

## 2. Análisis de Sistemas de Información. Ingeniería de Requerimientos.

La parte más difícil de construir un sistema es precisamente saber qué construir. Ninguna otra parte del trabajo conceptual es tan difícil como establecer los requerimientos técnicos detallados, incluyendo todas las interfaces con gente, máquinas y otros sistemas (Brooks, 1987):

- Ninguna otra parte del trabajo afecta tanto al sistema si es hecha mal.
- Ninguna es tan difícil de corregir más adelante.

Entonces, la tarea más importante que el ingeniero de software hace para el cliente es la extracción iterativa y el refinamiento de los requerimientos del producto (Brooks, 1987).

Los requerimientos para un sistema son descripciones de lo que el sistema debe hacer: el servicio que ofrece y las restricciones en su operación. Tales requerimientos reflejan las necesidades de los clientes por un sistema que atienda cierto propósito, como sería controlar un dispositivo, colocar un pedido o buscar información. Al proceso de descubrir, analizar, documentar y verificar estos servicios y restricciones se le llama ingeniería de requerimientos (IR).

Al término “requerimiento” se lo emplea de distintas formas en la industria del software. En algunos casos, un requerimiento es simplemente un enunciado abstracto de alto nivel en un servicio que debe proporcionar un sistema, o bien, una restricción sobre un sistema. En el otro extremo, consiste en una definición detallada y formal de una función del sistema. Davis (1993) explica por qué existen esas diferencias:

- Si una compañía desea otorgar un contrato para un gran proyecto de desarrollo de software, tiene que definir sus necesidades de una forma suficientemente abstracta para que una solución no esté predefinida. Los requerimientos deben redactarse de tal forma que muchos proveedores liciten en pos del contrato, ofreciendo, tal vez, diferentes maneras de cubrir las necesidades de organización del cliente.
- Una vez otorgado el contrato, el proveedor tiene que escribir con más detalle una definición del sistema para el cliente, de modo que éste comprenda y valide lo que hará el software. Estos documentos suelen nombrarse documentos de requerimientos para el sistema.

Algunos de los problemas que surgen durante el proceso de ingeniería de requerimientos son resultado del fracaso de hacer una separación clara entre esos diferentes niveles de descripción. En Sommerville (2011) se distinguen estos dos tipos de requerimientos bajo los términos: “requerimientos del usuario” para representar los requerimientos abstractos de alto nivel; y “requerimientos del sistema” para caracterizar la descripción detallada de lo que el sistema debe hacer.

Los requerimientos del usuario y los requerimientos del sistema se definen del siguiente modo:

1. Los requerimientos del usuario son enunciados en lenguaje natural junto con diagramas, acerca de qué servicios esperan los usuarios del sistema y de las restricciones con las cuales éste debe operar.
2. Los requerimientos del sistema son descripciones más detalladas de las funciones, los servicios y las restricciones operacionales del sistema de software. El documento de requerimientos del sistema (llamado en ocasiones especificación funcional) tiene que definir con exactitud lo que se implementará. Puede formar parte del contrato entre el comprador del sistema y los desarrolladores del software.

Los diferentes niveles de requerimientos son útiles, debido a que informan sobre el sistema a distintos tipos de lector. A continuación, se ilustra la diferencia entre los requerimientos del usuario y del sistema (Sommerville, 2011).

Definición del requerimiento del usuario:

1. El MHC-PMS elaborará mensualmente informes administrativos que revelen el costo de los medicamentos prescritos por cada clínica durante ese mes.

Especificación de los requerimientos del sistema:

- 1.1. En el último día laboral de cada mes se redactará un resumen de los medicamentos prescritos, su costo y las clínicas que los prescriben.
- 1.2. El sistema elaborará automáticamente el informe que se imprimirá después de las 17:30 del último día laboral del mes.
- 1.3. Se realizará un reporte para cada clínica junto con los nombres de cada medicamento, el número de prescripciones, las dosis prescritas y el costo total de los medicamentos prescritos.
- 1.4. Si los medicamentos están disponibles en diferentes unidades de dosis (por ejemplo, 10 mg, 20 mg) se harán informes por separado para cada unidad de dosis.
- 1.5. El acceso a los informes de costos se restringirá a usuarios autorizados en la lista de control de acceso administrativo.

Este ejemplo de un sistema de administración de pacientes para apoyar la atención a la salud mental (MHC-PMS) muestra cómo los requerimientos del usuario se extienden hacia varios requerimientos del sistema. En el ejemplo se observa que el requerimiento del usuario es muy general. Los requerimientos del sistema ofrecen información más específica sobre los servicios y las funciones del sistema que se implementará.

Es necesario escribir los requerimientos con diferentes niveles de detalle, ya que varios lectores los usarán de distintas formas. Por lo general, los lectores de requerimientos de usuarios no están interesados en la manera en que se implementará el sistema, y quizá sean administradores a quienes no les atraigan las facilidades detalladas del sistema, ejemplos de posibles lectores son:

- Gerentes del cliente
- Usuarios finales del sistema
- Ingenieros del cliente
- Gerentes de los contratistas
- Arquitectos del sistema

Mientras que los lectores de los requerimientos del sistema necesitan conocer con más precisión qué hará el sistema, ya que están preocupados sobre cómo apoyará los procesos de negocio o porque están inmersos en la implementación del sistema, posibles lectores son:

- Usuarios finales del sistema
- Ingenieros del cliente
- Arquitectos del sistema
- Desarrolladores de software

Sommerville (2011) en el capítulo 4 presenta un panorama “tradicional” de los requerimientos, más que de los requerimientos en los procesos ágiles. Para la mayoría de los sistemas grandes, todavía se presenta una fase de ingeniería de requerimientos claramente identificable, antes de

comenzar la implementación del sistema. El resultado es un documento de requerimientos que puede formar parte del contrato de desarrollo del sistema. Desde luego, por lo común hay cambios posteriores a los requerimientos, en tanto que los requerimientos del usuario podrían extenderse como requerimientos de sistema más detallados.

### 2.1. Requerimientos funcionales y no funcionales.

A menudo, los requerimientos del sistema de software se clasifican como requerimientos funcionales o requerimientos no funcionales:

1. **Requerimientos funcionales.** Son enunciados acerca de servicios que el sistema debe proveer, de cómo debería reaccionar el sistema a entradas particulares y de cómo debería comportarse el sistema en situaciones específicas. En algunos casos, los requerimientos funcionales también explican lo que no debe hacer el sistema.
2. **Requerimientos no funcionales.** Son limitaciones sobre servicios o funciones que ofrece el sistema. Incluyen restricciones tanto de temporización y del proceso de desarrollo, como impuestas por los estándares. Los requerimientos no funcionales se suelen aplicar al sistema como un todo, más que a características o a servicios individuales del sistema.

En realidad, la distinción entre los diferentes tipos de requerimientos no es tan clara como sugieren estas definiciones sencillas. Un requerimiento de un usuario interesado por la seguridad, como el enunciado que limita el acceso a usuarios autorizados, parecería un requerimiento no funcional. Sin embargo, cuando se desarrolla con más detalle, este requerimiento puede generar otros requerimientos que son evidentemente funcionales, como la necesidad de incluir facilidades de autenticación en el sistema.

Esto muestra que los requerimientos no son independientes y que un requerimiento genera o restringe normalmente otros requerimientos. Por lo tanto, los requerimientos del sistema no sólo detallan los servicios o las características que se requieren del mismo, sino también especifican la funcionalidad necesaria para asegurar que estos servicios y características se entreguen de manera adecuada.

#### Requerimientos funcionales.

Los requerimientos funcionales para un sistema refieren lo que el sistema debe hacer. Tales requerimientos dependen del tipo de software que se esté desarrollando, de los usuarios esperados del software y del enfoque general que adopta la organización cuando se escriben los requerimientos. Al expresarse como requerimientos del usuario, los requerimientos funcionales se describen por lo general de forma abstracta que entiendan los usuarios del sistema. Sin embargo, requerimientos funcionales más específicos del sistema detallan las funciones del sistema, sus entradas y salidas, sus excepciones, etcétera.

Los requerimientos funcionales del sistema varían desde requerimientos generales que cubren lo que tiene que hacer el sistema, hasta requerimientos muy específicos que reflejan maneras locales de trabajar o los sistemas existentes de una organización. Por ejemplo, veamos algunos casos de requerimientos funcionales para el sistema MHC-PMS, que se usan para mantener información de pacientes que reciben tratamiento por problemas de salud mental:

1. Un usuario podrá buscar en todas las clínicas las listas de citas.
2. El sistema elaborará diariamente, para cada clínica, una lista de pacientes que se espera que asistan a cita ese día.
3. Cada miembro del personal que usa el sistema debe identificarse de manera individual con su número de ocho dígitos.

Estos requerimientos funcionales del usuario definen las actividades específicas que debe proporcionar el sistema. Se tomaron del documento de requerimientos del usuario y muestran que los requerimientos funcionales pueden escribirse con diferentes niveles de detalle (contraste entre los requerimientos 1 y 3).

La inexactitud en la especificación de requerimientos causa muchos problemas en la ingeniería de software. Es natural que un desarrollador de sistemas interprete un requerimiento ambiguo de forma que simplifique su implementación. Sin embargo, con frecuencia, esto no es lo que desea el cliente. Tienen que establecerse nuevos requerimientos y efectuar cambios al sistema. Desde luego, esto aplaza la entrega del sistema y aumenta los costos.

Es el caso del primer ejemplo de requerimiento para el MHC-PMS que establece que un usuario podrá buscar las listas de citas en todas las clínicas. El motivo para este requerimiento es que los pacientes con problemas de salud mental en ocasiones están confundidos. Quizá tengan una cita en una clínica y en realidad acudan a una diferente. De ahí que, si tienen una cita, se registrará que asistieron, sin importar la clínica.

Los miembros del personal médico que especifican esto quizás esperen que “buscar” significa que, dado el nombre de un paciente, el sistema busca dicho nombre en las citas de todas las clínicas. Sin embargo, esto no es claro en el requerimiento. Los desarrolladores del sistema pueden interpretar el requerimiento de forma diferente e implementar una búsqueda, de tal modo que el usuario deba elegir una clínica y luego realizar la búsqueda. Evidentemente, esto implicará más entradas del usuario y tomará más tiempo.

En principio, la especificación de los requerimientos funcionales de un sistema debe ser completa y consistente. Completa significa que deben definirse todos los servicios requeridos por el usuario. Consistencia quiere decir que los requerimientos tienen que evitar definiciones contradictorias. En la práctica, para sistemas complejos grandes, es casi imposible lograr la consistencia y la totalidad de los requerimientos. Una causa para ello es la facilidad con que se cometen errores y omisiones al escribir especificaciones para sistemas complejos. Otra es que hay muchos participantes en un sistema grande. Un participante (“stakeholder”) es un individuo o una función que se ve afectado de alguna forma por el sistema y que tienen algún interés en él. Los participantes tienen diferentes necesidades, pero con frecuencia son inconsistentes. Tales inconsistencias tal vez no sean evidentes cuando se especifican por primera vez los requerimientos, de modo que en la especificación se incluyen requerimientos inconsistentes. Los problemas suelen surgir sólo después de un análisis en profundidad o después de que se entregó el sistema al cliente.

### Requerimientos no funcionales.

Los requerimientos no funcionales, como indica su nombre, son requerimientos que no se relacionan directamente con los servicios específicos que el sistema entrega a sus usuarios. Pueden relacionarse con propiedades emergentes del sistema, como fiabilidad, tiempo de respuesta y uso de almacenamiento. De forma alternativa, pueden definir restricciones sobre la implementación del sistema, como las capacidades de los dispositivos E/S o las representaciones de datos usados en las interfaces con otros sistemas.

Los requerimientos no funcionales, como el rendimiento, la seguridad o la disponibilidad, especifican o restringen por lo general características del sistema como un todo. Los requerimientos no funcionales a menudo son más significativos que los requerimientos funcionales individuales. Es común que los usuarios del sistema encuentren formas para trabajar en torno a una función del sistema que realmente no cubre sus necesidades. No obstante, el fracaso para cubrir los requerimientos no funcionales haría que todo el sistema fuera inútil. Por

ejemplo, si un sistema de aeronave no cubre sus requerimientos de fiabilidad, no será certificado para su operación como dispositivo seguro; si un sistema de control embebido fracasa para cubrir sus requerimientos de rendimiento, no operarán correctamente las funciones de control.

Aunque es posible identificar con regularidad cuáles componentes de sistema implementan requerimientos funcionales específicos (por ejemplo, hay componentes de formateo que implementan requerimientos de informe), por lo general es más difícil relacionar componentes con requerimientos no funcionales. La implementación de dichos requerimientos puede propagarse a lo largo del sistema. Para esto existen dos razones:

1. Los requerimientos no funcionales afectan más la arquitectura global de un sistema que los componentes individuales. Por ejemplo, para garantizar que se cumplan los requerimientos de rendimiento, quizá se deba organizar el sistema para minimizar las comunicaciones entre componentes.
2. Un requerimiento no funcional individual, como un requerimiento de seguridad, podría generar algunos requerimientos funcionales relacionados que definan nuevos servicios del sistema que se requieran. Además, también podría generar requerimientos que restrinjan los requerimientos ya existentes.

Los requerimientos no funcionales surgen a través de necesidades del usuario, debido a restricciones presupuestales, políticas de la organización, necesidad de interoperabilidad con otro software o sistemas de hardware, o factores externos como regulaciones de seguridad o legislación sobre privacidad. La Figura 1 es una clasificación de requerimientos no funcionales en término de características requeridas del software (requerimientos del producto), la organización que desarrolla el software (requerimientos de la organización) o de fuentes externas:

1. Requerimientos del producto. Estos requerimientos especifican o restringen el comportamiento del software. Los ejemplos incluyen requerimientos de rendimiento sobre qué tan rápido se debe ejecutar el sistema y cuánta memoria requiere (espacio), requerimientos de fiabilidad que establecen la tasa aceptable de fallas, requerimientos de seguridad y requerimientos de usabilidad.
2. Requerimientos de la organización. Son requerimientos de sistemas amplios, derivados de políticas y procedimientos en la organización del cliente y del desarrollador. Los ejemplos incluyen requerimientos del proceso operacional que definen cómo se usará el sistema, requerimientos del proceso de desarrollo que especifican el lenguaje de programación, estándares del entorno o el proceso de desarrollo a utilizar, y requerimientos ambientales que definen el entorno de operación del sistema.
3. Requerimientos externos. Este término cubre todos los requerimientos derivados de factores externos al sistema y su proceso de desarrollo. En ellos se incluyen requerimientos regulatorios que establecen lo que debe hacer el sistema para ser aprobado en su uso por un regulador, como sería un banco central; requerimientos legislativos que tienen que seguirse para garantizar que el sistema opere conforme a la ley, y requerimientos éticos que garanticen que el sistema será aceptable para sus usuarios y el público en general.

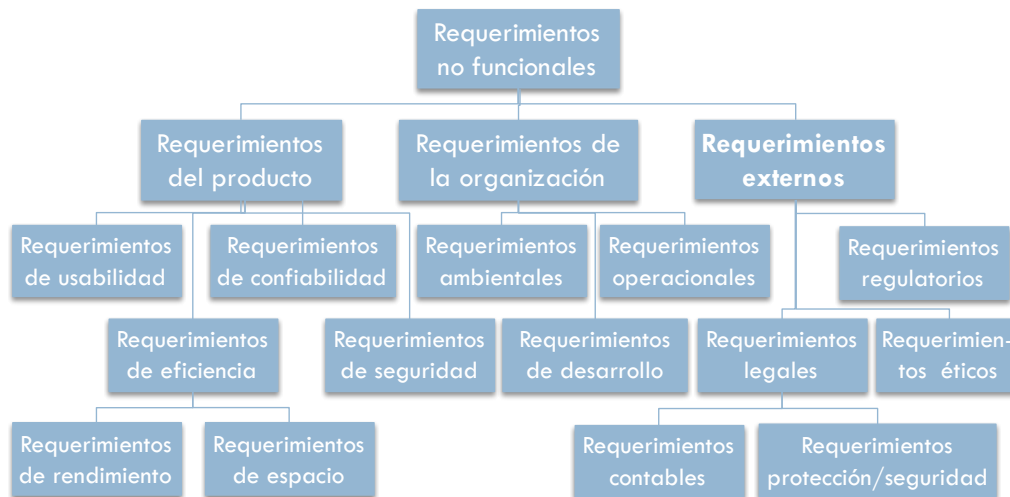


Figura 1. Tipos de requerimientos no funcionales (Sommerville, 2011).

A continuación, se muestran ejemplos de requerimientos del producto, de la organización y requerimientos externos tomados del MHC-PMS (para mayor detalle del caso de estudio, ver capítulo 1 de Sommerville, 2011).

#### REQUERIMIENTO DEL PRODUCTO

El MHC-PMS estará disponible en todas las clínicas durante las horas de trabajo normales (lunes a viernes, de 8:30 a 17:30). En cualquier día, los tiempos muertos dentro de las horas laborales normales no rebasarán los cinco segundos.

El requerimiento del producto es un requerimiento de disponibilidad que define cuándo estará disponible el sistema y el tiempo muerto permitido cada día. No dice algo sobre la funcionalidad del MHC-PMS e identifica con claridad una restricción que deben considerar los diseñadores del sistema.

#### REQUERIMIENTOS DE LA ORGANIZACIÓN

Los usuarios del sistema MHC-PMS se acreditarán a sí mismos con el uso de la tarjeta de identidad de la autoridad sanitaria.

El requerimiento de la organización especifica cómo se autentican los usuarios en el sistema. La autoridad sanitaria que opera el sistema se mueve hacia un procedimiento de autenticación estándar para cualquier software donde, en vez de que los usuarios tengan un nombre de conexión (login), pasan su tarjeta de identidad por un lector para identificarse a sí mismos.

#### REQUERIMIENTOS EXTERNOS

Como establece la HStan-03-2006-priv, el sistema implementará provisiones para la privacidad del paciente.

El requerimiento externo se deriva de la necesidad de que el sistema esté conforme con la legislación de privacidad de los datos. Evidentemente, la privacidad de datos es un asunto muy

importante en los sistemas de atención a la salud, y el requerimiento especifica que el sistema debe desarrollarse conforme a un estándar de privacidad nacional.

Un problema común con requerimientos no funcionales es que los usuarios o clientes con frecuencia proponen estos requerimientos como metas generales, como facilidad de uso, capacidad de que el sistema se recupere de fallas, o rapidez de respuesta al usuario. Las metas establecen buenas intenciones; no obstante, ocasionan problemas a los desarrolladores del sistema, pues dejan espacio para la interpretación y la disputa posterior una vez que se entregue el sistema. Por ejemplo, la siguiente meta del sistema es típica de cómo un administrador expresa los requerimientos de usabilidad:

Para el personal médico debe ser fácil usar el sistema, y este último debe organizarse de tal forma que minimice los errores del usuario.

Lo anterior se escribió para mostrar cómo podría expresarse la meta como un requerimiento no funcional “comprobable”. Aun cuando es imposible comprobar de manera objetiva la meta del sistema, en la siguiente descripción se puede incluir, al menos, la instrumentación de software para contar los errores cometidos por los usuarios cuando prueban el sistema:

Después de cuatro horas de capacitación, el personal médico usará todas las funciones del sistema. Después de esta capacitación, los usuarios experimentados no deberán superar el promedio de dos errores cometidos por hora de uso del sistema.

Siempre que sea posible, se deberán escribir de manera cuantitativa los requerimientos no funcionales, de manera que puedan ponerse objetivamente a prueba. La Tabla 1 muestra las métricas que se utilizan para especificar propiedades no funcionales del sistema.

Es posible medir dichas características cuando el sistema se pone a prueba para comprobar si éste cumple o no cumple con sus requerimientos no funcionales. En la práctica, los usuarios de un sistema suelen encontrar difícil traducir sus metas en requerimientos mensurables. Para algunas metas, como la mantenibilidad, no hay métricas para usarse. En otros casos, incluso cuando sea posible la especificación cuantitativa, los clientes no logran relacionar sus necesidades con dichas especificaciones. No comprenden qué significa algún número que define la fiabilidad requerida (por así decirlo), en términos de su experiencia cotidiana con los sistemas de cómputo. Más aún, el costo por verificar objetivamente los requerimientos no funcionales mensurables suele ser muy elevado, y los clientes que pagan por el sistema quizá piensen que dichos costos no están justificados.

Los requerimientos no funcionales entran a menudo en conflicto e interactúan con otros requerimientos funcionales o no funcionales. Por ejemplo, el requerimiento de autenticación visto recientemente requiere, indiscutiblemente, la instalación de un lector de tarjetas en cada computadora unida al sistema. Sin embargo, podría haber otro requerimiento que solicite acceso móvil al sistema desde las computadoras portátiles de médicos o enfermeras. Por lo general, las computadoras portátiles no están equipadas con lectores de tarjeta, de modo que, ante tales circunstancias, probablemente deba permitirse algún método de autenticación alternativo.

Propiedad	Medida
<b>Rapidez</b>	Transacciones/segundo procesadas Tiempo de respuesta usuario/evento Tiempo de regeneración de pantalla
<b>Tamaño</b>	Mbytes Número de chips ROM
<b>Facilidad de Uso</b>	Tiempo de capacitación Número de cuadros de ayuda
<b>Fiabilidad</b>	Tiempo medio para falla Probabilidad de indisponibilidad Tasa de ocurrencia de falla Disponibilidad
<b>Robustez</b>	Tiempo de reinicio después de falla Porcentaje de eventos que causan falla Probabilidad de corrupción de datos en falla
<b>Portabilidad</b>	Porcentaje de enunciados dependientes de objetivo Número de sistemas objetivo

Tabla 1. Métricas para especificar requerimientos no funcionales.

En la práctica, en el documento de requerimientos, resulta difícil separar los requerimientos funcionales de los no funcionales. Si los requerimientos no funcionales se expresan por separado de los requerimientos funcionales, las relaciones entre ambos serían difíciles de entender. No obstante, se deben destacar de manera explícita los requerimientos que están claramente relacionados con las propiedades emergentes del sistema, como el rendimiento o la fiabilidad. Esto se logra al ponerlos en una sección separada del documento de requerimientos o al distinguirlos, en alguna forma, de otros requerimientos del sistema.

Los requerimientos no funcionales, como los requerimientos de fiabilidad, protección y confidencialidad, son en particular importantes para los sistemas críticos y existen métodos particulares para su especificación (Sommerville, 2011).

## 2.2. El Documento de Requerimientos de Software.

El documento de requerimientos de software (o especificación de requerimientos de software o SRS) especifica lo que deben implementar los desarrolladores del sistema. Incluye tanto los requerimientos del usuario para un sistema, como una especificación detallada de los requerimientos del sistema. En ocasiones, los requerimientos del usuario y del sistema se integran en una sola descripción. En otros casos, los requerimientos del usuario se definen en una introducción a la especificación de requerimientos del sistema. Si hay un gran número de requerimientos, los requerimientos del sistema detallados podrían presentarse en un documento aparte.

Son esenciales los documentos de requerimientos cuando un contratista externo diseña el sistema de software. Sin embargo, los métodos de desarrollo ágiles argumentan que los requerimientos cambian tan rápidamente que un documento de requerimientos se vuelve



obsoleto tan pronto como se escribe, así que el esfuerzo se desperdicia en gran medida. En lugar de un documento formal, los enfoques como la programación extrema recopilan de manera incremental requerimientos del usuario y los escriben como historias de usuario. De esa manera, el usuario da prioridad a los requerimientos para su implementación en el siguiente incremento del sistema. Este enfoque es adecuado para sistemas empresariales, donde los requerimientos son inestables. Sin embargo, resulta útil escribir un breve documento de apoyo que defina los requerimientos de la empresa y los requerimientos de confiabilidad para el sistema; es fácil olvidar los requerimientos que se aplican al sistema como un todo, cuando uno se enfoca en los requerimientos funcionales para la siguiente liberación del sistema.

El documento de requerimientos tiene un conjunto variado de usuarios, desde el administrador ejecutivo de la organización que paga por el sistema, hasta los ingenieros responsables del desarrollo del software y los testers. La Tabla 2 presenta posibles usuarios del documento y cómo ellos lo utilizan.

Usuarios	Uso del documento de requerimientos
<b>Clientes del sistema</b>	Especifican los requerimientos y los leen para comprobar que cubren sus necesidades. Los clientes especifican los cambios a los requerimientos.
<b>Administradores</b>	Usan el documento de requerimientos para planear una cotización para el sistema y el proceso de desarrollo del sistema.
<b>Ingenieros del sistema</b>	Usan los requerimientos para entender qué sistema debe desarrollarse.
<b>Ingenieros de prueba de sistema (testers)</b>	Usan los requerimientos para desarrollar pruebas de validación para el sistema.
<b>Ingenieros de mantenimiento del sistema</b>	Usan los requerimientos para comprender el sistema y las relaciones entre sus componentes.

Tabla 2. Usuarios de un documento de requerimientos

La diversidad de posibles usuarios significa que el documento de requerimientos debe ser un compromiso entre la comunicación de los requerimientos a los clientes, la definición de los requerimientos con detalle preciso para desarrolladores y examinadores, y la inclusión de información sobre la posible evolución del sistema. La información de cambios anticipados ayuda tanto a los diseñadores del sistema a evitar decisiones de diseño restrictivas, como a los ingenieros de mantenimiento del sistema que deben adaptar el sistema a los nuevos requerimientos.

El nivel de detalle que se incluya en un documento de requerimientos depende del tipo de sistema a diseñar y el proceso de desarrollo utilizado. Los sistemas críticos necesitan tener requerimientos detallados porque la seguridad y la protección también deben analizarse de forma pormenorizada. Cuando el sistema lo desarrolla una compañía independiente (por ejemplo, mediante la subcontratación), deben detallarse y precisarse las especificaciones del sistema. Si se utiliza un proceso de desarrollo iterativo interno, entonces el documento de requerimientos suele ser mucho menos detallado y cualquier ambigüedad puede resolverse durante el desarrollo del sistema.

En la Tabla 3 se indica una posible organización para un documento de requerimientos basada en un estándar de la ISO/IEC/IEEE para documentos de requerimientos (ISO/IEC/IEEE 29148, 2011).

Sección	Descripción
<b>Identificación</b>	
Título	
Nota de revisión	El título y la nota de revisión permiten identificar de manera unívoca el documento. La nota de revisión incluye el nombre del proyecto, el número de versión del documento, y una lista de números de versiones y fechas de versiones previas al documento.
<b>Parte inicial</b>	
Tabla de contenidos	
Lista de figuras	
Lista de tablas	
<b>Introducción</b>	
Propósito	Describe la necesidad para el sistema.
Alcance	Describe el alcance del proyecto: especifica el nombre del producto de software, explica qué hará el software, describe la aplicación del software, sus beneficios, objetivos y metas.
Introducción del producto	
Perspectiva del producto	Define las relaciones del producto con otros productos relacionados. Es posible emplear un diagrama de bloques para ilustrar los componentes del sistema que lo engloba, las interconexiones e interfaces externas.
Funciones del producto	Detalla brevemente las funciones del sistema.
Características del usuario	Describe las características generales de los grupos de usuarios del producto, incluyendo aquellas características que puedan influir en la usabilidad, tales como nivel de educación, experiencia, discapacidades, y conocimiento técnico.
Limitaciones	Provee descripción de ciertas razones que limitarán las opciones posibles, tales como: leyes, limitaciones de hardware, interfaces a otras aplicaciones, paralelismo en operaciones, auditoría, control, requerimientos de lenguajes de orden superior, requerimientos de calidad, criticidad de la aplicación, seguridad.
Definiciones	Define los términos técnicos usados en el documento. No debe hacer conjeturas sobre la experiencia o la habilidad del lector.
Referencias	Provee una lista completa de los documentos referenciados. Identifica cada documento por el título, número de reporte (si es aplicable), fecha, organización que realizó la publicación, fuente desde donde puede ser obtenida.
<b>Requerimientos específicos</b>	
Interfaces externas	Especifica todos los requerimientos de software a un nivel de detalle suficiente que permita a los diseñadores diseñar un software que satisfaga los requerimientos, y a los testers probar el software.
Funciones	Define todas las entradas y salidas del sistema. La descripción debe complementar la información incluida en perspectiva del producto y no debería repetir la misma.
Requerimientos de usabilidad	Define las acciones fundamentales que tendrán lugar en el software. Incluye: controles en las entradas, secuencia exacta de las operaciones, respuestas a situaciones anormales, efecto de parámetros, relaciones de las salidas con las entradas.
Requerimientos de desempeño	Define los requerimientos de usabilidad, calidad en uso.
Requerimientos de base de datos lógica	Especifica requerimientos estáticos y dinámicos sobre performance. Requerimientos estáticos tales como el número de terminales a soportar, número de usuarios en simultáneo, cantidad y tipo de información a manejar. Requerimientos dinámicos pueden incluir al número de transacciones, tareas y cantidad de datos a ser procesados en un cierto periodo de tiempo, en condiciones de trabajo normales y de sobrecarga.
Restricciones de diseño	Especifica requerimientos lógicos para información que será almacenada en base de datos. Esto incluye tipo de información, frecuencia de uso, capacidad de acceso, restricciones de integridad.
Atributos del sistema de software	Especifica restricciones en el diseño del sistema impuestas por estándares externos, requerimientos regulatorios, o limitaciones del proyecto.
Información de soporte	Especifica atributos de software requeridos: confiabilidad, disponibilidad, seguridad, mantenibilidad, portabilidad.
	Contiene información adicional sobre ejemplos de entradas/salidas, descripciones de estudios de costo, resultados de encuestas a usuarios, información de soporte para la lectura de la SRS.
<b>Verificación</b>	
	Provee la metodología y los métodos planificados para calificar al software. Generalmente su estructura es la misma que la de Requerimientos específicos.
<b>Apéndices</b>	
Suposiciones y dependencias	Lista los factores que afectan los requerimientos especificados en la SRS.
Acrónimos y abreviaturas	Acrónimos y abreviaciones empleados en el documento.

Tabla 3. Estructura de un documento de requerimientos, adaptada de ISO/IEC/IEEE (2011).

Este estándar es genérico y se adapta a usos específicos. Naturalmente, la información que se incluya en un documento de requerimientos depende del tipo de software que se va a desarrollar y del enfoque para el desarrollo que se use. Si se adopta un enfoque evolutivo para un producto de software (por ejemplo), el documento de requerimientos dejará fuera muchos de los capítulos detallados que se sugirieron anteriormente. El enfoque estará en especificar los requerimientos del usuario y los requerimientos no funcionales de alto nivel del sistema. En este caso, diseñadores y programadores usan su criterio para decidir cómo cubrir los requerimientos establecidos del usuario para el sistema.

Sin embargo, cuando el software sea parte de un proyecto de sistema grande que incluya la interacción de sistemas de hardware y software será necesario, por lo general, definir los requerimientos a un nivel detallado. Esto significa que es probable que los documentos de requerimientos sean muy largos y deban incluir la mayoría, si no es que todas, las secciones que se detallan en la Tabla 3. Para documentos extensos, es muy importante incluir una tabla de contenido global y un índice del documento, de manera que los lectores encuentren con facilidad la información que necesitan.

### 2.3. Especificación de requerimientos.

La especificación de requerimientos es el proceso de escribir, en un documento de requerimientos, los requerimientos del usuario y del sistema. De manera ideal, los requerimientos del usuario y del sistema deben ser claros, sin ambigüedades, fáciles de entender, completos, consistentes, y verificables (Sommerville, 2011). Esto en la práctica es difícil de lograr, pues los participantes interpretan los requerimientos de formas diferentes y con frecuencia en los requerimientos hay conflictos e inconsistencias inherentes.

Los requerimientos del usuario para un sistema deben describir los requerimientos funcionales y no funcionales, de forma que sean comprensibles para los usuarios del sistema que no cuentan con un conocimiento técnico detallado. De manera ideal, deberían especificar sólo el comportamiento externo del sistema. El documento de requerimientos no debe incluir detalles de la arquitectura o el diseño del sistema. En consecuencia, al escribir los requerimientos del usuario no se debe emplear jerga de software, anotaciones estructuradas o formales. Se deben escribir los requerimientos del usuario en lenguaje natural, con tablas y formas sencillas, así como diagramas intuitivos.

Los requerimientos del sistema son versiones extendidas de los requerimientos del usuario que los ingenieros de software usan como punto de partida para el diseño del sistema. Añaden detalles y explican cómo el sistema debe brindar los requerimientos del usuario. Se pueden usar como parte del contrato para la implementación del sistema y, por lo tanto, deben ser una especificación completa y detallada de todo el sistema. Idealmente, los requerimientos del sistema deben describir de manera simple el comportamiento externo del sistema y sus restricciones operacionales. No tienen que ocuparse de cómo se diseña o implementa el sistema. Sin embargo, al nivel de detalle requerido para especificar por completo un sistema de software complejo, es prácticamente imposible excluir toda la información de diseño. Para ello existen varias razones:

1. Tal vez se tenga que diseñar una arquitectura inicial del sistema para ayudar a estructurar la especificación de requerimientos. Los requerimientos del sistema se organizan de acuerdo con los diferentes subsistemas que constituyen el sistema. Esta

definición arquitectónica es esencial si usted quiere reutilizar componentes de software al implementar el sistema.

2. En la mayoría de los casos, los sistemas deben interoperar con los sistemas existentes, lo cual restringe el diseño e impone requerimientos sobre el nuevo sistema.
3. Quizá sea necesario el uso de una arquitectura específica para cubrir los requerimientos no funcionales. Un regulador externo, que precise certificar que dicho sistema es seguro, puede especificar que se utilice un diseño arquitectónico ya avalado.

Los requerimientos del usuario se escriben casi siempre en lenguaje natural, complementado con diagramas y tablas adecuados en el documento de requerimientos. Los requerimientos del sistema se escriben también en lenguaje natural, pero de igual modo se utilizan otras notaciones basadas en formularios, modelos gráficos del sistema o modelos matemáticos del sistema. La Tabla 4 resume las posibles anotaciones que podrían usarse para escribir requerimientos del sistema.

Notación	Descripción
<b>Enunciados en lenguaje natural</b>	Los requerimientos se escriben al usar enunciados numerados en lenguaje natural. Cada enunciado debe expresar un requerimiento.
<b>Lenguaje natural estructurado</b>	Los requerimientos se escriben en lenguaje natural en una forma o plantilla estándar. Cada campo ofrece información de un aspecto del requerimiento.
<b>Lenguajes de descripción de diseño</b>	Este enfoque usa un lenguaje como un lenguaje de programación, pero con características más abstractas para especificar los requerimientos al definir un modelo operacional del sistema. Aunque en la actualidad este enfoque se usa raras veces, aún tiene utilidad para especificaciones de interfaz.
<b>Anotaciones gráficas</b>	Los modelos gráficos, complementados con anotaciones de texto, sirven para definir los requerimientos funcionales del sistema; los casos de uso del UML y los diagramas de secuencia se emplean de forma común.
<b>Especificaciones matemáticas</b>	Dichas anotaciones se basan en conceptos matemáticos como máquinas o conjuntos de estado finito. Aunque tales especificaciones sin ambigüedades pueden reducir la imprecisión en un documento de requerimientos, la mayoría de los clientes no comprenden una especificación formal. No pueden comprobar que representa lo que quieren y por ello tienen reticencia para aceptarlo como un contrato de sistema.

Tabla 4. Formas de escribir una especificación de requerimientos

Los modelos gráficos son más útiles cuando es necesario mostrar cómo cambia un estado o al describir una secuencia de acciones. Los diagramas de secuencia UML y los diagramas de estado exponen la secuencia de acciones que ocurren en respuesta a cierto mensaje o evento. En ocasiones, se usan especificaciones matemáticas formales con la finalidad de describir los requerimientos para sistemas de protección o seguridad críticos, aunque rara vez se usan en otras circunstancias.

### Especificación en lenguaje natural.

Desde los inicios de la ingeniería de software, el lenguaje natural se usa para escribir los requerimientos de software. Es expresivo, intuitivo y universal. También es potencialmente vago, ambiguo y su significado depende de los antecedentes del lector. Como resultado, hay muchas propuestas para formas alternativas de escribir los requerimientos. Sin embargo, ninguna se ha adoptado de manera amplia, por lo que el lenguaje natural seguirá siendo la forma más usada para especificar los requerimientos del sistema y del software.

Para minimizar la interpretación errónea al escribir los requerimientos en lenguaje natural, se recomienda seguir algunos lineamientos sencillos:

1. Elaborar un formato estándar y asegurarse de que todas las definiciones de requerimientos se adhieran a dicho formato. Al estandarizar el formato es menos probable cometer omisiones y más sencillo comprobar los requerimientos. El formato que usa Sommerville (2011) expresa el requerimiento en una sola oración. A cada requerimiento de usuario se asocia un enunciado de razones para explicar por qué se propuso el requerimiento. Las razones también pueden incluir información sobre quién planteó el requerimiento (la fuente del requerimiento), de modo que se conozca a quién consultar en caso de que cambie el requerimiento.
2. Utilizar el lenguaje de manera clara para distinguir entre requerimientos obligatorios y deseables. Los primeros son requerimientos que el sistema debe soportar y, por lo general, se escriben en futuro “debe ser”. En tanto que los requerimientos deseables no son necesarios y se escriben en tiempo pospretérito o como condicional “debería ser”.
3. Use texto resaltado (negrita, cursiva o color) para seleccionar partes clave del requerimiento.
4. No suponer que los lectores entienden el lenguaje técnico de la ingeniería de software. Es fácil que se malinterpreten palabras como “arquitectura” y “módulo”. Por lo tanto, se debe evitar el uso de jerga, abreviaturas y acrónimos.
5. Siempre que sea posible, asociar un motivo (“rationale”) con cada requerimiento de usuario. El motivo debe explicar por qué se incluyó el requerimiento. Es particularmente útil cuando los requerimientos cambian, pues ayuda a decidir cuáles cambios serían indeseables.

En ISO/IEC/IEEE 29148 (2011) se plantea que un requerimiento es una oración que traduce o expresa una necesidad, junto a sus condiciones y restricciones asociadas. Esta oración usualmente se escribe en lenguaje natural, por lo que debe poseer un sujeto, un verbo y un complemento. Un requerimiento debe explicitar el sujeto del requerimiento (por ejemplo, el sistema, el software) y qué debe ser realizado (por ejemplo, proveer un campo para). A continuación, se presentan tres ejemplos de sintaxis para expresar requerimientos:

[Condición][Sujeto][Acción][Objeto][Restricción]

Ejemplo: Cuando se recibe la señal x [Condición], el sistema [Sujeto] deberá activar [Acción] el bit de la señal x recibida [Objeto] dentro de los 2 segundos [Restricción].

[Condición][Acción o Restricción][Valor]

[Sujeto][Acción][Valor]

Las condiciones son atributos cualitativos o cuantitativos medibles que son estipulados para un requerimiento. Además, califican un requerimiento que es necesario (correcto) y proporcionan atributos que permiten que un requerimiento sea formulado y declarado de una manera que pueda ser validado y verificado. Las condiciones pueden limitar las opciones a un diseñador. Es importante transformar las necesidades en requerimientos sin imponer límites innecesarios al espacio de la solución.

Las restricciones restringen el diseño o la implementación. Las restricciones pueden aplicarse a todos los requerimientos, pueden especificarse en una relación con un requerimiento o conjunto de requerimientos específicos, o pueden identificarse como requerimientos independientes (es decir, sin limitar ningún requerimiento específico).

Cuando se especifiquen requerimientos de forma textual, se debe evitar el empleo de los siguientes términos:

- Superlativos, tales como el mejor, el máximo.
- Lenguaje subjetivo (depende del modo de ser visto), tales como amigable al usuario, fácil de usar.
- Pronombres imprecisos (o vagos), tales como este, aquella, sin que quede claro a quién o a qué hacen referencia.
- Adverbios y adjetivos ambiguos, tales como significativo, mínimo, casi siempre.
- Términos no verificables, abiertos, no concretos, tales como proveer soporte, pero no limitado a, como mínimo.
- Frases comparativas, tales como mejor que, mayor calidad que.

#### 2.4. Guía para la Escritura de Requerimientos, propuesta de INCOSE

En esta sección se presentan los conceptos esenciales de la propuesta (2017) para la especificación de requerimientos en lenguaje natural.

Al describir el desarrollo de sistemas, se debe poder distinguir entre "necesidades" y "requerimientos". Generalmente, las necesidades se consideran expectativas expresadas en el lenguaje de las personas a nivel de gestión empresarial o de las partes interesadas a nivel de operaciones comerciales. Las sentencias de requerimientos (requerimientos) se generan a partir de las necesidades a través de un proceso de análisis de requerimientos; puede haber más de un requerimiento definido para cualquier necesidad. Los requerimientos son declaraciones que se pueden utilizar para: verificar que el sistema que se está desarrollando cumple con los requerimientos que impulsaron las actividades de diseño y construcción / código, y validar que se satisfacen las necesidades y expectativas de los interesados a partir de las cuales se desarrollaron los requerimientos. En la Figura 2 se describen los conceptos empleados en la guía INCOSE.

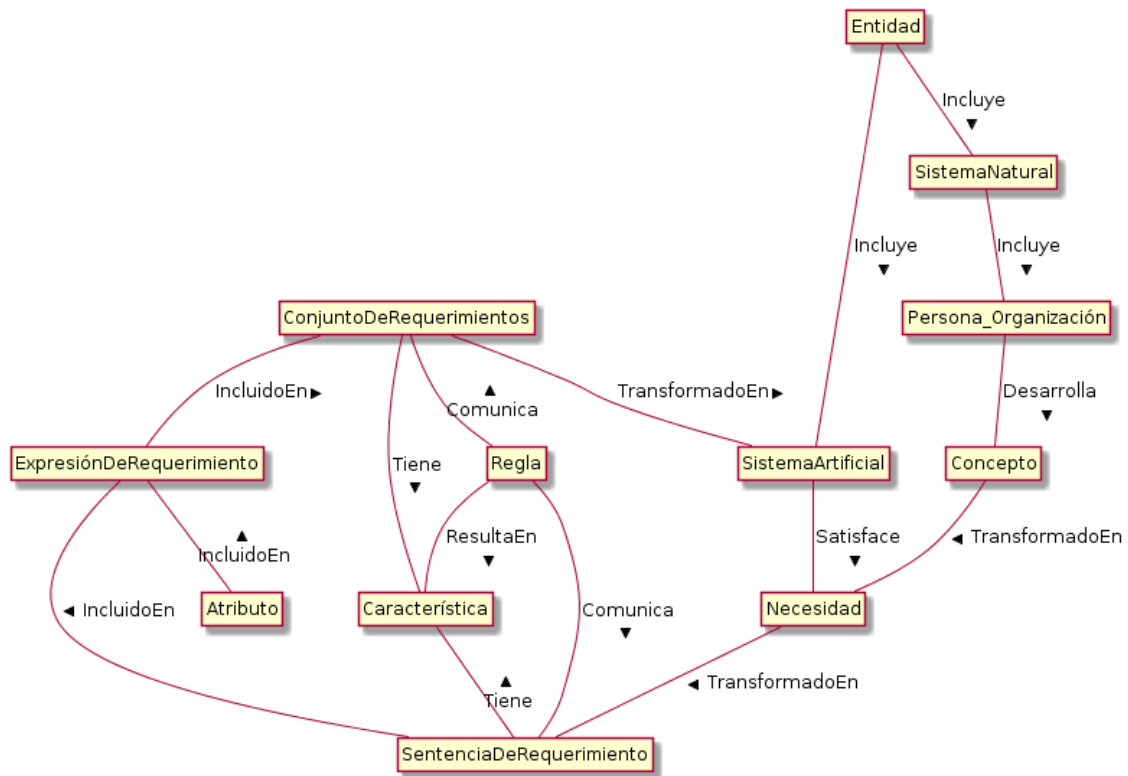


Figura 2. Conceptos y Relaciones empleados en la guía de escritura de requerimientos INCOSE.

Una **entidad** es una cosa a la que se refiere una necesidad o requerimiento, representa una empresa, unidad de negocio, sistema o elemento del sistema (que podría ser un producto, proceso, ser humano u organización).

Una **necesidad** es el resultado de una transformación de uno o más **conceptos** en una expectativa acordada, para que una entidad realice alguna función o posea cierta calidad (dentro de las limitaciones especificadas).

Desarrollar requerimientos es más que un ejercicio en escritura. El desarrollo de requerimientos es una actividad de ingeniería de sistemas donde el ingeniero o analista, a través del análisis, determina específicamente lo que el sistema debe hacer para satisfacer las **necesidades** de la **entidad**.

Una **declaración de requerimiento (sentencia de requerimiento)** es el resultado de una transformación de una o más **necesidades** en una obligación acordada para que una **entidad** realice alguna función o posea cierta calidad (dentro de las restricciones especificadas).

Un requerimiento es más que una simple declaración de requerimientos bien redactada que se escribe de manera muy sucinta (breve, concisa y precisa) en un formato estándar (siguiendo un patrón de requerimientos específico) que tiene las características definidas en esta guía. La expresión de requerimiento completo (**conjunto de requerimientos**) incluye atributos asociados que ayudan en el desarrollo y la gestión del requerimiento y el conjunto de requerimientos.

Una **expresión de requerimiento** se define como una **declaración de requerimiento** con un conjunto de **atributos** asociados

Aunque cada **expresión de requerimiento** individual es importante, es el **conjunto de requerimientos** que describirá a la **entidad** completa y se acordará como una obligación contractual; esa descripción está contenida en una especificación de requerimientos o documento para la entidad.

Un **conjunto de requerimientos** es un conjunto estructurado de **expresiones de requerimientos** acordadas para el sistema y sus interfaces externas documentadas en una Especificación de Requerimientos (Documento) del sistema.

Entonces, un **conjunto de requerimientos** es el resultado de una transformación de las **necesidades** y expectativas de los interesados en el sistema que está siendo desarrollado.

El proceso de creación de un **conjunto de requerimientos** implica:

- Analizar las **necesidades** y expectativas de los interesados para obtener los elementos necesarios para ser incluidos en el **conjunto de requerimientos**;
- Seleccionar un formato para la **expresión de requerimientos** y una organización del **conjunto de requerimientos**;
- Identificar las **características** del resultado deseado contra las pautas y **reglas** de la organización por las cuales se deben escribir las **declaraciones de requerimientos** y el **conjunto de requerimientos**; y
- Transformar las **necesidades** y expectativas de las partes interesadas en un **conjunto de requerimientos** que comunique y represente sin ambigüedad estas **necesidades** y expectativas de las partes interesadas.

En este contexto, la **verificación de requerimientos** confirma, mediante inspección, que los **requerimientos** contienen los elementos necesarios y poseen las **características** de un requerimiento bien formado, y que el requerimiento establece y se ajusta a las **reglas** establecidas en las pautas de desarrollo de requerimientos de la organización. La **validación de requerimientos** confirma, mediante inspección y análisis, que el **conjunto de requerimientos** resultante cumple con la intención de las **necesidades** de las partes interesadas de las cuales los requerimientos y el conjunto de requerimientos se descompusieron o derivaron. Por lo tanto, una declaración de requerimientos y un conjunto de requerimientos se confirman por verificación y validación.

- **Verificación** del requerimiento: el proceso para garantizar que el requerimiento cumpla con las reglas y características definidas para redactar un buen requerimiento. La atención se centra en la redacción y la estructura del requerimiento. “¿El requerimiento está redactado o estructurado correctamente de acuerdo con los estándares, pautas, reglas y listas de verificación de la organización?”.
- **Validación** de requerimientos: confirmación de que los requerimientos y el conjunto de requerimientos es una transformación acordada que comunica claramente las necesidades y expectativas de las partes interesadas en un lenguaje entendido por los desarrolladores. El foco está en el mensaje que los requerimientos y el conjunto de requerimientos están comunicando. “¿Los requerimientos establecidos comunican de manera clara y correcta las expectativas y necesidades de las partes interesadas?” “¿Estamos haciendo lo correcto?” O “¿Estamos construyendo lo correcto [según lo definido por el conjunto de requerimientos]?”

### Características de requerimientos individuales.

En esta sección se definen las características que deben poseer los requerimientos individuales. Se describen las características incluidas en la guía para la escritura de requerimientos propuesta por INCOSE (2017), resumidas en la Tabla 5.



Característica	Definición
<b>C1 - Necesario</b>	El requerimiento define una capacidad esencial, característica, restricción o factor de calidad. Si no se incluye en el conjunto de requerimientos, existirá una deficiencia en la capacidad o característica, que no puede cumplirse implementando otros requerimientos.
<b>C2 - Apropiado</b>	La intención específica y la cantidad de detalle del requerimiento es apropiado para el nivel (nivel de abstracción) de la entidad a la que se refiere.
<b>C3 - No ambiguo</b>	El requerimiento se establece de tal manera que se puede interpretar de una sola manera.
<b>C4 - Completo</b>	El requerimiento describe de manera suficiente la capacidad, característica, restricción o factor de calidad necesarios para satisfacer la necesidad de la entidad sin necesidad de otra información para comprender el requerimiento.
<b>C5- Singular (Atómico)</b>	El requerimiento debe indicar una única capacidad, característica, restricción o factor de calidad.
<b>C6 - Factible</b>	El requerimiento se puede cumplir dentro de las restricciones de la entidad (por ejemplo, costo, cronograma, técnico, legal, ético, regulatorio) con un riesgo aceptable.
<b>C7 - Verificable</b>	El requerimiento está estructurado y redactado de manera tal que su realización o implementación pueda probarse (verificarse).
<b>C8 - Correcto</b>	El requerimiento debe ser una representación precisa de la necesidad de la entidad a partir de la cual se transformó.
<b>C9 - Acorde a una forma</b>	Los requerimientos individuales deben cumplir con un patrón y estilo estándar aprobado para la escritura de los requerimientos.

Tabla 5. Características que deben reunir las especificaciones textuales de requerimientos.

### C1 - Necesario

El requerimiento define una capacidad esencial, característica, restricción o factor de calidad. Si no se incluye en el conjunto de requerimientos, existirá una deficiencia en la capacidad o característica, que no puede cumplirse implementando otros requerimientos.

Justificación:

La transformación debe dar como resultado un requerimiento que sea necesario para satisfacer una necesidad o un conjunto de necesidades para la entidad. La realización de cada requerimiento requiere recursos, esfuerzo y costo en forma de procesamiento, gestión y verificación. Los requerimientos innecesarios pueden conducir a un trabajo sin valor agregado, costos adicionales y riesgos innecesarios. Por lo tanto, solo los requerimientos necesarios deben incluirse en el conjunto de requerimientos. Una vez que se demuestra que cada requerimiento es necesario, el conjunto de requerimientos debe ser una solución suficiente para el conjunto de necesidades.

Guía de cómo alcanzarlo:

Un requerimiento debe poder rastrearse hasta una fuente que podría ser una o más necesidades de la entidad o un requerimiento principal asignado de nivel superior.

Un requerimiento no es necesario (no es necesario en el conjunto de requerimientos) si:

- el requerimiento puede eliminarse y el conjunto restante aún dará como resultado la entidad solicitada;
- la intención del requerimiento se cumplirá mediante la implementación de otros requerimientos; o
- el autor no puede comunicar la razón del requerimiento (es decir, no tiene una justificación válida).

Esta característica sólo se puede asegurar al revisar cada requerimiento en relación con la misión, las metas y los objetivos, así como los impulsores, las limitaciones, los conceptos y los escenarios definidos para la entidad. Si el requerimiento no se puede rastrear a una o más necesidades de la entidad (o requerimiento principal), no es necesario. La inclusión de la justificación y otros atributos, como el seguimiento (trazabilidad) a la fuente o al padre (requerimiento de nivel superior), para cada requerimiento también ayuda a comunicar la necesidad y la intención del requerimiento.

## C2 - Apropiado

La intención específica y la cantidad de detalle del requerimiento es apropiado para el nivel (nivel de abstracción) de la entidad a la que se refiere.

Justificación:

Se pueden imponer requerimientos a cualquier nivel; sin embargo, por regla general, un requerimiento debe expresarse a nivel de la entidad a la que se refiere. El requerimiento debe indicar lo que debe establecerse para el nivel de la entidad, no cómo debe cumplirse el requerimiento.

Guía de cómo alcanzarlo:

El requerimiento no debe ser más detallado o específico de lo necesario para el nivel en el que se establece. En particular, el tema del requerimiento debe ser apropiado para el nivel en el que vive el requerimiento. A menos que haya una buena razón, un sujeto / sustantivo de requerimiento debe referirse a la entidad al nivel del requerimiento (no más alto ni más bajo). El requerimiento debe evitar la imposición de restricciones innecesarias en el diseño arquitectónico en el nivel dado.

El objetivo es ser **independiente de la implementación**. Puede haber casos en los que exista una buena justificación para establecer restricciones sobre la implementación. Cuando esto sucede, se debe incluir la justificación para dejar en claro por qué es necesario establecer tales restricciones.

Una pregunta útil para formular sobre un requerimiento es “¿con qué propósito?” O “¿por qué?” Si el requerimiento se expresa en términos de implementación, la respuesta a esta pregunta puede ser el requerimiento real.

Es una buena práctica que, una vez que se escriba el siguiente nivel de requerimientos, el equipo haga un ejercicio de "nivelación", donde analicen cada requerimiento en el nivel superior y determinen si está en el nivel apropiado o si se debe bajar un nivel.

### C3 - No ambiguo

El requerimiento se establece de tal manera que se puede interpretar de una única manera.

#### Justificación:

Una declaración de requerimientos debe prestarse a una única interpretación de intención. Es difícil promulgar un acuerdo a menos que ambas partes tengan clara la obligación exacta. La ambigüedad conduce a múltiples interpretaciones de tal manera que las expectativas de las partes interesadas pueden no cumplirse.

La intención de un requerimiento debe ser entendida de la misma manera por el escritor, el diseñador y aquellos que realizan actividades de verificación. La ambigüedad lleva a interpretaciones de un requerimiento no previsto por el autor, y los problemas resultantes, incluyendo el retraso del proyecto e incluso tal vez litigios y pérdidas financieras.

#### Guía de cómo alcanzarlo:

Al escribir una declaración de requerimientos, pregunte si podría interpretarse, o no, de más de una manera. También pregunte si es verificable o no, es decir, si se establece de tal manera que se pueda obtener una prueba positiva basada en la redacción del requerimiento sin tener que interpretar el significado o hacer suposiciones sobre el significado. Además, es útil que las partes involucradas en la implementación de los requerimientos o la verificación y validación del sistema participen en el desarrollo, la revisión y la línea de base de los requerimientos. Cuando ven requerimientos que son ambiguos y su intención no es clara, pueden identificar el problema y sugerir una redacción alternativa e inequívoca del requerimiento. Como mínimo, se recomienda que los propietarios del requerimiento lleven al diseñador del sistema y al equipo de desarrollo a un recorrido del conjunto de requerimientos para garantizar que los requerimientos se entiendan, individualmente y como un conjunto.

Debido a las limitaciones del lenguaje, puede resultar difícil eliminar por completo toda ambigüedad. En este caso, el uso de la característica "rationale" puede proporcionar información adicional sobre la intención de eliminar la ambigüedad.

### C4 - Completo

El requerimiento describe de manera suficiente la capacidad, característica, restricción o factor de calidad necesarios para satisfacer la necesidad de la entidad sin necesidad de otra información para comprender el requerimiento.

#### Justificación:

Un acuerdo no es útil a menos que la obligación esté completa y no necesite más explicaciones. Un requerimiento bien formado no necesita más amplificación para implementar su intención. Cada requerimiento debe entenderse por derecho propio sin la sobrecarga de tener que entender a otros.

Las declaraciones de requerimientos no deben contener cláusulas para ser definidas (TBD - To Be Defined), para ser especificadas (TBS - To Be Specified) o para ser resueltas (TBR -To Be Resolved). TBx puede usarse durante el proceso de análisis para indicar el trabajo en curso, pero no debe estar en el requerimiento final.

Guía de cómo alcanzarlo:

Si bien apreciar completamente un requerimiento puede requerir algún contexto, la declaración en sí misma debe ser una oración completa que no requiera que se haga referencia a otras declaraciones para que se entiendan. Sin embargo, tenga en cuenta que un requerimiento puede referirse a otros documentos (por ejemplo, documentos de definición de interfaz, estándares y regulaciones). Al hacer estas referencias, sea específico a las secciones de los documentos que correspondan.

### C5 - Singular (Atómico)

El requerimiento debe indicar una única capacidad, característica, restricción o factor de calidad.

Justificación:

La transformación de necesidades a requerimientos es una transformación uno a uno o muchos a uno, por lo que la declaración de requerimiento resultante debe representar un solo pensamiento, aspecto o expectativa. La efectividad de varios procesos asociados con los requerimientos, como la descomposición, derivación, asignación, rastreo y verificación, depende de la capacidad de identificar declaraciones atómicas. Por ejemplo, la información de verificación definida para un requerimiento puede ser mucho más precisa cuando ese requerimiento aborda una única capacidad, característica, restricción o factor de calidad. Un requerimiento con múltiples ideas es difícil de asignar y rastrear a un padre o fuente.

Guía de cómo alcanzarlo:

Mantenga el requerimiento limitado a una calidad, característica, capacidad, función o restricción. Comprenda cómo las declaraciones se ajustan a la filosofía de trazabilidad del proyecto.

Si bien un único requerimiento debe consistir en una sola función, calidad o restricción, puede tener múltiples condiciones bajo las cuales se debe cumplir el requerimiento.

Evite el uso de la palabra "y" cuando una múltiples frases en la oración, cada una de las cuales puede asignarse y verificarse de manera diferente. La presencia de la conjunción "y" en una declaración de requerimiento siempre debe incitar al escritor a considerar si la declaración es singular o no.

### C6 - Factible

El requerimiento se puede cumplir dentro de las restricciones de la entidad (por ejemplo, costo, cronograma, técnico, legal, ético, regulatorio) con un riesgo aceptable.

Justificación:

No tiene mucho sentido aceptar una obligación para un requerimiento que no es factible. Estar de acuerdo con un requerimiento que no se puede cumplir con un riesgo aceptable dentro de las limitaciones a menudo resulta en excesos de costos del proyecto y retrasos en el cronograma.

Los requerimientos inherentemente inalcanzables, como el 100% de confiabilidad, son, en el mejor de los casos, una pérdida de tiempo y, en el peor, conducen a soluciones innecesariamente costosas.

Guía de cómo alcanzarlo:

El requerimiento establecido debe ser realizable por al menos una característica de diseño (dentro de las restricciones).

En general, no es posible determinar si una declaración de requerimiento individual es factible sin un análisis considerable de posibles soluciones. Sin embargo, generalmente podemos reconocer y evitar requerimientos que son imposibles o poco realistas.

Debido a que determinar la viabilidad no siempre se conoce completamente, la viabilidad a menudo se evalúa en términos de riesgo aceptable. Antes de permitir un requerimiento en su conjunto, se debe realizar una evaluación de viabilidad utilizando las tecnologías y métodos de ingeniería existentes. Si no es factible dentro de las restricciones establecidas, el requerimiento no debe incluirse en el conjunto. Hacerlo puede afectar negativamente el costo y el cronograma y puede dar lugar a un requerimiento que no se cumplirá ni se verificará.

## C7 - Verificable

El requerimiento está estructurado y redactado de manera tal que su realización pueda probarse (verificarse).

Justificación:

A menos que un requerimiento esté escrito de una manera que permita el diseño o la verificación del sistema, no hay forma de saber si se ha cumplido y si se ha cumplido la obligación.

Guía de cómo alcanzarlo:

Escriba cada requerimiento de una manera que permita verificar el diseño o sistema que el requerimiento se ha cumplido en el nivel apropiado mediante uno de los cuatro métodos de verificación estándar (inspección, análisis, demostración o prueba). Los diferentes tipos de requerimientos son verificables de diferentes maneras, y esto influirá en la forma en que se redacta el requerimiento. En general, para ser verificable, un requerimiento debe ser medible.

Un cliente puede especificar: "El rango de valores será lo más grande posible". Esta declaración es ambigua y no verificable. Este tipo de requerimiento es una señal de que se necesita un estudio para establecer un requerimiento de rango de valores máximo verificable.

Las causas más habituales para que un requerimiento no sea verificable son:

- no hay una especificación clara del comportamiento, las condiciones y los estados funcionales correctos.
- falta de precisión en los rangos de valores de rendimiento aceptable.
- uso de términos ambiguos.
- el requerimiento no es factible.

Al redactar requerimientos, utilice un punto de vista de verificación para imaginarse realizando el evento de verificación (análisis, inspección, demostración o prueba) y defina qué prueba dará

como resultado que se logre la intención del requerimiento como resultado del evento de verificación.

Las preguntas útiles para formular sobre un requerimiento son:

- ¿Cómo sabré si se ha cumplido el requerimiento? Si el requerimiento se ha cuantificado adecuadamente, proporcionará una respuesta precisa a esta pregunta.
- ¿Cuáles son los niveles obligatorios y deseables de desempeño requeridos? El resultado de esto puede ser que se proporcionan varios valores que describen la tolerancia permitida para este requerimiento.

Los métodos de verificación que se pueden emplear son:

**Inspección** tiene como objetivo detectar fallas en cualquiera de los documentos generados, se lleva a cabo mediante una revisión sistemática de los documentos generados.

**Análisis** de requerimientos consiste en el estudio de los requerimientos para verificar que estén correctamente adecuados a las características mencionadas, algunos autores también lo consideran inspección. En la misma se enfocan e intentan solucionar las deficiencias que los requerimientos puedan tener.

La **demostración** es parte de la verificación formal y consiste en probar matemáticamente que la especificación cumple con ciertas propiedades.

En la **prueba** se ejecutan casos de pruebas a partir de las especificaciones de requerimientos.

## C8 - Correcto

El requerimiento debe ser una representación precisa de la necesidad de la entidad a partir de la cual se transformó.

Justificación:

La transformación debe dar como resultado un requerimiento que debe poder validarse para que el requerimiento comunique lo correcto (por ejemplo, valor, tolerancia y unidades) de modo que se satisfagan las necesidades de la entidad. Debe poder demostrarse que el cumplimiento del requerimiento, tal como está escrito, dará como resultado la satisfacción de las necesidades a partir de las cuales se transformó.

Guía de cómo alcanzarlo:

Asegúrese de que la declaración de requerimientos refleje:

- descomposición o derivación que se basa en una interpretación correcta del requerimiento o necesidad de nivel superior;
- una comprensión correcta de las metas y objetivos subyacentes;

Utilice un proceso de validación de requerimientos definido para garantizar la corrección de la transformación en el contexto de los requerimientos individuales, así como el conjunto completo de requerimientos.

## C9 - Acorde a una forma

Los requerimientos individuales deben cumplir con un patrón y estilo estándar aprobado para su escritura.

**Justificación:**

Cuando los requerimientos dentro de la misma organización tienen el mismo aspecto, cada requerimiento es más fácil de escribir, comprender y revisar.

**Guía de cómo alcanzarlo:**

Se puede requerir que todos los requerimientos estén estructurados de acuerdo con un patrón específico definido para el tipo de declaración de requerimientos, por ejemplo: “Cuando <cláusula de condición>, la <cláusula de sujeto> deberá <cláusula de verbo de acción> <cláusula de objeto>, <calificación opcional cláusula>”.

Las organizaciones deben tener procesos bien definidos para el desarrollo y la gestión de requerimientos. Estos procesos deben incluir reglas para escribir requerimientos bien formados, listas de verificación para evaluar la calidad de los requerimientos, plantillas para estructuras permitidas para declaraciones de requerimientos.

Al desarrollar requerimientos para un cliente, el cliente puede tener su propio proceso, listas de verificación y patrones. Los escritores de requerimientos pueden necesitar ajustarse a estos procesos.

**Características de un conjunto de requerimientos.**

El conjunto de requerimientos especificados en un documento de especificación de requerimientos debe cumplir con las características resumidas en la Tabla 6.

Característica	Definición
<b>C10 - Completo</b>	El conjunto de requerimientos es independiente de tal manera que describe suficientemente las capacidades, características, restricciones, interfaces, estándares, regulaciones y / o factores de calidad necesarios para satisfacer las necesidades de la entidad sin necesitar otra información.
<b>C11 - Consistente</b>	El conjunto de requerimientos contiene requerimientos individuales que son únicos, no entran en conflicto ni se superponen con otros requerimientos del conjunto, y las unidades y sistemas de medición que utilizan son homogéneas. El lenguaje utilizado dentro del conjunto de requerimientos es coherente.
<b>C12 - Factible</b>	El conjunto de requerimientos puede realizarse dentro de las restricciones de la entidad con un riesgo aceptable.
<b>C13 - Comprensible</b>	El conjunto de requerimientos debe redactarse de manera que quede claro lo que espera la entidad y su relación con el sistema del que forma parte.
<b>C14 - Validable</b>	Debe poder probarse que el conjunto de requerimientos conducirá al logro de las necesidades de la entidad dentro de las restricciones.

Tabla 6. Características que deben reunir un conjunto de requerimientos.

### C10 - Completo

El conjunto de requerimientos es independiente, de tal manera que describe suficientemente las capacidades, características, restricciones, interfaces, estándares, regulaciones y / o factores de calidad necesarios para satisfacer las necesidades de la entidad sin necesitar otra información.

#### Justificación:

Si la transformación da como resultado requerimientos individuales que son necesarios, el conjunto de requerimientos debe ser una representación suficiente del conjunto de necesidades, es decir, se han incluido todos los requerimientos necesarios para satisfacer las necesidades y se han excluido todos los requerimientos innecesarios.

#### Guía de cómo alcanzarlo:

El objetivo es comunicar claramente las necesidades de las partes interesadas a través del conjunto mínimo de requerimientos que son necesarios y suficientes y no más.

Un conjunto de declaraciones de requerimientos representa una definición completa de las expectativas de los interesados. Abordar las necesidades implícitas es una parte clave para definir el conjunto de requerimientos y derivar los requerimientos que dan como resultado que se satisfagan esas necesidades, incluso si no se declararon originalmente como necesidades.

Las partes interesadas son una fuente primaria de requerimientos; dejar de lado a una parte interesada relevante podría resultar en requerimientos faltantes o incorrectos, resultando en un reprocesamiento costoso y lento.

En el caso de las interfaces, los requerimientos deben referirse a dónde se define la interacción entre el sistema a diseñar y otros sistemas. En el caso de las normas y reglamentos, se debe hacer referencia a los requerimientos específicos que se aplican al sistema a diseñar, en lugar de a todo el documento.

### C11 - Consistente

El conjunto de requerimientos contiene requerimientos individuales que son únicos, no entran en conflicto ni se superponen con otros requerimientos del conjunto, y las unidades y sistemas de medición que utilizan son homogéneas. El lenguaje utilizado dentro del conjunto de requerimientos es coherente (es decir, la misma palabra se utiliza en todo el conjunto para representar lo mismo).

#### Justificación:

Los requerimientos en conflicto conducen a un espacio de solución incompleto y, si no se identifican al principio del proceso de desarrollo, pueden llevar a un costoso reproceso.

El conjunto resultante de requerimientos individuales no debe entrar en conflicto y debe ser coherente entre sí.

Con frecuencia las necesidades de los clientes y otras partes interesadas relevantes o las restricciones de diseño entran en conflicto entre sí y deben conciliarse. Además, incluso si los requerimientos individuales no son ambiguos, el uso inconsistente de términos, abreviaturas,



unidades y sistemas de medición en diferentes requerimientos da como resultado una ambigüedad en el conjunto de requerimientos.

Guía de cómo alcanzarlo:

Es un desafío detectar conflictos entre requerimientos cuando el conjunto de requerimientos es grande.

Es importante identificar los requerimientos que tienen relaciones con otros requerimientos, ya sea directa o indirectamente. Esto es especialmente un problema cuando se cambia un requerimiento sin hacer un cambio correspondiente a los otros requerimientos dependientes. Una herramienta clave para ayudar a prevenir este tipo de inconsistencia es vincular (rastrear/trazar) los requerimientos dependientes entre sí.

También es difícil identificar conflictos simplemente a través del lenguaje utilizado para expresar requerimientos individuales, pero se puede facilitar clasificando y agrupando los requerimientos. Una estrategia es clasificar los requerimientos según los tipos de aspectos que cubren (por ejemplo, seguridad, calidad, área funcional). Luego, utilice la clasificación y el filtrado para reunir requerimientos que de otro modo serían diversos y examinarlos en busca de interacciones, compensaciones, dependencias y conflictos.

Utilice un glosario para garantizar el uso coherente de términos y abreviaturas en todo el conjunto de requerimientos.

## C12 - Factible

El conjunto de requerimientos puede realizarse dentro de las restricciones de la organización o empresa (por ejemplo, costo, cronograma, técnico, legal, ético, regulatorio) con un riesgo aceptable.

Justificación:

Del mismo modo que tiene poco sentido comprometerse a alcanzar un requerimiento individual que no es factible, el conjunto de requerimientos debe cumplirse dentro de las restricciones apropiadas, incluidos el costo y el cronograma. Si la inviabilidad no se identifica temprano en el proceso de desarrollo, puede conducir a un esfuerzo y costo desperdiciados.

Guía de cómo alcanzarlo:

Si bien las declaraciones de requerimientos individuales pueden parecer factibles, pueden no serlo cuando se colocan en combinación con otras. Es decir, la combinación de requerimientos individuales factibles no necesariamente conforma un conjunto factible de requerimientos.

Si los requerimientos individuales relacionados se distribuyen a través del conjunto de requerimientos, es aún más difícil para nosotros "entender" la viabilidad del conjunto. Como se indicó para las declaraciones de requerimientos individuales, determinar la viabilidad de un conjunto de requerimientos no siempre se puede realizar de manera certera y, a menudo, se evalúa en términos de riesgo aceptable.

En el espacio de la solución debe haber al menos una y preferiblemente múltiples soluciones alcanzables. La solución debe ser técnicamente alcanzable (como en términos de nivel de madurez y avance de la tecnología) y alcanzable dentro de las limitaciones del proyecto (como el costo, el calendario, el cumplimiento técnico, legal, ético y regulatorio).

### C13 - Comprensible

El conjunto de requerimientos debe redactarse de manera que quede claro lo que espera la organización o empresa y su relación con el sistema.

#### Justificación:

Es difícil promulgar un acuerdo, a menos que ambas partes tengan clara la obligación exacta y los resultados esperados con la implementación del conjunto de requerimientos. Por lo tanto, el conjunto debe estar escrito de modo que la audiencia relevante pueda comprender lo que se comunica mediante los requerimientos individuales, así como el conjunto de requerimientos.

#### Guía de cómo alcanzarlo:

La información necesaria para comprender el contexto de los requerimientos se incluye con el conjunto de requerimientos. Esto incluye información útil en el documento de requerimientos, así como atributos del requerimiento, tal como su justificación o razón ("rationale").

### C14 - Validable

Debe poder probarse que el conjunto de requerimientos conducirá al logro de las necesidades de la organización o empresa dentro de las restricciones (como el costo, el cronograma, el cumplimiento técnico, legal y regulatorio).

#### Justificación:

Debe poder mostrarse en cualquier etapa del desarrollo del sistema que el logro del conjunto de requerimientos dará como resultado la satisfacción del conjunto de necesidades a partir del cual se transformó. La validación del sistema se asegura de que el sistema construido y verificado cumpla con el propósito previsto en su entorno operativo. Por lo tanto, la validación del sistema vuelve a las necesidades para asegurarse de que se hayan cumplido, como un conjunto.

#### Guía de cómo alcanzarlo:

Los escenarios operativos, los conceptos y los casos de uso del ciclo de vida del sistema, a partir de los cuales se desarrollaron las necesidades y se transformaron los requerimientos, se pueden usar como casos de prueba para validar el conjunto de requerimientos. En otras palabras, el conjunto de requerimientos se puede validar entendiendo cómo cumple con los escenarios operativos que dan como resultado que la misión, las metas y los objetivos se cumplan dentro de los impulsores y limitaciones acordados.

Se debe tener cuidado para garantizar que también se validen los requerimientos de calidad.

Haga las preguntas: "¿La entidad desarrollada por este conjunto de requerimientos satisfará las necesidades de la entidad?" "¿Construimos lo correcto?".

## 2.5. Atributos de requerimientos

Arlow y Neustadt (2006) definen en el capítulo 3 que todos los requerimientos pueden tener un conjunto de atributos que captura información adicional (metadatos) sobre los requerimientos. Todo atributo de requerimiento tiene un nombre descriptivo y un valor. Por ejemplo, un requerimiento puede tener un atributo denominado fechaVencimiento que tiene como valor la fecha en que se tiene que entregar el requerimiento. El requerimiento también podría tener un atributo origen que posee como valor una descripción de dónde se ha originado ese requerimiento. El conjunto preciso de atributos que se decida emplear depende de la naturaleza y necesidades del proyecto y puede variar por tipo de requerimiento.

Quizás el atributo de requerimiento más común es prioridad. El valor de este atributo es la prioridad del requerimiento relativa a todos los otros requerimientos. Un esquema común para asignar prioridad es el siguiente conjunto de criterios:

- Debe tener. Requerimientos obligatorios que son fundamentales para el sistema.
- Debería tener. Requerimientos importantes que se pueden omitir.
- Podría tener. Requerimientos que son opcionales, se realizan si hay tiempo.
- Quiere tener. Requerimientos que pueden esperar para versiones posteriores del sistema.

Cuando se utilizan estos criterios, cada requerimiento tiene un atributo Prioridad que puede adoptar uno de los valores.

RUP propone emplear los siguientes atributos:

- Estado. Este atributo puede tener los siguientes valores:
  - Propuesto. Los requerimientos siguen estando bajo deliberación y todavía no se han acordado.
  - Aprobado. Los requerimientos que se han aprobado para implementación.
  - Rechazado. Los requerimientos que se han rechazado para la implementación.
  - Incorporado. Los requerimientos que se han implementado para una versión en particular.
- Beneficio (prioridad). Este atributo puede tener los siguientes valores:
  - Crítico. El requerimiento se debe implementar; de lo contrario el sistema no será adecuado para los grupos de decisión.
  - Importante. El requerimiento se podría omitir, pero esto afectaría a la usabilidad del sistema y la satisfacción de los grupos de decisión.
  - Útil. El requerimiento se podría omitir sin impacto importante en la aceptación del sistema.
- Esfuerzo. Es una estimación del tiempo y los recursos necesarios para implementar la característica medida en personas/días o cualquier otra unidad.
- Riesgo. Indica el riesgo que implica añadir esta característica. Su valor puede ser Alto, Medio, o Bajo
- Estabilidad. Es una estimación de la probabilidad de que el requerimiento cambiará de alguna forma. Su valor puede ser Alto, Medio, o Bajo.
- VersiónObjetivo. La versión del producto en la que el requerimiento se debería implementar.

Además de los atributos indicados, en INCOSE (2017) e ISO/IEC/IEEE (2011) destacan los siguientes atributos:

- Identificación. Cada requerimiento debe estar identificado de manera única.
- Autor, quién escribió el requerimiento.
- Fecha, fecha en la que se escribió el requerimiento.
- Stakeholder, lista de interesados en el requerimiento que estarán involucrados en el proceso de revisión y/o aprobación.
- Dependencia. Define la dependencia entre requerimientos, particularmente la traza a requerimiento padre.
- Trazabilidad a la fuente. La fuente puede ser un stakeholder o un modelo.
- Rationale (Razón). Explicita las razones por que el requerimiento es necesario.
- Método primario de verificación. Establece el método a emplear para su verificación.
- Dificultad. Indica una estimación sobre la dificultad de lograr el requerimiento. Sus valores pueden ser Fácil, Nominal, o Difícil.
- Estabilidad, indica qué probable es que el requerimiento cambie.

## 2.6. Procesos de ingeniería de requerimientos.

Sommerville (2011) indica que los procesos de ingeniería de requerimientos incluyen cuatro actividades de alto nivel. Éstas se enfocan en valorar si el sistema es útil para la empresa (estudio de factibilidad), descubrir requerimientos (adquisición y análisis), convertir dichos requerimientos en alguna forma estándar (especificación) y comprobar que los requerimientos definan realmente el sistema que quiere el cliente (validación):

1. Estudio de factibilidad. Se realiza una estimación sobre si las necesidades identificadas del usuario se cubren con las actuales tecnologías de software y hardware. El estudio considera si el sistema propuesto tendrá un costo-beneficio desde un punto de vista empresarial, y si éste puede desarrollarse dentro de las restricciones presupuestales existentes. Un estudio de factibilidad debe ser rápido y relativamente barato. El resultado debe informar la decisión respecto a si se continúa o no continúa con un análisis más detallado.
2. Obtención y análisis de requerimientos. Éste es el proceso de derivar los requerimientos del sistema mediante observación de los sistemas existentes, discusiones con los usuarios y proveedores potenciales, análisis de tareas, etcétera. Esto puede incluir el desarrollo de uno o más modelos de sistemas y prototipos, lo que ayuda a entender el sistema que se va a especificar.
3. Especificación de requerimientos. Consiste en la actividad de transcribir la información recopilada durante la actividad de análisis, en un documento que define un conjunto de requerimientos. En este documento se incluyen las dos clases de requerimientos: requerimientos del usuario y requerimientos de sistema.
4. Validación de requerimientos. Esta actividad verifica que los requerimientos sean realistas, coherentes y completos. Durante este proceso es inevitable descubrir errores en el documento de requerimientos. En consecuencia, deberían modificarse con la finalidad de corregir dichos problemas.

En la Figura 3 se muestra esto como proceso secuencial; sin embargo, en la práctica, la ingeniería de requerimientos es un proceso iterativo donde las actividades están entrelazadas. Desde

luego, las actividades en el proceso de requerimientos no se realizan simplemente en una secuencia estricta. El análisis de requerimientos continúa durante la definición y especificación, y a lo largo del proceso salen a la luz nuevos requerimientos; por lo tanto, las actividades de análisis, definición y especificación están vinculadas. En los métodos ágiles, como programación extrema, los requerimientos se desarrollan de manera incremental según las prioridades del usuario, en tanto que la obtención de requerimientos proviene de los usuarios que son parte del equipo de desarrollo.

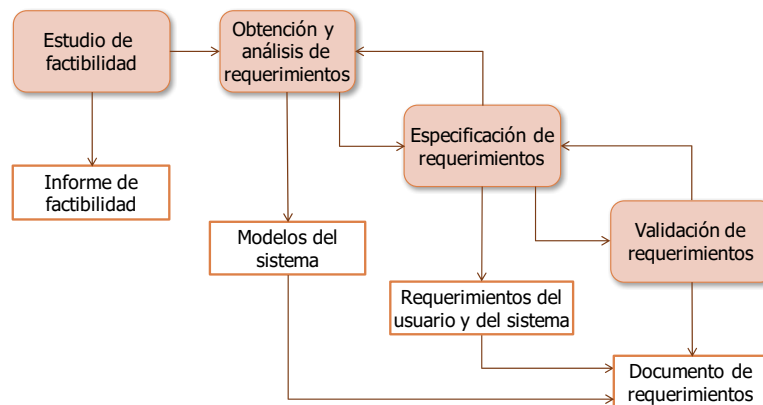


Figura 3. Proceso de ingeniería de requerimientos

La Figura 4 presenta este entrelazamiento. Las actividades están organizadas como un proceso iterativo alrededor de una espiral, y la salida es un documento de requerimientos del sistema. La cantidad de tiempo y esfuerzo dedicados a cada actividad en cada iteración depende de la etapa del proceso global y el tipo de sistema que está siendo desarrollado.

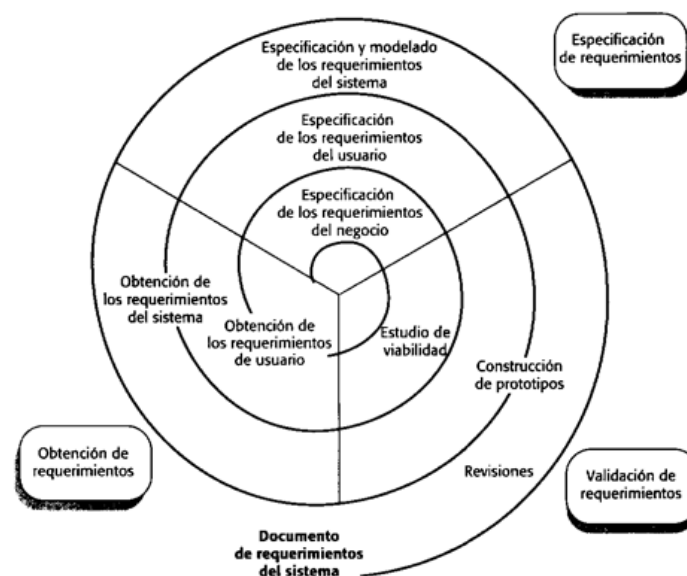


Figura 4. Vista en espiral del proceso de ingeniería de requerimientos

En el inicio del proceso, se empleará más esfuerzo para comprender los requerimientos empresariales de alto nivel y los no funcionales, así como los requerimientos del usuario para el

sistema. Más adelante en el proceso, en los anillos exteriores del espiral, se dedicará más esfuerzo a la adquisición y comprensión de los requerimientos detallados del sistema.

Este modelo en espiral acomoda enfoques al desarrollo, donde los requerimientos se elaboraron con diferentes niveles de detalle. El número de iteraciones del espiral tiende a variar, de modo que el espiral terminará después de adquirir algunos o todos los requerimientos del usuario. Se puede usar el desarrollo ágil en vez de la creación de prototipos, de manera que se diseñen en conjunto los requerimientos y la implementación del sistema.

Prácticamente en todos los sistemas cambian los requerimientos. Las personas implicadas desarrollan una mejor comprensión de qué quieren que haga el software; la organización que compra el sistema cambia; se hacen modificaciones al hardware, al software y al entorno organizacional del sistema. Al proceso de administrar tales requerimientos cambiantes se le llama administración de requerimientos.

## 2.7 Estudio de Factibilidad

Un estudio de factibilidad es un breve estudio sobre cuán beneficioso o práctico es desarrollar o adquirir un sistema para una organización. Debe responder tres preguntas clave: a) ¿El sistema contribuye con los objetivos globales de la organización? b) ¿El sistema puede implementarse dentro de la fecha y el presupuesto usando la tecnología actual? c) ¿El sistema puede integrarse con otros sistemas que se utilicen?

Si la respuesta a cualquiera de estas preguntas es negativa, probablemente no sea conveniente continuar con el proyecto.

En el informe de factibilidad se analiza la factibilidad técnica, factibilidad operativa, y factibilidad económica. Debería hacerse una recomendación sobre si debe continuar o no el desarrollo del sistema. En el informe, se pueden proponer cambios en el alcance, el presupuesto y la confección de agendas del sistema y sugerir requerimientos adicionales de alto nivel para éste.

## 2.8 Adquisición y Análisis de Requerimientos

Después de un estudio de factibilidad inicial, la siguiente etapa del proceso de ingeniería de requerimientos es la adquisición y el análisis de requerimientos. En esta actividad, los ingenieros de software trabajan con clientes y usuarios finales del sistema para descubrir el dominio de aplicación, qué servicios debe proporcionar el sistema, el desempeño requerido de éste, las restricciones de hardware, etcétera.

En una organización, la adquisición y el análisis de requerimientos pueden involucrar a diversas clases de personas. Un participante en el sistema, o “stakeholder”, es quien debe tener alguna influencia directa o indirecta sobre los requerimientos del mismo. Los participantes incluyen a usuarios finales que interactuarán con el sistema, y a cualquiera en una organización que resultará afectada por él. Otros participantes del sistema pueden ser los ingenieros que desarrollan o mantienen otros sistemas relacionados, administradores de negocios, expertos de dominio y representantes de asociaciones sindicales.

En la Figura 5 se muestra un modelo del proceso de adquisición y análisis. Cada organización tendrá su versión o ejemplificación de este modelo general, dependiendo de factores locales, tales como experiencia del personal, tipo de sistema a desarrollar, estándares usados, etcétera.

Las actividades del proceso son:

1. Descubrimiento de requerimientos. Éste es el proceso de interactuar con los participantes del sistema para descubrir sus requerimientos. También los requerimientos de dominio de los participantes y la documentación se descubren durante esta actividad. Existen numerosas técnicas complementarias que pueden usarse para el descubrimiento de requerimientos.
2. Clasificación y organización de requerimientos. Esta actividad toma la compilación no estructurada de requerimientos, agrupa requerimientos relacionados y los organiza en grupos coherentes. La forma más común de agrupar requerimientos es usar un modelo de la arquitectura del sistema, para identificar subsistemas y asociar los requerimientos con cada subsistema. En la práctica, la ingeniería de requerimientos y el diseño arquitectónico no son actividades separadas completamente.
3. Priorización y negociación de requerimientos. Inevitablemente, cuando intervienen diversos participantes, los requerimientos entrarán en conflicto. Esta actividad se preocupa por priorizar los requerimientos, así como por encontrar y resolver conflictos de requerimientos mediante la negociación. Por lo general, los participantes tienen que reunirse para resolver las diferencias y estar de acuerdo con el compromiso de los requerimientos.
4. Especificación de requerimientos. Los requerimientos se documentan e ingresan en la siguiente ronda de la espiral. Pueden producirse documentos de requerimientos formales o informales.

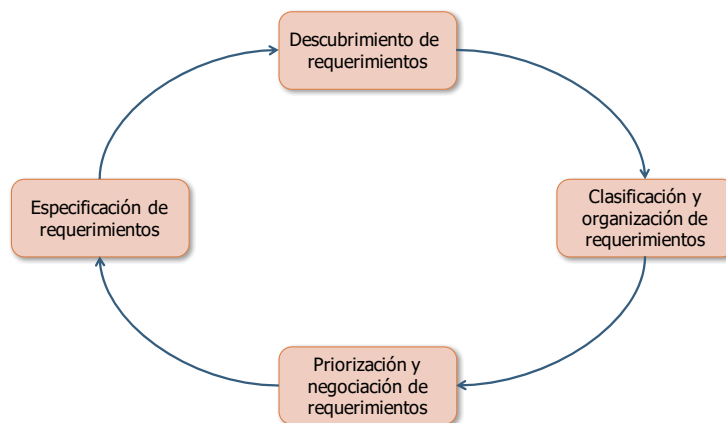


Figura 5. El proceso de adquisición y análisis de requerimiento

La Figura 5 muestra que la adquisición y el análisis de requerimientos es un proceso iterativo con retroalimentación continua de cada actividad a otras actividades. El ciclo del proceso comienza con el descubrimiento de requerimientos y termina con la documentación de los requerimientos. La comprensión de los requerimientos por parte del analista mejora con cada ronda del ciclo. El ciclo concluye cuando está completo el documento de requerimientos.

La adquisición y la comprensión de los requerimientos por parte de los participantes del sistema es un proceso difícil por diferentes razones:

1. Los participantes con frecuencia no saben lo que quieren de un sistema de cómputo, excepto en términos muy generales; pueden encontrar difícil articular qué quieren que haga el sistema; pueden hacer peticiones inalcanzables porque no saben qué es factible y qué no lo es.

2. Los participantes en un sistema expresan naturalmente los requerimientos con sus términos y conocimientos implícitos de su trabajo. Los ingenieros de requerimientos, sin experiencia en el dominio del cliente, podrían no entender dichos requerimientos.
3. Diferentes participantes tienen distintos requerimientos y pueden expresarlos en variadas formas. Los ingenieros de requerimientos deben descubrir todas las fuentes potenciales de requerimientos e identificar similitudes y conflictos.
4. Factores políticos llegan a influir en los requerimientos de un sistema. Los administradores pueden solicitar requerimientos específicos del sistema, porque éstos les permitirán aumentar su influencia en la organización.
5. El ambiente económico y empresarial donde ocurre el análisis es dinámico. Inevitablemente cambia durante el proceso de análisis. Puede cambiar la importancia de requerimientos particulares; o bien, tal vez surjan nuevos requerimientos de nuevos participantes a quienes no se consultó originalmente.

Resulta ineludible que diferentes participantes tengan diversas visiones de la importancia y prioridad de los requerimientos y, algunas veces, dichas visiones están en conflicto. Durante el proceso, usted deberá organizar negociaciones regulares con los participantes, de forma que se alcancen compromisos. Es imposible complacer por completo a cada participante, pero, si algunos suponen que sus visiones no se consideraron de forma adecuada, quizás intenten deliberadamente socavar el proceso de IR.

En la etapa de especificación de requerimientos, los requerimientos adquiridos hasta el momento se documentan de tal forma que puedan usarse para ayudar al hallazgo de requerimientos. En esta etapa, podría generarse una primera versión del documento de requerimientos del sistema, con secciones faltantes y requerimientos incompletos. De modo alternativo, los requerimientos pueden documentarse en una forma completamente diferente (por ejemplo, en una hoja de cálculo o en tarjetas).

## 2.9 Descubrimiento de requerimientos

El descubrimiento de requerimientos (llamado a veces adquisición de requerimientos) es el proceso de recopilar información sobre el sistema requerido y los sistemas existentes, así como de separar, a partir de esta información, los requerimientos del usuario y del sistema. Las fuentes de información durante la fase de descubrimiento de requerimientos incluyen documentación, participantes del sistema y especificaciones de sistemas similares. La interacción con los participantes es a través de entrevistas y observaciones, y pueden usarse escenarios y prototipos para ayudar a los participantes a entender cómo será el sistema.

Los participantes varían desde administradores y usuarios finales de un sistema hasta participantes externos como los reguladores, quienes certifican la aceptabilidad del sistema. Por ejemplo, los participantes que se incluyen para el sistema de información de pacientes en atención a la salud mental son:

1. Pacientes cuya información se registra en el sistema.
2. Médicos que son responsables de valorar y tratar a los pacientes.
3. Enfermeros que coordinan, junto con los médicos, las consultas y suministran algunos tratamientos.
4. Recepcionistas que administran las citas médicas de los pacientes.
5. Personal de Tecnología de la Información que es responsable de instalar y mantener el sistema.



6. Un director de ética médica que debe garantizar que el sistema cumpla con los lineamientos éticos actuales de la atención al paciente.
7. Encargados de atención a la salud que obtienen información administrativa del sistema.
8. Personal de archivo médico que es responsable de garantizar que la información del sistema se conserve, y se implementen de manera adecuada los procedimientos de mantenimiento del archivo.

Además de los participantes del sistema, se observa que los requerimientos también pueden venir del dominio de aplicación y de otros sistemas que interactúan con el sistema a especificar. Todos ellos deben considerarse durante el proceso de adquisición de requerimientos.

Todas estas diferentes fuentes de requerimientos (participantes, dominio, sistemas) se representan como puntos de vista del sistema, y cada visión muestra un subconjunto de los requerimientos para el sistema. Diferentes puntos de vista de un problema enfocan el problema de diferentes formas. Sin embargo, sus perspectivas no son totalmente independientes, sino que por lo general se traslapan, de manera que tienen requerimientos comunes. Usted puede usar estos puntos de vista para estructurar tanto el descubrimiento como la documentación de los requerimientos del sistema.

## 2.10 Validación de requerimientos

La validación de requerimientos es el proceso de verificar que los requerimientos definan realmente el sistema que en verdad quiere el cliente. Se traslapa con el análisis, ya que se interesa por encontrar problemas con los requerimientos. La validación de requerimientos es importante porque los errores en un documento de requerimientos pueden conducir a grandes costos por tener que rehacer, cuando dichos problemas se descubren durante el desarrollo del sistema o después de que éste se halla en servicio.

En general, el costo por corregir un problema de requerimientos al hacer un cambio en el sistema es mucho mayor que reparar los errores de diseño o codificación. La razón es que un cambio a los requerimientos significa generalmente que también deben cambiar el diseño y la implementación del sistema. Más aún, el sistema debe entonces ponerse a prueba de nuevo.

Durante el proceso de validación de requerimientos, tienen que realizarse diferentes tipos de comprobaciones sobre los requerimientos contenidos en el documento de requerimientos. Dichas comprobaciones incluyen:

1. Comprobaciones de validez. Un usuario quizá crea que necesita un sistema para realizar ciertas funciones. Sin embargo, con mayor consideración y análisis se logra identificar las funciones adicionales o diferentes que se requieran. Los sistemas tienen diversos participantes con diferentes necesidades, y cualquier conjunto de requerimientos es inevitablemente un compromiso a través de la comunidad de participantes.
2. Comprobaciones de consistencia. Los requerimientos en el documento no deben estar en conflicto. Esto es, no debe haber restricciones contradictorias o descripciones diferentes de la misma función del sistema.
3. Comprobaciones de totalidad. El documento de requerimientos debe incluir requerimientos que definan todas las funciones y las restricciones pretendidas por el usuario del sistema.
4. Comprobaciones de realismo. Al usar el conocimiento de la tecnología existente, los requerimientos deben comprobarse para garantizar que en realidad pueden

implementarse. Dichas comprobaciones también tienen que considerar el presupuesto y la fecha para el desarrollo del sistema.

5. Verificabilidad. Para reducir el potencial de disputas entre cliente y contratista, los requerimientos del sistema deben escribirse siempre de manera que sean verificables. Esto significa que usted debe ser capaz de escribir un conjunto de pruebas que demuestren que el sistema entregado cumpla cada requerimiento especificado.

Hay algunas técnicas de validación de requerimientos que se usan individualmente o en conjunto con otras:

1. Revisiones de requerimientos. Los requerimientos se analizan sistemáticamente usando un equipo de revisores que verifican errores e inconsistencias.
2. Creación de prototipos. En esta aproximación a la validación, se muestra un modelo ejecutable del sistema en cuestión a los usuarios finales y clientes. Así, ellos podrán experimentar con este modelo para constatar si cubre sus necesidades reales.
3. Generación de casos de prueba. Los requerimientos deben ser comprobables. Si las pruebas para los requerimientos se diseñan como parte del proceso de validación, esto revela con frecuencia problemas en los requerimientos. Si una prueba es difícil o imposible de diseñar, esto generalmente significa que los requerimientos serán difíciles de implementar, por lo que deberían reconsiderarse. El desarrollo de pruebas a partir de los requerimientos del usuario antes de escribir cualquier código es una pieza integral de la programación extrema.

No hay que subestimar los problemas incluidos en la validación de requerimientos. A final de cuentas, es difícil demostrar que un conjunto de requerimientos, de hecho, no cubre las necesidades de los usuarios. Estos últimos necesitan una imagen del sistema en operación, así como comprender la forma en que dicho sistema se ajustará a su trabajo. Es difícil, incluso para profesionales de la computación experimentados, realizar este tipo de análisis abstracto, y más aún para los usuarios del sistema. Como resultado, rara vez usted encontrará todos los problemas de requerimientos durante el proceso de validación de requerimientos. Es inevitable que haya más cambios en los requerimientos para corregir omisiones y malas interpretaciones, después de acordar el documento de requerimientos.

### 2.11 Administración de requerimientos

Los requerimientos para los grandes sistemas de software siempre cambian. Una razón es que dichos sistemas se desarrollaron por lo general para resolver problemas “horrorosos”: aquellos problemas que no se pueden definir por completo. Como el problema no se logra definir por completo, los requerimientos del software están condenados también a estar incompletos. Durante el proceso de software, la comprensión que los participantes tienen de los problemas cambia constantemente. Entonces, los requerimientos del sistema también deben evolucionar para reflejar esa visión cambiante del problema.

Una vez que se instala un sistema, y se utiliza con regularidad, surgirán inevitablemente nuevos requerimientos. Es difícil que los usuarios y clientes del sistema anticipen qué efectos tendrá el nuevo sistema sobre sus procesos de negocios y la forma en que se hace el trabajo. Una vez que los usuarios finales experimentan el sistema, descubrirán nuevas necesidades y prioridades. Existen muchas razones por las que es inevitable el cambio:

1. Los ambientes empresarial y técnico del sistema siempre cambian después de la instalación. Puede introducirse nuevo hardware, y quizá sea necesario poner en

interfaz el sistema con otros sistemas, cambiar las prioridades de la empresa (con los consecuentes cambios en el sistema de apoyo requerido) e introducir nuevas leyes y regulaciones que el sistema deba cumplir cabalmente.

2. Los individuos que pagan por un sistema y los usuarios de dicho sistema, por lo general, no son los mismos. Los clientes del sistema imponen requerimientos debido a restricciones organizativas y presupuestales. Esto podría estar en conflicto con los requerimientos del usuario final y, después de la entrega, probablemente deban agregarse nuevas características para apoyar al usuario, si el sistema debe cubrir sus metas.
3. Los sistemas grandes tienen regularmente una comunidad de usuarios diversa, en la cual muchos individuos tienen diferentes requerimientos y prioridades que quizás estén en conflicto o sean contradictorios. Los requerimientos finales del sistema inevitablemente tienen un compromiso entre sí y, con la experiencia, a menudo se descubre que el equilibrio de apoyo brindado a diferentes usuarios tiene que cambiar.

La administración de requerimientos es el proceso de comprender y controlar los cambios en los requerimientos del sistema. Es necesario seguir la pista de requerimientos individuales y mantener los vínculos entre los requerimientos dependientes, de manera que pueda valorarse el efecto del cambio en los requerimientos. También es preciso establecer un proceso formal para hacer cambios a las propuestas y vincular éstos con los requerimientos del sistema. El proceso formal de la administración de requerimientos debe comenzar tan pronto como esté disponible un borrador del documento de requerimientos. Sin embargo, hay que empezar a planear cómo administrar el cambio en los requerimientos durante el proceso de adquisición de los mismos.

Para la administración de requerimientos se debe decidir sobre:

1. Identificación de requerimientos. Cada requerimiento debe identificarse de manera exclusiva, de forma que pueda tener referencia cruzada con otros requerimientos y usarse en las evaluaciones de seguimiento.
2. Un proceso de administración del cambio. Éste es el conjunto de actividades que valoran el efecto y costo de los cambios. En la siguiente sección se estudia con más detalle este proceso.
3. Políticas de seguimiento. Dichas políticas definen las relaciones entre cada requerimiento, así como entre los requerimientos y el diseño del sistema que debe registrarse. La política de seguimiento también tiene que definir cómo mantener dichos registros.
4. Herramientas de apoyo. La administración de requerimientos incluye el procesamiento de grandes cantidades de información acerca de los requerimientos. Las herramientas disponibles varían desde sistemas especializados de administración de requerimientos, hasta hojas de cálculo y sistemas de bases de datos simples.

La administración de requerimientos necesita apoyo automatizado y herramientas de software, para lo cual deben seleccionarse durante la fase de planeación. Se necesitan herramientas de apoyo para:

1. Almacenamiento de requerimientos. Los requerimientos tienen que mantenerse en un almacén de datos administrado y seguro, que sea accesible para todos quienes intervienen en el proceso de ingeniería de requerimientos.

2. Administración del cambio. El proceso de administración del cambio se simplifica si está disponible la herramienta de apoyo activa.
3. Administración del seguimiento. Como se estudió anteriormente, la herramienta de apoyo para el seguimiento permite la identificación de requerimientos relacionados. Algunas herramientas que están disponibles usan técnicas de procesamiento en lenguaje natural, para ayudar a descubrir posibles relaciones entre los requerimientos.

Para sistemas pequeños, quizá no sea necesario usar herramientas especializadas de administración de requerimientos. El proceso de administración de requerimientos puede apoyarse con el uso de funciones disponibles en procesadores de texto, hojas de cálculo y bases de datos de PC. Sin embargo, para sistemas más grandes se requieren herramientas de apoyo más especializadas.

La administración del cambio en los requerimientos debe aplicarse a todos los cambios propuestos a los requerimientos de un sistema, después de aprobarse el documento de requerimientos. La administración del cambio es esencial porque es necesario determinar si los beneficios de implementar nuevos requerimientos están justificados por los costos de la implementación. La ventaja de usar un proceso formal para la administración del cambio es que todas las propuestas de cambio se tratan de manera consistente y los cambios al documento de requerimientos se realizan en una forma controlada.

Existen tres etapas principales de un proceso de administración del cambio:

1. Análisis del problema y especificación del cambio. El proceso comienza con la identificación de un problema en los requerimientos o, en ocasiones, con una propuesta de cambio específica. Durante esta etapa, el problema o la propuesta de cambio se analizan para comprobar que es válida. Este análisis retroalimenta al solicitante del cambio, quien responderá con una propuesta de cambio de requerimientos más específica, o decidirá retirar la petición.
2. Análisis del cambio y estimación del costo. El efecto del cambio propuesto se valora usando información de seguimiento y conocimiento general de los requerimientos del sistema. El costo por realizar el cambio se estima en términos de modificaciones al documento de requerimientos y, si es adecuado, al diseño y la implementación del sistema. Una vez completado este análisis, se toma una decisión acerca de si se procede o no con el cambio de requerimientos.
3. Implementación del cambio. Se modifican el documento de requerimientos y, donde sea necesario, el diseño y la implementación del sistema. Hay que organizar el documento de requerimientos de forma que sea posible realizar cambios sin reescritura o reorganización extensos. Conforme a los programas, la variabilidad en los documentos se logra al minimizar las referencias externas y al hacer las secciones del documento tan modulares como sea posible. De esta manera, secciones individuales pueden modificarse y sustituirse sin afectar otras partes del documento.

Si un nuevo requerimiento tiene que implementarse urgentemente, siempre existe la tentación para cambiar el sistema y luego modificar de manera retrospectiva el documento de requerimientos. Hay que tratar de evitar esto, pues casi siempre conducirá a que la especificación de requerimientos y la implementación del sistema se salgan de ritmo. Una vez realizados los cambios al sistema, es fácil olvidar la inclusión de dichos cambios en el documento

de requerimientos, o bien, agregar información al documento de requerimientos que sea inconsistente con la implementación.

Los procesos de desarrollo ágil, como la programación extrema, se diseñaron para enfrentar los requerimientos que cambian durante el proceso de desarrollo. En dichos procesos, cuando un usuario propone un cambio de requerimientos, éste no pasa por un proceso de administración del cambio formal. En vez de ello, el usuario tiene que priorizar dicho cambio y, si es de alta prioridad, decidir qué características del sistema planeadas para la siguiente iteración pueden eliminarse.

## 2.12 Modelado del sistema

El modelado de sistemas es el proceso para desarrollar modelos abstractos de un sistema, donde cada modelo presenta una visión o perspectiva diferente de dicho sistema. En esta sección se presenta el enfoque de modelado de sistema propuesto en el capítulo 5 de Sommerville (2011).

En general, el modelado de sistemas se ha convertido en un medio para representar el sistema usando algún tipo de notación gráfica, que ahora casi siempre se basa en notaciones en el Lenguaje de Modelado Unificado (UML). Sin embargo, también es posible desarrollar modelos formales (matemáticos) de un sistema, generalmente como una especificación detallada del sistema.

Los modelos se usan durante el proceso de ingeniería de requerimientos para ayudar a derivar los requerimientos de un sistema, durante el proceso de diseño para describir el sistema a los ingenieros que implementan el sistema, y después de la implementación para documentar la estructura y la operación del sistema. Es posible desarrollar modelos tanto del sistema existente como del sistema a diseñar:

1. Los modelos del sistema existente se usan durante la ingeniería de requerimientos. Ayudan a aclarar lo que hace el sistema existente y pueden utilizarse como base para discutir sus fortalezas y debilidades. Posteriormente, conducen a los requerimientos para el nuevo sistema.
2. Los modelos del sistema nuevo se emplean durante la ingeniería de requerimientos para ayudar a explicar los requerimientos propuestos a otros participantes del sistema. Los ingenieros usan tales modelos para discutir las propuestas de diseño y documentar el sistema para la implementación. En un proceso de ingeniería dirigido por modelo, es posible generar una implementación de sistema completa o parcial a partir del modelo del sistema.

El aspecto más importante de un modelo del sistema es que deja fuera los detalles. Un modelo es una abstracción del sistema a estudiar, y no una representación alternativa de dicho sistema. De manera ideal, una representación de un sistema debe mantener toda la información sobre la entidad a representar. Una abstracción simplifica y recoge deliberadamente las características más destacadas.

Desde diferentes perspectivas, usted puede desarrollar diferentes modelos para representar el sistema. Por ejemplo:

1. Una perspectiva externa, donde se modelen el contexto o entorno del sistema.
2. Una perspectiva de interacción, donde se modele la interacción entre un sistema y su entorno, o entre los componentes de un sistema.
3. Una perspectiva estructural, donde se modelen la organización de un sistema o la estructura de datos que procese el sistema.

4. Una perspectiva de comportamiento, donde se modele el comportamiento dinámico del sistema y cómo responde ante ciertos eventos.

Estas perspectivas tienen mucho en común con la visión 4 + 1 de arquitectura del sistema de Kruchten, la cual sugiere que la arquitectura y la organización de un sistema deben documentarse desde diferentes perspectivas.

En el curso emplearemos diagramas definidos en UML, que se han convertido en un lenguaje de modelado estándar para modelado orientado a objetos. El UML tiene numerosos tipos de diagramas y, por lo tanto, soporta la creación de muchos diferentes tipos de modelo de sistema. En particular abordaremos:

1. Diagramas de actividad, que muestran las actividades incluidas en un proceso o en el procesamiento de datos.
2. Diagramas de caso de uso, que exponen las interacciones entre un sistema y su entorno.
3. Diagramas de secuencias, que muestran las interacciones entre los actores y el sistema, y entre los componentes del sistema.
4. Diagramas de clase, que revelan las clases de objeto en el sistema y las asociaciones entre estas clases.
5. Diagramas de estado, que explican cómo reacciona el sistema frente a eventos internos y externos.

El detalle y el rigor de un modelo dependen de cómo lo use. Hay tres formas en que los modelos gráficos se emplean con frecuencia:

1. Como medio para facilitar la discusión sobre un sistema existente o propuesto.
2. Como una forma de documentar un sistema existente.
3. Como una descripción detallada del sistema que sirve para generar una implementación de sistema.

En el primer caso, el propósito del modelo es estimular la discusión entre los ingenieros de software que intervienen en el desarrollo del sistema. Los modelos pueden ser incompletos (siempre que cubran los puntos clave de la discusión) y utilizar de manera informal la notación de modelado. Cuando los modelos se usan como documentación, no tienen que estar completos, pues quizás usted sólo desee desarrollar modelos para algunas partes de un sistema. Sin embargo, estos modelos deben ser correctos: tienen que usar adecuadamente la notación y ser una descripción precisa del sistema.

En el tercer caso, en que los modelos se usan como parte de un proceso de desarrollo basado en modelo, los modelos de sistema deben ser completos y correctos. La razón para esto es que se usan como base para generar el código fuente del sistema.

## Modelos de contexto

En una primera etapa en la especificación de un sistema, debe decidir sobre las fronteras del sistema. Esto implica trabajar con los participantes del sistema para determinar cuál funcionalidad se incluirá en el sistema y cuál la ofrece el entorno del sistema. Tal vez decida que el apoyo automatizado para algunos procesos empresariales deba implementarse, pero otros deben ser procesos manuales o soportados por sistemas diferentes.

Debe buscar posibles interacciones en la funcionalidad con los sistemas existentes y determinar dónde tiene que implementarse nueva funcionalidad. Estas decisiones deben hacerse oportunamente durante el proceso, para limitar los costos del sistema, así como el tiempo necesario para comprender los requerimientos y el diseño del sistema.

En algunos casos, la frontera entre un sistema y su entorno es relativamente clara. Por ejemplo, donde un sistema automático sustituye un sistema manual o computarizado, el entorno del nuevo sistema, por lo general, es el mismo que el entorno del sistema existente.

En otros casos, existe más flexibilidad y usted es quien decide qué constituye la frontera entre el sistema y su entorno, durante el proceso de ingeniería de requerimientos. Por ejemplo, imagine que desarrolla la especificación para el sistema de información de pacientes para atención a la salud mental. Este sistema intenta manejar la información sobre los pacientes que asisten a clínicas de salud mental y los tratamientos que les prescriben. Al desarrollar la especificación para este sistema, debe decidir si el sistema tiene que enfocarse exclusivamente en reunir información de las consultas (junto con otros sistemas para recopilar información personal acerca de los pacientes), o si también es necesario que recopile datos personales acerca del paciente. La ventaja de apoyarse en otros sistemas para la información del paciente es que evita duplicar datos. Sin embargo, la principal desventaja es que usar otros sistemas haría más lento el acceso a la información. Si estos sistemas no están disponibles, entonces no pueden usarse en MHC-PMS.

La definición de frontera de un sistema no es un juicio libre de valor. Las preocupaciones sociales y organizacionales pueden significar que la posición de la frontera de un sistema se determine considerando factores no técnicos. Por ejemplo, una frontera de sistema puede colocarse deliberadamente, de modo que todo el proceso de análisis se realice en un sitio; puede elegirse de forma que sea innecesario consultar a un administrador particularmente difícil; puede situarse de manera que el costo del sistema aumente y la división de desarrollo del sistema deba, por lo tanto, expandirse al diseño y la implementación del sistema.

Una vez tomadas algunas decisiones sobre las fronteras del sistema, parte de la actividad de análisis es la definición de dicho contexto y las dependencias que un sistema tiene con su entorno. Normalmente, producir un modelo arquitectónico simple es el primer paso en esta actividad.

Los modelos de contexto, por lo general, muestran que el entorno incluye varios sistemas automatizados. Sin embargo, no presentan los tipos de relaciones entre los sistemas en el entorno y el sistema que se especifica. Los sistemas externos generan datos para el sistema o consumen datos del sistema. Pueden compartir datos con el sistema, conectarse directamente, a través de una red, o no conectarse en absoluto. Pueden estar físicamente juntos o ubicados en edificios separados. Todas estas relaciones llegan a afectar los requerimientos y el diseño del sistema a definir, por lo que deben tomarse en cuenta.

Por consiguiente, los modelos de contexto simples se usan junto con otros modelos, como los modelos de proceso. Éstos describen procesos humanos y automatizados que se usan en sistemas particulares de software.

## Modelos de interacción

Todos los sistemas incluyen interacciones de algún tipo. Éstas son interacciones del usuario, que implican entradas y salidas del usuario; interacciones entre el sistema a desarrollar y otros sistemas; o interacciones entre los componentes del sistema. El modelado de interacción del usuario es importante, pues ayuda a identificar los requerimientos del usuario. El modelado de la interacción sistema a sistema destaca los problemas de comunicación que se lleguen a presentar. El modelado de interacción de componentes ayuda a entender si es probable que

una estructura de un sistema propuesto obtenga el rendimiento y la confiabilidad requeridos por el sistema.

En el curso se cubren dos enfoques relacionados con el modelado de interacción:

1. Modelado de caso de uso, que se utiliza principalmente para modelar interacciones entre un sistema y actores externos (usuarios u otros sistemas).
2. Diagramas de secuencia, que se emplean para modelar interacciones entre componentes del sistema, aunque también pueden incluirse agentes externos.

Los modelos de caso de uso y los diagramas de secuencia presentan la interacción a diferentes niveles de detalle y, por lo tanto, es posible utilizarlos juntos. Los detalles de las interacciones que hay en un caso de uso de alto nivel se documentan en un diagrama de secuencia. UML también incluye diagramas de comunicación usados para modelar interacciones. Aquí no se analiza esto, ya que se trata de representaciones alternativas del diagrama de secuencia de mensajes. De hecho, algunas herramientas pueden generar un diagrama de comunicación a partir de un diagrama de secuencia.

### Modelos estructurales

Los modelos estructurales de software muestran la organización de un sistema, en términos de los componentes que constituyen dicho sistema y sus relaciones. Los modelos estructurales son modelos estáticos, que muestran la estructura del diseño del sistema, o modelos dinámicos, que revelan la organización del sistema cuando se ejecuta. No son lo mismo: la organización dinámica de un sistema como un conjunto de hilos en interacción tiende a ser muy diferente de un modelo estático de componentes del sistema.

Los modelos estructurales de un sistema se crean cuando se discute y diseña la arquitectura del sistema. El diseño arquitectónico es un tema particularmente importante en la ingeniería de software, y los diagramas UML de componente, de paquete y de implementación se utilizan cuando se presentan modelos arquitectónicos. Este curso se enfoca en el uso de diagramas de clase para modelar la estructura estática de las clases de objetos, en un modelo de dominio y en un sistema de software.

### Modelos de comportamiento

Los modelos de comportamiento son modelos dinámicos del sistema conforme se ejecutan. En ellos se muestra lo que sucede o lo que se supone que pasa cuando un sistema responde ante un estímulo de su entorno. Tales estímulos son de dos tipos:

1. Datos. Algunos datos que llegan se procesan por el sistema.
2. Eventos. Algunos eventos activan el procesamiento del sistema. Los eventos pueden tener datos asociados, pero esto no es siempre el caso.

Muchos sistemas empresariales son sistemas de procesamiento de datos que se activan principalmente por datos. Son controlados por la entrada de datos al sistema con relativamente poco procesamiento externo de eventos. Su procesamiento incluye una secuencia de acciones sobre dichos datos y la generación de una salida. Por ejemplo, un sistema de facturación telefónica aceptará información de las llamadas hechas por un cliente, calculará los costos de dichas llamadas y generará una factura para enviarla a dicho cliente. En contraste, los sistemas de tiempo real muchas veces están dirigidos por un evento con procesamiento de datos mínimo. Por ejemplo, un sistema de conmutación telefónico terrestre responde a eventos como



“receptor ocupado” al generar un tono de dial, o al presionar las teclas de un teléfono para la captura del número telefónico.

Los modelos dirigidos por datos muestran la secuencia de acciones involucradas en el procesamiento de datos de entrada, así como la generación de una salida asociada. Son particularmente útiles durante el análisis de requerimientos, pues sirven para mostrar procesamiento “extremo a extremo” en un sistema. Esto es, exhiben toda la secuencia de acciones que ocurren desde una entrada a procesar hasta la salida correspondiente, que es la respuesta del sistema. En UML esta secuencia puede ser modelada por diagramas de actividades como así también por diagramas de secuencia de mensajes.

El modelado dirigido por un evento muestra cómo responde un sistema a eventos externos e internos. Se basa en la suposición de que un sistema tiene un número finito de estados y que los eventos (estímulos) pueden causar una transición de un estado a otro. Por ejemplo, un sistema que controla una válvula puede moverse de un estado de “válvula abierta” a un estado de “válvula cerrada”, cuando recibe un comando operador (el estímulo). Esta visión de un sistema es adecuada particularmente para sistemas en tiempo real. UML soporta modelado basado en eventos usando diagramas de estado, que se fundamentaron en diagramas de estado propuestos por Harel a fines de los ´80s. Los diagramas de estado muestran estados y eventos del sistema que causan transiciones de un estado a otro. No exponen el flujo de datos dentro del sistema, pero suelen incluir información adicional acerca de los cálculos realizados en cada estado.

### Bibliografía

29148-2011 - ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes --Requirements engineering, 2011.

Arlow, J., I. Neustadt. UML 2.0. Anaya, 2006.

Brooks, Frederick. No Silver Bullet. Essence and Accidents of Software Engineering, Computer, 4, 10-19, 1987.

Davis, A. Software Requirements: Objects, Functions and States. Prentice Hall, 1993.

Requirements Working Group International Council on Systems Engineering (INCOSE). INCOSE-TP-2010-006-02.1. Guide for Writing Requirements, 2017, disponible en: <https://www.incose.org/>

Sommerville, I. Ingeniería de software, novena edición, Pearson, 2011.

Zielczynski, P. Requirements management using IBM Rational RequisitePro, IBM press, 2007.