



UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA

La Universidad Católica de Loja



MODALIDAD ABIERTA Y A DISTANCIA

Departamento de Ciencias de la Computación y Electrónica
Sección Ingeniería del Software y
Gestión de Tecnologías de la Información

Ingeniería de Requisitos

Texto-guía
4 créditos

Titulación	Ciclo
■ Ingeniero en Informática	VI

Autores:

Ing. Samantha Cueva
Ing. Manuel Sucunuta

Estimado estudiante recuerde que la presente guía didáctica está disponible en el EVA en formato PDF interactivo, lo que le permitirá acceder en línea a todos los recursos educativos.



18606

Asesoría virtual:

www.utpl.edu.ec

INGENIERÍA DE REQUISITOS

Texto-guía

Samanta Cueva
Manuel Sucunuta

UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA

 **Ecuador 3.0 By NC ND**

Diagramación, diseño e impresión:

EDILOJA Cía. Ltda.
Telefax: 593-7-2611418
San Cayetano Alto s/n

www.ediloja.com.ec
edilojainfo@ediloja.com.ec
Loja-Ecuador

Primera edición
Quinta reimpresión

ISBN-978-9942-08-212-1



Esta versión impresa, ha sido autorizada bajo las licencias Creative Commons Ecuador 3.0 de Reconocimiento -No comercial- Sin obras derivadas; la cual permite copiar, distribuir y comunicar públicamente la obra, mientras se reconozca la autoría original, no se utilice con fines comerciales ni se realicen obras derivadas. <http://www.creativecommons.org/licences/by-nc-nd/3.0/ec/>

Octubre, 2014



2. Índice.....	3
3. Introducción.....	5
4. Bibliografía.....	6
4.1. Básica	6
4.2. Complementaria	6
5. Orientaciones generales para el estudio	9
6. Proceso de enseñanza-aprendizaje para el logro de competencias.....	13

PRIMER BIMESTRE

6.1. Competencias genéricas	13
6.2. Planificación para el trabajo del alumno.....	13
6.3. Sistema de evaluación de la asignatura (primero y segundo bimestres).....	15
6.4. Orientaciones específicas para el aprendizaje por competencias.....	16
UNIDAD I: Fundamentos de la Ingeniería de Requerimientos	16
1.1. Requerimientos.....	16
1.2. Verificación y validación de requerimientos.....	19
1.3. Tipos de Requisitos	20
1.4. Formas de documentar los requerimientos	24
1.5. Características deseables de los requerimientos.....	25
1.6. Ingeniería de requisitos.....	26
1.7. Stakeholders (interesados)	31
Autoevaluación 1	34
UNIDAD 2: Preparar el Escenario para el Desarrollo de Requerimientos.....	38
2.1. Visionamiento.....	40
2.2. Glosario	45
2.3. Mitigar el riesgo en los requerimientos.....	48
Autoevaluación 2	52
UNIDAD 3: Elicitación de Requerimientos	54
3.1 Listar las fuentes de requerimientos	56
3.2. Categorías de los Stakeholders.....	57
3.3 Perfil de los stakeholders.....	62
3.4 Identificar técnicas combinadas de elicitación	64
3.5. Plan de elicitación de stakeholders	77
Autoevaluación 3	79

UNIDAD 4: : Análisis de Requerimientos	81
4.1. ¿Por qué se debería crear modelos de requisitos?	81
4.2. El ciclo de análisis de requerimientos.....	82
4.3. ¿Qué modelos se pueden crear?.....	83
4.4. ¿Cómo elegir el modelo correcto?	84
4.5. Herramientas y técnicas para analizar requerimientos	85
4.6. El modelado del negocio	86
4.7. Describir el alcance del proyecto.....	91
4.8. Agregar detalle a los requerimientos de usuario	95
4.9. Priorizar los requerimientos	118
Autoevaluación 4.....	120

SEGUNDO BIMESTRE

6.5. Competencias genéricas	123
6.6. Planificación para el trabajo del alumno	123
6.7. Orientaciones específicas para el aprendizaje por competencias	125
UNIDAD 5: Especificación de Requerimientos	125
5.1. Proceso de especificación de requerimientos	125
5.2. Técnicas y herramientas para documentar requerimientos.....	126
Autoevaluación 5.....	143
UNIDAD 6: Validación de Requerimientos	145
6.1 Validación de requerimientos	145
6.2. Técnicas de validación	147
6.3. Otros modelos de validación	153
Autoevaluación 6.....	157
UNIDAD 7: Gestión de Requerimientos	158
7.1 Gestión de Requisitos (Requirements Management, RM).....	158
7.2 Herramientas y técnicas que se utilizan en la gestión de requisitos.....	158
7.3. Ventajas y desventajas del uso de herramientas software en la IR.....	164
Autoevaluación 7	166
7. Solucionario	167
8. Anexos	175
Anexo 1: Desarrollo de requerimientos. (Gottesdiener E. , 2005)	175
Anexo 2: Factores a considerar al aplicar técnicas de obtención de requerimientos.....	176
Anexo 3: Esquema para documentar requerimientos de usuario. (Gottesdiener E. , 2005)	178
Anexo 4: Documento de especificación de requerimientos del caso de estudio, basado en el estándar IEEE Std 830-1998.	179
Apéndice A: Glosario	184
Apéndice B: Mapa de procesos	185
Apéndice C: Matriz de trazabilidad	186



3. Introducción

La presente asignatura consta de 4 créditos y forma parte del grupo de materias troncales de carrera de Ingeniería en Informática de la Escuela de Ciencias de la Computación en la modalidad de estudios Abierta y a Distancia.

La asignatura se presenta como una especialización al proceso de Ingeniería del Software en la parte de “Especificación de Requerimientos”, permitiendo a los estudiantes adquirir destrezas y habilidades en la captura de las necesidades de los stakeholders en un proyecto de desarrollo de software.

Considerando que la calidad de los proyectos de desarrollo de software depende en gran medida de la calidad de las especificaciones de software; Robertson S. y Roberson J. (2006)¹ sostiene que no importa la clase de proyecto que se esté desarrollando, ni tampoco si el proceso de desarrollo es ágil o pesado, si no se logra una correcta comprensión de los requerimientos por parte de los analistas y se asegura que el cliente también los comprende, el producto y el proyecto fallarán; por lo que el propósito de esta asignatura es instruir a los estudiantes en las técnicas, metodologías y estándares para descubrir y especificar requisitos basados en el proceso de desarrollo y gestión de requisitos.

Se desarrollará un trabajo colaborativo para socializar los diferentes criterios sobre los casos de estudio que se propondrán en el Entorno Virtual de Aprendizaje. La asignatura está compuesta por 7 unidades, distribuidas en 4 para el primer bimestre y 3 para el segundo.

En la primera unidad se abordan los fundamentos teóricos de la Ingeniería de Requisitos, partiendo de los niveles y categorías de requisitos, el proceso de ingeniería de requisitos, las características deseables de requisitos; la segunda se refiere a los escenarios de desarrollo; en la tercera unidad se refiere a elicitación de requisitos, la cuarta unidad se refiere a análisis.

En el segundo bimestre, la unidad cinco se enfoca a la especificación de requerimientos, en la unidad seis se analiza la validación de requisitos en donde se analizan las técnicas y herramientas que sirven para la realizar la validación de requisitos; finalmente en la unidad siete se analizan las técnicas de gestión de requisitos.

Estimado estudiante sea constante en el estudio ya que a través de esta materia podrá centrar las bases para construir un sistema de calidad acorde a las necesidades reales de los usuarios.

Recuerde que durante el proceso de aprendizaje le estaremos acompañando para orientarle en el proceso de aprendizaje.

1 S. Robertson & J. Robertson, Mastering the Requirements Process, Addison Wesley, USA, 2006, pp.



4. Bibliografía

4.1. Básica

Cueva, S.; Sucunuta, M. (2011). *Guía Didáctica de Ingeniería de Requisitos*. Loja – Ecuador. Editorial UTPL.

Texto guía diseñado para el estudio de Ingeniería de Requisitos en la carrera de Ingeniería en Informática de la Modalidad Abierta y a Distancia de la Universidad Técnica Particular de Loja.

4.2. Complementaria

- GOTTESDIENER, ELLEN (2009), *The Software Requirements Memory Jogger*, Goal QPC Inc.
Guía fácil de usar para el desarrollo y gestión de requisitos; proporciona a cada miembro del equipo del proyecto las herramientas y técnicas para fomentar la comunicación entre las empresas y los equipos técnicos sobre los requisitos necesarios para la producción de software con éxito.
- Lamsweerde, A., (2009). *Requirements Engineering: from system goals to UML models to software Specifications*. Inglaterra. Editorial Wiley.
En este libro se hace un especial referencia a los conceptos preliminares de la Ingeniería de Requisitos, además de un profundo análisis de los casos de los requisitos.
- Pressman, R., (2010). *Ingeniería del Software: Un enfoque práctico*. México. Editorial McGraw-Hill.
El texto se presenta como un medio de consulta a nivel general para entender a la ingeniería de requisitos en el contexto de la ingeniería del software.
- International Institute of Business Analysis. (2009) *A Guide to the Business Analysis Body of Knowledge*. Guía que recoge el análisis de negocios en donde se reflejan las mejores prácticas proporcionando un marco que describe las áreas de conocimiento, con actividades y tareas asociadas.
- Wiegers, K. (2003) *Software Requirements*. Microsoft Press.
Texto que reúne los principales componentes del proceso de gestión y administración de requerimientos.
- BOOCH, G.; RUMBAUGH, J. & JACOBSON, I. (2006), *El lenguaje unificado de modelado*, Addison Wesley.
El texto permite conocer sobre las diferentes formas de modelado.
- IBM Rational, 2003. *Mastering requirements with Use Cases*.
Texto sobre el análisis y diseño de Casos de Uso utilizando como software a Rational de IBM.
- Robertson, S. Robertson J. (2006) *Mastering the requirements Process*. Addison Wesley.
Texto enfocado al desarrollo de requisitos formales.
- Sommerville, I. (2005). *Ingeniería del Software*. Madrid: Pearson Education.
Libro que enfoca todos los procesos de la Ingeniería de Software.
- Loucopoulos, P., & Karakostas, V. (1995). *System Requirements Engineering*. McGraw-Hill.



Texto que presenta una visión equilibrada de los temas, conceptos, modelos, técnicas y herramientas que se encuentran en la investigación y práctica de la ingeniería de requisitos.

- IEEE Computer Society. (2004). *SWEBOK: Guide to the Software Engineering Body of Knowledge*. Los Alamitos, California: IEEE.
Guía de Ingeniería del Software que incluye Ingeniería de Software, Certificado de Desarrollo de Software Profesional, incluye prácticas completas.

OCW (Recurso Educativo Abierto)

- García, F. J., Conde, M. Á., & Bravo, S. (2008). *Ingeniería del Software, Introducción a la Ingeniería de Requisitos*. [En línea]. España. Disponible en: Sitio OCW de la Universidad de Salamanca: <http://ocw.usal.es/enseñanzas-tecnicas/ingenieria-del-software/contenidos/Tema3-IntroduccionalaIR-1pp.pdf> [Consulta 14-03-2011]
Recurso Educativo Abierto en el que se aborda la introducción a la Ingeniería de Requisitos.
- Ros J., Alvarez A. *Fundamentos de Ingeniería del Software* (2009). [En línea]. España. Disponible en: Sitio OCW de la Universidad de Murcia. <http://ocw.um.es/ingenierias/fundamentos-de-ingenieria-del-software>. [Consulta 14-03-2011].
Recurso Educativo Abierto en que se revisa el proceso de ingeniería del software y de la Ingeniería de Requisitos.

Direcciones electrónicas:

- SOFTWARE ENGINEERING INSTITUTE. [En línea]. Pittsburgh. Disponible en <http://www.sei.cmu.edu> [Consulta 15-04-2011].
Instituto de Ingeniería del Software, tiene como objetivo apoyar a las organizaciones a mejorar las capacidades en ingeniería del software de tal forma que el desarrollo y adquisición del software sea adecuado, libre de defectos, dentro del presupuesto. Para lo cual brinda soluciones en la adquisición, seguridad, desarrollo de software, manejo de procesos, riesgos y diseño de sistemas.
- NASA ONLINE DIRECTIVES INFORMATION. [En línea]. EEUU. Disponible en <http://nodis.hq.nasa.gov/> [Consulta 15-04-2011].
Sitio donde se dan lineamientos sobre la Ingeniería de Requisitos de Software.
- HANDBOOK OF SOFTWARE ARCHITECTURE. [En línea]. EEUU. Disponible en <http://www.handbookofsoftwarearchitecture.com/index.jsp?page=Main> [Consulta 15-04-2011].
Manual de Arquitectura de Software que hace referencia sobre el diseño de sistemas software, escritos para desarrolladores y administradores de proyectos.
- RESOURCES FOR SOFTWARE ARCHITECTS. [En línea]. Indiana. Disponible en <http://www.bredemeyer.com/index.html> [Consulta 15-04-2011].
Sitio que cuenta con una gran variedad de recursos en temas de arquitectura para las aplicaciones software, siendo de gran ayuda para los arquitectos de aplicaciones a profundizar y comprender su función en el desarrollo de los proyectos.
- AGILE ALLIANCE. [En línea]. Dallas. Disponible en <http://www.agilealliance.org/> [Consulta 15-04-2011].

Este sitio cuenta con importantes temas que tienen que ver con las tecnologías ágiles. Entre los recursos mas importantes están: presentaciones de conferencias, libros, artículos, grupos de usuarios, entre otros.

- AGILE MODELING. [En línea]. Canadá. Disponible en <http://www.agilemodeling.com/> [Consulta 15-04-2011].

Este sitio cuenta con documentación importante sobre el modelado ágil. En lo que tiene que ver a los requerimientos, existe un importante resumen de los temas del libro "Agile Software Requirements" de Dean Leffingwell.



5. Orientaciones generales para el estudio

Estudiar a distancia es un reto que requiere esfuerzo, dedicación y sobre todo de organización, por ello debe hacer de esta actividad un trabajo continuo y sistemático, organice su tiempo de manera que pueda verdaderamente aprovechar los contenidos que se le están ofreciendo. Le sugerimos hacer vida esta frase, que aunque le puede parecer trillada, es la que más se adapta a la realidad de las personas que estudian a distancia: “No deje para mañana lo que puede y debe hacer hoy”.

Le proponemos algunas orientaciones que le servirán en su proceso de aprendizaje:

Materiales

- Recuerde que los materiales con los que trabajará son este texto guía que le irá orientando sobre cómo y qué temas estudiar, además contiene ejemplos, ejercicios y autoevaluaciones que le permitirán medir su grado de comprensión y la necesidad de tutoría por parte del docente. Además se podrá apoyar en la bibliografía complementaria.
- Para reforzar el proceso de aprendizaje usted tiene un profesor-tutor que le guiará en el ciclo académico; al cual podrá hacer las consultas que requiera en un horario que oportunamente se estará publicando. Las tutorías las puede hacer mediante el EVA, correo electrónico o directamente a través de la línea telefónica, así que aproveche esta alternativa que la UTPL pone a su disposición.
- Los trabajos a distancia: Son actividades teóricas y prácticas que acompañan a la guía didáctica de cada una de las materias, le permiten aplicar y reforzar los conocimientos adquiridos mediante su desarrollo, y debe enviarlos a su profesor. La entrega de estos trabajos con su respectiva carátula es obligatoria y no recuperable, lo cual significa que si no entrega alguno de los mismos no tendrá opción a la evaluación presencial, su valoración es de 6 puntos; la misma que se compone de una parte objetiva, una parte de ensayo y una parte de interacción con el EVA, para esto revise las evaluaciones a distancia que acompañan a esta guía en donde se especifica exactamente las actividades que debe desarrollar.

¿Cómo estudiar?

- Programe un horario de estudio diario; para esta asignatura es necesario que dedique al menos una hora diaria de autoestudio.
- Para ayudarse en el proceso de aprendizaje utilice las técnicas de estudio que más se adapten a su manera de aprender: subrayado, resúmenes, cuadros sinópticos y trate, en lo posible, de estudiar en un horario y ambiente adecuado.
- Como una estrategia de aprendizaje le sugiero realice todas las autoevaluaciones y ejercicios que se plantean en el presente texto-guía, ya que esto le permitirá poner en práctica los conocimientos teóricos de la asignatura. No olvide que si surge alguna inquietud con respecto a estos ejercicios, puede pedir asesoría al profesor-tutor.

- En este texto-guía por cada bimestre existe la “planificación del trabajo del alumno” en donde especialmente se indica el tiempo que debe dedicar a la asignatura por cada tema que comprende el plan de asignatura, esto le permitirá avanzar organizadamente en el estudio y no tener problemas de acumulación de trabajo al final. Recuerde que esta es una asignatura troncal de carrera y por lo tanto el tiempo a dedicarle debe ser el necesario.

Apoyo tecnológico e interactividad

- Como parte del proceso de enseñanza se dispone del Entorno Virtual de Aprendizaje (EVA), de la biblioteca virtual, repositorio de documentos (OCW), entre otros por lo que es fundamental que revise continuamente estos recursos con el fin de ponerse al tanto de los materiales de su asignatura. Preste especial interés al EVA ya que con este recurso el profesor-tutor publicará algunos casos y ejercicios que le ayuden a reforzar los conocimientos, así como también de su participación ya sea en foros, tareas o el mismo desarrollo y registro del trabajo a distancia. Preste especial interés en los recursos ROA, ya que el profesor tutor estará publicando presentaciones, videos, archivos, etc., que tengan relación con la asignatura. Por lo que le animo a que ingrese de forma periódica para que este al tanto del avance del estudio de la asignatura.

Esperamos que todas y cada una de estas recomendaciones contribuyan al aprendizaje exitoso de esta asignatura. Como ayuda gráfica a la presente guía se utilizarán los siguientes focalizadores:

ÍCONO	DESCRIPCIÓN
	Para realizar lecturas recomendadas, texto complementarios, OCW, o anexos.
	Tiene que desarrollar las autoevaluaciones.
	Realizar los ejercicios y actividades recomendadas, no son ejercicios obligatorios pero se sugiere realizarlos.
	Para temas puntuales que se hacen referencia a algún componente.
	Para aspectos sumamente importantes y que tiene que prestar especial atención.



	Para poner atención a los ejemplos que se proponen, especialmente como caso de estudio.
	Para hacer referencia a una explicación importante.
	Para ubicar el tema en el contexto real.



6. Proceso de enseñanza-aprendizaje para el logro de competencias

PRIMER BIMESTRE

6.1. Competencias genéricas

- Capacidad de abstracción, análisis y síntesis.
- Habilidades para buscar, procesar y analizar información procedente de fuentes diversas.
- Capacidad para identificar, plantear y resolver problemas
- Capacidad para tomar decisiones
- Capacidad de trabajo en equipo
- Capacidad para formular, diseñar y gestionar proyectos.

6.2. Planificación para el trabajo del alumno

COMPETENCIAS ESPECÍFICAS	INDICADORES DE APRENDIZAJE	CONTENIDOS	ACTIVIDADES DE APRENDIZAJE	CRONOGRAMA ORIENTATIVO Tiempo Estimado
		UNIDADES/TEMAS		
Conocer la importancia de la Ingeniería de Requisitos en el proceso de desarrollo de software	• Reconoce las fases del proceso de ingeniería de requisitos. • Diferencia los roles de las personas que intervienen en el proceso de ingeniería de requisitos. • Reconoce las áreas de conocimiento que deben ser asumidas por los diferentes roles del proceso de ingeniería de requisitos • Reconoce los actores que participan en el ciclo de vida de la ingeniería de requisitos.	UNIDAD 1: FUNDAMENTOS DE INGENIERÍA DE REQUISITOS 1.1. Requerimientos 1.2. Verificación y validación de requerimientos. 1.3. Tipos de requisitos 1.4. Formas de documentar los requerimientos 1.5. Características deseables de los requerimientos 1.6. Ingeniería de requisitos 1.7. Stakeholders (involucrados) 1.8. Autoevaluación	• Revisión de contenidos en el texto guía. • Familiarización con el material. • Lectura comprensiva. • Desarrollo de actividades recomendadas en la guía y ejercicios propuestos en el texto básico. • Interacción en el EVA. • Inicio del desarrollo de la evaluación a distancia.	Semana 1 • 4 horas de autoestudio. • 4 hora de interacción.

Utilizar técnicas de captura y especificación de requisitos en el desarrollo de proyectos de software.	- Identifica la visión del producto y establece la problemática de la solución. - Establece un diccionario de términos comunes. - Identifica estrategias para mitigar los riesgos en los requerimientos de usuario.	UNIDAD 2: ESCENARIOS PARA DESARROLLO DE REQUERIMIENTOS 2.1. Visionamiento 2.2. Glosario 2.3. Estrategias para mitigar los riesgos en los requerimientos. 2.4. Autoevaluación	- Revisión de contenidos del capítulo 2 en el texto guía. - Lectura comprensiva. - Desarrollo de actividades recomendadas en la guía y ejercicios propuestos en el texto básico. - Interacción en el EVA.	Semana 2 4 horas de autoestudio 4 horas de interacción
		UNIDAD 3: ELICITACIÓN DE REQUERIMIENTOS 3.1. Fuentes de requerimientos. 3.2. Categorías de los Stakeholders. 3.3. Perfiles de los Stakeholders. 3.4. Técnicas de elicitación	- Revisión de contenidos del capítulo 3 en el texto guía. - Lectura comprensiva. - Desarrollo de actividades recomendadas en la guía. - Desarrollo de los ejercicios propuestos en el texto básico. - Interacción en el EVA.	Semana 3 y 4 8 horas de autoestudio 8 horas de interacción
	- Identifica los requisitos desde las fuentes de información como reportes de entrevista, solicitudes de desarrollo y documentación de los procesos de negocio. - Diferencia entre requisitos funcionales y no funcionales. - Prioriza los requisitos en función de la importancia para el cliente y el valor del negocio	UNIDAD 4: ANÁLISIS DE REQUERIMIENTOS 4.1. Modelos de requisitos. 4.2. Ciclo de análisis de requerimientos. 4.3. Modelos a crear. 4.4. Elegir el modelo. 4.5. Herramientas y técnicas.	- Revisión de contenidos del capítulo 4 en el texto guía. - Desarrollo de actividades recomendadas en la guía y ejercicios propuestos en el texto básico. - Interacción en el EVA. - Inicio del desarrollo de la evaluación a distancia.	Semana 5 y 6 8 horas de autoestudio 8 horas de interacción
		UNIDADES 1 a la 4	Revisión de contenidos como preparación para la evaluación presencial.	Semana 7 y 8 8 horas de autoestudio. 8 horas de interacción.

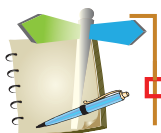


6.3. Sistema de evaluación de la asignatura (primero y segundo bimestres)

Formas de Evaluación Competencia: Criterio		1. Autoevaluación*	2. Heteroevaluación				3. Coevaluación	
			Evaluación a Distancia**			Evaluación Presencial		
			Parte Objetiva	Parte de Ensayo	Interacción en el EVA	Prueba Objetiva y de Ensayo		
Actitudes	Comportamiento ético	X	X	X	X	X		
	Cumplimiento, puntualidad y responsabilidad	X	X	X				
	Esfuerzo e interés en los trabajos	X	X	X	X	X		
	Respeto a las personas y a las normas de comunicación				X			
	Contribución en el trabajo colaborativo y de equipo							
	Presentación, orden y ortografía			X		X		
	Emite juicios de valor argumentadamente							
Conocimientos	Dominio del contenido	X	X	X	X	X		
	Investigación (cita fuentes de consulta)			X	X			
	Aporta con criterios y soluciones			X	X			
	Análisis y profundidad en el desarrollo de los temas							
PORCENTAJE		Estrategia de Aprendizaje	10%	20%	30%	Máximo 1 punto (Completa la evaluación a distancia)	70%	Actividades Presenciales y en el eva
Puntaje			2	4	6	14		
TOTAL					20 Puntos			
Para aprobar la asignatura se requiere obtener un puntaje mínimo de 28/40 puntos, que equivale al 70%.								
* Son estrategias de aprendizaje, no tienen calificación; pero debe responderlas con el fin de autocomprobar su proceso de aprendizaje.								
** Recuerde: que la evaluación a distancia del primer bimestre y segundo bimestre consta de dos partes: una objetiva y otra de ensayo, debe desarrollarla y entregarla en la fecha establecida.								

Señor estudiante:

Tenga presente que la finalidad de la valoración cualitativa es principalmente formativa.



6.4. Orientaciones específicas para el aprendizaje por competencias



UNIDAD I: Fundamentos de la Ingeniería de Requerimientos



A la hora de construir una aplicación software es fundamental que los desarrolladores conozcan de forma precisa el problema que van a resolver, de tal manera que la solución que se construya sea correcta y útil. Por tal motivo la correcta obtención de los requerimientos del sistema es uno de los aspectos clave en la construcción de proyectos de software, ya

sea en proyectos grandes o pequeños con complejidades diferentes la mala captura de los mismos es la causa de los problemas que surgen a lo largo del proceso de construcción. La ingeniería de requisitos como parte de la ingeniería del software permite la definición de los servicios y características que el sistema debe tener.



Estimado estudiante empecemos con el estudio de esta primera unidad donde se abordan los principales aspectos de la ingeniería de requerimientos los mismos que le permitirán conocer la importancia de la misma en el proceso de desarrollo de software.

1.1. Requerimientos

Empecemos definiendo el concepto de requisitos:

- Según Benet (2003:110) “Los requisitos son la especificación de lo que debe hacer el software, son descripciones del comportamiento, propiedades y restricciones del software que hay que desarrollar”.
- Los requisitos son descripciones de las propiedades necesarias y suficientes de un producto para que satisfaga las necesidades del consumidor. (Gottesdiener E. , 2005).
- “Un requisito del software es una característica que debe exhibir para solucionar cierto problema del mundo real. Por lo tanto, un requisito del software es una característica que se debe exhibir por el software desarrollado o adaptado para solucionar un problema particular.” (SWEBOK, 2004: 2-1).



Basado en estos conceptos, defina con sus propias palabras ¿Qué es un requisito de software?.

¿Por qué es importante tener “requisitos” de calidad en el ciclo de vida del desarrollo de un software?

Para entregar un producto de software con éxito, Ud. necesita desarrollar, documentar y validar los requisitos de software. Los requisitos bien entendidos son la base para determinar el éxito del software implementado, lo cual permite satisfacer las necesidades de los usuarios; dichas necesidades son las que se definen en los requisitos.



El no definir los requisitos correctamente tiene un precio bastante alto ya que se ocasionan requisitos mal definidos; estos errores pueden causar requisitos incompletos, incorrectos o requisitos contradictorios, entre los que se pueden mencionar a:

- Sobrecosto,
- Reproceso costoso,
- Mala calidad,
- Retraso en la entrega,
- Clientes descontentos,
- Miembros de equipo agotados y desmoralizados.

La importancia de tener requisitos de calidad radica en:

- Involucran del 10 al 15% del coste total del proyecto.
- Un error en los requisitos puede ser de 10 hasta 100 veces más costoso que un error en el código.
- Una equivocación en la etapa de requisitos se arrastra en las demás fases.

Para reducir el riesgo de fracasos en proyectos de software y los altos costos asociados a los mismos por causa de los requisitos defectuosos se deben definir correctamente los requisitos en el inicio del proceso del desarrollo del software para lo cual se debe realizar una estimación lo más real posible del tiempo que se necesita para realizar esta etapa.

Por estas razones los requerimientos deben tener las siguientes características:

- Ser una combinación compleja de los requisitos (necesidades) de diferentes personas (stakeholders) que pertenecen a diferentes niveles de una organización y entorno en donde se operará el software.
- Deben ser verificables.
- Deben ser lo más claros que se pueda y cuantificables en medida de lo posible.

Uno de los principales problemas al que nos enfrentamos como ingenieros de software en el desarrollo de sistemas es la ingeniería de requisitos, pues de esta fase depende el éxito del producto de software; considerando que si hay algún error en esta fase el resto de fases del ciclo de vida también se verán afectados y por ende el resultado es un producto de software que no cumple con las necesidades de los stakeholders.



“La correcta obtención de los requisitos es uno de los aspectos más críticos de un proyecto software, independientemente del tipo de proyecto que se trate, dado que una mala captura de los mismos es la causa de la mayor parte de los problemas que surgen a lo largo del ciclo de vida”. Johnson 1995: 2(1):41-47.

Observe la figura 1.1.

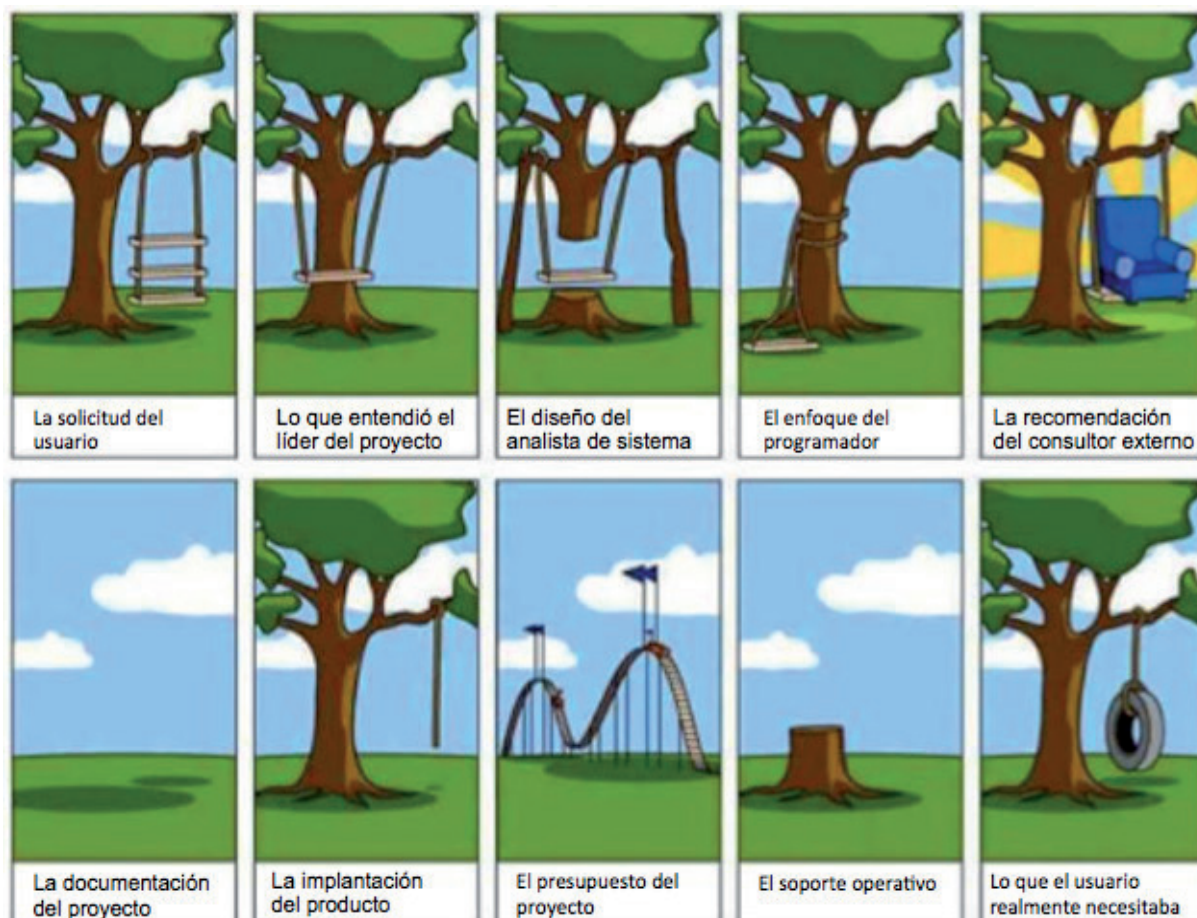


Figura 1.1: Dificultad de los requisitos

¿Qué puede concluir de dicha observación?

¡Efectivamente!, se están reflejando los problemas que existen en el proceso de recolección de datos. Los usuarios tienen una forma de expresar sus necesidades, el líder del proyecto, el analista de sistemas, el programador le dan un enfoque totalmente diferente; por otro lado la recomendación del consultor externo le añade otras funcionalidades al producto, además no existe documentación del proyecto y no se ha realizado el estudio de factibilidad económica, no hay un soporte operativo eficiente y finalmente el producto que el usuario necesitaba era algo sencillo, sin tantas complicaciones.

Resumiendo, las principales dificultades que se presentan en el proceso de recolección de requisitos se pueden mencionar que los requerimientos:

- No reflejan las necesidades reales de los clientes
- Son inconsistentes y/o incompletos.
- Realizar cambios sobre los requisitos ya definidos es muy costoso.
- Pueden existir malentendidos entre los stakeholders, y los ingenieros de software.
- Imprecisión de los requisitos, lo cual provoca que sean interpretados de diferentes formas por los stakeholders.
- Frecuentemente no está clara la frontera entre requisitos y diseño.



Para poder solucionar estos problemas surge la ingeniería de requisitos; a la cual Somerville, 2005:108; la define como “El proceso de establecer los servicios que el cliente requiere de un sistema y los límites bajo los cuales opera y se desarrolla”. La ingeniería de requisitos es la primera fase del ciclo de vida del software en la que se producen especificaciones a partir de ideas informales por parte de los stakeholders.

1.2. Verificación y validación de requerimientos

Los requisitos son fundamentales para el éxito del producto final. Se debe enfocar en el problema, es decir definir qué es lo que se desea construir y asegurar qué es necesario para satisfacer las necesidades del usuario; aunque las pruebas del software no se ejecutan durante el desarrollo la realización de pruebas conceptuales ayudará a descubrir la existencia de requisitos defectuosos, sin claridad o incompletos.

Por lo cual es aconsejable que de acuerdo a como se vayan desarrollando los requisitos estos sean verificados para comprobar si cumplen las condiciones o especificaciones de la actividad de desarrollo los requisitos.

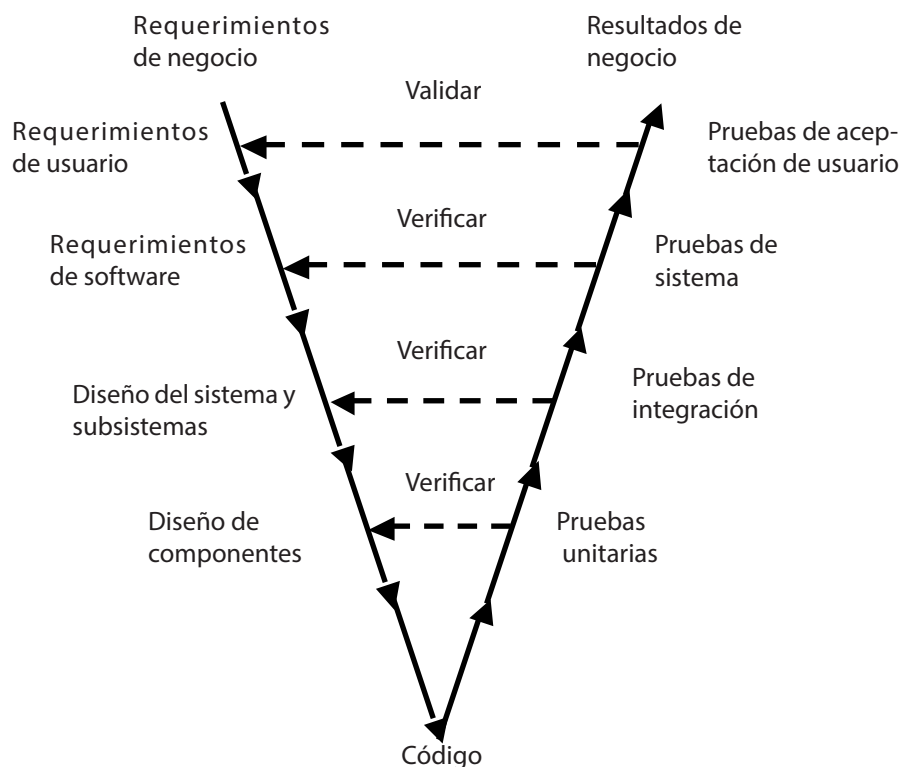


Figura 1.2: Proceso de verificar y validar requerimientos. (Gottesdiener E. , 2005)

Cuando se identifican los requisitos y se tiene la aceptación del usuario se asegura que se satisfacen las necesidades del usuario.

La *verificación* de requisitos representa el punto de vista del equipo de desarrollo asegurando que el software satisface los requisitos especificados, mientras que la validación de requerimientos está preocupada por el punto de vista del cliente asegurando que las necesidades del cliente se cumplan.

En la figura 1.2, se indica el proceso de verificar y validar requerimientos; según (Gottesdiener, 2005).

1.3. Tipos de Requisitos

Estos requisitos se pueden clasificar en: funcionales y no funcionales.

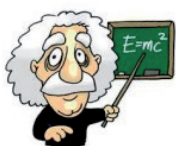
1.3.1 Requisitos Funcionales:

“Describen las funciones que el software debe ejecutar, a veces se conocen como capacidades”. SEWBOK, 2004: 2-2.

A los requisitos funcionales se los puede dividir en:

- **De usuario:** Los requerimientos de usuario, según Sommerville, 2005: 116; “Son declaraciones, en lenguaje natural y en diagramas, de los servicios que se espera que el sistema provea y de las restricciones bajo las cuales se debe operar.
- **Del sistema:** “Establecen con detalle los servicios y restricciones del sistema. El documento de requerimientos del sistema, algunas veces denominado especificación funcional, debe ser preciso”. Sommerville, 2005: 118

Ahora para comprender la diferencia de estos requisitos analice el siguiente ejemplo.



Ejemplo 1.1:

Dadas las siguientes premisas. Indique ¿Qué tipo de requisito es?

Enunciado: El sistema debe permitir al usuario introducir los datos de los estudiantes nuevos.

Análisis del problema:

Requisito de usuario expresado en términos generales. ¿Qué servicio debe prestar el sistema?

Enunciado: El sistema debe permitir a los usuarios buscar el producto por nombre, número de factura, código de barras.

Análisis del problema:

Requisito del sistema: Que define una parte de funcionalidad del sistema.

¿Le quedó clara la diferencia, entre los requisitos funcionales?. ¡Todavía no!, entonces le sugiero que se haga más ejercicios sobre el tema.



Ejercicio 1.1:

Dadas las siguientes premisas. Indique ¿Qué tipo de requisito es?

Plantéese, enunciados sobre requerimientos de un sistema e identifique los requerimientos funcionales. (Puede apoyarse en los ejercicios que se le ha colado en el EVA).



Ahora que ya quedó clara la diferencia entre los requisitos funcionales, revise a que se refieren los requisitos no funcionales.

1.3.2 Requisitos no funcionales:

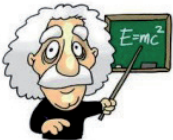
Estos requisitos incluyen áreas tales como el rendimiento, el diseño y las limitaciones de la aplicación; en SWEBOOK, 2004: 2-2 se los define como “son los que actúan para limitar la solución, se los conoce como restricciones o requisitos de calidad”.

Además los requisitos no funcionales pueden estar relacionados con propiedades emergentes del sistema, describen restricciones externas del sistema; definen las cualidades globales que el sistema ha de exhibir y son más críticos que los requisitos funcionales.

Los requisitos no funcionales son propiedades que el producto debe tener lo que no puede ser evidente al usuario, incluyendo atributos de calidad, coacciones, e interfaces externo. (Gottesdiener E., 2005) Sommerville, 2005:111; clasifica a los requisitos no funcionales en: “requisitos de producto, de organización y externos”.

- o **Requisitos de producto:** Estos especifican el comportamiento del producto.
- o **Requisitos de organización:** Se derivan de las políticas y procedimientos existentes en la organización del cliente y en la del desarrollador.
- o **Requisitos externos:** Son los requisitos que derivan de los factores externos al sistema y de su proceso de desarrollo, incluyen requerimientos de interoperabilidad que definen la manera en que el sistema interactúa con los otros sistemas de la organización.

Analice el siguiente ejemplo que le permitirá diferenciar entre los requisitos no funcionales.



Ejemplo 1.2:

Dadas las siguientes premisas. Indique ¿Qué tipo de requisito es?

Enunciado: El máximo espacio de almacenamiento ocupado por el sistema debe ser de 20 MB porque el sistema debe alojarse completamente en una memoria de solo lectura e instalarse en el palm.

Análisis del problema:

Requisito de producto que define una restricción en el tamaño del producto.

Enunciado: El proceso software y los documentos a realizar deben conformar el proceso y los estándares de documentación recogidos en la norma IEEE-830”.

Análisis del problema:

Requisito de organización que especifica que el sistema debe desarrollarse de acuerdo a un proceso estándar dentro de la empresa.

Enunciado: El sistema no debe revelar ninguna información personal sobre los clientes excepto su nombre y su número de referencia”

Análisis del problema:

Requisito externo se deriva de la necesidad del sistema de cumplir la legislación vigente sobre protección de datos.

Finalmente para terminar el tema de la categorización de requisitos, le invitamos a que revise la figura 1.3.

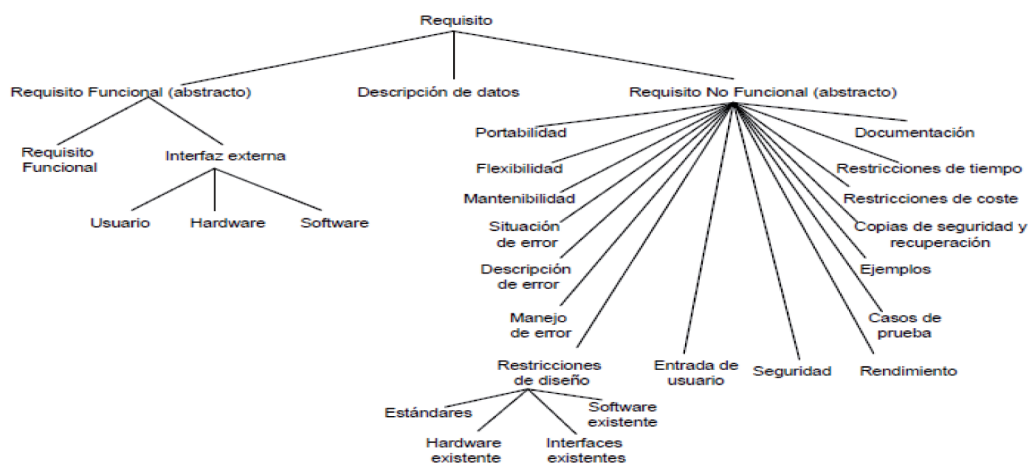


Figura 1.3: Jerarquía de especialización de requisitos³

Con la figura anterior, esperamos que le quede clara la clasificación de requisitos; recuerde no está solo en su proceso de aprendizaje, si se le presentan dudas en los temas tratados puede contactarse con sus profesores.

¿De dónde provienen los requisitos?

Los requisitos provienen de tres niveles:

Tabla 1.1: Niveles de procedencia de los requisitos

Nº Nivel	Nivel	Descripción
1	Requerimientos del negocio: ¿Por qué se está realizando el Proyecto?	Los requerimientos del negocio son las declaraciones de la empresa para justificar la ejecución del proyecto; incluye una visión del producto de software impulsado por los objetivos del negocio. Es decir describen el propósito y las necesidades a alto nivel que el producto debe satisfacer; además con la visión del producto se determinan las capacidades que el producto debe tener y también las limitaciones del mismo.

³ García, F. J., Conde, M. Á., & Bravo, S. (16 de 10 de 2008). *Ingeniería del Software, Introducción a la Ingeniería de Requisitos*. Recuperado el 14 de Abril de 2011, de Sitio OCW de la Universidad de Salamanca: <http://ocw.usal.es/enseanzas-tecnicas/ingenieria-del-software/contenidos/Tema3-IntroduccionalIR-1pp.pdf>



2	Requerimientos del usuario: Lo que los usuarios podrían hacer con el producto.	Los requerimientos de los usuarios son la definición de los requisitos de software desde el punto de vista del usuario. Se describen las tareas que los usuarios necesitan llevar a cabo con el software y las características de calidad necesarias del software. Los documentos que contienen los requisitos del usuario a menudo se llaman capacidades operativas, características del producto, el concepto de las operaciones, o casos de uso. Los requisitos de los usuarios son el puente entre los objetivos de negocio y los requisitos de software detallados. Por esta razón, es importante asegurar que los analistas que escriben los requisitos tengan excelentes habilidades de comunicación, así como conocimiento de modelos de requisitos de usuario.
3	Requerimientos de software: Lo que los desarrolladores necesitan construir	Los requisitos de software son las descripciones detalladas de todos los requisitos funcionales y no funcionales que el software debe cumplir para satisfacer las necesidades del negocio y del usuario, sin salirse de los límites del diseño e implementación. Los requisitos de software establecen un acuerdo entre el personal técnico y los empresarios sobre las características que el producto debe tener. Los documentos que contienen los requisitos de software se los conoce como especificación de requisitos de software, requisitos detallados, especificación, especificación técnica o especificación funcional. Por lo general, los autores de la especificación de requerimientos de software son analistas, sin embargo, los clientes también deben revisar y aprobar los requisitos de software.



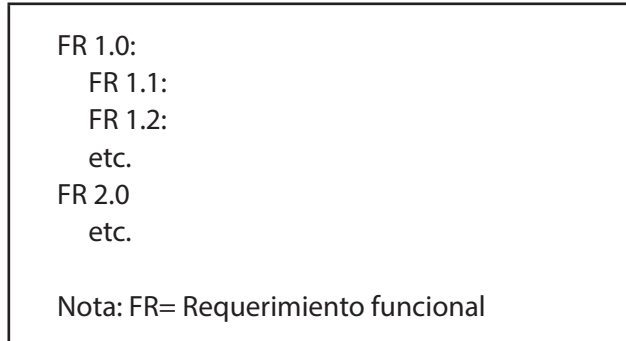
Ejercicios

Elabore un cuadro sinóptico de los tipos de requisitos que existen en el desarrollo de software.

1.4. Formas de documentar los requerimientos

Existen varias formas de documentar los requisitos:

a) En forma Textual



b) Diagrama de árbol

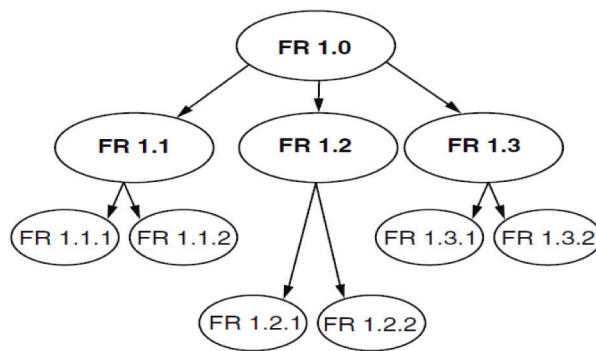


Figura 1.4: Diagrama de árbol

c) Requisitos como modelos

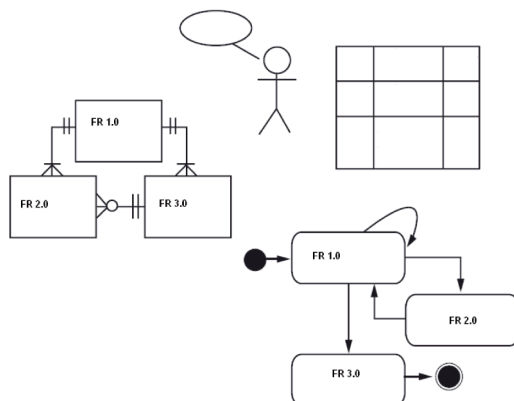


Figura 1.5: Requisitos como modelos



1.5. Características deseables de los requerimientos

Realizar una recolección y documentación de los requisitos de alta calidad es fundamental para el desarrollo de productos de software con éxito; para asegurarse de que están desarrollando los requisitos eficazmente, debe asegurarse de que todos sus requerimientos tengan las siguientes características:

1.5.1. Características de requisitos individuales

Las características deseables que todo requisito debe cumplir son:

- **Completo:** Un requerimiento está completo si no necesita ampliar detalles en su redacción, es decir, proporciona la información suficiente para su comprensión.
- **Correcto:** Cada requerimiento debe describir con exactitud la funcionalidad para ser construida (K., 2003).
- **Claro:** Pueden ser entendidos de la misma manera por todas las partes interesadas con un mínimo de explicación complementaria. (Gottesdiener E., 2005).
- **Factible:** Debe ser posible poner en práctica cada requerimiento dentro de las capacidades conocidas y las limitaciones del sistema en su entorno de operaciones (K., 2003).
- **Necesario:** Un requerimiento es necesario, si cuando se prescinde del mismo provoca una deficiencia en el sistema a construir; además cuando sus características físicas o factor de calidad no pueden ser reemplazados por otras capacidades del producto.
- **Priorización:** Dentro del conjunto de requerimientos, alguno de ellos debe ser más importante que los otros; en este proceso deben intervenir los stakeholders.
- **Sin ambigüedades:** Un requerimiento no tiene ambigüedades cuando se lo puede interpretar de una sola forma, y por lo tanto el lenguaje usado en su definición no causa confusiones al lector.
- **Verificable:** Un requerimiento es verificable cuando puede ser cuantificado de manera que se pueden utilizar los métodos de verificación de inspección, análisis, demostración o pruebas.

1.5.2. Características de especificaciones de requerimientos

Recuerde que no es suficiente tener excelentes declaraciones de los requisitos individuales; sino que también deben cumplir condiciones dentro del conjunto de requerimientos, las mismas que se detallan a continuación:

- **Completo:** Ningún requerimiento o información necesaria deberían estar ausentes, sin embargo los requisitos que faltan son difíciles de detectar porque no están descritos.
- **Consistente:** Los requisitos de conformidad no entren en conflicto con otros requisitos del mismo tipo o con un mayor nivel de negocios, sistema o necesidades de los usuarios (Wiegers, 2003).
- **Modificable:** Debe ser capaz de revisar en el SRS cuando sea necesario y para mantener un historial de los cambios realizados de acuerdo a cada necesidad surgida; cada requisito debe aparecer solo una vez en el SRS.
- **Trazable:** El requisito de trazabilidad puede estar vinculado a su origen hacia atrás y hacia adelante a los elementos de diseño y código fuente que aplicarla a uno de los casos de prueba que verifique la aplicación como correcta.

Los requisitos trazables están escritos en una forma estructurada, de granularidad fina en lugar de párrafos largos. Se debe evitar agrupar múltiples requerimientos en una sola instrucción, los diferentes requisitos pueden rastrear hasta el diseño y elementos de código.

(Gottesdiener E. , 2005), recomienda las siguientes prácticas clave que promueven desarrollar requisitos de calidad:

- Desarrollar una visión clara para el producto final.
- Desarrollar una comprensión bien definida, del alcance del proyecto.
- Involucrar a los stakeholders durante el proceso de requisitos.
- Representar y descubrir los requisitos usando múltiples modelos.
- Documentar los requisitos con claridad y coherencia.
- Validar continuamente que los requisitos sean correctos con el enfoque del proyecto.
- Verificar la calidad de los requisitos frecuentemente.
- Priorizar los requerimientos y eliminar los innecesarios.
- Establecer una línea base para los requerimientos, ya que pueden servir para futuros proyectos.
- Rastrear los orígenes de los requisitos y la forma de vinculación con otros requisitos y con los elementos del sistema.
- Anticipar y gestionar todos los cambios de los requisitos.

1.6. Ingeniería de requisitos



Existen algunas definiciones de ingeniería de requisitos, entre ellas se mencionan:

- “La ingeniería de requisitos es ampliamente utilizada para indicar el tratamiento sistemático de los requisitos”. SWEBOOK, 2004: cap. 2.
- Pressman, 2010:83 “El espectro amplio de tareas y técnicas que llevan a entender los requerimientos.”
- Gottesdiener, 2009:16; “La Ingeniería de requisitos es una disciplina dentro de los sistemas y de la ingeniería de software que abarca todas las actividades y resultados asociados a definir un producto de las necesidades – es una de las mejores maneras de desarrollar requisitos de excelencia. En definitiva podríamos decir que la Ingeniería de Requisitos es el conjunto de actividades para descubrir, documentar y mantener un conjunto de requisitos.



Recuerde que la ingeniería de requisitos no es solo un proceso técnico; debido a que los requisitos del sistema están influenciados por las preferencias, prejuicios de los usuarios, aspectos políticos y legales de la organización; es decir proporciona un mecanismo apropiado para entender las necesidades del cliente, evaluar la factibilidad de las mismas, ofrecer una solución acertada y finalmente especificar esta solución sin ambigüedades.

En el proceso de la ingeniería de requisitos el equipo de desarrollo se enfrenta al problema de identificar correctamente los requisitos, pero para realizar este proceso no existe una técnica estandarizada y estructurada que garantice la calidad del resultado.

El proceso de la ingeniería de requisitos

La ingeniería de requisitos se compone del desarrollo y de la gestión de requisitos

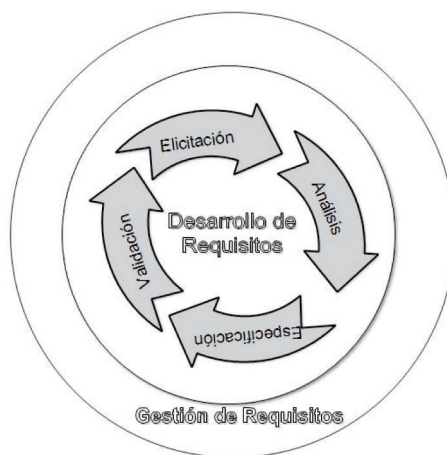


Figura 1.6: _Gestión de Requisitos

A continuación se analizan con mayor detalle cada una de las etapas:

Desarrollo de requisitos

En el proceso de desarrollo de requisitos las actividades que se deben ejecutar son:

- Elicitación
 - Análisis
 - Especificación
 - Validación de los requisitos.
- a) **Elicitación de requisitos:** En esta fase el equipo de desarrollo junto con los stakeholders identifican, articulan y entienden los requisitos de la aplicación a desarrollar. El descubrimiento de los requisitos implica entender el dominio de la aplicación, el servicio que va a prestar la aplicación, los problemas que se requieren resolver y las necesidades de los usuarios de la aplicación.

A esta fase también se la conoce como Descubrimiento de Requisitos; Según Gottersdiener (2009:17), esta fase consiste en: "Identificar las partes interesadas, la documentación y las fuentes externas de información sobre los requisitos, y solicitar los requisitos de esas fuentes".



Figura 1.7: Proceso de descubrimiento de requisitos

- b) **Análisis de Requisitos:** Es el proceso mediante el cual obtiene una comprensión precisa de los requisitos, se analizan las necesidades identificadas por parte de los stakeholders de tal forma que se obtiene el Documento de definición de requisitos Validado.



Figura 1.8: Proceso de análisis de requisitos

El análisis de requisitos comprende las siguientes actividades:

- Analizar los requisitos funcionales (RF) recolectados.
- Agrupar los requisitos funcionales recolectados y clasificarlos.
- De la clasificación de los requisitos determinar: los que no son necesarios, son incompatibles entre sí, no son completos, no son factibles y los que están repetidos.
- Aprobar la lista tentativa de requisitos funcionales definitivos por parte de los usuarios expertos en el dominio de la aplicación.
- Estructurar el contenido de Documentos de Definición de Requisitos (DDR).
- Elaborar el documento de Definición de Requisitos DDR con el listado de los requisitos funcionales; el cual debe estar aprobado por parte de los stakeholders.

En esta fase el cuidado se debe tomar para describir requisitos con exactitud, suficientemente como para permitir que los requisitos sean validados, su implementación sea verificada, y sus costes estimados. (IEEE Computer Society, 2004).

- c) **Especificación de requisitos:** “Se refiere a la producción de un documento a su equivalente electrónico que pueda estar sistemáticamente repasado, evaluado y aprobado”. (IEEE Computer Society, 2004).



La Especificación de requisitos se refiere a: “Diferenciar y documentar los requisitos funcionales y no funcionales, identificar los atributos de calidad, requisitos importantes y restricciones, y verificar que los requisitos documentados sean completos y no sean ambiguos”. (Gottesdiener, 2009:17)

En esta fase se elaboran tres tipos de documentos:

- **Documento de definición del sistema:** Define los requisitos del sistema de alto nivel desde las perspectiva del dominio, además se incluye información de fondo sobre los objetivos del sistema, su ambiente, declaración de limitaciones y los requisitos no funcionales.
- **Documento de requisitos del sistema:** En este documento se manifiesta lo que requieren los desarrolladores del sistema; además se incluyen los requerimientos del usuario para el sistema como una especificación detallada de los requerimientos del sistema.
- **Documento de requisitos de software:** Contiene una descripción completa de las necesidades y funcionalidades del sistema que se va a desarrollar además determina el alcance del sistema y la forma en la que realizará las funciones, definiendo los requerimientos funcionales y los no funcionales.

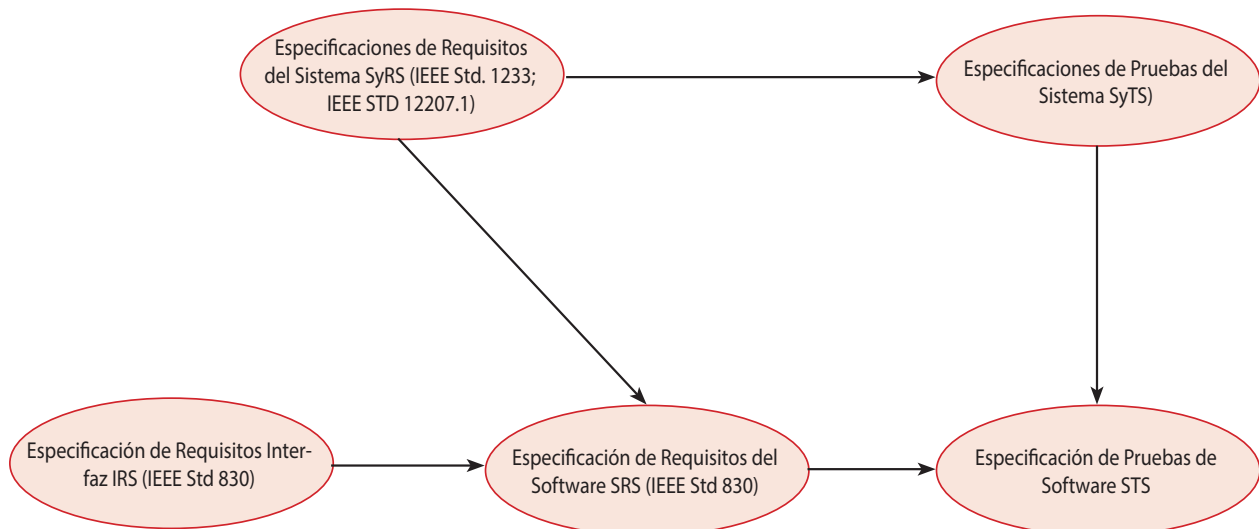


Figura 1.7: Documentos de la fase de especificación de requisitos



Ahora les invitamos a que revisen cada uno de los estándares IEEE que definen la documentación del proceso de ingeniería de requisitos. Esperamos su aporte al respecto a través del Entorno Virtual de Aprendizaje.

- d) **Validación de requisitos:** Los requisitos deben ser validados para asegurarse que el equipo de desarrollo de software haya entendido los requisitos; además se verifica que los documentos de los requisitos contempla estándares, es comprensible, constante y finito. Con esta premisa se puede concluir que la validación de los requisitos es el proceso de examinar el documento de los requisitos para asegurarnos que este define el software correctamente.

El proceso de validación implica las siguientes tareas: revisiones de requisitos, prototipado, validación del modelo, pruebas de aceptación, verificación de validez, verificación de consistencia, de integridad, realismo, verificabilidad; se profundizará en estos temas en el capítulo 6 de este texto guía.

Gestión de requisitos:

La gestión de requisitos es un conjunto de actividades que ayudan al equipo de desarrollo a identificar, controlar y seguir los requisitos y los cambios en cualquier momento.

La gestión o administración de requisitos se definió para sistemas grandes y cambiantes, debido a que durante el proceso del software, la comprensión del problema por el desarrollador está cambiando constantemente y estos cambios retroalimentan a los requerimientos. (Sommerville, 2005).

A continuación se detallan las actividades de esta fase:

- a) **Establecer una línea base:** Se establece una línea de base mediante la documentación del estado actual de las necesidades a un punto en el tiempo, para usar como punto de partida. La línea de base muestra una serie de requisitos con los atributos de estado acordados en un punto determinado en el tiempo y captura atributos importantes acerca de los requisitos. El desarrollo de una línea de base crea una referencia a utilizar para realizar un seguimiento de las necesidades evolucionan con el tiempo.
- b) **Control de cambios:** Se deben establecer mecanismos y políticas para reconocer, evaluar y decidir como integrar las nuevas necesidades e ir evolucionando hacia una línea de base de las necesidades existentes.
- c) **Seguimiento de requisitos:** Mediante la identificación y documentación de cómo los requisitos están relacionados de forma lógica y de los tipos de estos requisitos. La Trazabilidad de los requisitos le permite identificar la forma en el que los requisitos se relacionan con las metas y objetivos de negocio; y cuáles son los entregables del desarrollo futuro.

Para terminar esta sección sobre el proceso de ingeniería de software le invito a que revise el Anexo 1, dónde se indican los componentes del desarrollo de requerimientos y las actividades de gestión.



EJERCICIOS

Una vez analizadas las actividades del proceso de ingeniería del software, le invito a que complete la siguiente tabla en cuanto a los resultados de las actividades del desarrollo y la gestión de requisitos:



ACTIVIDAD	SALIDAS
Definición del alcance	• Visión del Producto • Glosario
Elicitación	• Categorías y perfiles de los stakeholders..
Análisis	•
Especificación	• Categorías y perfiles de los stakeholders..
Validación	
Gestión de Requisitos	• .

Si tuvo alguna dificultad de completar la tabla anterior, no dude en contactarse con sus profesores.

1.7. Stakeholders (interesados)

Los interesados o su equivalente en inglés stakeholders son personas u organizaciones que tienen influencia directa, indirecta, o se ven influenciados por un proceso de software. Muchos autores en los mismos proyectos de desarrollo utilizan el término en inglés stakeholders.

Los interesados más representativos y más fáciles de identificar son los clientes, usuarios finales y desarrolladores. Sin embargo existen otros que se relacionan con el proyecto como son: auditores, accionistas, proveedores, directivos, administradores, etc.

Cuando se desarrolla un proyecto de software inicialmente es sencillo identificar a los interesados obvios, como: el equipo de desarrollo, usuarios finales y clientes. Pero hay que descubrir otra gente que a simple vista no se relacionan con los anteriores pero que sus actividades giran en torno al sistema; como es el caso de la gente que está en el nivel más bajo del organigrama organizativo hasta aquellos que dirigen la organización.

En (Lamsweerde, 2009) se indica que para abordar una visión compartida de los problemas que se abordarán en la construcción del sistema se necesita de la cooperación de los interesados para obtener los requisitos completos, adecuados y realistas. Por tanto, hay que seleccionar a los interesados apropiados para definir un modus operandi que permitan superar dificultades en el proyecto. El desarrollo de requisitos y gestión de requisitos implica a muchos stakeholders en diversos cargos.

Recuerde que, por lo general un proyecto de software comienza con un patrocinador, que aprueba la justificación del proyecto y por lo tanto autoriza la ejecución del mismo; además intervienen el jefe del proyecto, analistas, los desarrolladores de software y evaluadores; a continuación se detallan las funciones de cada uno de ellos.



Lo invitamos a que identifique un proyecto de desarrollo de software e identifique los stakeholders del mismo.

Tabla 1.2: Funciones de los Stakeholders

Stakeholder	Función
Sponsor del proyecto	a) Asigna recursos (personal, materiales y fondos) para el proyecto. b) Asegura que las metas y objetivos del proyecto estén alineados con los de la organización. c) Guía apropiadamente la participación de los clientes y usuarios en el proyecto. d) Define o aprueba la visión y alcance del producto. e) Toma decisiones acerca del alcance del proyecto y problemas de versionamiento del producto. f) Resuelve conflictos en la priorización de requerimientos. g) Puede delegar autoridad para la aprobación de requisitos detallados a los expertos del negocio o administradores de empresas.
Gerente de proyecto o producto	a) Actúa como enlace entre el equipo de software y el administrador del negocio. b) Coordina la implicación del usuario. c) Asegura que el analista y los expertos tengan los recursos, herramientas, entrenamiento y conocimiento para desarrollar los requerimientos. d) Establece el proceso de control de cambios de los requerimientos. e) Supervisa la priorización de requerimientos. f) Monitoriza el progreso del desarrollo y gestión de requisitos.
Analista	a) Selecciona técnicas de elicitación. b) Colabora con los expertos del negocio. c) Coordina actividades de gestión de requisitos. d) Diseña modelos y documentos. e) Traduce requerimientos de usuario a especificaciones. f) Monitoriza el cambio de los requerimientos y coordina la negociación. g) Verifica que requerimientos son necesarios, correctos, completos y consistentes.



Experto en la materia	a) Provee detalles acerca de las necesidades de los usuarios. b) Provee detalles acerca de los procesos de negocio, reglas y datos. c) Identifica gente adicional que puede asesorar en los requerimientos. d) Representa a los usuarios quienes no pueden directamente involucrarse en el desarrollo. e) Identifica y consulta con otros expertos en la materia o asesores que tienen conocimiento relevante de los requerimientos. f) Aseguran que los requerimientos estén alineados a la visión del producto. g) Revisan la documentación de requerimientos para asegurarse que esta adecuada y completamente representando las necesidades de los usuarios. h) Participa en la creación o revisión de los modelos y documentos de requerimientos. i) Prioriza requerimientos.
Desarrollador y tester de software	a) Provee detalles acerca de restricciones de diseño y sugerencias respecto a la viabilidad de requerimientos no funcionales. b) Puede contribuir a escribir partes de la especificación de requerimientos de software. c) Revisa toda la documentación de requerimientos. d) Revisa las especificaciones de software para asegurarse de que se puede transformar en un diseño de software viable. e) Asegura que los requerimientos puedan ser probados.

Finalmente, a continuación se resumen los roles principales y lo que hacen los stakeholders en el proyecto de software.

Tabla 1.3: Roles de los stakeholders

	DESARROLLO DE REQUERIMIENTOS			GESTIÓN DE REQUERIMIENTOS
	Define requerimiento del negocio	Desarrolla requerimientos de usuario	Especificar requerimientos de software	
Sponsor del proyecto	Propietario aprobador	Revisor	Aprobador	Aprobador
Gerente del proyecto o producto	Productor	Revisor	Revisor	Revisor

Analista	Revisor	Productor	Productor	Productor
Usuario experto	Revisor	Propietario, aprobador, productor	Propietario Revisor	Propietario
Desarrollador y tester de software	Revisor	Revisor	Revisor Productor	Revisor

Ahora les invitamos a que realicen los siguientes ejercicios.



Actividad recomendada

Trabajemos ahora para reforzar los conocimientos adquiridos en esta primera unidad.

1. Describa tres tipos diferentes de requerimientos funcionales existentes en un sistema. Dé ejemplos de cada uno de estos tipos de requerimientos.
2. Describa tres tipos diferentes de requerimientos no funcionales existentes en un sistema. Dé ejemplos de cada uno de estos tipos de requerimientos.

Hemos concluido el estudio de la primera unidad. Conviene comprobar cuánto ha logrado asimilar de los temas revisados en esta parte, para lo cual es necesario que desarrolle el siguiente cuestionario.



Autoevaluación 1



En cada uno de los literales realice lo que se pide, conteste esta evaluación y luego compruebe las respuestas, las mismas que se encuentran al final de ésta guía

1. Elija la opción correcta.
1. La Ingeniería de Requisitos, es importante por:
 - a) Depende de esta etapa el éxito del producto de software.
 - b) Permite detectar solo las restricciones que puede tener un sistema.
 - c) Permite detectar únicamente los servicios que debe tener el sistema.
 - d) Permite definir las necesidades del sistema de una forma general.



2. **Indique a qué término o términos en el contexto de la ingeniería de requisitos corresponden los siguientes enunciados:**
- a) Cualquier persona que se beneficie en forma directa o indirecta del sistema en desarrollo.
 - b) Es un documento, un conjunto de modelos gráficos, un conjunto de escenarios de uso, que se crea para describir detalladamente los aspectos del software que se va a elaborar, antes de que el proyecto empiece.
 - c) Son declaraciones en lenguaje natural y en diagramas de los servicios que se espera que el sistema provea y de las restricciones bajo las cuales debe operar.
 - d) Establecen con detalle los servicios y restricciones del sistema.
3. **Los requerimientos funcionales describen**
- a) La funcionalidad o los servicios que se espera que este proveerá el sistema a desarrollarse.
 - b) Se refieren a las propiedades emergentes del sistema como fiabilidad, el tiempo de respuesta.
 - c) Son más críticos que los requerimientos funcionales particulares.
 - d) Proviene del dominio de aplicación del sistema y que reflejan las características de ese dominio.
4. **Señale la respuesta correcta. Los requerimientos de usuario**
- a) Describen con detalle las funcionalidades que proveerá el sistema a desarrollarse.
 - b) Describen de forma general la funcionalidad que proveerá el sistema a desarrollarse.
 - c) Se refieren a las propiedades emergentes del sistema como fiabilidad, el tiempo de respuesta.
 - d) Son más críticos que los requerimientos funcionales particulares.
5. **Los requerimientos no funcionales se los puede clasificar en:**
- a) Requerimientos del usuario y requerimientos del sistema.
 - b) Requerimientos del producto y requerimientos organizacionales.
 - c) Requerimientos del producto y requerimientos externos.
 - d) Requerimientos del producto, requerimientos organizaciones y requerimientos externos.

6. **La característica de factibilidad en un requisito individual se refiere a:**

- a) Si no necesita ampliar detalles en su redacción, es decir, proporciona la información suficiente para su comprensión.
- b) No tiene ambigüedades cuando se lo puede interpretar de una sola forma, y por lo tanto el lenguaje usado en su definición no causa confusiones al lector.
- c) Cuando se prescinde del mismo provoca una deficiencia en el sistema a construir; además cuando sus características físicas o factor de calidad no pueden ser reemplazados por otras capacidades del producto.
- d) Debe ser posible poner en práctica cada requerimiento dentro de las capacidades conocidas y las limitaciones del sistema en su entorno de operaciones.

7. **Un requisito individual es concreto cuando se refiere a:**

- a) No tiene ambigüedades cuando se lo puede interpretar de una sola forma, y por lo tanto el lenguaje usado en su definición no causa confusiones al lector.
- b) Cuando se prescinde del mismo provoca una deficiencia en el sistema a construir.
- c) Si no necesita ampliar detalles en su redacción, es decir, proporciona la información suficiente para su comprensión.
- d) Debe ser posible poner en práctica cada requerimiento dentro de las capacidades conocidas y las limitaciones del sistema en su entorno de operaciones.

8. **Un requisito tiene la característica de “consistente” cuando:**

- a) Ningún requerimiento o información necesaria deberían estar ausentes, sin embargo los requisitos que faltan son difíciles de detectar porque no están descritos.
- b) No entra en conflicto con otros requisitos del mismo tipo o con un mayor nivel de negocios, sistema o necesidades de los usuarios.
- c) Es capaz de revisar en el SRS cuando sea necesario y para mantener un historial de los cambios realizados de acuerdo a cada necesidad surgida; cada requisito debe aparecer solo una vez en el SRS.
- d) Puede estar vinculado a su origen hacia atrás y hacia adelante a los elementos de diseño y código fuente que aplicarla a uno de los casos de prueba que verifique la aplicación como correcta.



9. **Un requisito es trazable cuando:**

- a) Ningún requerimiento o información necesaria deberían estar ausentes, sin embargo los requisitos que faltan son difíciles de detectar porque no están descritos.
- b) Puede estar vinculado a su origen hacia atrás y hacia adelante a los elementos de diseño y código fuente que aplicarla a uno de los casos de prueba que verifique la aplicación como correcta.
- c) No entra en conflicto con otros requisitos del mismo tipo o con un mayor nivel de negocios, sistema o necesidades de los usuarios.
- d) Es capaz de revisar en el SRS cuando sea necesario y para mantener un historial de los cambios realizados de acuerdo a cada necesidad surgida; cada requisito debe aparecer sólo una vez en el SRS.

10. **Los requisitos deben ser validados:**

- a) Para asegurarse que el equipo de desarrollo de software haya entendido los requisitos.
- b) Para examinar el documento de los requisitos.
- c) Para ayudar al equipo de desarrollo a identificar, controlar y seguir los requisitos y los cambios en cualquier momento.
- d) Para la comprensión del problema únicamente por parte del desarrollador.



UNIDAD 2: Preparar el Escenario para el Desarrollo de Requerimientos



Antes de empezar con el desarrollo de requerimientos es necesario realizar ciertas actividades prioritarias, que permitan saber de forma preliminar el contexto de la solución que se va a desarrollar, además definir el problema visionando su solución dentro de los objetivos de negocio.

Las actividades preliminares tienen que ver con preparar el escenario sobre el cual se aplicará el proceso de licitación (obtención) de requerimientos. Es fundamental que para poder planificar una entrevista, un cuestionario, un taller, entre otras técnicas; se debe conocer que es lo que se preguntará, analizará o estudiará; además de identificar a los actores clave que pueden aportar al desarrollo del proyecto; por ello es importante realizar estas actividades preliminares.

Por lo tanto en esta unidad abordaremos las tareas preliminares que ayuden a enfocar la solución hacia los objetivos del negocio, de tal forma que permitan tanto a desarrolladores como a interesados (Stakeholders), contribuir en el desarrollo.

Recuerde, tal como se vio en la unidad anterior en el literal 1.6, el proceso de desarrollo de requerimientos consta de las fases de: Elicitación, Análisis, Especificación y Validación de forma interactiva; por lo que nos vamos a preparar antes de empezar con el proceso.

En la tabla 2.1. se indica las técnicas y herramientas que permiten sentar las bases para proceder con el desarrollo de los requerimientos. Esta tabla a sido tomada de (Gottesdiener E. , 2005).

Tabla 2.1: Técnicas y herramientas para definir el escenario para el desarrollo de requerimientos

Cuando necesite:	Entonces crear:
Definir la visión del producto	Visionamiento
Aclarar términos	Un glosario
Identificar los riesgos de los requerimientos	Una estrategia para mitigar los riesgos de los requerimientos

Por lo tanto la presente unidad se basará en el desarrollo de estas tres herramientas: Visionamiento, Glosario y Estrategia para los riesgos. Cabe señalar que previamente tanto el sponsor como el gerente de desarrollo habrán realizado el acercamiento respectivo a los Stakeholders de alto nivel que conocen aunque de una forma general el contexto de negocio; esto con el objeto de tener elementos de por dónde empezar.



Para orientar adecuadamente el estudio y desarrollo de los componentes se desarrollarán algunos ejemplos que tienen que ver con el desarrollo de una aplicación para la UTPL, donde intervienen los estudiantes de modalidad abierta. A continuación se hace una breve descripción de la misma.



Ejemplo: Caso de estudio

Sistema de gestión de trámites de la UTPL (autotrámites)

Los estudiantes de modalidad abierta de la UTPL, realizan trámites o reclamos que tienen que ver con:

- Cambios de centro.
- Cambios de lugar y fecha de evaluación.
- Registro de notas que el profesor olvidó registrar o registró de forma incorrecta.
- Gestionar una beca.
- Anulación de matrícula.
- Anulación de asignaturas.

Cada uno de estos trámites requiere de documentos que deben ser entregados en el centro al que pertenece el estudiante y estos a su vez entregados a la sede en Loja, para que dependiendo del trámite el departamento de Coordinación Académica canalice a las instancias correspondientes para que se proceda a resolver tal solicitud.

Esto evidentemente que al hacerlo de forma manual requiere de tiempo y el uso de varios recursos. El más afectado es el estudiante que tiene que esperar un tiempo excesivamente largo para saber si le aprobaron o no la solicitud.

Ante esta situación la Dirección de Modalidad Abierta de la UTPL ha creído conveniente desarrollar un sistema para automatizar estos procesos, en donde el estudiante se acerque al centro de la UTPL de su localidad y registre su solicitud adjuntando toda la documentación que se requiere de forma digital, y que el sistema gestione de acuerdo al trámite para que se notifique a quien corresponda el análisis y resolución del mismo.

Con esto se quiere evitar el molesto envío de la documentación de forma física para ahorrar tiempo, y que al momento que se registre la solicitud, se notifique de la misma a quienes son responsables del caso; esto permitiría al estudiante y demás interesados consultar en el sistema del estado del trámite.



De igual forma se requiere que desarrolle la especificación de requerimientos para un caso real. A continuación se dan algunas pautas para su desarrollo.

**Actividad recomendada:** Caso de desarrollo local**Caso de desarrollo local**

Se va a desarrollar una solución para un caso de su localidad. Lo llamaremos “Caso de desarrollo local”, con la finalidad de afianzar los conocimientos de forma práctica y acercados a la realidad.

Para lo cual deberá:

- Escoger un lugar donde le puedan facilitar información de los procesos que realizan.
- El lugar puede ser: el sitio donde usted trabaja, una institución, un departamento, un local comercial, un almacén, una institución educativa, etc.
- El lugar deberá tener un conjunto de procesos y actividades definidas, en lo posible que no sea muy complejo o grande.
- Explique al dueño, encargado o responsable de la entidad que requiere de cierta información con fines educativos.
- Pida solamente la información que sea necesaria.

Conforme avancemos con la materia se deberá desarrollar ciertos componentes como parte de proceso de desarrollo de requerimientos.

Estas actividades de desarrollo se combinarán con los trabajos a distancia y sobre todo con el Entorno Virtual de Aprendizaje, así que estar muy atentos a las disposiciones que se establezcan en el mismo por parte del profesor.



Si tiene algún inconveniente o no está seguro con el lugar escogido para desarrollar el caso de su localidad, pida asesoría a su profesor tutor para despejar las dudas, es necesario que tenga una descripción general de lo que se hace en el lugar. Puede hacerlo mediante el EVA, correo electrónico o teléfono.

2.1. Visionamiento

El visionamiento es una declaración concisa que resume el propósito a largo plazo y la intención del nuevo producto. Esta declaración podría reflejar una vista equilibrada que satisface las necesidades de diversos stakeholders. Desde un punto de vista general puede verse como algo idealista, pero debe basarse en realidades propias de la organización en la cual se va a desarrollar el producto. Este visionamiento puede ser parte de otros documentos como por ejemplo en el documento de inicio del proyecto, en alguna carta del proyecto, pero principalmente en el documento de visión del proyecto.

Como resultado del visionamiento tenemos:

- Asegura que las definiciones y problemática del producto se alinea con las metas y objetivos del negocio.



- Identificar los stakeholders del producto en forma general.
- Descripción del estado del negocio y cómo los usuarios se beneficiarán cuando el proyecto termine.
- Facilita a los miembros del equipo dar una descripción sencilla del proyecto.

Para tener una idea más clara de lo que hace esta actividad, es necesario plantearse las siguientes preguntas:



- ¿Quién comprará o usará este producto?
- ¿Qué hará el producto para sus Stakeholders?
- ¿Cuáles son las razones para comprar o utilizar este producto.
- ¿Cuál será el estado del entorno operacional o de negocios cuando el producto esté disponible?
- ¿Cómo se distinguirá en el mercado?

Como puede ver, cada pregunta requiere de captar información a un nivel general de situaciones fundamentales para empezar a desarrollar los requerimientos. A continuación se describen cuatro pasos que se deberán realizar para lograr el visionamiento de la solución. Le sugerimos considere las actividades de cada uno de los pasos que se indican.

1. Definir lo siguiente:

- **Cliente Objetivo (Target customer):** Describe las personas que usarán o comprarán el software.
- **Declaración de necesidad u oportunidad:** Describe lo que hace el cliente (Target customer) y explica como este producto le podría ayudar.
- **Nombre del producto:** Proporciona el nombre del producto que se creará.
- **Categoría del producto:** Describe el tipo de producto que construirá. Las categorías del producto podrían incluir: aplicaciones de software de gestión interna, software integrado, software de juegos, dispositivos de hardware, o sistemas complejos.
- **Los principales beneficios o las razones convincentes para comprar:** Describe qué podría hacer el producto para el cliente o la justificación para haber comprado el producto.
- **Principal alternativa competitiva, sistema actual, o proceso manual actual:** Describir los principales productos disponibles que compiten, o el sistema, o el proceso que el producto reemplazará.
- **Declaración de la diferencia de los productos primarios:** Explicar las diferencias entre el producto que se construirá y la competencia.

2. Crear la declaración de la visión mediante la introducción de los términos definidos en la siguiente plantilla.

Para <Cliente objetivo> **quien** <declaración de la necesidad u oportunidad>, **el** <nombre del producto> **es un** <categoría del producto> **que** <principales beneficios o razones convincentes para comprar>.

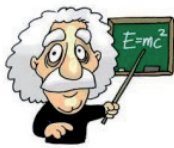
A diferencia de <principal alternativa competitiva, sistema actual, o proceso manual actual>, **nuestro producto** <declaración de las diferencias del producto primario>.

3. Revisar la declaración de la visión y verificar que se alinea con las metas y objetivos de la organización.

- Contar con un sponsor para asegurar que la visión se ajusta con los objetivos organizacionales y departamentales.
- Contar con un equipo de trabajo, idealmente en colaboración con el sponsor del proyecto, con la finalidad de revisar la declaración de la visión como sea necesario.

Adicionalmente es necesario hacer la declaración del problema, que consiste en una descripción del problema actual que la empresa está experimentando y aclara lo que una solución exitosa podría ser. Esto es útil cuando la solución incluye la ampliación de un software existente o cuando la implementación del producto crea una necesidad para un proceso de cambio del negocio. Use la siguiente plantilla para hacer una declaración del problema.

El problema de <Insertar el planteamiento del problema> **afecta** <nombre de las personas afectadas, organización, o grupo de clientes>. **El impacto de esto es** <nombre del impacto (por ejemplo, malas decisiones, los sobrecostos, información o procesos erróneos, tiempo de respuesta lento a los clientes, etc)>. **Una solución exitosa sería** <describir la solución>.



Ejemplo:

A continuación se pone a consideración un ejemplo del caso de estudio (Autotrámite): Inicialmente se procede a definir los elementos que permiten declarar la visión, para posteriormente acoplar a la plantilla.

**1. Definir lo siguiente:**

- **Cliente:**

Personal administrativo y estudiantes de la modalidad de estudios a distancia de la Universidad Técnica Particular de Loja.

- **Necesidad u oportunidad:**

Registran y resuelven los trámites que se generan en cada centro asociado mediante un proceso definido para cada trámite.

- **Nombre del producto:**

Sistema de automatización de trámites.

- **Categoría del producto:**

Aplicación Web.

- **Beneficios o razones convincentes:**

- Registra las solicitudes de trámites directamente en el sistema desde cualquier centro.
- Define un flujo de documentación desde los centros asociados a la sede central.
- Notifica a las personas involucradas en el trámite sobre las tareas que debe realizar.
- Permite conocer sobre el estado de un trámite en cualquier momento.

- **Principal alternativa competitiva, sistema actual, o proceso manual actual:**

- Los trámites existentes tienen que enviarse de forma física a la sede en Loja.
- No existe un flujo de actividades debidamente controlado para cada trámite.
- Tanto el estudiante como los involucrados en el trámite no conocen a ciencia cierta sobre el estado del mismo.

- **Declaración de la diferencia de los productos primarios:**

Permitirá:

- Que las solicitudes de trámites académico/contable se registren en el sistema.
- Que se notifique a los involucrados (personal UTPL-Estudiantes) del inicio, gestión y finalización de un trámite.
- Que se identifique a la persona que está atendiendo un trámite.
- Que se pueda obtener información académico/contable..
- Que se puedan ejecutar las solicitudes de forma automática de acuerdo al trámite solicitado.

2. Defina el visionamiento

Para personal administrativo y estudiantes de la modalidad de estudios a distancia de la Universidad Técnica Particular de Loja **quienes** registran y resuelven los trámites que se generan en cada centro asociado **el** sistema de automatización de trámites **es** un una aplicación Web **que** registra las solicitudes de trámites directamente en el sistema desde cualquier centro, define un flujo de documentación desde los centros asociados a la sede central, notifica a las personas involucradas en el trámite sobre las tareas que debe realizar y permite conocer sobre el estado de un trámite en cualquier momento.

A diferencia del proceso actual que requiere que los trámites existentes tienen que enviarse de forma física a la sede en Loja, no existe un flujo de actividades debidamente controlado para cada trámite y tanto el estudiante como los involucrados en el trámite no conocen a ciencia cierta sobre el estado del mismo.

3. Definición del problema

El problema de la atención de trámites académicos y contables que generan los estudiantes no tienen una adecuada atención, lo que genera excesiva demora en la resolución de los mismos debido a que no existe un flujo definido para el envío-recepción y a la falta de políticas para su ejecución.

Afecta a: Estudiantes, Directora MAD, Director Administrativo Financiero, Coordinación Académica MAD – Loja, Coordinación Académica MAD – CRQ, Directores de Escuela, Contabilidad MAD, Gestión de Procesos, Atención al Estudiante y Secretarías.

Cuyo impacto es:

- El promedio de tiempo para resolver un trámite es alto.
- El estudiante requiere mucha gestión para obtener el resultado de su trámite, causando malestar e insatisfacción.
- Se desperdicia tiempo, en tareas relacionadas a tratar de ubicar un trámite y conseguir su estado.
- Algunos trámites causan congestión en el procesos de matrículas, pues al no poderse resolver ágilmente, los estudiantes deben regresar al centro universitario varias veces entorpeciendo procesos de trabajo importantes (Matrículas).

Una solución satisfactoria permitirá:

- Que la atención de los casos se realice mediante un sistema de fácil uso, de tal forma de que se de solución a todos los trámites.
- Que las solicitudes de trámites sean registradas directamente en un sistema.
- Que las solicitudes de trámites fluyan electrónicamente hacia a las diferentes personas que deban revisarlo antes de su resolución.
- Que la documentación relacionada al trámite esté asociada a éste de forma digital de manera que pueda ser visualizada directamente en el sistema.
- Que se pueda conocer el estado en que se encuentra un trámite.
- Que se pueda disponer de estadísticas que permitan tomar decisiones para optimizar los procesos relacionados.
- Que exista seguimiento adecuado de los trámites, estableciendo tiempos de respuesta.



Actividad recomendada

Realice el visionamiento y la definición del problema de acuerdo a la plantilla que se describió anteriormente, para el caso de desarrollo local.

Si desea tener una retroalimentación sobre el desarrollo de esta actividad, envíe al profesor tutor la actividad desarrollada y solicite su comentario. Será necesario que describa de forma breve el contexto del caso que está desarrollando. Puede enviar por correo electrónico.

2.2. Glosario

Al glosario se lo considera como un diccionario de términos comúnmente relevantes con respecto al producto que se construye, se mejora o se compra. Los términos del glosario se utilizan a lo largo de todo el proyecto. Su uso es necesario ya que establece un vocabulario común para los términos clave que ayuda a los miembros del equipo a entender estos términos. Ya que diferentes Stakeholders pueden usar el mismo término para explicar diferentes cosas, o pueden usar diferentes términos para expresar la misma cosa, causando confusión y gasto de energía a la hora de comunicar los requerimientos.

Como resultado de construir un glosario se tiene:

- Proveer un conocimiento compartido del dominio del problema.
- Permitir a los dueños del negocio informen al personal técnico acerca de importantes conceptos del negocio.
- Provee los fundamentos para definir modelos de requerimientos tales como reglas de negocio, modelos de datos y casos de uso.
- Ahorra tiempo y esfuerzo durante el desarrollo de requerimientos, eliminando malos entendidos de lo que realmente significan los conceptos del negocio.

Para lograr estos resultados se debe hacer lo siguiente:

1. Determinar quien en el proyecto puede identificar una lista inicial de términos.

- Incluir expertos en la materia.
- Incluir los analistas de datos de la organización de desarrollo de software (que a menudo son los expertos en la definición de términos del negocio).

2. Identificar términos importantes que sean relevantes para el dominio del negocio.

- Examine los nombres en los documentos existentes del proyecto (por ejemplo, en la carta del proyecto, en la declaración de la visión, y declaración del problema).
- Incluya términos relacionados con la tabla que se indica.

Tabla 2.2: Términos clave y relevantes que se pueden utilizar para construir el glosario.

Término	Ejemplo
Negocio, o partes del negocio	clientes, compradores, clientes potenciales, vendedores, proveedores, distribuidores y proveedores de servicios
Lugares y ubicaciones	direcciones y sitios
Eventos	puestos de trabajo, órdenes de trabajo, solicitudes, envíos y producción
Acuerdos	contratos, estimaciones, y descuentos
Cuentas	cuentas de clientes, registros financieros, y balances
Productos y servicios	servicios a los empleados, materiales y bienes
Mercados y prospectos	partes de negocio, proveedores, contratistas, clientes, vendedores, y distribuidores
Recursos	edificios, bienes, máquinas, dispositivos, contratistas, empleados y sus definiciones

3. Elabore el borrador de definiciones de los términos

- Oriente cada definición a lectores quienes no tienen experiencia en el negocio o conocimiento acerca de los términos.
- Incluya "Alias" o nombres alternativos cuando múltiples términos tienen el mismo significado.
- Incluya acrónimos (siglas) después de cada término donde se utilice.
- Añada ejemplos para clarificar su utilidad. Por ejemplo use calificadores (como es: "cliente potencial" en lugar de "cliente") en términos que ayuden a clarificar.
- Pedir a una persona bosquejar una definición para cada término.

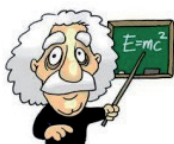
4. Identificar múltiples stakeholders para revisar las definiciones y revisar los términos como sea necesario para llegar a un acuerdo común para cada término.

- Reunirse con los de stakeholders para socializar, definir, revisar y aprobar los términos del glosario.

El glosario del proyecto puede ser creado por secciones de acuerdo a los términos del proyecto, como por ejemplo: roles del proyecto, nombres de organización, métodos de software, herramientas, etc.



Además un glosario evoluciona a medida que se van estableciendo los requisitos, por lo que alguien deberá mantener el glosario al día de manera que se use en todos los modelos de requisitos y en las discusiones de requerimientos. Idealmente esto lo podría realizar alguna persona del negocio, sin embargo un analista es un buen candidato para realizar esta tarea.

**Ejemplo:**

A continuación se presenta un ejemplo de definición de términos como parte del glosario.

Término	Definición	Alias	Ejemplos
Registro de notas	Proceso que permite que los profesores ingresen las notas de los estudiantes. Se pueden presentar los siguientes casos: - Nota no ingresada en el sistema - Nota registrada en el sistema difiere de la nota de la hoja de resumen de notas. - Nota de la evaluación difiere de la hoja de resumen de notas.	Registro de notas	
Registro de beca	El registro extemporáneo de becas se da cuando en el Departamento de becas no se ha ingresado la beca de un estudiante, a pesar de que este ha cumplido con todos los requisitos que se necesita para aplicar a una beca.	Asignación de beca	
Anulación académica	La anulación académica se presenta cuando el estudiante presenta repeticiones en determinadas asignaturas (3 en adelante); en esta caso se realiza una autorización para hacer la anulación del periodo académico en el que el estudiante reprobó la asignatura solicitada para anulación. Se debe tener presente que en esta anulación no existe devolución económica	Anular asignatura	
Digitalización	Proceso por el cual se capturan los datos de un formato físico y se lo expresa datos en forma digital.	Documento digital	
Cliente	Una persona o organización, interna o externa, quienes tienen la responsabilidad financiera del sistema. El cliente es el receptor de los productos desarrollados y sus entregables. Ver también stakeholder	Estudiante	
Defecto	Una anomalía dentro de un producto. Ejemplos como: omisión e imperfecciones encontradas durante fases tempranas del ciclo de vida. Un defecto puede ser cualquier tipo de novedad que se requiera registrar y resolverla. Ver también Requerimiento de cambios.	Error	



Actividad recomendada

Continuando con el problema escogido en el visionamiento, elabore un glosario de acuerdo a la tabla que se especificó anteriormente.

2.3. Mitigar el riesgo en los requerimientos

Para mitigar los riesgos en los requerimientos es necesario establecer estrategias que permitan evaluar los requerimientos relacionados con el riesgo e identificar acciones para evitar o minimizar estos riesgos. Los riesgos en los requerimientos son sucesos o condiciones que ponen en peligro el desarrollo satisfactorio del producto. Los riesgos deben ser evaluados, rastreados y controlados a lo largo del proyecto, esto ocasiona que se pueda tener un gran impacto de éxito en el proyecto.

La estrategia para mitigar el riesgo en los requerimientos, permite:

- Identificar los riesgos que podrían impedir el desarrollo efectivo y la gestión de requisitos.
- Involucrar múltiples stakeholders que categorizan el riesgo de cada requerimiento de acuerdo a su grado de ocurrencia y a su impacto.
- Al equipo comunicar honestamente acerca de los potenciales obstáculos.
- Identificar alternativas para administrar los riesgos por parte del equipo.

Para identificar los riesgos es necesario realizar lo siguiente:

1. Reunir los stakeholders para revisar y adaptar una lista de potenciales riesgos de requerimientos.

- Para esto es necesario usar una lista inicial de riesgos comunes, como por ejemplo:
 - Falta de involucramiento del usuario.
 - Expectativa del cliente poco realista.
 - Los desarrolladores añaden funcionalidades innecesarias.
 - Constante cambio de requerimientos.
 - Pobre análisis del impacto cuando las necesidades cambian y evolucionan.
 - Uso de nuevas técnicas o herramientas de requerimientos.
 - Requisitos poco claros, ambiguos.
 - Requisitos contradictorios (conflictivos).
 - Falta de requisitos.
 - Lluvia de ideas para identificar riesgos adicionales en base a la experiencia del equipo, como de la cultura de la empresa y su medio ambiente

2. Clasificar los riesgos

- Analizar cada riesgo de acuerdo a su probabilidad y su impacto.
- **La probabilidad.** Riesgo estimado para causar un problema. Usar una escala o rango como es:



- a. Bajo: Remota posibilidad que el riesgo se realizará. Entre 0% --- 25%.
- b. Medio: Probabilidad moderada que ocurra el riesgo. Entre 26% --- 74%.
- c. Alto: Gran probabilidad o certeza de que el riesgo ocurra. Entre 75% --- 100%
- **El impacto** es el grado en que el riesgo afectan negativamente al proceso de requisitos.
 - a. Bajo: Puede presentar algún impacto, insignificante.
 - b. Medio: Impacto manejable o marginal.
 - c. Alto: Impacto crítico o catastrófico. Principales problemas que necesitan ser analizados.
- Clasificar cada riesgo de acuerdo a cada dimensión (probabilidad e impacto)
- 3. **Representar gráficamente las clasificaciones y ponerse de acuerdo en los riesgos que se abordarán.**

Una forma de representar la clasificación final podría ser:

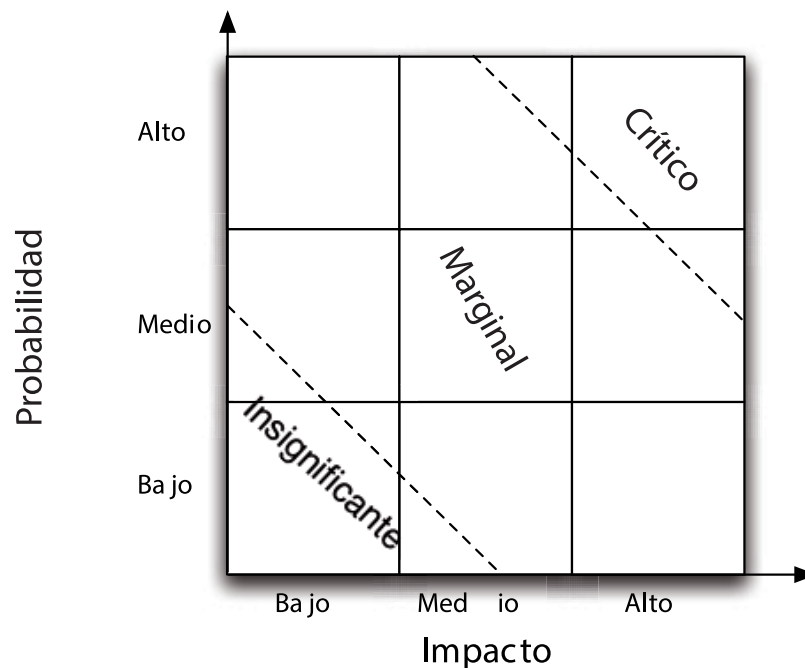


Figura: 2.1: Relación entre la probabilidad e impacto de los riesgos. (PMI-PMBOOK, 2008)



Actividad recomendada

Le sugerimos que proponga una representación de las clasificaciones de los riesgos, este aporte lo deberá entregar medio del EVA.

4. Determinar las formas de controlar, evitar o mitigar los posibles riesgos críticos

- Asignar cada riesgo crítico a un miembro del equipo quien tendrá la responsabilidad de monitorear el riesgo. Identificar las acciones que él o ella tendrá, los recursos necesarios para llevar a cabo las acciones, y la manera que él o ella comunicará las acciones al equipo.
- Asegurar que el sponsor y el líder del equipo estén de acuerdo con las acciones.
- Asegurarse que los miembros del equipo entienden como sus acciones afectan sus requerimientos.
- Monitorear los riesgos a lo largo del desarrollo y gestión de requisitos.
- Analizar y adicionar nuevos riesgos a los requerimientos como ocurran, y actualizar la estrategia de mitigación de riesgo como sea necesario.

En la siguiente tabla se indican algunos riesgos y estrategias que comúnmente ocurren en los proyectos de desarrollo.

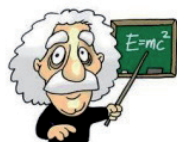
Tabla 2.3: Riesgos y estrategias

Riesgos comunes en los requerimientos	Estrategias de mitigación
Falta de participación del usuario	• Identificar a los interesados del producto. • Crear un plan de participación de interesados. • Usar técnicas de elicitación que atraigan a los usuarios en el proceso
Expectativas poco realistas de los clientes.	• Crear la visión del producto. • Desarrollar modelos de alcance del proyecto. • Validar requerimientos con prototipos operacionales.
Desarrolladores agregan funcionalidades innecesarias .	• Crear la visión del producto. • Priorizar requerimientos.
Constante cambio de requerimientos.	• Desarrollar modelos de alcance. • Crear una línea base para requerimientos y establecer mecanismos de control de cambios.
Pobre análisis del impacto cuando las necesidades cambian y evolucionan.	• Crear una línea base y seguimiento de requerimientos. • Formalizar documentos de requerimientos.
Uso de nuevas técnicas o herramientas de requerimientos.	• Adaptar los procesos de requerimientos para el proyecto. • Conducir una retrospectiva de los requerimientos.
Requisitos poco claros, ambiguos.	• Desarrollar la visión del producto. • Desarrollar múltiples modelos de requerimientos. • Validar los requerimientos.
Requisitos contradictorios (conflictivos).	• Formalizar el documento de visión. • Validar los requerimientos con el modelo de validación.
Falta de requisitos	• Desarrollar múltiples modelos de requerimientos. • Verificar la falta de requerimientos con el modelo de validación.



Muchas de las acciones de mitigación de riesgos involucra el establecimiento y seguimiento de las buenas prácticas de gestión de requerimientos. En proyectos pequeños, se puede gestionar los riesgos de manera informal siempre y cuando el equipo revise los riesgos de forma periódica.

Para tener un control efectivo de los riesgos es necesario poner en práctica las técnicas de mitigación y seguimiento de riesgos, revisando periódicamente para verificar si se están tomando las acciones correctas y si los nuevos riesgos están siendo considerados.



Ejemplo:

A continuación se presenta una tabla donde se especifican ciertos elementos que permiten catalogar a un riesgo, así como la estrategia de mitigación.

Factor de riesgo	Probabilidad	Impacto	Estrategia de mitigación	Responsable
Falta de disponibilidad del director de la MAD para aclarar el alcance	Media	Alto	Llevar a cabo un taller de visionamiento con el actual director y crear modelos de alcance en el taller.	Ellen Marshall (Gerente proyecto)
No involucramiento del contratista	Media	Alto	Llevar a cabo una visita (por ejemplo en el lugar de trabajo hacer el seguimiento por un medio día). Llevar a cabo una entrevista con el contratista cada cierto periodo de tiempo.	Adam Faith (Analista).
Poco interés de las secretarías de los centros	Media	Bajo	Realizar un taller para indicar las bondades de un sistema automatizado.	Ellen Marshall (Gerente proyecto)
Poco interés de las secretarías de escuela	Alto	Alto	Realizar reuniones informativas sobre el nuevo proceso automatizado para resolver los trámites de los estudiantes.	Ellen Marshall (Gerente proyecto) y el sponsor del proyecto.



Actividad recomendada

De acuerdo al ejemplo anterior defina las estrategias a utilizar para mitigar los riesgos en los requerimientos de acuerdo al caso de su localidad.

Hemos concluido el estudio de esta unidad, por lo que nos conviene comprobar cuanto hemos asimilado de los temas tratados para lo cual es necesario desarrollar las autoevaluaciones de conocimiento.



Autoevaluación 2



Conteste de acuerdo a lo que se pide, luego compare sus respuestas con el solucionario que se encuentra al final de esta guía.

Lea detenidamente la afirmación y escriba una V si considera que es Verdadera o una F si no lo es, en el paréntesis respectivo.

1.	El visionamiento es una declaración que resume el propósito a largo plazo del nuevo producto.	()
2.	El visionamiento es visto como algo idealista.	()
3.	El visionamiento es único y por ser definido de forma general no puede ser parte de otros documentos.	()
4.	El visionamiento asegura que las definiciones y problemática estén alineados con las metas y objetivos del negocio.	()
5.	Los términos del glosario se utilizan a lo largo de todo el proyecto.	()
6.	El glosario no puede ser creado separado por secciones.	()
7.	Para mitigar los riesgos en los requerimientos es necesario establecer estrategias que permitan evaluar los requerimientos.	()
8.	Al definir la estrategia para mitigar el riesgo se debe involucrar múltiples stakeholders que categoricen el riesgo para cada requerimiento.	()
9.	La clasificación del riesgo se puede hacer solamente de acuerdo a la probabilidad de ocurrencia.	()
10.	Una buena alternativa para mitigar los riesgos en los requerimientos es el establecimiento y seguimiento de las buenas prácticas de gestión de requerimientos.	()



2. Complete la siguiente tabla, con la herramienta apropiada

Para preparar el escenario para el desarrollo de requerimientos, es necesario desarrollar ciertas actividades.

Cuando necesite:	Entonces crear:
Definir la visión del producto	
Aclarar términos	
Identificar los riesgos de los requerimientos	



Ir a solucionario

UNIDAD 3: Elicitación de Requerimientos



Uno de los aspectos más cruciales y desafiantes en el desarrollo de proyectos de software es la definición de los requisitos para la aplicación propuesta, por lo que en la presente unidad se abordará la primera fase del proceso de desarrollo de requerimientos. La elicitación consiste en identificar las fuentes de obtención de requisitos y luego utilizando técnicas evocar esas fuentes.

La captura de requisitos es una actividad humana intensa que se basa en la participación de los stakeholders, como fuente principal de obtención de requisitos.

La captura de los requisitos es responsabilidad del analista, pero puede estar implicado otro personal técnico que desee beneficiarse de un conocimiento mucho más profundo de las necesidades de los interesados.



Tal como se indica en la unidad 1, existen diversos problemas que ocasionan que un proyecto de desarrollo de software no llegue a feliz término, entonces como responder a la pregunta ¿Por qué es difícil la captura de requisitos?

Los clientes y usuarios generalmente no entienden como diseñar y desarrollar el software, por lo que no estarán en capacidad de especificar sus necesidades de software de tal forma que entiendan los desarrolladores. Por su parte, los desarrolladores a menudo no entienden los problemas y necesidades de los clientes y usuarios, como para especificar las necesidades.

Entre las dificultades típicas, tenemos:

- Necesidades diferentes y a veces conflictivas de los diferentes tipos de usuarios.
- Requisitos no declarados o asumidos por parte de los stakeholders.
- Identificar a los stakeholders clave.
- Incapacidad para imaginar nuevas o diferentes formas de usar el software.
- Incertidumbre para adaptarse a las necesidades cambiantes del negocio.
- Contar con un gran número de requerimientos sumamente relacionados.
- Tiempo limitado para obtener los requisitos. Stakeholders ocupados – importantes.
- Superar la resistencia al cambio.

Para superar estas dificultades, se debe fomentar un ambiente de cooperación y comunicación entre los desarrolladores, clientes y usuarios, para asegurar que se eliciten los requerimientos apropiados. ¿Cómo elicitar requerimientos de software?



Para obtener los requisitos de forma eficaz, será necesario:

1. Elegir y planificar las técnicas de elicitación de requerimientos.
2. Establecer metas, expectativas y preparar
3. Elicitar los requerimientos
4. Verificar y corregir los hallazgos
5. Repetir los pasos 1-4 para profundizar el entendimiento de los requerimientos por parte del equipo.

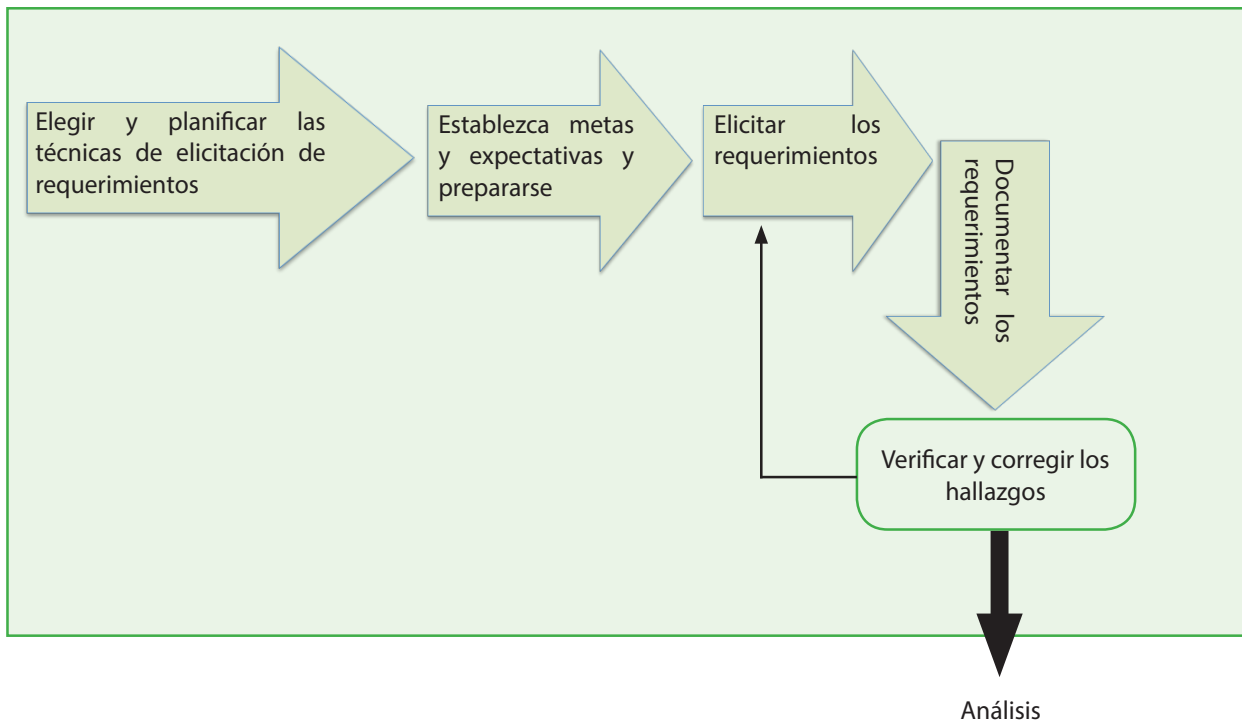


Figura 3.1: Proceso para elicitar requerimientos. (Gottesdiener E. , 2005)

Para que el proceso de elicitación sea manejable y no se convierta en una tarea difícil de sobrellevar por el equipo de desarrollo, es necesario utilizar ciertas herramientas y técnicas que faciliten la captura de los requisitos. A continuación en la tabla se indican estas técnicas y herramientas que sugiere. (Gottesdiener E. , 2005)

Tabla 3.1: Estrategias y herramientas para elicitar requerimientos

Cuando necesite: Entonces crear:	
Identificar fuentes	Una lista de fuentes de requerimientos
Identificar stakeholders del producto	Categorías de los stakeholders
Describir necesidades y criterios de éxito de los stakeholders	Perfiles de stakeholders
Revisar técnicas de elicitación	Identificar combinaciones de técnicas de elicitación: entrevistas, prototipos exploratorios, talleres facilitadores, focus groups, análisis de tareas de usuario, estudio de documentación existente
Plan de elicitación	Plan de elicitación de stakeholders

3.1 Listar las fuentes de requerimientos

El listado de fuentes es un inventario de personas, documentos específicos, y fuentes de información externa de donde se elicitarán los requerimientos. Se realiza para identificar fuentes potenciales de documentación de requerimientos y permitir a los analistas, elicitar, revisar, documentar, y verificar información de requerimientos con los stakeholders. Por lo que se deberá: identificar las fuentes de información y facilitar la planificación para una eficiente participación de los stakeholders.

Para realizar un proceso apropiado de elicitación se requiere:

- Identificar los stakeholders relevantes que proporcionarán los requerimientos.
- Identificar la documentación que se pueda usar como fuente de información para los requerimientos.
- Identificar fuentes externas de información.

A continuación se realiza una descripción más al detalle.

1. Identificar los stakeholders relevantes que proporcionarán los requerimientos.

- Asegúrese de considerar a todos los Stakeholders del proyecto. Incluya a los clientes quienes auspician y dirigen el desarrollo de software, los usuarios que interactuarán directamente con el software, y otros quienes tienen conocimiento o apuestan por el proyecto.
- Desarrolle un plan de elicitación para cada stakeholder. Tener en cuenta que los stakeholders están a menudo ocupados y necesitan de un aviso previo para participar en el levantamiento de requerimientos.



2. Identificar la documentación que se pueda usar como fuente de información de requerimientos.

- Documentación de interfaces del sistema existentes.
- Requerimientos de cambio, listas de defectos de software, registro de quejas de clientes, lista de inconvenientes.
- Guías de usuario, materiales de entrenamiento, guías y procedimientos de trabajo.
- Documentación help desk.
- Código de sistemas existentes.

3. Identificar fuentes externas de información

- Departamentos o compañías que proveen servicios de estudios de mercado y análisis de la industria.
- Análisis de productos de software del mercado
- Materiales de ventas, marketing y comunicaciones
- Regulaciones, guías y leyes provenientes de agencias gubernamentales o cuerpos regulatorios.

3.2. Categorías de los Stakeholders

Las categorías de stakeholders son grupos o individuos que están interesados en el producto de software que se está desarrollando. Es necesario definir las categorías de los Stakeholders para entender quienes están interesados o influyen en el proyecto, quienes utilizarán el producto y sus salidas; y a quien de alguna manera afectaría el producto. Por lo tanto estos grupos e individuos necesitarán estar informados acerca del progreso, conflictos, cambios y prioridades del proceso de desarrollo del producto.

Esto se hace:

- Especificando los tipos de personas quienes tienen requerimientos y necesidades a las que se denominan como involucrado o stakeholders.
- Distinguiendo a los clientes de los usuarios del producto.
- Clarificando que personas y agencias externas se deben consultar.



Tenga presente que si no logra una comprensión completa por parte de los Stakeholders puede ocurrir la falta de requisitos o erróneos o el desarrollo de la solución de forma incorrecta. Por tanto es necesario que se asegure de que todos los interesados comprenden además de incluir a todos los interesados antes de proceder con el desarrollo del software.

Esta actividad permite responder a las siguientes preguntas:



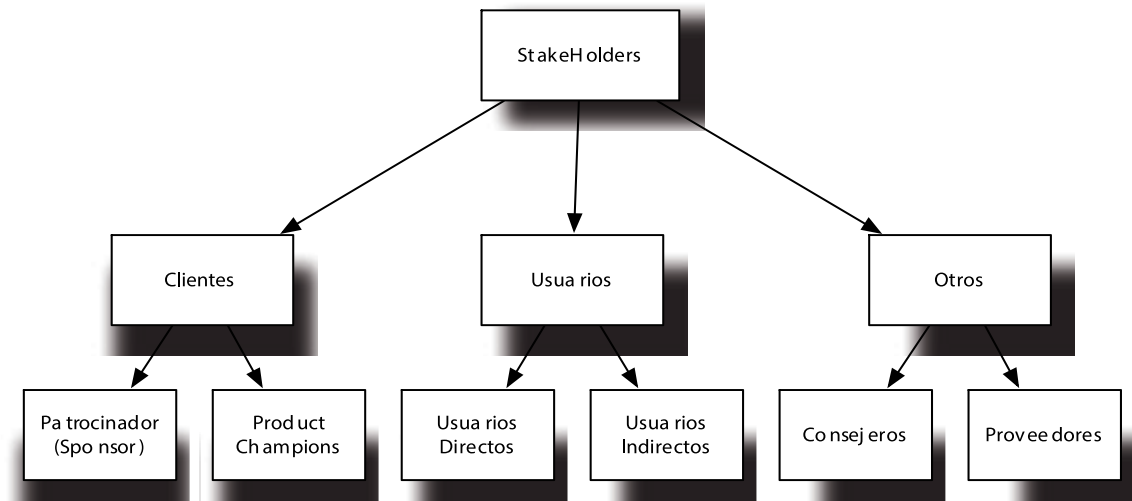
- ¿Quién afecta o es afectado por el sistema?
- ¿Quién o que interactúa con el sistema?

- ¿Quién tiene los conocimientos relevantes de los requerimientos?

¿Cuáles son las categorías de los Stakeholders?

Hay tres categorías de actores: clientes, usuarios y otras partes interesadas, tal como se indica en la figura 3.1.

Figura 3.2: Categoría de los Stakeholders. (Wiegers, 2003)



Clientes

Son responsables de aceptar y pagar por el producto de software.

Sponsor	• Es quien autoriza o legitima el desarrollo del producto contratando o pagando el proyecto. • La influencia del sponsor es a veces necesaria para lograr el involucramiento de los stakeholders. • Asegúrese de revisar la lista de stakeholders con el patrocinador para mantenerlo informado acerca de los Stakeholders pertinentes.
Product champions	• Es quien asegura que el software cumple con las necesidades de múltiples comunidades de usuarios. • Identifica los usuarios quienes deben participar en el desarrollo de requerimientos para asegurar que los requerimientos correctos son obtenidos. • Pueden ser los usuarios finales del producto. • El producto puede tener múltiples champions si se tienen múltiples tipos de usuarios directos.



Usuarios

Son aquellos que están en contacto con el producto de software o son afectados por este de alguna manera.	
Usuarios directos	Son las partes (por ejemplo: gente, organizaciones, componentes del sistema o dispositivos que directamente interactúan con el software) que directamente se involucran con el software (por ejemplo: una persona que solicita la información del sistema a través de una interfaz de usuario, un sistema que envía o recibe archivos de datos, o un dispositivo que envía o recibe señales o mensajes).
Usuarios indirectos	- Quienes no interactúan directamente con el software pero pueden entrar en contacto con los productos generados por el sistema (reportes, facturas, bases de datos, etc.). - También puede incluir a personas quienes pueden ser afectadas por las decisiones o acciones realizadas como resultado de las salidas del sistema.

Otros stakeholders

Poseen conocimiento acerca del producto o un interés en su desarrollo o mantenimiento.	
Consejeros	- Tienen información relevante acerca del software. - Puede incluir a expertos. - A menudo conocen las reglas de negocio vitales que deben ser incorporadas dentro del software, aun sino interactúan con el producto de software. - Asegúrese de identificar a los consejeros e involucrarlos en el proceso de elicitación de requerimientos.
Proveedores	- Quienes diseñan y producen el software transformando los requerimientos en el producto final - Incluye a los miembros del equipo (analistas, diseñadores, desarrolladores, finanzas, ventas y departamentos de soporte), proveedores de software y subcontratistas.



Los asesores a menudo conocen las reglas del negocio por lo que es vital incorporar a los asesores en el proceso de captura de requisitos, y evitar demasiado esfuerzo. La categoría de otros Stakeholders puede incluir a partes internas (por ejemplo, la gente de la fabricación legal, finanzas, ventas, y los departamentos de apoyo) y externos (por ejemplo, la gente de los organismos reguladores, auditores, y el público en general) a la organización del proyecto. Un solo actor puede pertenecer a varias categorías. Por ejemplo, un entrenador de producto de software puede ser un usuario directo (que tiene que usar el sistema como parte de su responsabilidad del trabajo), un usuario indirecto (que accede a los materiales de capacitación relacionados con el sistema), y un asesor (que proporciona asesoramiento en temas de usabilidad con una versión anterior del software).

Para realizar la categorización de los Stakeholders, es necesario realizar lo siguiente:

1. Identificar stakeholders ya sean clientes, usuarios u otros stakeholders.
 - Registre los stakeholders como roles en cada una de las categorías, más que como personas específicas. Recuerde que el mismo rol puede aparecer en varias categorías.
 - Recordar que el mismo rol puede aparecer en múltiples categorías.
2. Revisar el registro de categorías de stakeholders con los stakeholders del proyecto para asegurar que está completa y precisa.
3. Revisar la lista como sea necesario y compartir la misma con el equipo entero.

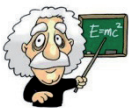


Use el listado de roles genéricos de stakeholders que se indican a continuación como punto de partida para ayudarlo a categorizar. Es necesario traducir estos roles genéricos en categorías específicas para el proyecto (por ejemplo: para el proyecto de Autotrámites, el rol del “Experto financiero” sería “Asesor fiscal de Autotrámites”).



Roles genéricos de Stakeholders

Auditor	Consultor medio
Comprador	Usuario multilingüe
Usuario de oficina	Personal de operaciones
Especialista en comunicación	Especialista de embalaje
Especialista en contratos	Especialista en nómina o salarios
Analista de servicio al cliente	Especialista en adquisiciones
Administrador de base de datos	Instalador del producto
Analista de documentación	Experto regulador
Especialista en medio ambiente	Reportero crítico
Experto financiero	Especialista en ventas
Futuro o posible usuario directo	Inspector de seguridad
Supervisor del gobierno	Programador
Usuario invitado	Administrador del sistema
Usuario deshabilitado	Arquitecto del sistema
Especialista en materiales peligrosos	Usuario del sistema
Especialista en mesa de ayuda	Especialista en seguridad
Especialista en recursos humanos	Especialista en soporte
Experto legal	Escritor técnico
Administrador crítico	Entrenador
Especialista de fabricación	Especialista en usabilidad
Especialista en marketing	Usuario formador



Ejemplo:

Categoría de stakeholders para el sistema de registro de trámites.

Clientes		Usuarios		Otros Stakeholders	
Sponsor	Product Champion	Usuarios Directos	Usuarios Indirectos	Consejeros	Proveedores
• Director de MAD	• Director de procesos	• Secretaria centro • Secretaria de carrera • Director de escuela • Profesor • Coordinación académica MAD • Contabilidad • Centro de evaluaciones	• Estudiante • Atención al estudiante • Call Center	• SRI • Senescyt • Auditor externo • Consultor	• Gerente del proyecto. • Analistas • Desarrolladores • Administrador de base de datos



Actividad recomendada

De acuerdo a la lista de los roles genéricos y al ejemplo anterior determine las diferentes categorías de Stakeholders para el caso de desarrollo local.

3.3 Perfil de los stakeholders

El perfil de los Stakeholders es una descripción que caracteriza a cada stakeholder y explica su relación con el proyecto. Es necesario definir el perfil para entender los intereses, preocupaciones y criterios de éxito del producto para cada stakeholder, para descubrir las posibles fuentes de conflicto entre las partes interesadas de los requisitos, y para poner de relieve temas relacionados con los requisitos que pueden requerir más tiempo y atención. Los perfiles de los Stakeholders también pueden revelar los posibles obstáculos para la implementación exitosa del producto y le ayudará a definir la cantidad de participación de cada Stakeholder debe tener en el levantamiento de requerimientos.

Los perfiles de los stakeholders permiten:

- Educar al equipo acerca de las expectativas del cliente.
- Provee al equipo un alto nivel de entendimiento de las necesidades del usuario.
- Destaca potenciales obstáculos en el proceso de aceptación del producto de software por parte de los stakeholders.



Las preguntas claves que deberá responder con esta herramienta son:

- ¿Cuáles son las responsabilidades de los Stakeholders clave en relación con el sistema en desarrollo o los cambios implementados?



- ¿Qué motivaciones, deseos y esperanzas tienen los stakeholders para el producto de software?
- ¿Qué características o capacidades de software deben estar presentes para cada uno de los stakeholders para ver el producto como un éxito?
- ¿Qué obstáculos, limitaciones, o los factores que limitan cada actor se prevé para él mismo o para otros que puedan amenazar la implementación exitosa?
- ¿Qué nivel de confort tienen los stakeholders con la tecnología?
- ¿Hay algún trabajo especial o condiciones ambientales que podrían afectar la capacidad de los stakeholders en utilizar eficazmente el sistema?

Para especificar el perfil de cada stakeholder deberá:

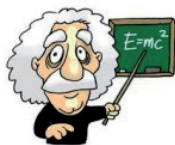
1. Escribir un breve perfil de cada actor. Describa su:

- **Rol:**
Lista la categoría de stakeholder (por ejemplo, el sponsor (patrocinador), Product Champion, usuario directo, usuario indirecto, asesor o proveedor) al que pertenece.
 - **Responsabilidades:**
Describa brevemente el rol de cada stakeholder en relación con el proyecto.
 - **Intereses:**
Liste las necesidades de los stakeholders, deseos y expectativas para el producto (por ejemplo, los intereses de un patrocinador puede incluir aumento de los ingresos, reducción de costos, mejora de los servicios y el cumplimiento de las normas reglamentarias).
 - **Los criterios de éxito:**
Describir las características o capacidades que el producto debe tener para ser considerado como exitoso.
 - **Preocupaciones:**
Lista de todos los obstáculos, limitaciones, o los factores limitantes que puedan impedir o inhibir al stakeholder a aceptar el producto.
 - **Capacidad técnica:**
Describir al usuario directo el grado de familiaridad con la tecnología.
 - **Características del entorno de trabajo y limitaciones:**
Describir las condiciones de trabajo relevante que pueda afectar el uso del sistema (por ejemplo, un entorno de trabajo ruidoso o el uso del móvil o al aire libre).
- 2. Incluir los perfiles de los Stakeholders en el documento de requerimientos de usuario (si se utiliza) y el documento de especificaciones de requisitos de software.**



Si los perfiles contienen gran cantidad de información, documente un perfil por cada stakeholder en una tabla o sección en el documento de requisitos correspondientes.

La combinación de las categorías para proyectos pequeños o de menor complejidad, se puede crear una versión más corta de los perfiles de los stakeholders mediante la combinación de los intereses y los criterios de éxito.



Ejemplo:

Perfiles de Stakeholders del sistema de trámites.

Stakeholder	Rol	Responsabilidad	Intereses	Criterios de éxito	Preocupación	Competencias técnicas/ Relación de ambiente de trabajo
Director de la MAD	- Sponsor - Usuario indirecto	Aprobar (proyecto)	Construya un sistema de acuerdo a las normas de la MAD	Satisfacer las necesidades de los estudiantes	Tiempo de construcción de la aplicación	N/A
Coordinación Académica	- Usuario directo - Product champion	Aprobar proceso	Perfeccionar y validar los procesos	- Trámites registrados en cada centro. - Cada actor realiza las actividades. - Satisfacción de los estudiantes	- Involucramiento de los actores que intervienen en el proceso. - No se registre de forma apropiada la información	Liberación del producto.



Actividad recomendada

Complementando la actividad anterior determine los perfiles para el caso de desarrollo de su localidad, basándose en la tabla del ejemplo anterior.

3.4 Identificar técnicas combinadas de elicitación

Ahora vamos a enfocarnos en las técnicas que permitan obtener información relevante a partir de los stakeholders, por lo que es importante analizar tales técnicas que la ingeniería de requisitos considera apropiadas para obtener información.

Además recalcar que estas técnicas son complementarias entre sí, ya que en determinado momento alguna de ellas servirá para determinar situaciones generales, otra servirá para detalles concretos. Por esta razón le sugiero que cuidadosamente analice cada una de ellas. Entre las técnicas más comunes tenemos:

- Entrevistas con los stakeholders.
- Talleres.



- Prototipos de exploración.
- Entrevistas a grupos focales.
- Observación.
- Análisis de tareas del usuario.
- Estudio de la documentación existente.
- Encuestas.



Es necesario que desarrolle su plan de elicitación para seleccionar y combinar las técnicas aplicables de esta lista. Por ejemplo se pueden desarrollar talleres con prototipos de exploración para encontrar errores en los requisitos y validarlos, o realizar un análisis mediante la observación a las tareas del usuario para ver en lo que se tiene que centrar.

3.4.1 Entrevistas con los Stakeholders

Las entrevistas son consideradas una de las técnicas de mayor importancia para la captura de requisitos ya que es una de las formas de comunicación más naturales entre las personas. El propósito de una entrevista está orientado a establecer un enfoque sistemático diseñado para obtener información de una persona o grupo de personas en un ambiente formal o informal, haciendo preguntas pertinentes y apropiadas. Las entrevistas son conversaciones cara a cara en la que un entrevistador hace preguntas para obtener información del entrevistado.

De acuerdo a (Lamsweerde, 2009) las entrevistas pueden ser de dos tipos: estructuradas (con preguntas preparadas con anterioridad), o no estructuradas (sin preguntas predefinidas).

- **Entrevistas estructuradas:**

Se dan cuando el entrevistador ha elaborado un conjunto de preguntas acorde a un propósito asociado con el entrevistado. Algunas de las preguntas pueden ser abiertas o preguntas en donde la respuesta tenga un conjunto de posibilidades conocidas.

- **Entrevistas sin estructurar:**

Donde el entrevistado y el entrevistador sin ninguna pregunta predefinida discuten de temas de interés abiertamente.

Frecuentemente las entrevistas se realizan para:

- Obtener información general sobre las necesidades de los interesados
- Preguntar a los clientes y usuarios sobre el estado de sus necesidades.
- Ayudar a descubrir conflictos en los requerimientos de software.

El hacer una entrevista y que esta sea exitosa depende de ciertos factores:

- Nivel de comprensión del dominio por parte del entrevistador
- La experiencia del entrevistador

- Habilidad del entrevistador en la documentación de los debates
- Disposición del entrevistado para proporcionar la información pertinente
- Grado de claridad del entrevistado acerca de lo que la empresa requiere del sistema.
- Armonía entre el entrevistador y el entrevistado

¿Cómo hacer la entrevista?

Deberá:

1. Identificar a las personas que le gustaría entrevistar

- Elija a las personas transversalmente. Incluya sponsors, clientes, y usuarios con experiencia en el tema.
- Establecer los entrevistados y entrevistadores con los que se empezará. Asegúrese que los entrevistadores se sentirán a gusto entrevistando a directivos y clientes, y viceversa.

2. Prepare las preguntas de la entrevista

- Aclare el objetivo de cada entrevista (por ejemplo: para obtener información de antecedente y características de alto nivel del software, o para obtener un conocimiento detallado del flujo de trabajo del usuario o las necesidades de datos).
- Elabore las preguntas de la entrevista desde lo general a lo más particular. Organice de forma fácil empezando con preguntas de hecho. (por ejemplo: ¿Cuál ha sido su participación en el proyecto hasta la fecha?. Al final las preguntas más difíciles como preguntas de interpretación.



En este aspecto debe adaptar las preguntas para captar la atención del entrevistado. Para un alto ejecutivo o a los clientes, pregunte, “¿Por qué es este proyecto (o software) importante para usted (o sus clientes)?” O “¿Qué debe hacer este producto para que usted lo llame exitoso?” Para los usuarios que van a interactuar directamente con el software, por ejemplo, “Hábleme de su forma ideal de <realizar una tarea>”.

Incluya preguntas de contexto libre (preguntas de alto nivel sobre el producto y el proceso) al inicio de la entrevista y todas las entrevistas abiertas. Esto permite entender el panorama. Por ejemplo:

- ¿Qué problema resuelve este sistema?
- ¿Qué problemas puede crear este sistema?
- ¿Cual es el ambiente que se encuentra este sistema?
- ¿Qué grado de precisión es requerido o deseado en este producto?

Incluir meta-preguntas (preguntas acerca de preguntas) que le permitan ajustar sus preguntas durante la entrevista. Por ejemplo:

- ¿Estoy haciendo demasiadas preguntas?
- ¿Mis preguntas son relevantes?
- ¿Es usted la persona adecuada para responder estas preguntas?
- ¿Quién más podría responder estas preguntas?



- ¿Hay algo más que le podría preguntar o responder?

3. Programe la entrevista y organice la logística para la reunión.

- Encuentre un lugar donde la entrevista no sea interrumpida.
- Prepare a los entrevistados. Indique el objetivo de la entrevista, si es posible proporcione las preguntas de la entrevista con uno o mas días de anticipación. Además pídale a reunir todos los documentos que pueden ser útiles para consultar durante la entrevista (por ejemplo, manuales, referencias, planes o informes).
- Asegúrese de que los entrevistadores están familiarizados con la terminología de la empresa. Comparta un glosario con los entrevistados para asegurar que están de acuerdo con los términos y definiciones.
- Asegúrese de comunicar al entrevistado cuánto tiempo tomará la entrevista. Típicamente una entrevista debería durar entre 45 y 60 minutos.

4. Realizar la entrevista

- Preséntese y realice una pregunta inicial.
- Indique a las personas entrevistadas que usted tomará notas durante la entrevista.
- Practique la escucha activa. Por ejemplo, repetir las respuestas en sus propias palabras y mantenga sus ojos comprometidos con el entrevistado.
- Evite preguntas capciosas (Por ejemplo, ¿No cree usted que...? O ¿por qué no hace sólo ...?)
- Sea flexible, haciendo las preguntas necesarias.
- Finalice la entrevista con un agradecimiento, e indique los siguientes pasos que realizará. Pida permiso para hacer el seguimiento de las preguntas si fuera necesario.

5. Documente los resultados

- Revise sus notas inmediatamente después de la entrevista, cuando la información está aún “fresca” en su mente.
- Conjuntamente con los entrevistados para resolver algún conflicto de información.
- Analizar sus notas de varias entrevistas para descubrir patrones y conflictos.
- Generar un conjunto de modelos o necesidades de forma textual para revisiones preliminares por parte del equipo, basado en las entrevistas.



Grabación de audio y vídeo puede parecer eficaz, pero generalmente no lo son. El tiempo que tarda en escuchar o ver cada entrevista y tomar notas es mal utilizado. Use cinta sólo si usted desea aprender sobre estilos de entrevista o si los comentarios incluidos son importantes para su proyecto. Si usted decide grabar las entrevistas debe obtener el permiso del entrevistado previamente.



Actividad recomendada

Considerando los lineamientos anteriormente descritos elabore un guión para realizar una entrevista al encargado del departamento financiero de una determinada entidad.

3.4.2 Talleres

Un taller es una reunión de las partes interesadas cuidadosamente seleccionados que trabajan juntos bajo la guía de un experto y neutral que produce y documenta modelos de requerimientos. Los talleres se realizan de manera rápida y eficiente para definir, refinar, priorizar y **cerrar las necesidades con los usuarios**. Compromete a los usuarios a descubrir las necesidades del proceso e identifica a los usuarios del producto.

Para tener éxito en la realización del taller, se deben realizar los siguientes pasos:

1. Determinar el propósito del taller y los participantes.

- Defina los roles de las personas que participarán en el taller.
 - Participantes: Usuarios y expertos.
 - Facilitadores, grabadora, sponsor y observadores.
- Talleres pequeños, con una docena o menos de participantes.
- Utilice un experto si tiene un grupo con problemas políticos o conflictos.

2. Identifique las reglas del taller

- El facilitador indique las reglas para los participantes al inicio del taller, algunas reglas pueden ser:
 - Inicio y fin a tiempo.
 - Estar preparados.
 - Centrarse en los intereses y no en las posiciones.
 - Compartir toda la información relevante.
 - ¡Participa!.
- Confirmar las reglas. Asegúrese que los propios participantes hagan cumplir las reglas, con la ayuda del facilitador.
- Defina reglas de decisión y un proceso de toma de decisiones para el taller. Por ejemplo: “La persona a cargo toma la decisión final después de consultar al grupo”.

3. Defina los entregables del taller

- Incluya entregables tangibles (como son los modelos de análisis), como también modelos intangibles (como son las decisiones).
- Determine cómo va a saber cuando los entregables son bastante buenos. Hacer que los resultados sean lo suficientemente específicos.

4. Diseño de la agenda

- Crear una agenda que abra el taller, conduzca actividades de descubrimiento de requerimientos, y cierre el taller.



- Envíe la agenda a los participantes antes del taller.
- Pida a los participantes traer los documentos importantes de la empresa.

5. Conducir la reunión

- Pídale al sponsor del proyecto o al patrocinador del proyecto abrir la reunión con una breve descripción del propósito del taller.
- Utilice un plan de interacción con los participantes.
- Use una variedad de medios y herramientas para captar el interés de las personas durante la reunión.
- Use herramientas adecuadas (portátil, proyector).
- Hacer una corta retrospectiva de forma periódica en el taller para tener una retroalimentación del mismo.
- Imprimir los resultados del taller.
- Pedir al sponsor e interesados clave para que se unan al final del taller para permitir a los participantes presentar los resultados y compartir problemas que necesitan ser resueltos.
- Cierre el taller asignando cuestiones pendientes a participantes específicos con fechas de vencimiento y planes de comunicación.

6. Seguimiento, próximos pasos y acciones

- Hacer responsables a los participantes de dar seguimiento a las tareas asignadas.
- Evaluar el taller luego de completar la elicitación de requerimientos.



Además de la realización de taller para proyectos de desarrollo interno o compra de software para uso interno, los esfuerzos para el desarrollo de software comercial se puede llevar a cabo talleres facilitados por la participación de usuarios sustitutos o realizar talleres en los sitios de trabajo de los clientes.

También se pueden realizar talleres para revisar prototipos.



Actividad recomendada

Considerando los lineamientos anteriormente descritos elabore un guión para realizar un taller para el departamento financiero de una determinada entidad.

3.4.3 Prototipos de exploración

Los prototipos exploratorios, son versiones parciales o preliminares del software creado para explorar o validar los requisitos.

Cuando los usuarios tienen dificultad para expresar descripciones del sistema es posible mostrar un esquema reducido del sistema que ayude a visualizar la forma en que se vera. En concreto es una interfaz de usuario que se integra con casos de uso, escenarios, datos y reglas de negocio. (Lamsweerde, 2009).

Los prototipos son un software de rápida implementación para definir aspectos de “cómo es el sistema”. Las clases de prototipos dependen de los aspectos que se desean prototipar

Tipos de prototipos

Los prototipos horizontales y verticales abordan el contenido del sistema propuesto de diferente manera. Los prototipos horizontales imitar los interfaces de usuario o una porción superficial de la funcionalidad del sistema. Los prototipos verticales se sumerge en los detalles de la interfaz, la funcionalidad, o ambas.

Tabla 3.2: Tipos de prototipos

Tipo	Prospecto	Evolutivo
Horizontal	• Aclarar los requisitos funcionales • Identificar las funciones que faltan. • Explorar la interfaz de usuario o enfoque de navegación	• Implementa importantes casos de uso. • Implementa casos de uso adicionales dependiendo de la prioridad • Perfeccionar los sitios Web • Adaptar el sistema a la rápida evolución de las necesidades
Vertical	• Demostrar la viabilidad técnica de rendimiento, facilidad de uso, u otros atributos de calidad	• Implementa la compleja funcionalidad del software de comunicación (por ejemplo basada en web o cliente-servidor) y las capas. • Implementar y optimizar los algoritmos básicos. • Probar y ajustar el rendimiento.

Para realizar los prototipos realice lo siguiente:

1. Seleccione una parte del alcance del producto para el prototipo.
 - Escoja los requerimientos que están poco claros, contradictorios, o que impliquen la interacción del usuario.
 - Escoja un pequeño conjunto de funcionalidad.
 - Use algún requisito representado de forma textual u otras representaciones.
2. Determinar si se creará un prototipo de prospecto o prototipo evolutivo.
 - Aclarar el propósito del prototipo con los usuarios y miembros del equipo.
 - Establecer las condiciones técnicas para el desarrollo del prototipo, en su caso.



3. Diseñar y construir el prototipo.

- Utilizar los datos reales de los clientes (no datos ficticios), si es posible. (Tenga cuidado con la privacidad o los problemas de seguridad cuando se utilizan datos reales.) Cuando construya la interfaz de usuario, considere adicionar datos de ejemplo que muestren a los usuarios que están probando el prototipo.

4. Conduzca la evaluación del prototipo con los usuarios.

- Empiece con una declaración sobre los objetivos del prototipo y los próximos pasos.
- Demostrar o simular interactuar el usuario con el sistema. Mostrar las maquetas de las interfaces de alto nivel.
- Registre los problemas de usuario y sugerencias.
- Que la revisión del prototipo dure dos horas o menos.
- Crear un resumen de las conclusiones y pasos a seguir, y acordar un calendario para la próxima revisión (si es necesario).

5. Documente los resultados.

- Corregir los modelos y documentos relacionados con los requisitos.



Debido a que los prototipos de exploración se desarrollan a menudo utilizando diferentes herramientas a las utilizadas para generar el software, tenga cuidado de no permitir a los usuarios o miembros del equipo sacar conclusiones sobre el desempeño esperado del producto final basado en el rendimiento del prototipo.



Actividad recomendada

Realice un prototipo de interfaz de usuario que permita registrar en una ficha, los datos de cada estudiante. Considerar criterios de presentación y distribución, así como obligatoriedad o no de los mismos.

3.4.4 Grupos focales

Los grupos focales son entrevistas en grupo cuidadosamente planeado que plantean problemas y hacen preguntas abiertas para obtener información de los participantes. A menudo consisten en una serie de reuniones entre un moderador y un grupo de seis a doce personas. La participación es voluntaria y los resultados son confidenciales. (Gottesdiener E. , 2005).

Se lo hace para obtener la reacción del usuario a nuevos productos o ideas de productos en un ambiente controlado, y revelar la información subjetiva y la percepción acerca de las características del producto. Los grupos de enfoque exploran opciones requisitos y obtienen las reacciones a los nuevos componentes e interfaces. También ayudan a las organizaciones de desarrollo de productos a priorizar las necesidades e identificar las áreas de estudio de forma cualitativa o cuantitativa.

Para realizar las entrevistas a estos grupos focales, se debe realizar lo siguiente:

1. Definir los objetivos y los participantes del grupo de enfoque.
2. Planificar y organizar la logística para la sesión.
3. Dirigir el grupo de enfoque.
4. Analizar y documentar la información recogida.



Se pueden llevar a cabo un grupo de enfoque exploratorio haciendo preguntas abiertas acerca de un producto ya existente y luego pidiendo a los participantes imaginar un mejor producto en el futuro. Pídeles que describan su uso, funcionalidad y características.

3.4.5 Observación

Esta técnica se caracteriza por entender una tarea mediante la observación directa, a la vez permitir que alguien que conoce la tarea nos explique. Esto provoca que se evalúe el entorno de trabajo de las personas. Esta técnica se la utiliza especialmente cuando se tiene la intención de mejorar el proceso en curso o el proyecto. (Lamsweerde, 2009).

La observación se aplica especialmente a los procesos de negocio o procedimientos de trabajo que implica a personas que no pueden darse cuenta de lo que hacen los demás o tiene dificultad para explicarlo. En algunos proyectos es importante comprender el procesos actual para hacer los ajustes necesarios.

La técnica de observación considera dos enfoque básicos: Enfoque pasivo o invisible y el enfoque activo o visible.

- **Pasivo:**

En este enfoque la ingeniería de requisitos no interfiere con la gente involucrada en las tareas, sino que son analizadas en base a notas o video cámaras, etc., Estos registros deben ser analizados e interpretados apropiadamente. Cuando un sujeto está realizando una tarea y al mismo instante va explicando sus actividades se denomina Análisis de Protocolo. De igual manera los estudios etnográficos son otro caso particular de observación pasiva donde el ingeniero de requisitos intenta durante largos períodos descubrir propiedades emergentes de grupos sociales relacionados con los procesos observados. En la observación también se analiza las actitudes de los participantes, sus reacciones a situaciones específicas, sus gestos, conversaciones, chistes, etc.

- **Activo:**

En este enfoque el ingeniero de requisitos se involucra en las tareas, pasando a formar parte del equipo de trabajo.

Para realizar esta actividad es necesario realizar los siguientes pasos:



1. Preparar la observación: consiste en determinar las personas a las cuales se les observará. En grandes empresas se tendrá que escoger apropiadamente la muestra de personas. Además como parte de esta fase se debe elaborar los cuestionarios que se aplicarán durante o después del desarrollo de actividades de la gente.
2. Observar: El observador indica a la persona que esta siendo observada, y:
 - Asegura al usuario que su trabajo no está siendo cuestionado, sino que la observación de su trabajo y documentación servirá para el análisis de requisitos.
 - Se informa al usuario que el observador está presente sólo para estudiar sus procesos y se abstendrá de intervenir.
 - Indicar al usuario que puede detener el proceso de observación en cualquier momento si consideran que está interfiriendo con su trabajo.
 - Sugerir al usuario que puede “pensar en voz alta” cuando este trabajando, esto como una manera de compartir sus intenciones, retos y preocupaciones.

En cuanto a la conducta de la observación.

- Tomar notas detalladas.
 - Si se utiliza la observación con el enfoque activo, hacer preguntas deductivas acerca de por qué son así los procesos y tareas que se llevan a cabo.
3. Post observación, documentación y confirmación:
 - Obtener respuestas a las preguntas nuevas que surgieron durante la observación.
 - Proporcionar al usuario un resumen de las notas lo antes posible para su revisión y aclaración.
 - Revisar los hallazgos con todo el grupo para asegurarse que los detalles finales representan a todo el grupo y no solamente a los individuos seleccionados.



Un analista puede actuar como un aprendiz, para realizar tareas de un usuario bajo la supervisión de un usuario experimentado y bien informado. El “aprendiz de analista” podría conocer las necesidades de los usuarios para hacer las necesidades reales de trabajo y aprender a ser un usuario experimentado.



Actividad recomendada

Continuando con el caso de estudio de su localidad, ubique a una persona a la que pueda aplicar esta técnica y luego documente lo ocurrido (Utilice cualquier formato para documentar).

1.4.6 Estudio de la documentación existente

Basados en (IIBA, 2005), el propósito del análisis de documentos es la obtención de los requisitos mediante el estudio de la documentación disponible sobre las soluciones existentes y la identificación de información relevante. Este análisis comprende: los planes de negocio, estudios de mercado, contratos, solicitudes de propuestas, declaraciones de trabajo, procedimientos, guías de capacitación, competencia técnica del producto, informes de problemas, bitácoras de las sugerencias de los clientes, entre otros. Al identificar y consultar a todas las probables fuentes de información dará lugar a una mejor cobertura de las necesidades, siempre y cuando la documentación esté al día.

Para realizar el análisis de los documentos se reúnen todos los detalles de las soluciones existentes, como son: las reglas del negocio, las entidades y los atributos que deberán ser considerados en la nueva solución o ser actualizados en la solución actual.

De forma general en el análisis de documentos es necesario considerar los siguientes:

1. Identificar las fuentes de documentación adecuadas para su uso.
 - Utilice sólo una documentación fiable para descubrir las necesidades y generar modelos de análisis.
 - Localizar la documentación del usuario que podrían estar en forma física (por ejemplo, manuales de capacitación y guías de procedimiento), así como en forma flexible (por ejemplo, pantallas de ayuda y mensajes de error).
 - Considere usar:
 - Documentación de respaldo.
 - Pantallas de ayuda.
 - Descripciones de trabajo.
 - Manuales de operación y lineamientos.
 - Planes de estrategia y negocio.
 - Reglamentos, estándares industriales y políticas de la compañía.
 - Publicaciones en revistas de software comercial y en revistas técnicas.
 - Procedimientos operativos estándar (SOP).

En la documentación (incluso antes de los requerimientos del usuario, documentos de especificaciones de usuario y especificaciones de los sistemas de interconexión y subsistemas).

Documentación de apoyo (por ejemplo, help desk, soporte de campo, instalación, mantenimiento y guías de solución de problemas).

- Informes de usuarios problema, los registros de quejas y peticiones de mejora.
- Los materiales de capacitación, manuales de usuario, y tutorial.
- Sitios web y material de marketing.
- Interfaz de usuario en línea y grupos de discusión.



- Búsqueda de información sobre productos de la competencia, especialmente aquellos con: una funcionalidad atractiva para los clientes, los más utilizados, los menos utilizados, los que causan molestia, entre otros.

2. Revise y analice la documentación.

- Busque información sobre los requisitos no funcionales (por ejemplo, rendimiento, usabilidad y seguridad).
- Considere la posibilidad de los posibles requisitos que se asignan a los programas y que se destinará a las personas como parte de un proceso de negocio.
- Comparta y revise los resultados con los clientes y usuarios.

3. Cree el borrador de los modelos de análisis.



Una vez revisada la técnica, sería conveniente hacernos las siguientes preguntas: ¿Qué le parece la técnica?, ¿Cómo cree que debería estar la documentación para poder hacer el análisis respectivo?. ¿Qué sucede si el usuario responsable de la documentación no proporciona los documentos apropiados?. ¡Qué problema! Verdad.



Cuando estamos ante una situación real son varios los problemas a los que deberemos enfrentarnos, por tanto el analista encargado del levantamiento de información deberá tener ciertas características que le permitan manejar los inconvenientes que se puedan presentar.

Actividad recomendada



Continuando con el caso de su localidad, indique cuáles son los documentos que se puede estudiar para levantar información y por ende definir requisitos. (Liste y describa cada documentos).

3.4.7. Cuestionarios

Esta técnica está orientada a un determinado grupo de Stakeholders (Interesados) cuya lista de preguntas debe considerar lo siguiente:

- Cada pregunta debe ser elaborada de forma corta y en base a un contexto.
- Estandarizar la respuesta en base a una lista preestablecida de posibles respuestas.
- Las preguntas pueden ser de diferente tipo.
- Los Stakeholders deben entregar los cuestionarios marcando las respuestas apropiadas.
- Las preguntas de selección múltiple deben fácilmente permitir seleccionar una respuesta asociada a la lista de posibles respuestas.

- Las preguntas de ponderación deben responderse de acuerdo al grado de importancia observado, preferencias o riesgos. Las respuestas que podrían tener estas preguntas pueden ser valores cualitativos (como es muy alto, alto, bajo, etc.) o valores cuantitativos (como porcentajes o rangos de valores).

Son varias las ventajas que nos ofrecen los cuestionarios ya que nos permite obtener información subjetiva de forma rápida, a un bajo costo, de forma remota a un gran número de personas.

La forma de aplicar el cuestionario también es diverso, desde una forma tradicional mediante un papel impreso hasta utilizando herramientas tecnológicas con diseños acordes al grupo que se desea aplicar.

Contrariamente una de las desventajas de los cuestionarios es que la información obtenida sea sesgada por diversos motivos, como por ejemplo, la muestra de personas que se escogieron para aplicar el cuestionario, las personas que estaban dispuestos a responder, no hay una relación directa con los encuestados por lo que no se puede contextualizar las preguntas, preguntas ambiguas, etc.

Consecuentemente debemos diseñar las preguntas cuidadosamente y validar el cuestionario para evitar y mitigar los posibles errores. De acuerdo a (Lamsweerde, 2009) los criterios de validación incluyen:

- Los encuestados deben ser una muestra representativa y significativa.
- La cobertura de la lista de preguntas y posibles respuestas.
- Ausencia de sesgos en las preguntas y en las posibles respuestas.
- Formular preguntas y respuestas no ambiguas.

Para diseñar y aplicar los cuestionarios (o encuestas), se debe hacer lo siguiente:

1. Identifique el propósito del cuestionario.
2. Determinar la muestra (grupo) y el método de recolección.
3. Diseñe las preguntas del cuestionario.
4. Pruebe el cuestionario antes de distribuirlo.
5. Aplique el cuestionario.
6. Analice y documente los datos.



Cuando realizamos cuestionarios de alta calidad realmente son de gran utilidad ya que sirven como complemento para otras técnicas, como es el caso de las entrevistas ya sea para un primer acercamiento o para diseñar una posterior entrevista con un mayor grado de detalle.

Es importante respetar el tiempo de los stakeholders cuando se utilizan técnicas que implican la interacción directa a los interesados. Asegúrese de que comience y termine a tiempo para: entrevistar, la facilitación de talleres, la realización de los grupos focales, realización de análisis de tarea de usuario, o la observación a los usuarios. Cada proyecto es diferente. Al seleccionar las técnicas de captura de requisitos, tenga en cuenta los factores que se indican en el Anexo 2.



3.5. Plan de elicitación de stakeholders

Es un plan que considera la importancia de los requisitos de los diferentes Stakeholders y sus contribuciones al proceso de desarrollo de requisitos. Es necesario hacerlo para decidir quién debe participar en las actividades de los distintos requisitos y la forma en que debería contribuir. El desarrollo de esta estrategia le ayuda a evitar pasar por alto a los Stakeholders y los requisitos faltantes. También ayuda a obtener el compromiso de los "grupos de interés por su tiempo y participación.



La participación de los Stakeholders es esencial para los proyectos de software exitosos. Las personas son la fuente principal de información de los requisitos, esto es importante para obtener la participación de los interesados centrados en las necesidades.

¿Qué hace el plan?

- Permite que el equipo centre sus esfuerzos en los requerimientos de alta prioridad por parte de los stakeholders.
- Establece relaciones de colaboración entre los técnicos y actores del proyecto.
- Alienta a los patrocinadores y los champions asegurar que las personas con conocimientos de requerimientos críticos estará disponible para el equipo del proyecto.
- Promueve el uso eficaz del tiempo de las personas.



Las preguntas clave que esta herramienta debe responder:

¿Cuán importantes son las necesidades de cada actor?

¿Cómo podemos involucrar a cada uno de los interesados en el proceso de desarrollo de requisitos?

Para establecer el plan, se debe hacer:

1. Catalogue la importancia de cada stakeholder en las categorías de Stakeholder. Utilice un esquema de clasificación, como MoSCoW (Siglas en inglés):
 - Debe (M -> Must): Esencial para el éxito.
 - En caso de (S -> Should): Es muy importante para recoger y comprender los requerimientos de este grupo de interés.
 - Podría (C -> Could): Es bueno tener la participación de este grupo de interés, pero de menor importancia.
 - No (W -> Won't): no debe ser considerada.
2. Determinar cómo va a involucrar a cada stakeholder que posee el M, S o C. tenga en cuenta:
 - Grado de participación: Decida la medida en que cada actor va a participar. Él o ella pueden participar plenamente, tienen algún grado de participación limitada, o sea indirectamente si un sustituto representa sus necesidades.
 - Método de la participación: Determinar cómo el actor va a participar:

- *Activamente:* Participa en las necesidades de talleres, encuestas, entrevistas, grupos focales, o prototipos.
- *Pasiva:* Obtiene informes de los sustitutos o revisa mensajes de correo electrónico sobre el progreso del desarrollo de los requisitos.
- *Indirectamente:* Suministra la mesa de servicio o logs de peticiones del cliente o encuestas anónimas o datos de marketing.
- La frecuencia de la participación: Decidir si el involucramiento del stakeholder será continua o periódica.

3. Registre el plan de elicitación en una tabla u otro documento.

En el (PMI-PMBOOK, 2008) se establece que para determinar a las personas interesadas (stakeholders), se debe conocer su influencia e interés en el proyecto y catalogarlos de la siguiente manera:

- **Stakeholders directos:** Los que están involucrados en el ciclo de vida del proceso, por lo que se verán afectados y tendrán interés en una finalización exitosa.
- **Stakeholders indirectos:** Estos muestran cierta preocupación por el proyecto ya que su nivel de interés e influencia es bajo.

En la siguiente figura 3.1 se hace una relación que existe entre el interés y la influencia de los interesados. En (Wieggers, 2003) se establecen algunos criterios para agrupar a los usuarios y establecer ciertos grupos a los que pueden pertenecer cada uno de ellos. Además un usuario en determinados momentos del desarrollo del proyecto, puede pertenecer a un determinado grupo de stakeholders y en otro instante pertenecer a otro grupo.

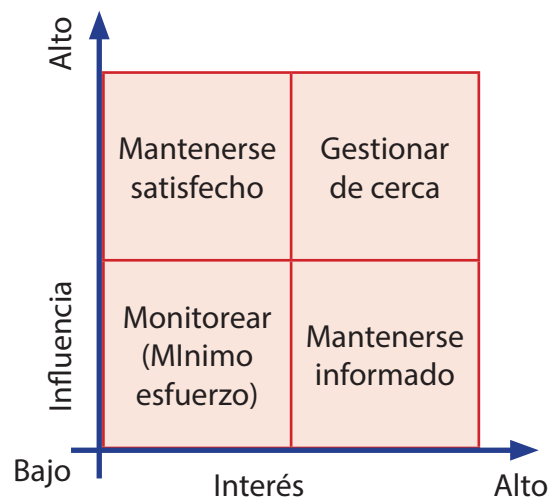


Figura 3.3: Relación entre interés e influencia de los Interesados, tomado de (PMI-PMBOOK, 2008).

En el (PMI-PMBOOK, 2008) se indican los Stakeholders que pueden existir en una organización, cabe señalar que dependiendo del contexto de funcionamiento de la organización se pueden definir otros.

- Clientes/Usuarios: Personas u organizaciones que usarán la solución.
- Patrocinador: Persona o grupo de personas que financian el proyecto.



- Directores de portafolio: Ejecutivos de la organización que manejan los proyectos.
- Directores de programa: Responsable de la gestión y coordinación de los proyectos.
- Oficina de Dirección de proyectos: PMO – Dirección centralizada de los proyectos.
- Directores del proyecto: Personas que tienen a cargo todos los aspectos del proyecto.
- Equipo del proyecto: Grupo de personas que tienen a su cargo el desarrollo del proyecto.
- Gerentes funcionales: Expertos que proporcionan servicios al proyecto.
- Gerentes operacionales: Personas encargadas de investigar, desarrollar, diseñar, fabricar, aprovisionar, probar los productos en la organización.
- Vendedores/Socios de negocio: Compañías externas a la organización que celebran un contrato para proporcionar servicios al proyecto.



Actividad recomendada

Realice un análisis del interés e influencia que podrían proporcionar los stakeholders en un proyecto de desarrollo de sistemas. Apóyese en la figura 3.3 de este capítulo.

Se ha concluido el estudio de esta unidad, por lo que conviene comprobar cuanto ha asimilado de los temas tratados para lo cual es necesario desarrollar las autoevaluaciones de conocimiento.



Autoevaluación 3



Realice las actividades que se indican en cada uno de los literales.

Lea detenidamente la afirmación y escriba una V si considera que es verdadera o una F si no lo es, en el paréntesis respectivo.

1.	La elicitación de requerimientos consiste en identificar las fuentes de información y luego utilizando técnicas evocar esas fuentes.	()
2.	Listar las fuentes consiste en un inventario de personas, documentos específicos y fuentes de información externa.	()
3.	Las categorías de stakeholders son grupos o individuos que están interesados en el producto de software que se esta desarrollando.	()
4.	Los grupos de stakeholders (categorizados) necesitarán estar informados acerca del progreso, conflictos, cambios y prioridades del proceso de desarrollo del producto.	()
5.	Hay dos categorías de stakeholders: usuarios y clientes.	()

6.	Los clientes son aquellos que estan en contacto con el producto de software o son afectados por este de alguna manera.	()
7.	Los stakeholders expertos pueden catalogarse en la categoría de proveedores.	()
8.	El Product Champion es quien asegura que el software desarrollado cumple con las necesidades de los usuarios.	()
9.	El perfil de los stakeholders es una descripción que caracteriza a cada stakeholder y explica su relación con el proyecto.	()
10.	Los prototipos exploratorios, son versiones parciales o preliminares del software creado para explorar o validar los requisitos.	()

2. Complete la siguiente tabla, con la herramienta apropiada.

Cuando necesite:	Entonces crear:
a) Identificar fuentes	
b) Identificar Stakeholders del producto	
c) Describir necesidades y criterios de éxito de los stakeholders	
d) Revisar técnicas de elicitación	
e) Plan de elicitación	





UNIDAD 4: : Análisis de Requerimientos



El análisis de requerimientos consiste en refinar los requisitos para asegurar que todos los Stakeholders entienden y examinan errores, omisiones, y otras deficiencias. El análisis incluye la descomposición al detalle de requisitos de alto nivel, la construcción de prototipos, evaluación de viabilidad, y la negociación de prioridades.

El objetivo es desarrollar los requisitos de suficiente calidad y detalle que los gerentes pueden construir estimaciones realistas del proyecto y el personal técnico pueda proceder con el diseño, construcción y pruebas.

A menudo es útil representar algunos de los requisitos de múltiples maneras: por ejemplo, tanto en forma textual y gráfica. Los resultados del análisis están en los modelos de requisitos. Los modelos de requisitos (también conocidos como modelos de análisis) son las necesidades de los usuarios representados por diagramas, texto estructurado (por ejemplo, listas, tablas o matrices), o combinados. El análisis también implica dar prioridad a las necesidades mediante el análisis de los requisitos para tomar decisiones sobre su importancia y oportunidad.

Los requisitos elicitados desde los stakeholders y articulados usando modelos de análisis necesitan ser lo suficientemente claros y completos para posteriormente validar el proceso de requerimientos de software.

El análisis de los requisitos es principalmente responsabilidad del analista, pero puede involucrar a los actores clave, tales como: usuarios, clientes y personal técnico quienes son necesarios para entender las necesidades del usuario.



Como puede ver, una vez identificados a los Stakeholders clave y establecidos los mecanismos para recopilar la información, se tiene un conocimiento en cuanto a los requisitos, por lo que realizaremos el análisis de los mismos.

4.1. ¿Por qué se debería crear modelos de requisitos?.

Los modelos de requisitos le ayudarán a:

- Facilitar la comunicación entre personal técnico y empresarios. Los modelos permiten que el equipo vea diferentes aspectos de las necesidades del usuario desde diferentes perspectivas.
- Descubrir requisitos faltantes, erróneos, ambiguos y contradictorios.
- Los modelos de requerimientos se juntan, lo que permite al equipo dar a conocer los requisitos relacionados e inconsistentes entre modelos. Descubrir y corregir estos errores resulta en requerimientos de alta calidad.
- Hacer que el proceso de desarrollo de requisitos sea más interesante y atractivo para los stakeholders. El uso de modelos textuales y visuales ofrece y permite los interesados entender los requisitos de más de un ángulo.

- Utilice los diferentes modos de pensamiento humano. Algunas personas piensan con más precisión con las palabras, mientras que otras son más capaces de entender los conceptos con los diagramas.

4.2. El ciclo de análisis de requerimientos

Es importante analizar las necesidades a medida que los provocan: las personas, documentos y fuentes externas.

Para analizar los requerimientos:

1. Modelo de negocio (Si es necesario).
 - Determinar si los modelos de negocios se necesitan.
 - Elegir uno o más modelos de negocio.
 - Crear los modelos, verificar su exactitud a medida que evolucionan.
2. Defina el alcance del proyecto.
 - Crear una combinación de modelos para describir el alcance del proyecto.
 - Compruebe los modelos entre sí para descubrir los defectos de los requisitos.
 - Revisar y obtener un acuerdo con el patrocinador del proyecto.
3. Crear modelos de requerimientos de usuario detallados.
 - Seleccione varios modelos que ayudan a los usuarios expresar sus necesidades.
 - Repetidamente refinar los modelos, validando sus correcciones.
 - Aproveche el plan de elicitación de Stakeholders para hacer mejor uso del tiempo de las personas.
 - Revisar el alcance de los modelos cuando nuevas necesidades tiende a surgir.
4. Priorizar las necesidades.
 - Organizar los requisitos para que fácilmente puedan ser priorizados.
 - Reunir a los Stakeholders para negociar los requerimientos.
 - Determinar criterios para la toma de decisiones acerca de la importancia relativa de los requisitos.
 - Priorizar los requerimientos basados en estos criterios.
5. Repita los pasos 3 y 4 como detalles sobre los requisitos que surgen o se revisan.

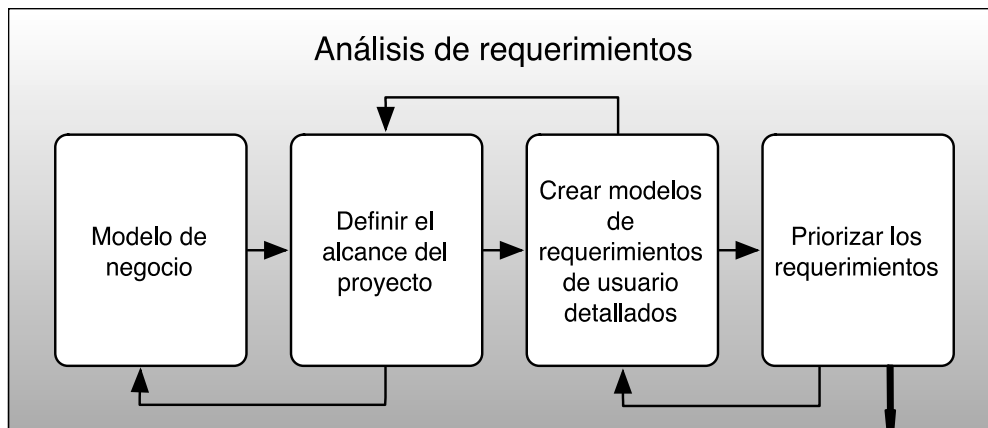


Figura 4.1: Proceso para realizar la fase de análisis de requerimientos. (Gottesdiener E. , 2005)

4.3. ¿Qué modelos se pueden crear?

Se puede usar una variedad de modelos de requerimientos de usuarios para analizar los requerimientos. Los modelos representan respuestas a las “4W’s + H” (¿Quién? ¿Qué? ¿Cuándo? ¿Por qué? y ¿Cómo?). (Siglas en inglés).

Tabla 4.1: Modelos de requerimientos basados en el enfoque del usuario

Enfoque de la pregunta Ejemplo de pregunta		Modelo de requerimiento de usuario para este enfoque
¿Quién?	• ¿Quiénes son los Stakeholders del proyecto? • ¿Quiénes interactuarán directamente con el sistema? • ¿Quién supervisará la interactividad con el sistema?	• Categoría de stakeholders • Tabla de actores o personas • Mapa de dialogo
¿Qué?	• ¿Qué hace importante el significado de los términos de negocio? • ¿Qué funciones de la organización interactúan para compartir información? • ¿Qué información entra y sale del sistema? • ¿Qué son los elementos de datos estáticos que se deben almacenar y cómo se relacionan?	• Glosario • Mapa de relaciones (un modelo de negocio) • Diagrama de contexto • Modelo de datos
¿Cuándo?	• ¿Cuándo el sistema necesita responder o actuar? • ¿Cuándo se realizaron las tareas y cuando se cambia la información?	• Tabla evento-respuesta • Diagrama de estado

¿Por qué?	• ¿Por qué estamos motivados para hacer cumplir las normas, políticas, reglamentos y legislación? • ¿Por qué son las decisiones las que influyen en el comportamiento y hacen valer la estructura empresarial?	• Políticas de negocio • Reglas del negocio
¿Cómo?	• ¿Cómo hacer operar los procesos en el negocio para alcanzar los objetivos de negocio? • ¿Cómo son las tareas realizadas y en qué secuencia?	• Mapa de procesos (un modelo de negocio) • Los casos de uso y posiblemente los mapas de casos de uso y los paquetes de caso.

4.4. ¿Cómo elegir el modelo correcto?

Ciertos modelos son más apropiados para determinar los requerimientos de ciertos dominios de negocio. Se escoge el modelo de acuerdo a una pregunta de enfoque (¿Quién? ¿Qué? ¿Cuándo? ¿Por qué? y ¿Cómo?), que proporcione una potencial comprensión entre los requerimientos y el desarrollo del modelo. Por ejemplo:

- Dominios transaccionales de negocio:

Que manejan procesos de negocio y tareas tales como: operaciones de negocios y administración, procesamiento de pedidos y gestión de inventario. Son muy adecuadas para los modelos “¿Cómo?”. Por ejemplo, casos de uso y escenarios. Relacionados con modelos “¿Quién?” y “¿Por qué?”, que también son útiles para estos dominios. Por ejemplo: los actores y reglas de negocio.

- Dominios estructurales:

Existen para almacenar y analizar datos, tales como: sistemas de minería de datos, generar consultas e informes. Son adecuados los modelos “¿Qué?”. Por ejemplo, modelos de datos. También se puede complementar con los modelos “¿Por qué?”. Por ejemplo, con reglas de negocio.

- Dominios dinámicos:

Que responden a los acontecimientos en constante cambio para almacenar datos y actuar en función de su estado en un punto en el tiempo. Por ejemplo: los sistemas que gestionan el tráfico en la red, la adjudicación, las operaciones de un dispositivo mecánico, y otras operaciones en tiempo real. Es muy adecuado los modelos “¿Cuándo?”. Por ejemplo: tablas de evento-respuesta y diagramas de estado. También se pueden incluir los modelos: “¿Por qué?” (por ejemplo, con reglas de negocio), “¿Qué?” (por ejemplo, con modelos de datos), y con “¿Cuándo?” el usuario esta involucrado con interfaces.



- Dominios orientados al control:

Evalúan las condiciones para tomar medidas o decisiones como la logística, la detección de fraudes, la configuración del producto, y el diagnóstico. La mejor forma de describir es utilizando los modelos “¿Por qué?”. Por ejemplo: reglas de negocio y las tablas de decisión. Debe ser complementado con modelos “¿Qué?”. Por ejemplo, modelos de datos.



Estas son sólo directrices. Cada dominio es diferente por lo que debe determinar qué modelos son los más útiles en el desarrollo de un subconjunto de modelos en forma preliminar y validación de ellos, y luego ajustar sus selecciones. No es necesario usar todos los modelos de requerimientos. Puede escoger un subconjunto que sea adecuado al dominio del problema. Ahorre tiempo a los Stakeholders elaborando unos modelos de alto nivel, y luego consulte si son útiles.



Actividad recomendada

De acuerdo a la explicación de este apartado, analice e indique cuáles son los modelos que podría utilizar para el desarrollo de requerimientos, para el caso de estudio de su localidad.

4.5. Herramientas y técnicas para analizar requerimientos

A continuación en la tabla se indican las técnicas y herramientas para analizar requerimientos de acuerdo a (Gottesdiener E. , 2005).

Tabla: 4.2: Herramientas y técnicas para análisis de requerimientos

Cuando necesite:	Entonces crear:
Modelar el negocio	Combinar entre mapa de relaciones y/o mapa de procesos.
Entender el alcance del proyecto	Combinar entre diagramas de contexto, tablas de evento-respuesta y/o políticas de negocio.
Adicionar detalle a los requerimientos de usuario	Combinar o variación de: tabla de actores, casos de uso, mapas de dialogo, modelo de datos, diagramas de estado y/o reglas de negocio.
Negociar la importancia entre los requisitos	Priorizar requerimientos.

Las técnicas y herramientas que se indican en la tabla 4.2, se desarrollan de acuerdo a la forma en que se apliquen las técnicas de levantamiento de información, por ejemplo: ¿Cómo obtener la información necesaria para construir un “Mapa de Procesos”? Evidentemente que es necesario realizar entrevistas, cuestionarios, revisar documentación existente, etc. Con la finalidad de tener la información necesaria para poder construir estos modelos, en ciertos casos de ser necesario se tendrá que utilizar técnicas con un mayor grado de especialización.

De acuerdo a la tabla 4.2, hay algunos modelos que se desarrollan en base a los requerimientos ya establecidos e incluso ya especificados, por ejemplo los casos de uso.



Entonces:

- ¿No es el proceso apropiado?
- ¿No es necesario especificar los requerimientos?
- ¿Podemos pasar ya al diseño y desarrollo del sistema?

Recuerde que en la primera unidad hablamos sobre el proceso de desarrollo de requerimientos en la cual decíamos que era un desarrollo de elaboración progresiva o incremental; por ende en cada interacción se podrá hacer la especialización y definición de los requerimientos. Deberá utilizar las técnicas de acuerdo al avance y progreso en la definición de los requerimientos.

A continuación se indican algunas técnicas que deben ser utilizadas y planificadas conforme se realicen las interacciones en el proceso de desarrollo de requerimientos y de acuerdo a lo que se necesite crear.

4.6. El modelado del negocio



De acuerdo a (Jacobson, Booch, & Rumbaugh, 2004) cuando se construyen los sistemas, estos deben dar un verdadero soporte y agregar valor al proceso en una organización, sin embargo los usuarios desconocen cuáles son los requisitos por lo que la captura no solamente consiste en una sencilla entrevista con los usuarios, sino que es necesario establecer los elementos necesarios para planificar las actividades que garanticen conocer el dominio del problema y los contextos tanto organizacional como operacional, en definitiva la situación actual.

El modelado de negocio le ayuda a entender cómo una aplicación de software apoya los procesos de negocio, y descubre los requisitos que se deben asignar (por ejemplo, la actualización de los documentos oficiales, guías que hay que volver a trabajar, realizar actividades de capacitación, y la revisión de los procedimientos operativos). Un proceso de negocio es un conjunto de tareas relacionadas que crea algo de valor para el negocio. El modelado de negocio también ayuda a definir procesos eficientes para el uso del nuevo software.

El software que se propone debe integrarse con los procesos de negocios existentes o nuevos, pero no todos los proyectos de software requieren modelos de negocio. Usted debe considerar el modelado de negocios, cuando:

- El alcance del proyecto no es claro o es demasiado grande.
- Hay un patrocinio no claro o difuso.
- La gestión empresarial quiere repensar o rediseñar cómo se hace el trabajo.
- El proyecto consiste en la investigación o la aplicación de software comercial.
- La empresa debe cumplir con las políticas legales o reglamentarias que requieren la intervención manual, procesos y documentación.



El modelado de negocios requiere del patrocinio y participación de los clientes. Muchos proyectos de software requieren de significativos procesos de negocio y cambio organizacional. Cuando una empresa tiene cargas regulatorias o legales, los procesos no automatizados son necesarios para la supervivencia y los puestos de trabajo y roles cambian con frecuencia por lo que la gente tiene que ser comunicada con tiempo y con frecuencia. Los empresarios necesitan definir e implementar nuevos procedimientos y documentar para comunicar y gestionar el cambio. El modelado de negocio le permite hacer frente a estos temas difíciles desde el principio.

4.6.1. Mapa de relación

Un mapa de relaciones es un diagrama que muestra qué tipo de información y productos que se intercambian entre los clientes externos, proveedores y las funciones clave de la organización.

Se usa para entender el contexto de la organización mediante la identificación de las funciones de negocios afectadas y sus entradas y salidas. Un mapa de relaciones de negocios pueden revelar oportunidades de mejora de proceso antes de definir el alcance de un proyecto de software.

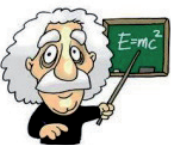


Las preguntas que se intenta responder con este modelo son:

- ¿Qué entradas recibimos de nuestros clientes externos y los proveedores?
- ¿Qué resultados ofrecemos a nuestros clientes externos y los proveedores?
- ¿Qué áreas funcionales están involucradas en el manejo de las entradas y salidas?
- ¿Cuáles son los traspasos (entradas y salidas) dentro de nuestra organización?

Para construir un mapa de relaciones, realice lo siguiente:

1. Dibuje las funciones clave, departamentos y grupos de trabajo involucrados en el proceso de negocio en cajas.
2. Liste las principales entradas y salidas que cada función recibe o produce.
3. Conecte las entradas y salidas de las funciones que usan y producen.



Ejemplo:

A continuación se presenta un mapa de relación para el trámite registro de nota como parte del caso de estudio.

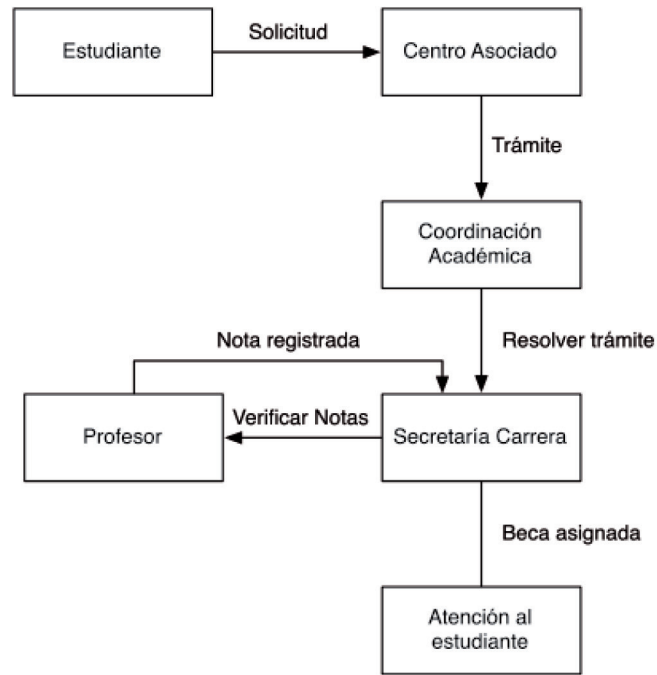
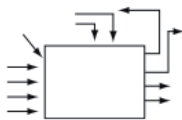


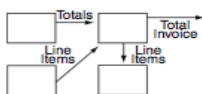
Figura 4.2: Ejemplo de mapa de relaciones

Mejora de procesos de negocio:

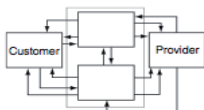
Se puede utilizar el mapa de relaciones para identificar mejoras a los procesos de negocio antes de iniciar los esfuerzos de la construcción del nuevo software. Así:



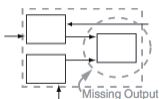
Funciones sobrecargadas: un gran número de entradas y salidas a través del diagrama.



Repetición: la misma entrada o salida o similar utilizada en múltiples funciones.



Múltiples interfaces externas: muchas entradas y salidas de todas las funciones van y vienen de los clientes externos y proveedores.



Funciones desnudas: falta de entradas o salidas.



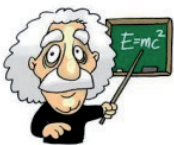
4.6.2. Mapa de procesos

Un mapa de procesos muestra la secuencia de pasos, entradas y salidas necesarias para manejar un proceso de negocio a través de múltiples funciones, organizaciones o roles de trabajo.

Se utiliza para identificar qué procesos se asignan a la empresa (procesos manuales) y qué se destinará al software. Los mapas de procesos también sirven como base para la mejora de procesos de negocio.

Para construir un mapa de procesos, realice lo siguiente:

1. Nombre a los procesos de negocio a ser modelados, comenzando con un verbo de acción.
2. Definir el evento que dispara o inicia el proceso.
Nombre del evento de negocios en un formato “sujeto + verbo + objeto”
Por ejemplo: “El cliente ordena lugares”.
3. Nombre el punto final o resultado del proceso.
Darle un nombre sencillo y directo, en positivo (por ejemplo, “La orden está completa”, “El trabajo está programado,” o “La factura se paga”).
4. Liste los participantes en el proceso de negocio (es decir, áreas funcionales, departamentos o funciones) en una columna en el lado izquierdo del diagrama.
5. Crear filas horizontales o “carriles” para cada participante, para representar a la entidad de la organización o el papel donde se realiza el trabajo.
6. Identificar todos los pasos del proceso que se producen entre la activación de eventos y el resultado.
7. Identificar las salidas de cada etapa.
8. Revise el diagrama como sea necesario.



Ejemplo:

A continuación se indica un ejemplo de mapa de procesos que aborda el trámite Asentamiento de Notas, que realiza el estudiante desde el centro al que pertenece. Analice el proceso y las actividades que cada actor realiza.

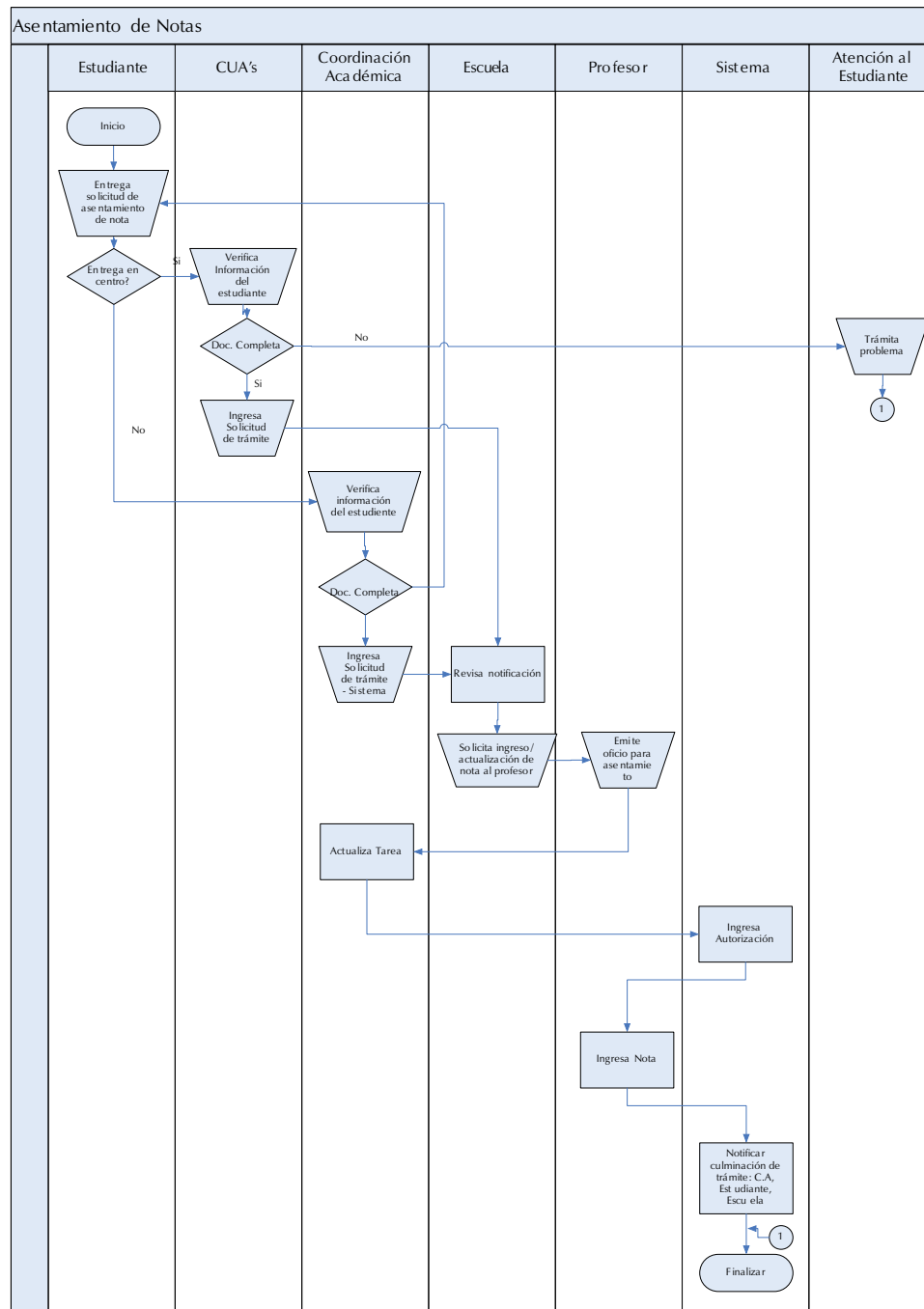


Figura 4.3: Ejemplo de mapa de procesos.



Actividad recomendada

Continuando con el caso de su localidad, y en base a la información recopilada al aplicar las técnicas de elicitación, determine las actividades que realiza cada actor para un determinado proceso y construya un mapa de proceso.



4.7. Describir el alcance del proyecto

Definir que requerimientos están en el ámbito para reducir en gran medida el riesgo de los proyectos de software. Evitar por ejemplo la expansión incontrolada de los requisitos a medida que avanza el proyecto. Representar los requerimientos a un mismo nivel permite establecer un lenguaje común acerca de los requerimientos y ayuda a articular el límite entre lo que es y lo que está fuera del alcance del producto.

Una definición clara del alcance del producto también limita el enfoque del proyecto para permitir una mejor planificación, uso del tiempo y el uso de los recursos. Si el alcance del proyecto no está claro, es mejor cancelar o retrasar el proyecto hasta que pueda ser aclarado y acordado por los Stakeholders clave.

A continuación se describen algunos modelos que permiten describir el alcance de los proyectos.

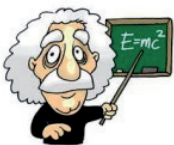
4.7.1. Diagrama de contexto

Un diagrama de contexto muestra al sistema en su entorno, con las entidades externas (es decir, personas y sistemas) que proporcionan y reciben información o material desde y hacia el sistema.

Se lo utiliza para ayudar a los stakeholders de forma rápida y simple a definir el alcance del proyecto, y centrarse en los insumos que necesita el sistema así como las salidas que ofrece. El diagrama de contexto ayuda al equipo a obtener modelos de requisitos (por ejemplo, los actores, casos de uso, y la información de los datos del modelo) y pueden surgir posibles problemas de alcance como nuevas entidades externas.

Para construir un diagrama de contexto, realice lo siguiente:

1. Dibuje un círculo para representar el sistema, ponga una etiqueta con el nombre del sistema.
2. Identificar las entidades externas.
3. Añadir los flujos de información.
4. Después trace otros requerimientos de usuario (como es el glosario, tabla evento-respuesta, actor, y casos de uso), verifique estos modelos contra el diagrama de contexto, y revisar si es necesario.



Ejemplo:

A continuación se indica un diagrama de contexto para el caso del sistema de gestión de trámites de la UTPL, en la que se establecen las entidades externas que proporcionan y reciben información al sistema.

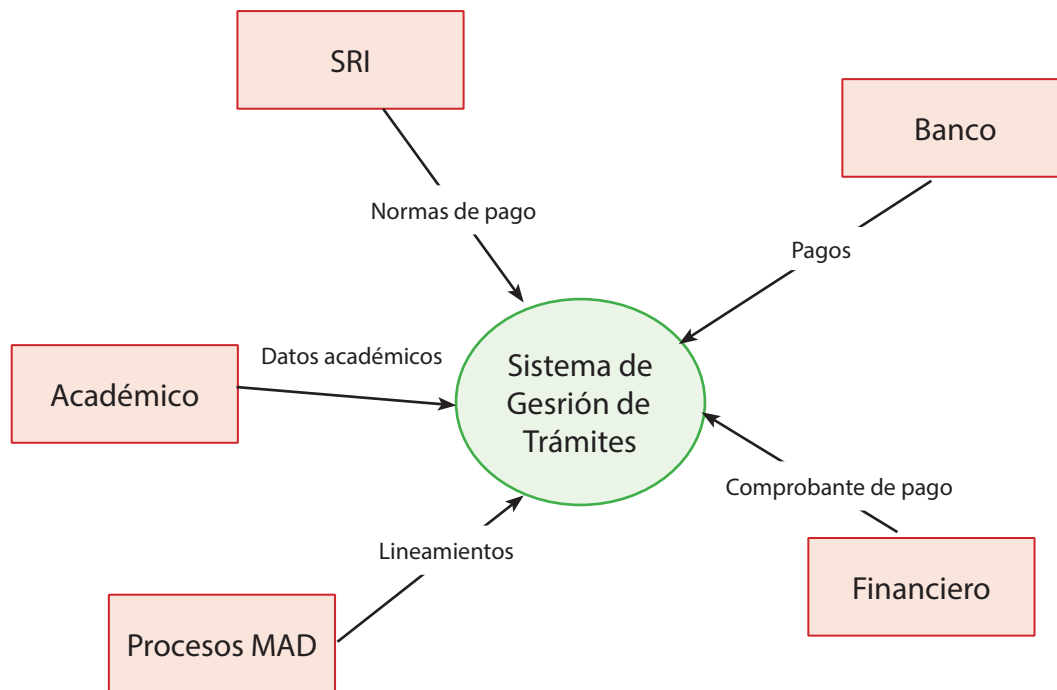


Figura 4.4. Ejemplo de un diagrama de contexto.



Actividad recomendada

Realice un diagrama de contexto del caso de desarrollo de su localidad para indicar cuales son las entidades externas que dan o reciben información del sistema.

4.7.2. Tabla evento - respuesta

Una tabla evento-respuesta identifica cada evento (por ejemplo: un estímulo de entrada que activa el sistema para llevar a cabo alguna función) y las respuestas de eventos resultantes de esas funciones.

Se usa para definir todas las condiciones para las que el sistema debe responder, para la definición de los requerimientos funcionales. (Cada caso requiere una respuesta predecible del sistema.). La creación de una tabla evento-respuesta también puede surgir como necesidad de acceso a bases de datos externas o fuentes de archivos.

Para construir una tabla evento-respuesta, realice lo siguiente:

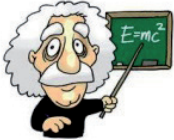
1. Nombre los eventos y clasifíquelos como negocio, temporal o eventos de señal.
 - Para los nombres de los eventos de negocios use el formato “sujeto + verbo + objeto”. Los eventos de negocios hacen que los usuarios humanos inicien una interacción con el sistema.
 - Para los nombres de los eventos temporales use el formato “tiempo para <verbo + objeto>”. Los eventos temporales son en función del tiempo. Estos eventos temporales son completamente predecibles y dará lugar a trabajos programados o que se ejecutan por lotes.
 - Para los nombres de los eventos de señal use el formato “sujeto + verbo + objeto”. Estos eventos de señales se originan en los dispositivos de hardware.



- Para cada evento, describir su respuesta apropiada.

El formato de las respuestas de evento es "<objeto> siempre a <asunto>" o "el sistema almacena <Información>".

- Verifique la tabla de eventos-respuesta frente a los modelos existentes y revisar si es necesario.



Ejemplo:

Tabla de evento-repuesta del sistema de Autotrámites. Caso de estudio.

Evento	Tipo de evento	Respuesta
Registrar el trámite	Negocio	Entregar al estudiante el comprobante de registro del trámite.
Tiempo para registrar el trámite y notificar al estudiante del registro	Temporal	Correo de notificación enviado al estudiante.
Coordinación académica notifica al profesor, secretaria o director de la carrera sobre la resolución del trámite	Negocio	El involucrado recibe la notificación vía correo electrónico sobre la decisión del trámite, y los próximos pasos a dar.

4.7.3. Políticas de negocio

Las políticas de negocio son las directrices, normas y reglamentos que rigen o condicionan la conducta de un negocio. Las políticas son la base para la toma de decisiones y el conocimiento que se implementan en el software y en los procesos manuales. Ya sea impuesta por una agencia externa o desde dentro de la empresa, las políticas de negocio se usan para agilizar las operaciones, aumentar la satisfacción y lealtad del cliente, reducir el riesgo, mejorar los ingresos, y cumplir con los requisitos legales (y por lo tanto mantenerse en el negocio).

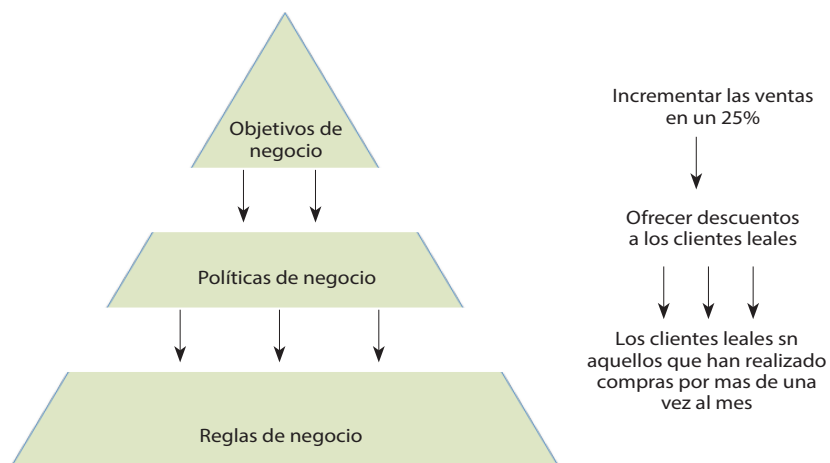


Figura 4.5. Clasificación de las reglas, políticas y objetivos del negocio. (Gottesdiener E. , 2005)

Se usan para identificar las políticas destinados a los empresarios, que permitirá a la comunidad empresarial preparar la aplicación de actualización de procedimientos de software, guías, capacitación, formularios y otros bienes necesarios para hacer cumplir las políticas.

Las políticas asignadas al software deben estar explícitamente definidas como reglas de negocio y deben estar incluidas al final de la especificación de requerimientos de software. Las reglas de negocio evolucionan a partir de políticas de alto nivel que a su vez proceden de los objetivos de negocio.

Las políticas se originan ya sea desde el interior de una organización o de una entidad externa, como una agencia gubernamental.

Para definir las políticas de negocio, realice lo siguiente:

1. Identificar los grupos de políticas de negocio para el dominio del problema.
2. Determine dónde localizar las políticas.
3. Documentar las políticas que están en el ámbito del proyecto.



Algunos equipos de proyecto eligen comenzar identificando las reglas de negocio y luego asociarlos a las políticas de negocio, en lugar de comenzar con las políticas. Cuestionar las reglas es una forma de construir normas. Para replantear cada política: ¿Es necesario? ¿Promueve los objetivos de negocio? ¿Se puede identificar claramente por qué la política es necesaria? ¿Puede la política ser aplicada en el software?. La adición o la aplicación de las políticas requiere tiempo y dinero.



Ejemplo:

A continuación se establecen ciertas políticas de negocio del caso de estudio (Autotrámite).

Política de grupo	Ident.	Política	Propietario	Fuente
Registro	BP-01	Los trámites se registran dentro de un cronograma.	Secretaría centro	Manual de procesos de la MAD
	BP-02	Solo se registran trámites con toda la documentación.		
	BP-03	El estudiante es el único que puede registrar el trámite.		
Autorización	BP-04	Coordinación académica autoriza y notifica que se ejecute el trámite.	Coordinación académica	Manual de procesos de la MAD
	BP-05	Revisa la documentación de acuerdo al plan.		
	BP-06	Notifica al estudiante sobre el estado del trámite.		
	BP-07	Redirecciona el trámite de acuerdo al reclamo.		



4.8. Agregar detalle a los requerimientos de usuario

Una vez que el alcance del producto está claro, es necesario que analice los requerimientos más al detalle. Use múltiples modelos, combine modelos para crear un excelente conocimiento de las necesidades del usuario. Los modelos se unen para comparar los detalles y poder encontrar defectos.

Planifique una secuencia para crear modelos que mejor se articulen a las necesidades. La secuencia, sin embargo, importa menos que el acto de la iteración entre los modelos para conocer los requisitos y revelar los defectos de los requisitos. Comience por definir y analizar un modelo, y luego cambie a otro, de forma periódica regrese a cada modelo a detallar o corregir.

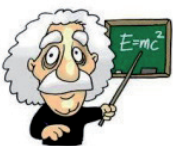
4.8.1. Tabla de actores

Una tabla de actor identifica y clasifica a los usuarios del sistema en términos de sus funciones y responsabilidades. Se usa para detectar la falta de los usuarios del sistema, para identificar los requisitos funcionales como los objetivos del usuario (casos de uso), y para ayudar a los empresarios de aclarar las responsabilidades del trabajo.

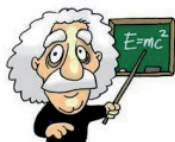
Los actores no representan a personas físicas o a sistemas, sino su rol. Esto significa que cuando una persona interactúa con el sistema de diferentes maneras (asumiendo diferentes papeles), estará representado por varios actores. Por ejemplo, una persona que proporciona servicios de atención telefónica a clientes y realiza pedidos para los clientes estaría representada por un actor «equipo de soporte» y por otro actor «representante de ventas».

Para definir la tabla de actores, realice lo siguiente:

1. Liste los roles que se desempeñan en el sistema y coloque el nombre del rol en la columna "Actor" de la tabla. No liste como actores a los títulos de trabajo o personas específicas. En su lugar, abstraiga un listado basado en las necesidades de los actores para conseguir algo específico con respecto al sistema. (Los actores incluyen personas, otros sistemas, dispositivos hardware, y temporizadores).
2. Coloque los atributos de cada actor en columnas adicionales. Escriba una descripción breve de las responsabilidades de cada actor. Agregue más columnas para incluir los otros atributos (si es necesario), tales como:
 - Profesiones relacionadas
 - Ubicación
 - Nombres de personas
 - Nivel de experiencia en el uso del sistema
 - Dominio del entorno
 - Frecuencia de uso
3. Revise la tabla de actores para encontrar actores faltantes o extraños.



Durante el diseño, se amplía la tabla actores, permitiendo a la base de datos acceder a las reglas de cada actor (por ejemplo: los derechos de una entidad de datos: crear-leer-actualizar-eliminar).



Ejemplo:

A continuación se indica la tabla de actores para uno de los trámites, del sistemas de gestión de trámites de la UTPL.

Actor	Atributos y Responsabilidades	Título del empleado
Estudiante	Inicia el proceso. Facilita los documentos para registrar el trámite.	Estudiante
CUA's	Registra el trámite en el sistema. Recibe y verifica que los documentos para el trámite sean los correctos. Notifica a coordinación académica sobre el registro de un trámite.	Coordinador del CUA's
Coordinación académica	Decide en base a los documentos presentados, calendario académico y tipo de tramite si aprueba o niega el trámite. Notifica a otras instancias sobre la decisión del trámite.	Secretaría
Profesor	Resuelve tramites relacionados con la parte académica. Actualiza datos en el sistema de gestión académica.	Profesor



Actividad recomendada

Realice la tabla de actores en función del mapa de proceso desarrollado en el apartado 4.6.2. Indique el nombre del actor, las responsabilidades y el título que utiliza.

Relaciones de los actores

Utilice un mapa de actores (también conocida como una jerarquía de actor o modelo de usuario) para mostrar las interrelaciones del actor. Los actores pueden ser escritos en tarjetas (una por ficha) o dibujados con el Lenguaje de Modelado Unificado (UML).

Cuando varios actores, como parte de sus papeles, también representan un papel más generalizado, se describe mediante una relación de **generalización**. El comportamiento del papel general se describe en una superclase actor. Los actores especializados heredan el comportamiento de la superclase y extienden ese comportamiento de algún modo.

4.8.2. Casos de uso

A decir de (Jacobson, Booch, & Rumbaugh, 2004) "Un caso de uso es una descripción de un conjunto de secuencias de acciones, incluyendo variantes, que ejecuta un sistema para producir un resultado observable de valor para un actor". A continuación detallamos algunas de las características que nacen de la definición de los casos de uso:



Un caso de uso, se puede expresar que es uno de los usos que se quiere dar al sistema, lo cual también se puede saber respondiendo a la pregunta ¿Para qué usáramos el sistema?, para efectos del ejemplo, piense en un sistema que todos conocemos, como un teléfono celular inteligente (Smartphone), y hagamos la pregunta ya mencionada ¿Para qué usaría un teléfono inteligente?, piénselo un momento y haga su propia lista de posibles respuestas.

Ahora bien, la lista podría ser extensa sobre todo por las posibilidades de aplicaciones que se pueden incluir, sin embargo hagamos una lista básica de cosas que se podrá hacer con este sistema.

1. Recibir llamadas
2. Realizar llamadas
3. Leer mensajes de texto.
4. Escribir mensajes de texto.
5. Capturar fotografías.
6. Capturar video.
7. Enviar mensajes de correo electrónico.
8. Recibir mensajes de correo electrónico.
9. Navegar en internet.
10. Reproducir música.
11. Reproducir video.
12. Manejar agenda de actividades diarias con alarmas.

Si lo analiza un poco y aunque la lista podría estar incompleta, piense por un momento en cada ítem de esta lista, todos representan algo que el usuario puede hacer con el teléfono celular. Cada uno de estos ítems es lo que conocemos como casos de uso.



Note que esta lista no incluye aspectos demasiado generales que no nos dejen claro lo que hace o debería hacer el sistema, por ejemplo “gestionar llamadas telefónicas” o “administrar mensajes de correo electrónico”, tampoco acciones específicas que el usuario debe hacer para obtener el resultado, como por ejemplo presionar la tecla llamar (Send), de hecho el presionar la tecla es un paso del caso de uso “Realizar llamadas”, por tanto las acciones o interacciones con el caso de uso según la definición deben dar valor al trabajo del usuario, y desde esta óptica sólo pueden ser casos de uso las tareas que se puede hacer con el sistema.

Nombres de los casos de uso

Los nombres de los casos de uso deben ser expresiones verbales que describen algún comportamiento del vocabulario del sistema que se está modelando, es muy común al menos al inicio colocar nombres que describen las actividades necesarias para completar el funcionamiento de un caso de uso, o por el contrario definir nombres muy amplios que reflejan módulos completos de una aplicación y reflejan ninguna acción específica del sistema.

Ejemplos de nombres de casos de uso pueden ser: registrar matrícula, crear ficha de estudiante, anular matrícula, solicitar matrícula; como nombres incorrectos por ser actividades tendríamos: Ingresar datos personales, validar contraseña, ingresar fecha de nacimiento, fijar monto a depositar, seleccionar contacto; y finalmente ejemplos de nombres incorrectos demasiado generales: Gestión académica, cobranzas, realizar inventario.

La notación que usa UML para representar los casos de uso es una figura de óvalo. En la figura 4.6 se muestra algunos de los casos de uso identificados para el teléfono celular. Tenga en cuenta los nombres que se han colocado para cada uno de los casos de uso.

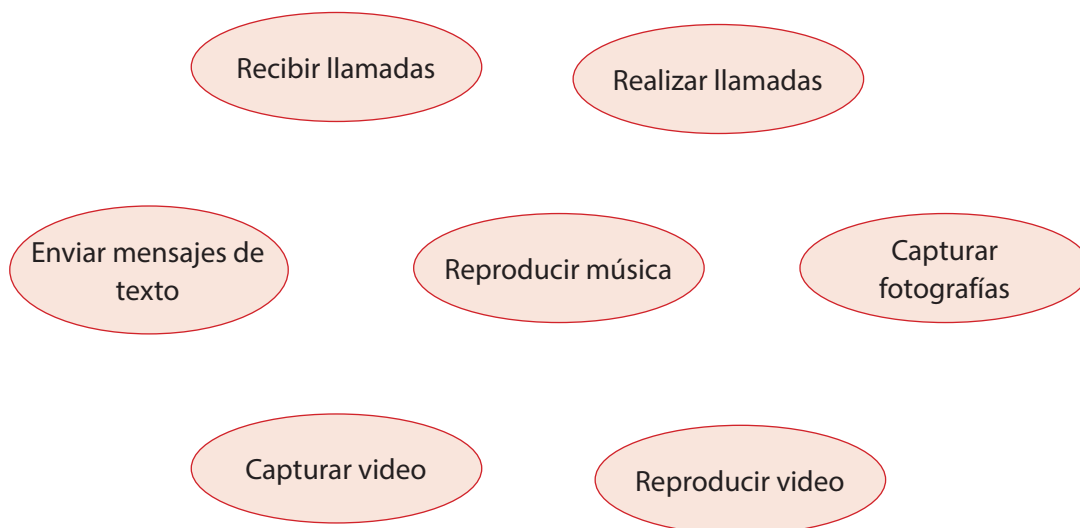


Figura 4.6: Representación gráfica de los casos de uso que implementa un teléfono celular

Ahora bien, los casos de uso, siempre van asociados a un actor que podríamos decir que es un usuario, un dispositivo u otro sistema que está en condiciones de interactuar con la funcionalidad que representa el caso de uso.

Si analizamos la definición siguiente “un actor representa un conjunto coherente de roles los usuarios y los casos de uso representan una interacción con estos. Normalmente, un actor representa un rol que es desempeñado por un humano, un dispositivo de hardware o cualquier otro sistema al interactuar con nuestro sistema”⁴, podemos llegar a las siguientes conclusiones:

- Un actor no es necesariamente una persona, es un rol o un conjunto de roles, por lo tanto no hace referencia a un individuo en particular, sino a varios individuos representados en el mismo rol.
- Una misma persona puede jugar varios roles, pero el actor en cada caso es diferente.
- La relación que une al actor con el caso de uso se denomina interacción, y este siempre es de dos vías, es decir está en condiciones de enviar o recibir mensajes desde y hacia los casos de uso.
- Un actor es un elemento externo al sistema, es decir nunca forma parte del mismo.

3Un actor se representa con un monigote, y por tratarse de un rol, se lo describe en singular. La figura 4.7 representa la relación entre un actor y varios casos de uso.

4 Reproductor de audio y video fabricado por Apple

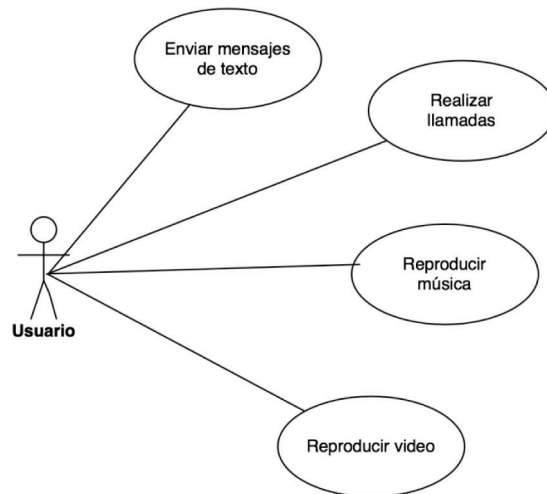


Figura 4.7: Representación gráfica de los casos de uso con un actor.

El modelado de casos de uso es una técnica que no se limita a un proceso de desarrollo o a una tecnología específica, por el contrario pueden usarse para cualquier paradigma de desarrollo de software, esto por la sencilla razón de que los casos de uso especifican el qué, no el cómo.

Hasta ahora hemos identificado tres componentes del modelo: los actores, los casos de uso y las asociaciones, veamos un ejemplo que nos permita comprender el significado de cada uno de estos elementos, notará que más allá de la definición, lo que importa es la interpretación que se le debe dar a cada uno de ellos.



El comportamiento de un caso de uso se especifica describiendo flujos de eventos en forma textual, el cual debe ser expresado en un lenguaje que favorezca la comprensión de todos los involucrados en el sistema incluyendo cliente, usuarios analistas, programadores, etc. En esta descripción se especifica cómo inicia y cuando acaba el caso de uso, además de describir la interacción entre los diferentes elementos del sistema para producir el resultado deseado.

En la figura 4.8, podemos apreciar algunas de las funcionalidades de un iPod4 video.

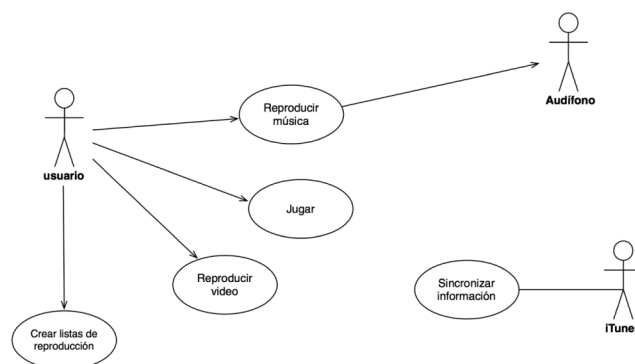


Figura 4.8: Modelo de casos de uso para un reproductor de audio y video⁵

⁵ Reproductor de audio y video fabricado por Apple

Un caso de uso además incluye todas las situaciones que pueden darse al momento de ejecutarlo, a cada una de estas situaciones se la conoce como una instancia del caso de uso y entre estas instancias podemos distinguir el flujo básico de eventos al cual también podemos decir que es el flujo normal y las demás instancias se pueden definir como flujos alternos o excepcionales. A la combinación de flujo normal con uno o más flujos alternos se conoce como escenario, y los escenarios representan por tanto todo lo que puede darse en el caso de uso y en la gran mayoría de las veces es preciso definir los escenarios exitosos y los escenarios fallidos.

La especificación de los flujos de eventos y escenarios se verá con detalle mas adelante. Mientras tanto avancemos estudiando algunos otros elementos del modelado de casos de uso.

Actores



Conforme se ha indicado anteriormente, un actor es un rol que representa un usuario, un dispositivo u otro sistema que se comunica con nuestro sistema. Este rol puede ser desempeñado por varias personas, y a su vez una persona puede ejecutar varios roles. Gráficamente se representa con un monigote como se aprecia en las figuras anteriores. Ej. En un centro Universitario, el coordinador del centro puede ingresar al sistema con el rol de coordinador y hacer las cosas que le competen en ese rol, y a la vez como estudiante puede ingresar al sistema con ese rol, quedando claro de que no puede usar sus dos roles a la vez, ver figura 4.9.

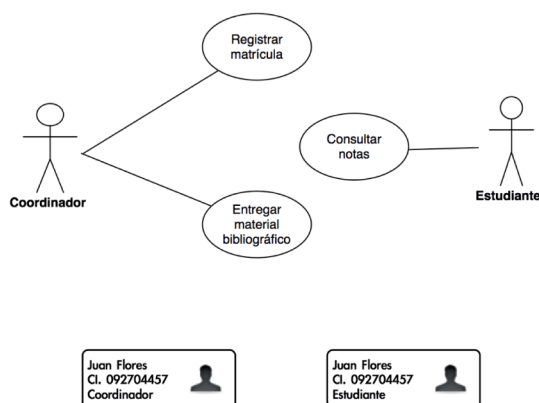


Figura 4.9: Un usuario desempeñando varios roles

Cuando hablamos de comunicación, decimos que es el paso de mensajes entre el actor y el caso de uso.



Importante:

Un actor siempre es externo al sistema, es decir no forma parte del mismo, es preciso tenerlo en cuenta al momento de definirlos.

La única relación posible entre casos de uso y actores es la asociación, en la figura 4.8 se aprecia las relaciones entre actor y caso de uso. La asociación es una relación bidireccional por lo tanto el actor y el caso de uso pueden enviar o recibir información.



Las puntas de flecha determinan el origen de la comunicación, normalmente es un detalle que se suele pasar por alto, pero puede ser valioso al momento de modelar las demás partes del sistema.

Relaciones entre actores

Los actores no se relacionan directamente, por el hecho de ser externos, no se representan relaciones entre ellos, sino únicamente la relación de estos con los casos de uso. Sin embargo pueden existir categorías generales de actores así como también categorías especializadas, las cuales se representan a través de una relación de generalización, que en principio tiene las mismas características que en los modelos de clases y se representa con una flecha que va desde el actor especializado hacia el actor general, como se aprecia en la figura 4.10, Donde se ve que un actor docente puede especializar como investigador, y esto en el sistema significa que el docente investigador puede trabajar con los casos de uso de docente, mas los específicos de investigador.

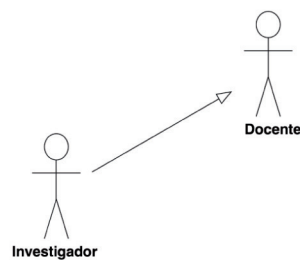


Figura 4.10: Relación de generalización entre actores



Actividad recomendada

Conteste lo que se solicita.

a) Analice el diagrama de la figura 4.8 y conteste las siguientes interrogantes:

1. ¿Cuántos actores tiene el sistema?
2. ¿Cuáles actores son humanos?
3. ¿Cuáles actores son dispositivos?
4. ¿Cuáles actores son sistemas externos?
5. ¿Qué cosas puede hacer el actor usuario?
6. ¿Qué interpretación se les puede dar a las puntas de las flechas?
7. ¿Por qué la asociación entre el caso de uso "Sincronizar información" y el actor iTunes no tiene flecha?
8. ¿Puede el actor usuario ejecutar el caso de uso "Sincronizar información"?

Dedique unos minutos a responder a estas interrogantes, y si lo hace correctamente, su comprensión del modelado de casos es adecuada para poder continuar. Puede comprobar sus respuestas más adelante.

- b) Modifique el diagrama de casos de uso, de modo que permita obtener un actor especializado que sea capaz de transmitir el audio a través de una frecuencia en FM, además agregue un caso de uso que le permita grabar notas de voz ¿será necesario un nuevo actor? ¿Se agregan nuevas funcionalidades al actor usuario?

Con el fin de que pueda comparar sus respuestas, vamos a responderlas explicando detalles que le servirán de guía para entender el modelo.

Solución a las preguntas a)

1. ¿Cuántos actores tiene el sistema?
Un actor se representa con un monigote, por lo tanto el sistema cuenta con 3 actores que son: iTunes⁷, usuario y audífono.
2. ¿Cuáles actores son humanos?
El único actor humano es usuario.
3. ¿Cuáles actores son dispositivos?
El único dispositivo que interactúa con el sistema en este caso es audífono, en este punto vale aclarar que un iPod no dispone de parlantes internos, pero si una salida de audio a la cual se conectan los audífonos, los cuales actúan como actores, los audífonos no son parte del iPod, y de hecho podría conectarse a otro dispositivo como un emisor FM o a unos parlantes que en este caso también actuarían como actores.
4. ¿Cuáles actores son sistemas externos?
Si revisamos la información acerca que lo es iTunes, llegamos a la conclusión de que este es un sistema, y puesto que puede interactuar con nuestro iPod, se constituye en un actor.
5. ¿Qué cosas puede hacer el actor usuario?
Las funcionalidades que puede ejecutar un actor en este caso son las representadas en los casos de uso, por lo tanto las cosas que puede hacer el actor usuario son: reproducir música, reproducir video, crear listas de reproducción y jugar.
6. ¿Qué interpretación se les puede dar a las puntas de las flechas?
Las asociaciones entre casos de uso y actores son bidireccionales, es decir que la interacción se da en ambos sentidos, por lo tanto las puntas de flecha denotan quien inicia la interacción.
7. ¿Por qué la asociación entre el caso de uso Sincronizar información y el actor iTunes no tiene flecha?⁵⁷
Porque no es importante representar quien inicia la interacción, y cualquiera de los dos puede iniciarla.
8. ¿Puede el actor usuario ejecutar el caso de uso “Sincronizar información”?
Según el modelo no es posible, esto sólo puede ejecutarlo el actor iTunes, aunque sea el mismo usuario quien controle a iTunes, no lo hace directamente

⁷ iTunesTM, es un aplicativo de Apple para reproducir música en el ordenador y que contiene funciones que permiten sincronizar el contenido del iPodTM



Modificación del diagrama. Literal b)

Para solucionar el planteamiento, es preciso considerar varias de las características de los actores, en este caso cuando hablamos de actores especializados nos referimos a la relación de herencia entre audífono y un nuevo actor al que denominaremos transmisor fm, en este caso para tener una mejor comprensión hemos cambiado el nombre al actor audífono por el de salida audio, con lo cual puede ser cualquier dispositivo capaz de producir el sonido necesario para que sea escuchado por un usuario, el nuevo actor entonces sería capaz al igual que su actor general de interpretar la señal de audio y en lugar de producir los sonidos las convertiría en ondas de radio de frecuencia modulada para que sea interpretadas por cualquier radio, que en este caso no es necesario colocar en nuestro diagrama debido a que no interactúan directamente con el sistema.

Por otro lado se nos pide que agreguemos un caso de uso que sea capaz de grabar notas de voz, ello implica que alguien debe activar este caso de uso; en este proceso sería el actor usuario, por lo tanto si se agregan una nueva funcionalidad, pero el dispositivo no tiene un micrófono, al menos no hemos definido que lo tenga, por tal motivo agregaremos un actor que no forma parte del sistema, pero está en condiciones de recoger los sonidos de los usuarios y procesarlos en el sistema, por lo tanto colabora con el caso de uso grabar notas de voz, que se aprecia en la figura 4.11

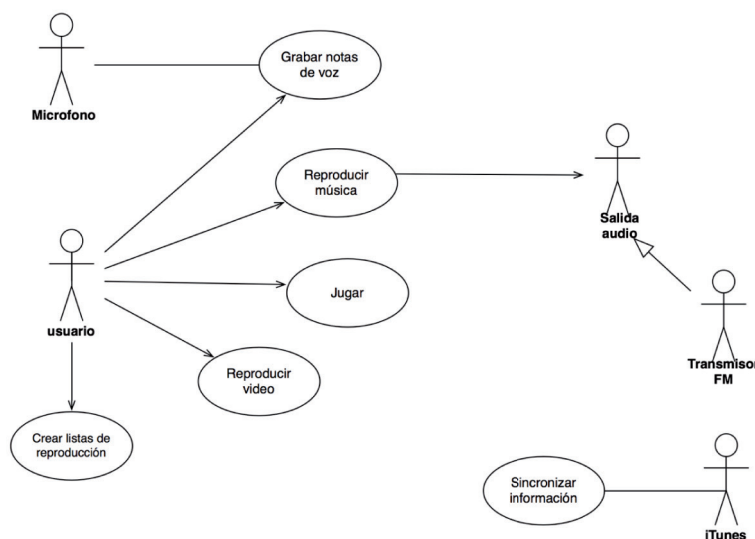


Figura 4.11: Solución a la parte B del ejercicio.

¿Qué le pareció el ejercicio? Con toda seguridad le ayudó a clarificar el significado de los casos de uso, actores y las asociaciones entre ellos, sin embargo esto no es suficiente para aplicarlo en el modelado de todos los requerimientos del sistema, hacen falta estudiar las relaciones entre casos de uso, que tocamos en el siguiente tema.

Recuerde que si tuvo alguna dificultad con el contenido de esta asignatura, es momento de recurrir a su profesor tutor.

Relaciones entre casos de uso

Conforme avanza el modelado de requisitos, es preciso optimizar los modelos de casos de uso, esto con el propósito de mejorar la reutilización de requerimientos sin sacrificar la claridad y la comprensión de los mismos, además una estructuración adecuada de los casos de uso puede favorecer la administración de requerimientos.



La estructuración entre casos de uso consiste en estudiar el comportamiento de los casos de uso con el propósito de establecer aquellos casos en los que resulta mejor separar ciertas partes del comportamiento de algunos casos de uso para optimizarlos y reutilizarlos, a este proceso se lo conoce como factorización de casos de uso y puede incluir los elementos que se muestran a continuación:

- **Comportamiento común** a varios casos de uso, se lo identifica porque aparece en los flujos de varios casos de uso.
- **Comportamiento opcional**, es cuando cierta parte del flujo se ejecuta solo en ciertos casos y no puede ser puesto como un flujo alterno.
- **Comportamiento excepcional**, aquel que no sucede en condiciones normales del caso de uso.
- **Comportamiento que se desarrollará en iteraciones futuras**, esto con el fin de no retrasar la implementación de los casos de uso considerados prioritarios.



Importante :

Es recomendable no empezar a estructurar los casos de uso hasta que tenga un grupo de requerimientos comprendidos y aceptados por los usuarios, de lo contrario puede crear serias confusiones.

En virtud de esta factorización se puede identificar tres tipos de relaciones entre casos de uso:

- Relación de inclusión.
- Relación de extensión.
- Relación de herencia.

Solamente para reconocer el origen de los casos de uso llamaremos como caso de uso base al caso de uso original y caso de uso agregado a aquel que se obtuvo producto de la optimización del modelo.

A continuación vamos a estudiar cada una de estas relaciones con ejemplos ilustrativos para cada caso, ponga especial atención sobre todo para distinguir entre las relaciones de inclusión y extensión.

a) Relación de inclusión (include)

La relación de inclusión nace como producto de la factorización del comportamiento de varios casos de uso, a los cuales se les extrae el comportamiento común y se lo coloca en un nuevo caso de uso, y para que los casos de uso base obtengan ese comportamiento deben invocarlo mediante la inclusión.

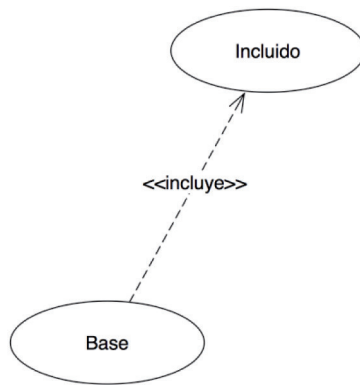


Figura 4.12: Relación de inclusión entre casos de uso.

En la inclusión, el caso de uso base invoca explícitamente al caso de uso añadido y su comportamiento depende de él, pero ninguno de los dos tiene acceso a los atributos del otro. El comportamiento del caso de uso agregado se dice entonces que está encapsulado y por tanto puede ser reutilizado por varios otros casos de uso base.

Recuerde:



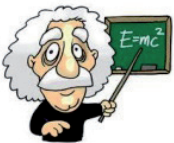
Los casos de uso añadidos no nacen como los casos de uso base de las necesidades de los usuarios, estos nacen como producto de la optimización del modelo, por lo cual estos no tienen relación directa con los actores, sino solo con otros casos de uso.

La relación de inclusión puede usarse para reutilizar el comportamiento, o simplemente para optimizar el modelo de forma que cierto comportamiento quede encapsulado.

El ejemplo más común de un caso de uso que es incluido en otros caso de uso es el de autenticar usuario, sin embargo este no necesariamente es requerido en todos los sistemas.

Para comprenderlo con más claridad observe el siguiente ejemplo:

Ejemplo:



A nuestro modelo del reproductor de audio y video iPod, vamos a analizarlo para identificar comportamiento común que pueda factorizarse y más aun reutilizarse.

Obviamente es necesario tener las especificaciones de los casos de uso, al menos los esquemas para determinar este comportamiento, sin embargo vamos a intuir un poco para obtener casos de uso añadidos para nuestro modelo.

Si considera el diagrama de la figura 4.8, vemos dos casos de uso cuyas funcionalidades podría tener algo en común, se trata de Reproducir música y reproducir video, en ambos casos será necesario buscar el archivo de audio o de video que se desea reproducir, y si se lo implementa tal cual está, tocará repetir la funcionalidad que busca los archivos que el usuario desea reproducir, esto nos da la oportunidad de factorizar estos casos de uso y obtener un nuevo caso de uso al que podríamos denominar como "seleccionar archivos".

Con lo cual el diagrama podría quedar como se muestra en la figura 4.13.

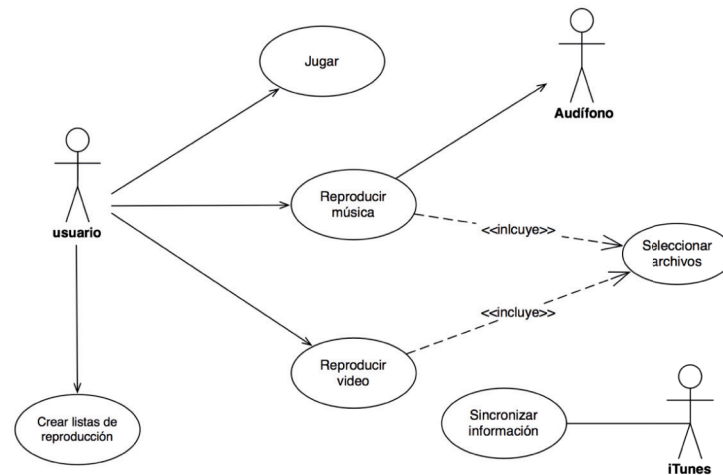


Figura 4.13: Modelo de casos de uso con inclusión 8

Si agregamos parcialmente el flujo de eventos a estos casos de uso debería verse de la siguiente manera:

Reproducir música	
1.	Incluir "Seleccionar archivos"
2.	Seleccionar play
3.	

Seleccionar archivos	
1.	Solicitar criterio de clasificación de archivos.
2.	Elegir de la lista los criterios disponibles.
3.	Seleccionar 1 de los álbumes mostrados.
4.	Finalizar selección.

Para comprender mejor cómo funciona la relación de inclusión observe en la figura 4.14 cómo el flujo de eventos se transfiere.

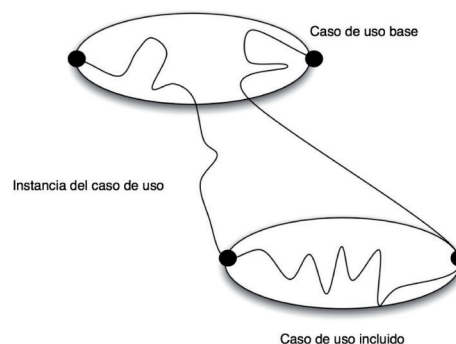


Figura 4.14: Flujo de eventos en una relación de inclusión⁸

En un determinado punto del flujo de eventos del caso de uso base, se incluye el comportamiento del caso de uso incluido y una vez que este se ejecuta totalmente, retorna el control al siguiente punto donde quedó el caso de uso original.

8 IBM Corporation (2003). Mastering requirements with use cases. Ob. cit. p. 10-9



Como se aprecia en el ejemplo, la relación de inclusión no es condicional, si el flujo de eventos del caso de uso base llega al punto de inclusión, el caso de uso incluido siempre se ejecuta, así mismo puede suceder que el flujo de eventos nunca llega a un escenario que deba incluir el caso de uso.

b) Relación de extensión (extend)

La relación de extensión conecta una extensión de caso de uso con el caso de uso base, es necesario definir en qué lugar del caso de uso base incorporar el punto de extensión. En UML esta relación se representa con una flecha que va desde la extensión del caso de uso hacia el caso de uso base, como se aprecia en la figura 4.15.

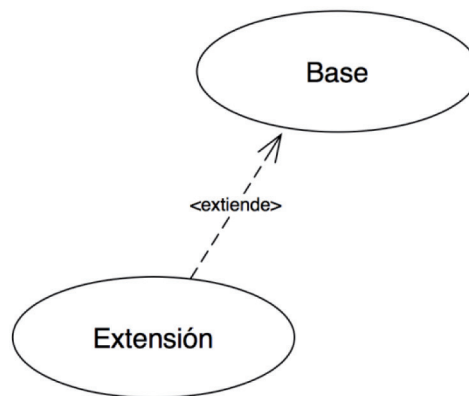
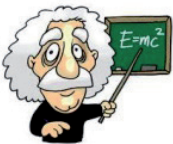


Figura 4.15: Representación en UML de la relación de extensión

El uso de la relación de extensión “significa que un caso de uso base incorpora implícitamente el comportamiento de otro caso de uso en el lugar especificado indirectamente por el caso de uso que extiende al base. El caso de uso base puede estar aislado pero, en algunas condiciones, su comportamiento puede extenderse con el comportamiento de otro caso de uso”, los lugares donde se incorpora el comportamiento del otro caso de uso se denominan puntos de extensión.”⁹

Desde el punto de vista del usuario, una relación de extensión denota un comportamiento opcional del sistema, el cual es requerido en determinadas condiciones, también es posible utilizar esta relación para modelar varios flujos que se pueden insertar en un punto dado y que son controlados explícitamente por el actor.



Ejemplo:

Retomemos nuestro ejemplo del reproductor de música, si observamos el caso de uso sincronizar información, cuyo propósito es copiar los archivos de audio y video al dispositivo de modo que se mantenga igual la biblioteca del computador con la biblioteca del dispositivo, en este proceso puede suceder que se tiene una nueva versión del software del dispositivo, por lo que opcionalmente se podría ejecutar un caso de uso denominado actualizar software. Esto lo podemos apreciar en la figura 4.16.

9 Booch, G. Et Al (2006).Ob. cit. p. 253

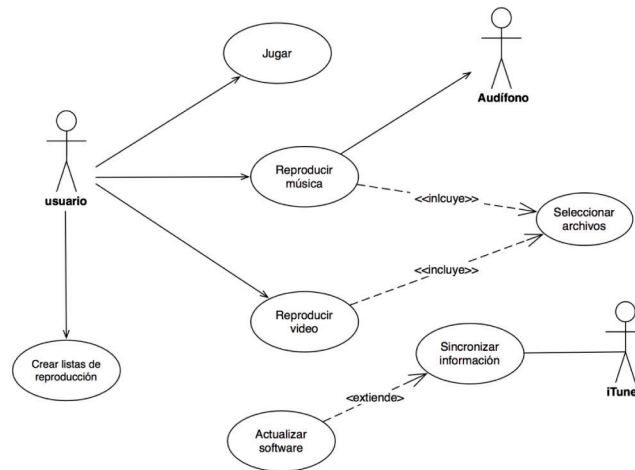


Figura 4.16: Ejemplo de relación de extensión

Para representar de mejor forma el comportamiento de esta relación observe la figura 4.17.

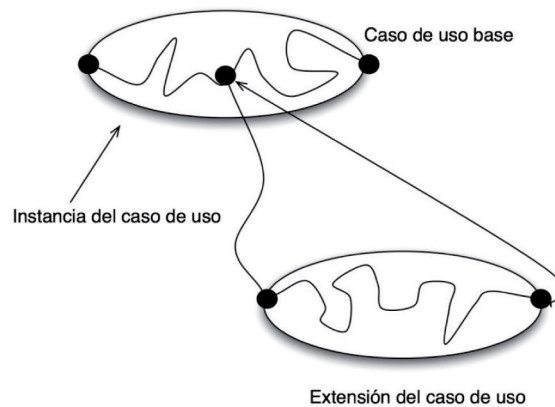


Figura 4.17: Flujo de eventos en una relación de extensión¹⁰

5.3.2 Relación de generalización

Otra variante de las relaciones entre casos de uso es la generalización, que al igual que las dos anteriores factoriza cierto comportamiento de los casos de uso para poder obtener variantes de los mismos. Esta relación es similar a la generalización de las clases es decir hay un caso de uso padre del cual uno o más casos de uso hijos heredan sus características y especializan cierto comportamiento por tanto una instancia de un caso de uso hijo se puede usar en cualquier lugar donde se requiera una instancia del caso de uso padre, pero no lo contrario.

La relación de generalización se representa en UML con una línea con una punta de flecha dirigida hacia la clase padre, como se muestra en la figura 4.18.

¹⁰ Rational University (2003). Mastering requirements with use cases. Ob. cit. p. 10-14

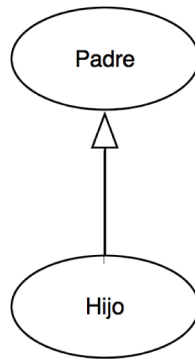
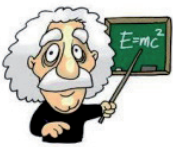


Figura 4.18: Notación de la relación de herencia entre casos de uso



Ejemplo:

En nuestro ejemplo, optimizaremos el modelo de casos de uso para tener un solo caso de uso reproducir medio que tenga la funcionalidad de reproducir el archivo de audio y de este obtendremos un caso de uso hijo que se especialice en reproducir video que funciona igual que su padre, pero agrega características relacionadas con el control de video que no dispone su padre, además se evita la relación de inclusión con el caso de uso seleccionar archivos haciendo que el modelo sea mucho más claro. Además se agrega otro caso de uso denominado grabar notas de voz junto con un actor adicional que permita dicho comportamiento para el sistema, con ello podrá visualizar todas las relaciones estudiadas en un solo diagrama.

El diagrama para este ejemplo se muestra en la figura 4.19.

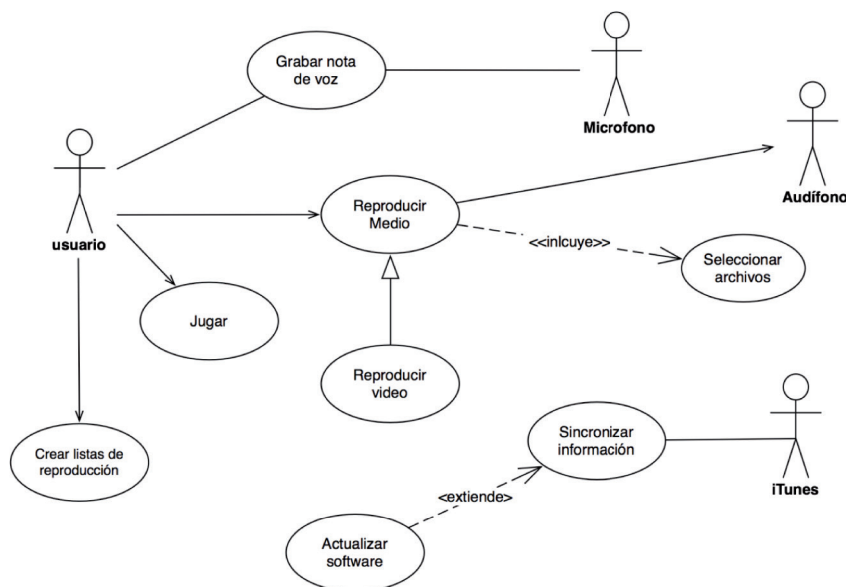


Figura 4.19: Modelo de casos de uso que todas las relaciones entre casos de uso

Como puede apreciar los diagramas de casos de uso son muy expresivos y sobre todo versátiles, en el ejercicio que hemos desarrollado hasta ahora ha sido fácil ir actualizando el modelo.

Diagramas de casos de uso

Los diagramas de casos de uso representan el comportamiento externo del sistema, en este sentido permiten la representación del contexto del sistema y también los requisitos. En el primero caso se usa una línea que establece los límites del sistema diferenciándolo de los elementos externos al mismo y en el segundo caso se usan para especificar lo que el sistema debería hacer sin detallar el cómo lo hace.

Según (Jacobson, Booch, & Rumbaugh, 2004), para modelar el contexto del sistema deben seguirse los siguientes pasos:

1. Identificar las fronteras del sistema decidiendo los comportamientos que formarán parte de él y cuáles serán ejecutados por entidades externas.
2. Hay que identificar los actores en torno al sistema, considerando qué grupos requieren ayuda del sistema para llevar a cabo sus tareas; qué grupos son necesarios para ejecutar las funciones del sistema; qué grupos interactúan con el hardware externo o con otros sistemas de software; y qué grupos realiza funciones secundarias de administración y mantenimiento.
3. Hay que organizar los actores similares en jerarquías de generalización/especialización.
4. Hay que proporcionar un estereotipo para cada uno de los actores, si se ayuda a entender el sistema.

Al diagrama de casos de uso de nuestro reproductor automático agreguémosle la información de contexto.

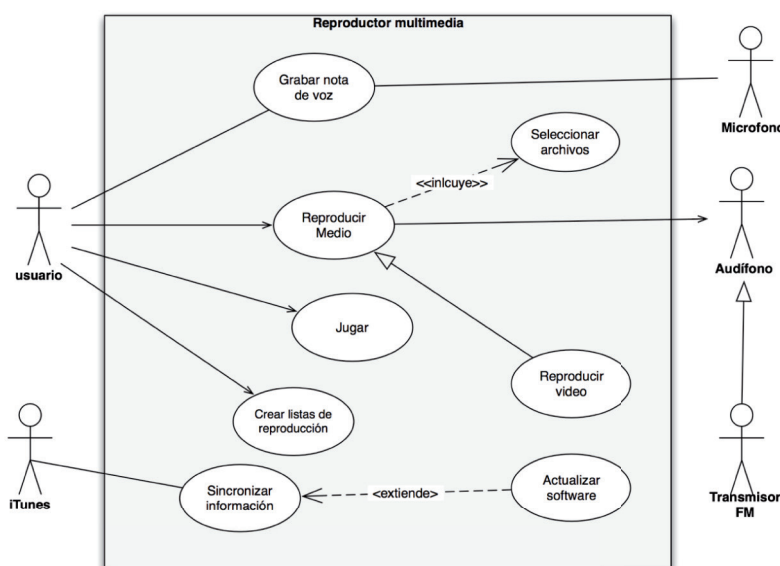


Figura 4.20: Modelo de casos de uso de contexto

Para modelar los requisitos del sistema con casos de uso, (Jacobson, Booch, & Rumbaugh, 2004) propone los siguientes pasos:

1. Establecer el contexto del sistema, identificando los actores a su alrededor.
2. Hay que considerar el comportamiento que cada actor espera del sistema o requiere que se le proporcione.



3. Hay que nombrar esos comportamientos comunes como casos de uso.
4. Hay que factorizar el comportamiento común en nuevos casos de uso que puedan ser utilizados por otros.

Ahora bien si se toma en cuenta los requerimientos el proceso para crear los casos de uso es el siguiente:

1. Identificar actores y casos de uso.

ACTORES

Recordemos que los actores son los roles que desempeñan los usuarios, dispositivos u otros sistemas que son capaces de interactuar con nuestra aplicación, por lo tanto en este paso se precisa identificarlos y asociarlos con los requisitos. Esta información ya la tenemos establecida en el documento de visión que expresa las necesidades junto con el documento de especificación de requisitos.

A continuación se presentan algunas preguntas útiles preguntas que nos pueden ayudar a identificar a los actores.

- ¿Quién o qué usa el sistema?
- ¿Quién o qué obtiene información del sistema?
- ¿Quién o que provee información al sistema?
- ¿En qué lugares de la compañía se usa el sistema?
- ¿Quién o qué soporta y mantiene el sistema?
- ¿Qué otros sistemas utilizan se valen de este sistema?

Una vez de identificados los casos caso de uso, es preciso describirlos, para ello se puede usar la siguiente estructura:

Tabla 4.2 Plantilla para definición de actores

Nombre	Coloque aquí el nombre del actor	Tipo	U/S/D
Descripción			
Describa la función del actor en el proceso de negocio			
Funciones con el sistema			
Enuncie las funciones que el actor debe desempeñar con el sistema y sus requerimientos			

Conforme se aprecia la tabla 4.2, los datos necesarios en una primera etapa son: nombre del actor, el tipo que corresponde a una de las categorías indicadas (usuario, sistema externo, dispositivo externo), la descripción especificada en términos del proceso de negocio vinculado al sistema y las funciones o requerimientos que cumpliría con el sistema. Esta información es fundamental para el momento de identificar los casos de uso.

Ahora es necesario validar si todos los actores han sido definidos, si no se lo ha hecho existe la probabilidad de omitir aspectos importantes que debe tener el sistema. Para ello IBM Corp. Propone que se puede intentar responder las siguientes preguntas:

- ¿Se han identificado todos los actores?
- ¿Se ha reportado para su modelado todos los roles en el entorno del sistema?
- ¿Se ha asociado cada uno de los actores con funcionalidades del sistema?
- ¿Se puede nombrar al menos dos personas para asumir un rol cualquiera?
- ¿Puede alguno de los actores desempeñar roles similares frente al sistema? Si es posible, júntelos en un solo actor.

CASOS DE USO:

Los casos de uso nacen del proceso de abstracción mediante el cual se establecen las funcionalidades de los actores. Puesto que en el paso anterior ya se definió lo que los actores harían frente al sistema, ahora es momento de refinar y organizar esas funciones en casos de uso. Para ello pueden servir de guía las siguientes interrogantes:

- **¿Cuáles son las metas de cada actor?, es decir**
 - o ¿Por qué el actor requiere el sistema?
 - o ¿Será el actor capaz de crear, almacenar, cambiar, remover o simplemente leer información del sistema? Si es capaz ¿por qué?
 - o ¿Necesitará el actor informar al sistema acerca de eventos externos o cambios?
 - o ¿Requerirá el actor ser informado acerca de ciertas ocurrencias en el sistema?
- ¿Podrá el sistema solventar todo el comportamiento requerido de parte del negocio?

Las respuestas a estas preguntas permitirá definir una lista de casos de uso o funcionalidades que deberá tener el sistema para atender los requisitos, importante en este caso es asegurarse de que para cada uno de los requisitos funcionales hay uno o más casos de uso que los soportan y que cada caso de uso es la respuesta a al menos un requisito de los usuarios, a estos casos de uso habrá que agregar otros propios del sistema como aquellos que tienen que ver con temas de seguridad, controles de acceso, configuración, etc.

Para tener una visión más clara de las funcionalidades, es recomendable representar visualmente los casos de uso, y luego intentar asociarlos con los actores.

El siguiente paso es realizar una descripción breve de los casos de uso, para ello es recomendable numerar cada uno de los casos de uso, asignarles un nombre y realizar una descripción de alto nivel.

Algunos autores llaman a estos como casos de uso de alto nivel. La tabla 5.2 es una plantilla adecuada para este propósito.



Desde el punto de vista de los requerimientos los casos de uso son fundamentales para definir el comportamiento dinámico del sistema, los escenarios se pueden representar mediante diagramas de interacción y la especificación de los comportamientos específicos con diagramas de actividades, además en base a las descripciones de los casos de uso se puede modelar las interfaces del sistema, es decir las pantallas que verá el usuario, estas se pueden usar para desarrollar prototipos, casos de prueba etc.

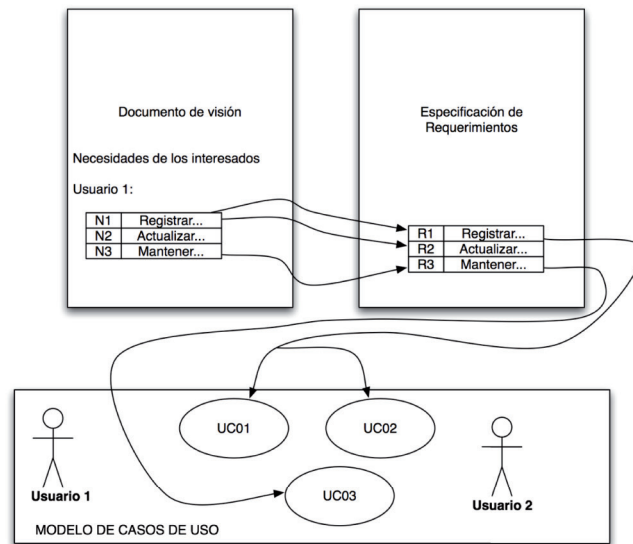


Figura 4.21: Proceso de Abstracción de casos de uso. (Jacobson, Booch, & Rumbaugh, 2004)

4.8.3. Reglas de negocio

Cada organización opera en base a un amplio conjunto de políticas, leyes y normas. Muchas de las organizaciones dependiendo del ámbito y el lugar de operación deben cumplir con leyes gubernamentales. Todos estos principios de control para la operación de las organizaciones se las conocen como reglas del negocio. Ciertas reglas tienen que considerarse como parte del control del software, tal es el caso de nuestro país (Ecuador) en el tema de la facturación; pero otras reglas son parte del proceso y tienen que hacerse un control humano.

Las reglas del negocio son importantes en la definición de requisitos funcionales ya que permiten establecer las capacidades que el sistema deberá tener para cumplir con las normas. Por tanto es imprescindible que estas reglas estén a mas de bien definidas, bien documentadas para su correcta interpretación y aplicación. Cuando las normas no están documentadas y están solamente en la cabeza de expertos o dirigentes, cuando salgan o se jubilen existirá un vacío del conocimiento de éstas reglas.

Existen diferentes criterios para hacer una clasificación de los diferentes tipos de las reglas del negocio. En la figura 4.22 se muestra una clasificación de cinco reglas de negocio que se consideran en una organización.

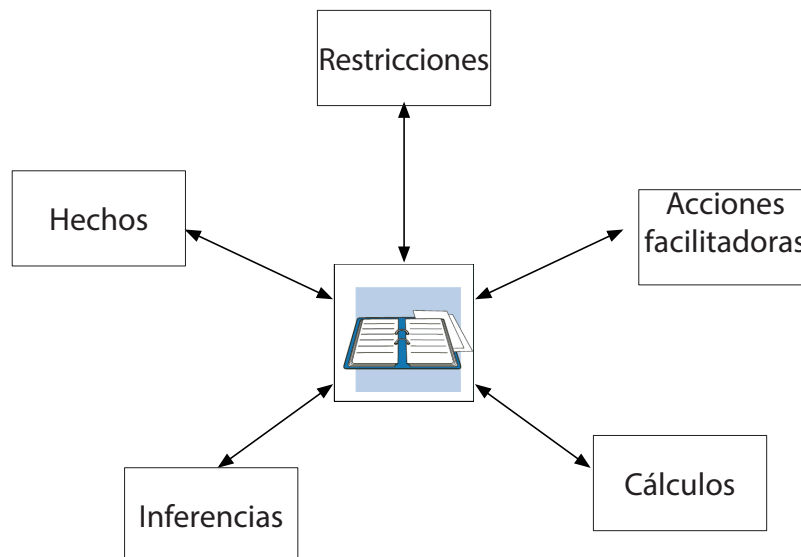


Figura 4.22: Clasificación de las reglas de negocio, tomado de (Wiegiers, 2003)

A continuación vamos a describir las reglas más comunes que se pueden identificar de cara a los procesos de una organización. Este enfoque se basa en lo expresado en (Wiegiers, 2003).

Hechos

Los hechos son afirmaciones verdaderas acerca del negocio. Los hechos describen asociaciones o relaciones entre los términos del negocio. Cabe indicar que los hechos no se traducen en requisitos funcionales. Los hechos son verdades inmutables por cuanto definen los datos y sus atributos. A continuación tenemos algunos ejemplos:

- Cada contenedor de productos químicos tiene un código de barras de identificador único.
- Cada orden debe tener un cargo de envío.
- Cada elemento de línea en un pedido representa una combinación específica de químicos, grado, tamaño del envase, y el número de contenedores.
- De que los boletos no reembolsables sin pagar tasa alguna cuando el comprador cambia el itinerario.
- Los impuestos no se calcula sobre los gastos de envío.

Restricciones

Las restricciones sirven para limitar las acciones que el sistema o los usuarios deban realizar. Los ejemplos sobre restricciones son:

- Un prestatario que es menor de 18 años de edad debe tener un padre o un tutor legal como aval en el préstamo.
- Un usuario de la biblioteca puede poner hasta 10 anuncios en espera.
- Un usuario puede solicitar un producto químico en la lista de riesgo de nivel 1 sólo si ha recibido entrenamiento con peligrosos químicos en los últimos 12 meses.



- d) Todas las aplicaciones de software deben cumplir con las regulaciones gubernamentales para el uso de personas con discapacidad visual.
- e) La correspondencia no puede mostrar más de cuatro dígitos del seguro social el número de la póliza de seguro.
- f) Las tripulaciones de vuelo de avión comercial debe recibir por lo menos ocho horas de descanso ininterrumpido en cada período de 24 horas.

Acción de facilitadores (posibilita)

Las acciones facilitadoras son aquellas posibilidades en las que desencadena una regla bajo determinadas condiciones. La norma podría dar lugar a la especificación de algunas funciones de software, cuyas condiciones conducen a la acción de complejas combinaciones de valores lógicos.

En (Wiegers, 2003) se indica que las tablas de decisión son una buena alternativa para documentar las reglas de negocio que consideran cuestiones lógicas. Cuando se realiza una declaración de la forma “Si-entonces” es un indicio de que se está especificando una acción facilitadora.

A continuación se presentan algunos ejemplos de acciones facilitadoras.

- Si el almacén tiene el producto A que se solicita en stock, **entonces** se ofrece el producto en existencia al solicitante.
- Si la fecha de vencimiento del producto A en stock ha expirado, **entonces** notificar al personal encargado de bodega.

Inferencias

La inferencia (también llamada inferir el conocimiento), consiste en la generación de nuevo conocimiento en base a unas reglas que cumplen con ciertas condiciones. Una inferencia permite crear un hecho nuevo a partir de otros hechos o cálculos.

Una inferencia también consiste en declararla usando la forma “si - entonces” de acuerdo a las acciones que permitan las reglas de negocio. Para este caso la cláusula “entonces” implica un hecho o parte de información, no una acción a tomar. Algunos ejemplos de inferencias son:

- Si el pago no es recibido dentro de los 30 días calendario a partir de la fecha, entonces la cuenta está en mora.
- Si no se puede enviar el pedido en un plazo de 10 días a partir de la notificación de pago, entonces el pedido está en estado pendiente de envío.

Cálculos

Esta clase de reglas de negocio se ejecutan usando fórmulas matemáticas o algoritmos. Muchos cálculos se derivan de reglas externas a la organización, como es el caso, en nuestro medio las retenciones por concepto de venta. Estas reglas de cálculo permiten especificar requerimientos de software de acuerdo a como se las va descubriendo.

Las reglas de cálculo pueden ser expresadas en formato de texto o alternativamente representarse de forma simbólica, a continuación se especifican algunos ejemplos:

- Hay un descuento a los productos de categoría A:
 - o del 5% al valor unitario cuando el pedido está entre 10 y 15 unidades,
 - o del 10% al valor unitario cuando el pedido está entre 16 y 20 unidades y
 - o del 15% al valor unitario cuando son más de 20 unidades.
- El envío de la orden dentro de la localidad y que pesa hasta 2 kilos es de 12.50 dólares, a partir de allí 15 centavos por cada 100 gramos.



Actividad recomendada

Defina algunos ejemplos de reglas de negocio de acuerdo a la clasificación que se realiza en la figura 4.22.

Documentar las reglas del negocio

Al inicio del proyecto un simple catálogo será suficiente pero de acuerdo al desarrollo de las actividades del proyecto o aquellas grandes organizaciones cuyas operaciones de negocio o sistemas de información son complejas, las reglas deben establecerse en una base de datos de reglas de negocio. Existen herramientas comerciales que permiten la gestión de las reglas cuando el catalogo es demasiado grande y no se puede realizar con un tradicional procesador de texto.

La documentación de las reglas de negocio tiene que ser de lo más sencillo posible, de tal manera que el equipo de desarrollo lo utilice con eficacia; la documentación se realizará sin utilizar complejos catálogos que confundan a cualquier persona que haga uso de las mismas.

Al adquirir cierta experiencia para definir las reglas de negocio se deben ir documentando en plantillas estructuradas para ir definiendo y clasificando en los tipos apropiados. Comúnmente las plantillas describen patrones de palabra clave y frase que estructuran las normas de forma coherente. También es factible utilizar una base de datos, una herramienta de gestión, un gestor de reglas, un motor de reglas; sin embargo para empezar se puede probar con la plantilla que se muestra a continuación. (Wieggers, 2003)

La plantilla para documentar las reglas deberá tener los siguientes datos:

- Un identificador.
- Una definición de la regla.
- Pertenecer a una categoría (Hecho, Restricción, Acción facilitadora, Inferencia o Cálculo).
- Pertenecer a un grupo.
- Si es estática o dinámica.
- Fecha en que será efectiva (si es el caso)
- Fuente (de ser necesario)
- En que caso de uso se utiliza (en el momento apropiado).



Ejemplo:

A continuación se presenta una tabla en la que se indica mediante un ejemplo las reglas de negocio.

ID	Definición de la regla	Categoría	Grupo	Estática o Dinámica
HEC-001	Cada contenedor de productos químicos tiene un código de barras de identificador único.	Hecho	Política corporativa	Estática
HEC-002	Cada elemento de línea en un pedido representa una combinación específica de químicos, grado, tamaño del envase, y el número de contenedores.	Hecho	Política corporativa	Estática
REC-012	Todas las aplicaciones de software deben cumplir con las regulaciones gubernamentales para el uso por personas con discapacidad visual	Restricción	Política corporativa	Dinámica

Como se puede ver en la tabla anterior la columna de ID nos permite identificar y hacer el seguimiento respectivo de la norma con respecto a los requerimiento funcionales. La columna Tipo para saber la categoría o clase de regla. La columna de ESTATICA O DINAMICA para hacer el seguimiento de la regla y ver si cambia con el tiempo. La columna de ORIGEN para saber si son políticas corporativas o de gestión.

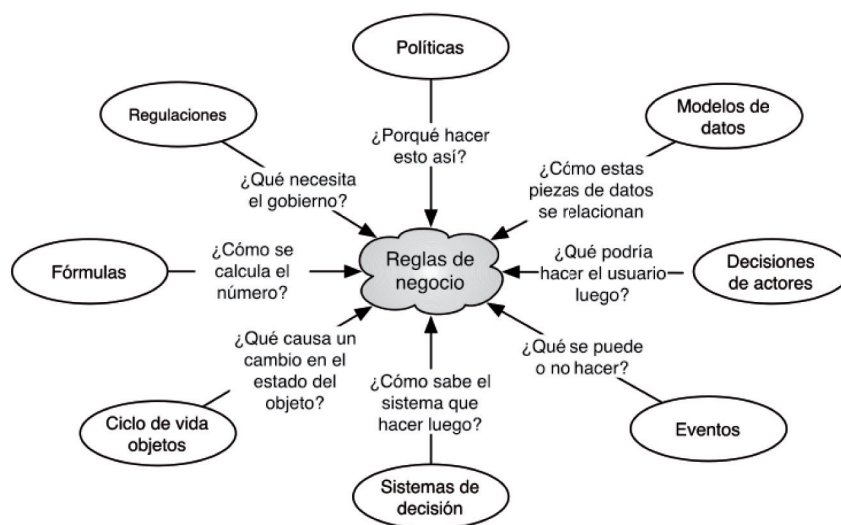


Figura 4.23: Descubrir reglas de negocio mediante preguntas. (Wiegiers, 2003)

Durante la captura de requisitos, al aplicar cualquier técnica el analista deberá descubrir las reglas, ya que en la mayoría de los casos, estas no están definidas. El analista deberá establecer las fuentes así como las preguntas necesarias para descubrirlas. En (Wiegiers, 2003) se sugiere algunas fuentes en la que se relaciona una pregunta para descubrir el contexto; en la figura 4.23 se muestra estas fuentes.

Luego de identificar y documentar las reglas de negocio es necesario determinar cuáles serán implementadas en la solución. Algunas normas se incluirán en los casos de uso por lo que se incluirán en los requisitos funcionales que harán cumplir la norma.



Al iniciar un proyecto de desarrollo de software, es necesario conocer el contexto de forma general de tal manera que cada vez se aplique alguna de las técnicas de levantamiento de información se pueda ir conociendo al detalle a la empresa u organización.

El analista de requerimientos juega un importante rol a la hora de utilizar las estrategias para dominar el contexto de la organización, será quien defina cuales son los interesados más idóneos y que pueden colaborar en el equipo de trabajo. Para poder trabajar de forma coordinada y armónica los interesados deberán estar dispuestos a realizar los esfuerzos necesarios para que el proyecto salga adelante.

Otro aspecto que es importante es lo relacionado a todas aquellas normas, políticas y leyes tanto a nivel interno o externo que hacen que se cumplan bajo ciertos criterios cada una de las actividades de un proceso determinado. Las restricciones son tan importantes que deben de ser debidamente tratadas y analizadas. El obviar alguna de ellas será motivo de serios inconvenientes ya que no enfocaría apropiadamente los requerimientos.

4.9. Priorizar los requerimientos

La priorización de requerimientos es el proceso de asignación, un precedente que ordene o clasifique un requerimiento sobre otro. Los Stakeholders necesitan conocer sobre la importancia de los requerimientos que les permitan tomar decisiones inteligentes y defendibles acerca de que los requisitos para implementarlos o aplazarlos. No todos los requisitos son igual de importantes y rara vez hay suficiente tiempo y dinero para poder desarrollar todos los requisitos (Gottesdiener E. , 2005).

Es necesario hacer la priorización para asignar los recursos a los requisitos más importantes y tomar decisiones acerca de qué y cuándo implementar los requerimientos . La priorización también ayuda a determinar cuando implementar requerimientos en caso que se desarrolle de forma incremental.

Para hacer la priorización se necesita realizar lo siguiente:

1. Identificar y organizar los requerimientos que necesita priorizar.
 - Asegúrese que todos los requisitos se encuentran en el mismo nivel de detalle.
 - Identificar los componentes o grupos de requisitos que son suministrados por otros.
 - Identifique los requerimientos que son dependientes.
 - Identificar los requerimientos que son interdependientes y que actúan solos.
 - Documente los requerimientos en una matriz de trazabilidad.
2. Reúna al equipo de Stakeholders a participar en el proceso de priorización.
 - Para software comercial incluya usuarios finales y personas de desarrollo del producto, vendedores, reguladores del departamento y agencias, y soporte técnico. Para desarrollos de sistemas internos incluya a usuarios finales, personas del departamento regulatorio, al product champion y personal de soporte técnico.
 - Incluir al personal técnico (por ejemplo, los diseñadores, los desarrolladores, y el director del proyecto) como asesores del proceso de priorización. El personal técnico debe estar familiarizado con el software existente y con los riesgos asociados con el proyecto.



- Mantenga un equipo de priorización pequeño (es decir, menos de siete personas). En proyectos grandes, puede que tenga que aumentar este número para asegurarse de que el proceso de priorización son bien equipadas, que todos los interesados están familiarizados con los requisitos que dan prioridad, y con el uso de las reglas de decisión y un proceso de toma de decisiones.

3. Identifique los criterios a considerar.

Los criterios son factores que ayudan a evaluar la importancia relativa de los requisitos basados en qué tan bien cumplen con la exigencia que se desea alcanzar.

4. Determinar la importancia relativa de los criterios.

- Comparar los criterios en pares. Pregunte: "¿Es el <Criterio A> más importante que el <Criterio B>?"
- Asignar un mayor peso a los criterios que son más importantes que otros.

5. Crear una matriz de criterios para demostrar la fuerza de la correlación entre la exigencia y los criterios.

- Coloque las características (es decir, conjuntos de casos de uso, lógicamente relacionadas declaraciones requisitos, o agrupaciones de cualquier requisitos relacionados lógicamente que se define en el primer paso) en las filas de la matriz. Cada fila debe incluir una o más características que pueden ser liberadas de forma independiente.
- Utilice una escala de 3, 6, 9 (o mecanismo de clasificación de otro tipo) para separar el valor de la clasificación. Los requisitos con una puntuación baja correlacione con un 3, los que tienen una correlación media con un 6, y los que tienen una fuerte correlación con un 9.
- Pida a los participantes evaluar cada criterio y discutir su clasificación.
- Calcular la importancia de cada característica para totalizar la clasificación de cada criterio.
- Repita este proceso para cada característica.
- Haga que los interesados revisen los resultados discutirlos, explicando sus puntuaciones.

6. Decida qué requerimiento desarrollar.

También se puede usar un conjunto de criterios, como es: valor, costo y riesgos para construir una matriz de priorización.



- Valor: Esto compuesto por: el beneficio o la penalización para el cliente o el negocio si el requerimiento es implementado.
- Costo: Es el costo de implementar la función, teniendo en cuenta el esfuerzo, recursos y costos de capital.
- Los riesgos: Son los riesgos técnicos relacionados con el desarrollo del requisito.

Se ha concluido el estudio de esta unidad, por lo que conviene comprobar cuánto se ha asimilado de los temas tratados para lo cual es necesario desarrollar las autoevaluaciones de conocimiento.



Autoevaluación 4



Realice las actividades que se indican en cada uno de los literales

Lea detenidamente la afirmación y escriba una V si considera que es verdadera o una F si no lo es, en el paréntesis respectivo.

1.	Un proceso de negocio es un conjunto de tareas relacionadas que crea algo de valor para el negocio.	()
2.	Cuando el alcance del proyecto no es claro o es demasiado grande no es conveniente realizar un modelado del negocio.	()
3.	El modelado de negocios requiere del patrocinio y participación de los clientes.	()
4.	Los modelos de requisitos le ayudan a descubrir requisitos faltantes, erróneos, ambiguos y contradictorios.	()
5.	Los dominios estructurales manejan procesos de negocio y tareas tales como: operaciones de negocios y administración, procesamiento de pedidos y gestión de inventario	()
6.	Los dominios dinámicos responden a los acontecimientos en constante cambio para almacenar datos y actuar en función de su estado en un punto en el tiempo	()
7.	Un mapa de relaciones es un diagrama que muestra qué tipo de información y productos que se intercambian entre los clientes externos, proveedores y las funciones clave de la organización.	()
8.	Los mapas de proceso se utiliza para identificar qué procesos se realizan de forma manual y qué se destinará al software.	()
9.	Un caso de uso muestra al sistema en su entorno, con las entidades externas (es decir, personas y sistemas) que proporcionan y reciben información o material desde y hacia el sistema.	()
10.	Las políticas de negocio son las directrices, normas y reglamentos que rigen o condicionan la conducta de un negocio.	()

2. Liste las partes del ciclo de análisis de requerimientos.

- _____
- _____
- _____
- _____
- _____



2. Complete la siguiente tabla acerca de las herramientas y técnicas para análisis de requerimientos.

Cuando necesite:	Entonces crear:
a) Modelar el negocio	Combinar entre mapa de relaciones y/o mapa de procesos.
b)	Combinar entre diagramas de contexto, tablas de evento-respuesta y/o políticas de negocio
c)	Combinar o variación de: tabla de actores, casos de uso, mapas de dialogo, modelo de datos, diagramas de estado y/o reglas de negocio
b)	Priorizar requerimientos





SEGUNDO BIMESTRE

6.5. Competencias genéricas

- Capacidad de abstracción, análisis y síntesis
- Habilidades para buscar, procesar y analizar información procedente de fuentes diversas.
- Capacidad para identificar, plantear y resolver problemas
- Capacidad para tomar decisiones
- Capacidad de trabajo en equipo
- Capacidad para formular, diseñar y gestionar proyectos

6.6. Planificación para el trabajo del alumno

COMPETENCIAS ESPECÍFICAS	INDICADORES DE APRENDIZAJE	CONTENIDOS	ACTIVIDADES DE APRENDIZAJE	CRONOGRAMA ORIENTATIVO Tiempo Estimado
		UNIDADES/TEMAS		
Aplicar técnicas de modelado y documentación de especificaciones de software.	• Documenta las necesidades de los usuarios. • Identifica medios de verificación de las necesidades de los usuarios • Identifica los elementos para documentar los requerimientos de software • Identifica modelos de verificación y validación de los requerimientos de Software	UNIDAD 5: ESPECIFICACIÓN DE REQUERIMIENTOS 5.1 Proceso de especificación de requerimientos. 5.2 Técnicas y herramientas para especificar requerimientos.	• Revisión de contenidos del capítulo 5 en el texto guía. • Lectura comprensiva. • Desarrollo de actividades recomendadas en la guía y ejercicios propuestos en el texto básico. • Interacción en el EVA.	Semana 1, 2 y 3 • 12 horas de autoestudio • 12 horas de interacción.
		UNIDAD 6: VALIDACIÓN DE REQUERIMIENTOS 6.1. Contexto de la validación de requisitos. 6.2. Validación de Requisitos	• Revisión de contenidos del capítulo 6 en el texto guía. • Lectura comprensiva. • Desarrollo de actividades recomendadas en la guía y ejercicios propuestos en el texto básico. • Interacción en el EVA.	Semana 4 • 4 horas de autoestudio • 4 horas de interacción.



Gestionar adecuadamente los requisitos.	• Establecer criterios de sobre la gestión de requisitos • Utilizar técnicas y herramientas de gestión de requisitos.	UNIDAD 7: GESTIÓN DE LOS REQUISITOS 7.1 Gestión de Requisitos. 7.2 Herramientas y técnicas de gestión de requisitos.	• Revisión de contenidos del capítulo 7 en el texto guía. • Desarrollo de actividades recomendadas en la guía y ejercicios propuestos en el texto básico. • Interacción en el EVA.	Semana 5 y 6 • 8 horas de autoestudio • 8 horas de interacción.
		Revisión de las unidades 5, 6 y 7		Semana 7 y 8 • 8 horas de autoestudio • 8 horas de interacción.



6.7. Orientaciones específicas para el aprendizaje por competencias



UNIDAD 5: Especificación de Requerimientos



La Especificación de Requerimientos de Software (ERS) es también llamada una especificación funcional, la especificación de un producto, el documento de requerimientos, o la especificación del sistema, sin embargo las organizaciones no utilizan estos términos de la misma manera. La ERS establece las funciones y capacidades que el sistema de software debe proveer y las restricciones a tomar en consideración.

La ERS es la base para la planificación del proyecto, diseño y codificación, así como la base para las pruebas del sistema y la documentación de usuario. Se debe describir completamente el comportamiento del sistema bajo diferentes circunstancias, este, no debe contener detalles de diseño, construcción, pruebas o de gestión del proyecto y también restricciones de implementación.



La especificación de requerimientos es el proceso de elaboración, refinamiento y organización de los requisitos plasmados en un documento. La especificación es responsabilidad principal del analista de sistemas, pero se pueden incluir otros usuarios que permitan hacer las revisiones y validaciones e incluso aportar con la redacción de los requisitos.

Estimado estudiante, es necesario que ponga especial atención al proceso que se indica en la figura 5.1; ya que cualquiera sea la naturaleza del proyecto o el ámbito, los principales documentos que se elaboran en esta fase, es el objetivo del proceso de desarrollo de requisitos. Recuerde tal como se indicó en la unidad 1, el resultado del desarrollo es la especificación de los requisitos y la gestión gira en torno a estas especificaciones, por lo tanto la calidad del documento de especificación de requerimientos reflejará la calidad del sistema que se construya.

Recuerde que en la elicitación se aplican técnicas para poder obtener información, mientras que en el análisis se utilizan modelos que permiten organizar esa información; con todo esto se procede a documentar de acuerdo al proceso que se indica a continuación.

5.1. Proceso de especificación de requerimientos

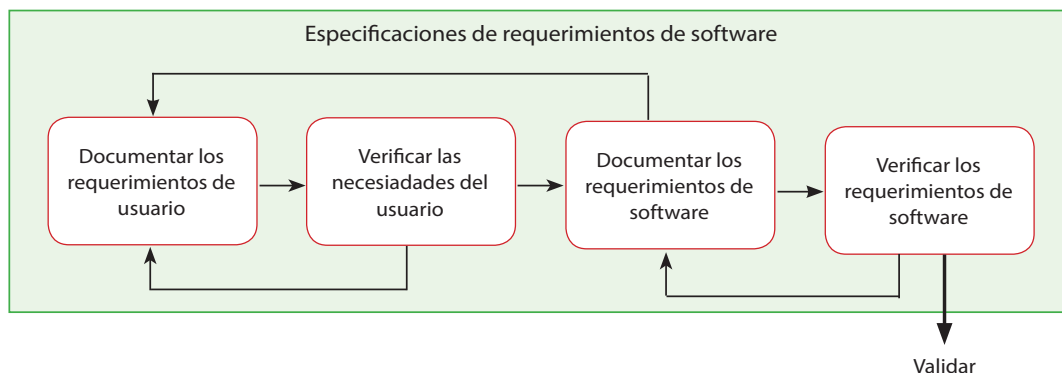


Figura 5.1. Proceso de especificación de requerimientos de software. (Gottesdiener E., 2005)

5.1.1. Documentar los requerimientos de usuario

Documentar los requisitos desde el punto de vista del usuario consiste en elaborar un documento de necesidades de los usuarios (que se describe más adelante). Incluyen los modelos de análisis y de la prosa narrativa. La documentación permite Describir las características y el comportamiento del sistema que se propone desde el punto de vista del usuario. Esta descripción actúa como un puente entre las necesidades del usuario y la especificación de requisitos de software. Más adelante se estará analizando y describiendo la plantilla utilizada para el efecto.

5.1.2. Verificar las necesidades del usuario

Compruebe que los requisitos de los usuarios describen lo que los usuarios tienen que ver con el sistema. Asegúrese de que los requisitos se derivan de los requerimientos del negocio (es decir, la visión del producto, metas y objetivos del proyecto). También es necesario que los Stakeholders comprueben que los requisitos estén completos, consistentes y de alta calidad. Revise la documentación, según sea necesario.

5.1.3. Documentar los requerimientos de software

Documentar o registrar los requisitos de software en definitiva es realizar una especificación de requisitos de software que consiste en escribir el documento de especificación para el público profesional (que proporcionan el software). Esta especificación describe los requisitos funcionales, los atributos de calidad, las interfaces del sistema, y las limitaciones de diseño e implementación. Mas adelante se analizará la plantilla utilizada para el efecto.

5.1.4. Verificar los requerimientos de software

En esta fase asegúrese de que la documentación describe correctamente las capacidades de intención y las características del sistema. Comprobar que los requisitos de software han sido derivados de forma precisa de las necesidades del usuario, de los requisitos del sistema, y otras fuentes que se utilizaron.

Ahora deberá centrarse en la documentación de los requerimientos; para esto la IEEE a través de sus estándares facilita plantillas que pueden ser ajustadas a los estándares de la organización; de igual forma las metodologías de hoy en día como son: RUP, Volere, entre otras han establecido sus plantillas para proceder con ésta documentación. Las plantillas que se indican en esta guía tómelas como referencia para desarrollar el caso práctico y ajústelas de acuerdo a su conveniencia.

5.2. Técnicas y herramientas para documentar requerimientos



Como se vio en el apartado anterior y concretamente en la figura 5.1, es necesario realizar la documentación de los requerimientos tanto a nivel de usuario, como a nivel de profesionales (especificación). Por lo que (Gottesdiener E., 2005) propone desarrollar dos documentos, tal como se indica en la tabla 5.1.

Es necesario comprobar que la documentación se ajuste a las plantillas de la organización, las normas y convenciones con los nombres (fundamental para contextualizar). Establecer procedimientos para el control de cambios en la documentación para controlar cualquier cambio a los documentos.


Tabla: 5.1: Herramientas y técnicas para especificación de requerimientos

Cuando necesite:	Entonces crear:
Documentar y verificar requerimientos de usuario.	El documento de requerimientos de usuario.
Documentar y verificar requerimientos de software	La especificación de requerimientos de usuario

5.2.1. Documento de requerimientos de usuario

Un documento de requerimientos de usuario es un registro de los requisitos escritos para una audiencia de usuarios que describen lo que los usuarios necesitan y porque son necesarios.

Este documento se lo usa para obtener un acuerdo sobre lo que el producto tiene que hacer para satisfacer las necesidades de los usuarios, para consolidar las necesidades de las comunidades de usuarios diversos, y para proporcionar detalles adicionales no descritos por los modelos de análisis. Este documento de requerimientos de los usuarios también actúa como un puente entre la definición del negocio y los requisitos de software.

Para hacer esto debe realizar lo siguiente:

- Identificar las fuentes para el documento de requisitos de usuario.
- Organizar los requerimientos del usuario en el documento de requisitos de usuario.

1. Identificar las fuentes para el documento de requisitos de usuario

- Incluya la visión del producto, carta del proyecto, los modelos de análisis, el usuario de procedimiento de documentación (por ejemplo, manuales, procedimientos operativos estándar y materiales de capacitación), la documentación del sistema actual, y cualquier otra documentación acerca de las necesidades del usuario.
- Decida sobre el formato del documento para los requisitos. (se sugiere un formato que se indica mas adelante, o puede utilizar plantillas de documentos estándar de la organización, si están disponibles). Utilice la documentación más rica cuando la externalización del desarrollo de un proveedor externo, o si el producto es un sistema complejo o crítico. En el anexo 2 se especifica el “Esquema de requerimientos e usuario” que sería de mucha utilidad.

2. Organizar los requerimientos del usuario en el documento de requisitos de usuario

- Utilice los modelos de análisis (por ejemplo, casos de uso de los, un diagrama de contexto para ilustrar “Interfaces con otros sistemas”, y reglas de negocio en el “Impacto de las políticas, normas y reglas de negocio”) para estructurar el documento.
- Revisar el documento desde la perspectiva de los lectores de varios negocios (es decir, el patrocinador del proyecto, administradores de empresas, gerentes de marketing o de producto, instructores y usuarios).

- Compruebe que en el documento se utilice la terminología de los usuarios, eliminando la jerga técnica.
- Asegúrese de que el lenguaje en el documento sea claro.



Actividad recomendada

Revise la plantilla que se indica en el Anexo 3, y desarrolle cada uno de los puntos de acuerdo a la información obtenida al aplicar los modelos de análisis de la unidad 4.

5.2.2. Documento de especificación de requerimientos de software

El documento para la especificación de requisitos de software (ERS) es un registro preciso de los requisitos que permite a los proveedores de software diseñar, desarrollar y probar la solución de software. El documento de ERS⁹ contiene el conjunto completo de los requerimientos funcionales y no funcionales que el producto de software debe cumplir.

El estándar IEEE 830 define los beneficios de una buena especificación de requerimientos de software:

- Establece las bases para lograr un acuerdo entre los clientes y los proveedores en lo que el producto de software debe hacer. La descripción completa de las funciones a realizar por el software especificado en el documento de especificación de requerimientos asistirá a los usuarios potenciales a determinar si las especificaciones cumplen con sus necesidades o como el software debe ser modificado para cumplir las mismas.

Tabla: 5.2: Plantilla para especificar requerimientos de software (ERS)

1. Introducción
 - 1.1. Propósito
 - 1.2. Convenciones del documento
 - 1.3. Público objetivo y sugerencias de lectura
 - 1.4. Alcance
 - 1.5. Referencias
2. Descripción general
 - 1.1. Perspectiva del producto
 - 1.2. Funciones del producto
 - 1.3. Clases y características de usuario
 - 1.4. Ambiente de operación
 - 1.5. Restricciones de diseño e implementación
 - 1.6. Documentación de usuario
 - 1.7. Suposiciones y dependencias

⁹ Referencia: IEEE STD 830-1998



3. Características del sistema
 - 2.1. Características del sistema X
 - 2.1.1. Descripción y prioridades
 - 2.1.2. Secuencia estímulo/respuesta
 - 2.1.3. Requerimientos funcionales
 - 2.2. Características del sistema ...
 3. Requerimientos de interfaz externa
 - 3.1. Interfaz de usuario
 - 3.2. Interfaz de hardware
 - 3.3. Interfaz de software
 - 3.4. Interfaz de comunicaciones
 4. Requerimientos no funcionales
 - 4.1. Requisitos de rendimiento
 - 4.2. Requerimientos de seguridad
 - 4.3. Atributos de calidad del software
 5. Otros requerimientos
- Apéndice A: Glosario
- Apéndice B: Modelos de análisis
- Apéndice C: Listado de temas

- Reducir el esfuerzo en el desarrollo: La preparación de la especificación de requerimientos de software (ERS) forza a los diversos grupos interesados dentro de la organización a considerar rigurosamente todos los requerimientos antes de que el diseño inicie y con esto reducir el rediseño, recodificación y retesteo. La revisión cuidadosa de los requerimientos (ERS) puede revelar de forma temprana omisiones, malentendidos e inconsistencias en el ciclo de desarrollo cuando estos problemas son fáciles de corregir.
- Proveer bases para la estimación de costos y cronogramas: La descripción del producto a ser desarrollado tal como se indica en la ERS es una base real para la estimación de costos del proyecto y puede ser usada para obtener aprobación de las ofertas o estimaciones de precios.
- Proveer una línea base para la validación y verificación: Las organizaciones pueden desarrollar sus planes de validación y verificación de manera más productiva desde un buen documento de especificación de requerimientos. Como parte del contrato de desarrollo, el ERS provee una línea base contra la cual se puede medir el cumplimiento. (NOTA: Se utiliza el documento de especificación de requerimientos ERS para crear el Plan de Pruebas).
- Facilitar la transferencia: El ERS hace fácil la transferencia del producto de software a nuevos usuarios o nuevos equipos. Para los clientes por lo tanto es más fácil transferir el programa a otras partes de su organización, y a los proveedores les resulta más fácil de transferir a los nuevos clientes.

Nuevamente tomando como referencia el estándar IEEE 930 un ERS debe considerar algunos atributos de calidad que permitan gestionar el documento durante su construcción, entre estos tenemos:

Estimado estudiante es necesario que nuevamente tome como referencia el estándar IEEE 930, un ERS debe considerar algunos atributos de calidad que permitan gestionar el documento durante su construcción, entre estos tenemos:

- **Correcto:** La ERS es correcta si y solo si todo requisito que figura aquí (y que será implementado en el sistema) refleja alguna necesidad real. La corrección de la ERS implica que el sistema implementado será el sistema deseado.
- **No ambiguo:** Cada requisito tiene una sola interpretación. Para eliminar la ambigüedad inherente a los requisitos expresados en lenguaje natural, se deberán utilizar gráficos o notaciones formales. En el caso de utilizar términos que, habitualmente, poseen más de una interpretación, se definirán con precisión en el glosario.
- **Completo:** Todos los requisitos relevantes han sido incluidos en la ERS. Conviene incluir todas las posibles respuestas del sistema como los datos de entrada, tanto fallidos como los válidos.
- **Consistente:** Los requisitos no pueden ser contradictorios. Un conjunto de requisitos contradictorio no es implementable.
- **Clasificado:** Normalmente, no todos los requisitos son igual de importantes. Los requisitos pueden clasificarse por su importancia (esencial, condicional u opcional) o por su estabilidad (cambios que se espera que afecten al requisito). Esto sirve, ante todo, para no emplear excesivos recursos en implementar requisitos no esenciales.
- **Verificable:** La ERS es verificable si y solo si todos sus requisitos son verificables. Un requisito es verificable (testeable) si existe un proceso finito y no costoso para demostrar que el sistema cumple con el requisito. Un requisito ambiguo no es en general verificable. Expresiones como: a veces, bien, adecuado, etc., introducen ambigüedad en los requisitos. Requisitos como: en caso de accidente la nube tóxica no se extender más allá de 25Km” no es verificable por el alto costo que conlleva.
- **Modificable:** La ERS es modificable si y sólo si se encuentra estructurada de forma que los cambios a los requisitos pueden realizarse de forma fácil, completa y consistente. La utilización de herramientas automáticas de gestión de requisitos (por ejemplo RequisitePro) facilitan enormemente esta tarea.
- **Trazable:** La ERS es trazable si se conoce el origen de cada requisito y se facilita la referencia de cada requisito a los componentes del diseño y de la implementación. La trazabilidad hacia atrás indica el origen (documento, persona, etc.) de cada requisito. La trazabilidad hacia delante de un requisito R indica que componentes del sistema son los que realizan el requisito R.

Tenga presente que para documentar los requisitos en el documento de especificación es necesario realizar los siguientes pasos:

1. Identificar y etiquetar las características necesarias para alcanzar los objetivos del software.
2. Descomponer y documentar los requerimientos funcionales dentro de cada función.



3. Verificar las declaraciones de los requisitos funcionales.
4. Identificar y cuantificar los atributos de calidad.
5. Cuantificar los requerimientos funcionales.
6. Asociar requisitos relacionados.
7. Identificar las limitaciones de diseño e implementación.
8. Identificar los requisitos de interfaz externa.
9. Quitar las soluciones de diseño.
10. Identificar y corregir los requisitos omitidos, en conflicto, y la superposición.
11. Priorizar todos los requisitos, agregar atributos a los requerimientos, y trazar las prioridades y atributos de cada requerimiento.
12. Organizar los requerimientos en una plantilla de ERS, completando cada sección.
13. Chequear la calidad del documento ERS.

Dada la importancia del documento, le invitamos a realizar una revisión detallada de las actividades que se indican en el listado anterior.

1. Identificar y etiquetar las características necesarias para alcanzar los objetivos del software

- Identifique las características de los requisitos de información, para lo cual deberá revisar: la declaración de la visión del producto, los modelos de análisis, la documentación del usuario, e información de las actividades realizadas como parte de la elicitación de requerimientos. Por cuanto las características son un conjunto de capacidades necesarios para alcanzar los objetivos del negocio.
- Asigne un nombre a las características con una descripción corta por ejemplo “Horario de trabajo” o utilizar un gerundio como “programación”. Incluir una etiqueta mediante letras, números o abreviaturas. Por ejemplo: “REQ-F-001”, “PRO.HOR”. Las etiquetas son importantes para distinguir unos de otros requisitos al mismo tiempo que permite documentar cómo los requisitos se descomponen.

2. Descomponer y documentar los requerimientos funcionales dentro de cada función

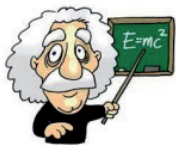
- Descomponga las características en los requerimientos funcionales. Visualizar el resultado final como una jerarquía.
- Escribir frases cortas y concisas usando imperativos para describir los requisitos funcionales (por ejemplo, “El sistema le pedirá que indique el programador de la fecha de empleo”). Utilice una estructura de la frase estándar, tales como:

[<restricción>] <asunto> <acción del verbo>
[<resultado observable>] [<calificador>]

Donde:

- [**<restricción>**] Identifica las condiciones en que debe cumplir el requerimiento.
- **<asunto>**: Muestra quién o qué está haciendo la tarea (por lo general “el sistema” o un actor).
- **<acción del verbo>**: es la tarea a ejecutar.
- [**<resultado observable>**] muestra el resultado de la acción.
- [**<calificador>**]: Identifica los atributos de calidad para la exigencia.

Nota: Los corchetes “[]” indican componentes opcionales de la frase.



Ejemplo

Describiendo requerimientos funcionales

Ejemplo sin restricción: “El sistema permitirá que una secretaria seleccione el trámite”.

Ejemplo con restricción: “Cuando no esté el profesor la secretaria de carrera deberá registrar la nota correcta al estudiante.”

Ejemplo con restricciones, resultado observable y calificador: “Cuando una secretaria de centro registra el trámite el sistema deberá enviar un correo electrónico al estudiante con los datos acerca del trámite”.

- Descomponer las declaraciones de requisitos complejos o compuestos en múltiples sentencias. En las instrucciones complejas describir la secuencia y el flujo. En las sentencias compuestas utilice las conjunciones (por ejemplo: y, o, también, con).
- Use las continuaciones (es decir, frases que siguen a un imperativo e introducir un requisito de nivel inferior) para descomponer los requisitos de las declaraciones (por ejemplo, “El sistema deberá proporcionar una lista de contratistas disponibles en el siguiente orden :...”). Las continuaciones son “más adelante:”, “de la siguiente manera:”, “lo siguiente:”, “lista”, y “en particular”.
- Proporcionar ejemplos para ilustrar. Utilice “por ejemplo” o “este es un ejemplo”.
- Divida las cláusulas condicionales anidadas en declaraciones por separado.
- Buscar excepciones (como: “no”, “sí”, “pero”, “a menos”, “a pesar” y “menos”) y dividirlos en declaraciones de distintos requisitos (los requisitos con excepciones puede reflejar una regla de negocio y resultar reglas de negocio adicional).
- Citar el documento de reglas de negocio como una referencia externa o hacer referencia a ellos en el apéndice.
- Utilice tablas o gráficos para explicar los requisitos complejos. Establezca un título a cada tabla o gráfico con un identificador único para su fácil identificación. Cite las referencias de forma clara y correctamente con un identificador único (por ejemplo, “Ver Estados de empleo, el Apéndice B,



Figura 5") y los documentos externos citar totalmente con el nombre del documento, la ubicación, y un identificador único.



Un caso de uso puede traducirse en múltiples declaraciones de requisitos funcionales dentro de una característica. Cada paso de casos de uso es un candidato de bajo nivel de requerimiento funcional dentro de cada función.

Ejemplo:

Múltiples declaraciones de requerimientos funcionales dentro de una característica

1. Registrar trámite	
1.1. Registrar documentación trámite	Nombre de la característica
1.1.1. El sistema permitirá que la secretaria del centro seleccione el tipo de documento	Nombre del caso de uso Paso del caso de uso



Actividad recomendada

De acuerdo al caso de desarrollo, identifique:

- Requerimientos funcionales sin restricción
- Requerimientos funcionales con restricción
- Requerimientos funcionales con restricción, resultado y calificador.

3. Verificar las declaraciones de los requisitos funcionales.

- Asegúrese de definir cada término de negocio utilizadas en los requisitos en el glosario.
- Asegúrese que pueda verificar cada declaración del requisito. Asegúrese de escribir de forma clara y distintivamente los algoritmos, las decisiones y condiciones y cada documento en un solo lugar de la ERS.
- Involucrar a los probadores en la revisión o el desarrollo de requisitos para asegurar que los requisitos pueden ser probados.
- Asegúrese de definir todos los datos necesarios para satisfacer los requisitos funcionales en el modelo de datos (o modelo de clases o diccionario de datos). Incluya el modelo de datos, ya sea en el apéndice del modelo de análisis o de la sección de requisitos funcionales de la ERS.
- Eliminar o aclarar cualquier requisito señalado como "por determinar".

4. Identificar y cuantificar los atributos de calidad.

- Describir los atributos de calidad como las características de funcionamiento del software, el desarrollo y despliegue.
- Revise la lista de atributos de calidad y seleccionar aquellas que resulten aplicables. (Figura 5.2).

- Especificar métricas para todos los atributos de calidad. Proporcionan una escala de medición (es decir, las unidades que se utilizan para pruebas de conformidad del producto), junto con los plazos, los valores de aceptación, los valores mínimo y máximo, o cualquier otras medidas necesarias para pruebas de conformidad.

5. Cuantificar los requerimientos funcionales

- Asignar medidas o criterios explícitos a los requerimientos funcionales. Relacionar la cuantificación de la exactitud de los resultados, aspecto y sensación (usabilidad), seguridad, mantenimiento, portabilidad, o la realización de la funcionalidad. Considere la velocidad de respuesta (por ejemplo, el tiempo de respuesta en segundos), el rendimiento (por ejemplo, número de transacciones por período de tiempo), capacidad (por ejemplo, el número de usuarios concurrentes), y el tiempo de ejecución para hardware y software (por ejemplo, completar una operación de un brazo robótico en menos de 100 milisegundos) en su cuantificación del rendimiento.

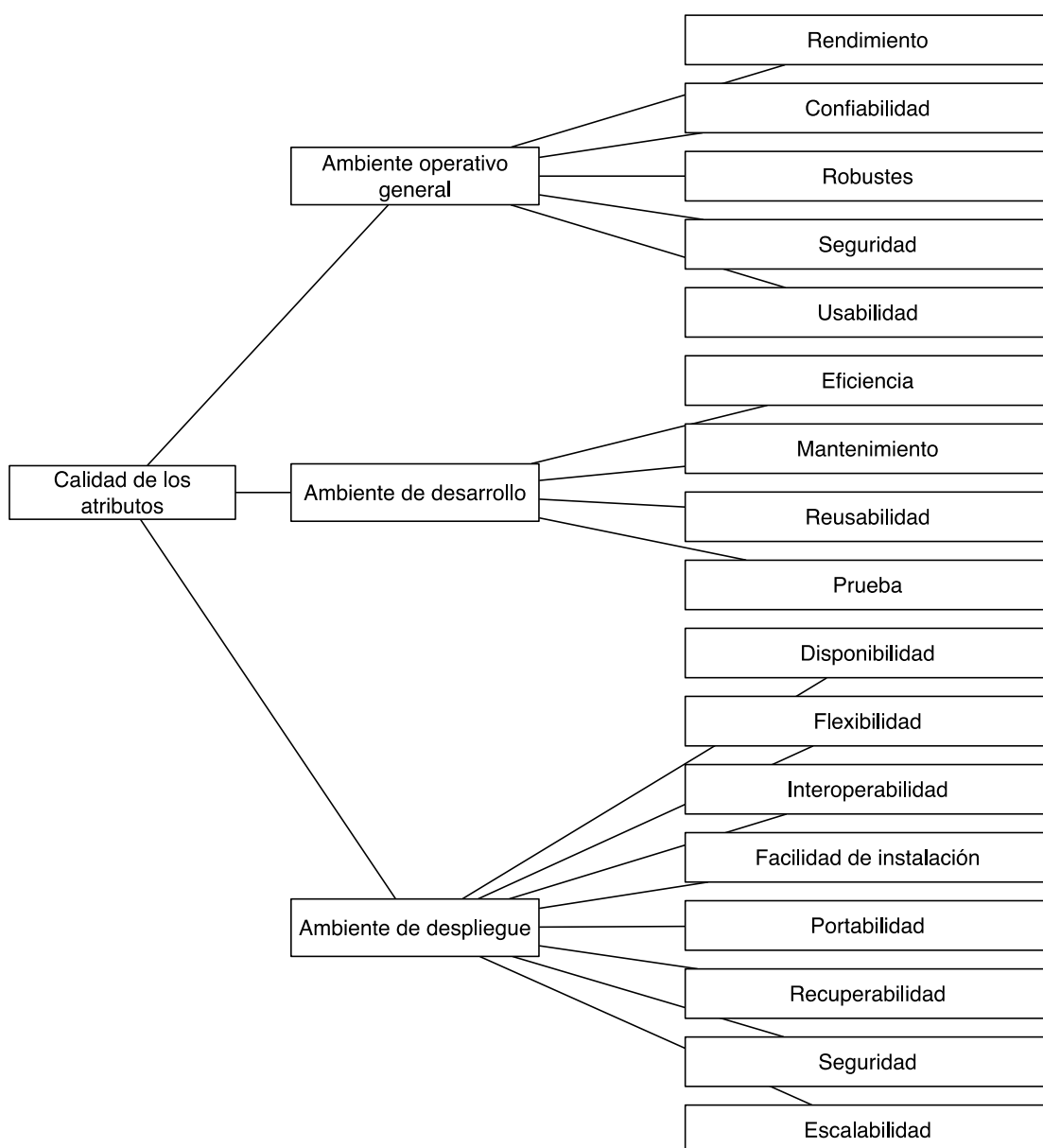


Figura 5.2. Calidad de los atributos.



Actividad recomendada

De acuerdo a la figura 5.2, identifique los atributos de calidad que deberán asociarse al caso de desarrollo.

6. Asociar requisitos relacionados.

- Proporcionar los requisitos funcionales, con referencias a los atributos relacionados con la calidad. (Un atributo de calidad puede ser requerido por varios requisitos funcionales. Por ejemplo, los tiempos de respuesta de ciertos requisitos de fiabilidad y puede ser requerido por una serie de requisitos funcionales).
- Se pueden mostrar los requerimientos relacionados en una matriz.

7. Identificar las limitaciones de diseño e implementación.

- Considere un lenguaje apropiado de diseño y desarrollo, herramientas, formatos de intercambio de datos, y tanto convenios como normas para la programación y el diseño.
- Regulaciones y políticas.
- Las restricciones impuestas por las interfaces de hardware, tales como límites de utilización de la memoria, los límites del procesador, el tamaño o peso.
- Especifique las versiones, proveedores, y cualquier otra información de identificación.

8. Identificar los requisitos de interfaz externa.

- Documento de cada componente de la interfaz y defina el formato de cada interfaz. Especifique cada interfaz con el nombre, versión, vendedor, y cualquier otra información de identificación.
- Considere características de la apariencia de todas las interfaces de usuario y de los componentes de hardware.
- Considere los componentes de software, tales como: los sistemas operativos y navegadores, el software de comunicación que representa y la transferencia de datos entre sistemas informáticos o redes, el software de red que monitorea y controla el intercambio de información en un entorno de red, el directorio de software que mantiene el conocimiento de la ubicación de los recursos de la red, y componentes e interfaces comerciales de otros sistemas de software.

9. Quitar las soluciones de diseño.

- Eliminar todas las restricciones sobre la forma en que el producto debe ser implementado a menos que sean las limitaciones legítimas de diseño para el desarrollo o la aplicación de la organización. Permita a los diseñadores encontrar las mejores alternativas, teniendo en cuenta las necesidades y limitaciones conocidos del diseño.

10. Identificar y corregir los requisitos omitidos, en conflicto, y la superposición.

- Usar escenarios para descubrir requisitos faltantes. Realizar escenarios de paseos virtuales del documento de requerimientos para detectar errores.
- Asociar los casos de uso con las declaraciones de requisitos, si los casos de uso están disponibles. Almacenar la información en una matriz de trazabilidad de los requisitos como una ayuda para la detección de los posibles solapamientos o requisitos faltantes.

11. Priorizar todos los requisitos, agregar atributos a los requerimientos, y trazar las prioridades y atributos de cada requerimiento.

- Revisar las prioridades del análisis de requerimientos (revise la unidad anterior) y corregirlos cuando sea necesario. Asignar una prioridad a las necesidades en un nivel adecuado de detalle.
- Identificar los atributos que son importantes para definir acerca de los requisitos.

12. Organizar los requerimientos en una plantilla de ERS

- Use una plantilla como la que se muestra en la tabla 5.2; y dar formato al documento con la información de las especificaciones. Establecer la nomenclatura y numeración apropiadas para estructurar el documento.
- A continuación se realiza una descripción al detalle de la plantilla que se indica en la tabla 5.2. para especificar requerimientos de software (Wiegers, 2003).

1. Introducción

En esta sección se presenta una introducción a todo el documento de especificación de requisitos software(ERS). Consta de varias subsecciones.

1.1. Propósito

Identificar al producto o aplicación cuyos requerimientos son especificados en este documento, incluyendo la revisión o el número de versión. Si este ERS pertenece a una parte de un sistema mas grande, identifique la parte o subsistema que corresponde.

1.2. Convenciones del documento

Describe algunos estándares o convenciones tipográficas, incluye estilos del texto, partes importantes, o notaciones significativas.

1.3. Público objetivo y sugerencias de lectura

Lista los diferentes lectores para quienes está dirigido el ERS. Describe de forma general el contenido del ERS y la forma en que está organizado.



1.4. Alcance

Provee una pequeña descripción del software y su propósito. Es importante relacionar el software con el usuario o con los objetivos corporativos y de negocio, además de sus estrategias. Si se ha desarrollado un documento de visión y alcance por separado, es necesario referirse a este sin necesidad de incluir su contenido en este documento.

1.5. Referencias

Especifique una lista de documentos u otros recursos a los que se refiere el SRS, incluyendo enlaces a ellos si es posible. Estos pueden incluir: guías de estilos para interfaz de usuario, contratos, normas, especificación de requisitos del sistema, documentos de casos de uso, especificaciones de interfaz, documentos de conceptos de operaciones, o el ERS para un producto relacionado. Es necesario que se proporcione información suficiente para que el lector pueda acceder a cada referencia incluyendo el título, autor, número de versión, fecha y ubicación de origen o (como la carpeta de red o URL).

2. Descripción general

En esta sección se describen todos aquellos factores que afectan al producto y a sus requisitos. No se describen los requisitos, sino su contexto. Esto permitirá definir con detalle los requisitos en la siguiente sección, haciendo que sean más fáciles de entender.

2.1. Perspectiva del producto

En esta subsección deben relacionar el futuro sistema (producto software) con otros productos. Si el producto es totalmente independiente de otros productos, es aquí donde se debe especificar. Si la ERS define un producto que es parte de un sistema mayor, esta subsección relacionará los requisitos del sistema mayor con la funcionalidad del producto descrito en la ERS, y se identificarán las interfaces entre el producto mayor y el producto aquí descrito. Se recomienda utilizar diagramas de bloques.

2.2. Funciones del producto

En esta subsección se mostrará un resumen, a grandes rasgos, de las funciones del futuro sistema. Por ejemplo, en una ERS para un programa de trámites de la UTPL, esta subsección mostrará que el sistema soportará la actualización de datos del estudiante, registro de documentación, registro de acuerdo al trámite; todo esto sin mencionar el gran detalle que cada una de estas funciones requiere. Las funciones deberán mostrarse de forma organizada, y pueden utilizar gráficos, siempre y cuando dichos gráficos reflejen las relaciones entre funciones y no el diseño del sistema.

2.3. Clases y características de usuario

Se identifican los diferentes tipos de usuario que podrían utilizar el producto, describiendo sus características pertinentes. Algunos de los requisitos podría pertenecer sólo a determinado tipo de usuario. Identificar las clases de usuarios favorecidos. Las clases de usuarios representan un subconjunto de Stakeholders descritos en el documento de visión y alcance.

2.4. Entorno de funcionamiento

Describir el entorno en el que el software funcionará, incluida la plataforma de hardware, los sistemas operativos y versiones, la ubicación geográfica de los usuarios, servidores y bases de datos. También es necesario hacer una lista de otros componentes de software o aplicaciones con las que el sistema deben coexistir pacíficamente. El documento de visión y el alcance puede contener parte de esta información en un alto nivel.

2.5. Restricciones de diseño e implementación

Se describen los factores y justificativos que limitan a los desarrolladores del producto. Las restricciones incluyen:

- Tecnologías específicas, herramientas, lenguajes de programación y bases de datos que podrían ser usadas.
- Restricciones debido al entorno operativo del producto, tales como los tipos y versiones de los navegadores Web que se utilizará.
- Convenciones de desarrollo requeridos o normas. (Por ejemplo, si la organización cliente será quien dará el mantenimiento del software, la organización puede especificar anotaciones de diseño y estándares de codificación que un subcontratista debe seguir)
- Compatibilidad con productos anteriores.
- Las limitaciones impuestas por las reglas de negocio (las cuales están documentadas en otras partes).
- Las limitaciones del hardware, tales como los requisitos de tiempo, la memoria o restricciones del procesador, tamaño, peso, materiales, o el costo.
- Las convenciones de interfaz de usuario a seguir cuando se mejora un producto existente.
- Estándar de los formatos de datos de intercambio, como XML.

2.6. Documentación de usuario

Lista de los componentes de la documentación de usuario que se entregará junto con el software ejecutable. Estos podrían incluir manuales de usuario, ayuda en línea y tutoriales. Deberá identificar los formatos de entrega de la documentación requerida, normas, o las herramientas.

2.7. Suposiciones y dependencias

Una suposición es una declaración en la que se cree que es cierto en ausencia de la prueba o el conocimiento definitivo. Pueden surgir problemas si las suposiciones son incorrectas, no se comparten, o cambian, obviamente se traducirá en los riesgos del proyecto. Un lector de la ERS podría suponer que el producto se ajustará a una convención particular de interfaz de usuario, mientras que otro asume algo diferente. Un desarrollador puede asumir un cierto conjunto de funciones de forma personalizada por escrito para esta aplicación, pero el analista supone que van a ser reutilizados de un proyecto anterior, mientras que el director del proyecto espera conseguir una librería de funciones comerciales.



Identificar ciertas dependencias del proyecto de factores externos fuera de su control, tales como la fecha de lanzamiento de la próxima versión del sistema operativo o la emisión de un estándar de la industria. Si usted espera a integrarse en el sistema algunos de los componentes que otro proyecto está en desarrollo, que dependen de ese proyecto para suministrar los componentes que funcione correctamente en la fecha prevista. Si estas dependencias ya están documentados en otros lugares, como en el plan del proyecto, se refieren a aquellos otros documentos aquí.

3. Características del sistema

La plantilla que se indica en la figura 5.2 esta organizada por características del sistema, que es solamente una posible forma de organizar los requerimientos funcionales. Otras opciones incluyen organización mediante casos de uso, modos de operación, clases de usuario, estímulos, la respuesta, clases de objeto o jerarquía funcional. Son posibles también hacer combinaciones como es, los casos de uso dentro de las clases de usuario. No existe una opción única, mas bien se debe seleccionar un método de organización que facilite a los lectores entender las capacidades del producto. Por lo tanto las subsecciones que pueda tener este apartado dependerá de dicha organización. Vamos a describir esto mediante un ejemplo.

3.1. Características del sistema X.

Indique el nombre de la función en pocas palabras, por ejemplo: "3.1. Registrar trámite", luego para cada subsección numere: 3.1.1 , 3.1.2, ...

3.1.1. Descripción y prioridades

Realice una breve descripción de la función e indique si es de prioridad alta, media o baja. Las prioridades son dinámicas ya que cambian en el transcurso del proyecto. Si está utilizando una herramienta de gestión de requisitos, definir un atributo de exigencia de prioridad.

3.1.2. Secuencia de entrada/respuesta

Liste la secuencia de entrada (las acciones del usuario, las señales de los dispositivos externos, o de otros factores desencadenantes) y las respuestas del sistema que definen el comportamiento de esta característica. Estas entradas corresponden a los pasos de diálogo inicial de casos de uso o de los eventos del sistema externo.

3.1.3. Requerimientos funcionales

Detalle de los requisitos funcionales asociados con esta función. Estas son las capacidades de software que deben estar presentes para el usuario para llevar a cabo la función de los servicios o llevar a cabo un caso de uso. Describir cómo el producto debe responder anticipadamente a las condiciones de error y entradas no válidas y acciones. Establezca una única etiqueta de cada requisito funcional. Esta es la sección más larga e importante de la ERS. Deberán aplicarse los siguientes principios:

- El documento debería ser perfectamente legible por personas de muy distintas formaciones e intereses.
- Deberán referenciarse aquellos documentos relevantes que poseen alguna influencia sobre los requisitos.

- Todo requisito deberá ser unívocamente identificable mediante algún código o sistema de numeración adecuado.
- Lo ideal, aunque en la práctica no siempre realizable, es que los requisitos posean las características que se indican al inicio en el literal 5.2.2. (Correctos, no ambiguos, completos, consistentes, clasificables, verificables, modificables y trazables).

Se deberán definir tantas funciones como sean necesarias.

4. Requerimientos de interfaz externa

Richard Thayer (2002), indica que “los requisitos de interfaz externa especifican hardware, software, o los elementos de base de datos con la que un sistema o el componente de interfaz”, por tanto en esta sección se proporciona información para asegurar que el sistema se comunica correctamente con los componentes externos. Si las diferentes partes del producto tienen diferentes interfaces externas, es necesario incorporar una instancia de esta sección dentro del detalle de los requisitos para cada una de dichas partes.

Alcanzar un acuerdo sobre las interfaces del sistema externo e interno ha sido identificada como una buena práctica en la industria del software. Se realiza una descripción detallada de los datos y componentes de control de las interfaces en el diccionario de datos. Un sistema complejo con múltiples subcomponentes debe utilizar una especificación de interfaz por separado o especificación de la arquitectura del sistema. La documentación de la interfaz puede incorporar material de otros documentos de referencia. Por ejemplo, podría apuntar a una aplicación de programación diferente de interfaz (API) o un manual de dispositivos de hardware que muestra los códigos de error que el dispositivo puede enviar al software.

4.1. Interfaz de usuario

Describe las características lógicas de cada interfaz de usuario que el sistema necesita. Algunos ítems que se pueden incluir son:

- Referencias a normas GUI o lineamientos de estilos que se deben seguir.
- Normas para las fuentes, iconos, etiquetas, imágenes, esquemas de color, secuencias de campo de tabulación, los controles más utilizados, etc.
- Diseño de pantalla o las limitaciones de resolución.
- Botones estándar, funciones o enlaces de navegación que aparecen en cada pantalla, como un botón de ayuda.
- Teclas de acceso directo.
- Convenciones de mensaje de la pantalla.
- Normas de diseño para facilitar la localización de software.
- Alojamiento para los usuarios con discapacidad visual.



4.2. Interfaz de hardware

Se describe las características de cada interfaz entre el software y los componentes de hardware del sistema. Esta descripción podría incluir los tipos de dispositivos compatibles, las interacciones de datos y control entre el software y el hardware, y los protocolos de comunicación a utilizar.

4.3. Interfaz de software

Se describen las conexiones entre este producto y otros componentes de software (identificado por su nombre y la versión), incluyendo bases de datos, sistemas operativos, herramientas, bibliotecas y componentes integrados comerciales. Manifestar el propósito de los mensajes, datos y elementos de control que se intercambian entre los componentes de software. Describir los servicios que necesitan los componentes de software externos y la naturaleza de las comunicaciones entre componentes. Identificar los datos que serán compartidos a través de componentes de software. Si el mecanismo de intercambio de datos deben ser implementadas de una manera específica, como un área de datos global, especificar esto como una limitación.

4.4. Interfaz de comunicaciones

Establecer los requisitos para cualquier comunicación que las funciones del producto podría utilizar, incluyendo el correo electrónico, explorador Web, protocolos de red de comunicaciones, y los formularios electrónicos. Definir cualquier formato de mensaje. Especificar la seguridad de la comunicación o problemas de cifrado, las tasas de transferencia de datos, y los mecanismos de sincronización.

5. Requerimientos no funcionales

En esta sección se especifican los requerimientos no funcionales y otros requerimientos de interfaces externas.

5.1. Requisitos de desempeño

La especificación de los requisitos de desempeño se utilizan para diferentes operaciones del sistema. Es necesario explicar su relación para guiar a los desarrolladores en la toma de decisiones apropiadas para el diseño. Por ejemplo, la demanda en el uso de la base de datos con respecto a los tiempos de respuesta puede llevar a los diseñadores a establecer a la base de datos en múltiples localizaciones geográficas o para desnormalizar las tablas relacionales para que las consultas resulten mas rápidas. También se podría especificar requisitos de memoria y espacio en disco, carga de usuarios concurrentes, o el número máximo de datos almacenados en las tablas de base de datos.

5.2. Requerimientos de protección

Protección y seguridad son ejemplos de calidad de atributos, por lo que se analizan estos atributos en secciones diferentes de la plantilla de la ERS, ya que son importantes en absoluto, por lo general son críticos. En este apartado se especifica los requisitos que tienen que ver con la posible pérdida, daño o perjuicio que pudieran derivarse del uso del producto. Deberá definir las salvaguardias o acciones que se deben tomar, así como las acciones potencialmente peligrosas que deben ser prevenidos. Identificar las certificaciones de seguridad, políticas o regulaciones a las que el producto debe ajustarse. Algunos ejemplos de los requisitos de seguridad son:

- **PR-1** El sistema pondrá fin a cualquier operación dentro de 1 segundo, si la presión media del tanque supera el 95 por ciento de la presión máxima especificada.
- **PR-2** El escudo de radiación de haz estará abierto solo a través del control informático permanente. El escudo bajará automáticamente a su lugar si el control de equipo se pierde por cualquier razón.

5.3. Requerimientos de seguridad

Se especificará los requisitos con respecto a cuestiones de seguridad, integridad o la vida privada que afectan el acceso al producto, el uso del producto, y la protección de datos que utiliza el producto o se crea. Los requisitos de seguridad que normalmente se originan en las reglas de negocio, que identifican las políticas de seguridad o privacidad, o regulaciones a las que el producto debe ajustarse. Como alternativa, puede cumplir estos requisitos a través del atributo de calidad llamado integridad. Por ejemplo:

- **SE-1** Cada usuario debe cambiar su contraseña de inicio de sesión asignado inicialmente, inmediatamente después de su ingreso exitoso la primera vez. La contraseña inicial no puede ser reutilizada.
- **SE-2** Al abrir una puerta mediante una tarjeta magnética, esta se mantenga abierta durante un lapso de 8 segundos.

5.4. Atributos de calidad del software

Indicar la existencia de las características de calidad del producto adicionales que serán importantes para los clientes o desarrolladores. Estas características deben ser específicas, cuantitativas y verificables. Indicar las prioridades relativas de los distintos atributos, como la facilidad de uso, la facilidad de aprendizaje, o la portabilidad sobre la eficiencia.

6. Otros requerimientos

Definir otros requisitos que no están en la ERS. Por ejemplo se pueden incluir requisitos de internacionalización (moneda, formato de fecha, el idioma, las normas internacionales, y los aspectos culturales y políticos) y requisitos legales. También puede añadir secciones en las operaciones, administración y mantenimiento para cubrir las necesidades de instalación del producto, configuración, puesta en marcha y la tolerancia de cierre, la recuperación y responsabilidad, y el registro y seguimiento de operaciones. Agregue todos los nuevos tramos de la plantilla que son pertinentes a su proyecto. Omite esta sección si todas sus necesidades están en otras secciones.

En el anexo 4, se han desarrollado partes del caso de estudio de la especificación de requerimientos de software basados en la plantilla del estándar IEEE Std 830-1998.



En el sitio: <http://users.dsic.upv.es/asignaturas/facultad/lsi/ejemplorup/Requisitos.html> existe un ejemplo utilizando la metodología RUP basada en casos de uso. Revise detenidamente cada uno de los artefactos y relacione con respecto a la especificación que hemos desarrollado en esta parte.



De igual forma en el sitio <http://www.volere.co.uk/template.htm> se indica las partes que contiene la plantilla para especificar requerimientos utilizando Volere.



Actividad recomendada

Busque las plantillas que utiliza RUP y Volere para especificar requerimientos de software, y establezca semejanzas y diferencias.

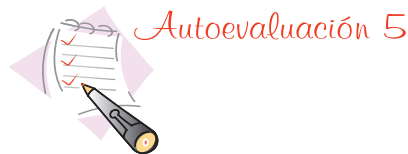
7. Chequear la calidad del documento ERS

- Llevar a cabo revisiones de la SRS para detectar problemas de calidad en los requisitos y en el propio documento. En la unidad siguiente se establecen los criterios de calidad.



Como puede ver los pasos que se necesitan para realizar la ERS, requiere del uso de los modelos de análisis ya sea para documentar las necesidades o para definir los requerimientos. Por lo tanto el uso adecuado de las técnicas permitirán que se logren definir los requisitos.

Hemos concluido el estudio de esta unidad, por lo que nos conviene comprobar cuanto ha asimilado los temas tratados para lo cual es necesario desarrollar las autoevaluaciones de conocimiento.



Realice las actividades que se indican en cada uno de los literales.

- a) Lea detenidamente la afirmación y escriba una V si considera que es verdadera o una F si no lo es, en el paréntesis respectivo.

1	El ERS obliga a que los grupos interesados dentro de la organización consideren todos los requerimientos.	()
2	El siguiente enunciado: "Cada requisito tiene una sola interpretación" corresponde al atributo: Correcto	()
3	La característica de consistencia en un ERS, consiste en que los requisitos no pueden ser contradictorios.	()
4	En el alcance del ERS, se pueden obviar requisitos no funcionales.	()
5	¿Al momento de realizar el proceso de aseguramiento de calidad, el documento de especificación de requisitos será obligatorio?	()

6	Si un cliente necesita que el sistema realice una transacción en un tiempo máximo determinado por él. Este requisito es considerado como requisito funcional.	()
7	Si al momento de iniciar el proyecto se habla de la construcción del mismo en un lenguaje de programación determinado por las licencias que posee el cliente, se podrá hablar de un requisito de portabilidad.	()
8	Si el sistema no cuenta con otras interfaces hacia los otros sistemas se podría obviar la sección de interfaz de hardware.	()
9	Que el sistema este accesible a los usuarios en un 95% del tiempo es un requerimiento no funcional de fiabilidad.	()
10	Al identificar los logos a utilizar en el sistema, se podría catalogar como un requerimiento de interfaz de usuario.	()

b) Indique los requerimientos:

- a. Requerimientos funcionales sin restricción
- b. Requerimientos funcionales con restricción
- c. Requerimientos funcionales con restricción, resultado y calificador
- c) De acuerdo a la plantilla de la tabla 5.2, establezca un esquema de las partes que desarrollaría como parte del documento de especificación de requerimientos del caso de desarrollo.
- d) Reúna todos los documentos desarrollados como parte del análisis de requerimientos aplicando las diferentes técnicas de obtención de requerimientos.
- e) Qué criterio le merece las plantillas del RUP y Volere para la especificación de requerimientos de software.





UNIDAD 6: Validación de Requerimientos

Estimado estudiante, revisemos los aspectos principales de la validación de requisitos, que es la fase en la cual se determina si un producto satisface las necesidades del usuario y se ajusta a los requisitos; además la validación de requisitos garantiza que los requisitos son necesarios y están suficientemente especificados para satisfacer las necesidades del usuario antes de que el diseño y desarrollo del software comience. Las actividades de validación de los requisitos son detectar y corregir cualquier requisito innecesario e incorrecto.



Se entiende como validación de los requisitos al proceso de comprobación de que estos requisitos fueron especificados de acuerdo a las necesidades de los clientes. Esta etapa es de suma importancia si no se quiere correr el riesgo de realizar una mala implementación debido a que no se hizo una buena especificación de requisitos.

Todo esto se verá reflejado tanto en la calidad del software como en el costo puesto que se deberá corregir el sistema. El proceso de validación de requisitos comprende actividades que generalmente se realizan una vez obtenida una primera versión de la documentación de especificación de requisitos.

De esta manera es de suma importancia que al terminar la etapa de modelado de requerimientos se realice el proceso de comprobación realizando una comparación de las conversaciones iniciales que se realizaron a las especificaciones que se obtuvieron.

6.1 Validación de requerimientos

Para validar los requisitos se debe realizar lo siguiente:

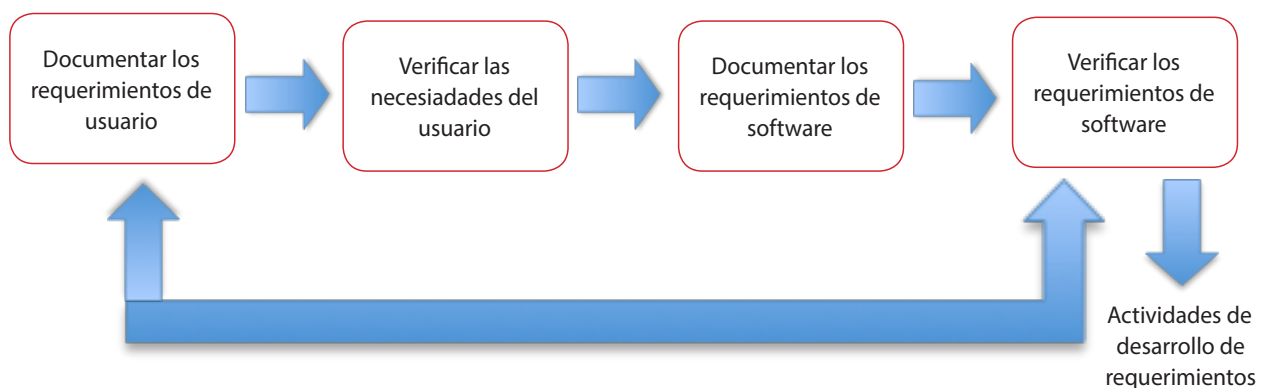


Figura 6.1: Actividades de validación de requerimientos (Gottesdiener E. , 2005)

A continuación se detallan cada una de estas actividades:

a) Seleccionar e integrar la técnica de validación de requisitos:

Que comprende la identificación de las técnicas de validación más eficaces; además se debe elegir una combinación de técnicas de validación y diferentes porciones de los requisitos; así mismo se debe validar los requisitos de alto riesgo a principios del proceso, para reducir los riesgos generales del proyecto y finalmente se deben integrar las actividades de validación a través del desarrollo de requisitos.

b) Asegurar la participación adecuada de usuario:

En esta etapa se debe verificar que los requerimientos del usuario describan la forma en que interactúan con los usuarios. La participación activa de los usuarios es crucial ya que la validación implica el chequeo de la conformidad de las necesidades del usuario. Se recomienda que los interesados comprueben que los requisitos estén completos, consistentes y sean de alta calidad, para lo cual es necesario la revisión de la documentación. Además se debe asegurar de obtener los requerimientos funcionales, los del negocio y los del usuario.

c) Validar los requisitos:

Implica comprobar que un subconjunto de los requisitos estén bien definidos, esto se lo debe realizar a principios del desarrollo de los requisitos y no cuando se tenga el detalle de los modelos de análisis.

d) Revisar la documentación de requerimientos:

Se debe revisar la documentación inmediatamente basados en la retroalimentación de la etapa anterior (validación de requisitos); se debe realizar el análisis de los requisitos para entender cómo los cambios afectan a los planes del proyecto; priorizar nuevamente los requisitos luego de las actividades de validación y finalmente hay que repetir el ciclo a medida que avanza el proceso de desarrollo de requerimientos.

Como objetivo primordial de la validación de requisitos se puede mencionar que se intenta descubrir los problemas en el documento de especificación de requisitos antes de llegar a su implementación.



Roger Pressman en su libro Ingeniería del software, presenta un conjunto de preguntas para validar los requerimientos, considera que estas ayudan a evaluar el desarrollo de esta fase, a continuación se listan estas preguntas:

- “¿Es coherente cada requerimiento con los objetivos generales del sistema o producto?.
- Se han especificado todos los requerimientos en el nivel apropiado de abstracción? Es decir ¿algunos de ellos tiene un nivel de detalle técnico que resulta inapropiado en esta etapa?.
- El requerimiento ¿Es realmente necesario o representa una característica agregada que tal vez no sea esencial para el objetivo del sistema?.
- ¿Cada requerimiento está acotado y no es ambiguo?.



- ¿Tiene atribución cada requerimiento?, es decir, ¿hay una fuente (por lo general individual y específica) clara para cada requerimiento?.
- ¿Hay requerimientos en conflicto con otros?
- ¿Cada requerimiento es asequible en el ambiente técnico que albergará el sistema o producto?.
- Una vez implementado cada requerimiento ¿Puede someterse a prueba?.
- El modelo de requerimientos ¿Refleja de manera apropiada la información, la función y el comportamiento del sistema que se va a construir?
- ¿Se ha particionado el modelo de requerimientos en forma que exponga información cada vez más detallada sobre el sistema?
- ¿Se ha usado el patrón de requerimientos para simplificar el modelo de estos? ¿se han validado todos los patrones de manera apropiada? ¿son conscientes todos los patrones con los requerimientos del cliente?"¹²



Al trabajar con estas preguntas se puede de alguna manera ir validando los requisitos, para evaluar el cumplimiento de algunas especificaciones además de comprobar una buena especificación de requisitos.



Actividad recomendada

Realice la validación de los requerimientos indicados en el anexo 4, utilizando las preguntas de la sección 6.1.

6.2. Técnicas de validación



Recordemos, que a mediados de los años setenta se realizó un estudio en donde se estima que el coste de corregir un error en un sistema software aumenta a medida que se desarrolla el sistema. El costo de corregir un error en las etapas finales es del 10 a 100 veces por encima que el costo de corregirlo en las etapas iniciales, estas estimaciones siguen teniendo plena vigencia, por este motivo trataremos estos temas para que cuando diseñemos software no cometamos los mismos errores.

Sin embargo, los costos de corrección de errores en la fase de prueba son mucho mayores que si los errores se corrigen en fases tempranas como la fase de requisitos o de análisis. De aquí que es necesario adelantar el proceso de prueba en las primeras etapas de desarrollo claro está donde sea posible.

La revisión de requisitos se realiza con un grupo de personas que lee, analiza y discute el documento de especificación de requisitos. Entonces es importante seleccionar a las personas que han de trabajar en el proceso de validación.

Se puede realizar una comparación entre el análisis y la validación, tomando en cuenta que en el análisis se cuenta como entrada requisitos incompletos mientras que en la validación un conjunto comprobado de requisitos. Además en el análisis se verifica si los requisitos satisfacen las necesidades del cliente o si

12 Pressman Roger S. Ingeniería del Software: un enfoque práctico, Séptima edición. McGraw Hill.

se han levantado los requisitos correctos mientras que en la validación se verifica si están bien descritos los requisitos o si el documento de requisitos describe el sistema, como se puede observar la etapa de validación permite tener una definición precisa de los requisitos.

En la tabla 6.1, se indican las técnicas que se utilizan de acuerdo a lo que se está verificando.

Tabla 6.1: Técnicas de Validación de Requisitos

Cuándo Ud. necesite	Se debe Realizar
Revisión de pares	Revisar requerimientos
	Documentar requerimientos
Crear pruebas de validación	Pruebas de aceptación de usuario
Pruebas sobre modelos de requerimientos	Valide los modelos
Mostrar partes del sistema	Prototipos Operacionales

Ahora, revisemos a detalle cada una de estas técnicas:

6.2.1. Revisión de pares

La revisión por pares es la formación de un grupo de partes interesadas para evaluar la documentación de los requisitos o de los modelos con el fin de encontrar errores y mejorar la calidad de los mismos.

Un tipo de revisión por pares son las inspecciones; las cuales constituyen el tipo más formal de revisión por pares; las inspecciones se pueden incorporar en las fases de: Planificación, Reunión de Información, Preparación de inspección individual, Reunión de Inspección, Seguimiento, Análisis causal (para determinar la causa de los defectos y decidir la forma de prevenirlos en un trabajo futuro), inspecciones sobre roles formales y herramientas de inspección.



Analicemos, ¿por qué es importante realiza la revisión de pares?; es necesario para detectar errores e inconsistencias en los requisitos, para asegurar que los requisitos representan adecuadamente las necesidades del usuario, para reducir los costos asociados con la corrección de defectos de ejecución que se originan en las necesidades, y para aumentar la calidad del software.



Para realizar esta revisión se debe:

- Proporcionar un entorno de aprendizaje en el que los interesados puedan entender mejor los requisitos de los requisitos y dominio del negocio.
- Obligar a los analistas a centrar los esfuerzos de validación en las partes en que los requisitos tengan mayor riesgo y ambigüedad.
- Definir las expectativas de calidad para las necesidades a través de la creación de listas de comprobación o revisiones de inspección.
- Identificar los requisitos de las oportunidades de mejora de procesos.



A lo anterior hay que decidir qué partes de los requisitos de un producto se deben revisar y el tipo de enfoque que se dará a dichos tests; identificar los stakeholders que intervendrán en la revisión, planificar la revisión a través de la organización de reuniones para las revisiones y finalmente se deben preparar las revisiones de las reuniones usando listas de chequeo.

6.2.2. Pruebas de aceptación de usuario

Las pruebas de aceptación de los usuarios se los denomina también como criterios de aceptación, pruebas de aceptación, pruebas de usuario final o pruebas funcionales; las mismas que constituyen casos específicos de prueba que los usuarios utilizan para decidir si aceptan la entrega de un sistema, cada prueba de aceptación es una prueba de caja negra (es decir, una prueba escrita respecto a la aplicación de software) que representa las entradas del sistema y los resultados esperados para los que el sistema está diseñado para ejecutarse.

Estas pruebas de aceptación del usuario involucra analizar los resultados de las pruebas de revisión, verificar la exactitud de las pruebas de aceptación, decidir qué pruebas de aprobación se deben realizar y cuáles no, y decidir que los que no pasaron las pruebas tienen la prioridad más alta para corrección; luego de las pruebas se llega a las conclusiones en donde los usuarios aceptan o rechazan el sistema.

Estas pruebas son importantes de ejecutar porque permiten definir la aceptación del sistema; para lo cual se deben realizar: guías para los usuarios para describir de manera más explícita la forma en que espera que el software trabaje; asegurar la existencia de pruebas para demostrar que el sistema se ajusta a las expectativas del cliente; proporcionar una representación concreta de los datos necesarios para que los usuarios acepten el sistema final; proporcionar una base para el desarrollo de manuales de usuario, y finalmente proporcionar pruebas útiles para la validación del modelo. ¿Pero cómo alcanzamos esto?.

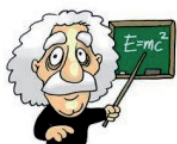
A través de la definición de criterios de aceptación del sistema, de la aceptación de las pruebas de casos, de la determinación de métodos de pruebas de aceptación; los métodos que se pueden utilizar son:

- Manuales
- Herramientas de pruebas con interfaz gráfica de usuario
- Codificación y pruebas.
- Scripting.
- Hojas de cálculo
- Plantilla

En cuanto a la parte de validar el análisis de los modelos utilizando pruebas de aceptación, se puede mencionar que estas pruebas tienen diferentes niveles de aceptación, para lo cual se debe considerar el nivel de severidad de la prueba a la hora de priorizar las pruebas para su corrección. Los niveles de severidad se encuentran detallados en la tabla 6.2.

Tabla 6.2 Niveles de Severidad

NIVEL DE SEVERIDAD	DEFINICIÓN
1	Crítico: Es imposible continuar con las pruebas o aceptar el sistema a causa de este error.
2	Alto: Las pruebas pueden continuar, pero el sistema no se puede implementar con este problema.
3	Mediano: Las pruebas pueden continuar y el sistema es probable que se siga desarrollando con algunas salidas de funcionalidad del negocio acordados.
4	Mínimo: Las pruebas y el despliegue pueden progresar. El problema debería ser corregido, pero no afectará la funcionalidad de negocio.
5	Visual: Los errores como colores, fuentes, y las demostraciones de interfaz que son menos que deseables pueden ser corregidos en algún futuro tiempo.



Ejemplo:

Para una mejor comprensión del tema, revise el siguiente ejemplo:

Datos de Entrada		
Resumen de servicios	Fecha	Apellidos del contratista
	15 de Marzo	Espinoza
	20 de Abril	Armijos
	30 de Agosto	Suárez
Resultados esperados de las pruebas (Buscar trabajos programados (por fecha, servicio, o el contratista apellidos)		
Consulta	Registros devueltos	Comentarios
Marzo y Agosto	6	Buscar en la fecha.
Armijos	1	Buscar en el nombre del contratista



Actividad recomendada

A partir de la actividad recomendada en la sección 6.1, determine los niveles de aceptación que tienen cada uno de los requerimientos además aplique la plantilla que se indica en el ejemplo anterior.

6.2.3. Modelos de validación

A esta técnica también se la denomina como: resumen de pruebas, pruebas conceptuales o análisis lógico. La validación del modelo utiliza simulaciones de pruebas en lugar de casos de prueba real para pasar por múltiples modelos de análisis que permitan descubrir la falta de información y corregir errores de documentación.



¿Qué se logra con esta validación?

- Permitir a usuarios y miembros de equipo simular la operación de sistema y probar el código, en lugar de casos de pruebas reales.
- Demuestra si los modelos son compatibles entre ellos.
- Comprobar que los requerimientos cubren situaciones típicas del usuario.
- Descubre errores, inconsistencias, o requerimientos que fallan en el modelo de análisis y en la documentación de requisitos.
- Permite a una forma de pruebas cuando un prototipo es no disponible o factible

¡Y! ¿Entonces?

¿Cómo se hace la validación?

Para realizar la validación se deben seguir los siguientes pasos:

- Identificar y crear casos de pruebas.
- Seleccionar los modelos de análisis para validar.
- Rastrear los modelos a través de los casos de prueba de una manera paso a paso.
- Corregir los modelos de requisitos

6.2.4. Prototipos operacionales

En este apartado se abordará el tema de Prototipos operacionales, que también reciben el nombre de: Demostración, Prueba de Concepto, Prototipo estructural o prototipo vertical.

El prototipado permite capturar información y validar la solución o producto que se ha desarrollado mediante prototipos. Es una representación gráfica clara de lo que el usuario quiere o va a recibir. A través del prototipado se logra reducir costos en desarrollar o reconstruir un sistema o proyecto. Son muy necesarios para comunicarse con el usuario o cliente, permitiendo determinar el nivel de usabilidad e interacción del mismo, e identificar o redefinir ciertos aspectos de la solución.

Los prototipos sirven para demostrar cómo una parte del software funcionará una vez que esté en desarrollo, para demostrar si los requisitos satisfacen a los clientes, y para validar los requisitos de interfaz externa. Además permite evaluar la viabilidad de los atributos de calidad tales como el rendimiento, facilidad de uso, o la seguridad; detectar las funcionalidades innecesarias, los pasos que faltan, o interfaces de usuario muy complejo que podría inhibir la satisfacción de las necesidades del usuario, además permite a los desarrolladores obtener experiencia en el diseño y desarrollo temprano en el proyecto y reduce el riesgo global del proyecto mediante la revelación de los requisitos faltantes, erróneos.



Según el estándar ISO 13407, el prototipo se define como: “una representación de todo o parte de un producto o sistema que, aunque limitado de algún modo, puede utilizarse con fines de evaluación”¹⁶.

16 FERRÉ, Xavier. Marco de Integración de la Usabilidad en el Proceso de desarrollo de software. Facultad de Informática. Universidad Politécnica de Madrid. Resultados de Tesis Doctoral. *Disponible en:* <http://is.fi.upm.es/xavier/usabilityframework/tecnicas/prototipos.php>

Existen tres tipos de prototipos:

- **Prototipo de la interfaz del usuario:** Es un modelo evolutivo de cómo va quedando la interfaz del usuario en base a los requisitos o necesidades planteados. Puede ser desarrollado en papel, ejecutable del sistema o proyecto en desarrollo o software específicos que se los encuentra libremente en el mercado.
- **Prototipos de rendimiento:** Estos no incluyen la interfaz del usuario. Va orientado a los desarrolladores, con el fin de comprobar la funcionalidad de los componentes de la solución que se está desarrollando. No se aplica a la ingeniería de requisitos.
- **Prototipo funcional:** Es una versión limitada de la solución desarrollada. No se aplica a la ingeniería de requisitos.

El proceso para llevar a cabo el prototipado es:

- Determinar que requisitos se van a validar mediante un prototipo operativo.
- Desarrollar el prototipo.
- Evaluar el prototipo.

Las técnicas que se utilizan son:

- Revisión por pares.
- Pruebas de aceptación de usuario.
- Validación de modelos.
- Prototipos operacionales

Los prototipos pueden llegar a ser muy útiles si se los sabe aplicar correctamente y en el momento oportuno. En la captura y validación de requisitos se puede llegar a determinar realmente lo que el usuario desea y lo que quiere ver. Deben ser fiables y describir exactamente los componentes o funcionalidades tomados en cuenta en el prototipado.

El uso de prototipos puede tener ventajas, así como también desventajas al aplicarlo en la ingeniería de requisitos. Dentro de las principales ventajas y desventajas de esta técnica están:

Tabla 6.3. Ventajas y Desventajas del Prototipado

VENTAJAS	DESVENTAJAS
• Viabilidad y utilidad del sistema o proyecto. • Permite desarrollar interfaces de usuario en base a los requisitos. • Permite encontrar requisitos incorrectos o inconsistentes.	• Elevados costos en capacitación de prototipado. • Costos en el desarrollo de prototipos. • Se alarga el tiempo de entrega de la solución. • Al no ser reales ni considerar el rendimiento de la solución, los clientes o stakeholders pueden tener una mala impresión de lo que realmente se desea desarrollar.

Las pruebas del sistema en etapas tempranas del desarrollo permiten mejorar la calidad de los requisitos detectando errores, omisiones, inconsistencias y sobre especificaciones en los requisitos funcionales cuando aún es fácil y económico corregirlos.



Para capturar los requisitos funcionales se utiliza mayoritariamente casos de uso. Los casos de uso proporcionan un medio de expresar la interacción entre un sistema y su entorno. Esto permite estructurar los requisitos de acuerdo con los objetivos de los usuarios.

Las pruebas de requisitos nos permiten diseñar pruebas sobre cada requisito individual, estas pruebas pueden estar descritas en el mismo documento de especificación de requisitos.



Por ejemplo en: <http://hex.nosololinux.com/modulos/descargas/Anexo%202.pdf> en la página 29, encontramos:

6. Pruebas de aceptación

Aunque posiblemente en un sistema con estas características no procesa establecer unas reglas de aceptación en forma de pruebas, se considerará aceptado el sistema si cumple con las condiciones siguientes:

1. Es capaz de jugar contra un humano en un tablero 5x5 con un tiempo de decisión menor a 1 minuto en la jugada más complicada (exceptuando la primera, que puede obtenerse de tablas), ejecutándose en un modelo concreto de equipo.
2. Cumple los requisitos establecidos en la sección anterior, pudiéndose ejecutar cada caso de uso documentando sobre el sistema.
3. El sistema se ejecuta satisfactoriamente sobre Mac OS, Linux y Windows.

Figura 6. 2. Ejemplo de pruebas de aceptación



Actividad recomendada

Indique como serían las pruebas de aceptación utilizando los requerimientos del anexo 4.

6.3. Otros modelos de validación

Además existen otros modelos de validación que están contemplados en la ingeniería de requisitos, los cuales van a ser analizados para poder determinar el proceso de validación en cada uno de ellos.

6.3.1. Modelo de POHL

Es un modelo iterativo en el que entran en juego cuatro actividades: Captura, Negociación, Especificación-Documentación y Validación-Verificación. El orden en que se ejecutan y realizan estas actividades puede ser cualquiera, pero siempre es aconsejable seguir una secuencia.¹⁴

¹⁴ Información recopilada de:

- GARCÍA, Jesús. Introducción a la Ingeniería de requisitos. Ingeniería de Software. Escuela Superior Politécnica de Albacete. España. Disponible en: www.info-ab.uclm.es/ asignaturas/42530/pdf/M1tema3.pdf
- BERNARDEZ, Beatriz. (2001). Modelo aplicado a la calidad de la ingeniería de requisitos. Universidad de Sevilla. Documento digital.

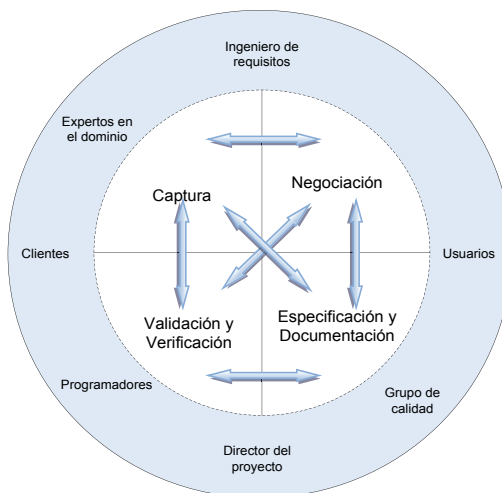


Figura 6.3. Modelo de Pohl de IR. 14

6.3.2. Modelo en espiral

Este modelo plantea tres ejes: borrador de requisitos, conflictos de requisitos, documento de requisitos; que se cumplen o complementan en base a tres actividades: captura, análisis-validación y negociación de requisitos.

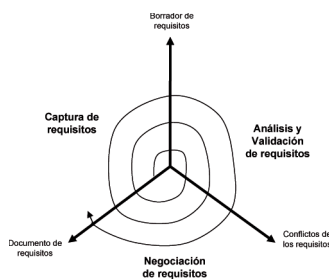


Figura 6.4. Modelo Espiral de IR14

6.3.3. Modelo SWEEBOK (Software Engineering Body of Knowledge Model)

Nació como un proyecto desarrollado por la IEEE “para producir un cuerpo de conocimiento sobre Ingeniería de Software que sienta las bases sobre dicha ingeniería como una profesión”¹⁵.

Dentro de este proyecto existen diez áreas de conocimiento, diseño del software, construcción del software, testeo del software, mantenimiento, entre otras. Una de ellas son los Requisitos del Software, es aquí donde nace el modelo SWEEBOK como una visión consistente y consensuada de la Ingeniería del Software.

¹⁵ GARCÍA, Jesús. Introducción a la ingeniería de requisitos. Ingeniería de Software. Escuela Superior Politécnica de Albacete. España. Disponible en: www.info-ab.uclm.es/asignaturas/42530/pdf/M1tema3.pdf

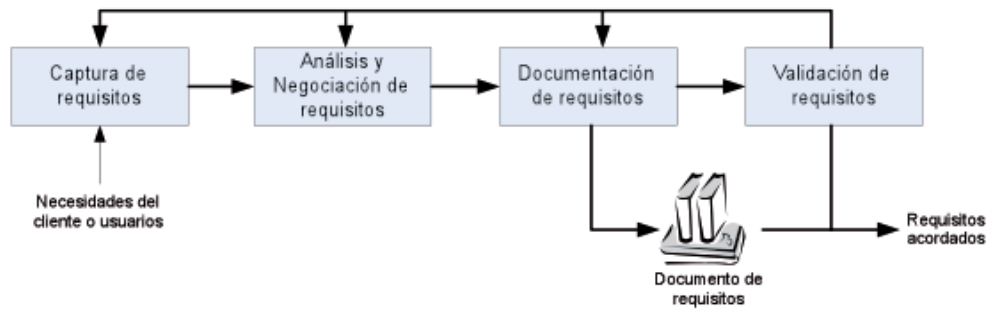


Figura 6.5 Modelo SWEBOK de IR (IEEE Computer Society, 2004)

El modelo consiste en cuatro actividades: captura, análisis y negociación, documentación y, validación de los requisitos.

6.3.4. Modelo de VOLERE

Este quizá es el modelo que ha servido de base para la construcción de los demás modelos, posee un conjunto de actividades que se relacionan e interactúan de manera iterativa, lo que hace que el proceso de Ingeniería de requisitos se vuelva eficaz y eficiente. Este es un modelo de procesos genérico, útil para definir, analizar, especificar y validar los requisitos.

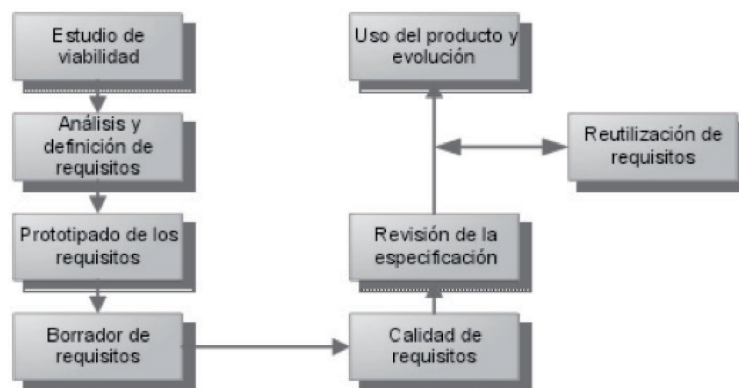


Figura 6.6 Modelo de Volere¹⁴

Dentro de las principales actividades de este modelo están: Estudio de viabilidad, Análisis y definición de requisitos, Prototipado de los requisitos, borrador de requisitos, calidad de los requisitos, revisión de la especificación, diseño y construcción, reutilización de requisitos, uso del producto y evolución.

¹⁴ Información tomada de: ROBERTSON, Z., ROBERTSON, J. (2006). Mastering the Requirements Process. Editorial Addison-Wesley. Sexta Edición.

6.3.5. Comparación entre los modelos

Tabla 6.4. Comparación entre Modelos de Proceso de IR

ACTIVIDADES	MODELOS					
	GENÉRICO	POHL	ESPIRAL	SWEEBOK	REAIMS	VOLERE
Captura	x	x	x	X	x	x
Análisis	x	x	x	X	x	x
Borrador de requisitos			x		x	x
Negociación		x	x	X	x	x
Descripción de requisitos					x	x
Modelado de requisitos					x	x
Especificación de requisitos		x	x	X		x
Validación	x	x	x	X	x	x
Gestión					x	x

Todos los modelos son similares en cuanto a las actividades que realizan, pero tienen sus diferencias, las que radican en la aplicación de un número determinado de actividades y en la forma de hacerlo, sea esto iterativo o secuencial.

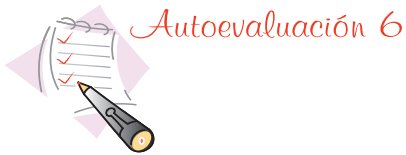
Como se pudo observar el proceso de validación está presente en todos los modelos de procesos de requisitos. Esto ayuda a determinar la importancia de aplicar esta actividad.

Para finalizar esta unidad le solicitamos que realice la siguiente actividad.



Actividad recomendada

El estándar IEEE 1028 proporciona orientación sobre la revisión y auditoría de software; y el estándar IEEE 1465 norma el tratamiento de los requisitos en la calidad del desarrollo de software; por lo cual se solicita que revise estos estándares e indique cuáles son las normativas que se aplican a la validación de requisitos. Esta actividad debe ser entregada a través del Entorno Virtual de Aprendizaje (EVA).



Autoevaluación 6



Realice las actividades que se indican en cada uno de los literales

6.1 Responda verdadero o falso según el enunciado que se presenta:

1.	Alexistir varias versiones de la especificación de requisitos las pruebas de validación de los mismos se deben realizar a partir de la versión final de los mismos.	
2.	Las especificaciones que se obtienen deben ser validadas para determinar que lo que está en el documento es lo que pide el cliente.	
3.	Si no se cumple (su validación no fue total) un requisito y este no afecta a la implementación del sistema, podrá dejarse sin efecto el mismo.	
4.	Podrá existir algún requerimiento que por su naturaleza no puede ser implementado.	
5.	Diseñar o construir un prototipo del sistema que se desea es una forma de probar los requisitos.	
6.	Se puede utilizar la especificación de casos de uso para probar los requisitos del Sistema.	
7.	La validación de requisitos está presente en todos los modelos de procesos de IR.	
8.	Todos los procesos de los modelos de IR se basan en igual número de actividades.	
9.	Puede el equipo de aseguramiento de calidad probar los requisitos.	
10.	Los prototipos orientados a los desarrolladores son los llamados prototipos de rendimiento.	

6.2 Aplique el estándar IEEE 1028 y del estándar IEEE 1465 para validar los requisitos que se indican en el anexo 4.



UNIDAD 7: Gestión de Requerimientos



esta unidad.

Estimado estudiante, en esta unidad, se va a analizar la gestión de requisitos, las actividades que implica y las técnicas que se utiliza en la gestión de los requisitos. El manejo inadecuado de los requisitos cambiantes son a menudo una fuente de retrasos en los proyectos, es por esta razón que es muy importante que revise minuciosamente

7.1 Gestión de Requisitos (Requirements Management, RM)

Por lo general los requisitos de software son volátiles, ya que pueden cambiar debido a varios factores tales como errores u omisiones en la fase de elicitación, la naturaleza cambiante, la comprensión compleja de sistemas, las tecnologías emergentes, los requisitos cambiantes del negocio o las exigencias reglamentarias, y las presiones competitivas.



La gestión de requisitos es el proceso de seguimiento de la situación de los requisitos y controlar los cambios de los requisitos de la línea de base. La gestión de requisitos es una actividad del ciclo de vida, comenzando por el desarrollo de los requisitos y continuando en el desarrollo del software.

Para gestionar los requisitos, es necesario establecer procedimientos que permiten al equipo de forma rápida entender el impacto de los cambios, decidir cómo hacer frente a las nuevas necesidades, y renegociar los acuerdos sobre los requisitos.

Como se ha explicado anteriormente en proyectos de desarrollo de software considerados de nivel medio-alto es imprescindible realizar una gestión de los requisitos, puesto que con la correcta administración de los mismos no se permiten errores en las etapas posteriores en el desarrollo de los mismos.

Esta etapa consiste, primordialmente, en gestionar los cambios a los requisitos, puesto que este es uno de los atributos de calidad que deben cumplirse y es de suma importancia, puesto que asegura la consistencia entre los requisitos y el sistema que se está desarrollando, se ha de considerar las cantidades de tiempo y esfuerzo que se utilizará puesto que abarca todo el ciclo de vida del software.

Entonces la gestión de requisitos implica la definición de procedimientos de gestión de cambios: definen los pasos y los análisis que se realizarán antes de aceptar los cambios propuestos o si es necesario cambiar los atributos de los requisitos afectados, como se realizará el proceso, cuáles serán los responsables y de esta manera contar con una trazabilidad: hacia atrás, hacia delante y entre requisitos, con la aplicación de métodos de versionamiento de los documentos de requisitos.

7.2 Herramientas y técnicas que se utilizan en la gestión de requisitos

En la actualidad existen herramientas que facilitan las tareas de la escritura, trazabilidad y gestión de cambios en los requisitos, además de mantener una adecuada administración de estos en bases de datos u otros repositorios.



Estas herramientas aunque son estándares, nos permiten también modificar atributos de los requisitos como para poder adaptar los mismos a los cambios de cada organización. Dentro de las herramientas utilizadas en la ingeniería de requisitos están:

- Rational Requisite Pro
- IRqA (Analizador integral de requisitos)
- LEAP SE
- Visual Requisite

La elección de las técnicas y herramientas de software, dependen de la metodología de desarrollo de software que se vaya a aplicar y del tipo de proyecto a construir, es así que, en un proyecto de gestión académica universitario en el que se esté aplicando RUP como metodología de desarrollo, es posible que el ingeniero de requisitos aplique Rational Requisite Pro, ya que es una herramienta propia de esta metodología y, le permite manejar y administrar de forma adecuada la gran cantidad de requisitos que se generan del proyecto.

Dentro de las principales características de las herramientas de gestión de requisitos están:

- **Captura de requisitos:** Comprende la obtención de los requisitos o necesidades planteados por el cliente.
- **Análisis:** Se realiza la identificación de los requisitos y la clasificación por tipo: funcional, no funcional, de software, de usuario, de interfaz, de negocio.
- **Negociación:** Implica contrastar y dar opiniones respecto a los requisitos que se han obtenido, con el fin de determinar las funciones y las restricciones que tendrá el sistema o proyecto.
- **Especificación:** Consiste en la definición, descripción detallada y completa de cada uno de los requisitos que serán sometidos a validación.
- **Validación y verificación:** Comprobar que los requisitos obtenidos cumplen con los objetivos del cliente y si fueron construidos en base a estándares o criterios desarrollados por el equipo de desarrollo.
- **Gestión de requisitos:** Se realiza la comprensión y control de los cambios de cada uno de los requisitos.
- **Trazabilidad:** Consiste en determinar el ciclo de vida de los requisitos, además de que cada uno de estos posee su propia identificación, diferenciándolos de los demás.
- **Facilidad de uso:** Quiere decir facilidad en el manejo de herramientas para la administración de requisitos por parte de aquellas personas que realicen el mantenimiento y explotación de los mismos, además de su fácil modificación tanto en su estructura como en su estilo o tipo.
- **Redundancia:** No se debe permitir redundancia, cada uno de los requisitos se debe utilizar en un solo lugar y ser identificados de manera diferente.
- **Generación de modelos:** Algunas de las herramientas utilizadas en la ingeniería de requisitos deben permitir generar modelos como: casos de uso, conceptuales, de estado entre otros; de tal manera que ayude a validar y verificar cada uno de los requisitos y a la identificación de potenciales errores.

- **Comunicación de requisitos:** Lo más importante en el desarrollo de todo proyecto es la comunicación que exista en el equipo de desarrollo, toda la información debe ser transmitida de manera oportuna, en el caso de los requisitos, a través de las herramientas se puede lograr esto con la generación de usuarios, estos tendrán acceso a cada uno de los requisitos con el fin de analizarlos, refinarlos de ser necesario y sobre todo obtener coherencia en los requisitos.
- **Generar informes:** Es importante para el respectivo análisis de los requisitos y de un proyecto.



Actividad recomendada

Realice un cuadro comparativo de las siguientes herramientas: Rational Requisite Pro, IRqA (Analizador integral de requisitos), LEAP SE, Visual Requisite e incluya al menos otras 3 herramientas. Indique la herramienta con la que trabajaría Ud. para la gestión de requisitos, fundamente su selección.

En cuanto a las técnicas para manejar requerimientos, ponemos a su disposición el siguiente cuadro resumen:

Tabla 7.1: Técnicas para manejar requerimientos

Cuando necesite:	Entonces crear:
Establecer mecanismos para manejar los cambios de los requerimientos	Cambio de control de políticas y procedimientos.
Identificar información suplementaria de requisitos	Atributos de requerimientos
Entender la procedencia de los requerimientos y sus relaciones	Matrices de trazabilidad de requerimientos

A continuación vamos a detallar cada una de estas técnicas:

7.2.1 Control de cambios

Cambiar las políticas y procedimientos de control se refiere a establecer mecanismos y reglas para la identificación, evaluación y decidir la forma de integrar los requisitos de las nuevas y cambiantes necesidades en la línea de base.



Es necesario realizar el control de cambios, para anticiparse y responder a las necesidades cambiantes, establecer procedimientos eficaces que permitan los cambios legítimos a los requisitos de un mínimo de perturbaciones para los planes del proyecto, y para hacer el uso más eficaz del tiempo de las partes interesadas para evaluar y resolver los cambios.

Al aplicar el control de cambios se tiene:

- Alinear el proyecto de software con el cambio de necesidades de negocio.
- Asegurar que los clientes entienden y acepten los cambios de requisitos.



- Definir que y cómo los cambios de requisitos serán registrados.
- Establecer los procedimientos para la comprensión de cuáles son los requisitos y los resultados de desarrollo que están asociadas a las nuevas necesidades, para ayudar en el análisis del impacto.
- Programar la planificación de la aplicación o aplazamientos de requisitos.

Para lo cual se debe: Identificar los procedimientos de control de cambio y crear un tablero de control de cambios (CCB), crear la línea base de los requisitos, y finalmente implemente su proceso de control de cambio una vez que usted ha creado una línea base de requerimientos, que se debe ver reflejado en un informe y supervisar cualquier cambio.

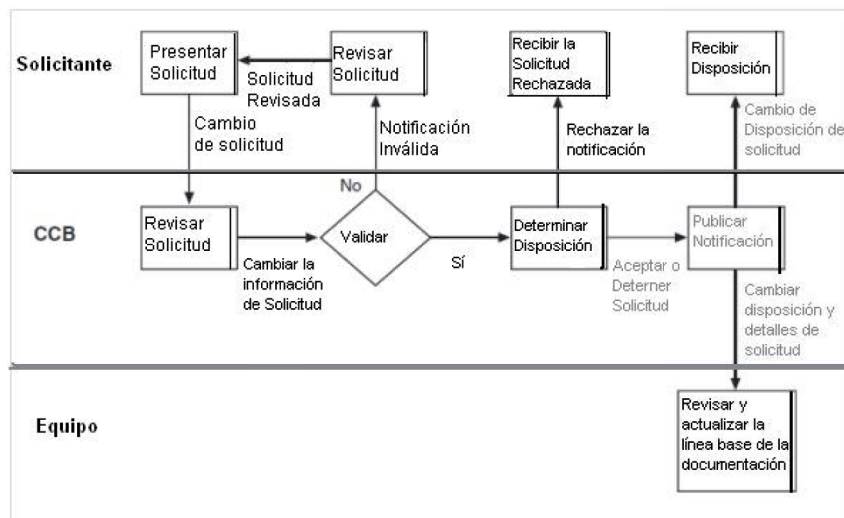


Figura 7.1: Mapa del Proceso de Requerimientos (Gottesdiener E. , 2005)



Existen algunas herramientas de control de cambios, para obtener mayor información al respecto de herramientas de gestión revise el siguiente link: <http://www.ebgconsulting.com>

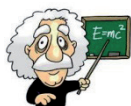
Para realizar el proceso de control de cambio, se debe:

1. Identificar el procedimiento para control de cambios.
Incluir procedimientos para: proponer requerimientos de cambio, realizar un análisis de impacto, actualizar requerimientos, etc.
2. Creación de una línea base para los requisitos
Para crear la línea base de los requisitos es necesario, identificar de forma única cada requisito, pero hay que asegurarse de registrarse todos los atributos necesarios de los requisitos; se recomienda utilizar una herramienta de gestión de requisitos para registrar la información de requisitos.
3. Implementación del proceso de control de cambios
Se debe usar procesos informales de control de cambios para pequeños y mínimos riesgos en los proyectos; como por ejemplo un pequeño desarrollo de un par de semanas para entregar un componente de un software.

Por cada incremento, el equipo debe explorar y priorizar los requerimientos; además se deben desarrollar pruebas, diseño e implementación de código, y presentar el producto a los clientes y usuarios para su evaluación y aceptación.

Pida el personal de negocio y técnico actuar como un consejo de control de cambio, en la decisión de que requerimientos deben desarrollarse prioritariamente.

Típicamente el control de cambio es manejado durante cada incremento diciendo «No» (por ejemplo no permiten a ningunos cambios a los requisitos), aunque el cambio de requisitos que se solicite debería ser registrado.



Ejemplo

A continuación se muestra un ejemplo de cómo se debe desarrollar un control de cambios.

Tabla 7.2: Plantilla de control de cambios

Cambio en la identificación del requisito	Identificación de requerimientos	Pequeña descripción	Fecha solicitada	Disposición	Fecha de disposición	Justificación de la disposición
CR-3	EST.BID-2.0	The system Deferred shall provide contractors with the ability to bid on outstanding jobs	5 de Agosto	Retrasado	8 de Agosto	Depende del apoyo de varias empresas, y añade complejidad técnica y el riesgo. Los objetivos primarios para optimizar las operaciones se deben lograr en primer lugar.



Actividad recomendada

De acuerdo a la solicitud de cambios que se le indica en el Entorno Virtual de Aprendizaje, realice la plantilla de control de cambios y envíela por este mismo entorno.

7.2.2. Atributos de los requerimientos

Los atributos de requisitos son la información adicional asociada con los requisitos; es necesario realizarlo para recopilar información útil para explicar, justificar, el seguimiento y la presentación de informes sobre los requisitos; con lo cual se logra:

- Dar a los interesados información útil para filtrado, selección y análisis de requisitos.



- Proporcionar información a los responsables de control de decisión sobre el impacto de los cambios en los requisitos.
- Ayuda a educar a los nuevos miembros del equipo acerca de los requisitos.
- Proporciona información histórica acerca de los requisitos que ayuda a los equipos a mantener o mejorar el software entregado.

El proceso a realizar es:

- Identificar los atributos que Ud. necesita para hacer un seguimiento.
- Definir y mantener los atributos de todos los requisitos.

7.2.3. Matrices de seguimiento de requerimientos

Identifica cómo los requisitos están relacionados con resultados de desarrollo de software y otros requisitos; Las matrices de seguimiento de requisitos muestran los requisitos relacionados con el linaje hacia adelante y hacia atrás a los entregables del proyecto.

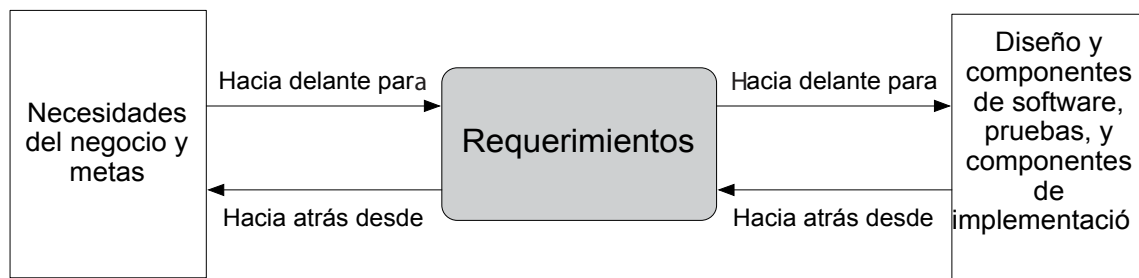


Figura 7.2: Trazabilidad de requisitos (Gottesdiener E. , 2005)

Estas matrices son útiles para entender cómo los cambios de requisitos tienen un impacto y en otras prestaciones posteriores de desarrollo de software.

¿Qué se debe hacer?

- Muestra las interdependencias entre los requisitos.
- Ofrece una visión de gestión mediante la identificación de lo que se debe entregar para satisfacer los requisitos.
- Proporciona informes que sean útiles para la vigilancia del cumplimiento del contrato.
- Demuestra que las necesidades han sido satisfechas por la asociación a los componentes del sistema y las pruebas.

El proceso que se debe seguir es:

- Determinar qué resultados de desarrollo de software se deben dar seguimiento.
- Crear las matrices de seguimiento de los requisitos.



Ejemplo:

A continuación se indica un ejemplo de la construcción de una matriz de seguimiento de requisitos.

Matriz de seguimiento de requisitos durante el desarrollo de requisitos, muestra los requisitos funcionales derivados de los Casos de Uso				
Identificación de requerimientos	CASOS DE USO			
	UC1	UC2	UC3	UC4
SCH-3.2	X			
EST-3.2		X		
CLO-2.3			X	X

Matriz de seguimiento de requerimientos en el desarrollo de software						
Requisitos	Diseño de elementos		Código	Pruebas del sistema	Pruebas de aceptación	
Identificación de requerimientos	Identificación del paquete	Número de versión	Módulo	Script	Pruebas de casos	Fecha de pruebas de aceptación
SCH-1.2	DE-436	1.8	CVCS48491	SSCVSC01	ACTSC124	Julio 5
				SSCVC04	ACTSC249	Julio 8
SEC-1.0	DE-887	2.1	LBR309	SSSR9	ACTSR01	Agosto2

7.3. Ventajas y desventajas del uso de herramientas software en la IR

Para finalizar este tema de gestión de requisitos, se hace una evaluación de las ventajas y desventajas de las herramientas de software que se utilizan en la ingeniería de requisitos:

VENTAJAS:

- Algunas de las herramientas permite generar grandes repositorios de requisitos que pueden ser utilizados no solo en uno, sino en varios proyectos dependiendo del contexto en el que se manejan.
- Permite tener una visibilidad clara de cada uno de los requisitos, es decir, la identificación de funcionalidades y restricciones que tendrá un proyecto o sistema.
- Permite dar seguimiento a los requisitos durante todo el desarrollo de un proyecto, el seguimiento implica trazabilidad, es decir, que los requisitos tendrán vida durante todo el proyecto.
- Algunas de las herramientas permiten el manejo de versiones de los requisitos, es una característica importante a la hora de analizar los requisitos y verificar la evolución de cambios que se ha desarrollado.
- Permiten generar modelos como casos de uso, de estado, diagramas de flujo entre otros, lo que facilita el análisis de los requisitos, su especificación y validación.
- Manejo de un grupo de usuarios, cada usuario registrado en el proyecto tendrá acceso a todos los requisitos capturados, podrá modificarlos, analizarlos, validarlos y notificar los cambios realizados.



- g. Controlar, administrar los requisitos y sus cambios para luego reutilizarlos en cualquier actividad del proceso de software que se necesite.
- h. Reducción de costos y tiempo en el proceso de requisitos de los proyectos.

DESVENTAJAS:

- Una de las principales desventajas es la cultura de uso de las herramientas de ingeniería de requisitos, dichas herramientas no son muy utilizadas en la administración y gestión de requisitos en el desarrollo de proyectos, en la mayoría de casos este proceso es manual, y en otros casos se usa plantillas elaboradas en Microsoft Word solo para captura de requisitos.
- La infraestructura necesaria para usar estas herramientas es otro impedimento, se necesita comunicar los equipos en red, en algunos casos, la utilización de bases de datos potentes para la obtención de grandes repositorios de requisitos es muy necesario.
- Los costos por licenciamiento de las herramientas como por ejemplo Rational Requisite Pro e IRqA (que ofrecen grandes beneficios), son altos. Adicional a esto, se suma los costos de las herramientas que se incorporen o interactúen con las de requisitos.
- No permiten incorporar procesos ni modelos de ingeniería de requisitos.

En la siguiente tabla se presentan algunas características de las herramientas para la gestión de requisitos, entonces dependiendo de las necesidades que se tenga deberá seleccionar la herramienta que mejor se ajuste a sus necesidades.

Tabla 7.3: Contrastación de herramientas frente a características de administración de requisitos

Características	Requisite Pro	IRqA	LEAP SE	Visual Requisite
Captura	X	X	X	X
Análisis	X	X		
Negociación				
Especificación	X	X		
Validación y verificación		X		
Gestión de requisitos	X	X	X	X
Trazabilidad	X	X		
Facilidad de uso			X	X
No redundancia				
Generación de modelos	X	X		X
Comunicación de requisitos	X	X		
Generación de informes	X	X		X



Estas herramientas están en Internet y presentan versiones de prueba que permitirán evaluarlas, descargue entonces la herramienta de requisite pro y realice el proceso de especificación de requisitos esto ayudará a la mejor comprensión de este tema.

Se ha concluido el estudio de esta unidad, por lo que conviene comprobar su aprendizaje en los temas tratados para lo cual es necesario desarrollar las autoevaluaciones de conocimiento.



Autoevaluación 7



Realice las actividades que se indican en cada uno de los literales.

7.1 Responda verdadero o falso según el enunciado que se presenta:

1	La gestión de requisitos es el proceso de seguimiento de la situación de los requisitos y controlar los cambios de los requisitos de la línea de base.	()
2	Para proyectos de desarrollo de software de un nivel medio-alto no es necesario realizar una gestión de requisitos.	()
3.	La característica de trazabilidad de las herramientas de gestión de requisitos se refiere a que cada uno de los requisitos posee su propia identificación, diferenciándolos de los demás.	()
4.	El control de cambios permite alinear el proyecto de software con el cambio de necesidades de negocio.	()
5.	El control de cambios permite crear una línea base de los requisitos.	()
6.	Para crear la línea base de los requisitos es necesario, identificar de forma única cada requisito.	()
7.	El personal de negocio y técnico actúan como un consejo de control de cambio.	()
8.	Los atributos de los requisitos son necesarios para explicar, justificar, el seguimiento y la presentación de informes.	()
9.	En las matrices de seguimiento se reflejan únicamente los cambios de requisitos	()
10.	Las matrices de seguimiento de requisitos no son aptas para vigilar el cumplimiento del contrato.	()

7.2 Basado en los conceptos que se han explicado en este capítulo determine los siguientes aspectos tomando como problema el ejemplo mencionado en el capítulo 5.

- a) Línea base.
- b) Control de cambios()
- c) Matriz de seguimientos de los requisitos.



Ir a solucionario



7. Solucionario

Unidad 1: Fundamentos de la ingeniería de requerimientos

1. Elija la opción correcta, en los literales de opción múltiple en la pregunta2 escriba lo que se pide.

UNIDAD 1	
PREGUNTA	RESPUESTA
1	1.1
2	a) Stakeholder, b) Escenario, c) Requisitos de usuario, d) Requisitos del sistema
3	3.1
4	4.2
5	5.4
6	6.3
7	7.3
8	8.2
9*	9.2
10	10.1

Unidad 2: Preparar el escenario para el desarrollo de requerimientos

1. Lea detenidamente la afirmación y escriba una V si considera que es verdadera o una F si no lo es, en el paréntesis respectivo.

UNIDAD 2	
PREGUNTA	RESPUESTA
1	V
2	F
3	F
4	V
5	V
6	F
7	V
8	V
9	F
10	V

2. Complete la siguiente tabla, con la herramienta apropiada.

- a) Visionamiento
- b) Un glosario de términos
- c) Crear las estrategias para mitigar los riesgos



Unidad 3: Elicitación de requerimientos

1. Lea detenidamente la afirmación y escriba una V si considera que es verdadera o una F si no lo es, en el paréntesis respectivo.

UNIDAD 3	
PREGUNTA	RESPUESTA
1	V
2	V
3	V
4	V
5	F
6	F
7	F
8	V
9	V
10	V

2. Complete la siguiente tabla, con la herramienta apropiada.

- Listado de fuentes de requerimientos.
- Categoría de los stakeholders.
- Perfil de los stakeholders.
- Entrevistas, Prototipos, Talleres, Observación, Documentación, etc.
- Plan de elicitación de stakeholders.

Unidad 4: Análisis de requerimientos

1. Lea detenidamente la afirmación y escriba una V si considera que es verdadera o una F si no lo es, en el paréntesis respectivo.

UNIDAD 4	
PREGUNTA	RESPUESTA
1	V
2	F
3	V
4	V
5	F
6	V
7	V
8	V
9	F
10	V

2. Liste las partes del ciclo de análisis de requerimientos.
- Modelado del negocio
 - Defina el alcance del proyecto
 - Crear al detalle los modelos de requerimientos de usuario
 - Priorizar los requerimientos
3. Complete la siguiente tabla acerca de las herramientas y técnicas para análisis de requerimientos.
- Modelar el negocio
 - Entender el alcance del proyecto
 - Adicionar detalle a los requerimientos de usuario
 - Negociar la importancia entre los requisitos



Unidad 5: Especificaciones de requerimientos

1. Lea detenidamente la afirmación y escriba una V si considera que es verdadera o una F si no lo es, en el paréntesis respectivo.

UNIDAD 5	
PREGUNTA	RESPUESTA
1	V
2	F
3	V
4	V
5	V
6	F
7	V
8	F
9	F
10	V

Unidad 6: Validación de requerimientos

1. Lea detenidamente la afirmación y escriba una V si considera que es verdadera o una F si no lo es, en el paréntesis respectivo.

UNIDAD 6	
PREGUNTA	RESPUESTA
1	V
2	V
3	F
4	F
5	V
6	V
7	V
8	F
9	F
10	V



Unidad 7: Gestión de requerimientos

1. Lea detenidamente la afirmación y escriba una V si considera que es verdadera o una F si no lo es, en el paréntesis respectivo.

UNIDAD 7	
PREGUNTA	RESPUESTA
1	V
2	F
3	V
4	V
5	V
6	V
7	V
8	V
9	F
10	F

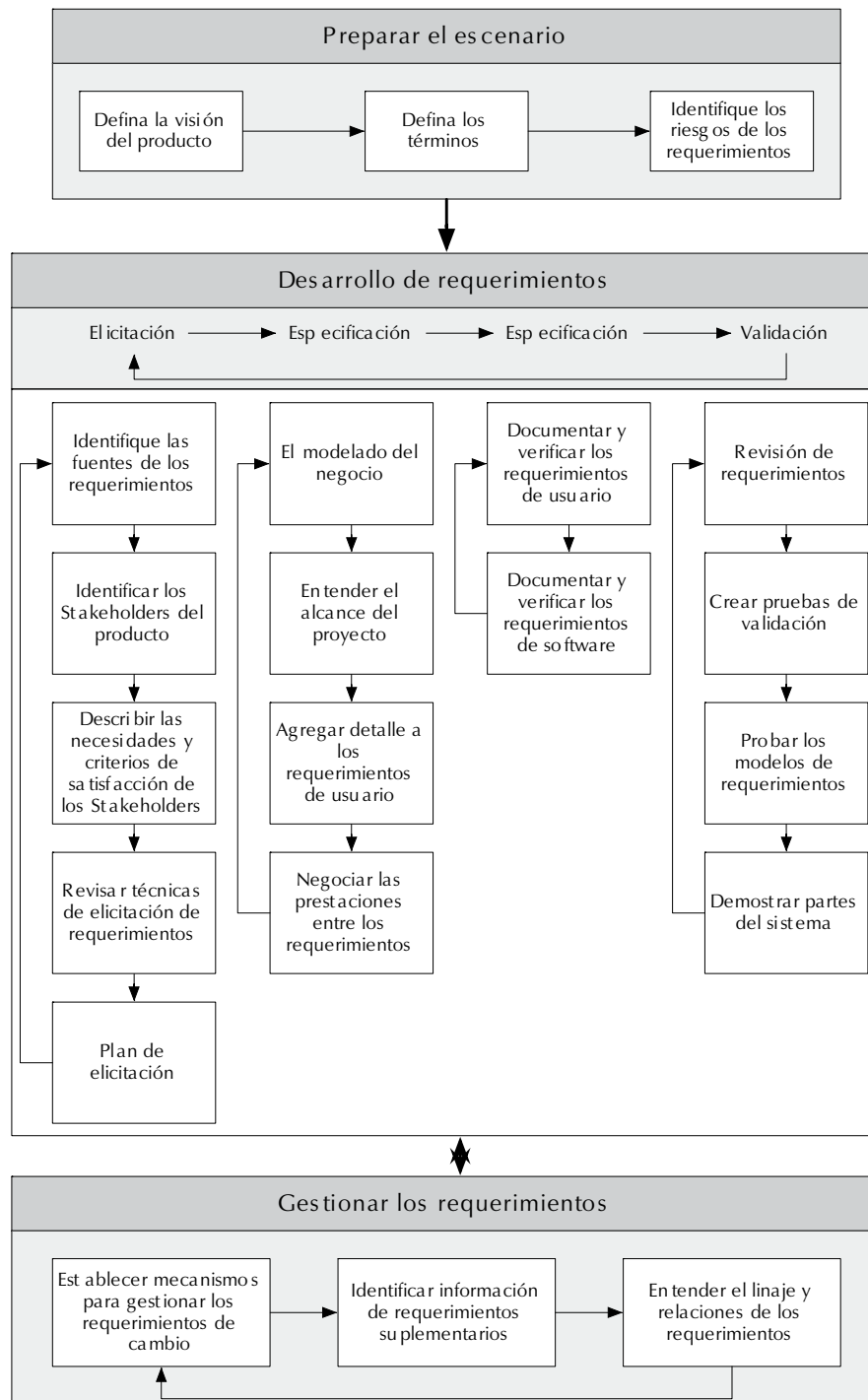


8. Anexos



El presente material ha sido reproducido con fines netamente didácticos, cuyo objetivo es brindar al estudiante mayores elementos de juicio para la comprensión de la materia, por lo tanto no tiene fin comercial.

Anexo 1: Desarrollo de requerimientos. (Gottesdiener E., 2005)



Anexo 2: Factores a considerar al aplicar técnicas de obtención de requerimientos.

Cada proyecto es diferente. Al seleccionar la técnica para levantar información considere los factores que se indican en la siguiente tabla, para guiarse y obtenga requerimientos satisfactorios. (Gottesdiener E. , 2005)

Técnica	Factores a considerar		
	Número de usuarios finales	Accesibilidad a los expertos en la materia	Tiempo para recopilar requerimientos
Entrevistas	No es factible con un gran número de usuarios y expertos; use representantes.	Se requiere acceso a los entrevistados, puede hacer entrevistas telefónicas, aunque se limita la calidad de información. Los entrevistadores deben trasladarse al lugar donde están los usuarios.	Tiempo total para conducir las entrevistas, clasificar los resultados, y aclarar conflictos de datos que puede tomar días o semanas.
Prototipos exploratorios	Seleccione uno o más usuarios representativos de cada grupo de usuarios.	Se requiere de acceso directos a los usuarios para revisar los prototipos, a menos que utilice herramientas en línea para la revisión. Lo ideal es combinar los prototipos con talleres o análisis de tareas de usuario, que requiera acceso directo con el usuario.	Los prototipos que son desechados pueden ser elaborados en horas, mientras que los prototipos evolutivos pueden tomar días o semanas. Revisar solamente toma horas. Múltiples revisiones podrían ser conducidos como un desarrollo de prototipo interactivo.
Talleres	Seleccione uno o más usuarios representativos de cada grupo de usuarios.	Se basa en una interacción cara a cara para ser más efectivo; usualmente se requiere de múltiples talleres dentro de un marco de tiempo corto. Los usuarios podrían tener que viajar para reunirse en los talleres.	Conseguir a las personas apropiadas para los talleres planificados reduce el tiempo de desarrollo de requerimientos a días o semanas e incrementa la calidad de los requerimientos.
Grupos focales	Seleccione uno o más usuarios representativos de cada grupo de usuarios.	Se basa en una interacción cara a cara; por lo general se requiere del enfoque de múltiples grupos focales.	Usualmente toma semanas planificar, conducir, y analizar los datos.



Observación	Seleccione uno o más usuarios representativos de cada grupo de usuarios.	Se basa en el acceso en tiempo real al ambiente de trabajo de los usuarios. Los observadores tendrán que trasladarse a los sitios de trabajo de los usuarios.	Se puede hacer en días o semanas, dependiendo de la predisposición del usuario.
Análisis de tareas de las del usuario	Seleccione uno o más usuarios de cada grupo de usuarios directos o clientes externos.	Se basa en una interacción cara a cara para ser más efectivos. Usualmente se requiere de reuniones de corto tiempo.	Las reuniones de usuarios para documentar las tareas generalmente toma días.
Estudio de documentación existente	No aplicable	No aplicable	El análisis y la documentación puede tomar varios días.
Encuestas	Útil para una muestra de un gran número de stakeholders	Acceso físico no requerido	Diseñar la encuesta, obtener las respuestas y organizar los datos puede tomar semanas o meses.

Es muy importante respetar el tiempo de los stakeholders cuando se utilizan técnicas que implican la interacción directa de estos. Asegúrese de que comience y termine a tiempo tanto las entrevistas, los talleres, los grupos focales, entre otras.

Habilidades y características necesarias

Independientemente de la técnica de obtención utilizada, necesita sólidas habilidades de análisis y capacidad de ser neutral. Adicionalmente habilidades y características para cada técnica de obtención incluye:

Técnica	Habilidades de facilitación	Habilidades interpersonales	Habilidades para entrevistar	Habilidades de observación/ escuchar	Habilidades de escritura técnica
Entrevistas		X	X	X	
Prototipos					
Talleres	X	X		X	
Grupos focales	X	X	X	X	
Análisis de tareas de usuario		X	X	X	X
Observación		X		X	
Estudio de documentación					X
Encuestas			X		X

Anexo 3: Esquema para documentar requerimientos de usuario. (Gottesdiener E. , 2005)

1. Introducción
 - 1.1. Objetivo y antecedentes
 - 1.2. Información del negocio y necesidades del usuario
 - 1.3. Documento de información general y convenio
 - 1.4. Referencias
2. Sistema o situación actual
 - 2.1. Antecedentes, objetivos, y alcance o situación del sistema actual.
 - 2.2. Sistema actual o procesos de negocio
 - 2.3. Personas, organizaciones y lugares.
3. Justificación para el nuevo sistema
 - 3.1. Razones fundamentales para el nuevo sistema
 - 3.2. Descripción general del sistema y los procesos de negocio afectados
 - 3.3. Las personas afectadas, las funciones y las organizaciones
 - 3.4. Las prioridades y el alcance del cambio
 - 3.5. El impacto en las operaciones, la organización, la gente, y soporte
 - 3.6. Impacto en las políticas, normas y reglas de negocio.
4. Nuevas funcionalidades
 - 4.1. Usuarios y perfiles de usuario
 - 4.2. Nuevas o actualizadas capacidades de usuario (Apéndice D)
 - 4.3. Impacto de las nuevas capacidades en los procesos de usuario y en el sistema
 - 4.4. Interfaces con otros sistemas
 - 4.5. Entorno de soporte de usuario y documentación de usuario
5. Evaluación del sistema propuesto
 - 5.1. Ventajas, desventajas y limitaciones.
 - 5.2. Plan de administración del cambio del negocio y la organización
 - 5.3. Cuestiones operativas
 - 5.4. Alternativas a considerar

Apéndices

A: Glosario de términos

B: Diccionario de datos

C: Diagrama de contexto

D: Casos de uso y escenarios.



Anexo 4: Documento de especificación de requerimientos del caso de estudio, basado en el estándar IEEE Std 830-1998.

1. Introducción

1.1. Propósito

La gestión de trámites por parte de los estudiantes de MAD de la UTPL, requieren de una atención adecuada por parte de cada uno de los estamentos universitarios que intervienen en la resolución de los mismos. La aplicación "Autotrámites" permite el aprovechamiento de los recursos que la UTPL posee en cuanto a personas, como a tecnología. En esta versión, la 1.0 se establecen las necesidades de los usuarios, secretarías de centros, secretarías de carrera, coordinación académica y profesores para gestionar y responder al flujo de eventos que se realizan por cada trámite. La aplicación utilizará la información académica del sistema académico "Syllabus" y del sistema financiero "Baan".

1.2. Convenciones del documento

Al escribir este documento se basa en la plantilla del estándar IEEE-830.

1.3. Público objetivo y sugerencias de lectura

Este documento está dirigido para:

- Director de MAD
- Coordinador académico de la MAD
- Director del área de procesos de la MAD
- Coordinador del centro asociado de la MAD
- Profesores
- Secretarías
- Estudiantes

1.4. Alcance

"Autotrámites" es una aplicación web que permitirá:

- Que las solicitudes de trámites académico / contable se registren en el sistema.
- Que los requisitos asociados a un trámite se ingresen obligatoriamente.
- Que se notifique a los involucrados (personal UTPL-estudiantes) del inicio de un trámite.
- Que se visualicen las solicitudes de trámite en formato digital.
- Que las personas involucradas puedan gestionar un trámite.
- Notificar al estudiante de la ejecución de un trámite.
- Identificar a la persona que está atendiendo un trámite.
- Que se disponga de estadísticas que permitan la toma de decisiones.
- Que se pueda obtener información académico/contable.
- Que los trámites puedan ser configurados.
- Que se puedan ejecutar trámites automáticos de acuerdo al trámite solicitado.

1.5. Referencias

- Acta de constitución del proyecto.
- Documento de visionamiento y problemática.
- Plan de especificación de requerimientos, entre otros.

2. Descripción general

2.1. Perspectiva del producto

El sistema “Autotrámites” será integrado en el Sistema de Gestión Académica y tendrá como objetivo la gestión automática de trámites a través de un motor de workflow que proveerá la infraestructura para diseñar, construir, ejecutar, monitorear a cada uno de los trámites.

Debido a que el sistema se deberá integrar con el Sistema de Gestión Académica, interactuará con sus módulos principales: Gestión Académica, Gestión Financiera, Configuración.

SISTEMA DE GESTIÓN ACADÉMICA

Módulo de	Descripción
Gestión Académica	Maneja información referente los asuntos académicos como por ejemplo: planes de estudio, periodos académicos, notas, expedientes de los estudiantes, etc.
Gestión Financiera	Este módulo permite consultar información de los saldos a favor y en contra de los estudiantes que realicen solicitudes de trámite.
Configuración	Este módulo permite la parametrización del sistema, por ejemplo cierre de matrículas, cierre de ingreso de notas, restricciones a funcionalidades, creación de catálogos, etc.
Servicios en Línea al Estudiante	Permite al consultar información académico/contable al estudiante (Notas, saldos a favor, saldos en contra etc.).

MÓDULO DE AUTOMATIZACIÓN DE TRÁMITES

Módulo de	Descripción
Funcionalidades generales	Permite invocar a cualquier trámite y además controlar las actividades que lo gestionan.
Funcionalidades específicas	Permite llevar adelante en el sistema las actividades específicas definidas en cada uno de los trámites.



2.2. Funciones del producto

El sistema de Autotrámites constará de las siguientes funcionalidades:

Registro de trámites

Contempla servicios como:

- Búsqueda de estudiantes.
- Escoger tipo de trámite.
- Digitalizar documentos.
- Emitir notificaciones.
- Determina el flujo de eventos para cada trámite.
- Asigna cada trámite al rol de usuario.

Resolución de trámites

Contempla los siguientes servicios:

- Presenta una bandeja de trámites con los trámites por resolver.
- Presenta información dependiendo del trámite.
- Resuelve la actividad encomendada.
- Notifica de la solución de la actividad.

Consulta de trámites

Dependiendo del trámite y rol del usuario se presenta información referente al trámite.

2.3. Clases y características de usuario

Los usuarios que intervienen son:

- Estudiante: Recibe notificación de las actividades respecto al trámite.
- Secretaria centro: Quien registra el trámite en el sistema.
- Profesor: Quien registra los datos cuando existe un trámite que tiene que ver con su materia.
- Coordinación académica: Resuelve ciertas actividades que tienen que ver con la coordinación.

2.4. Ambiente de operación

El sistema autotrámites funcionará envevido en el sistema de gestión académica de la UTPL.

2.5. Restricciones de diseño e implementación

- El sistema deberá adaptarse al proceso que se realiza en el grupo de software de la UTPL.
- Se acoplará a la arquitectura del sistema académico de la UTPL.
- Utilizará los servicios del sistema financiero.

2.6. Asunciones y dependencias

Asunciones

- El acceso a las características del sistema web se realizará previamente un registro de usuarios.
- Las actividades que pueda realizar cada usuario dependerá del rol que se asigne en el sistema.

Dependencias

- El sistema se basará en el uso de herramientas que esta licenciado la UTPL, y como motor central el workflow de Oracle.

3. Características del sistema Autotrámites

3.1. Registrar solicitud de trámite

3.1.1. Descripción.

El estudiante inicia el proceso al entregar todos los documentos (dependiendo del trámite), en el centro asociado, para que la secretaria con la ayuda del sistema registre en línea el trámite y se notifique tanto al estudiante como a quienes deben resolver sobre la existencia del trámite.

3.1.2. Requerimientos funcionales

Los requerimientos que se identifican son:

FUN-AUT-01. Seleccionar estudiante

Entrada:

La selección del estudiante consiste en buscar y presentar si tiene registrado previamente algún trámite. Se presenta un formulario con el tipo de búsqueda a realizar: (básica o avanzada), y los campos son:

- Cédula del estudiante
- Nombres del estudiante
- Apellidos del estudiante

Proceso:

Para realizar la búsqueda se puede realizar de dos formas:

- Búsqueda básica: Se la realiza ingresando el número de cédula del estudiante.
- Búsqueda avanzada: Se realiza ingresando: nombres y/o apellidos del estudiante.

**Salida:**

Si el estudiante fue encontrado presentar:

- Datos académicos adicionales del estudiante: carrera, período académico, centro universitario.
- Datos de trámites ya registrados anteriormente: tipo trámite, fecha inicio, fecha fin, responsable, actividad, estado.

FUN-AUT-02. Registrar documentos

...

FUN-AUT-03. Registrar tipo trámite

...

FUN-AUT-04. Consultar estado del trámite

...

3. Requerimientos no funcionales

- 3.1. La aplicación deberá ser tan familiar como sea posible a los usuarios que han usado otras aplicaciones web (Syllabus) y aplicaciones de escritorio en Windows.
- 3.2. La aplicación debe ser fácil de entender y utilizar.

Se deberá diseñar una aplicación amigable, que minimice la sobrecarga cognitiva y perceptiva del usuario de la aplicación, o sea, diseñar de forma óptima el conjunto de operaciones, niveles, dispositivos y lenguajes resumidos en la etapa de análisis.

- 3.3. El flujo de trabajo de cada trámite deberá ser fácilmente interpretado en el motor de workflow.

Cada una de las actividades de trámite, deberán ser especificadas de acuerdo al proceso definido para la resolución de este, con la finalidad de que no existan actividades adicionales a considerar en el flujo de trabajo .

Apéndice A: Glosario

1) Términos generales

- **Actividad de un trámite**

Representa cada uno de los estados y diligencias que la solicitud de un estudiante tiene que recorrer en el flujo de trabajo hasta su conclusión.

- **Digitalización**

Proceso por el cual se capturan los datos de un formato físico y se lo expresa datos en forma digital.

- **Estatus de trámite**

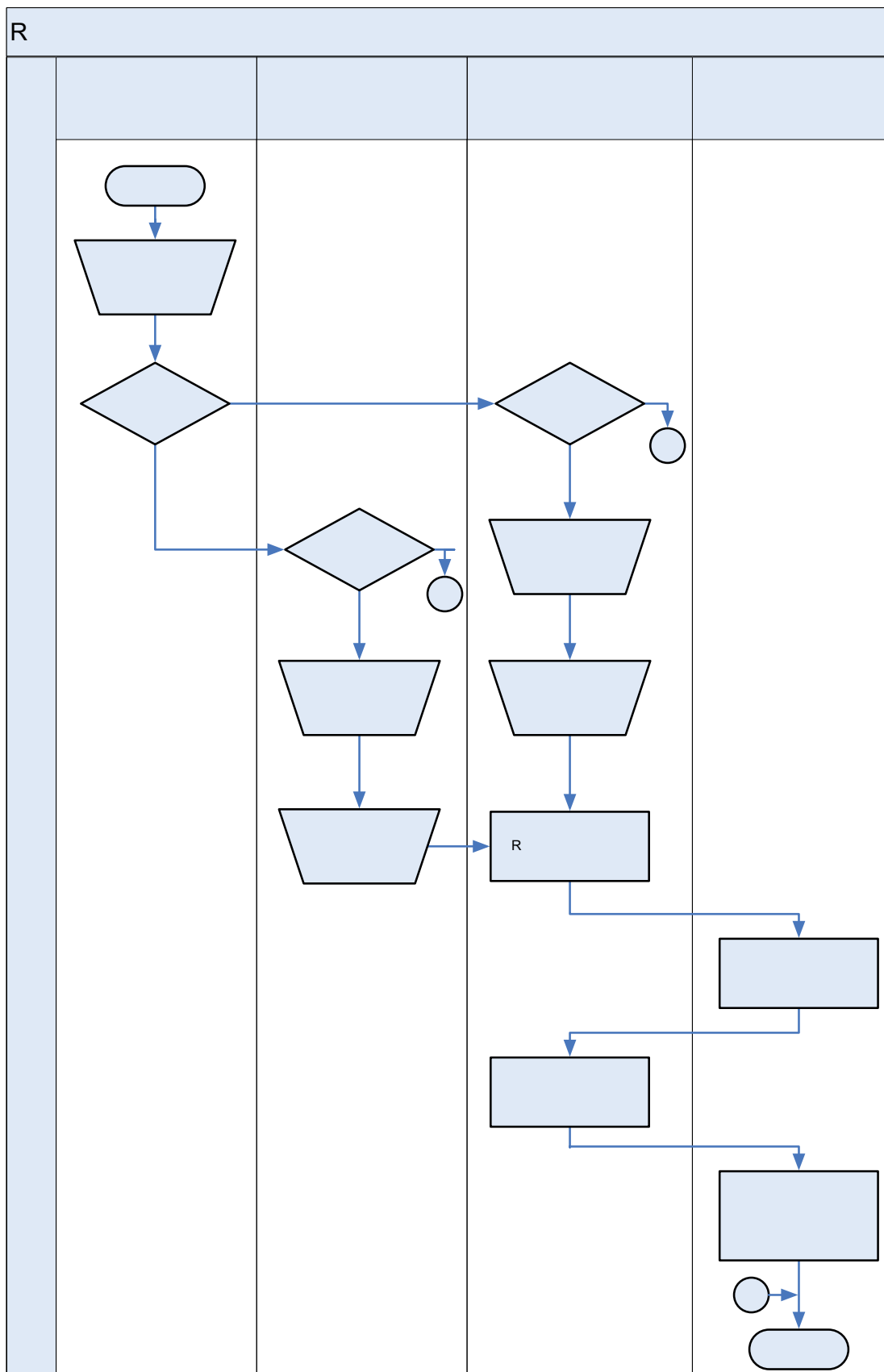
Representa la situación relativa de una actividad de trámite dentro de un determinado flujo de trabajo.

- **Flujo de trabajo**

Representa los aspectos operacionales de una actividad de trabajo: cómo se estructuran las tareas, cómo se realizan, cuál es su orden correlativo, cómo se sincronizan, cómo fluye la información que soporta las tareas y cómo se le hace seguimiento al cumplimiento de las tareas.



Apéndice B: Mapa de procesos



Apéndice C: Matriz de trazabilidad

Necesidades	Características	Requerimientos	Casos de Uso
Que se registren las solicitudes de trámites académicos y/o contables en el sistema	Facilidad para ingresar solicitudes de trámites en el sistema	La aplicación debe permitir ingresar solicitudes de trámites en el sistema	Ingresar trámite
Que los requisitos necesarios para llevar a cabo un trámite se presenten al inicio de cada trámite y se ingresen obligatoriamente	- Facilidad para identificar todos los requisitos asociados a un trámite. - Facilidad para que el ingreso de requisitos asociado a un trámite sea obligatorio	- La aplicación debe permitir que uno o varios requisitos se puedan asociar a un trámite - La aplicación debe considerar el ingreso obligatorio de requisitos.	- Ingresar trámite - Configurar Trámites
Se notifique a las personas involucradas de la UTPL del inicio de un trámite	- Facilidad para comunicar electrónicamente al personal involucrado del inicio de una solicitud de trámite antes de su ejecución - Facilidad para consultar notificaciones que tiene un involucrado en la resolución de trámites	- La aplicación debe comunicar electrónicamente al personal involucrado del inicio de una solicitud de trámite antes de su ejecución - La aplicación debe permitir consultar notificaciones que tiene un involucrado en la resolución de trámites	- Ingresar trámite - Consultar Notificaciones
Que las solicitudes de trámites puedan ser visualizadas en formato digital	- Facilitar la digitalización de las solicitudes de trámite. - Facilidad para visualizar las solicitudes de trámites	La aplicación debe realizar la digitalización de las solicitudes de trámite.	Ingresar un trámite



Necesidades	Características	Requerimientos	Casos de Uso
Que las personas involucradas puedan gestionar un trámite.	- Facilidad para realizar la aceptación de solicitud de trámite - Facilidad para realizar las reprobación o notificación de trámite - Facilidad para conocer el status de un trámite en determinada instancia, ingresando Actividad-Estado-Responsable-Tiempos de Actividad. - Facilidad para visualizar status de un trámite en el motor de workflow.	- La aplicación debe permitir la aceptación de una solicitud de trámite - La aplicación permitir realizar las reprobación o notificación de trámite - La aplicación debe permitir conocer el status de un trámite en determinada instancia, ingresando Actividad-Estado-Responsable-Tiempos de Actividad. - La aplicación debe permitir la visualización del status de un trámite a través de un motor de workflow	- Ejecutar Tramites - Consultar Status
Que se notifique al solicitante la ejecución de un trámite	- Facilidad para presentar información ágil de los trámites. - Facilidad para que el estudiante pueda verificar el resultado del trámite desde los servicios en línea. - Facilidad para que desde los CUA se pueda verificar el trámite	- La aplicación debe permitir que el estudiante pueda verificar el resultado del trámite desde los servicios en línea. - La aplicación debe permitir que desde los CUA's se pueda verificar el trámite	Consultar status

Necesidades	Características	Requerimientos	Casos de Uso
Que se pueda identificar a las persona que esta atendiendo un trámite	Facilidad para identificar al personal involucrado en la resolución de un trámite	La aplicación debe permitir identificar al personal involucrado en la resolución de un trámite	Consultar status de trámite
Que se disponga de estadísticas que permitan la toma de decisión	- Facilidad para obtener información estadística de trámites en ejecución. - Facilidad para obtener información estadística de trámites ejecutados. - Facilidad para obtener información estadística del tiempo promedio de culminación de un trámite. - Facilidad para visualizar la información estadística realizando combinaciones de los criterios establecidos.	- La aplicación debe permitir obtener información estadística de trámites ejecutados. - La aplicación debe permitir la obtención de información estadística del tiempo promedio de culminación de un trámite. - La aplicación debe visualizar la información estadística realizada mediante combinaciones de los criterios establecidos.	Obtener estadísticas



Necesidades	Características	Requerimientos	Casos de Uso
Qué se pueda acceder a información académico/contable.	- Facilidad para verificar información contable (Facturas, Saldos a favor, saldos en contra, etc.) - Facilidad para verificar información académica (Expediente, Matriculas, Asignaturas, etc.)	- La aplicación debe permitir la verificar de información contable (Facturas, Saldos a favor, saldos en contra - La aplicación debe permitir verificar información académica (Expediente, Matriculas, Asignaturas, etc.)	Consultar información
Que lo trámites puedan ser configurados: número de pasos, inicio – fin	- Facilidad para registrar Trámites - Facilidad para registrar los requisitos asociados a un trámite - Facilidad para registrar tiempos de respuesta por trámite. - Facilidad para categorizar un trámite. - Facilidad para definir el flujo de trabajo de un trámite.	- La aplicación debe permitir registrar cada uno de los trámites - La aplicación debe permitir registrar tiempos de respuesta por trámite. - La aplicación debe permitir categorizar un trámite.	Configurar Trámite
Que se puedan generar procesos automáticos con el SGA para la ejecución de solicitudes de trámite (ej. anulación de matriculas.)	Facilidad para integrar los trámites que se realizan en el SGA, para que su ejecución sea automática.	La aplicación debe permitir la resolución automática de trámites mediante la interacción con el SGA	Ejecutar Trámite Automáticos

SC, MS/gg/2012-01-12/183
cII/2014-06-16

