones técnicas. En el componente Cómo(s) se hace la traducción de requisitos de clientes a requisitos de producto y además, se mantiene la relación entre los requisitos. Por estas razones, con la construcción de la casa se asegura el cumplimiento de la práctica específica "Establecer requisitos de producto y de componente" del área de proceso Desarrollo de Requisitos en el modelo CMMI.

RD SP 3.1 - Establecer conceptos y escenarios operacionales
Los casos de uso que se definen en los Cómo(s) permiten ver la secuencia de eventos que pueden ocurrir al usar el producto. De forma similar, las estrategias de diseño, incluidas en este componente de la casa, definen el ambiente en el cual el producto operará. Con estas actividades se puede asegurar el cumplimiento de la práctica específica "Establecer conceptos y escenarios operacionales" del área de proceso Desarrollo de Requisitos.

3.4.7.4 Paso 4: Matriz de Relaciones – Relaciones entre Qué(s) y Cómo(s)

Después de establecer los Qué(s) y los Cómo(s), la construcción de la Casa de la Calidad continúa con el establecimiento de las relaciones entre la voz del cliente (Qué) y los requisitos técnicos (Cómo). Este paso debe ser realizado con la colaboración de todo el equipo de desarrollo, con el fin de discutir la contribución de cada característica técnica a las necesidades.

En la matriz de relaciones se debe determina el nivel de alcance o relevancia que tiene cada caso de uso o estrategia en función de una necesidad. Estas relaciones pueden ser categorizadas como **fuertes, medianas y débiles**. Para representar esta categorización se propone utilizar los símbolos mostrados en la figura 53; sin embargo, el ingeniero de requisitos puede optar por utilizar otros símbolos o

categorías (1, 3 y 9) para indicar la fuerza de la relación. Si una celda es dejada en blanco quiere decir que no existe ninguna relación.

Figura 53. *Tipos de Relaciones*
Fuente: Elaboración propia

Cuando una fila de los Qué(s) tiene todas las celdas en blanco significa que una necesidad no ha sido direccionada a ningún Cómo y que por lo tanto es necesario identificar un requisito técnico que la satisfaga.

Igualmente, cuando una columna de los Cómo tiene todas sus celdas en blanco, significa que una característica que ha sido adicionada, no satisface ninguna necesidad de los clientes y usuarios y que por lo tanto debe ser eliminada. De forma similar, cuando una columna contiene sólo enlaces débiles, puede ser que la característica técnica no sea relevante para cumplir con la satisfacción del cliente; sin embargo, se debe tener muy en cuenta que al cambiar esta característica puede producir un efecto adverso en otra que sí sea muy relevante para la construcción del producto. Por esta razón, se aconseja verificar las relaciones existentes entre las características técnicas en la matriz de correlación, que será explicada en el siguiente paso de la construcción de la Casa de la Calidad.

También existe la posibilidad de que el ingeniero de requisitos identifique ciertos casos de uso que cuenten con un gran número de relaciones. Cuando esto sucede se debe pensar en fragmentar dicho caso de uso, con el fin de quitarle complejidad al momento de su implementación y, con esto, obtener una arquitectura más modular y fácil de mantener.

En la figura 54 se muestra la estructura de la matriz de relación.

			Casos de Uso	CU-001 Caso de uso 1	CU-002 Caso de uso 2	CU-003 Caso de uso 3	Requisitos No Funcionales	Estrategia 1	Estrategia 2	Estrategia 3
Requisitos-C Funcionales		**Prioridad**								
Objetivo 1	Necesidad 1.1	X_1			⊙	○				△
	Necesidad 1.2	X_2		⊙	△					
Requisitos-C No Funcionales										
Usabilidad	Necesidad 1	X_3		△						○
	Necesidad 2	X_4				○		⊙		
Seguridad	Necesidad 3	X_5						△	⊙	

Figura 54. *Estructura de la Matriz de Relaciones de la Casa de la Calidad*

Fuente: Elaboración propia

3.4.7.5 Paso 5: Matriz de Correlación – El techo de la Casa

La matriz de correlación se establece al comparar cada Cómo con los demás, con el fin de detectar posibles conflictos y su influencia sobre las demás características técnicas.

Como se dijo anteriormente, en la sección 2.1.4 *"La Casa de la Calidad"*, el techo contiene la información más crítica para la trazabilidad de los requisitos, pues en él se refleja la correlación entre las características técnicas, es decir, permite conocer el efecto que un cambio o mejora de un requisito, tiene sobre los demás.

El grado y dirección de los efectos identificados tienen gran influencia sobre los esfuerzos de desarrollo, especialmente cuando existen relaciones negativas de un Cómo sobre otros, pues es posible que se genere un cuello de botella en el diseño

del producto. Por esta razón, el equipo desarrollador deberá tener especial cuidado con este tipo de características, pues el aumento de una, representa directamente la disminución de otra que puede tener igual importancia para los clientes y usuarios, por tanto se debe hacer una planeación adecuada para cumplir con ambas características sin afectar la satisfacción del cliente. En otras palabras, la matriz de correlación mostrará qué funciones técnicas necesitan mayor comunicación y colaboración.

Esta matriz puede convertirse en una herramienta muy eficaz para la gestión de los cambios de los requisitos, dado que cuando cambia una necesidad (Qué), deberán cambiar una o más características técnicas (Cómo) en la casa. Cuando esto sucede, la matriz le permitirá al ingeniero de requisitos identificar qué características técnicas se verán afectadas por los cambios realizados en los Cómo(s).

Otra característica muy importante de la matriz de correlación surge cuando, por ejemplo, una característica del producto, que parece ser irrelevante para la satisfacción del cliente por tener relaciones débiles en la matriz de relaciones, puede generar efectos adversos en otras características que son muy relevantes para la satisfacción de los clientes y usuarios cuando hay un cambio en ella.

La matriz de correlación será dividida en tres secciones diferentes como se muestra en la figura 55. La principal motivación para hacer esta división es que no todas las características técnicas se relacionan de la misma manera.

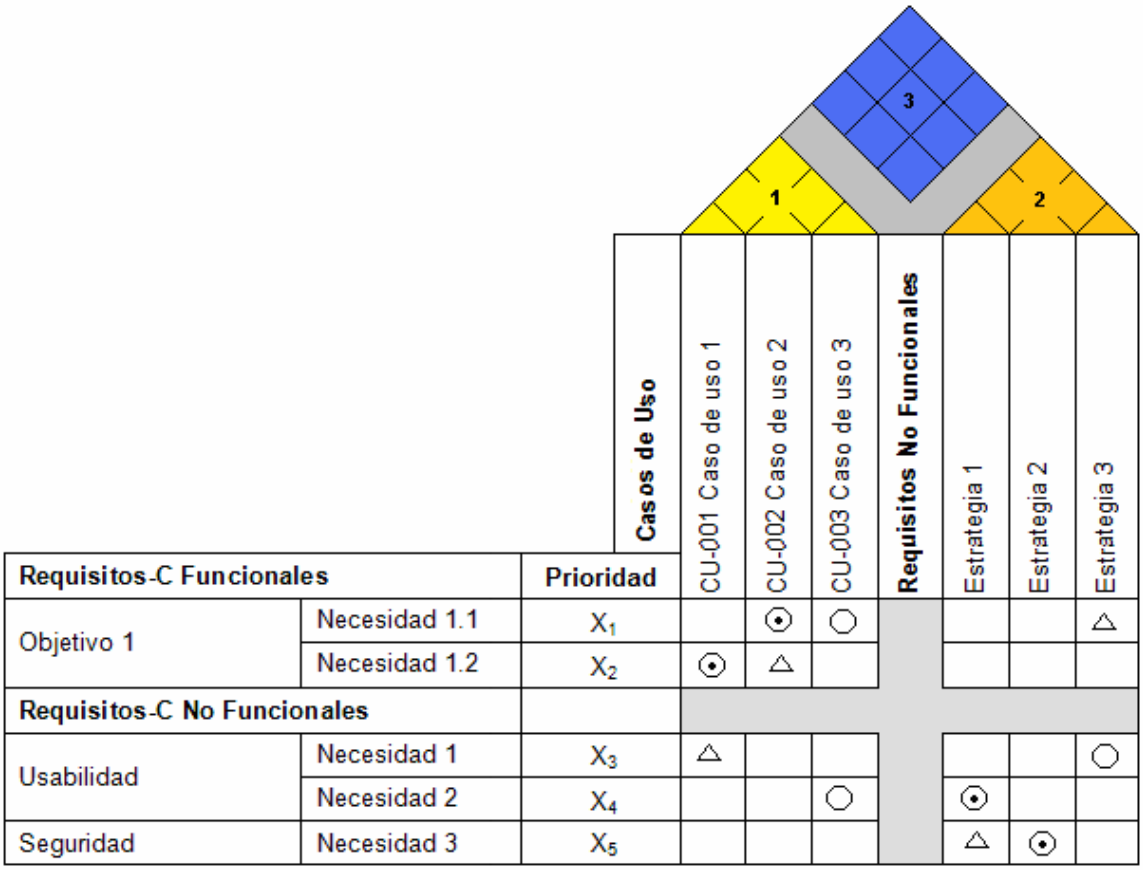


Figura 55. *Estructura de la matriz de correlación en la Casa de la Calidad*

Fuente: Elaboración propia

En la sección 1 de dicha figura se identifican las relaciones existentes entre los casos de uso, dado que un caso de uso no afecta positiva ni negativamente a otro; esta sección sirve para la trazabilidad de los requisitos funcionales, pues a través de la matriz se puede determinar qué casos de uso deben ser revisados cuando se realiza un cambio en otro requisito. Estas relaciones de representarán con el símbolo: ✓

La sección 2 de la misma figura corresponde a las relaciones entre las estrategias planteadas para cumplir con los requisitos no funcionales. Las correlaciones se representan por “O” para indicar que existe una correlación positiva y “X” para indicar que existe una correlación negativa.

Con el siguiente ejemplo se pretende mostrar este tipo de relaciones (positivas y negativas) entre las característica no funcionales de un producto de software.

“Cuando se dirigen lo esfuerzos para maximizar la eficiencia de una sistema de software, es probable que tenga una relación negativa con factores como la mantenibilidad. Por otro lado, cuando se dirigen los esfuerzos a cumplir la conformidad, es probable que se tenga un efecto positivo en factores como la usabilidad y la facilidad de prueba. Sin embargo, es muy probable que no se tengan relaciones explícitas entre la usabilidad y la portabilidad” [Karlsson, 1996].

Los conflictos presentados en el ejemplo anterior, pueden ser resueltos con mayor facilidad con ayuda de la columna de prioridad de la Casa de la Calidad, en la cual se determina la importancia asignada por los clientes y usuarios a cada característica no funcional.

En la tercera y ultima sección de la figura 55, se deberán determinar las relaciones entre los requisitos no funcionales y los funcionales, con ayuda de la matriz de relación “*Requisito-C No Funcional x Requisitos-C Funcionales*” realizada en la tarea 3.

En esta sección de la matriz de correlación se podrán utilizar los 3 tipos de símbolos que fueron definidos para las otras secciones, puesto que al mezclar los requisitos funcionales con los no funcionales se pueden presentar ambas situaciones, es decir, relaciones de trazabilidad o de efecto (positivo o negativo).

Un ejemplo de una relación de trazabilidad se presentaría cuando una estrategia de usabilidad - Facilidad de aprendizaje está ligada con un caso de uso como “solicitar ayuda”.

El segundo tipo de relación puede verse cuando los clientes y usuarios solicitan la generación de reportes en los cuales es necesario acceder a mucha información y al mismo tiempo se solicita prontitud en la respuesta. Como puede verse, entre

estos dos requisitos existe una relación negativa, dado que acceder a grandes volúmenes de información aumenta considerablemente el tiempo de respuesta.

La correlación positiva se presenta cuando se crean casos de uso para cumplir con un requisito no funcional, o se crean estrategias para cumplir con características no funcionales inmersas en los requisitos funcionales. Un ejemplo del primer tipo de relación se puede ver cuando se crea el caso de uso "Validar usuario" para cumplir una característica no funcionales de seguridad; un ejemplo del segundo tipo de relación podría ser incluir entre las estrategias de facilidad de uso la utilización de formatos similares a los de la organización, para los casos de uso que realicen el ingreso de la información al sistema.

***RD SP 2.2* - Asignar los requisitos para cada componente de producto**
La construcción de la Casa de la Calidad se relaciona con la práctica específica "Asignar los requisitos para cada componente de producto" de la segunda meta del área de proceso RD, en el sentido en que las relaciones de todos los requisitos están definidas y documentadas claramente; además, los requisitos funcionales cuentan con asignaciones a casos de uso y los requisitos no funcionales a estrategias de diseño, las cuales facilitan la elaboración del producto.

3.4.7.6 Paso 6: El cuánto – Ponderación de características técnicas

El último paso en la realización de la casa es determinar la ponderación de cada característica técnica propuesta, con el fin de obtener la información necesaria para dirigir las actividades siguientes de diseño o desarrollo; es decir, se determina la fuerza que tiene cada una de las características, con el fin de enfocar los esfuerzos en desarrollar dichas características y de esta manera satisfacer al cliente.

Para cumplir con el objetivo descrito anteriormente, se crean 3 componentes nuevos en la Casa de la Calidad:

- Una columna en la cual se especifica el peso de cada necesidad, según las relaciones que tenga.
- De forma similar, se crea una fila en la cual se especifica el peso de cada una de las características técnicas.
- Una fila que contiene el grado de importancia total de cada característica técnica, teniendo en cuenta la prioridad asignada por los clientes y usuarios a cada una de las necesidades y sus respectivas relaciones.

Requisitos-C Funcionales		Prioridad	CU-001 Caso de uso 1	CU-002 Caso de uso 2	CU-003 Caso de uso 3	Requisitos No Funcionales	Estrategia 1	Estrategia 2	Estrategia 3	Peso de la fila
Objetivo 1	Necesidad 1.1	X_1		⊙	◯				△	13
	Necesidad 1.2	X_2	⊙	△						10
Requisitos-C No Funcionales										
Usabilidad	Necesidad 1	X_3	△						◯	4
	Necesidad 2	X_4			◯		⊙			12
Seguridad	Necesidad 3	X_5					△	⊙		10
	Peso de la columna		10	10	6		10	9	4	
	Importancia		Y_1	Y_2	Y_3		Y_4	Y_5	Y_6	

Figura 56. *Casa de la Calidad con todos sus componentes*

Fuente: Elaboración propia

El resultado del peso de cada una de las filas o las columnas es, simplemente, la sumatoria de los valores de las relaciones encontradas en cada una de ellas. Como se ilustra en la figura 56, el peso de la primera fila es 13; este resultado se obtiene al sumar los valores equivalentes a los símbolos de la relación (9 + 3 + 1). De forma similar, el resultado en la columna 1 es de 10 (9 + 1).

Para poder calcular la importancia de cada una de las características técnicas, se debe multiplicar la prioridad asignada por los clientes y usuarios con el total del peso de la fila. De este modo se obtiene un valor ponderado de la relación, y la sumatoria de estos valores da como resultado la importancia de cada una de las características. A continuación se expone la fórmula para obtener la importancia de una característica técnica.

$$\sum_{i=1}^{n} X_i _ P_i$$

donde P_i es el peso de la fila i
y X_i es la prioridad de la fila i

Fórmula 4. *Importancia de la característica técnica i*

Para ejemplificar, tomando la figura 56, los valores de Y_1 y de Y_3 serían el resultado de las siguientes ecuaciones:

$Y_1 = (9 \times X_2) + (1 \times X_3)$

$Y_3 = (3 \times X_1) + (3 \times X_4)$

Concluyendo, un valor de ponderación total alto sugiere que esta característica técnica debe atenderse con alta prioridad y cuidado por parte del equipo de trabajo.

3.4.8 Tarea 8: Elaborar el Documento de Análisis de Requisitos

Este documento se utiliza para especificar los elementos del modelo, así mismo, incluye las matrices de relación elaboradas en la tarea 3, donde se analizan los requisitos desde diferentes perspectivas. Igualmente, estas matrices permiten establecer relaciones de trazabilidad entre los requisitos elicitados. También se deben incluir las plantillas de los conflictos identificados durante el proceso.

3.4.8.1 Partes del documento de Análisis

A continuación se hará una descripción detallada de las partes que conformarán el documento de análisis de requisitos, basados en algunas de las indicaciones dadas por Amador Durán en [Durán, 2000].

- **Portada:** Las partes que conforman la portada se pueden visualizar en la figura 57.

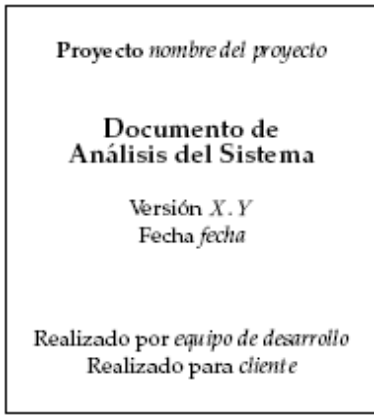


Figura 57. *Portada del documento de análisis del sistema*
Fuente: [Durán, 2000]

La explicación de cada uno de los campos de la portada es similar a la mostrada en la sección 3.3.4 "*Elaborar el Documento de Requisitos del Sistema*".

- Igualmente se deben incluir en el documento las siguientes secciones, explicadas anteriormente en la sección 3.3.4:

  - Lista de cambios
  - Índice
  - Lista de figuras y tablas
  - Introducción

- **Requisitos del sistema clasificados:** En esta sección las necesidades elicitadas deberán clasificarse en requisitos funcionales, no funcionales y de información, basándose en los resultados de la tarea 1 del análisis de requisitos.

- **Conflictos:** En esta sección se deben incluir todos los conflictos que se vayan identificando durante la etapa de análisis, especificando la situación presentada, el nombre de los objetivos o requisitos en conflicto, posibles alternativas, soluciones tomadas y responsable. En este punto se incluyen los resultados de la sección 3.4.2 *"Identificación de conflictos en los requisitos".*

- **Matrices de relación:** En este punto se incluyen las matrices identificadas en la sección 3.4.3 *"Elaborar matrices de relación".*

- **Resultados del modelo de Kano:** Se especifican los resultados obtenidos con la aplicación del modelo y las matrices de clasificación de los requisitos.

- **Casa de la Calidad:** En esta sección se incluye la Casa de la Calidad que logró construirse, luego de ejecutar una a una las tareas propuestas hasta ese punto.

- **Casos de Uso:** En esta sección se incluyen las especificaciones y los diagramas de los casos de uso que se hayan identificado durante la construcción de la Casa de la Calidad.

- **Glosario de términos:** En esta parte se amplía el glosario de términos que sea necesario especificar luego de la etapa de análisis y sumárselo a los términos ya definidos en la tarea 4 de la sección 3.2 *"Modelado del Negocio"*, preferiblemente en orden alfabético.

- **Apéndices:** De forma similar a los apéndices del documento de requisitos, deben ser incluidos en el documento de análisis si se ve la necesidad de adicionar información que pueda ser relevante para el documento. Por ejemplo, los cuestionarios utilizados para la realización del modelo de Kano.

En la figura 58 se puede observar un esquema del documento análisis de requisitos:

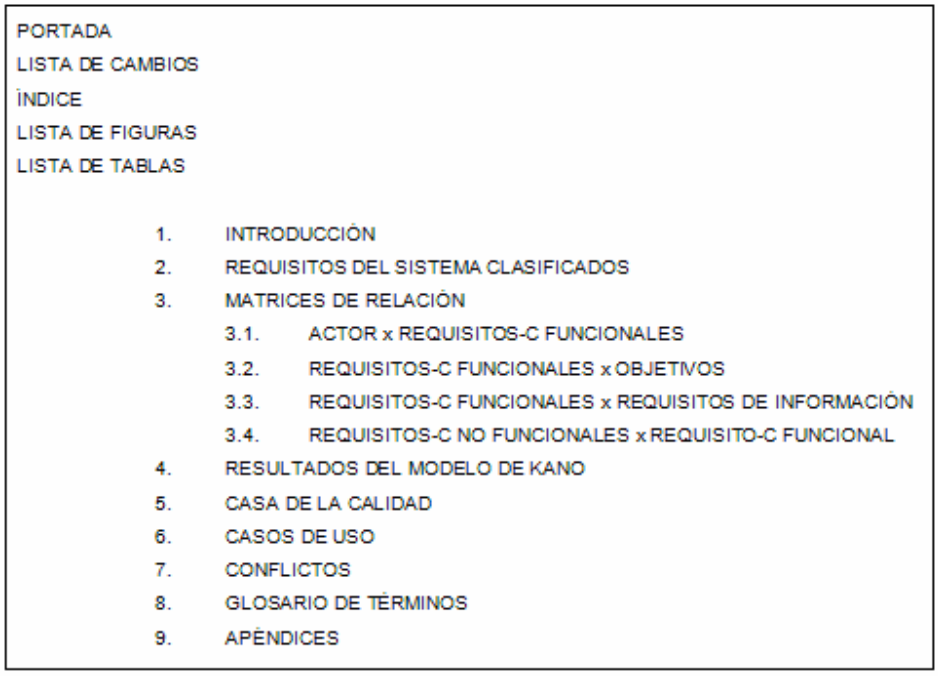


Figura 58. *Estructura del documento de análisis de requisitos*

Fuente: Elaboración propia

RD SP 3.2 - Establecer una definición de la funcionalidad requerida
La definición del documento anterior permitirá generar una primera consolidación sobre el proyecto que se va a realizar, de esta forma el equipo tendrá claramente definido e identificado qué es lo que se pretende desarrollar y cuál será la funcionalidad del producto. Lo anterior, permite darle cumplimiento a la práctica de la tercera meta del área de proceso RD: "Establecer y mantener una definición de la funcionalidad requerida".

3.5 VALIDACIÓN DE REQUISITOS

La validación es un proceso en el cual todos los stakeholders participan con el fin de asegurar que el conjunto de requisitos de software incluidos en el documento de especificación de requisitos, define realmente el sistema que el usuario necesita o el cliente desea. Esta actividad debe ser realizada a conciencia, dado que, en muchos casos, la especificación de los requisitos actúa como un contrato entre los stakeholders, pues en él se define qué es lo que se va a construir. En síntesis, el objetivo principal de la validación es descubrir problemas en el documento de requisitos antes de comprometer recursos para su implementación.

Generalmente es muy difícil, o incluso imposible, evaluar una especificación de requisitos y decir con absoluta seguridad si dicha especificación contiene las necesidades implícitas y explícitas de los clientes y usuarios. En [Nuseibeh y Easterbrook, 2000] se expone que esta actividad es difícil, principalmente por dos razones. La primera razón es de naturaleza filosófica y tiene que ver con la cuestión de la verdad y lo que es conocible. La segunda razón está relacionada con el aspecto social y concierne la dificultad para alcanzar un acuerdo entre diferentes participantes con objetivos y metas que se encuentran en conflicto.

Como resultado de esta validación se produce una línea base (baseline). En el contexto de la Ingeniería de Requisitos, una línea base es un conjunto de requisitos que han sido formalmente aceptados por todas las personas implicadas

en el proyecto. Una vez que se establece una línea base, futuros cambios a tales requisitos sólo podrán realizarse por medio de un proceso formal de gestión y aprobación de cambios.

A continuación se explican brevemente algunas técnicas existentes para realizar la validación de los requisitos:

- **Reviews o Walk-throughs**: Esta técnica consiste en la lectura y corrección de la documentación o modelado de la definición de requisitos. Con ello se puede validar la correcta interpretación de la información transmitida; sin embargo se hace difícil verificar la consistencia de la documentación e identificar información faltante.

- **Auditorías:** La revisión de la documentación con esta técnica consiste en un chequeo de los resultados contra un checklist definido a comienzos del proceso; es decir, que con esta técnica sólo es revisada una muestra de los requisitos.

- **Matrices de trazabilidad**: Esta técnica consiste en marcar los objetivos del sistema y chequearlos contra los requisitos del mismo. Es necesario ir viendo qué objetivos cubre cada requisito y de esta forma se podrán detectar inconsistencias u objetivos no cubiertos.

- **Prototipos**: La construcción de prototipos del sistema a desarrollar puede facilitar enormemente tanto la elicitación de requisitos como la validación de los mismos, dado que permiten comprobar la adecuación o necesidad de un requisito antes de desarrollarlo.

  Los prototipos de descarte generalmente están relacionados con las interfaces de usuario (cada vez más importantes, por ejemplo, para el desarrollo de aplicaciones soportadas en Internet). Además, el uso de prototipos tiene las siguientes ventajas:

- Minimizan los malentendidos entre usuarios e ingenieros de requisitos.
- Facilitan la detección de funcionalidad no considerada o poco refinada.
- Dan una idea anticipada de las posibilidades del sistema final.
- Facilitan el entrenamiento anticipado de los usuarios para el manejo de la aplicación final.

Como argumento en contra, se puede decir que el empleo de prototipos representa un esfuerzo adicional en el desarrollo de software y las personas involucradas, por lo tanto su utilización implica un mayor compromiso en la mejora de la relación costo/beneficio por parte del grupo. Además, es muy importante que los clientes y usuarios estén conscientes de que los prototipos no constituyen el sistema final, sino una herramienta para obtener más información sobre sus necesidades.

Es importante resaltar que algunas de las herramientas utilizadas para realizar el análisis de requisitos también son útiles en la validación de los mismos. Por ejemplo, la matriz de relación Requisitos-C x Objetivos permite ver la trazabilidad de las necesidades elicitadas con cada uno de los objetivos del sistema, facilitándose, de esta manera, la identificación de aquellos objetivos que no están cubiertos por ningún requisito. Igualmente, a través de la Casa de la Calidad se puede observar la trazabilidad de las necesidades (Requisitos-C) a requisitos de producto (Requisitos-D).

Para realizar esta actividad, se tomará como base la propuesta de Amador Durán, en la cual se propone el uso de dos técnicas de validación conjuntamente: *walk – throughs y prototipos de interfaz de usuario*.

Se considera el uso de prototipos especialmente útil para efectuar las labores de validación de los requisitos, dado que para los clientes y usuarios es más difícil hacerse una idea de cómo funcionará el software a partir de una especificación formal de requisitos. Este hecho se refleja perfectamente en una frase tomada de [Durán, 2000]: “Esta técnica [prototipos] permite que los usuarios tengan una idea

más clara del producto que van a recibir, ya que su grado de comunicabilidad es más alto que el de las especificaciones textuales de requisitos".

La importancia de los prototipos radica en que ellos proveen una dimensión visual de la especificación de requisitos.

A continuación se listan algunas situaciones, expuestas en [López, 2004], en las cuales la utilización de prototipos puede ser fundamental para el éxito de un proyecto.

- El área de aplicación no está bien definida.
- El coste de rechazo por parte de los usuarios, por no cumplir sus expectativas, es muy alto.
- Es necesario evaluar previamente el impacto del sistema en los usuarios y en la organización.
- Se usan nuevos métodos, técnicas, tecnología.

La actividad de validación está divida en las siguientes tareas:

1. Construcción de prototipos
2. Preparar y realizar las sesiones de walkthrough
3. Validar los requisitos de almacenamiento de información y funcionales
4. Validar los requisitos no funcionales
5. Cerrar la versión del documento de especificación de requisitos

A continuación se describe detalladamente cada una de estas tareas.

3.5.1 Tarea 1: Construcción de prototipos

El objetivo de esta tarea es construir un prototipo que ilustre el conjunto de requisitos elicitados y analizados previamente, con el fin de utilizarlo en las sesiones de validación.

Como se dijo anteriormente, los prototipos pueden ser de dos tipos: *De usar y tirar (desechables) o Evolutivos.* A continuación se enuncian las principales características de cada uno de ellos.

- **De usar y tirar (desechable)**
  - Sirven para el análisis y validación de los requisitos, especialmente aquellos relacionados con la interfaz de usuario.
  - Después de que se redacta y acepta la especificación del sistema, se desecha el prototipo.
  - La aplicación se desarrolla siguiendo un paradigma diferente.
- **Evolutivos:**
  - Comienzan con un sistema relativamente simple que implementa los requisitos más importantes o mejor conocidos.
  - El prototipo se aumenta o cambia en cuanto se descubren nuevos requisitos.
  - Finalmente, se convierten en el sistema requerido.
  - Actualmente se usan en el desarrollo de sitios Web y en aplicaciones de comercio electrónico.

Adicionalmente, los prototipos pueden tomar muchas formas, tales como:

- **Storyboard:** El término storyboard es muy utilizado en publicidad y en producción de video para referirse a una serie secuencial de figuras o fotos, creada para comunicar la apariencia deseada en una escena o película. En software, este término se usa para referirse a un bosquejo o borrador que representa cada página en un sitio Web. Como un diagrama de flujo, un storyboard muestra las relaciones de una página con las otras.

- **Modelos de Pantallas:** Se dibujan o se toman de otros sistemas "pantallazos" o screenshots que ilustren cómo quedará el sistema.

- **Bocetos en Papel:** El ingeniero de requisitos dibuja un boceto para mostrar al usuario algunos aspectos importantes en la interfaz de la aplicación. Aunque esta técnica es muy restringida, puede ser muy útil en sesiones de elicitación donde se necesite ejemplificar rápidamente algunos aspectos de la aplicación.

Para desarrollar los prototipos se pueden utilizar herramientas tales como:

- Generadores de interfaz (4GLs, Visual Basic, Java, etc.).
- Editores de páginas Web (Dreamweaver, Frontpage, etc.).
- Herramientas CASE (Formularios, pantallas, generación de código, etc.)
- Bocetos en papel, como se dijo anteriormente.
- Aplicaciones de dibujo (Harward Graphics, etc.).
- MS PowerPoint.
- Otras.

RD SP 3.4 - Analizar los requisitos para alcanzar el balance
Con la construcción de prototipos se esta siendo congruente con la práctica específica "Analizar los requisitos para alcanzar el balance", pues en ellos se podrán cruzar las necesidades de los stakeholders con las restricciones identificadas durante estos análisis.

3.5.2 Tarea 2: Preparar y realizar las sesiones de walkthrough

El objetivo principal de una sesión de walkthrough es encontrar conflictos (defectos, omisiones, contradicciones) en el producto que se revisa, de forma que puedan plantearse alternativas y los participantes aumenten su conocimiento sobre el producto en cuestión.

Durante las sesiones de walkthrough, el autor del producto *recorre* el producto a revisar en detalle, permitiendo que los participantes pongan de manifiesto sus opiniones sobre el mismo.

3.5.2.1 Participantes del walkthrough

Para lograr una sesión exitosa de validación de los requisitos por medio de walkthrough, es muy importante primero que todo, identificar aquellas personas que deberán participar en dichas sesiones. En [Durán, 2000] se exponen los siguientes tipos de participantes o roles:

- **Autor**: Es el responsable de la elaboración del producto que se evalúa durante la sesión de walkthrough. Debe asegurarse que el producto tiene el tamaño adecuado para poder ser evaluado en un máximo de tiempo (aproximadamente dos horas) y que no contiene faltas ortográficas ni defectos, de forma que no se desvíe la atención durante la sesión de walkthrough. Debe también preparar la documentación complementaria para situar a los participantes en el contexto del producto que se va a revisar.

  El autor debe asistir a la reunión para resolver las cuestiones y dudas que puedan surgir en el transcurso de ésta, de forma que, una vez finalizada la reunión, incorpore al producto los cambios sugeridos en la reunión y lo reelabore, en el caso de que haya que someterlo a una nueva sesión.

- **Presentador:** Es el responsable de organizar y dirigir la sesión, de seleccionar a los participantes entre los que debe encontrarse el autor del producto a revisar, y de acordar con ellos la fecha y el lugar de la reunión.

  También debe elegir al moderador y al registrador, entregar la documentación a los revisores, dar una visión general del producto, resumir el estado de salida de la última sesión de walkthrough si la hubo, presentar el producto y tomar la decisión final si no hay consenso sobre el estado del producto. Además, elaborará la documentación resultante de la reunión.

  Es relativamente frecuente que el autor y el presentador sean la misma persona.

- **Revisores:** Son los principales participantes de la sesión. Antes de la reunión deben revisar cuidadosamente la información que han recibido y anotar los comentarios que desean exponer durante la reunión. En el transcurso de ésta serán participantes activos explicando dichos comentarios y realizando al autor las preguntas que consideren necesarias para entender el producto y poder calificarlo. Son los responsables de decidir si el producto se acepta, si es necesario corregirlo y volver a convocar una reunión o si es suficiente con revisarlo. En el caso de los walkthrough sobre especificaciones de requisitos, los revisores serán principalmente los clientes y usuarios del producto final.

- **Moderador:** Antes de la reunión, debe aclarar los roles y responsabilidades de los asistentes con el presentador. También es responsable de reconducir la reunión si se desvía de los puntos principales y de que se realice de forma ordenada. Una vez terminada la sesión, debe encargarse de consultar a los participantes su opinión sobre el resultado de la reunión.

- **Registrador:** Es el responsable de registrar la hora de comienzo de la sesión, los conflictos expresados por los revisores durante la sesión, las acciones sugeridas por éstos y la hora de finalización de la sesión.

3.5.2.2 Fases del walkthrough

Dentro de la técnica de walkthrough se distinguen tres fases principales [Durán, 2000]: *Preparación, Revisión y Conclusiones.*

1. **Preparación:** Para que la sesión se desarrolle satisfactoriamente, los participantes deben recibir con anterioridad a la celebración de la reunión el material correspondiente al producto a revisar para que puedan estudiarlo y preparar las preguntas o los temas a discutir que consideren oportunos.

   Por lo tanto se recomienda enviar previamente el prototipo y el documento de requisitos obtenido en la tarea 4 de la sección 3.3 "*Elicitación de Requisitos*", y anexar a este documento los casos de uso identificados en la tarea 6 del análisis de requisitos, pues estos guiarán posteriormente el walkthrough.

2. **Revisión:** Durante la celebración de la sesión de revisión el presentador debe *recorrer* (*walk through*) con detalle, tanto el prototipo como el documento de requisitos, de forma que los participantes puedan plantear las cuestiones que consideren oportunas.

   El registrador tomará nota de éstas elaborando una lista de conflictos (defectos, omisiones, etc.). El moderador debe intentar que los participantes no busquen soluciones a los conflictos planteados sino que se limiten a exponerlos. Como resultado de la reunión, el documento de especificación de requisitos puede aceptarse, recomendar ciertos cambios o acordarse la celebración de una nueva sesión, si la naturaleza de los cambios necesarios lo requiriese.

3. **Conclusión:** Una vez terminada la sesión es necesario transformar la documentación generada en documentos formales que deben recoger la información referente a la sesión: nombre del producto revisado, fecha de la sesión, participantes, participantes implicados en la siguiente sesión de

walkthrough si procede, y la lista enumerada de aportaciones realizadas por los participantes, así como las acciones correctivas que se hayan propuesto.

Como resultado de la realización de las sesiones de walkthrough pueden detectarse conflictos y descubrirse nuevos requisitos.

Las dos tareas siguientes se deben realizar durante las sesiones de walkthrough; sin embargo, se decidió explicarlas separadamente para un mejor entendimiento.

REQM SP 1.1 - Obtener un entendimiento de los requisitos
Durante las sesiones de walkthrough dirigidas con los prototipos y los casos de uso, los clientes y usuarios pueden identificar si el equipo desarrollador comprendió completamente sus necesidades, a su vez, el equipo desarrollador puede determinar si los casos de uso realizados hasta el momento cumplen con las necesidades elicitadas y, de este modo, llegar a un acuerdo entre las partes. Estas sesiones aseguran el cumplimiento de la práctica específica "Obtener un entendimiento de los requisitos" del área de proceso REQM.

3.5.3 Tarea 3: Validar los requisitos de almacenamiento de información y funcionales

El objetivo de esta tarea es validar los requisitos de almacenamiento de información y funcionales, es decir, asegurarse que representan realmente las necesidades de clientes y usuarios.

Estos dos tipos de requisitos se agruparon en una única tarea, dado que las sesiones de walkthrough asistidas con prototipos de interfaz de usuario y guiadas por la especificación de los requisitos permiten una validación conjunta de forma sencilla de los requisitos mencionados, teniendo siempre en cuenta las relaciones de trazabilidad establecidas entre los requisitos de almacenamiento de

información y los requisitos funcionales en la matriz Requisitos-C Funcionales x Requisitos de Información expuesta en la sección 3.4.3.3.

Para validar los requisitos funcionales se recomienda recorrer los casos de uso identificados durante la construcción de la Casa de la Calidad, dado que en ellos está condensada toda la informa concerniente a las necesidades funcionales elicitadas.

Los productos resultantes de esta tarea serían los requisitos de almacenamiento de información, los requisitos funcionales y el prototipo, validados total o parcialmente. En el caso que se descubran conflictos, éstos deberán resolverse en nuevas sesiones de elicitación/negociación (ver sección 3.3).

3.5.4 Tarea 4: Validar los requisitos no funcionales

El objetivo de esta tarea es validar los requisitos no funcionales. Sus productos, de forma similar a la tarea anterior, serían los requisitos no funcionales validados total o parcialmente, y los posibles conflictos que pudieran aparecer.

Para realizar esta tarea los clientes y usuarios, ayudados por los ingenieros de requisitos, deberán revisar los requisitos no funcionales y su trazabilidad con los requisitos funcionales (ver matriz de relación Requisitos-C No Funcional X Requisitos Funcionales, sección 3.4.3.4) para aclarar las posibles dudas que surjan durante la sesión y de ser necesario, adicionar nuevos requisitos no funcionales o agregar relaciones con otros requisitos funcionales que no habían sido contempladas.

3.5.5 Tarea 5: Cerrar la versión del documento de especificación de requisitos

Si no han aparecido nuevos conflictos durante el proceso de validación, se debe llegar a un acuerdo entre clientes y desarrolladores para cerrar la versión actual de los requisitos y establecer la línea base de los mismos.

Como se dijo anteriormente, una vez que se establece esta línea base, futuros cambios a tales requisitos sólo podrán realizarse por medio de un proceso formal de gestión y aprobación de dichos cambios.

REQM SP 1.2 - Obtener un compromiso con los requisitos
Una vez los clientes y usuarios aceptan el conjunto de requisitos, se cierra la versión de requisitos con el fin de que el equipo desarrollador se comprometa a construir el producto descrito hasta el momento por los participantes y se establece una línea base para que los futuros cambios se vean reflejados en todo el proyecto (documentos, tiempo y costo). Con esta práctica se asegura el cumplimiento de la práctica específica "Obtener un compromiso con los requisitos" del área de proceso REQM.

RD SP 3.5 - Validar los requisitos
Esta etapa es concerniente con la quinta práctica específica de la tercera meta del área de proceso Desarrollo de Requisitos, en la cual se propone "Validar los requisitos", pues las diferentes tareas apuntan hacia la exploración de la adecuación de los requisitos definidos durante las etapas de elicitación y análisis; así mismo, porque permitirá refinar aquellos que lo requieran.

RD SG 3 - Analizar y validar los requisitos
Con el cumplimiento de las cinco prácticas específicas "Establecer conceptos y escenarios operacionales", "Establecer una definición de funcionalidad requerida", "Analizar los requisitos", "Analizar los requisitos para alcanzar el balance" y "Validar los requisitos", se asegura el cumplimiento de la tercera meta del área de proceso Desarrollo de Requisitos.

3.6 GESTIÓN DE REQUISITOS

Debido a que un proyecto informático es susceptible de cambios, habría que proceder a su actualización por medio de la incorporación o eliminación de ciertas funcionalidades, lo cual indica que es necesario mantener controlado y documentado el producto. La Gestión de Requisitos es el proceso encargado de relacionar los usuarios, los requisitos y el negocio, asegurando el cumplimiento de las especificaciones; asimismo, tiene como tarea la identificación, asignación y seguimiento de los requisitos, incluyendo verificación, modificación y control a lo largo del ciclo de vida del producto de software.

Un estudio realizado por Meta Group descubrió que aproximadamente entre el 60% y 70% de los proyectos fallan por la pobre recopilación, análisis y gestión de requisitos, esto es porque un proyecto de software exitoso es aquel que satisface al usuario; si sus requisitos no son completamente definidos y documentados, esto puede afectar a todo el proyecto. [McDonald, 2005]

Este proceso aumenta la comunicación y la colaboración, garantizando que todos los participantes tengan acceso a los objetivos del negocio comunes que guían la empresa o a los requisitos del cliente que definen el proyecto de desarrollo. De esta forma, los jefes de proyecto obtienen la capacidad de realizar un seguimiento del estado del proyecto al controlar el progreso del trabajo que se está realizando, con el fin de cumplir los requisitos.

La gestión de requisitos constituye un instrumento estratégico con enfoque proactivo, que permite al gestor del proyecto adoptar, desarrollar e implementar adecuadamente las actividades de este proceso de una forma disciplinada, coherente y repetitiva. Esto permite proveer una relación entre requisitos, diseño e implementación del sistema y el reconocimiento temprano de aquellos que no son satisfechos.

En [Monografías, Gestión], se expone que la gestión de requisitos implica una serie de aspectos relevantes para: evitar altos niveles de retrabajo, controlar el número de peticiones de cambios y evitar que se incremente el tiempo y costos del proyecto al tener que revisar una y otra vez los cambios en los requisitos.

Para alcanzar estos objetivos se propone:

- Definir procedimientos que establezcan los pasos y los análisis que se realizarán antes de aceptar los cambios propuestos
- Establecer un grupo encargado del control de cambios
- Crear líneas base y controlar las versiones de los requisitos
- Mantener la trazabilidad hacia atrás y hacia adelante entre los requisitos, realizando un seguimiento a su estado y midiendo su volatilidad
- Controlar las versiones de los documentos de requisitos y mantener un historial de los cambios
- Usar herramientas para la gestión de requisitos
- Crear matrices de trazabilidad de requisitos

De esta forma, una eficiente gestión de requisitos a lo largo de todo el ciclo de vida del proyecto contribuye eficazmente a la calidad del producto final y al grado de satisfacción del cliente.

Desde el inicio hay que establecer una **línea base** de requisitos, como un canal simple para el control de cambios. Esta se refiere al conjunto de requisitos funcionales y no funcionales que el equipo del proyecto se ha comprometido a implementar en una versión específica, previamente aprobada en la especificación de requisitos. A continuación se describirán las tareas que se deben llevar a cabo en el proceso de gestión de requisitos, cada una de estas provee beneficios, tanto

al cliente como al desarrollador, con el fin de asegurar que el proyecto se está moviendo en la dirección correcta [McDonald, 2005] y [Monografías, Gestión].

3.6.1 Gestión de cambios

Los cambios de requisitos deben ser gestionados para asegurar que la calidad de los mismos se mantenga, debido a que los problemas suscitados por dichos cambios podrían incurrir en altos costos, por lo cual puede considerarse el cambio en los requisitos como un factor crítico de riesgo. [McDonald, 2005]

La gestión de cambios es un proceso formal para identificar, evaluar, trazar y reportar cambios propuestos y aprobados a la especificación del producto. Como el proyecto va evolucionando, los requisitos pueden cambiar o expandirse para ajustar algunas modificaciones en el alcance o diseño del proyecto. Un proceso de gestión de cambios proporciona un rastreo completo y preciso de todos los cambios que son pertinentes al proyecto. El valor en el control de cambios en los requisitos es que el proyecto se mantiene encaminado, reduciendo o eliminando costos y posibles retrasos en los cronogramas.

Esta tarea involucra modificar el tiempo en el que se va a implementar una característica en particular, lo cual puede tener impacto en otros requisitos. Por esto, la gestión de cambios involucra actividades como establecer políticas, guardar históricos de cada requisito, identificar dependencias entre ellos y mantener un control de versiones.

Otra ventaja significativa se da mientras se prueba el sistema, pues cada involucrado ha logrado comprender cuáles requisitos fueron aprobados y qué cosas hará, utilizará y tendrá el sistema. De esta forma los límites establecidos por los requisitos son una forma de medir el éxito del cumplimiento del desarrollo del proyecto.

En vista que las peticiones de cambios provienen de muchas fuentes, las mismas deben ser enrutadas en un solo proceso. Esto se hace con la finalidad de evitar problemas y conseguir estabilidad en los requisitos.

REQM SP 1.3 - Administrar los cambios en los requisitos
Este paso es uno de los que requieren más cuidado en el proceso de gestión de requisitos, siendo coherente con la tercera práctica del área de proceso REQM del modelo CMMI "Administrar cambios en los requisitos" al tener en cuenta que se deben gestionar eficiente y efectivamente los cambios en las especificaciones de requisitos y que es importante analizar el impacto que puede ocasionar un cambio en un requisito antes de aceptarlo.

Algunos de los pasos por los cuales pasa una solicitud de cambio, se explicarán a continuación y se podrá visualizar mejor este proceso en la figura 59.

- **Identificación del cambio**

  La identificación del cambio se inicia desde el análisis de requisitos, nuevas necesidades del cliente y problemas operacionales; después se realiza el análisis del cambio y evaluación de costos, debiendo quedar claro cuántos requisitos se verán afectados y cuánto costará (tiempo y recursos); sólo después se toma la decisión de la implementación del cambio, que luego será documentada. Para identificar los requisitos afectados por un cambio, se pueden utilizar las matrices de relación y la Casa de la Calidad.

- **Ejecución de cambios**

  Aprobado el cambio, se procede a su implementación, de acuerdo con la fase del proyecto a que corresponda. En caso de que los cambios aprobados impliquen la implementación de nuevos requisitos, entonces será necesario retomar la etapa de elicitación de requisitos.

- **Trazabilidad**

  La trazabilidad de requisitos se define como la habilidad para describir y seguir la vida de un requisito en ambos sentidos, hacia sus orígenes o hacia su implementación, a través de todas las especificaciones generadas durante el proceso de desarrollo de software. Se compone de dos actividades importantes [Anaya y Letelier, 2000]:

  - Configuración de la trazabilidad, de acuerdo con las necesidades concretas del proyecto, para así conseguir un resultado positivo respecto del costo-beneficio asociado.

  - Especificación de la trazabilidad en el proyecto y la posterior explotación de dicha información.

  Los requisitos deben ser trazables por su origen (¿quién lo propuso?); necesidad (¿por qué existe?); por su relación con otros requisitos; por su relación con elementos del diseño y/o la implementación.

  La trazabilidad se cumple en la metodología por medio de las matrices de relación, elaboradas en la tarea 3 de la elicitación, y posteriormente con la construcción de la Casa de la Calidad en la tarea 6 del análisis de requisitos.

REQM SP 1.4 - Mantener una trazabilidad bidireccional de los requisitos
Este punto concuerda con la cuarta práctica del área de proceso REQM del modelo "Mantener una trazabilidad bidireccional de los requisitos", puesto que la metodología propuesta permite visualizar de forma fácil la trazabilidad de las necesidades y objetivos del software con las funcionalidades del producto. Se debe aclarar que la metodología da las pautas para mantener la trazabilidad durante la etapa de ingeniería de requisitos, sin embargo, para cumplir con la práctica, se debe conservar esta trazabilidad en el resto del ciclo de vida.

- **Control de versiones**

  Habitualmente el documento de especificación de requisitos necesitará ser modificado a medida que progresa el producto software, como resultado de los cambios aprobados en requisitos individuales o grupos de ellos. Estos cambios se ven reflejados en el campo de versión descrito en el documento de requisitos (Ver tarea 4 *"Elaborar el documento de requisitos del sistema"* de la sección 3.3) y en el documento de análisis de requisitos (Ver tarea 8 *"Elaborar el documento de análisis de requisitos"* de la sección 3.4).

  Entre algunos de los beneficios que proporciona el control de versiones se encuentran: prevenir cambios no autorizados, guardar revisiones de los documentos de requisitos, recuperar versiones previas de los documentos y prevenir la modificación simultánea a los requisitos.

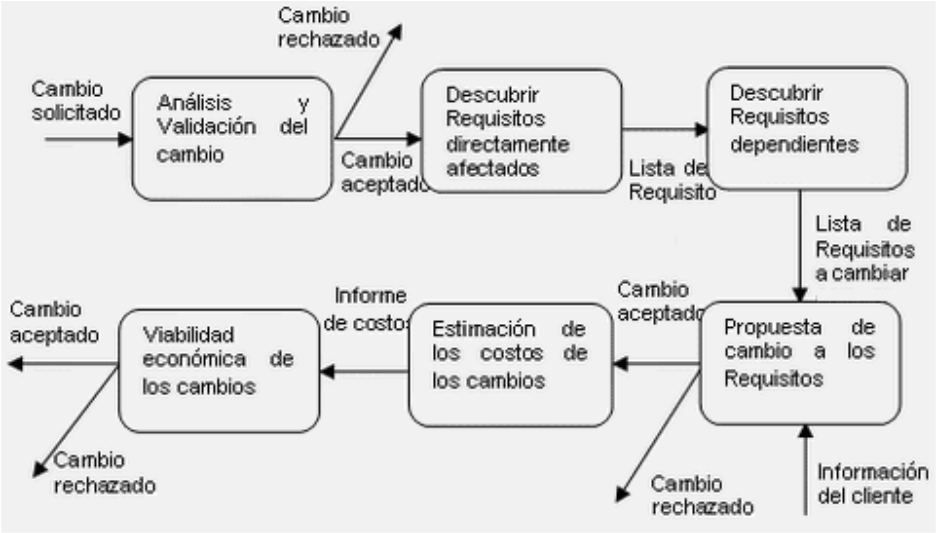


Figura 59. *Proceso de control de cambios*

Fuente: [Monografías, Gestión]

Herramientas para gestionar requisitos

El uso de herramientas de gestión de requisitos permite mejorar, tanto la productividad como la calidad en el desarrollo de un proyecto software.

En un proyecto de desarrollo de software la gestión de requisitos es una tarea intensiva que debe contar con la habilidad de relacionar diferentes documentos, de obtener una visión sinóptica de esta relación y de controlar los cambios de una manera consistente.

Estas herramientas permiten a los desarrolladores importar grandes documentos de una variedad de formatos estándar de procesadores de palabras. Los elementos del documento están sujetos a rigurosos cambios y a un régimen de control de versiones. Se puede establecer una relación entre los elementos del documento, y los atributos pueden ser asociados con los elementos del sistema y a menudo relacionados. Pueden ser generadas una variedad de vistas de documentos utilizando tanto los atributos como las relaciones, generalmente vistas específicas de trazabilidad, tales como matrices de trazabilidad.

Existen varias herramientas en el mercado que apoyan las tareas de gestión de requisitos; en [McDonald, 2005] sugieren algunas que pueden servir como apoyo para el proceso de gestión de requisitos y que serán explicadas a continuación:

RequisitePro[9]: Es una herramienta centrada en documentos, que almacena los requisitos asociándolos a documentos (aunque también permite guardarlos directamente en la base de datos). Apoya el control de cambio de requisitos, con trazabilidad para especificaciones de software y pruebas. Está muy unido a MS Word, así mismo es compatible con Oracle sobre Unix o Windows y SQL Server sobre Windows.

IRqA[10]: Es una herramienta especialmente diseñada para soportar el proceso completo de ingeniería de requisitos. En IRqA el ciclo de especificación incluye la captura de requisitos, análisis, especificación de sistema, validación y organización de requisitos, la cual es soportada por modelos estándares.

[9] RequisitePro: http://www-306.ibm.com/software/awdtools/reqpro/
[10] IRqA: http://www.irqaonline.com/

CaliberRM[11]**:** Esta herramienta, para sistemas grandes y complejos, proporciona una base de datos de requisitos con trazabilidad. Se ve a los requisitos como parte del proceso de gestión de la calidad del software; incluye pruebas (testing) y el trazado de defectos (defect tracking). Caliber está basado en Internet y maneja referencia de documentos, responsabilidad de usuario, trazabilidad, prioridad, estado, entre otros.

DOORS[12]**:** Considera los requisitos como objetos y los documentos como módulos. Tiene una orientación basada en objetos, frente a RequisitePro y Caliber-RM, que manejan solamente requisitos y sus atributos. Es una herramienta para organizaciones grandes que necesitan controlar complejos conjuntos de usuarios y requisitos de sistemas con una completa trazabilidad.

[11] CaliberRM: http://www.borland.com/us/products/caliber/index.html
[12] DOORS: http://www.telelogic.es/products/doors/doorsnet/index.cfm

CAPITULO 4: CASO PRÁCTICO

En este capítulo se exponen los resultados obtenidos al aplicar la metodología propuesta en el Colegio Anglo Español.

Dado que la Institución educativa tiene la necesidad de implantar un sistema que automatice parte de sus procesos, la información y colaboración ofrecida por ellos permitirá cumplir con el objetivo de evaluar la propuesta metodológica, identificando sus principales ventajas y falencias.

El capítulo está dividido en tres secciones; en la primera sección se hace una breve descripción de la historia y los programas ofrecidos por el Colegio Anglo Español; las secciones dos y tres corresponden con el documento de requisitos del sistema, explicado en la sección 3.3.4, y con el documento de análisis de requisitos elaborado en la octava tarea del análisis de requisitos. Además de la información contenida en los documentos, se anexan algunos datos recolectados en la realización de las diferentes tareas del proceso de ingeniería de requisitos.

4.1 CONTEXTO DE LA ORGANIZACIÓN

4.1.1 HISTORIA

En 1993 el Colegio Anglo Español surge como la materialización de un sueño de educadores egresados de diferentes Universidades de la Ciudad de Medellín, quienes en su trayectoria por la educación tradicional y masiva, concluyeron que los jóvenes actuales demandan de una metodología, unos contenidos curriculares que respondan a las necesidades del mundo actual y un acompañamiento personalizado que permita el desarrollo de sus potencialidades individuales para que se constituyan en seres triunfadores, líderes de la sociedad que les corresponda vivir.

El Colegio inició labores en una antigua casa del tradicional barrio Laureles de Medellín donde funcionó durante 6 años, luego la Sociedad fundadora adquirió la Finca "El Laguito" en Llanogrande, Rionegro, donde construyó la primera parte del proyecto de la Sede Campestre del Colegio Anglo Español. Después, la Institución es trasladada a una nueva sede en la ciudad de Rionegro, donde se encuentra situada actualmente.

Hasta el momento, la Institución ha egresado un conjunto numeroso de bachilleres y ha obtenido grandes satisfacciones en la aplicación del modelo pedagógico personalizado que la caracteriza.

Desde su inicio ha estado orientado por los Licenciados Nazly Rosa Medina Buelvas y Rubén Darío Orozco Palacio.

4.1.2 PROGRAMAS

El Colegio Anglo Español ofrece educación de carácter formal para niños, jóvenes y adultos, y programas técnicos laborales de carácter no formal para adultos.

1. **Para niños y jóvenes:** Los niveles de Preescolar, Básica (tanto primaria como secundaria) y Media Académica. Los grados correspondientes a estos niveles cumplen Ciclos Lectivos Regulares, es decir, anualizados.

2. **Para jóvenes en edad extraescolar y adultos laborales:** Los niveles de Básica Primaria y Secundaria y la Educación Media para optar el título de bachiller. Los grados en esta modalidad se cumplen en Ciclos Lectivos Especiales Integrados (CLEI) que permiten flexibilidad en la jornada de estudio, con modalidades presenciales de lunes a viernes o semi-presenciales en jornadas sabatinas o dominicales.

4.2 DOCUMENTO DE REQUISITOS DEL SISTEMA

4.2.1 PROCESOS DE NEGOCIO

Aunque en el Colegio Anglo Español se hace necesario automatizar gran parte de los procesos de negocio, pues actualmente todo se hace manualmente, para la realización de este caso práctico sólo se tendrán en cuenta tres procesos principales, pues se considera que son suficientes para evaluar la propuesta metodológica.

Los tres procesos seleccionados para aplicar la metodología son:

- Proceso de Matrícula
- Proceso de Administración de Folios
- Proceso de Calificación

A continuación se describen brevemente estos procesos, indicando las personas y entidades involucradas en cada uno de ellos y el conjunto de actividades que deben ser realizadas por dichas personas para conseguir cierto objetivo.

4.2.1.1 Proceso de Matrícula

Este proceso inicia con la llegada de las personas interesadas en ingresar a la Institución. La secretaria les hace entrega de una solicitud de ingreso que debe ser diligenciada por completo por los interesados y entregada de nuevo a la secretaria. Acto seguido, se les otorga una cita con la Rectora de la Institución, quien los entrevista y revisa cada uno de los aspectos relevantes del estudiante para saber si puede o no ser aceptado en el Colegio. Después de la entrevista con la rectora, se envía al estudiante a una entrevista con la psicóloga de la Institución para conocer más aspectos de su personalidad.

Con el fin de conocer cuál es el diagnóstico actual del conocimiento del alumno en ciertas materias relevantes, la Institución realiza tres exámenes para evaluar su estado; las materias examinadas son matemáticas, inglés y español.

Cuando el estudiante ha terminado exitosamente cada uno de estos procesos, se diligencia el libro de matrícula donde se almacena toda la información relevante del alumno y se le asigna un número de identificación interno (Folio).

4.2.1.2 Proceso de Administración de Libros

En todo momento, la Institución es contactada por estudiantes egresados, alumnos retirados u otras instituciones educativas, para consultar información sobre los estados financieros de algún alumno, calificaciones obtenidas en un lapso de tiempo, permanencia en la Institución, entre otras. Estos datos generalmente son entregados a los interesados en actas, informes, reportes o certificados, los cuales son realizados uno por uno por la Secretaria.

La consulta de la información siempre es realizada por la Secretaria y se debe buscar en cada uno de los libros que la contenga, para poder alcanzar los datos necesarios.

Entre la información más consultada en estos libros se encuentra aquella relacionada con los egresados de la Institución, quienes necesitan el informe de calificaciones final para su ingreso a la universidad, y la información de los alumnos que se retiran o son expulsados del Colegio, quienes tienen la obligación de llevar a cualquier Institución donde vayan a continuar sus estudios, toda la información referente a su permanencia en el Colegio.

Un aspecto importante de los informes y reportes generados por la Institución, es que deben cumplir con ciertas normas dictaminadas por el Ministerio de Educación Nacional.

4.2.1.3 Proceso de Calificación

El año lectivo de la Institución está divido en 4 periodos, en cada uno de los cuales se realiza una entrega de calificaciones a los padres de familia. Al comenzar el periodo, el profesor entrega a la Secretaria los logros que van a ser evaluados, ella los almacena para que posteriormente sean revisados por la Rectora. Al finalizar el periodo, el profesor confirma los logros que efectivamente fueron evaluados y la Secretaria con esta información, genera un formato de calificación completo por alumno, en el cual se encuentran la totalidad de logros evaluados en cada una de las materias.

Cuando el formato está totalmente elaborado, la Secretaria se los entrega a los profesores para que estos evalúen a todos los alumnos en una jornada pedagógica, en la cual la hoja académica de cada estudiante es pasada profesor por profesor para que cada uno asigne la calificación obtenida por el alumno. Después se hace entrega de estas calificaciones a los padres de familia o acudientes en una fecha estipulada por la Institución.

El Ministerio de Educación es el encargado de establecer la forma de representación de las calificaciones para los estudiantes de todo el país. Actualmente, las Instituciones deben calificar de forma cualitativa a los estudiantes, de acuerdo con el Decreto 230 del 2002, es decir, asignarles una letra para cada valor de la evaluación. Para el caso del Colegio Anglo Español, se realiza una valoración, en primera instancia, por números en el intervalo de 0 a 5, haciendo después una conversión de estos números a su respectivo valor cualitativo, con el fin de cumplir con la reglamentación. Cabe aclarar que para la Institución, los valores de 3.6 hacia abajo se consideran una calificación “en proceso”, es decir, que debe ser nivelada por el estudiante.

4.2.2 REGLAS DE NEGOCIO

- El sistema de calificaciones está dado por las estipulaciones del Ministerio de Educación. Actualmente, las calificaciones deben ser expresadas de forma cualitativa (letras), sin embargo, el Colegio mantiene una correspondencia entre las calificaciones cuantitativas (números de 0 a 5) y las cualitativas. La tabla 21 muestra la escala de conversión utilizada en el Colegio.

Calificación cuantitativa	Calificación cualitativa
0 – 1.9	Deficiente (D)
2 – 2.9	Insuficiente (I)
3 – 3.5	Aceptable (A)
3.6 – 4.5	Sobresaliente (S)
4.6 – 5	Excelente (E)

Tabla 21. *Correspondencia entre las calificaciones cuantitativas y cualitativas*
Fuente: Elaboración propia

- Las calificaciones por debajo de 3.6 se consideran no aprobadas.
- Un alumno debe repetir un grado cuando tiene más de dos materias por debajo de 3.6.
- Cuando un alumno pierde un logro debe nivelarlo durante el resto de año, si no lo hace y pasa de grado “debiendo” logros del año anterior, se le dan dos fechas para nivelarlo. En caso de que el alumno no nivele los logros entre las fechas estipuladas por la Institución, se devuelve al grado anterior, hasta cuando cumpla todos los logros.
- Las calificaciones en las hojas académicas de los alumnos del tutorial siempre deben ser superiores a 3.6, es decir, los profesores sólo pueden firmar la hoja académica del alumno cuando éste gana el logro.

- Un alumno puede cambiarse del Programa Regular al tutorial en cualquier momento.
- Durante el proceso de matrícula un alumno puede cambiarse de programa.
- Las matrículas en el Programa Tutorial son permanentes.

4.2.3 ROLES Y RESPONSABILIDADES

A continuación serán explicados los principales roles identificados para los procesos descritos anteriormente:

- **Rector:** Aunque la responsabilidad principal de este rol es de carácter administrativo, pues necesita información para la gestión, en el proceso de matrícula debe realizar entrevistas a los estudiantes interesados en ingresar a la Institución y esto le crea una participación más activa en este proceso. Para el proceso de calificación, el rector debe conocer los logros que los profesores realizan para revisarles el grado de calidad y observar si cumplen a cabalidad con los objetivos generales de las materias de la Institución.
- **Secretaria:** Es el rol que más participa en cada uno de los procesos de la Institución, dado que es el que administra los libros donde está contenida la información más importante para el Colegio. El proceso de matrícula, descrito anteriormente, comienza y termina con las actividades realizadas por este rol. Además las calificaciones deben pasar en cada una de sus etapas por las manos de la Secretaria, quien debe hacer los informes de calificaciones que van a ser entregados a los acudientes. La realización de documentos, el ingreso y digitalización de información, la elaboración de certificados, la recepción de documentación, entre otras, son las actividades principales de este participante.

- **Profesores:** Sólo son activos para el proceso de calificaciones. Ingresan los logros que van a ser calificados y diligencian las notas para cada uno de los alumnos.

- **Psicóloga:** Es encargada de realizar entrevistas para los alumnos interesados en ingresar al Colegio. Sólo es parte del proceso de matrícula y debe recopilar información de diagnóstico para cada estudiante, ayudando a determinar las posibles personas que no se acomodan a la metodología utilizada en la Institución.

- **Alumnos (Actuales, Egresados y Retirados):** Son el eje central de cada uno de los procesos. Son los "clientes" de la Institución y todas las soluciones planteadas deben cumplir con sus expectativas. Son partícipes de los tres procesos a sistematizar, pues son ellos quienes deben ser matriculados y calificados y adicionalmente realizan las peticiones de información y certificados.

- **Secretaría de Educación Nacional:** Este rol legisla la forma de comportamiento general de la Institución. Cuando se presenta una modificación en las leyes de educación debe cambiar, de cierta forma, algún proceso de la Institución; por consiguiente, se debe tener en cuenta como un rol participante en el proyecto, dado que tiene influencia sobre todos los procesos a sistematizar.

- **Acudientes:** Cada alumno menor de edad que es matriculado en la Institución debe tener un acudiente que se responsabilice de él. Las calificaciones son entregadas al alumno en compañía de su acudiente, quienes son informados por los profesores sobre las debilidades y/o fortalezas que se hayan presentado en la formación del estudiante.

- **Otras Instituciones educativas:** Cuando un alumno desea ingresar a la Institución, debe traer la información académica correspondiente al periodo de

tiempo de permanencia en otras Instituciones educativas. Igualmente, cuando un alumno se retira o es expulsado del Colegio, debe llevar esta información a la Institución donde desea continuar sus estudios.

Los alumnos egresados necesitan el reporte de calificaciones final de todos los años de bachillerato y el acta de grado para ingresar a la educación superior, por tanto estas Instituciones también deben ser tenidas en cuenta.

4.2.4 DESCRIPCIÓN DEL SISTEMA ACTUAL

Actualmente toda la información manejada por la Institución se encuentra en papel y está almacenada principalmente en 3 libros:

1. **Libro de matrículas:** En este libro se guardan los datos básicos de los alumnos como nombre, apellidos, nombre del acudiente, fecha de ingreso, etc. Este libro es llenado cuando un alumno acuerda ingresar a la Institución.

2. **Libro de calificaciones:** Existe uno por cada año, contiene las calificaciones obtenidas por cada uno de los alumnos en dicho año; las calificaciones son organizadas por período académico.

3. **Libro de estado financiero:** En este libro está contenido el estado financiero de cada uno de los alumnos que han pasado por el Colegio. Este libro es llenado mensualmente, con base en los pagos realizados por los alumnos de la Institución.

Estos libros son administrados por la Secretaria de la Institución, quien se encarga de diligenciar los diferentes folios, organizarlos en el libro que corresponda y de llevar todos los registros en un orden cronológico, según su ingreso al libro.

Se pudo detectar que este proceso es totalmente dependiente de las personas que lo realizan y se han encontrado problemas cuando el personal que diligencia los folios no es organizado, puesto que si no se referencia bien la ubicación de toda la información, es muy probable que posteriormente sea muy difícil de encontrar, esto sucede principalmente cuando los alumnos se cambian de un programa a otro. Además, cuando se cambian los empleados, este proceso se entorpece enormemente, pues la consulta de los libros indicados depende en gran medida de la memoria de las personas que ingresaron la información y muchas veces es necesario contactarlas para encontrar ciertos folios.

Dado que el manejo de la información es realizado en forma manual por los empleados de la Institución, implica un mayor consumo de tiempo y algunos reprocesos al ejecutar cada una de las operaciones que componen los procesos de negocio identificados. Así mismo, y con el pasar del tiempo, se ha detectado que el volumen de información se ha incrementado, lo cual ha dificultado la búsqueda, recuperación y análisis de la misma.

Una solución al problema identificado permitiría automatizar el ingreso y modificación de la información personal, tanto de los alumnos como de sus acudientes, al igual que el control de las calificaciones obtenidas por los alumnos durante toda su vida académica. De esta forma, se reduciría el tiempo que se emplea en realizar operaciones manuales y se podrían generar reportes y análisis que le permitan a las directivas del Colegio, tomar decisiones que aumenten la productividad del negocio.

4.2.5 PARTICIPANTES Y RESPONSABILIDADES

Participante	Rol	Responsabilidad
Nazly Medina	Rectora (Cliente / Usuario)	- Describir los proceso de negocio - Identificar reglas de negocio - Identificar problemas - Expresar sus expectativas del sistema - Expresar sus necesidades
Cristina Valencia	Secretaria (Usuario Líder)	- Describir los proceso de negocio que realiza directamente - Identificar problemas - Expresar sus expectativas del sistema - Expresar sus necesidades
Rubén Darío Orozco	Profesor (Usuario)	- Expresar sus expectativas del sistema - Expresar sus necesidades
César Darío Orozco	Profesor (Usuario)	- Expresar sus expectativas del sistema - Expresar sus necesidades
Carlos Darío Orozco	Ingeniero de Requisitos Desarrollador Tester	- Elicitación, análisis y validación de requisitos - Guiar las sesiones de elicitación - Resolver conflictos e incongruencias en los requisitos - Desarrollo de prototipos
Alejandra Maya	Ingeniero de Requisitos Desarrollador	- Elicitación, análisis y validación de requisitos - Guiar las sesiones de elicitación - Resolver conflictos e incongruencias en los requisitos - Desarrollo de prototipos
Carolina Arroyave	Ingeniero de Requisitos	- Análisis y validación de requisitos - Identificar conflictos en los requisitos

4.2.6 CATÁLOGO DE NECESIDADES

4.2.6.1 Diagramas de Afinidad

Sistematizar los libros académicos y de matrícula	
Digitalizar la información existente en los libros	Secretaria
Guardar los nuevos folios de las matrículas	Secretaria
Almacenar las notas finales de los alumnos del Programa Regular	Sistema
Almacenar las notas finales de los alumnos del Programa Tutorial	Sistema
Ingresar las nivelaciones de los alumnos (Regular)	Secretaria
Actualizar los datos de las matrículas	Secretaria
Permitir cambiar el estado de los alumnos a retirado, egresado o activo	Secretaria
Permitir cambiar a un alumno de programa	Secretaria
Ver toda la información de un alumno al ingresar el nombre	Secretaria, Rector

Listas después del proceso de matrícula	
Generar la lista de alumnos por grado en orden alfabético automáticamente	Secretaria
Generar el número de alumnos por grado y por programa	Secretaria
Generar la planilla de notas y asistencia para cada grado automáticamente	Secretaria
Generar la lista de alumnos con su acudiente y los teléfonos	Secretaria

Calificaciones del Programa Regular	
Registrar la planeación académica de cada profesor para cada materia y grado (logros)	Profesor
Quitar o modificar los logros de su planeación académica	Profesor
Crear hojas académicas para todos los alumnos	Sistema
Permitir que cada profesor evalúe a los alumnos (logrado o en proceso)	Profesor
Convertir automáticamente de calificación cuantitativa a cualitativa y guardar las dos notas	Sistema
Imprimir los informes de calificaciones para los acudientes	Secretaria
Imprimir los logros perdidos de cada alumno	Secretaria
Cambiar las notas de los logros perdidos de un alumno	Secretaria
Almacenar las notas finales de los alumnos del Programa Regular	Sistema
Imprimir informe final de notas	Secretaria
Permitir a los acudientes acceder a la hoja académica por Internet	Acudiente
Los alumnos no pueden pasar el año con más de dos materias perdidas	Sistema

Ingresar logros de comportamiento	Secretaria
Ingreso del servicio social para los estudiantes cursantes del grado Once	Secretaria

Calificaciones del Programa Tutorial	
Digitalizar los logros de las guías para cada CLEI	Secretaria
Permitir modificar los logros	Secretaria
Pasar las calificaciones de un alumno	Secretaria
Imprimir la hoja académica general	Secretaria
El sistema no debe permitir ingresar notas menores de 3.6	Sistema
Convertir automáticamente de calificación cuantitativa a cualitativa y guardar las dos notas	Sistema
Deberá generar un reporte automático cuando se cumplan todos los logros	Sistema
Deberá calcular la nota parcial automáticamente en la mitad del semestre	Sistema
Si el alumno se retira antes de terminar el CLEI, se debe calcular la nota parcial con lo que lleve	Sistema
Cambiar la modalidad de los alumnos (presencial o semipresencial)	Secretaria

Informes y reportes	
Generar actas de grado	Secretaria
Generar certificados de estudio	Secretaria
Generar certificados académicos	Secretaria
Imprimir informe final de notas	Secretaria
Imprimir lista de graduandos	Secretaria

Aspectos generales del sistema
Hacer respaldos de la información
Mostrar la información según la persona que entre al sistema
Cada usuario debe tener una contraseña
Los profesores no pueden ingresar notas por fuera de la Institución
Un profesor puede dictar varias materias
Varios profesores deben poder evaluar un alumno al mismo tiempo
El sistema debe tener ayudas
El sistema debe ser fácil de usar
El sistema debe tener los colores del Colegio

4.2.6.2 Diagramas de Árbol

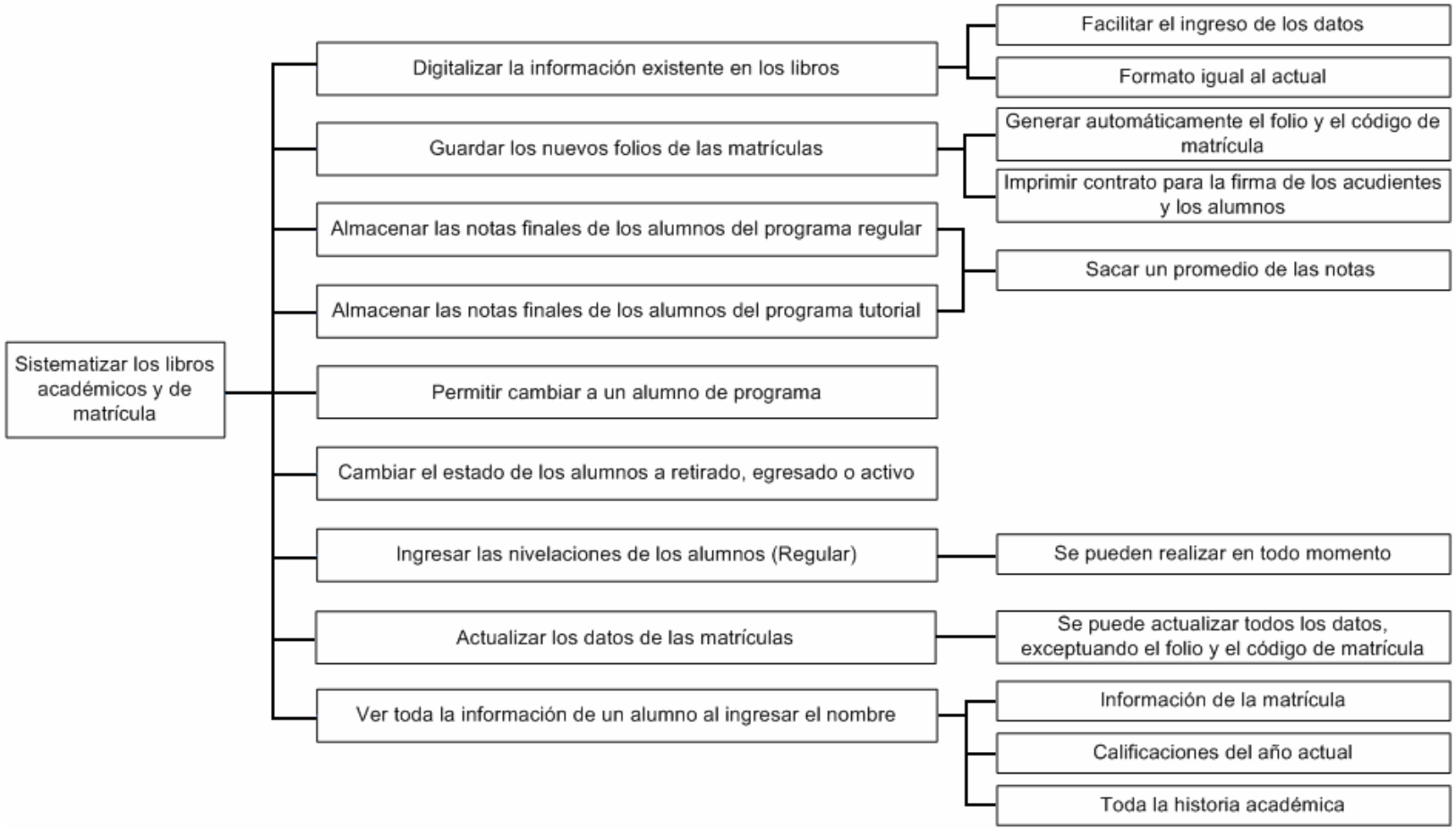


Figura 60. *Diagrama de árbol - Sistematizar los libros*

Fuente: Elaboración propia

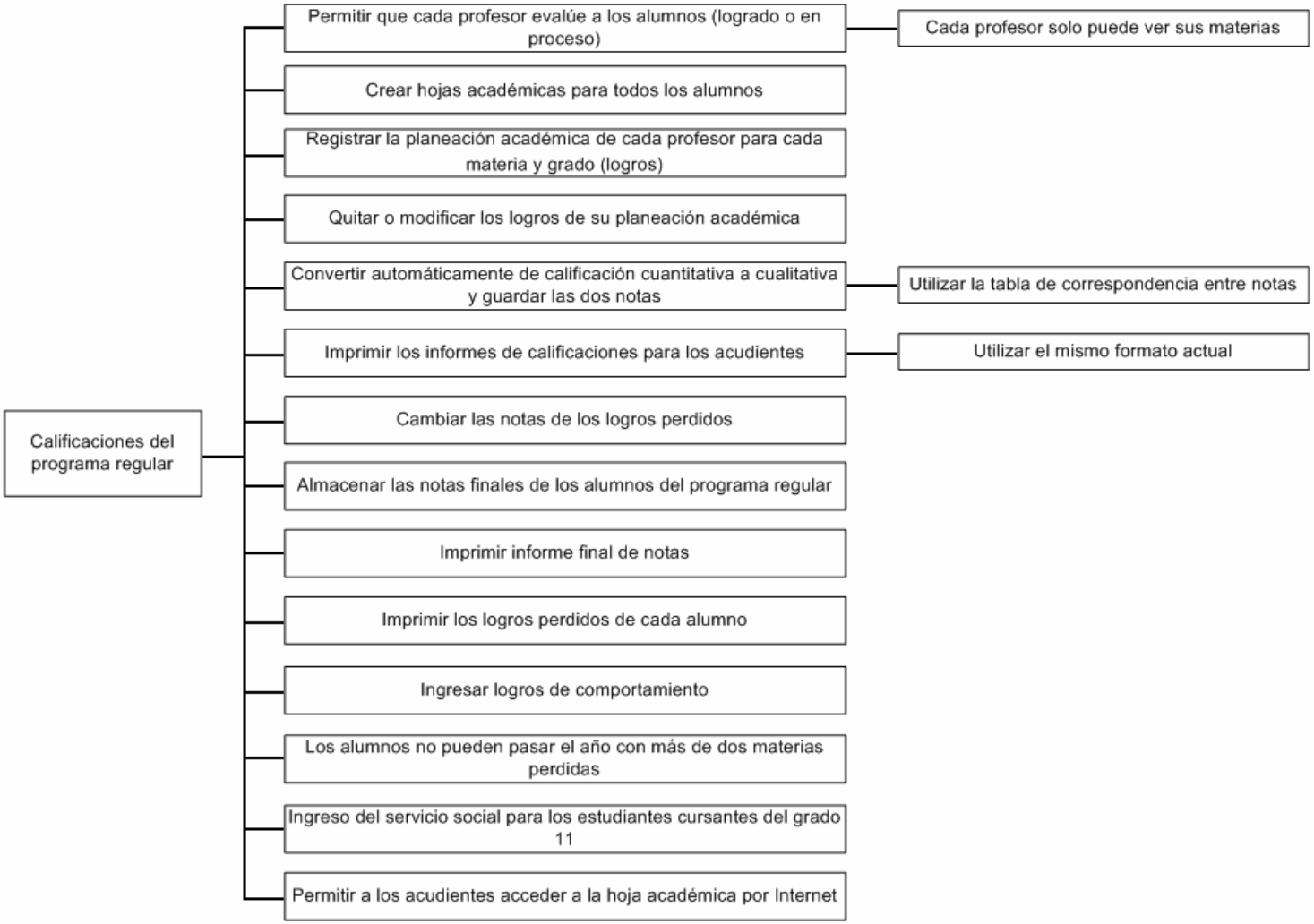


Figura 61. *Diagrama de árbol - Calificaciones del Programa Regular*

Fuente: Elaboración propia

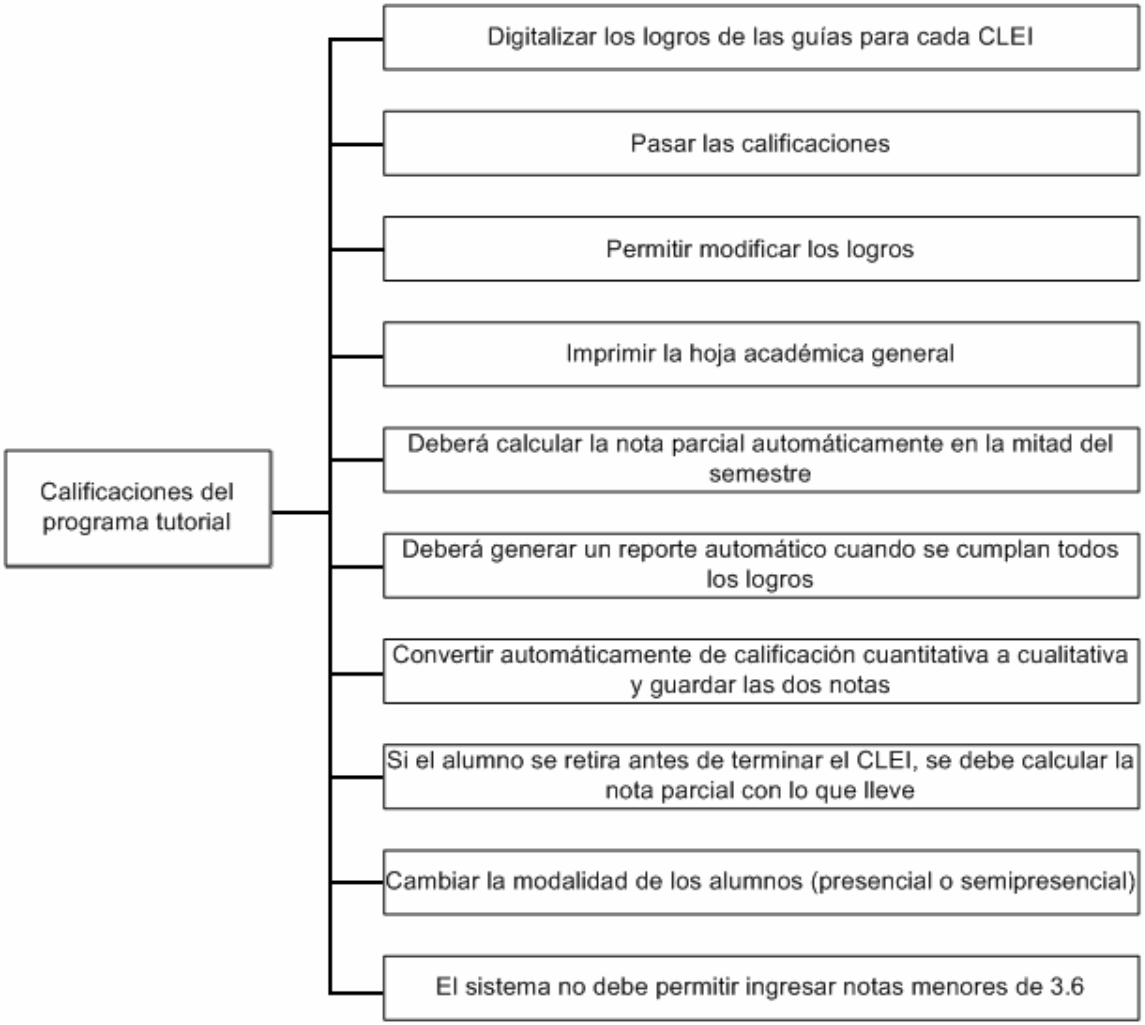


Figura 62. *Diagrama de árbol - Calificaciones del Programa Tutorial*

Fuente: Elaboración propia

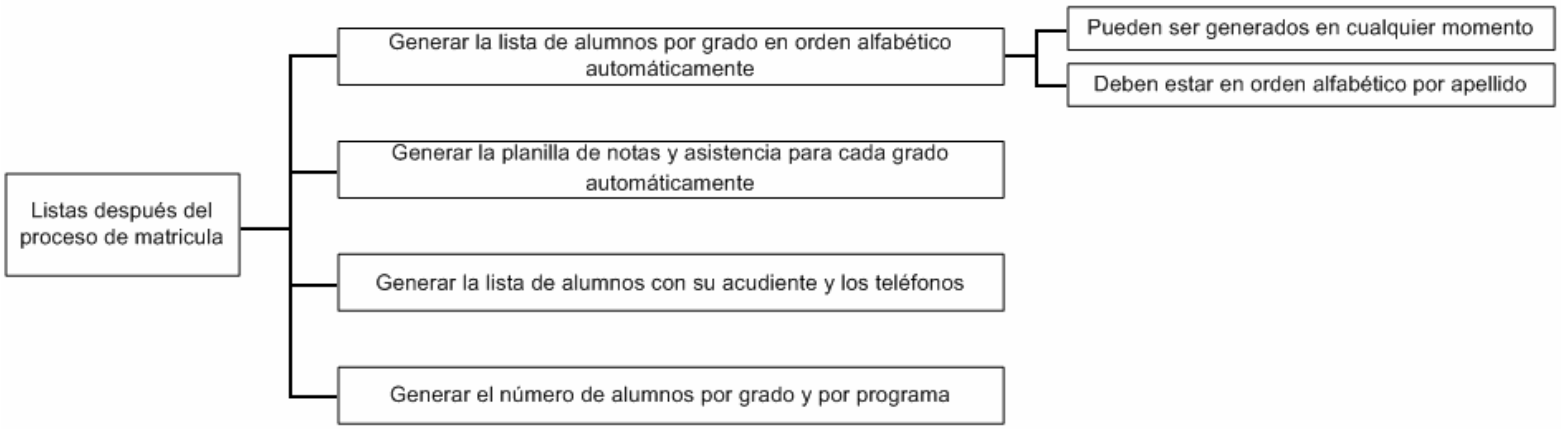


Figura 63. *Diagrama de árbol - Listas después del proceso de matrícula*

Fuente: Elaboración propia

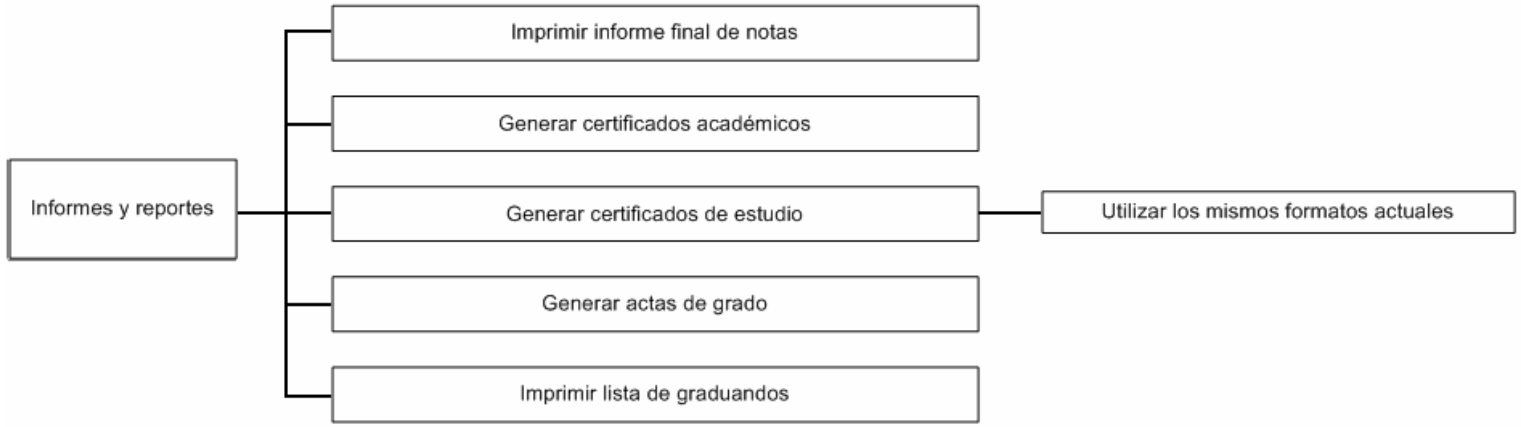


Figura 64. *Diagrama de árbol - Informes y reportes*

Fuente: Elaboración propia

Figura 65. *Diagrama de árbol - Aspectos generales del sistema*

Fuente: Elaboración propia

4.2.6.3 Definición de los Actores

- **Secretaria:** Este actor representa a la Secretaria del Colegio. Es el actor que tiene más interacciones con el sistema, ya que en el futuro gran parte de las tareas que realiza actualmente estarán automatizadas.

- **Profesor:** Este actor representa a los profesores del Colegio. Su principal interacción con el sistema la realiza durante el proceso de calificación, pues a través del sistema deberá ingresar los logros que va a evaluar durante el periodo y las calificaciones obtenidas por los alumnos en cada uno de ellos.

- **Rector:** Este actor representa al Rector del Colegio. Este actor tiene acceso a toda la información del sistema y puede ejecutar gran parte de las funcionalidades ofrecidas, sin embargo, es un actor de carácter más administrativo que de operación, por esta razón la mayoría de reportes generados por la Secretaria deben ser firmados o revisados por el Rector.

- **Acudiente:** Representa los acudientes de los alumnos del Colegio, que generalmente son los padres de familia. La única interacción de este actor con el sistema está representada por la revisión, por parte del acudiente, de la hoja académica del alumno.

4.2.6.4 Requisitos de Almacenamiento de Información

Gran parte de estos requisitos fueron extraídos de los formatos utilizados actualmente en la Institución, para llevar a cabo los diferentes procesos.

Dado que este es un ejercicio práctico, no se van a tener en cuenta los campos de Autor, Fuente, y Urgencia de la plantilla de requisitos de información, expuesta en la tarea 3 de la elicitación de requisitos “Identificar Requisitos de Información”.

RI-001	**Información sobre los alumnos**
Versión	1.0 (30/04/2007)
Requisitos Asociados	• **Objetivo:** Sistematizar los libros académicos y de matrículas • **Objetivo:** Listas después del proceso de matrícula • **Objetivo:** Informes y reportes • Crear hojas académicas para todos los alumnos • Permitir que cada profesor evalúe a los alumnos (logrado o en proceso) • Imprimir los logros perdidos de cada alumno • Cambiar las notas de los logros perdidos de un alumno • Almacenar las notas finales de los alumnos del Programa Regular • Imprimir informe final de notas • Permitir a los acudientes acceder a la hoja académica por Internet • Ingreso del servicio social para los estudiantes cursantes del grado Once • Pasar las calificaciones de un alumno • Cambiar la modalidad de los alumnos (presencial o semipresencial)
Descripción	El sistema deberá almacenar la información correspondiente a *los alumnos del Colegio.* En concreto:
Datos específicos	• Número de Matrícula • Código • Nombres • Apellidos • Identificación • Lugar de expedición • Fecha de nacimiento • Lugar de nacimiento • Notaría • Folio • Año • Religión que practica • Colegio de procedencia • Residencia • Teléfono residencia • Nombre del padre • Ocupación del padre • Teléfono del padre • Nombre de la madre • Ocupación de la madre • Nombre del acudiente • Parentesco

	• Ocupación del acudiente • Teléfono del acudiente • Estado: Activo, Retirado, Egresado • Programa: Tutorial, Regular • Modalidad: Presencia, Semipresencial
Importancia	Vital
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

RI-002	**Información sobre la planeación académica**
Versión	1.0 (30/04/2007)
Requisitos Asociados	• Registrar la planeación académica de cada profesor para cada materia y grado (logros) • Quitar o modificar los logros de su planeación académica • Ingresar logros de comportamiento • Digitalizar los logros de las guías para cada CLEI • Permitir modificar los logros
Descripción	El sistema deberá almacenar la información correspondiente a *la planeación académica de los profesores por periodo* En concreto:
Datos específicos	• Área • Asignatura • Profesor • Grado • Periodo • Tiempo • Fecha - Del - Al • Logros • Indicadores de procesos de aprendizaje y formación • Actividades complementarias • Semana
Importancia	Vital
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

RI- 003	**Información sobre la hoja académica (Regular)**
Versión	1.0 (30/04/2007)
Requisitos Asociados	• **Calificaciones del Programa Regular** • Almacenar las notas finales de los alumnos del Programa Regular • Imprimir informe final de notas • Generar certificados académicos
Descripción	El sistema deberá almacenar la información correspondiente a *la hoja académica de los alumnos del Programa Regular* En concreto:
Datos específicos	• Periodo • Año • Alumno • Grado • Área • Profesor • Logros específicos • Valoración • Rango Integral • Observaciones
Importancia	Vital
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

RI-004	**Información sobre la hoja académica (Tutorial)**
Versión	1.0 (30/04/2007)
Requisitos Asociados	• Almacenar las notas finales de los alumnos del Programa Tutorial • Calificaciones del Programa Tutorial
Descripción	El sistema deberá almacenar la información correspondiente a *la hoja académica de cada alumno.* En concreto:
Datos específicos	• Alumno • Área • Ciclo • Número de Guía • Logros • Seguimiento • Sustentación • Fecha • Profesor
Importancia	Vital

Estado	Validado
Estabilidad	Media
Comentarios	Ninguno

RI-005	**Información sobre la valoración escolar (Regular)**
Versión	1.0 (30/04/2007)
Requisitos Asociados	• Almacenar las notas finales de los alumnos del Programa Regular • Ingresar las nivelaciones de los alumnos (Regular) • Ver toda la información de un alumno al ingresar el nombre • Imprimir informe final de notas • Ingreso del servicio social para los estudiantes cursantes del grado Once • Generar certificados académicos • Imprimir informe final de notas
Descripción	El sistema deberá almacenar la información correspondiente a *la valoración escolar de los alumnos de la Institución* En concreto:
Datos específicos	• Folio • Código de Matrícula • Año • Alumno • Documento de Identidad • Grado • Áreas • Intensidad Horaria (semanal) • **Valoración:** - Periodo 1 - Periodo 2 - Periodo 3 - Periodo 4 - Valoración final • **Nivelaciones:** - Área - Fecha - Valoración • **Servicio Social:** - Nombre del Proyecto - Número de Horas
Importancia	Vital

Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

RI- 006	**Información sobre la valoración escolar (Tutorial)**
Versión	1.0 (30/04/2007)
Requisitos Asociados	• Almacenar las notas finales de los alumnos del Programa Tutorial • Ver toda la información de un alumno al ingresar el nombre • El sistema no debe permitir ingresar notas menores de 3.6 • Deberá generar un reporte automático cuando se cumplan todos los logros • Deberá calcular la nota parcial automáticamente en la mitad del semestre • Cambiar la modalidad de los alumnos (presencial o semipresencial) • Imprimir informe final de notas
Descripción	El sistema deberá almacenar la información correspondiente a *la valoración escolar de los alumnos del Programa Tutorial de la Institución* En concreto:
Datos específicos	• Folio • Código de Matrícula • Año • Alumno • Documento de Identidad • CLEI • Áreas • Modalidad (Presencial o Semipresencial) • **Valoración:** - Parcial - Final • **Servicio Social:** - Nombre del Proyecto - Número de Horas
Importancia	Vital
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

4.2.7 GLOSARIO DE TÉRMINOS

- **Acudiente:** Es la persona que está a cargo de la relación alumno - Colegio, generalmente son los padres. Un alumno puede tener como máximo un acudiente.
- **Año Lectivo:** Es el periodo de tiempo en el cual se están dictando clases en la Institución. Tiempo de actividad escolar. Para el Colegio Anglo Español es de 40 semanas.
- **Certificado Académico:** Documento en el cual se consignan las calificaciones de un alumno en un rango de tiempo determinado. Se especifica área y valoración.
- **Certificado de Estudio:** Documento que especifica el estado del estudiante en la Institución, grado que está cursando o cursó y su permanencia en el Colegio.
- **CLR:** Ciclo Lectivo Regular
- **CLEI:** Ciclo Lectivo Especial Integrado
- **Guías:** Documentación académica entregada a los alumnos para que estudien un conjunto de temas específicos.
- **Hoja Académica:** Formato que contiene las calificaciones de un alumno en un periodo determinado. Generalmente el formato está dividido por logros.
- **Modalidad Presencial:** Programa del Colegio que es realizado con una intensidad horaria de 20 horas semanales.
- **Modalidad Semipresencial:** Programa de la Institución que es realizado con una intensidad horaria de 10 horas semanales.
- **Nivelación:** Actividad que es realizada por los alumnos cuando no cumplieron con la valoración mínima recomendada por el Colegio para completar un logro

de una materia. Se nivela para aumentar la calificación obtenida con el fin de cumplir con los objetivos del curso.

- **Periodo:** Intervalo de tiempo, correspondiente a 2 meses, que es utilizado para dividir el año lectivo. Al final de cada uno de estos se entrega un reporte de calificaciones a los acudientes de los alumnos.
- **Planeación Académica:** Conjunto de logros definidos por los profesores para llevar a cabo en un periodo. Esta lista es revisada por la Rectora de la Institución para revisar la calidad y el verdadero cumplimiento de los objetivos académicos del Colegio.
- **Retirado:** Es un alumno que termina el contrato con la Institución de forma voluntaria o impuesta por el Colegio.
- **Reingreso:** Estado por el cual pasa un alumno si se encontraba en estado retirado y vuelve a estar activo en la Institución.
- **Servicio Social:** Requisito del Ministerio de Educación para los estudiantes de la Educación Media. Es un requisito obligatorio.
- **Sistema de calificaciones:** Forma impuesta por el Gobierno para cuantificar o cualificar el desempeño de los alumnos.
- **Tutorial:** Programa de la Institución que permite realizar dos grados académicos en un año lectivo.

4.3 DOCUMENTO DE ANÁLISIS DE REQUISITOS

4.3.1 REQUISITOS DEL SISTEMA CLASIFICADOS

4.3.1.1 Necesidades de Información

La siguiente es la lista de las necesidades de información encontradas para el sistema. Estos requisitos están definidos más ampliamente en las plantillas para requisitos de información en el documento de requisitos (ver sección 4.2.6.4).

- RI-001 Información sobre los alumnos
- RI-002 Información sobre la planeación académica
- RI-003 Información sobre la hoja académica (Regular)
- RI-004 Información sobre la hoja académica (Tutorial)
- RI-005 Información sobre la valoración escolar (Regular)
- RI-006 Información sobre la valoración escolar (Tutorial)

4.3.1.2 Necesidades Funcionales

- **OBJ-001 Sistematizar los libros académicos y de matrícula**

  RF-001 Digitalizar la información existente en los libros

  RF-002 Guardar los nuevos folios de las matrículas

  RF-003 Almacenar las notas finales de los alumnos del Programa Regular

  RF-004 Almacenar las notas finales de los alumnos del Programa Tutorial

  RF-005 Ingresar las nivelaciones de los alumnos (Regular)

RF-006 Actualizar los datos de las matrículas

RF-007 Permitir cambiar el estado de los alumnos a retirado, egresado o activo

RF-008 Permitir cambiar a un alumno de programa

RF-009 Ver toda la información de un alumno al ingresar el nombre

- **OBJ-002 Generar listas después del proceso de matrícula**

RF-010 Generar la lista de alumnos por grado en orden alfabético automáticamente

RF-011 Generar el número de alumnos por grado y por programa

RF-012 Generar la planilla de notas y asistencia para cada grado automáticamente

RF-013 Generar la lista de alumnos con su acudiente y los teléfonos

- **OBJ-003 Gestionar calificaciones del Programa Regular**

RF-014 Registrar la planeación académica de cada profesor para cada materia y grado (logros)

RF-015 Quitar o modificar los logros de su planeación académica

RF-016 Crear hojas académicas para todos los alumnos

RF-017 Permitir que cada profesor evalúe a los alumnos (logrado o en proceso)

RF-018 Convertir automáticamente de calificación cuantitativa a cualitativa y guardar las dos notas

RF-019 Imprimir los informes de calificaciones para los acudientes

RF-020 Imprimir los logros perdidos de cada alumno

RF-021 Cambiar las notas de los logros perdidos de un alumno

RF-022 Imprimir informe final de notas

RF-023 Los alumnos no pueden pasar el año con más de dos materias perdidas

RF-024 Ingresar logros de comportamiento

RF-025 Ingreso del servicio social para los estudiantes cursantes del grado Once

RF-026 Permitir a los acudientes acceder a la hoja académica por Internet

RF-027 Un profesor puede dictar varias materias

- **OBJ-004 Gestionar calificaciones del Programa Tutorial**

RF-028 Pasar las calificaciones de un alumno

RF-029 Imprimir la hoja académica general

RF-030 El sistema no debe permitir ingresar notas menores de 3.6

RF-031 Deberá generar un reporte automático cuando se cumplan todos los logros

RF-032 Deberá calcular la nota parcial automáticamente en la mitad del semestre

RF-033 Si el alumno se retira antes de terminar el CLEI, se debe calcular la nota parcial con lo que lleve

RF-034 Cambiar la modalidad de los alumnos (presencial o semipresencial)

- **OBJ-005 Generar informes y reportes**

  RF-035 Generar actas de grado

  RF-036 Generar certificados de estudio

  RF-037 Generar certificados académicos

  RF-038 Imprimir lista de graduandos

4.3.1.3 Necesidades No Funcionales

- **Portabilidad**

  RNF-001 Permitir a los acudientes acceder a la hoja académica por Internet

- **Seguridad**

  RNF-002 Hacer respaldos de la información

  RNF-003 Mostrar la información según la persona que entre al sistema

  RNF-004 Cada usuario debe tener una contraseña

  RNF-005 Los profesores no pueden ingresar notas por fuera de la Institución

- **Rendimiento**

  RNF-006 Varios profesores deben poder evaluar un alumno al mismo tiempo

- **Usabilidad**

  RNF-007 El sistema debe tener ayudas

  RNF-008 El sistema debe ser fácil de usar

  RNF-009 El sistema debe tener los colores del Colegio

RNF-010 Iconos fáciles de interpretar

RNF-011 Realizar búsquedas por nombres

RNF-012 Creación de manual de usuario

4.3.2 MATRICES DE RELACIÓN

4.3.2.1 Actor x Requisitos-C Funcionales

	Secretaria	Rector	Profesor	Acudiente
RF-001	⦿	△		
RF-002	⦿	△		
RF-003	△		○	
RF-004	△		○	
RF-005	⦿			
RF-006	⦿			
RF-007	⦿	△		
RF-008	⦿	△		
RF-009	⦿	○		
RF-010	⦿		△	
RF-011	⦿			
RF-012	⦿	○	○	
RF-013	⦿	⦿		
RF-014	△		⦿	
RF-015	△		⦿	
RF-016	△		○	
RF-017			⦿	
RF-018			△	
RF-019	⦿			
RF-020	⦿			
RF-021	⦿	○		
RF-022	⦿	○		

RF-023	△			
RF-024	○	△	◉	
RF-025	◉			
RF-026				◉
RF-027			○	
RF-028			◉	
RF-029	◉			
RF-030			△	
RF-031	△		△	
RF-032	△		△	
RF-033	△		△	
RF-034	◉	○		
RF-035	◉	○		
RF-036	◉	○		
RF-037	◉	○		
RF-038	◉	○		

4.3.2.2 Requisitos-C Funcionales x Objetivos

	OBJ-OO1	OBJ-OO2	OBJ-OO3	OBJ-OO4	OBJ-OO5
RF-001	◉		○	○	○
RF-002	◉	◉	○	○	○
RF-003	◉		◉		○
RF-004	◉		◉	○	
RF-005	○		◉		△
RF-006	○				
RF-007	○				
RF-008	◉			△	△
RF-009	○			△	△
RF-010		◉			
RF-011		○			
RF-012		◉		○	
RF-013		○			

RF-014			◉		○
RF-015			◉		○
RF-016			◉		○
RF-017			◉		○
RF-018			◉		
RF-019			◉		◉
RF-021			◉		
RF-022	○		◉		◉
RF-023	○		○		
RF-024			○		
RF-025	○		○		
RF-026			△		
RF-027			○		
RF-028				◉	○
RF-029				◉	
RF-030				△	
RF-031				◉	○
RF-032				◉	○
RF-033				○	○
RF-034				○	
RF-035					○
RF-036					◉
RF-037			○	○	◉
RF-038					○

4.3.2.3 Requisitos-C Funcionales X Requisitos De Información

	RI-001	**RI-002**	**RI-003**	**RI-004**	**RI-005**	**RI-006**
RF-001	◉	◉	◉	◉	◉	◉
RF-002	◉				△	△
RF-003			○		◉	
RF-004				○		◉
RF-005			○		◉	
RF-006	◉					

RF-007	⊙					
RF-008	⊙					
RF-009	⊙		○		○	
RF-010	⊙					
RF-011	⊙					
RF-012	⊙	○				
RF-013	⊙					
RF-014		⊙				
RF-015		⊙				
RF-016			⊙			
RF-017			⊙		⊙	
RF-018			⊙	⊙	⊙	⊙
RF-019			⊙			
RF-020		⊙				
RF-021			⊙		○	
RF-022					⊙	
RF-023					△	
RF-024		⊙				
RF-025	⊙				○	
RF-026			⊙		⊙	
RF-027		○				
RF-028				⊙		⊙
RF-029		⊙				
RF-030				⊙		
RF-031		⊙		⊙		⊙
RF-032		⊙		⊙		⊙
RF-033				⊙		⊙
RF-034	⊙					
RF-035					△	△
RF-036	○					
RF-037			○	○	○	○
RF-038	⊙					

4.3.2.4 Requisitos-C No Funcionales X Requisitos-C Funcionales

Existen varios requisitos no funcionales que no tienen relación con los requisitos funcionales, por tanto fueron excluidos de la matriz.

	RNF-001	RNF-002	RNF-003	RNF-005	RNF-006	RNF-007	RNF-011
RF-001		⊙	⊙			⊙	
RF-002		⊙	⊙			⊙	
RF-003	△	⊙	⊙				
RF-004		⊙	⊙				
RF-005		⊙	⊙	⊙			
RF-006			⊙				⊙
RF-007			⊙			△	⊙
RF-008			⊙				⊙
RF-009			⊙				⊙
RF-010			⊙				
RF-011			⊙				
RF-012			⊙				
RF-013			⊙				
RF-014		⊙	⊙			⊙	
RF-015			⊙			○	
RF-016		⊙					
RF-017			⊙	⊙	⊙	⊙	
RF-018							
RF-019			⊙				⊙
RF-020			⊙				⊙
RF-021			⊙			△	⊙
RF-022		⊙	⊙				⊙
RF-023							
RF-024		○	⊙			○	
RF-025		○	⊙			○	⊙
RF-026	⊙		⊙			△	
RF-027							
RF-028		⊙	⊙	⊙	⊙	⊙	⊙
RF-029			⊙				

RF-030							
RF-031		⊙					
RF-032		⊙					
RF-033							
RF-034			⊙				⊙
RF-035			⊙				⊙
RF-036			⊙				⊙
RF-037			⊙				⊙
RF-038			⊙				

4.3.3 RESULTADOS DEL MODELO DE KANO

En el apéndice de este documento (Sección 4.3.8) se muestra el cuestionario de atribución de importancia realizado al personal del Colegio Anglo Español.

El cuestionario fue diligenciado por seis personas, entre las que se encontraban la Rectora, la Secretaria, una asistente y tres profesores. Una vez se recogieron todos los cuestionarios, se calculó el promedio con las prioridades asignadas por los participantes a cada requisito.

A continuación se muestran los resultados obtenidos al tabular todas las respuestas.

Requisitos Funcionales	Prioridad
RF-001 Digitalizar la información existente en los libros	8,8
RF-002 Guardar los nuevos folios de las matrículas	7,6
RF-003 Almacenar las notas finales de los alumnos del Programa Regular	8,2
RF-004 Almacenar las notas finales de los alumnos del Programa Tutorial	8,2
RF-007 Permitir cambiar el estado de los alumnos a retirado, egresado o activo	6,5
RF-008 Permitir cambiar a un alumno de programa	7,1
RF-011 Generar el número de alumnos por grado y por programa	4,8
RF-012 Generar la planilla de notas y asistencia para cada grado automáticamente	8,5
RF-014 Registrar la planeación académica de cada profesor para cada materia y grado (logros)	7,5
RF-016 Crear hojas académicas para todos los alumnos	7
RF-017 Permitir que cada profesor evalúe a los alumnos (logrado o en proceso)	8,7
RF-019 Imprimir los informes de calificaciones para los acudientes	8,7
RF-020 Imprimir los logros perdidos de cada alumno	6,3
RF-021 Cambiar las notas de los logros perdidos de un alumno	8,5
RF-022 Imprimir informe final de notas	8,3
RF-025 Ingreso del servicio social para los estudiantes cursantes del grado Once	6,4
RF-026 Permitir a los acudientes acceder a la hoja académica por Internet	4,1
RF-027 Un profesor puede dictar varias materias	6,4
RF-028 Pasar las calificaciones de un alumno	8,6
RF-031 Deberá generar un reporte automático cuando se cumplan todos los logros	7,4
RF-032 Deberá calcular la nota parcial automáticamente en la mitad del semestre	7,9
RF-033 Si el alumno se retira antes de terminar el CLEI, se debe calcular la nota parcial con lo que lleve	7,9
RF-037 Generar certificados académicos	8,7

Tabla 22. *Prioridades asignadas a los Requisitos Funcionales*

Fuente: Elaboración propia

Requisitos No Funcionales	Prioridad
RNF-001 Permitir a los acudientes acceder a la hoja académica por Internet	4,1
RNF-002 Hacer respaldos de la información	9
RNF-003 Mostrar la información según la persona que entre al sistema	8,7
RNF-004 Cada usuario debe tener una contraseña	8,5
RNF-005 Los profesores no pueden ingresar notas por fuera de la Institución	5,4
RNF-006 Varios profesores deben poder evaluar un alumno al mismo tiempo	7,8
RNF-007 El sistema debe tener ayudas	6,8
RNF-008 El sistema debe ser fácil de usar	9
RNF-009 El sistema debe tener los colores del Colegio	7,7
RNF-010 Iconos fáciles de interpretar	8,3
RNF-011 Realizar búsquedas por nombres	8,7
RNF-012 Creación de manual de usuario	8,1

Tabla 23. *Prioridades asignadas a los Requisitos No Funcionales*

Fuente: Elaboración propia

4.3.4 CASA DE LA CALIDAD

La Casa de la Calidad puede verse en el Anexo 1.

4.3.4.1 Casos de Uso

Diagrama de Casos de Uso

Los casos de uso fueron agrupados según la similitud por su funcionalidad, con el fin de simplificar el diagrama de casos de uso. Los grupos son los siguientes:

- Matrículas
- Calificaciones
- Listas, Informes y Reportes
- Validación de usuarios

Validación de usuarios

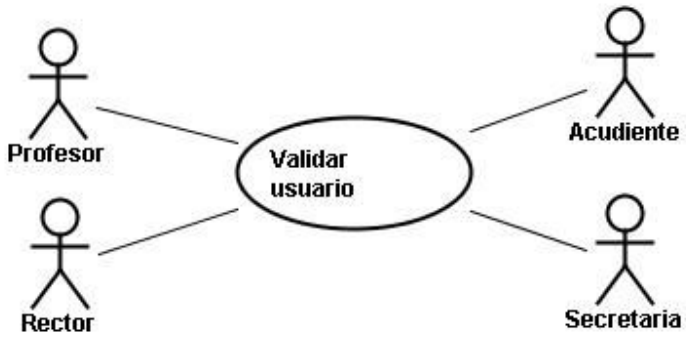


Figura 66. *Casos de uso de la funcionalidad Calificaciones*
Fuente: Elaboración propia

Calificaciones

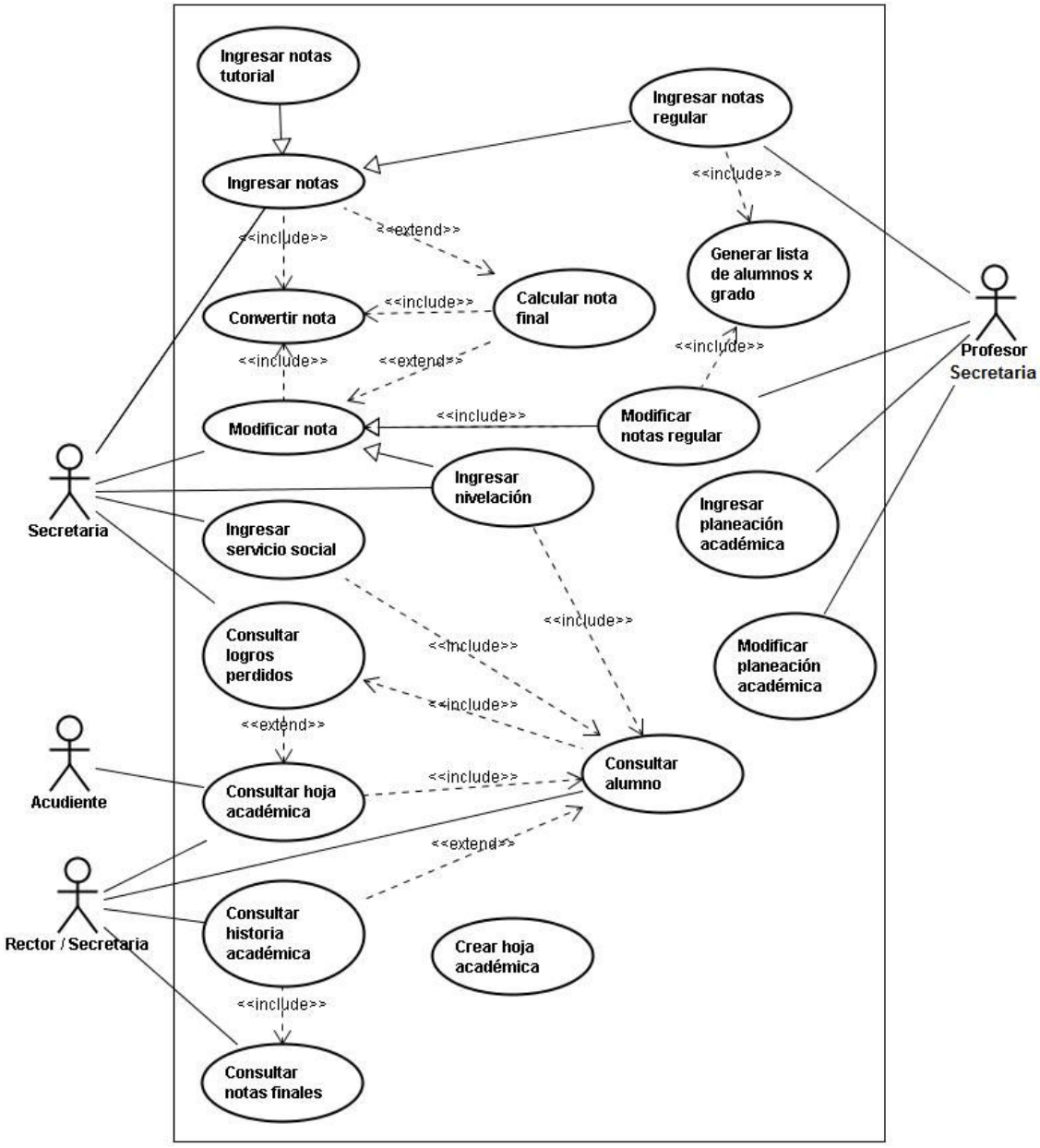


Figura 67. *Casos de uso de la funcionalidad Calificaciones*

Fuente: Elaboración propia

Listas, Informes y Reportes

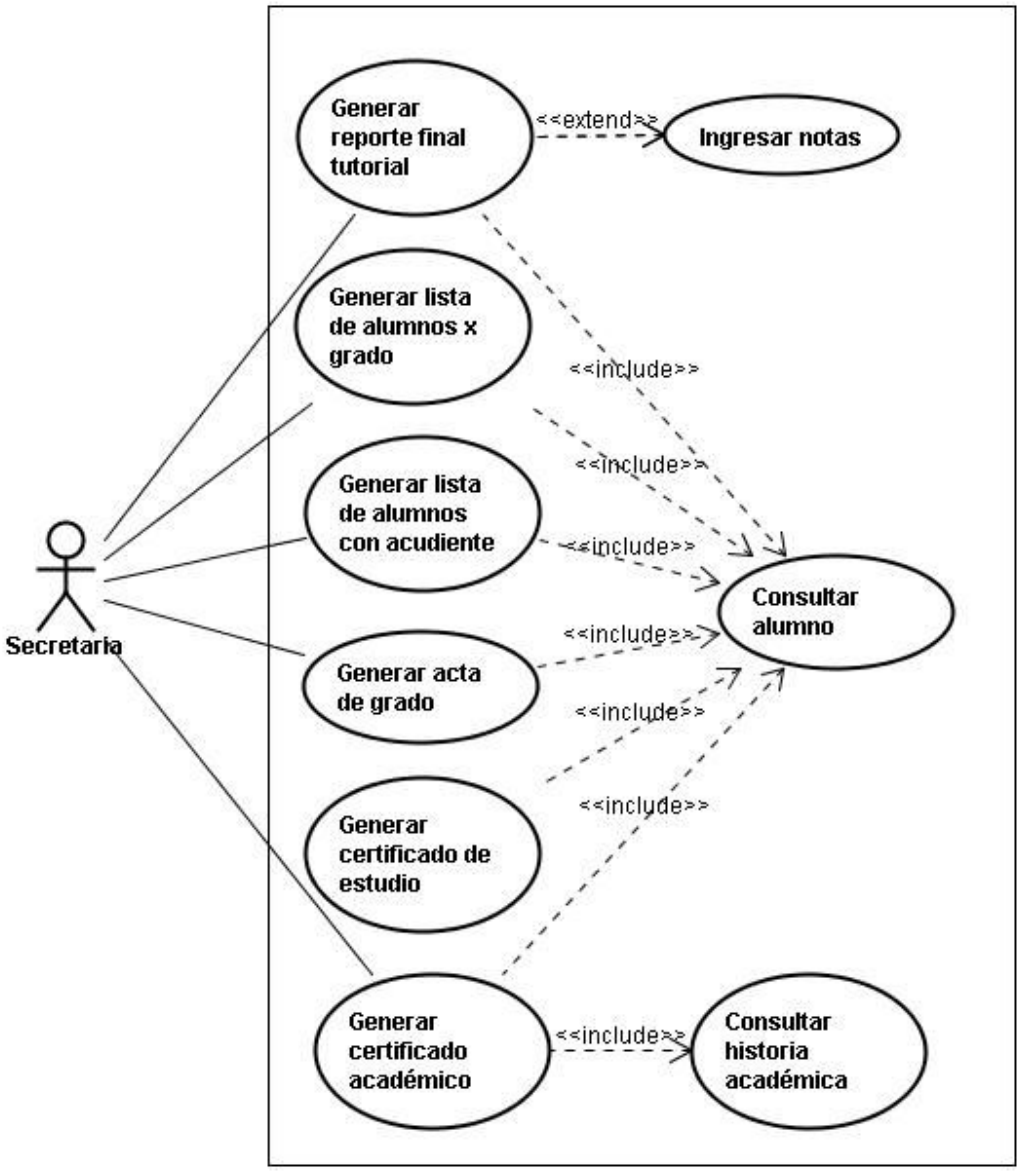


Figura 68. *Casos de uso de la funcionalidad Calificaciones*

Fuente: Elaboración propia

Matrículas

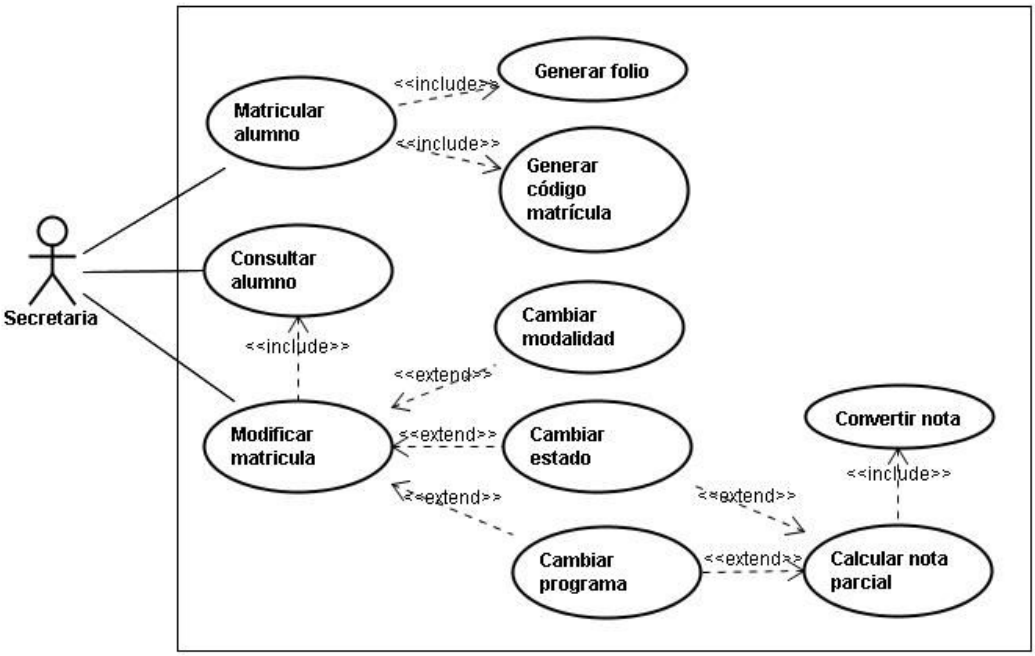


Figura 69. *Casos de uso de la funcionalidad matrículas*

Fuente: Elaboración propia

Especificación de los Casos de Uso

Dado que este es un ejercicio práctico, no se van a tener en cuenta los campos de Autor y Fecha de la plantilla para la especificación de los casos de uso, expuesta en la sección 3.4.7.3. Tampoco se van a especificar los actores que se relacionan con cada caso de uso, pues estas relaciones pueden verse claramente en los diagramas mostrados en las figuras anteriores.

Cuando un caso de uso no presente precondiciones o poscondiciones, estos campos también son suprimidos de la plantilla.

Adicionalmente, los flujos alternos sólo se describirán cuando representen una funcionalidad importante para el sistema.

CU-001	Matricular Alumno
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando una persona decide estudiar en el Colegio
Precondiciones	La persona no está matriculada y tiene toda la documentación solicitada por la Institución
Flujo Normal	**1.** El actor Secretaria solicita al sistema empezar el proceso de matrícula de un alumno **2.** El sistema utiliza los casos de uso CU-002 Generar Código Matrícula y CU-003 Generar Folio **3.** El sistema muestra a la Secretaria el código de matrícula y número de folio asignados al alumno **4.** El sistema solicita la información descrita en RI-001 Información sobre los alumnos **5.** El actor Secretaria ingresa los datos al sistema y solicita al sistema que los guarde **6.** El sistema almacena los datos ingresados por la Secretaria **7.** El sistema informa a la Secretaria que el proceso ha terminado con éxito
Poscondiciones	- La información de la matrícula es almacenada en el sistema - Se genera un contrato de matrícula para ser impreso

CU-002	Generar Código Matrícula
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso durante la realización del caso de uso CU-001 Matricular Alumno
Precondiciones	Un alumno está en proceso de matrícula
Flujo Normal	**1.** El sistema genera un código de matrícula para el alumno

CU-003	Generar Folio
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso durante la realización del caso de uso CU-001 Matricular Alumno
Precondiciones	Un alumno está en proceso de matrícula
Flujo Normal	**1.** El sistema genera un código de matrícula para el alumno

CU-004	Modificar Matrícula
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un alumno solicite la modificación de sus datos
Flujo Normal	**1.** El actor Secretaria solicita al sistema empezar el proceso de modificación de los datos de un alumno **2.** Se realiza el caso de uso CU-005 Consultar Alumno

	3. El sistema muestra todos los datos correspondientes al alumno a modificar **4.** El sistema permite a la Secretaria modificar los datos del alumno **5.** El actor Secretaria modifica los datos y solicita al sistema que los almacene **6.** El sistema modifica los datos correspondientes al alumno **7.** El sistema informa a la Secretaria que el proceso ha terminado con éxito
Flujo Alterno	**1.1** Si la Secretaria modifica la modalidad del alumno, se utiliza el caso de uso CU-006 Cambiar Modalidad **1.2** Si la Secretaria modifica el programa del alumno, se utiliza el caso de uso CU-007 Cambiar Programa **1.3** Si la Secretaria modifica el estado del alumno, se utiliza el caso de uso CU-008 Cambiar Estado
Poscondiciones	El sistema almacena los nuevos datos del alumno

CU-005	**Consultar Alumno**
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando la Secretaria o el rector desean consultar la información de un alumno o durante la realización de los siguientes casos de uso: CU-004 Modificar Matrícula, CU-019 Generar Lista de Alumnos x Grado, CU-020 Generar Lista de Alumnos con Acudiente, CU-021 Generar Acta de Grado, CU-022 Generar Certificado de Estudio, CU-015 Generar Certificado Académico, CU-027 Consultar Hoja Académica, CU-030 Ingresar Servicio Social
Flujo Normal	**1.** El sistema solicita el documento de identificación y/o el nombre completo del alumno **2.** El actor Secretaria ingresa los datos al sistema **3.** El sistema muestra al actor la información de la matrícula correspondiente al alumno y ofrece la opción de ver la historia académica
Flujo Alterno	**1.1** Los otros casos de uso utilizan este caso de uso para consultar alumno para obtener la información que necesita para poder operar **3.1** La Secretaria selecciona la opción de ver la historia académica **3.2** El sistema realiza el caso de uso CU-017 Consultar Historia Académica

CU-006	**Cambiar Modalidad**
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un alumno del tutorial desee cambiar su modalidad
Precondiciones	El cambio de modalidad debe ser solicitado por un alumno del tutorial
Flujo Normal	**1.** La Secretaria cambia la modalidad del alumno **2.** El sistema almacena la nueva modalidad

Poscondiciones	La nueva información del alumno es almacenada

CU-007	**Cambiar Programa**
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un alumno del tutorial desee cambiar de programa
Precondiciones	- El rector autorizó que un alumno del Programa Regular se cambie al Programa Tutorial - El rector autorizó que un alumno del Programa Tutorial se cambie, durante el proceso de matrícula, al Programa Regular
Flujo Normal	**1.** La Secretaria cambia el programa del alumno y selecciona la opción guardar **2.** El sistema almacena el nuevo programa
Flujo Alterno	**1.1** Si un alumno del Programa Regular se cambia al Programa Tutorial, se realiza el caso de uso CU-016 Calcular Nota Parcial para todas las materias **1.2** El sistema almacena las notas parciales en la nueva hoja académica del alumno
Poscondiciones	La nueva información del alumno es almacenada

CU-008	**Cambiar Estado**
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un alumno - Decide retirarse voluntariamente del Colegio - Es expulsado - Se gradúa - Decide reingresar al Colegio después de haber estado algún tiempo por fuera
Flujo Normal	**1.** La Secretaria cambia el estado de un alumno **2.** El sistema almacena el nuevo programa
Flujo Alterno	**1.1** Cuando se cambia el estado a retirado o expulsado, se realiza el caso de uso CU-016 Calcular Nota Parcial para todas las materias **1.2** El sistema almacena las notas parciales en la historia académica del alumno
Poscondiciones	La nueva información del alumno es almacenada

CU-009	**Ingresar Notas**
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando la Secretaria o un profesor deseen ingresar una nota al sistema

Flujo Normal	**1.** El usuario solicita al sistema empezar el proceso de calificación
Flujo Alterno	**1.1** Si el usuario selecciona calificar alumnos del tutorial, se realiza el caso de uso CU-032 Ingresar nota tutorial **1.2** Si el usuario selecciona calificar alumnos del Programa Regular, se realiza el caso de uso CU-033 Ingresar nota regular

CU-010	**Calcular Nota Final**
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso durante la realización de los siguientes casos de uso: CU-009 Ingresar Notas, CU-014 Generar Reporte Final Tutorial
Flujo Normal	**1.** El sistema calcula el promedio de las calificaciones obtenidas en cada logro **2.** Se realiza el caso de uso CU-011 Convertir Nota **3.** La nota final del alumno se almacena en su historia académica
Poscondiciones	Se almacena la nota final del alumno

CU-011	**Convertir Nota**
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso durante la realización de los siguientes casos de uso: CU-009 Ingresar Notas, CU-010 Calcular Nota Final
Flujo Normal	**1.** El sistema convierte la calificación numérica obtenida por el alumno a su respectiva calificación cualitativa (letras)

CU-012	**Modificar Nota**
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando la Secretaria o un profesor deseen modificar una nota en el sistema
Flujo Normal	**1.** El usuario solicita al sistema empezar el proceso de modificación de una nota
Flujo Alterno	**1.1** Si el usuario selecciona modificar las calificaciones de los alumnos del tutorial se realiza el caso de uso CU-025 Modificar nota tutorial **1.2** Si el usuario selecciona modificar las calificaciones de los alumnos del Programa Regular, se realiza el caso de uso CU-035 Modificar nota regular

CU-013	**Ingresar Nivelación**
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando la Secretaria desea ingresar una nivelación

Precondiciones	El alumno lleva la hoja de nivelación firmada por el profesor
Flujo Normal	**1.** La Secretaria solicita al sistema empezar el proceso de ingresar una nivelación **2.** El sistema solicita el documento de identificación y/o el nombre completo del alumno **3.** La Secretaria ingresa los datos solicitados **4.** El sistema realiza el caso de uso CU-028 Consultar Logros Perdidos **5.** La Secretaria modifica la calificación del logro nivelado por el alumno y selecciona la opción de guardar **6.** Se realiza el caso de uso CU-011 Convertir Nota **7.** El sistema almacena la nueva calificación del alumno
Flujo Alterno	**5.1** Si el alumno niveló todos los logros perdidos de una materia, se realiza el caso de uso CU-010 Calcular Nota Final
Poscondiciones	La nueva calificación del alumno es almacenada en el sistema

CU-014	**Generar Reporte Final Tutorial**
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso durante la realización del siguiente caso de uso: CU-032 Ingresar nota tutorial
Precondiciones	Se han ingresado las calificaciones para todos los logros de un alumno del Programa Tutorial
Flujo Normal	**1.** Se realiza el caso de uso CU-010 Calcular Nota Final para cada una de las materias **2.** Se genera el reporte final del tutorial, según el requisito de información RI-006 Información sobre la valoración escolar (Tutorial) y el formato requerido por el Colegio

CU-015	**Generar Certificado Académico**
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un alumno necesita un certificado académico
Flujo Normal	**1.** La Secretaria selecciona la opción de generar certificado académico **2.** El sistema solicita el documento de identificación y/o el nombre completo del alumno **3.** La Secretaria ingresa los datos solicitados **4.** Se realiza el caso de uso CU-017 Consultar Historia Académica **5.** Se genera el certificado académico según el formato requerido por el Colegio

CU-016	Calcular Nota Parcial
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso durante la realización de los siguientes casos de uso: CU-007 Cambiar Programa, CU-008 Cambiar Estado
Precondiciones	Existen notas parciales del alumno almacenadas en el sistema
Flujo Normal	**1.** El sistema calcula el promedio de las calificaciones obtenidas en cada logro almacenado en el sistema **2.** Se realiza el caso de uso CU-011 Convertir Nota

CU-017	Consultar Historia Académica
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso durante la realización de los siguientes casos de uso: CU-005 Consultar Alumno, CU-015 Generar Certificado Académico
Flujo Normal	**1.** Se realiza el caso de uso CU-018 Consultar Notas Finales para todos los grados realizados por el alumno en el Colegio

CU-018	Consultar Notas Finales
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso durante la realización del siguiente caso de uso: CU-017 Consultar Historia Académica
Flujo Normal	**1.** Se consultan las notas finales obtenidas por el alumno en un año determinado

CU-019	Generar Lista de Alumnos x Grado
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando la Secretaria necesita la lista de alumnos de un grado
Flujo Normal	**1.** La Secretaria selecciona la opción de generar lista de alumnos **2.** El sistema le solicita ingresar un rango de grados **3.** La Secretaria ingresa los datos solicitados **4.** El sistema genera las listas de los alumnos de los grados solicitados en orden alfabético

CU-020	Generar Lista de Alumnos con Acudiente
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando la Secretaria necesita la lista de alumnos con sus acudientes
Flujo Normal	**1.** La Secretaria selecciona la opción de generar lista de alumnos con acudiente **2.** El sistema genera las listas de los alumnos con sus acudientes y los

	teléfonos de localización

CU-021	**Generar Acta de Grado**
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el rector o un alumno le solicita a la Secretaria las actas de grado
Flujo Normal	**1.** La Secretaria selecciona la opción de generar actas de grado **2.** El sistema pregunta si es para un alumno o para todos los graduandos **3.** La Secretaria selecciona la opción de actas para todos los graduandos **4.** El sistema genera las actas de grado
Flujo Alterno	**3.1** Si selecciona acta para un alumno, el sistema solicita el nombre o documento de identidad del alumno **3.2** La Secretaria ingresa la información solicitada **3.3** El sistema genera el acta de grado del alumno

CU-022	**Generar Certificado de Estudio**
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el rector o un alumno le solicita a la Secretaria las actas de grado
Precondiciones	El alumno está activo en la Institución
Flujo Normal	**1.** La Secretaria selecciona la opción de generar certificado de estudio **2.** El sistema solicita el documento de identificación y/o el nombre completo del alumno **3.** La Secretaria ingresa los datos solicitados **4.** El sistema genera el certificado de estudio del alumno según el formato utilizado por el Colegio

CU-023	**Ingresar Planeación Académica**
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un profesor va a ingresar su planeación académica
Flujo Normal	**1.** El profesor solicita al sistema empezar el proceso de planeación académica **2.** El sistema lista las materias asignadas al profesor **3.** El profesor selecciona una materia **4.** El sistema solicita los datos del requisito de información RI- 002 Información sobre la planeación académica **5.** El profesor ingresa los logros y selecciona la opción de guardar **6.** El sistema almacena los logros
Poscondiciones	Los logros ingresados por el profesor son almacenados en el sistema

CU-024	Modificar Planeación Académica
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un profesor desea modificar su planeación académica
Flujo Normal	**1.** El profesor solicita al sistema empezar el proceso de modificación de la planeación académica **2.** El sistema lista las materias asignadas al profesor **3.** El profesor selecciona una materia **4.** El sistema lista los logros ingresados previamente por el profesor para el periodo actual **5.** El usuario modifica, elimina o ingresa logros y selecciona la opción de guardar **6.** El sistema almacena los logros
Poscondiciones	Los logros ingresados por el profesor son almacenados en el sistema

CU-025	Modificar notas regular
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un profesor desea modificar las notas ingresadas previamente al sistema
Flujo Normal	**1.** El profesor solicita al sistema empezar el proceso de modificación de las calificaciones del Programa Regular **2.** El sistema lista las materias asignadas al profesor **3.** El profesor selecciona una materia **4.** El sistema muestra al usuario la planilla de calificaciones y sólo le permite modificar las notas del periodo actual **5.** El usuario modifica o ingresa las calificaciones de los alumnos y selecciona la opción de guardar **6.** Se realiza el caso de uso CU-011 Convertir Nota **7.** El sistema almacena las calificaciones e informa al usuario que la operación se realizó correctamente
Flujo Alterno	**5.1** Cuando se ingresan las calificaciones de todos los logros del periodo se realiza el caso de uso CU-010 Calcular Nota Final
Poscondiciones	Las calificaciones de los alumno son almacenadas en el sistema

CU-026	Crear Hoja Académica
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando la Secretaria necesita imprimir la hoja académica para un grado del tutorial
Precondiciones	Los logros para el grado han sido previamente ingresados al sistema
Flujo Normal	**1.** La Secretaria solicita al sistema empezar el proceso de crear hoja

	académica para el tutorial **2.** El sistema solicita un rango de grados **3.** La Secretaria ingresa los datos solicitados **4.** El sistema genera la(s) hoja(s) académica(s) para cada grado, con base en los logros ingresados previamente

CU-027	**Consultar Hoja Académica**
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando: - Un acudiente ingresa a través de Internet para revisar la información académica de un alumno - Se va a realizar la entrega de calificaciones a los padres de familia y acudientes
Flujo Normal	**1.** El usuario solicita al sistema empezar el proceso de consultar la hoja académica de un alumno **2.** El sistema solicita el nombre y/o documento de identidad del alumno **3.** La Secretaria ingresa los datos solicitados **4.** El sistema genera la hoja académica para el alumno

CU-028	**Consultar Logros Perdidos**
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso durante la realización del siguiente caso de uso: CU-013 Ingresar Nivelación
Flujo Normal	**1.** El sistema revisa los logros con calificación por debajo de 3.6 y los lista

CU-029	**Ingresar ayuda**
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario accede a la ayuda del sistema
Flujo Normal	**1.** El usuario selecciona la opción de ayuda del sistema **2.** El sistema genera la ayuda dependiendo del caso de uso que se esté utilizando

CU-030	**Ingresar Servicio Social**
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un alumno tenga la información requerida del servicio social que está realizando
Flujo Normal	**1.** La Secretaria solicita al sistema empezar el proceso de ingreso del servicio social

	2. El sistema solicita el nombre y/o documento de identidad del alumno **3.** La Secretaria ingresa los datos solicitados **4.** El sistema solicita la información del servicio social **5.** La Secretaria ingresa los datos solicitados y selecciona la opción de guardar **6.** El sistema almacena la información del servicio social del alumno
Flujo Alterno	El sistema almacena la información del servicio social del alumno

CU-031	**Validar Usuario**
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso siempre que un usuario vaya a ingresar al sistema
Flujo Normal	**1.** El sistema solicita nombre se usuario y contraseña **2.** El usuario ingresa los datos solicitados **3.** El sistema valida que el usuario exista
Poscondiciones	El sistema retorna el perfil asignado al usuario que ingresó al sistema

CU-032	**Ingresar notas tutorial**
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando la Secretaria desea ingresar al sistema las calificaciones de un alumno del Programa Tutorial
Flujo Normal	**1.** La Secretaria solicita al sistema empezar el proceso de calificación del tutorial **2.** El sistema le solicita al usuario ingresar el nombre o documento de identidad del alumno **3.** La Secretaria ingresa la información solicitada **4.** El sistema muestra la plantilla de calificación para el alumno con las notas que han sido ingresadas hasta el momento **5.** La Secretaria ingresa las calificaciones y selecciona la opción de guardar **6.** Se realiza el caso de uso CU-011 Convertir Nota **7.** El sistema almacena las calificaciones e informa al usuario que la operación se realizó correctamente
Flujo Alterno	**5.1** Si el alumno cumplió todos los logros de un grado el sistema alerta al usuario sobre este hecho y se realiza el caso de uso CU-014 Generar Reporte Final Tutorial
Poscondiciones	Las calificaciones del alumno son almacenadas en el sistema

CU-033	**Ingresar notas regular**
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un profesor o la Secretaria deseen ingresar al sistema las

	calificaciones de un alumno del Programa Regular
Flujo Normal	**1.** El usuario solicita al sistema empezar el proceso de calificación del Programa Regular **2.** El sistema lista las materias asignadas al usuario **3.** El usuario selecciona una materia **4.** El sistema muestra al usuario la planilla de calificaciones **5.** El usuario ingresa las calificaciones de los alumnos y selecciona la opción de guardar **6.** Se realiza el caso de uso CU-011 Convertir Nota **7.** El sistema almacena las calificaciones e informa al usuario que la operación se realizó correctamente
Flujo Alterno	**5.1** Cuando se ingresan las calificaciones de todos los logros del periodo se realiza el caso de uso CU-010 Calcular Nota Final
Poscondiciones	Las calificaciones de los alumno son almacenadas en el sistema

4.3.5 CONFLICTOS

CFL-001	**Ingreso de planeación académica**
Versión	1.0 (2007)
Autores	Carlos Orozco Alejandra Maya Carolina Arroyave
Fuentes	Nazly Medina María Cristina Valencia
Descripción	Se encontró, en medio del análisis, que las necesidades “Registrar la planeación académica de cada profesor para cada materia y grado (logros)” y “Digitalizar los logros de las guías para cada CLEI” son muy similares.
Alternativas	- Eliminar alguna de las dos necesidades - Descartar del análisis alguna de las dos necesidades
Solución	Fue utilizada la segunda opción, por tanto fue descartada una de las dos necesidades del análisis. No fue eliminada porque se cree que la necesidad debe quedar como parte del documento de requisitos.
Importancia	Media
Urgencia	No
Estado	Validado
Comentarios	Ninguno

CFL-002	**Generar reporte automático**

Versión	1.0 (2007)
Autores	Carlos Orozco Alejandra Maya Carolina Arroyave
Fuentes	Nazly Medina María Cristina Valencia
Descripción	En el análisis no se pudo llegar a un consenso sobre cuál era la verdadera necesidad de los clientes y usuarios, con respecto a la siguiente necesidad "Deberá generar un reporte automático cuando se cumplan todos los logros".
Alternativas	- Hablar con el cliente y definir concretamente qué es lo que se necesita - Descartar la necesidad del análisis
Solución	Se hizo una llamada al cliente para que definiera el requisito de una forma más clara y así poder realizar el análisis correcto. El cliente dijo que este reporte no es escrito, es una notificación que debe hacer el sistema para indicarle a la Secretaria que el alumno ya terminó todos los logros de su CLEI.
Importancia	Media
Urgencia	No
Estado	Validado
Comentarios	Ninguno

CFL-003	**Imprimir informe final**
Versión	1.0 (2007)
Autores	Carlos Orozco Alejandra Maya Carolina Arroyave
Fuentes	Nazly Medina María Cristina Valencia
Descripción	Se encontró, en medio del análisis, que existían dos necesidades "Imprimir informe final de notas" en objetivos diferentes y se necesitó conocer para cual de los objetivos era más relevante tener esta necesidad, con el fin de no incidir negativamente en la construcción de la Casa de la Calidad (QFD).
Alternativas	- Hablar al cliente y saber para cuál de los objetivos era más relevante - Descartar la necesidad a criterio de los ingenieros de requisitos
Solución	Se realizó una llamada al cliente y se pudo concluir que la necesidad era más relevante dejarla en el objetivo "Gestionar calificaciones del Programa Regular", dado que es en este donde se realiza esta operación más repetidamente.
Importancia	Alta
Urgencia	No
Estado	Validado
Comentarios	Ninguno

4.3.6 APÉNDICES

4.3.6.1 Cuestionario de Atribución de Importancia

Requisitos Funcionales	Para cada necesidad, marque con un círculo el número de la escala que mejor refleje su opinión								
	Nada importante		Algo importante		Normal	Importante		Muy importante	
RF-001 Digitalizar la información existente en los libros	1	2	3	4	5	6	7	8	9
RF-002 Guardar los nuevos folios de las matrículas	1	2	3	4	5	6	7	8	9
RF-003 Almacenar las notas finales de los alumnos del Programa Regular	1	2	3	4	5	6	7	8	9
RF-004 Almacenar las notas finales de los alumnos del Programa Tutorial	1	2	3	4	5	6	7	8	9
RF-007 Permitir cambiar el estado de los alumnos a retirado, egresado o activo	1	2	3	4	5	6	7	8	9
RF-008 Permitir cambiar a un alumno de programa	1	2	3	4	5	6	7	8	9
RF-011 Generar el número de alumnos por grado y por programa	1	2	3	4	5	6	7	8	9
RF-012 Generar la planilla de notas y asistencia para cada grado automáticamente	1	2	3	4	5	6	7	8	9
RF-014 Registrar la planeación académica de cada profesor para cada materia y grado (logros)	1	2	3	4	5	6	7	8	9
RF-016 Crear hojas académicas para todos los alumnos	1	2	3	4	5	6	7	8	9
RF-017 Permitir que cada profesor evalúe a los alumnos (logrado o en proceso)	1	2	3	4	5	6	7	8	9
RF-019 Imprimir los informes de calificaciones para los acudientes	1	2	3	4	5	6	7	8	9
RF-020 Imprimir los logros perdidos de cada alumno	1	2	3	4	5	6	7	8	9
RF-021 Cambiar las notas de los logros perdidos de un alumno	1	2	3	4	5	6	7	8	9
RF-022 Imprimir informe final de notas	1	2	3	4	5	6	7	8	9
RF-025 Ingreso del servicio social para los estudiantes cursantes del grado Once	1	2	3	4	5	6	7	8	9
RF-026 Permitir a los acudientes acceder a la hoja académica por Internet	1	2	3	4	5	6	7	8	9
RF-027 Un profesor puede dictar varias materias	1	2	3	4	5	6	7	8	9
RF-028 Pasar las calificaciones de un alumno	1	2	3	4	5	6	7	8	9
RF-031 Deberá generar un reporte automático cuando se cumplan todos los logros	1	2	3	4	5	6	7	8	9
RF-032 Deberá calcular la nota parcial automáticamente en la mitad del semestre	1	2	3	4	5	6	7	8	9
RF-033 Si el alumno se retira antes de terminar el CLEI, se debe calcular la nota parcial con lo que lleve	1	2	3	4	5	6	7	8	9
RF-037 Generar certificados académicos	1	2	3	4	5	6	7	8	9
Requisitos No Funcionales									
RNF-001 Permitir a los acudientes acceder a la hoja académica por Internet	1	2	3	4	5	6	7	8	9
RNF-002 Hacer respaldos de la información	1	2	3	4	5	6	7	8	9
RNF-003 Mostrar la información según la persona que entre al sistema	1	2	3	4	5	6	7	8	9
RNF-004 Cada usuario debe tener una contraseña	1	2	3	4	5	6	7	8	9
RNF-005 Los profesores no pueden ingresar notas por fuera de la Institución	1	2	3	4	5	6	7	8	9
RNF-006 Varios profesores deben poder evaluar un alumno al mismo tiempo	1	2	3	4	5	6	7	8	9
RNF-007 El sistema debe tener ayudas	1	2	3	4	5	6	7	8	9
RNF-008 El sistema debe ser fácil de usar	1	2	3	4	5	6	7	8	9
RNF-009 El sistema debe tener los colores del Colegio	1	2	3	4	5	6	7	8	9
RNF-010 Iconos fáciles de interpretar	1	2	3	4	5	6	7	8	9
RNF-011 Realizar búsquedas por nombres	1	2	3	4	5	6	7	8	9
RNF-012 Creación de manual de usuario	1	2	3	4	5	6	7	8	9

Figura 70. *Cuestionario de atribución de importancia*

Fuente: Elaboración propia

TRABAJO FUTURO

Dentro de un trabajo de investigación es importante identificar las líneas de trabajo para dar continuidad al esfuerzo invertido. Por esto, esta sección pretende mostrar el trabajo futuro que es necesario realizar para seguir avanzando en la construcción de una metodología completa de desarrollo de software basada en QFD, teniendo como eje principal la Casa de la Calidad.

En la metodología QFD, la construcción de la casa se divide en cuatro fases:

- Fase 1: Planeación del producto: Casa de la Calidad.
- Fase 2: Diseño del producto: despliegue de las partes.
- Fase 3: Planeación del proceso.
- Fase 4: Control del proceso (cartas de control de calidad).

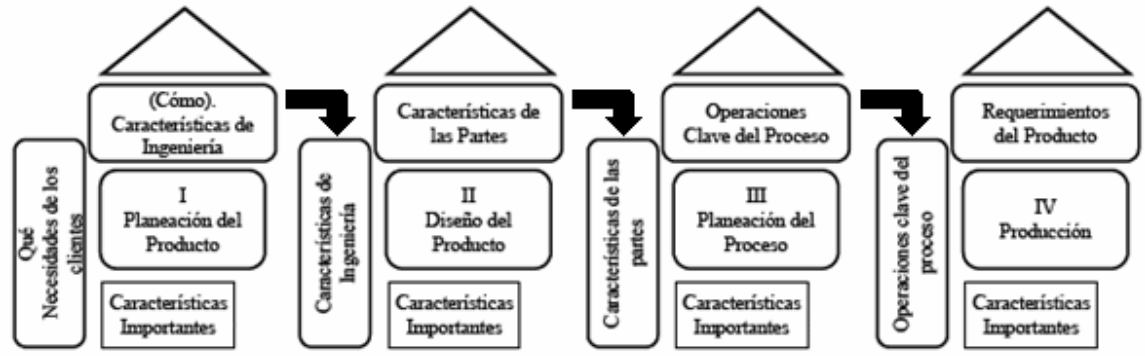


Figura 71. *Las cuatro fases de QFD*

En esta tesis se abarcó la primera fase del proceso QFD, la planeación del producto. Según la metodología QFD, la planeación del producto consiste en convertir los requisitos de los clientes y usuarios (la voz del cliente) en características técnicas que permitan desarrollar un producto que cumpla las necesidades que los clientes brindaron. Para este caso, dichas características técnicas son casos de uso, los cuales permitirán la comunicación entre todos los participantes del proyecto de desarrollo.

Como es sabido, el ciclo de vida del software comprende un conjunto de etapas, las cuales generan un resultado utilizado en las etapas posteriores del desarrollo de software. En el desarrollo del proyecto de grado se completó la primera etapa del ciclo de vida, la ingeniería de requisitos. Como trabajo futuro, se plantea la posibilidad de ampliar la metodología, de tal forma que se siga utilizando la Casa de la Calidad y la metodología QFD en las etapas posteriores del desarrollo: análisis y diseño, implementación y pruebas.

En la figura 72, se ilustra la forma en la cual se podría plantear una metodología completa para el desarrollo de software utilizando QFD.

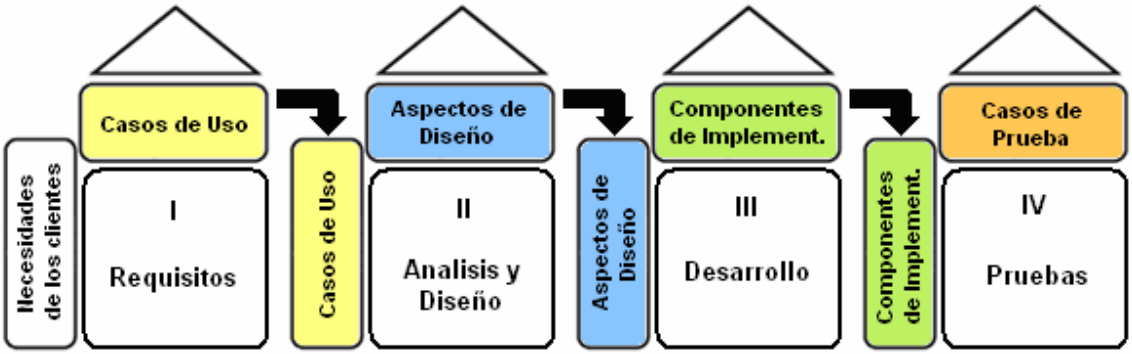


Figura 72. *Fases del QFD aplicadas a software*

En la segunda etapa se podrían traducir los casos de uso en aspectos de diseño, como: Clases, componentes, interfaces, modelos entidad relación, diagramas de secuencia, de actividades, entre otros. De forma similar, en la etapa de desarrollo los componentes de diseño enunciados anteriormente, se podrían mapear a componentes de implementación, tales como: código fuente, ejecutables, librerías, documentos, etc. Finalmente, durante la fase de pruebas se utilizan los aspectos de diseño y los casos de uso realizados en las primeras etapas del desarrollo, para crear los casos de prueba y posteriormente ejecutarlos sobre los componentes de implementación.

CONCLUSIONES

- La utilización de la “voz del cliente” en el análisis de los requisitos, las relaciones entre los casos de uso y las estrategias para satisfacer requisitos no funcionales y las atribuciones de importancia brindadas por los clientes y usuarios, son características particulares de la propuesta desarrollada, las cuales complementan las metodologías tradicionales de ingeniería de requisitos de una forma significativa, dado que resuelven algunos problemas que normalmente no son contemplados ni profundizados. Estos aspectos de la metodología permiten identificar oportunamente posibles conflictos y definiciones incompletas de las necesidades, logrando, de esta forma, disminuir costos, tiempo y esfuerzo en los proyectos. Así mismo, dichos aspectos facilitan que el producto final cumpla con las especificaciones requeridas por los clientes y usuarios, aumentando su nivel de satisfacción.

- La conformación de un equipo de personas con habilidades y perspectivas técnicas diferentes, es decir, desarrolladores, testers, analistas e ingenieros de requisitos, conduce a una definición integral de las características técnicas que deben ser incluidas en el producto software a construir, dado que cada uno de los participantes puede aportar una posible solución para el problema, desde su punto de vista, y, además, puede refutar los aportes de los demás participantes, según la viabilidad de la solución planteada.

- Teniendo en cuenta que la voz del cliente es el insumo más importante dentro de la propuesta metodológica, es necesario que los ingenieros de requisitos que participan en el proceso de elicitación tengan adecuadas habilidades de comunicación y escucha, de tal manera que faciliten la captura de todos los aspectos importantes que deben ser considerados en el producto de software, con el fin de satisfacer completamente las expectativas de los clientes y

usuarios. Un ingeniero que no posea estas habilidades, muy posiblemente generará problemas en lugar de aportar soluciones.

- El desarrollo de nuevas técnicas y metodologías que vayan en pro del mejoramiento de cualquier rama de la ingeniería es un reto que los estudiantes de las diferentes universidades deben enfrentar, de tal forma que contribuyan a que en el futuro, las empresas a las cuales ellos estén vinculados, puedan emprender estrategias enfocadas en la innovación, especialmente de los productos y servicios ofrecidos, lo cual les permitirá generar ventajas competitivas en el mercado.

- Actualmente, las empresas deben ser conscientes de que las necesidades y expectativas de los clientes y usuarios constituyen los aspectos más importantes a considerar al momento de realizar los diferentes negocios. Las estrategias para atraer a los clientes potenciales e influir en su decisión de compra, deben estar dirigidas a la generación de valor, con el fin de que los productos sean realmente atractivos para el público objetivo. La metodología propuesta es una herramienta que permite conocer efectivamente las expectativas de los clientes y usuarios, al plasmarlas como objetivos a cumplir desde el inicio del proceso de ingeniería de requisitos, asegurando el éxito del proyecto de construcción de software.

- Podría decirse que los problemas en la mayoría de las técnicas de elicitación se derivan de su inexactitud, al realizar una abstracción inmediata a características técnicas durante este proceso, lo cual conduce a una especificación de requisitos que no concuerda con lo que realmente quiere y necesita el cliente. La propuesta metodológica planteada contrarresta dicha inexactitud, al conservar las necesidades tal y como fueron expresadas por los clientes y usuarios, durante todo el proceso de ingeniería de requisitos.

- Uno de los aportes más significativos de la propuesta metodológica consiste en la definición de la **matriz de correlación.** Este componente de la Casa de la Calidad es, probablemente, uno de los que menos se profundiza y, en muchos casos, es omitido, debido a que incrementa el tiempo de seguimiento de la metodología. Sin embargo, la información que allí se relaciona, es crítica y fundamental para conocer claramente cómo afectan los cambios realizados sobre un requisito, en los demás requisitos e identificar todas las interacciones existentes entre ellos. El verdadero valor de esta matriz radica en que se está haciendo un análisis minucioso de las relaciones entre los casos de uso, las estrategias para darle cumplimiento a los requisitos no funcionales y las relaciones entre los requisitos funcionales y no funcionales. El beneficio de construir esta matriz se halla en que rápidamente será posible identificar el grado en el cual las características técnicas definidas serán consecuentes con las exigencias y necesidades del cliente/usuario. Así mismo, le proporcionará al equipo de desarrollo habilidad para identificar requisitos que presenten conflictos.

- Un factor crítico de éxito en la aplicación de la metodología es el compromiso que deben adquirir los clientes y usuarios para participar activamente durante todas las etapas del proceso de ingeniería de requisitos, dado que la contribución de cada una de las personas involucradas en el desarrollo del sistema software es crucial para el éxito del proyecto.

- La correcta identificación de los requisitos en un proyecto de desarrollo de software es fundamental en la creación de un sistema, dado que de esta labor dependen las etapas posteriores de implementación, pruebas, integración, operación y mantenimiento. En el caso práctico realizado en el Colegio Anglo Español, se pudo observar que con la utilización de la propuesta metodológica el equipo de desarrollo logró obtener un conjunto de requisitos consistente, el cual representa una base sólida para continuar con las etapas mencionadas.

- En el transcurso de la elicitación de requisitos realizada en el Colegio Anglo Español, se pudo concluir que cuando los clientes y usuarios no están familiarizados con ningún sistema de información, se hace muy difícil que tomen la iniciativa en la búsqueda de soluciones a sus problemas por medio de un producto de software, debido a que se han acostumbrado a realizar labores manuales y no encuentran en ello mayores inconvenientes. Por esta razón, el ingeniero de requisitos debe tomar una posición más activa en las sesiones de lluvia de ideas, explicando funciones que podría ofrecer el sistema y, de esta manera, promover la participación de todas las personas, teniendo especial cuidado de no influenciar las soluciones brindadas por los participantes.

- Durante la elaboración de los diagramas de árbol en las sesiones de elicitación, se pudo apreciar que cuando se está especificando el tercer nivel de jerarquía, generalmente, los clientes y usuarios definen restricciones que debe tener el sistema, muy útiles para los detalles de implementación.

- La aplicación de una nueva técnica requiere que aquellos que la van a usar entiendan muy claramente cuáles son los beneficios y limitaciones de la misma. Luego de haber definido completamente la metodología y haberla aplicado, paso a paso, en un caso real, se ha podido identificar que QFD en el proceso de ingeniería de requisitos, ofrece los siguientes beneficios:

  - Una de las grandes ventajas de la metodología es su flexibilidad, ya que ha permitido la incorporación de varias técnicas, métodos y modelos, facilitando, por ejemplo, la etapa de elicitación en donde los resultados obtenidos de las diferentes técnicas utilizadas han podido integrarse y complementarse, de tal forma que las necesidades vitales para el desarrollo del proyecto se identifiquen claramente.

- La técnica de priorización de las necesidades por medio de cuestionarios de Kano, ayuda al equipo desarrollador a enfocarse en aquellas características del producto que tienen un impacto positivo en la satisfacción del cliente.

- La utilización de los prototipos, servirá como medio de retroalimentación para las etapas posteriores de diseño y desarrollo, puesto que el ingeniero de requisitos podrá identificar los aspectos de usabilidad más relevantes para los clientes y usuarios, que no fueron expresados desde el inicio del proyecto, pero que deberán ser atendidos al momento de la implementación.

- El cumplimiento de algunas metas y prácticas de un modelo formal de calidad como CMMI, es un buen indicio de que las tareas que se van a llevar a cabo, apuntarán hacia el logro de los objetivos que giran en torno a la satisfacción del cliente y, por ende, al éxito del proyecto.

❖ Así mismo, se encontraron algunas falencias en la metodología. Su corrección podría agregarse al trabajo futuro, propuesto anteriormente. A continuación son descritas las falencias encontradas:

- La construcción de la Casa de la Calidad en las etapas posteriores, propuesta en el trabajo futuro, podría llegar a volverse compleja, pues es posible que las entradas para cada una de las etapas se incrementen sustancialmente. Por esta razón se requiere de un análisis profundo que permita continuar con la metodología, definiendo un modelo simple, con el que se sigan aprovechando las ventajas de QFD.

- Aunque existen herramientas que facilitan la construcción de la Casa de la Calidad, ninguna se acomoda completamente a las adecuaciones que se hicieron de ella, especialmente, en la matriz de correlación.

- Es posible que con la utilización de esta propuesta, los clientes y usuarios deban invertir más tiempo y dedicación durante el proceso de ingeniería de requisitos, en comparación con los utilizados en otras metodologías. Este incremento podría estar dado por la identificación de la fuerza de las relaciones en las matrices de relación y por el diligenciamiento de los cuestionarios de Kano. Sin embargo, con el desarrollo de estas actividades se obtiene información muy significativa para la implementación de un producto que satisfaga, de la mejor manera, las necesidades de los clientes y usuarios.

BIBLIOGRAFÍA

[Akao, 1997] Akao, Yoji. *QFD: Past, Present, and Future*. Asahi University

[Alvarado, 2004] Alvarado Márquez, José Antonio. *Mejora del proceso de desarrollo de software en México, un estudio exploratorio.* Instituto Tecnológico y de Estudios Superiores de Monterrey. Monterrey, julio de 2004.

[Anaya y Letelier, 2000] Anaya Victor y Letelier Patricio. *Trazabilidad de requisitos en proyectos basados en UML.* Universidad Politécnica de Valencia, 2000.

[Asoc. Latin, 2005] Asociación Latinoamericana de QFD 2005. *Herramientas interactivas.* Disponible en Web: <http://www.qfdlat.com/Herramientas_QFD/herramientas_qfd.html#MRelaciones>

[Bahamonde y Rossel, 2003] Bahamonde, José Manuel y Rossel Richard. *Un acercamiento a la ingeniería de requisitos*. Universidad Técnica Federico Santa María, Noviembre de 2003.

[Berezin, 1999] Berezin, Tanya. *Writing a Software Requirements Document.* Junio de 1999.

[Berrios, 2000] Berrios, Nelson. *Mejoramiento Continuo de Procesos: Una estrategia de competencia basada en las personas*. Revista Bit, Junio 2000. Disponible en Web: <http://www.revistabit.cl/pdf/18articulo3.pdf>

[Boehm, 1981] Barry W. Boehm. *Software Engineering Economics.* New Jersey: Prentice-Hall,1981.

[Borland, 2005] *Mitigating Risk With effective Requirements Engineering*. Borland, April 2005.

[Contreras y Laciar, 2004] Contreras Jaquelina Pamela, Laciar Verónica Lorena. *Procesos unificados para modelar sistemas de educación a distancia.* San Juan, Argentina 2004.

[Cook, 1997] Cook, T.D. Reichardt. *Métodos cualitativos y cuantitativos en la investigación evaluativa*. Ediciones Morata, España, 1997.

[Durán, 2000] Durán Toro, Amador. *Un Entorno Metodológico de Ingeniería de Requisitos para Sistemas de Información*. Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla, 2000.

[Goetsch y Davis, 1994] Goetsch, David L. y Davis, Stanley. *Introduction to Total Quality*. Macmillan College Publishing Company, Inc. Editorial Merrill, 1994.

[González Bosch et al., 2002] González Bosch, Verónica y Tamayo, Francisco, Blitz. *QFD: Un Vistazo Relámpago al Poder del QFD*. Asociación Latinoamericana de QFD: Formato Electrónico, 2002. Disponible en Web: <http://www.qfdlat.com/Herramientas_QFD/herramientas_qfd.html#BlitzQFD>

[Granollers, 2004] Granollers i Saltiveri, Toni. *MPIu+a. Una metodología que integra la ingeniería del software, la interacción persona-ordenador y la accesibilidad en el contexto de equipos de desarrollo multidisciplinares.* Julio de 2004. Disponible en Web: <http://griho.udl.es/castella/equip/invest/granollers.html>

[Hagg, 1992] Hagg, Stephen Eugene. *A field study of the use of quality function deployment (QFD) as applied to software development.* The University of Texas, Arlington 1992.

[Harty, 2001] Harty, David. *Quality Function Deployment. An Overview of QFD and its Applications to Software Engineering.* 2001.

[Hernández] Hernández Jesús. *Ciclo de vida del software: Gestión de Requisitos*

[Hernández, 2003] Hernández del Ángel, José Luis. *Modelo conceptual para asegurar el cumplimiento de ISO/TS 16949 a través del uso del QFD (quality function deployment)*. Instituto Tecnológico y de Estudios Superiores de Monterrey. Monterrey, Diciembre 2003.

[Herrero, 2005] Herrero Obeso, Miguel. *Fundamentos de Ingeniería del Software.* Febrero de 2005.

[ITESM, 1998] Centro de calidad ITESM. *QFD Quality Function Deployment.* 1988

[Jacobson et al., 2000] Jacobson Ivar, Booch Grady y Rumbaugh James. *El proceso unificado de desarrollo de software.* Pearson Education. Madrid, 2000.

[Jap. Society] *The Japanese Society for Quality Control – JSQC*. Disponible en Web: <http://www.jsqc.org/en/about_us/about_us.html>

[Karlsson, 1996] Karlsson, Joachim. *Managing software requirements using Quality function deployment*. Linköping, Suecia. Mayo, 1996

[Lamia, 1995] Lamia W.M. *Integrating QFD with Object Oriented Software Design Methodologies.* Pp. 417-434. 1995.

[León, 2005] León Duarte, Jaime Alfonso. *Metodología para la detección de requisitos subjetivos en el diseño de producto*. 2005

[López, 2003] López Solís, Rosa Idolina. *Modelo para medir la madurez de procesos y funciones del help desk.* Instituto Tecnológico y de Estudios Superiores de Monterrey. Monterrey, diciembre de 2003.

[López, 2004] Juan Antonio. *Ingeniería del Software – Prototipado*. Universidad de Murcia, Facultado de Informática y Sistemas.

[López, 2005] López Rabadán José Jaime. *Análisis para el Proceso de Diseño de Sistemas.* México DF, Agosto de 2005. Disponible en Web: **<**http://www.cs.cinvestav.mx/Estudiantes/TesisGraduados/2005/tesisJoseJaime.pdf>

[Martin, 1984] Martin, James. *An Information Systems Manifesto. N*ew York: Prentice-Hall, 1984.

[Mayol, 2003] Mayol Sevilla, José Antonio. *CMMI y la Plataforma de Desarrollo Software de IBM Racional*. IBM Software Group, 2003

[Mazur, 2000] Mazur Glenn. *Comprehensive Quality Function Deployment Overview*. 2000.

[Mazur, 2001] Mazur, Glenn. *QFD Black Belt Notes.* Japan Business Consultants, E.U., 2001.

[McDonald, 2005]] McDonald Landazuri, Bárbara A. *Definición de Perfiles en Herramientas de Gestión de Requisitos.* Septiembre de 2005.

[Mead *et al*., 2005] Mead, Nancy R., Hough, Eric D. y Stehney, Theodore R. *Security Quality Requirements Engineering (SQUARE) Methodology.* Software Engineering Institute, Carnegie Mellon University, 2005.

[Min. Fomento, 2005] Ministerio de Fomento. *Despliegue de la calidad demandada y planificada (QFD)*. 2005. Disponible en Web: <http://www.fomento.es/NR/rdonlyres/FEEE7FB6-52BE-45C0-9CF2-6A3BC187F382/19417/VA1.pdf>

[Menks, Ahmed y Fu, 2000] Menks, David. Ahmed, Anwar y Fu, Kaijun. *Quality Function Deployment*, 2000. Disponible en Web: <http://sern.ucalgary.ca/~kjfu/courses/SENG613/teamwork.html>

[Michael Trick, 1996] Trick, Michael. *Multiple Criteria Decision Making for Consultants.* 1996. Disponible en Web: <http://mat.gsia.cmu.edu/mstc/multiple/multiple.html>

[Monografías, Gestión] Monografias.com. *Gestión de Requisitos.* Disponible en Web: <http://www.monografias.com/trabajos41/requisitos-software/requisitos-software2.shtml>

[Monografías, Requisitos] Monografias.com. *Ingeniería de Requisitos – Ingeniería de Software.* Disponible en Web: < http://www.monografias.com/trabajos6/resof/resof.shtml#intro>

[NPD-NET, 2005] NPD – NET. *Modelo de Kano.* 2005. Disponible en Web: <http://www.vrc.gr:8080/npd-net/es/npd/page.html?page_id=1076>

[Nuseibeh y Easterbrook, 2000] Nuseibeh, Bashar y Easterbrook, Steve. *Requirements Engineering: A Roadmap.* 2000

[Ortín et al., 2001] Ortín Maria José, García Molina Jesús, Moros Begoña, Joaquín Nicolás. *El Modelo del Negocio como base del Modelo de Requisitos.* Grupo de Investigación de Ingeniería del Software, 2001. Disponible en Web: <http://www.lsi.us.es/~amador/JIRA/Ponencias/JIRA_Ortin.pdf>

[Rojas, 2006] Rojas Pobrete, Cristián. *Comunicación: El aspecto humano de la Ingeniería de Requisitos,* Universidad de Chile, 2006.

[Salvador Echeverría et al.] Salvador Echeverría Gómez, Rogelio Amezola Luna. *Enseñanza de Metodologías de diseño y problemática en su aplicación.* Disponible en Web: <http://www.ai.org.mx/ensenanza%20de%20metodologias.pdf>

[Schuldt, 1998] Jurgen E. Schuldt. *El premio Deming*. Luxembourgo, 1998. Disponible en Web: <http://www.geocities.com/wallstreet/exchange/9158/pdeming.htm>

[SEI, 2006] Software Engineering Institute. *CMMI for Development, Version 1.2*. Carnegie Mellon University, Agosto de 2006.

[Standish Group, 2004] The Standish Group. *The Chaos Report*. 2004 Disponible en Web: <http://www.standishgroup.com>

[Tan, Xie y Chia, 1997] K.C. Tan, M. Xie y E. Chia. *Quality function deployment and its use in designing information technology systems*. National University of Singapore, Kent Ridge, Singapore, 1997.

[Valdés, 2007] Valdés Herrera, Clemente. *Teoría de la Motivación – Higiene*. 2007. Disponible en Web: <http://www.wikilearning.com/teoria_de_la_motivacion_higiene-wkccp-21391-8.htm>

[Wikipedia, Casos de Uso] Wikipedia, The Free Encyclopedia. *Diagramas de Casos de Uso UML*. Disponible en Web: <http://es.wikipedia.org/wiki/Diagrama_de_casos_de_uso>

[Wikipedia, CMMI] Wikipedia, The Free Encyclopedia. *Capability Maturity Model Integration (CMMI)*. Disponible en Web: <http://es.wikipedia.org/wiki/CMMI>

[Wikipedia, TQM] Wikipedia, The Free Encyclopedia. *Total Quality Management (TQM)*. Disponible en Web: <http://en.wikipedia.org/wiki/Total_Quality_Management>

[Yacuzzy y Martín, 2002] Yacuzzy, E., y Martín, F. *Aplicación del modelo de Kano en el diseño de un producto farmacéutico. 2002*. Disponible en Web: <http://netec.mcc.ac.uk/WoPEc/data/Papers/cemdoctra224.html>

ANEXOS

ANEXO 1

La casa de la calidad realizada para el caso práctico se encuentra en formato digital junto al documento del proyecto de grado.

ANEXO 2

El prototipo del sistema a desarrollar para el Colegio Anglo Español, construido durante el proceso de validación de requisitos, se encuentra en la carpeta Anexo 2 – Prototipo.