Certified Professional for Requirements Engineering

Nivel Básico

Programa de estudios

Stan Bühne
Martin Glinz
Hans van Loenhoud
Stefan Staal

3.3.0 | 1 de abril de 2026

Condiciones de uso:

1. Los particulares y proveedores de formación pueden utilizar este programa de estudios como base para seminarios, siempre que se citen los derechos de autor y se incluyan en los materiales del seminario. Cualquiera que utilice este documento con fines publicitarios necesita autorización por escrito de IREB e.V. para ello.
2. Cualquier persona o colectivo puede utilizar este documento como base para artículos, libros u otra publicación siempre que se citen los derechos de autor y reconociendo al IREB e.V. como fuente y propietario de este documento en dichas publicaciones.

© IREB e.V.

Todos los derechos reservados. Ninguna parte de esta publicación puede ser reproducida, almacenada en un sistema de recuperación de información, o transmitida bajo ninguna modalidad o a través de ningún medio, electrónico, mecánico, fotocopia, grabación o de cualquier otra naturaleza sin el permiso previo y por escrito de los autores o del IREB e.V.

Agradecimientos

Este programa de estudios fue creado inicialmente en 2007 por Karol Frühauf, Emmerich Fuchs, Martin Glinz, Rainer Grau, Colin Hood, Frank Houdek, Peter Hruschka, Barbara Paech, Klaus Pohl y Chris Rupp. Estos autores contaron con el apoyo de los miembros de IREB Ian Alexander, Joseph Bruder, Samuel Fricker, Günter Halmans, Peter Jaeschke, Sven Krause, Steffen Lentz, Urte Pautz, Suzanne Robertson, Dirk Schüpferling, Johannes Staub y Thorsten Weyer.
La traducción al español del presente programa de estudios ha sido realizada por el equipo integrado por Sergio España, Xavier Franch (coordinador), Aurelio Gandarillas, Gustavo Márquez and Jose Luis de la Vara. Dado que el mismo programa se ha traducido, o se está traduciendo, a diversas lenguas, se ha seguido la directriz de respetar lo máximo posible el texto original. Aquellos términos cuya traducción puede ser motivo de controversia, se ha optado por introducir una nota a pie de página indicando otras aceptaciones comunes.

La versión 3 es una revisión importante, creada por Stan Bühne, Martin Glinz, Hans van Loenhoud y Stefan Staal. Estos autores contaron con el apoyo de Karol Frühauf, Rainer Grau, Kim Lauenroth, Chris Rupp y Camille Salinesi.

Durante esta revisión, Xavier Franch, Karol Frühauf, Rainer Grau, Frank Houdek y Thorsten Weyer proporcionaron retroalimentación. Wim Decoutere y Hans-Jörg Steffe proporcionaron retroalimentación adicional.

Las revisiones fueron realizadas por Christoph Ebert, Barbara Paech y Chris Rupp.

Aprobado para su publicación el 22 de julio de 2020 por el Consejo del IREB por recomendación de Xavier Franch y Frank Houdek.

Agradecemos a todos sus contribuciones.

Los derechos de autor © 2007 – 2026 de este syllabus se les atribuye a los autores mencionados con anterioridad. Los derechos de autor de este programa de estudios

pertenecen a los autores antes mencionados y se han transferido al International Requirements Engineering Board e.V. (IREB), Karlsruhe, Alemania.

Preámbulo

En el verano de 2017, desde IREB llevamos a cabo una encuesta para investigar la relevancia de la certificación vigente de nivel básico del Profesional Certificado en Ingeniería de Requisitos (CPRE) (versión 2.2). El objetivo de la encuesta fue obtener retroalimentación sobre la pertinencia práctica para el mercado de la certificación desde la perspectiva de los proveedores de formación, así como de los profesionales certificados con CPRE que trabajan como ingenieros de requisitos [MFeA2019]. La encuesta reflejó que, en general, el programa de estudios de nivel básico de la CPRE vigente, v2.2, todavía respondía a las necesidades más importantes del mercado e impartía a los candidatos los conocimientos pertinentes en materia de Ingeniería de Requisitos (IR). No obstante, recibimos retroalimentación de que varias técnicas mencionadas en el programa de estudios ya no se utilizan en la práctica, mientras que faltaban otras, en la búsqueda de un desarrollo más iterativo y adaptativo. Esta retroalimentación estaba alineada con la propia percepción del IREB sobre los cambios en el campo de la IR. Por lo tanto, hemos decidido llevar a cabo una revisión importante del programa de estudios del Nivel Básico del CPRE, eliminando el contenido desactualizado y añadiendo nuevos elementos. La versión 3 del programa de estudios refleja el estado de la IR contemporánea, abarcando tanto los enfoques dirigidos por un plan como los enfoques ágiles para especificar y gestionar requisitos.

Se espera que los candidatos que deseen obtener la certificación como CPRE de acuerdo a este programa de estudios dispongan de conocimientos básicos acerca del desarrollo de sistemas con enfoques dirigidos por un plan y ágiles.

Objetivo del Documento

Este programa de estudios define los objetivos educativos y un resumen del contenido educativo para el Foundation Level de la certificación Certified Professional for Requirements Engineering establecida por el International Requirements Engineering Board (IREB). El programa de estudios puede ser utilizado por parte de los proveedores de formación como base para crear sus materiales para cursos. Los estudiantes pueden utilizar este documento para prepararse para el examen.

Contenido del Programa de Estudios

Este programa de estudios de nivel básico aborda las necesidades de aquellas personas involucradas en la temática de Ingeniería de Requisitos. Esto incluye a personas con roles tales como Ingeniero de Requisitos, analista de negocios, analista de sistemas, dueño de producto o gestor de producto, desarrollador, gestor de proyectos o de TI, o experto en el dominio.

Este programa de estudios y el manual relacionado utilizan la abreviatura "IR" para Ingeniería de Requisitos.

Alcance del Contenido

El Nivel Básico del CPRE comunica principios básicos que son igualmente válidos para cualquier tipo de sistema (por ejemplo, aplicaciones para dispositivos móviles, sistemas de información o sistemas ciberfísicos). Además, el nivel básico del CPRE no asume ningún proceso de desarrollo específico, ni está orientado a un dominio de aplicación específico. Los proveedores de formación pueden ofrecer una formación que se centre en tipos específicos de sistemas, procesos o dominios de aplicación, siempre y cuando se cubran plenamente los objetivos educativos de este programa de estudios.

Grado de Detalle

El grado de detalle de este programa de estudios permite una formación y evaluación consistentes a nivel internacional. Para lograr este objetivo, el programa de estudios contiene los siguientes elementos:

- Objetivos educativos generales
- Contenidos con la descripción de los objetivos educativos
- Referencias bibliográficas complementarias (cuando sea necesario)

Formulaciones sensibles al género

En este documento, nos hemos abstenido deliberadamente de utilizar formulaciones específicas de género.

No hace falta decir que en IREB apoyamos las formulaciones sensibles al género. Sin embargo, el objetivo primordial de este documento es presentar y comunicar el contenido de forma clara y precisa, para ayudar a los lectores a mantener el enfoque en el mensaje subyacente. Por ello, en este documento usamos deliberadamente solo la forma masculina de las personas, pues el uso de ambas formas podría dificultar el entendimiento de ciertos textos complejos.

La decisión adoptada por IREB en referencia a las formulaciones sensibles al género no debe tomarse como una expresión de falta de respeto.

Objetivos Educativos / Niveles Cognitivos

A todos los módulos y objetivos educativos de este programa se les asigna un nivel cognitivo. Los niveles se clasifican del siguiente modo:

- **N1: Saber** (describir, enumerar, caracterizar, reconocer, nombrar, recordar, ...) — recordar o recuperar material previamente aprendido.
- **N2: Comprender** (explicar, interpretar, completar, resumir, justificar, clasificar, comparar, ...) — comprender/construir el significado de un material o situaciones dadas.
- **N3: Aplicar** (especificar, escribir, diseñar, desarrollar, implementar, ...) — aplicar los conocimientos y habilidades en situaciones dadas.

Los niveles superiores incluyen los niveles inferiores. Obsérvese que todos los términos del glosario que se designen como términos básicos deben ser conocidos (N1), aunque no se mencionen explícitamente en los objetivos educativos. El glosario se puede descargar en la página web del IREB: https://cpre.ireb.org/downloads-and-resources/downloads#cpre-glossary

Estructura del Programa de Estudios

El programa de estudios está formado por siete capítulos. Cada capítulo cubre una unidad educativa (UE). Los títulos de los capítulos principales incluyen su nivel cognitivo, que es el nivel más alto de sus secciones. La duración indica el tiempo que un curso de formación debería invertir en ese capítulo. Los proveedors de formación son libres de asignar más tiempo, pero deberían mantener las proporciones entre las UEs. Los términos importantes utilizados en un capítulo se enumeran al principio del mismo.

Ejemplo

UE 4 Prácticas para la elaboración de requisitos (N3)

Duración: 4 horas y 30 minutos

Términos: Fuente de requisitos, frontera del sistema, contexto del sistema, educción de requisitos, negociación de requisitos, validación de requisitos, parte implicada, modelo de Kano, conflicto, resolución de conflictos

En este ejemplo se muestra que el Capítulo 4 contiene objetivos educativos en el nivel N3 y que se deberían dedicar cuatro horas y media a la enseñanza del material de este capítulo.

Cada capítulo contiene secciones. Sus títulos también contienen el nivel cognitivo de su contenido.

Los objetivos educativos (OE) son enumerados antes del texto. El esquema de numeración de estos objetivos muestra la sección a la que pertenecen. Por ejemplo, el objetivo educativo OE 4.2.1 está descrito en la Sección 4.2.

Orden de los temas en el programa de estudios

El orden de los capítulos de este programa de estudios constituye un orden lógico de temas. Sin embargo, los temas no tienen que ser enseñados exactamente en este orden. Los proveedores de formación tienen libertad para enseñar el material en cualquier orden (incluida la intercalación de temas de diferentes UEs) que consideren apropiado en el contexto de su capacitación y que se ajuste a sus conceptos didácticos.

El Examen

Este programa de estudios es la base del examen para el certificado de Nivel Básico de CPRE.

> Las preguntas del examen pueden basarse en el contenido de más de un capítulo del programa de estudios. Todos los capítulos (UE 1 – UE 7) pueden ser objeto de evaluación.

El examen consiste en un test de selección múltiple.

Los exámenes se pueden realizar inmediatamente después de completar los cursos de formación, pero también de manera independiente de los cursos (por ejemplo, en un centro de evaluación). En la página web https://www.ireb.org se puede encontrar una lista de los organismos de certificación autorizados por el IREB.

Historial de versiones

Versión	Fecha	Comentario
2.1.4	Noviembre 201	Versión inicial, basado en la versión inglesa 2.
2.1.5	Junio 2012	El término diagrama de contextoañadido a la UE 6.6
2.2.0	Junio 2012	Corrección de erratas y errores gramaticales
3.0.0	Enero 1, 2022	Segunda actualización importante para reflejar el estado de la IR contemporánea, cubriendo tanto las propuestas dirigidas por un plan como las propuestas ágiles para la especificación y gestión de requisitos.
3.1.0	Septiembre 1, 2022	Errores ortográficos corregidos y referencias completadas en todo el documento para mejorar su legibilidad UE 1: Objetivo educativo OE 1.3.2 movido a OE 1.1.2 OE 1.2.2 y 1.3.1 actualizados UE 3 Objetivo educativo OE 3.1.3 actualizado UE 3.1.2: Párrafo sobre niveles de abstracción reformulado UE 3.4: Todos los tipos de modelos que no necesitan ser aplicados en el Nivel Básico han sido movidos a una nueva subsección 3.4.6 UE 3.6: Cabecera actualizada a "Documentos de Requisitos y Estructuras de Documentación" UE 4 UE 4.1: La descripción de las partes interesadas y de sus roles se ha hecho más precisa UE 4.2: La introducción y justificación para las técnicas de diseño y generación de ideas se ha actualizado para que sean más precisas UE 4.3: Cabecera actualizada a "Resolución de conflictos relativos a los requisitos" UE 6 OE 6.3.1 y OE 6.5.2 se han modificado ligeramente

3.1.1	Enero 1, 2024	Diseño corporativo actualizado
3.2.0	Febrero 26, 2024	Correcciones de maquetación para el nuevo diseño corporativo
3.3.0	Abril 1, 2026	Cambios relacionados con el contenido ▪ EO 3.4.3 y EU 3.4.6: Se ha eliminado el modelo de interacción de los objetivos de aprendizaje. ▪ EU 3.8: se ha aclarado que los criterios de calidad mencionados son criterios recomendados y no constituyen una lista exhaustiva. Cambios editoriales ▪ Actualizaciones editoriales de las referencias y de la sección 4.1, incluida la adición de una referencia retrospectiva al Principio 3. ▪ Se ha actualizado la sección de Referencias y se han verificado todos los hipervínculos.

Contenido

Contenido 9

1 Introducción y visión general de la ingeniería de requisitos (N2) 12

1.1 Ingeniería de Requisitos: Qué (N1) 12

1.2 Ingeniería de Requisitos: Por qué (N2) 13

1.3 Ingeniería de Requisitos: Dónde (N2) 13

1.4 Ingeniería de Requisitos: Cómo (N1) 13

1.5 El rol y las tareas de un ingeniero de requisitos (N1) 14

1.6 Qué aprender sobre la Ingeniería de Requisitos (N1) 14

2 Principios Fundamentales de la Ingeniería de Requisitos (N2) 15

2.1 Descripción General de los Principios (N1) 15

2.2 Explicación de los Principios (N2) 15

3 Productos del Trabajo y Prácticas de Documentación (N3) . 20

3.1 Productos del Trabajo en la Ingeniería de Requisitos (N2) 21

3.1.1 Características de los Productos del Trabajo (N1) 21

3.1.2 Niveles de abstracción (N2) 22

3.1.3 Nivel de Detalle (N2) 22

3.1.4 Aspectos a Considerar en los Productos del Trabajo (N1) 23

3.1.5 Directrices Generales de Documentación (N1) 23

3.1.6 Planificar los Productos del Trabajo a Utilizar (N1) 24

3.2 Productos del Trabajo Basados en Lenguaje Natural (N2) 24

3.3 Productos del Trabajo Basados en Plantillas (N3) 25

3.4 Productos del Trabajo Basados en Modelos (N3) 26

3.4.1 El Papel de los Modelos en la Ingeniería de Requisitos (N2) 26

3.4.2 Modelando el Contexto (N2) 27
3.4.3 Modelando la Estructura y los Datos (N3) 28
3.4.4 Modelando Función y Flujo (N3) 28
3.4.5 Modelando Estado y Comportamiento (N2) 29
3.4.6 Otros tipos de modelos en la ingeniería de requisitos (L1) 29
3.5 Glosarios (N2) 30
3.6 Documentos de Requisitos y Estructuras de Documentación (N2) 30
3.7 Prototipos en Ingeniería de Requisitos (N1) 31
3.8 Criterios de Calidad para los Productos del Trabajo y los Requisitos (N1) 32
4 Prácticas para la Elaboración de Requisitos (N3) 33
4.1 Fuentes para Requisitos (N3) 33
4.2 Educción de Requisitos (N2) 35
4.3 Resolución de conflictos relativos a los requisitos (N2) 36
4.4 Validación de Requisitos (N2) 37
5 Estructura de Proceso y Trabajo (N3) 39
5.1 Factores de Influencia (N2) 39
5.2 Facetas del Proceso de Ingeniería de Requisitos (N2) 40
5.3 Configuración de un Proceso de Ingeniería de Requisitos (N3) 42
6 Prácticas de Gestión de Requisitos (N2) 44
6.1 ¿Qué es la Gestión de Requisitos? (N1) 44
6.2 Gestión del Ciclo de Vida (N2) 44
6.3 Control de Versiones (N2) 45
6.4 Configuraciones y Líneas Base (N1) 45
6.5 Atributos y Vistas (N2) 45
6.6 Trazabilidad (N1) 46

6.7 Tratamiento del Cambio (N1) 46

6.8 Priorización (N1) 47

7 Apoyo de Herramientas (N2) 48

7.1 Herramientas en la Ingeniería de Requisitos (N1) 48

7.2 Introducción de Herramientas (N2) 49

Referencias 50

1 Introducción y visión general de la ingeniería de requisitos (N2)

Objetivo: Saber de qué se trata la IR y entender el valor de la IR

Duración: 1 hora

Términos: Requisito, especificación de requisitos, Ingeniería de Requisitos (IR), parte interesada, sistema, Ingeniero de Requisitos

Objetivos educativos

OE 1.1.1 Recordar la terminología fundamental (N1)
OE 1.1.2 Comprender los diferentes tipos de requisitos (N2)
OE 1.2.1 Explicar el valor de la IR (N2)
OE 1.2.2 Enumerar los síntomas de una IR inadecuada (N1)
OE 1.3.1 Conocer dónde se puede aplicar la IR y dónde se dan los requisitos (N1)
OE 1.4.1 Conocer las principales tareas de la IR y que un proceso de IR tiene que ser adaptado para realizarlas (N1)
OE 1.5.1 Caracterizar el rol y las tareas de un Ingeniero de Requisitos (N1)
OE 1.6.1 Recordar lo que necesita aprender un Ingeniero de Requisitos (N1)

1.1 Ingeniería de Requisitos: Qué (N1)

Las personas y las organizaciones tienen deseos y necesidades de que se construyan nuevas cosas o evolucionen las ya existentes. A esas necesidades las llamamos *requisitos*.

Las cosas que se construyen o evolucionan pueden ser:

- *Los productos* proporcionados a los clientes
- *Los servicios* puestos a disposición de los clientes
- Cualquier otro *entregable* tales como dispositivos, procedimientos o herramientas que ayuden a las personas y organizaciones a alcanzar un objetivo específico
- *Composiciones* o *componentes* de productos, servicios u otros entregables

Todas estas cosas pueden ser consideradas como *sistemas*. En este programa de estudios, utilizamos el término *sistema* para denotar todo tipo de cosas para las que *las partes implicadas* tienen requisitos. *Las partes implicadas* son personas u organizaciones que influyen en los requisitos de un sistema o que se ven afectadas por dicho sistema.

El objetivo de la IR es especificar y gestionar los requisitos de los sistemas de manera que los sistemas implementados y desplegados satisfagan los deseos y necesidades de las partes interesadas.

En IR, distinguimos tres tipos de requisitos [Glin2025]:

- Los r*equisitos funcionales* se refieren a un resultado o comportamiento que será proporcionado por una función de un sistema. Esto incluye los requisitos de datos o la interacción de un sistema con su entorno.

- Los *requisitos de calidad* se refieren a cuestiones de calidad que no están cubiertas por los requisitos funcionales, como el rendimiento, la disponibilidad, la seguridad o la fiabilidad.
- *Las restricciones* son requisitos que limitan el espacio de solución más allá de lo necesario para cumplir los requisitos funcionales y de calidad dados.

1.2 Ingeniería de Requisitos: Por qué (N2)

Una IR adecuada añade *valor* en el proceso de desarrollo y evolución de un sistema:

- Reduciendo el riesgo de desarrollar un sistema incorrecto
- Mejorando la comprensión del problema
- Siendo la base para estimar el esfuerzo y el coste del desarrollo
- Siendo prerrequisito para probar el sistema

Los síntomas típicos de una IR inapropiada son la ausencia, poca claridad o incorrección de los requisitos. Esto es particularmente debido a:

- Apresurarse a construir directamente el sistema
- Problemas de comunicación entre las partes involucradas
- La suposición de que los requisitos son evidentes por sí mismos
- Educación y habilidades inadecuadas en materia de IR

1.3 Ingeniería de Requisitos: Dónde (N2)

La IR puede aplicarse a los requisitos de cualquier tipo de sistema. Sin embargo, el caso de aplicación dominante de la IR en la actualidad está representado por los sistemas en los que el software desempeña un papel importante. Esos sistemas suelen constar de componentes de software, elementos físicos y elementos organizativos.

Los requisitos pueden darse como:

- *Requisitos del sistema* — lo que un sistema debe hacer
- *Requisitos de las partes implicadas* — lo que las partes implicadas quieren desde su perspectiva
- *Requisitos de los usuarios* — lo que los usuarios quieren desde su perspectiva
- *Requisitos del dominio* — propiedades del dominio requeridas
- *Requisitos de negocio* — metas de negocio, objetivos y necesidades de una organización

1.4 Ingeniería de Requisitos: Cómo (N1)

Las principales tareas de la IR son la educción (4.2), la documentación (3), la validación (4.4), y la gestión (6) de los requisitos. El apoyo de herramientas (7) puede ayudar a realizar estas tareas. El análisis de los requisitos y la resolución de conflictos entre requisitos (4.3) se consideran parte de la educción. Para realizar correctamente las actividades de la IR, un proceso de IR adecuado debe ser adaptado a partir de una amplia gama de posibilidades (5).

1.5 El rol y las tareas de un ingeniero de requisitos (N1)

Típicamente, "Ingeniero de Requisitos" no es un puesto de trabajo, sino un *rol* que las personas desempeñan y que:

- Educe, documenta, valida y/o gestiona requisitos como parte de sus obligaciones.
- Tiene un profundo conocimiento de la IR.
- Puede salvar la brecha entre el problema y sus posibles soluciones.

En la práctica, los analistas de negocios, los especialistas en aplicaciones, los dueños de producto, los ingenieros de sistemas e incluso los desarrolladores actúan en el rol de Ingeniero de Requisitos.

1.6 Qué aprender sobre la Ingeniería de Requisitos (N1)

Este programa de estudios cubre las habilidades fundamentales que un Ingeniero de Requisitos debe aprender. Abarca los principios fundamentales de la IR (2), cómo documentar los requisitos de diversas formas (3), cómo elaborar los requisitos con diversas prácticas (4), cómo definir y trabajar con procesos de IR adecuados (5), cómo gestionar los requisitos existentes (6) y cómo utilizar herramientas de apoyo (7).

2 Principios Fundamentales de la Ingeniería de Requisitos (N2)

Objetivo: Conocer y comprender los principios de la IR

Duración: 1 hora y 30 minutos

Términos: Contexto, requisito, Ingeniería de Requisitos (IR), parte implicada, entendimiento compartido, validación

Objetivos educativos

OE 2.1.1 Enumerar los principios de la IR (N1)
OE 2.2.1 Recordar los términos asociados a los principios (N1)
OE 2.2.2 Explicar los principios y las razones por los que son importantes (N2)

2.1 Descripción General de los Principios (N1)

La IR se rige por un conjunto de principios que se aplican a todas las tareas, actividades y prácticas de la IR. Los nueve principios siguientes constituyen la base de las prácticas presentadas en los capítulos sucesivos de este programa de estudio.

1. Orientación al Valor: Los requisitos son un medio para conseguir un fin, no un fin en sí mismo
2. Partes implicadas: La IR consiste en satisfacer los deseos y necesidades de las partes implicadas
3. Entendimiento compartido: Es imposible el desarrollo de sistemas exitosos sin una base común
4. Contexto: Los sistemas no se pueden entender de forma aislada
5. Problema – Requisito – Solución: Una tripleta inevitablemente entrelazada
6. Validación: Los requisitos no validados son inútiles
7. Evolución: El cambio en los requisitos no es un accidente, sino el caso habitual
8. Innovación: Más de lo mismo no es suficiente
9. Trabajo sistemático y disciplinado: No se puede prescindir de la IR

2.2 Explicación de los Principios (N2)

Principio 1 - Orientación al Valor: Los requisitos son un medio para un fin, no un fin en sí mismo

El valor de un requisito es igual a su beneficio menos el coste de educir, documentar, validar y gestionar el requisito. El beneficio de un requisito es el grado con el que contribuye a:

- Construir sistemas que satisfagan los deseos y necesidades de sus partes implicadas.
- Reducir el riesgo de fallos y reconstrucciones costosas al desarrollar el sistema.

Principio 2 - Partes Implicadas: La IR consiste en satisfacer los deseos y necesidades de las partes implicadas

Dado que la IR consiste en comprender los deseos y necesidades de las partes implicadas, tratar a las partes implicadas de forma adecuada es una tarea esencial de la IR. Cada parte implicada tiene un rol en el contexto del sistema que se va a construir — por ejemplo, como usuario, cliente, consumidor, operador o regulador. Una parte implicada también puede tener más de un rol al mismo tiempo. En el caso de roles de partes implicadas con demasiados individuos o cuando se desconocen los individuos, pueden definirse descripciones arquetípicas ficticias conocidas como *personas* en su sustitución. No es suficiente considerar sólo los requisitos de los usuarios finales o clientes. Hacer esto significaría que podríamos pasar por alto requisitos críticos de otras partes implicadas. Los usuarios que proporcionan retroalimentación sobre un sistema en uso también deben ser considerados partes implicadas.

Las partes implicadas pueden tener necesidades y puntos de vista diferentes, lo que puede dar lugar a requisitos conflictivos. Es tarea de la IR identificar y resolver tales conflictos.

Involucrar a la gente adecuada en los roles de las partes implicadas es crucial para el éxito de la IR. Las prácticas para identificar, priorizar y trabajar con las partes implicadas se describen en el apartado 4.

Principio 3 - Entendimiento Compartido: Es imposible el desarrollo de sistemas exitosos sin una base común

La IR crea, promueve y asegura el entendimiento compartido entre las partes involucradas: las partes implicadas, Ingenieros de Requisitos y desarrolladores. Hay dos formas de entendimiento compartido:

- *Entendimiento compartido explícito* (logrado mediante requisitos documentados y acordados)
- *Entendimiento compartido implícito* (basado en el conocimiento compartido de las necesidades, las visiones, el contexto, etc.)

El conocimiento del dominio, la colaboración exitosa previa, la cultura y los valores compartidos, y la confianza mutua son factores que facilitan el entendimiento compartido, mientras que la distancia geográfica, la subcontratación o los grandes equipos con una alta rotación son obstáculos.

Entre las prácticas contrastadas para lograr un entendimiento compartido se encuentran la creación de glosarios (3.5), la creación de prototipos (3.7) o la utilización de un sistema existente como punto de referencia. Las prácticas para evaluar el entendimiento compartido incluyen ejemplos de los resultados esperados, la exploración de prototipos o la estimación del coste de la implementación de un requisito. La práctica más importante para

reducir el impacto de los malentendidos es utilizar un proceso con bucles de retroalimentación cortos (5).

Principio 4 - Contexto: Los sistemas no se pueden entender de forma aislada

Los sistemas se encuentran incrustados en un *contexto*. Es imposible especificar un sistema de forma adecuada si no se comprende ese contexto. En la IR, el contexto de un sistema es la parte del entorno de un sistema que es relevante para comprender el sistema y sus requisitos. La *frontera del sistema* es la frontera entre un sistema y su contexto circundante. Inicialmente, la frontera del sistema a menudo no está clara, e incluso puede cambiar con el transcurso del tiempo.

Determinar la frontera del sistema y definir las interfaces externas entre el sistema y los elementos del contexto con los que interactúa son auténticas tareas de la IR. Al mismo tiempo, es necesario determinar el *alcance* del sistema — es decir, el rango de cosas a las que se puede dar forma y diseñar al desarrollar el sistema. También debemos considerar la así denominada *frontera de contexto* que separa la parte del entorno de un sistema relevante para la IR del resto del mundo.

Al especificar un sistema, no es suficiente considerar sólo los requisitos dentro de la frontera del sistema. En la IR, también deben tenerse en cuenta:

- Cambios en el contexto que pueden afectar a los requisitos del sistema.
- Requisitos del mundo real relevantes para el sistema (y cómo mapearlos en los requisitos del sistema).
- Supuestos de contexto que deben satisfacerse para que el sistema funcione y cumpla con los requisitos del mundo real.

Principio 5 - Problema - Requisito - Solución: Una tripleta inevitablemente entrelazada

Se produce un *problema* cuando las partes implicadas no están satisfechas con la situación tal y cómo es. Los *requisitos* recogen lo que las partes implicadas necesitan para eliminar el problema o para simplificarlo. Un sistema socio-técnico que satisface estos requisitos constituye una *solución*.

Problemas, requisitos y soluciones no se presentan necesariamente en este orden. Las ideas relativas a la solución pueden crear necesidades de usuario que deben ser elaboradas como requisitos e implementadas en una solución real. Este es el caso típico de la innovación.

- Los problemas, los requisitos y las soluciones están estrechamente entrelazados: no pueden abordarse de forma aislada.

- No obstante, los Ingenieros de Requisitos tienen como objetivo separar los problemas, los requisitos y las soluciones entre sí en la medida de lo posible al pensar, comunicar y documentar. Esta separación de asuntos hace que las tareas de la IR sean más fáciles de tratar.

Principio 6 - Validación: Unos requisitos no validados son inútiles

Eventualmente, tenemos que validar que el sistema desplegado satisface los deseos y necesidades de las partes implicadas. Para controlar el riesgo de tener a las partes implicadas insatisfechas desde el principio, la validación de los requisitos debe comenzar desde la misma IR. Tenemos que comprobar si:

- Se ha logrado un acuerdo entre las partes implicadas respecto a los requisitos,
- Los deseos y necesidades de las partes implicadas están cubiertos de forma adecuada por los requisitos,
- Los supuestos de contexto (véase el Principio 4 anterior) son razonables.

Las prácticas para validar requisitos se desarrollan en 4.4.

Principio 7 - Evolución: El cambio en los requisitos no es un accidente, sino el caso habitual

Los sistemas y sus requisitos están sujetos a *evolución*. Esto significa que cambian de forma constante. Las solicitudes de cambio de un requisito o de un conjunto de requisitos de un sistema pueden estar causadas, por ejemplo, por:

- Cambios en los procesos de negocio
- Lanzamiento de nuevos productos o servicios por parte de la competencia
- Cambios en las prioridades u opiniones de los clientes
- Cambios en la tecnología
- Cambios legales o normativos
- Retroalimentación de los usuarios del sistema pidiendo cambiar características ya existentes o introducir nuevas características

Además, los requisitos pueden cambiar debido a la retroalimentación de las partes implicadas al validar los requisitos, debido a la detección de defectos en los requisitos educidos previamente, o debido a cambios en las necesidades de las partes implicadas.

En consecuencia, los Ingenieros de Requisitos deben perseguir dos objetivos aparentemente contradictorios:

- Permitir cambios en los requisitos
- Mantener los requisitos estables

Los detalles sobre cómo lograr esto se abordan en 6.7.

Principio 8 - Innovación: Más de lo mismo no es suficiente

Dar a las partes implicadas exactamente lo que quieren hace que se pierda la oportunidad de construir sistemas que satisfagan las necesidades de las partes implicadas mejor de lo que esperan. Una buena IR se esfuerza no sólo por satisfacer a las partes implicadas, sino también por hacer que se sientan satisfechas, entusiasmadas o seguras. De esto trata, en última instancia, la innovación.

La IR da forma a sistemas innovadores:

- A pequeña escala, buscando nuevas características que resulten estimulantes y fáciles de usar.
- A gran escala, buscando nuevas ideas disruptivas.

En 4.2 se discuten varias técnicas para fomentar la innovación en la IR.

Principio 9 - Trabajo sistemático y riguroso: No se puede prescindir de la IR

Necesitamos emplear procesos y prácticas adecuados para educir, documentar, validar y gestionar requisitos de forma sistemática, independientemente del proceso de desarrollo que se esté utilizando. Incluso cuando un sistema se desarrolla de forma ad hoc, un enfoque sistemático y disciplinado de la IR mejora la calidad del sistema resultante.

No hay un solo proceso o práctica de la IR que funcione bien en toda situación dada o, al menos, en la mayoría de las situaciones. El trabajo sistemático y riguroso significa que los Ingenieros de Requisitos:

- Adaptan sus procesos y prácticas al problema, el contexto y el entorno dados.
- No siempre utilizan el mismo proceso y conjunto de prácticas.
- No reutilizan procesos y prácticas de trabajos de IR exitosos del pasado sin una reflexión previa.

Para cada empeño de IR, los procesos, prácticas y productos del trabajo deben ser elegidos para que se adecuen mejor a la situación específica. Los detalles se abordan en 3, 4, 5 y 6.

3 Productos del Trabajo y Prácticas de Documentación (N3)

Objetivo: Comprender el papel fundamental de los productos del trabajo en la IR y ser capaces de crear productos del trabajo

Duración: 7 horas

Términos: Producto del trabajo, productos del trabajo basados en lenguaje natural, productos del trabajo basados en plantillas, productos del trabajo basados en modelos, glosario, criterios de calidad, especificación de requisitos

Objetivos educativos

OE 3.1.1 Conocer las características de los productos del trabajo de la IR y enumerar los tipos de productos del trabajo frecuentemente utilizados (N1)

OE 3.1.2 Saber para qué se puede utilizar cada producto del trabajo y conocer el tiempo de vida de los productos del trabajo (N1)

OE 3.1.3 Explicar los diferentes niveles de abstracción para los requisitos, incluyendo cómo elegir los niveles de abstracción y los niveles de detalle apropiados (N2)

OE 3.1.4 Conocer los aspectos a considerar en los productos del trabajo y las interrelaciones entre estos aspectos (N1)

OE 3.1.5 Enumerar las directrices generales de documentación (N1)

OE 3.1.6 Describir por qué es conveniente planificar los productos del trabajo que se utilizarán (N1)

OE 3.2.1 Conocer los productos del trabajo basados en lenguaje natural, y sus ventajas y desventajas (N1)

OE 3.2.2 Explicar las reglas para redactar buenos requisitos basados en el lenguaje natural (N2)

OE. 3.3.1 Conocer las categorías de productos del trabajo basados en plantillas, y sus ventajas y desventajas (N1)

OE 3.3.2 Especificar un requisito individual y una historia de usuario utilizando una plantilla de frases (N3)

OE 3.3.3 Especificar un caso de uso utilizando una plantilla de formulario (N3)

OE 3.4.1 Comprender el papel, el propósito y el uso de los modelos en la IR (N2)

OE 3.4.2 Comprender las ventajas y limitaciones del modelado en IR (N2)

OE 3.4.3 Conocer los términos modelo, lenguaje de modelado, modelo de actividades, diagrama de actividades, modelo de clases, diagrama de clases, modelo de contexto, diagrama de contexto, modelo de dominio, modelo de objetivos, modelo de procesos, diagrama de secuencias, gráfico de estados, máquina de estados, diagrama de máquina de estados, caso de uso, diagrama de casos de uso (N1)

OE 3.4.4 Comprender cómo seleccionar un tipo de modelo apropiado para especificar los requisitos en una situación determinada (N2)

OE 3.4.5 Comprender e interpretar modelos simples, escritos en UML cuando sea aplicable, de los tipos siguientes: modelos de contexto, casos de uso y diagramas de casos de uso, modelos de dominio, modelos de clases, modelos de actividades, modelos de procesos y gráficos de estados (N2)

OE 3.4.6 Especificar un modelo simple de los datos de un sistema o de los objetos de un dominio, utilizando un diagrama de clases UML (N3)

OE 3.4.7 Especificar una función del sistema simple o un proceso de negocio, mediante un diagrama de actividades UML (N3)

OE 3.5.1 Explicar el propósito de los glosarios y cómo crear uno (N2)

OE 3.6.1 Conocer documentos de especificación de requisitos de uso frecuente (N1)
OE 3.6.2 Explicar los propósitos de cada estructura de documento y los criterios para estructurar documentos (N2)
OE 3.7.1 Conocer los diferentes tipos de prototipos y para qué se utilizan (N1)
OE 3.8.1 Conocer los criterios de calidad para requisitos individuales (N1)
OE 3.8.2 Conocer los criterios de calidad para productos del trabajo (N1)

3.1 Productos del Trabajo en la Ingeniería de Requisitos (N2)

Un producto del trabajo es un resultado intermedio o final, que se genera en un proceso de trabajo y que queda registrado. Hay una variedad de productos del trabajo en la IR, que van, por ejemplo, desde bocetos gráficos de corta duración, pasando por colecciones de historias de usuario que van evolucionando, hasta documentos de especificación de requisitos contractuales con cientos de páginas que son entregados oficialmente.

3.1.1 Características de los Productos del Trabajo (N1)

Los productos del trabajo se caracterizan por su propósito, representación, tamaño y tiempo de vida. Los productos del trabajo siguientes se utilizan con frecuencia en la práctica, para los fines indicados. Tenga en cuenta que un producto del trabajo puede contener otros productos del trabajo.

- Los productos del trabajo para un solo requisito incluyen requisitos individuales e historias de usuario.
- Los productos del trabajo para un conjunto coherente de requisitos incluyen casos de uso, modelos gráficos de distinto tipo (3.4), descripciones de tareas, descripciones de interfaces externas y épicas.
- Los productos del trabajo que constituyen documentos o estructuras de documentación completos incluyen las especificaciones de los requisitos del sistema, los backlogs del producto y del sprint, y los story maps.
- Otros productos del trabajo incluyen glosarios, notas textuales, bocetos gráficos, y prototipos.

Los productos del trabajo pueden *representarse* de varias formas:

- Basado en lenguaje natural (3.2)
- Basado en plantillas (3.3)
- Basado en modelos (3.4)
- Otras representaciones, como dibujos o prototipos (3.7)

La mayoría de los productos del trabajo se *almacenan* electrónicamente en forma de archivos, en bases de datos o en herramientas de IR. Los productos del trabajo informales y temporales también pueden almacenarse en otros medios — por ejemplo, en papel o en notas post-it en una pizarra Kanban.

Al observar el tiempo de vida de los productos del trabajo, distinguimos tres categorías:

- *Productos del trabajo temporales:* creados para apoyar la comunicación y crear un entendimiento compartido.

- *Productos del trabajo en evolución:* típicamente crecen en varias iteraciones a lo largo del tiempo; necesitan algunos metadatos (6.5); puede aplicarse control de cambios.
- *Productos del trabajo duraderos:* se han *entregado* o forman parte de una *línea base*; necesitan un conjunto completo de metadatos (6.5); debe seguirse un proceso de cambio (6.3, 6.4).

Un producto del trabajo temporal puede convertirse en un producto del trabajo en evolución (conservándolo y añadiendo metadatos). Análogamente, un producto del trabajo temporal o en evolución puede convertirse en duradero al ser incorporado en una línea base o entregado.

3.1.2 Niveles de abstracción (N2)

Los requisitos suelen existir en muchos *niveles de abstracción diferentes* — desde, por ejemplo, requisitos de alto nivel para un nuevo proceso de negocio, hasta requisitos en un nivel muy detallado, como la reacción de un componente de software específico ante un evento excepcional.

La elección del nivel de abstracción adecuado depende del tema a especificar y del objetivo de la especificación. Sin embargo, es importante no mezclar requisitos que estén en diferentes niveles de abstracción. En los productos del trabajo pequeños y medianos, los requisitos deben estar más o menos al mismo nivel de abstracción.

En los productos del trabajo de gran envergadura, como la especificación de los requisitos del sistema, los requisitos de los diferentes niveles de abstracción deben mantenerse separados estructurando la especificación en consonancia (3.6). Un requisito de alto nivel de abstracción puede refinarse en varios requisitos detallados a niveles más bajos y concretos.

Cuando los requisitos de negocio y los requisitos de partes interesadas se expresan en productos del trabajo duraderos —como las especificaciones de requisitos de negocio, especificaciones de requisitos de partes interesadas, o documentos de visión—, preceden a la especificación de los requisitos del sistema. En otros proyectos, los requisitos de negocio, los requisitos de partes interesadas, y los requisitos del sistema pueden evolucionar conjuntamente.

3.1.3 Nivel de Detalle (N2)

El nivel de *detalle* con el que se debe especificar los requisitos depende de varios factores, en particular:

- El problema y el contexto de desarrollo
- El grado de entendimiento compartido del problema
- El grado de libertad que se deja a los diseñadores y programadores
- Disponibilidad de una rápida retroalimentación de las partes interesadas durante el diseño y la implementación
- Coste vs. valor de una especificación detallada
- Estándares impuestos y restricciones regulatorias

Cuanto mayor es el nivel de detalle de los requisitos especificados, menor será el riesgo de obtener eventualmente algo inesperado o no especificado. Sin embargo, el coste de la especificación aumenta a medida que aumenta el nivel de detalle.

3.1.4 Aspectos a Considerar en los Productos del Trabajo (N1)

Cuando se especifican requisitos en los productos del trabajo, es necesario considerar diferentes aspectos.

1. Los requisitos se clasifican por su tipo (1.1) en:
   a) Requisitos funcionales
   b) Requisitos de calidad
   c) Restricciones
2. Los requisitos funcionales se centran en diferentes aspectos de la funcionalidad de un sistema:
   a) Estructura y datos
   b) Función y flujo
   c) Estado y comportamiento
3. Por último, los requisitos sólo pueden entenderse en su contexto (Principio 4 en 2):
   a) *Contexto del sistema*, incluidos los actores externos
   b) Frontera del sistema e *interfaces externas*

Hay muchas interrelaciones y dependencias entre los aspectos mencionados. Por ejemplo, una solicitud emitida por un usuario (contexto) puede desencadenar una transición de estado (estado y comportamiento) que inicie una acción seguida de otra acción (función y flujo) que requiera datos (estructura y datos) para proporcionar un resultado al usuario (contexto) dentro de un intervalo de tiempo determinado (calidad).

Algunos productos del trabajo se centran en un aspecto específico y se abstraen de los demás aspectos. Esto es particularmente cierto en el caso de los modelos de requisitos (3.4). Otros productos del trabajo, como la especificación de los requisitos del sistema, cubren todos estos aspectos. Cuando se documentan diferentes aspectos en productos del trabajo separados o en capítulos separados del mismo producto del trabajo, estos productos del trabajo o capítulos deben mantenerse consistentes entre sí.

3.1.5 Directrices Generales de Documentación (N1)

Independientemente de las técnicas utilizadas, las guías siguientes son de aplicación durante la creación de productos del trabajo:

- Seleccione un tipo de producto del trabajo que se ajuste al *objetivo previsto*.
- *Evite la redundancia* haciendo referencia al contenido en lugar de repetir el mismo contenido de nuevo.
- *Asegúrese de que no haya inconsistencias* entre los productos del trabajo, particularmente cuando cubren diferentes aspectos.

- *Utilice los términos de manera consistente*, conforme están definidos en el glosario.
- *Estructure* adecuadamente los productos del trabajo.

3.1.6 Planificar los Productos del Trabajo a Utilizar (N1)

Cada entorno de proyecto y cada dominio es diferente, por lo que el conjunto de productos del trabajo resultantes debe ser definido para cada caso. Por lo tanto, se debe llegar a acuerdos acerca de las cuestiones siguientes:

- ¿En qué productos del trabajo se documentarán los requisitos y con qué propósito?
- ¿Qué niveles de abstracción es necesario considerar?
- ¿Hasta qué nivel de detalle deben documentarse los requisitos en cada nivel de abstracción?
- ¿Cómo se representarán los requisitos en estos productos del trabajo?

Los productos del trabajo que se utilizarán deben definirse una etapa temprana de un proyecto. Esto tiene varias ventajas:

- Ayuda a planificar los esfuerzos y los recursos.
- Asegura que se utilicen las notaciones apropiadas.
- Asegura que todos los resultados se documenten en los productos del trabajo correctos.
- Asegura que no se necesite luego un reajuste importante de la información y una "edición final".
- Ayuda a evitar la redundancia, lo que resulta en menos trabajo y mantenibilidad más sencilla.

3.2 Productos del Trabajo Basados en Lenguaje Natural (N2)

Desde el inicio de la IR sistemática, el lenguaje natural ha sido un medio fundamental para especificar los requisitos en la práctica.

Los productos del trabajo basados en el lenguaje natural tienen grandes ventajas:

- El lenguaje natural sin restricciones es extremadamente expresivo y flexible.
- Casi cualquier requisito que se conciba, relacionado con cualquier aspecto, puede expresarse en lenguaje natural.
- El lenguaje natural se utiliza en la vida cotidiana y se enseña en la escuela, por lo que no se requiere una formación específica para leer y comprender textos en lenguaje natural.

Estas ventajas se dan a expensas de que los textos escritos en lenguaje natural frecuentemente son susceptibles de ser interpretados de diferentes maneras, lo que constituye un problema a la hora de especificar los requisitos. Además, la detección de ambigüedades, omisiones e inconsistencias en tales textos es difícil y costosa.

Las recomendaciones siguientes facilitan la redacción de buenos requisitos basados en el lenguaje natural:

- Escribir frases cortas y bien estructuradas.
- Definir y utilizar de forma consistente una terminología uniforme (3.5).
- Evitar términos y frases vagos o ambiguos.
- Conocer los escollos de la escritura técnica que se enumeran a continuación.

Cuando se redactan documentos técnicos en lenguaje natural, hay algunos escollos bien conocidos que deben evitarse o utilizarse con cuidado [GoRu2003].

Cosas a evitar:

- Descripciones incompletas
- Nombres no específicos
- Condiciones incompletas
- Comparaciones incompletas

Cosas que deben usarse con cuidado:

- Voz pasiva
- Cuantificadores universales (como "todo" o "nunca")
- Nominalizaciones (es decir, sustantivos derivados de un verbo, por ejemplo, "autenticación")

3.3 Productos del Trabajo Basados en Plantillas (N3)

Los productos del trabajo basados en plantillas se utilizan para superar algunas de las deficiencias de los productos del trabajo basados en lenguaje natural, proporcionando estructuras predefinidas para los requisitos.

- Las *plantillas de frases* proporcionan una estructura sintáctica predefinida para una frase que expresa un requisito, en particular un requisito individual o una historia de usuario.
- Las *plantillas de formulario* proporcionan un conjunto de campos predefinidos en un formulario que debe rellenarse, por ejemplo, para redactar un caso de uso o un requisito de calidad que se pueda medir.
- *Las plantillas de documento* proporcionan una estructura predefinida para un documento de requisitos.

Se han descrito varias plantillas en la literatura. [ISO29148], [MWHN2009], y [Rupp2014] proporcionan plantillas de frases para los requisitos individuales. [Cohn2004] define una plantilla de frases ampliamente utilizada para las historias de usuario y [Cock2001] describe plantillas de formulario para los casos de uso. [Laue2002] ha propuesto una plantilla para la descripción de las tareas. [ISO29148] y [RoRo2012] proporcionan plantillas de documento para especificaciones completas. Además, un cliente podría prescribir el uso de plantillas específicas para sus proyectos.

Ventajas de los productos del trabajo basados en plantillas:

- Proporcionar una estructura clara y reutilizable
- Ayudar en la captura de la información más relevante
- Hacer que los requisitos y las especificaciones de los requisitos se vean uniformes
- Mejorar la calidad general de los requisitos y de las especificaciones de requisitos

Desventajas y escollos de los productos del trabajo basados en plantillas:

- La gente a menudo se centra en cumplimentar formalmente la plantilla más que en el contenido.
- Aumenta la probabilidad de que se omitan los aspectos que no se incluyen en la plantilla.

3.4 Productos del Trabajo Basados en Modelos (N3)

Los requisitos representados en lenguaje natural tienen limitaciones [Davi1993], en particular con respecto a proporcionar una visión general de un conjunto de requisitos y comprender las relaciones entre los mismos. Modelar los requisitos aborda estas limitaciones.

3.4.1 El Papel de los Modelos en la Ingeniería de Requisitos (N2)

Un *modelo* es una representación abstracta de una parte de la realidad existente o de una parte de la realidad que se desea crear. La noción de realidad incluye cualquier conjunto concebible de elementos, fenómenos, o conceptos, incluidos otros modelos. Con respecto a un modelo, la parte modelada de la realidad se llama el *original*. Ejemplos de modelos fuera del ámbito de la ingeniería de software son los modelos de información de construcción (BIM) [ISO19650], que modelan los elementos necesarios para planificar, construir y gestionar edificios y otros elementos de construcción.

En la IR, los modelos ayudan a comprender las relaciones e interconexiones entre los requisitos y proporcionan una visión general de un conjunto de requisitos. Esto se logra principalmente centrándose en algunos aspectos — por ejemplo, el comportamiento — mientras se abstrae de todos los demás aspectos. El uso de una notación gráfica para un modelo facilita la obtención de una visión general. Sin embargo, los modelos también pueden representarse de manera no gráfica, por ejemplo, con tablas.

Los modelos de requisitos tienen ventajas en comparación con los requisitos representados en lenguaje natural:

- Las relaciones e interconexiones entre los requisitos son más fáciles de entender con modelos gráficos que cuando se especifican en lenguaje natural.
- Centrarse en un solo aspecto reduce la carga cognitiva para comprender los requisitos modelados.
- Los lenguajes de modelado de requisitos tienen una sintaxis restringida que reduce la posibilidad de ambigüedades y omisiones.

Los modelos también tienen limitaciones:

- Mantener modelos que se centran en diferentes aspectos coherentes entre sí es un desafío.
- Es necesario integrar la información de los diferentes modelos para un entendimiento causal.
- Los modelos se centran principalmente en los requisitos funcionales; la mayoría de los requisitos de calidad y de las restricciones no pueden expresarse en los modelos con un esfuerzo razonable.
- La sintaxis restringida de un lenguaje de modelado gráfico implica que no todos los elementos de información relevantes pueden ser expresados en un modelo.

Por lo tanto, es frecuente combinar modelos de requisitos y requisitos en lenguaje natural.

En IR, los modelos pueden ser usados para:

- *Especificar* los requisitos (principalmente funcionales) parcialmente o incluso de forma completa, como medio para sustituir los requisitos representados textualmente.
- *Descomponer* una realidad compleja en aspectos bien definidos y complementarios; representando cada aspecto mediante un modelo específico.
- *Parafrasear* los requisitos representados textualmente a fin de mejorar su comprensibilidad, en particular con respecto a las relaciones entre ellos.
- *Validar* los requisitos representados textualmente con el objetivo de descubrir omisiones, ambigüedades e inconsistencias.

Los *lenguajes de modelado* se utilizan para expresar modelos. Varios lenguajes de modelado, por ejemplo, UML [OMG2017] o BPMN [OMG2013], se han estandarizado. Cuando los requisitos se especifican en un lenguaje de modelado no estándar, se requiere una leyenda que explique la sintaxis y la semántica del lenguaje de modelado utilizado.

Hay muchos tipos de modelos que se pueden usar en la IR. Un Ingeniero de Requisitos debe comprender qué tipo de modelo es el más adecuado para especificar los requisitos en una situación determinada.

En una fase temprana, el ingeniero de requisitos suele empezar por modelar el contexto (3.4.2) o los objetivos del sistema previsto.

3.4.2 Modelando el Contexto (N2)

Los modelos que se centran en el aspecto del contexto especifican la integración estructural de un sistema en su entorno, así como la interacción entre un sistema y los actores en el contexto del sistema.

Los *modelos de contexto* especifican un sistema y los actores en el contexto del sistema que interactúan con el sistema. Un modelo de contexto también esboza las interfaces entre un sistema y su contexto (por ejemplo, en términos de qué información se intercambia).

Los *diagramas de contexto* se utilizan como lenguaje de modelado gráfico para expresar modelos de contexto. No existe una notación estandarizada para los diagramas de contexto.

Los modelos de contexto pueden expresarse mediante los diagramas de contexto del análisis estructurado [DeMa1978] o diagramas de cajas y líneas adaptados [Glin2019].

En el lenguaje de modelado UML [OMG2017], los *diagramas de casos de uso* proporcionan un medio para modelar un sistema y su contexto, en términos de los casos de uso del sistema y de los actores en el contexto del sistema que interactúan con el sistema mediante estos casos de uso.

Los *casos de uso* modelan la interacción dinámica entre un actor en el contexto del sistema y un sistema, desde la perspectiva del actor. Los casos de uso se escriben en su mayoría utilizando plantillas de formulario en lenguaje natural (3.3) o utilizando diagramas de actividades UML (3.4.4).

3.4.3 Modelando la Estructura y los Datos (N3)

Los modelos que se centran en aspectos de estructura y de datos especifican los requisitos relativos a las propiedades estructurales estáticas de un sistema o un dominio.

Los modelos de dominio estáticos especifican los objetos (de negocio) y sus relaciones en un dominio de interés. Pueden expresarse con diagramas de clases UML [OMG2017].

Los *modelos de clases* especifican principalmente las clases de un sistema y sus atributos y relaciones. Las clases representan entidades tangibles e intangibles en el mundo real, que el sistema debe conocer para llevar a cabo sus tareas. Los diagramas de clases UML se utilizan típicamente como lenguaje de modelado para los modelos de clases.

3.4.4 Modelando Función y Flujo (N3)

Los modelos que se centran en aspectos de función y flujo especifican los requisitos relativos a la secuencia de acciones que son necesarias para producir los resultados requeridos a partir de determinados insumos, o las acciones necesarias para ejecutar un proceso (de negocio), incluido el flujo de control y de datos entre las acciones y quién es responsable de cada acción.

Los *modelos de actividades* se utilizan para especificar las funciones del sistema. En el lenguaje de modelado UML [OMG2017], se utilizan *diagramas de actividades* para expresar los modelos de actividad. Proporcionan elementos para modelar las acciones y el flujo de control entre las acciones. Los diagramas de actividades también pueden expresar quién es responsable de cada acción. Los elementos de modelado avanzados (no cubiertos por el CPRE Nivel Básico) proporcionan los medios para modelar el flujo de datos.

Los modelos de procesos se utilizan para describir los procesos de negocio o los procesos técnicos. Pueden expresarse con diagramas de actividades UML o con lenguajes específicos de modelado de procesos como BPMN [OMG2013]. En CPRE Nivel Básico, sólo usamos diagramas de actividades UML para el modelado de procesos.

3.4.5 Modelando Estado y Comportamiento (N2)

Los modelos que se centran en el estado y el comportamiento especifican los requisitos para el comportamiento de un sistema o de un componente del dominio; lo hacen en términos de las reacciones del sistema a eventos, siendo estas reacciones dependientes del estado del sistema, o en términos de la dinámica de la interacción entre los componentes del sistema.

Las *máquinas de estados* modelan los eventos que desencadenan la transición desde un estado a otro, y las acciones que deben realizarse cuando se produce una transición de estado. Los *gráficos de estados* [Hare1988] son máquinas de estados con estados que se descomponen jerárquicamente y/o ortogonalmente. Las máquinas de estados, incluidos los gráficos de estados, pueden expresarse en el lenguaje de modelado UML [OMG2017] mediante *diagramas de máquinas de estados* (también llamados *diagramas de estados*).

3.4.6 Otros tipos de modelos en la ingeniería de requisitos (L1)

En CPRE nivel básico, la comprensión y la aplicación de los modelos se limitan a algunos tipos de modelos importantes. Existen otros tipos de modelos que se utilizan en la Ingeniería de Requisitos. En CPRE nivel básico, basta con conocerlas y saber para qué se usan.

Los modelos de objetivos representan un conjunto de objetivos, subobjetivos, y las relaciones entre ellos. Los modelos de objetivos también pueden incluir las tareas y los recursos necesarios para alcanzar un objetivo, los actores que desean alcanzarlo y los obstáculos que impiden su consecución.

En SysML [OMG2019], los diagramas de definición de bloques pueden adaptarse para expresar diagramas de contexto utilizando bloques estereotipados para el sistema y para los actores. Los diagramas de definición de bloques también pueden utilizarse para modelar la estructura de un sistema en términos de las entidades conceptuales del sistema y las relaciones entre ellas.

Los modelos de historias de dominio pueden utilizarse para modelar la función y el flujo, especificando historias visuales sobre cómo los actores interactúan con los dispositivos, artefactos y otros elementos de un dominio, normalmente utilizando símbolos específicos del dominio [HoSch2020]. Son un medio para comprender el dominio de aplicación en el que operará un sistema.

Los diagramas de secuencia UML se usan con frecuencia para especificar la interacción entre objetos.

3.5 Glosarios (N2)

En cada esfuerzo de IR que involucra a más de una persona, existe el riesgo de una falta de entendimiento compartido de la terminología — es decir, que algunas personas interpreten los mismos términos de diferentes maneras. Para mitigar este problema, el entendimiento compartido de los términos relevantes se registra en un glosario.

Un *glosario* es una colección centralizada de definiciones de: términos específicos del contexto, términos cotidianos con un significado especial en el contexto dado, abreviaturas, y acrónimos. Los sinónimos (diferentes términos que denotan la misma cosa) deben marcarse como tales. Los homónimos (usos del mismo término para cosas diferentes) deben evitarse o marcarse como tales.

Las reglas siguientes se aplican a los glosarios:

- Gestione el glosario de forma centralizada.
- Mantenga el glosario durante todo el proyecto de desarrollo del sistema.
- Otorgue la responsabilidad de definir y mantener el glosario a una persona o a un grupo pequeño.
- Use un estilo y una estructura uniforme para el glosario.
- Involucre a las partes interesadas y busque el acuerdo sobre la terminología.
- Haga el glosario accesible para todas las personas involucradas.
- Prescriba el uso obligatorio del glosario.
- Revise que los productos del trabajo hacen un uso adecuado del glosario.

3.6 Documentos de Requisitos y Estructuras de Documentación (N2)

Los documentos de especificación de requisitos (3.1.1) comprenden varios productos del trabajo de la IR. Por consiguiente, es importante organizar esos documentos con una estructura bien definida, a fin de crear un conjunto coherente y sostenible de requisitos. Además de los requisitos, un documento de requisitos puede contener también información y explicaciones adicionales — por ejemplo, un glosario, condiciones de aceptación, información sobre el proyecto, o características de la implementación técnica.

Los requisitos también pueden organizarse en estructuras de documentación distintas a los documentos clásicos.

Los documentos utilizados con frecuencia son los siguientes:

- Especificación de Requisitos de Partes Interesadas
- Especificación de Requisitos de Usuarios (un subconjunto de una Especificación de Requisitos de Partes Interesadas, que abarca sólo los requisitos de los usuarios)
- Especificación de Requisitos del Sistema
- Especificación de Requisitos de Negocio
- Documento de Visión

Las estructuras de documentación alternativas que se utilizan con frecuencia son las siguientes:

- Product backlog
- Sprint backlog
- Mapa de historias [N. del T.: el término inglés story map también se usa con frecuencia]

Tanto la selección de una estructura de documentación como la organización interna de la estructura elegida dependen de lo siguiente:

- El proceso de desarrollo elegido (5)
- El tipo de desarrollo y el dominio
- El contrato (un cliente puede prescribir el uso de una estructura de documentación determinada)
- El tamaño del documento

Las plantillas de documento pueden ayudar a estructurar una especificación de requisitos. Existen varias plantillas disponibles en la literatura [Vole2020], [RoRo2012] y en estándares [ISO29148]. También es posible reutilizar plantillas de proyectos anteriores similares al actual, o los clientes pueden imponer el uso de una plantilla determinada. Una organización también puede decidir crear una plantilla como un estándar interno.

3.7 Prototipos en Ingeniería de Requisitos (N1)

En la IR, los *prototipos* son un medio para especificar los requisitos mediante el ejemplo, y para validar los requisitos. En particular, se pueden utilizar prototipos si las partes interesadas no desean redactar y revisar productos del trabajo en lenguaje natural, basados en plantillas, o basados en modelos.

Los prototipos *exploratorios* [LiSZ1994] se utilizan para crear un entendimiento compartido, aclarar requisitos, o validar requisitos a diferentes niveles de fidelidad. Se desechan después de su uso.

- Los *wireframes* son prototipos de baja fidelidad construidos con materiales sencillos o con herramientas de dibujo, que sirven principalmente para discutir y validar ideas de diseño y bosquejos de interfaz de usuario.
- Las *maquetas* son prototipos de mediana fidelidad [N. del T.: el término inglés *mock-up* también se usa con frecuencia]. Al especificar sistemas digitales, las maquetas utilizan pantallas reales y soportan navegación mediante clics, pero sin una funcionalidad real. Sirven principalmente para especificar y validar las interfaces de usuario.
- Los *prototipos nativos* son prototipos de alta fidelidad que implementan partes críticas de un sistema de forma que los interesados pueden utilizar el prototipo para comprobar si la parte prototipada del sistema funcionará y se comportará como se espera.

Dependiendo del grado de fidelidad, los prototipos de exploración pueden ser un producto del trabajo costoso, por lo que siempre existe una compensación entre el coste y el valor obtenido.

Los *prototipos evolutivos* [LiSZ1994] son sistemas piloto que forman el núcleo de un sistema a desarrollar. El sistema final evoluciona ampliando y mejorando gradualmente el sistema piloto en varias iteraciones.

3.8 Criterios de Calidad para los Productos del Trabajo y los Requisitos (N1)

Un requisito debe cumplir criterios de calidad específicos para ser considerado un buen requisito. En la IR moderna, con enfoques orientados al valor (Principio 1 en 2), el grado de cumplimiento de un criterio de calidad se corresponderá con valor creado por este requisito. Esto significa que los requisitos no tienen que satisfacer completamente todos los criterios de calidad — pero, cuanto más alto sea el valor de un requisito, más relevantes son sus criterios de calidad asociados, con el fin de reducir el riesgo de fracaso.

La adecuación y la comprensibilidad son los criterios de calidad más importantes para los requisitos individuales. Sin ellos, un requisito es inútil o incluso perjudicial, independientemente del cumplimiento de todos los demás criterios.

Criterios de calidad recomendados para *requisitos individuales*:

- Adecuado (describe necesidades reales y acordadas con las partes interesadas)
- Necesario
- No ambiguo
- Completo (auto-contenido)
- Comprensible
- Verificable

Como se describe en 3.1.1, es usual registrar los requisitos en diversos productos del trabajo que contienen un requisito individual o múltiples requisitos. Los criterios de calidad anteriores se utilizarán para crear buenos requisitos individuales en un producto del trabajo. En el caso de los productos del trabajo que contengan más de un requisito, se considerarán además los criterios de calidad siguientes.

Criterios de calidad recomendados para productos de trabajo que cubren *múltiples requisitos*:

- Consistente
- No redundante
- Completo (no faltan requisitos conocidos y relevantes)
- Modificable
- Trazable
- Conforme (a normas y estándares)

4 Prácticas para la Elaboración de Requisitos (N3)

Objetivo: Comprender el uso de prácticas para identificar fuentes de requisitos, educir requisitos, identificar y resolver conflictos, y validar los requisitos

Duración: 4 horas y 30 minutos

Términos: Fuente de requisitos, frontera del sistema, contexto del sistema, educción de requisitos, negociación de requisitos, validación de requisitos, parte implicada, modelo de Kano, conflicto, resolución de conflictos

Objetivos educativos

OE 4.1.1 Determinar las fronteras del sistema para centrarse en los requisitos relevantes (N3)
OE 4.1.2 Recordar las fuentes relevantes para el sistema a crear (N1)
OE 4.1.3 Identificar a las partes implicadas y redactar una lista de partes implicadas (N3)
OE 4.1.4 Comprender los beneficios de la gestión de las partes implicadas (N2)
OE 4.2.1 Comprender cómo el modelo de Kano puede ayudar a educir los requisitos adecuados (N2)
OE 4.2.2 Comprender la diferencia entre las técnicas de recolección y las técnicas de diseño y generación de ideas (N2)
OE 4.2.3 Comprender cómo elegir una técnica de educción adecuada para una situación dada (N2)
OE 4.3.1 Recordar los diferentes tipos de conflicto (N1)
OE 4.3.2 Comprender qué actividades son necesarias para resolver conflictos (N2)
OE 4.3.3 Comprender cómo aplicar las técnicas de resolución de conflictos apropiadas (N2)
OE 4.4.1 Comprender por qué es necesario validar los documentos de requisitos (N2)
OE 4.4.2 Recordar los cuatro aspectos importantes para la validación de requisitos (N1)
OE 4.4.3 Comprender cómo aplicar técnicas apropiadas para la validación de requisitos (N2)

4.1 Fuentes para Requisitos (N3)

La calidad y la completitud de los requisitos dependen en gran medida de las fuentes de requisitos involucradas. La falta de una fuente relevante conducirá a una comprensión incompleta de los requisitos o a requisitos incompletos. La identificación de las fuentes de requisitos es un proceso iterativo y recursivo que requiere una reconsideración constante.

El Capítulo 2, Principio 3, hace hincapié en la necesidad de un entendimiento compartido (explícito e implícito) entre todas las partes implicadas. Comprender el contexto del sistema que se va a desarrollar en un determinado dominio de aplicación es un requisito previo para poder identificar las fuentes de requisitos pertinentes. La zona comprendida entre la frontera del sistema y la frontera del contexto se denomina el contexto (del sistema) (Principio 4 en 2). El contexto (del sistema) es necesario para comprender la naturaleza de los requisitos que se han de elaborar y, por consiguiente, para identificar las fuentes originales para requisitos.

Las fuentes de requisitos se clasifican en tres tipos:

- Partes implicadas
- Documentos
- Sistemas

Las partes implicadas de un sistema (véase [Glin2025] para una definición; véase también Principio 2 en 2) son la principal fuente de requisitos. Los roles típicos de las partes interesadas son [BiSp2003]:

- Usuarios (también llamados usuarios finales)
- Patrocinadores
- Gestores
- Desarrolladores
- Autoridades
- Clientes

Además, las personas u organizaciones que se *ven afectadas* por un sistema deben considerarse partes interesadas (indirectas).

La identificación sistemática de las partes implicadas debería tener lugar al comienzo de un proyecto de desarrollo y los resultados deberían gestionarse durante todo el desarrollo como una actividad continua. Esto incluye la identificación tanto de los roles de las partes interesadas como de las personas que los desempeñan.

En todos los sistemas con interfaz de usuario, los *usuarios finales* del sistema constituyen un grupo de partes implicadas que reviste especial interés para el ingeniero de requisitos. Los usuarios finales deben agruparse (por ejemplo, por roles, tareas o responsabilidades similares).

Cuando los usuarios finales puedan ser identificados individualmente, deberán elegirse representantes de cada grupo. De lo contrario, se pueden definir personas para representar a los grupos de usuarios finales pertinentes [Coop2004].

Las fuentes potenciales para identificar a las partes implicadas relevantes son:

- Listas de control de grupos y roles típicos de las partes implicadas
- Estructuras organizativas
- Documentación de procesos de negocio
- Informes de mercado
- Partes implicadas iniciales para identificar partes implicadas *adicionales*

Las partes implicadas deberían documentarse en una lista actualizada de partes implicadas con (al menos) la información siguiente:

- Nombre
- Función (rol)
- Información personal y de contacto adicional
- Disponibilidad durante el progreso del proyecto (cuándo y dónde)
- Relevancia
- Áreas y nivel de conocimiento

- Objetivos e intereses en cuanto al proyecto

Pueden surgir problemas con las partes implicadas si los derechos y obligaciones de una parte implicada no están claros o si no se atienden suficientemente las necesidades de la parte implicada. La gestión de las relaciones de las partes implicadas [Bour2009] es una forma eficaz de contrarrestar los problemas con las partes implicadas.

En la mayoría de contextos de sistema, hay más fuentes disponibles. También deben considerarse para que un sistema nuevo tenga éxito, ya que la mayoría de las partes implicadas no hablan de lo obvio: sus requisitos "subconscientes" (4.2).

Las fuentes de requisitos adicionales incluyen:

- Sistemas existentes y legados
- Documentos de procesos
- Documentos legales o regulatorios
- Regulaciones específicas de una compañía
- Información (de marketing) sobre posibles usuarios futuros

Se puede encontrar otra fuente para requisitos examinando situaciones similares en dominios completamente diferentes.

4.2 Educción de Requisitos (N2)

Dentro de la educción, la tarea del Ingeniero de Requisitos es comprender los deseos y necesidades de las partes implicadas y, al mismo tiempo, asegurarse de que se han recogido los requisitos de todas las fuentes relevantes de requisitos, aplicando técnicas apropiadas para educirlos. Un punto importante en la educción es convertir las demandas, deseos, y expectativas implícitas en requisitos explícitos.

Para educir los requisitos, es crucial conocer la naturaleza y la importancia de cada requisito. Éstas pueden cambiar de un proyecto a otro y también con el tiempo. El modelo de Kano [KaeA1984] clasifica los requisitos en tres categorías relevantes:

- Encantadores (sinónimos: factores de entusiasmo, requisitos inconscientes)
- Satisfactores (sinónimos: factores de desempeño, requisitos conscientes)
- Insatisfactores (sinónimos: factores básicos, requisitos subconscientes)

Hay muchas técnicas diferentes para educir estas categorías de requisitos. Nosotros diferenciamos entre:

- Técnicas de recolección
- Técnicas de diseño y de generación de ideas

Las *técnicas de recolección* son técnicas establecidas para la educción de requisitos [BaCC2015] que ayudan a educir satisfactores e insatisfactores mediante la investigación de diferentes fuentes.

Se pueden distinguir cuatro categorías principales:

- Técnicas de interrogatorio
- Técnicas de colaboración

- Técnicas de observación
- Técnicas basadas en artefactos

Las *técnicas de diseño y generación de* ideas tienen como objetivo estimular la creatividad durante la educción de requisitos. Su objetivo es crear ideas para resolver un problema y explorar ideas de diseño [Kuma2013]. Esto puede dar lugar a requisitos nuevos o innovadores que a menudo son encantadores. (N. del T.: en la acepción del modelo de Kano.) Ejemplos populares de estas técnicas son la lluvia de ideas [Osbo1979], las analogías, la creación de prototipos (por ejemplo, maquetas), los escenarios y los storyboards.

Un concepto más amplio relacionado con el diseño y la generación de ideas es el *design thinking*. Existen diferentes aproximaciones, como *d.school* [Sdsc2012] y *Designing for Growth* [LiOg2011], que ofrecen un amplio conjunto de técnicas que pueden utilizarse para la educción de requisitos.

Las técnicas de educción deben ser capaces de detectar todo tipo de requisitos — requisitos funcionales y de calidad, y restricciones, del mismo modo. En la práctica, los requisitos de calidad y las restricciones suelen recibir menos atención.

Para educir *requisitos de calidad*, debería utilizarse un modelo de calidad como el estándar ISO 25010 [ISO25010] como lista de control. Este modelo también puede ser de ayuda para cuantificar requisitos.

Las *restricciones* pueden encontrarse considerando posibles limitaciones del espacio de solución potencial — por ejemplo, cuestiones técnicas, legales, organizativas, culturales, o ambientales.

Elegir las técnicas de educción adecuadas es una competencia clave crítica que depende de muchos factores diferentes, como:

- Tipo de sistema
- Modelo de ciclo de vida de desarrollo de software
- Personas involucradas
- Configuración de la organización

Los mejores resultados se logran con la combinación de diferentes técnicas de educción. [CaDJ2014] representan un enfoque sistemático para seleccionar las técnicas.

4.3 Resolución de conflictos relativos a los requisitos (N2)

Las técnicas de educción por sí mismas no aseguran que el conjunto de requisitos resultantes como un todo sea consistente, completo, conforme, etc. (3.8). Para el conjunto final, sin embargo, todas las partes implicadas deben comprender y acordar todos los requisitos que son relevantes para ellas. Si algunas partes implicadas no están de acuerdo, esta situación debe reconocerse como un conflicto que debe resolverse en consecuencia. Se deberían seleccionar técnicas de resolución de conflictos adecuadas en función del tipo de conflicto e información contextual. Esto requiere comprender a fondo la naturaleza del conflicto de requisitos y las actitudes de las partes implicadas.

Las tareas para identificar y resolver conflictos son:

- Identificación de conflictos
- Análisis de conflictos
- Resolución de conflictos
- Documentación de la resolución de conflictos (decisiones tomadas)

Es útil distinguir entre diferentes tipos de conflicto [Moor2014]. Los tipos de conflicto siguientes suelen requerir la atención del Ingeniero de Requisitos:

- Conflicto del tema en cuestión
- Conflicto de datos
- Conflicto de intereses
- Conflicto de valor
- Conflicto de relaciones
- Conflicto estructural

Para resolver los conflictos con éxito, se pueden aplicar técnicas comunes:

- Acuerdo
- Compromiso
- Votación
- Criterio de autoridad
- Definición de variantes

Además, existen varias técnicas auxiliares, por ejemplo:

- Tener en cuenta todos los hechos (circunstancias)
- "Más-menos interesante"
- Matriz de decisión

4.4 Validación de Requisitos (N2)

Validar requisitos es un paso importante para el éxito de un sistema (Principio 6 en 2). Asegurar la calidad de los requisitos desde el principio reducirá el esfuerzo desperdiciado más adelante. Validar los requisitos significa comprobar la calidad de los productos del trabajo en cuestión así como de los requisitos individuales que contiene (véase 3.8 para más detalles).

Los aspectos importantes que deben considerarse en la validación de requisitos incluyen:

- Intervención de las partes implicadas adecuadas
- Separar la identificación y la corrección de los defectos
- Validación desde distintos puntos de vista
- Validación repetida

Existen varias técnicas de validación (por ejemplo, [GiGr1993], [OleA2018]). Estas técnicas de validación suelen clasificarse en:

- *Técnicas de revisión*, incluyendo:
  - Revisiones guiadas
  - Inspecciones
- *Técnicas exploratorias*, por ejemplo:
  - Prototipado
  - Pruebas alfa y beta
  - Pruebas A/B [KoTh2017]
  - Construcción de un producto mínimo viable (MVP)
- *Desarrollo de una muestra*

Estas técnicas difieren en formalidad y esfuerzo. La técnica a seleccionar depende de factores como el modelo de ciclo de vida de desarrollo de software, la madurez del proceso de desarrollo, la complejidad y el nivel de riesgo del sistema, cualquier requisito legal o regulatorio, y/o la necesidad de informes de auditoría.

5 Estructura de Proceso y Trabajo (N3)

Objetivo: Explicar los conceptos de un proceso de IR y aplicar configuraciones de proceso apropiadas

Duración: 1 hora y 15 minutos

Términos: Proceso, proceso de IR

Objetivos educativos

OE 5.1.1 Conocer los factores importantes que influyen en un proceso de IR (N1)
OE 5.1.2 Comprender cómo y por qué estos factores influyen (N2)
OE 5.2.1 Comprender las facetas a considerar para la configuración de un proceso de IR (N2)
OE 5.3.1 Conocer configuraciones típicas del proceso de IR (N1)
OE 5.3.2 Comprender los pasos para configurar un proceso de IR (N2)
OE 5.3.3 Para situaciones con sistemas y desarrollos sencillos, seleccionar y aplicar una configuración del proceso de IR apropiada (N3)

Se requiere un proceso para dar forma y estructurar el trabajo de IR a realizar en un contexto dado. Como no hay un proceso de IR que sea adecuado para cualquier situación (1.4), se debe configurar un proceso de IR a medida que se ajuste al contexto de desarrollo y de sistema dado.

El proceso de IR da forma al flujo de información y al modelo de comunicación entre los diversos participantes (por ejemplo, clientes, usuarios, Ingenieros de Requisitos, desarrolladores, y probadores) y también define los productos del trabajo que se utilizarán o producirán. Así, pues, el proceso de IR proporciona el marco para educir, documentar, validar y gestionar requisitos.

5.1 Factores de Influencia (N2)

Muchos factores influyen en la configuración de un proceso de IR. Los principales factores son:

- Encaje global del proceso: el proceso de IR debe encajar en el proceso global de desarrollo del sistema.
- Contexto de desarrollo
- Capacidad y disponibilidad de las partes implicadas
- Entendimiento compartido
- Complejidad y criticidad del sistema a desarrollar
- Restricciones
- Tiempo y presupuesto disponibles
- Volatilidad de los requisitos
- Experiencia de los Ingenieros de Requisitos

Un análisis de los factores de influencia proporciona información sobre cómo configurar el proceso de IR. Los factores de influencia también restringen el espacio de las posibles configuraciones de proceso. Por ejemplo, cuando las partes implicadas sólo están

disponibles al principio del proyecto, no se puede elegir un proceso que se base en la retroalimentación continua de las partes implicadas.

5.2 Facetas del Proceso de Ingeniería de Requisitos (N2)

Hay tres facetas decisivas que se deben considerar cuando se configura un proceso de IR[Glin2019].

Faceta de Tiempo: Lineal frente a Iterativo

En un proceso lineal, los requisitos se especifican por adelantado en una sola fase del proceso. En un proceso iterativo, los requisitos se especifican de manera incremental, comenzando por las metas generales y algunos requisitos iniciales, y luego añadiendo o modificando requisitos en cada iteración.

Criterios para elegir un proceso *lineal* de IR:

- El proceso de desarrollo para el sistema sigue un plan y es principalmente lineal.
- Las partes implicadas conocen sus requisitos y pueden especificarlos por adelantado.
- Se requiere una especificación de requisitos exhaustiva como base contractual para la externalización del diseño y la implementación del sistema.
- Las autoridades regulatorias exigen una especificación de requisitos exhaustiva y entregada oficialmente en una etapa temprana del desarrollo.

Criterios para elegir un proceso *iterativo* de IR:

- El proceso de desarrollo para el sistema es iterativo y ágil.
- Muchos de los requisitos no se conocen de antemano, pero surgirán y evolucionarán durante el desarrollo del sistema.
- Las partes implicadas están disponibles de tal manera que se pueden establecer bucles de retroalimentación breves como un medio para mitigar el riesgo de desarrollar el sistema equivocado.
- La duración del desarrollo permite más de una o dos iteraciones.
- La capacidad de cambiar los requisitos fácilmente es importante.

Faceta de Propósito: Preceptivo frente a Exploratorio

En un proceso de IR preceptivo, la especificación de requisitos constituye un contrato: todos los requisitos son vinculantes y deben implementarse. En un proceso de IR exploratorio, sólo se conocen a priori las metas, mientras que los requisitos concretos tienen que explorarse.

Criterios para elegir un proceso de IR *preceptivo*:

- El cliente requiere un contrato fijo para el desarrollo del sistema.
- La funcionalidad y el alcance tienen prioridad sobre el coste y los plazos.
- El desarrollo del sistema especificado puede ser licitado o externalizado.

Criterios para elegir un proceso de IR *exploratorio*:

- Inicialmente, las partes implicadas sólo tienen una vaga idea de sus requisitos.
- Las partes implicadas participan activamente y proporcionan retroalimentación continua.
- Los plazos y el coste tienen prioridad sobre la funcionalidad y el alcance.
- No está claro a priori qué requisitos se implementarán realmente y en qué orden se hará.

Faceta de Objetivo: Específico para un Cliente frente a Orientado al Mercado

En un proceso de IR específico para un cliente, el sistema es encargado por un cliente y desarrollado por un proveedor. En un proceso de IR orientado al mercado, el sistema se desarrolla como un producto o servicio para un mercado, dirigido a segmentos de usuarios específicos.

Criterios para elegir un proceso de IR *específico para un cliente*:

- El sistema será utilizado principalmente por la organización que ha encargado el sistema y que paga por su desarrollo.
- Las partes implicadas importantes están principalmente asociadas a la organización del cliente.
- Se pueden identificar personas concretas para los roles de los partes implicadas.
- El cliente quiere una especificación de requisitos que pueda servir como contrato.

Criterios para elegir un proceso de IR *orientado al mercado*:

- La organización que realiza el desarrollo pretende vender el sistema como producto o servicio en algún segmento de mercado .
- Los posibles usuarios no son identificables individualmente.
- Los Ingenieros de Requisitos tienen que diseñar los requisitos de manera que se ajusten a las necesidades previstas de los usuarios a los que se dirigen.
- Dueños del producto, personal de marketing, diseñadores digitales, y arquitectos de sistema son las partes implicadas principales.

Pistas y Advertencias

- Los criterios presentados anteriormente son *heurísticas* más que reglas fijas. Por ejemplo, la externalización del desarrollo del sistema se hace preferiblemente con un proceso de IR preceptivo en lugar de uno exploratorio porque el contrato entre el cliente y el proveedor se basa normalmente en una especificación de requisitos exhaustiva. Sin embargo, también es posible negociar un contrato de externalización basado en un proceso de IR exploratorio.
- Los procesos de IR lineales funcionan sólo si hay implantado un proceso sofisticado para cambiar los requisitos.
- Los procesos de IR lineales implican bucles de retroalimentación largos: Para mitigar el riesgo de desarrollar el sistema equivocado, los requisitos se deben validar intensamente.

- Cuando se define un proceso de IR, a menudo se elige que sea *lineal* y *preceptivo* a la vez.
- Los procesos de IR exploratorios también son típicamente procesos iterativos (y viceversa).
- En un proceso orientado al mercado, la retroalimentación de los usuarios es el único medio para validar si el producto satisfará realmente las necesidades del segmento de usuarios al que va dirigido.
- La faceta orientada al mercado no combina bien con las facetas lineal y preceptiva.

5.3 Configuración de un Proceso de Ingeniería de Requisitos (N3)

En un contexto concreto de desarrollo de sistema, los responsables de la IR tienen que configurar el proceso de IR a aplicar. Basado en un análisis de los factores de influencia en 5.1, se puede utilizar una combinación adecuada de las facetas de proceso descritas en 5.2 [Glin2019]. A continuación se describen tres combinaciones típicas.

Proceso de IR Participativo: Iterativo y Exploratorio y Específico para un Cliente

Caso de aplicación principal:	Proveedor y cliente colaboran estrechamente; las partes implicadas participan activamente tanto en el proceso de IR como en el de desarrollo
Productos del trabajo típicos:	Product backlog con historias de usuarios y/o descripciones de tareas, prototipos
Flujo de información típico:	Interacción continua entre partes implicadas, dueños del producto, Ingenieros de Requisitos, y desarrolladores; puede incluir retroalimentación de usuarios

Proceso de IR Contractual: Típicamente Lineal (a veces Iterativo) y Preceptivo y Específico para un Cliente

Caso de aplicación principal:	La especificación de requisitos constituye la base contractual para el desarrollo de un sistema por personas no involucradas en la especificación y con poca interacción de las partes implicadas después de la fase de requisitos.
Productos del trabajo típicos:	Especificación de requisitos de sistema clásica, con requisitos basados en lenguaje natural y modelos
Flujo de información típico:	Principalmente de las partes implicadas a los Ingenieros de Requisitos

Proceso de IR orientado al Producto: Iterativo y Exploratorio y Orientado al Mercado

Caso de aplicación principal:	Una organización especifica y desarrolla software para venderlo o distribuirlo como un producto o servicio
Productos del trabajo típicos:	Product backlog, prototipos
Flujo de información típico:	Interacción entre dueño de producto, marketing, Ingenieros de Requisitos, diseñadores digitales, y desarrolladores; y (tal vez) retroalimentación rápida de clientes/usuarios

Obsérvese que puede haber contextos de sistema y de desarrollo en los que ninguna de las configuraciones mencionadas encaje. Por ejemplo, restricciones regulatorias pueden imponer el uso de un proceso que se ajuste a determinados estándares como el ISO/IEC/IEEE 29148 [ISO29148].

Cuando se configura un proceso de IR, recomendamos usar un procedimiento de cinco pasos:

1. Analizar los factores de influencia (5.1)
2. Evaluar los criterios de faceta (5.2)
3. Configurar el proceso (5.3)
4. Determinar los productos del trabajo (3)
5. Seleccionar las prácticas apropiadas

6 Prácticas de Gestión de Requisitos (N2)

Objetivo: Comprender la necesidad y el beneficio de la gestión de requisitos

Duración: 2 horas

Términos: Gestión de requisitos, gestión del cambio, trazabilidad, atributos de los requisitos, ciclo de vida de los requisitos, priorización

Objetivos educativos

OE 6.1.1 Conocer en qué consiste la gestión de requisitos y por qué es necesaria (N1)
OE 6.2.1 Explicar porqué los productos del trabajo de los requisitos necesitan un modelo de estado/ciclo de vida (N2)
OE 6.3.1 Explicar cómo es el concepto de versionado de requisitos en una situación determinada del proyecto (N2)
OE 6.4.1 Conocer el uso de configuraciones y líneas base de requisitos (N1)
OE 6.5.1 Conocer el propósito de los atributos de requisitos (N1)
OE 6.5.2 Explicar cómo es un conjunto adecuado de atributos para los requisitos en una situación de proyecto determinada (N2)
OE 6.5.3 Explicar el propósito de las vistas y mencionar las diferentes vistas de requisitos (N2)
OE 6.6.1 Mencionar las razones para la trazabilidad de los requisitos (N1)
OE 6.6.2 Describir de forma resumida las diferencias entre la trazabilidad implícita y explícita (N1)
OE 6.6.3 Conocer cómo se puede documentar la trazabilidad explícita (N1)
OE 6.7.1 Conocer cómo tratar los cambios en los enfoques lineales (basados en un plan) y ágiles (N1)
OE 6.8.1 Conocer el motivo del establecimiento de prioridades y conocer los criterios de evaluación significativos (N1)
OE 6.8.2 Nombrar los pasos de la priorización de los requisitos (N1)
OE 6.8.3 Nombrar las diferentes categorías de técnicas de priorización (N1)

6.1 ¿Qué es la Gestión de Requisitos? (N1)

La gestión de requisitos es el proceso de gestionar los requisitos existentes registrados en varios productos del trabajo. En particular, esto incluye el almacenamiento, el cambio y el trazado de requisitos [Glin2025]. La gestión de requisitos puede llevarse a cabo de diferentes maneras y a diferentes niveles, dependiendo del proceso de desarrollo y el contexto elegidos — véase, por ejemplo,[Leff2011], [Rupp2014], [WiBe2013]. Independientemente de las circunstancias, la tarea de la gestión de requisitos es mantener los requisitos de tal manera que todos los roles de un proyecto puedan trabajar de manera efectiva y eficiente.

6.2 Gestión del Ciclo de Vida (N2)

La gestión del ciclo de vida se refiere al proceso de seguimiento de todos los productos del trabajo con respecto a la situación en su ciclo de vida. Cada requisito documentado y cada producto del trabajo tiene su propio ciclo de vida: se crea, se evalúa y se refina antes de ser revisado, reelaborado, consolidado, acordado, y así en lo sucesivo. Para permitir la identificación de qué producto del trabajo está en qué estado, se requiere un modelo de

ciclo de vida que defina cada estado de ciclo de vida y transición de estado permitidos. El estado actual de un producto del trabajo siempre debe estar claro, incluyendo (normalmente) la historia de sus transiciones.

6.3 Control de Versiones (N2)

El control de versiones de los requisitos se refiere al proceso de seguimiento de todos los productos del trabajo durante su evolución. Cualquier cambio en un producto del trabajo debería verse reflejado en una nueva versión. El versionado permite trazar la historia de un producto del trabajo hasta su origen y restaurar un producto de trabajo a cualquier versión anterior. Para ello, el control de versiones requiere la adopción de tres medidas:

- Un número de versión para identificar de forma única la versión de un producto del trabajo.
- Una historia de lo que se cambió.
- Un concepto para el almacenamiento de productos del trabajo.

Se debe considerar el versionado para todos los productos del trabajo [WiBe2013]. Generalmente, un número de versión consta de al menos dos partes: la versión y el incremento.

6.4 Configuraciones y Líneas Base (N1)

Una *configuración de requisitos* es un conjunto consistente de productos del trabajo que contienen requisitos. Cada configuración se define para un propósito específico y contiene como máximo una versión de cada producto del trabajo [Glin2025]. La finalidad de las configuraciones es, por ejemplo, revisar un conjunto de productos del trabajo o facilitar una estimación del esfuerzo de desarrollo.

Una *línea base* es una *configuración* estable y controlada en cuanto a cambios de los productos del trabajo, que se utiliza para la planificación de la entrega u otros hitos de entrega en un proyecto [Glin2025].

Las configuraciones tienen las propiedades siguientes:

1. Conexión lógica
2. Consistencia
3. Unicidad
4. Inalterabilidad
5. Base para el restablecimiento

6.5 Atributos y Vistas (N2)

Los *atributos* son necesarios para documentar los metadatos importantes de un producto del trabajo y suelen utilizarse para responder a un número de preguntas importantes durante el ciclo de vida del proyecto o del producto.

El objetivo de usar atributos para caracterizar los requisitos es permitir que los miembros del equipo y otras partes implicadas obtengan la información que necesitan sobre los requisitos en cualquier momento del proyecto.

La definición del conjunto de atributos relevantes depende de las necesidades de información de las diferentes partes implicadas en el proyecto. Los estándares existentes, por ejemplo [ISO29148], proporcionan una visión general de los atributos más relevantes.

Las *vistas* son un extracto del conjunto total de requisitos que sólo incluyen el contenido de interés en un momento dado. Desde una perspectiva técnica, una vista es una combinación de filtros y configuraciones de ordenación que pueden ponerse a disposición o reutilizarse para otros participantes almacenando la combinación seleccionada.

Distinguimos tres tipos de vistas:

- *Vistas selectivas*
- *Vistas proyectivas*
- *Vistas agregadas*

En la mayoría de los casos, las vistas de requisitos son combinaciones de vistas selectivas, proyectivas y agregadas para elaborar los informes.

6.6 Trazabilidad (N1)

La trazabilidad [GoFi1994] es la capacidad de trazar un requisito *hasta su origen* (es decir, las partes implicadas, los documentos, las justificaciones, etc.) y *hacia adelante hasta los productos del trabajo subsiguientes (por ejemplo, los casos de prueba)*, así como hasta *otros requisitos* de los que depende este requisito.

La trazabilidad es un pre-requisito para la gestión de requisitos y, a menudo, se exige de forma explícita en estándares, leyes y reglamentos. Implementar la trazabilidad significa básicamente mantener las dependencias entre los diferentes productos del trabajo (3.1) en diferentes niveles de abstracción (3.1.2), niveles de detalle (3.1.3) y hacia todos los predecesores y sucesores relevantes por razones de análisis, conformidad e información.

La trazabilidad puede documentarse de forma *implícita*, estructurando y normalizando los productos del trabajo, o de forma *explícita*, relacionando los productos del trabajo entre sí sobre la base de sus identificadores únicos en diversas formas [HuJD2011]. Las formas de representación más comunes son los hipervínculos, las referencias, las matrices, las tablas o los grafos.

6.7 Tratamiento del Cambio (N1)

Los requisitos no son estáticos. Los cambios en los requisitos se producen por muy diversas razones y deben ser tratados de forma adecuada (Principio 7 en 2), por ejemplo, mediante la creación de una *solicitud* formal *de cambio*, o añadiendo un nuevo elemento al *product backlog*.

La toma de decisiones, la planificación y el control de la implementación de un cambio depende del enfoque de desarrollo y del momento en que se produce el cambio.

En un enfoque *lineal*, la decisión sobre un cambio suele ser adoptada por un Comité de Control del Cambio (en los proyectos) o un Comité Asesor del Cambio (en operaciones). En un enfoque más *iterativo*, el dueño del producto incluye el cambio en el product backlog y prioriza el nuevo elemento en consecuencia.

6.8 Priorización (N1)

No todos los requisitos son igualmente importantes [Davi2005]. La evaluación y la priorización se utilizan para determinar los requisitos más relevantes para una próxima entrega o incremento del producto.

La *evaluación* de los requisitos es la base para su priorización, a menudo determinada mediante el uso de múltiples criterios de evaluación como el valor de negocio, la urgencia, el esfuerzo, las dependencias y otros.

La *prioridad* de un requisito describe la importancia de un requisito individual en comparación con otros requisitos de acuerdo con determinados criterios [Glin2025]. La *priorización* propiamente dicha se realiza en base a un solo criterio o a múltiples criterios; esto depende principalmente de la técnica de priorización elegida.

Pasos para la priorización:

- Definir los objetivos y restricciones más importantes para la priorización
- Definir los criterios de evaluación deseados
- Definir las partes implicadas que deben participar
- Definir los requisitos que deben ser priorizados
- Seleccionar la técnica de priorización
- Llevar a cabo la priorización

Las *técnicas* de priorización se pueden clasificar en:

- Técnicas de priorización *ad-hoc*
- Técnicas de priorización *analíticas*

7 Apoyo de Herramientas (N2)

Objetivo: Ofrecer una visión general de la función de las herramientas de IR y los aspectos para la implementación

Duración: 30 minutos

Términos: Herramienta, herramienta de IR

Objetivos educativos

OE 7.1.1 Conocer los diferentes tipos de herramientas de IR (N1)
OE 7.2.1 Explicar qué se debe considerar cuando se introducen las herramientas de IR (N2)

7.1 Herramientas en la Ingeniería de Requisitos (N1)

El proceso de la IR puede ser apoyado por herramientas que respalden tareas y actividades dedicadas. Dado que los procesos de la IR son bastante individuales (5), las herramientas de IR existentes se centran a menudo sólo en ciertos aspectos de la IR y rara vez apoyan todas las actividades. Antes de seleccionar una herramienta, los Ingenieros de Requisitos deben decidir qué tareas y actividades deben ser apoyadas durante el proceso de IR, y cómo. Diferenciamos entre las herramientas que soportan:

- Gestión de requisitos:
  - Definición y almacenamiento de los atributos de los requisitos
  - Priorización de Requisitos
  - Gestión de versiones y configuraciones
  - Trazado y seguimiento de requisitos
  - Gestión de los cambios en los requisitos
- Gestión del proceso de IR:
  - Medición y presentación de informes sobre el proceso de IR
  - Medición y presentación de informes sobre la calidad del producto
  - Gestionar el flujo de trabajo de la IR
- Documentación deñ conocimiento sobre los requisitos:
  - Compartir requisitos
  - Creación un entendimiento compartido de los requisitos
- Modelización de requisitos
- Colaboración en la IR
- Prueba/simulación de los requisitos

Las herramientas suelen tener una mezcla de las características mencionadas. Para asegurar que todas las tareas de la IR se cubren adecuadamente, se podrían combinar diferentes herramientas.

A veces se utilizan otros tipos de herramientas, por ejemplo, herramientas ofimáticas o de seguimiento de incidencias, para documentar o gestionar los requisitos. Estas herramientas

tienen limitaciones y deben utilizarse sólo cuando el proceso de IR está bajo control y los requisitos están alineados y son suficientemente estables.

7.2 Introducción de Herramientas (N2)

Seleccionar una herramienta de IR no es diferente de seleccionar una herramienta con cualquier otro propósito. El objetivo, el contexto y los requisitos deben describirse antes de la selección para que pueda tener éxito [Fugg1993].

Sólo se puede buscar una herramienta apropiada una vez que se hayan introducido los procedimientos y técnicas de IR adecuados. La introducción de una herramienta requiere responsabilidades y procedimientos en la IR claramente establecidos. En el proceso de introducción de una herramienta de IR, los aspectos siguientes son relevantes:

- Tener en cuenta los costes de todo el ciclo de vida más allá de los costes de licencia
- Considerar los recursos necesarios
- Utilizar proyectos piloto para evitar riesgos
- Evaluar la herramienta según criterios definidos
- Instruir a los empleados sobre cómo usar la herramienta

Referencias

[AnPC1994] Annie I. Antón, W. Michael McCracken, Colin Potts: Goal Decomposition and Scenario Analysis in Business Process Reengineering. CAiSE (Conference on Advanced Information Systems Engineering), 1994, 94–104.

[BaCC2015] K. Baxter, C. Courage, K. Caine: Understanding Your Users: A Practical Guide to User Research Methods, 2nd edition. Morgan Kaufmann, Burlington, 2015.

[BiSp2003] K. Bittner, I. Spence: Use Case Modelling. Pearson Education, Boston, 2003.

[Bour2009] L. Bourne: Stakeholder Relationship Management: A Maturity Model for Organisational Implementation. Gower Publishing Ltd, Birlington, 2009.

[CaDJ2014] D. Carrizo, O. Dieste, N. Juristo: Systematizing requirements elicitation technique selection. Information and Software Technology 2014, 56(6): 644–669.

[Cock2001] A. Cockburn: Writing Effective Use Cases. Addison-Wesley, Boston 2001.

[Cohn2004] M. Cohn: User Stories Applied – For Agile Software Development. Addison-Wesley, Boston, 2004.

[Coop2004] A. Cooper: The Inmates Are Running the Asylum: Why High-Tech Products Drive Us Crazy and How to Restore the Sanity. Que, Indianapolis, 2004.

[Davi2005] A. M. Davis: Just Enough Requirements Management – Where Software Development Meets Marketing. Dorset House Publishing, New York, 2005.

[Davi1993] A. M. Davis: Software Requirements – Objects, Functions, & States, 2nd edition, Prentice Hall, New Jersey, 1993.

[DeMa1978] T. DeMarco: Structured Analysis and System Specification. Yourdon Press, New York, 1978.

[Fugg1993] A. Fuggetta: A classification of CASE technology. IEEE Computer 1993, 26 (12): 25–38.

[GiGr1993] T. Gilb, D. Graham: Software Inspection. Addison Wesley, Boston, 1993.

[Glin2019] M. Glinz: Requirements Engineering I. Course Notes, University of Zurich, 2019. https://www.ifi.uzh.ch/en/rerg/courses/archives/hs19/re-i.html#resources. Visitada por última vez en enero de 2026.

[Glin2025] M. Glinz: A Glossary of Requirements Engineering Terminology. Versión 2.2. https://cpre.ireb.org/downloads-and-resources/downloads#cpre-glossary. Visitada por última vez en enero de 2026.

[GoFi1994] O. Gotel, A. Finkelstein: An Analysis of the Requirements Traceability Problem. 1st International Conference on Requirements Engineering, Colorado Springs, 1994. 94–101.

[GoRu2003] R. Goetz, C. Rupp: Psychotherapy for System Requirements. 2nd IEEE International Conference on Cognitive Informatics (ICCI'03), London, 2003. 75–80.

[GRL2020] Goal oriented Requirement Language. University of Toronto, Canada https://www.cs.toronto.edu/km/GRL. Visitada por última vez en enero de 2026.

[Hare1988] D. Harel: On Visual Formalisms. Communications of the ACM 1988, 31 (5): 514–530.

[HoSch2020] S. Hofer, H. Schwentner: Domain Storytelling — A Collaborative Modeling Method. Available from Leanpub, http://leanpub.com/domainstorytelling. Visitada por última vez en enero de 2026.

[HuJD2011] E. Hull, K.Hull, K. Jackson, and J. Dick: Requirements Engineering. Springer, 3rd Ed, 2011.

[ISO29148] ISO/IEC/IEEE 29148: Systems and Software Engineering – Life Cycle Processes – Requirements Engineering, International Organization for Standardization, 2018.

[ISO19650] ISO 19650: Organization and Digitization of Information about Buildings and Civil Engineering Works, including Building Information Modelling (BIM)– Information Management Using Building Information Modelling – Part 1 and 2, International Organization for Standardization, 2018.

[ISO25010] ISO/IEC/IEEE25010:2011: Systems and software Quality Requirements and Evaluation (SQuaRE) –– System and software quality models. International Organization for Standardization, Geneva, 2011.

[Jack1995] M. A.A. Jackson: Software Requirements and Specifications: A Lexicon of Practice, Principles and Prejudices. Addison-Wesley, New York, 1995.

[Jack1995b] M. Jackson: The World and the Machine. 17th International Conference on Software Engineering 1995 (ICSE 1995). 287–292.

[KaeA1984] N. Kano et al.: Attractive quality and must-be quality. Journal of the Japanese Society for Quality Control 1984, 14(2): 39–48. (en japonés)

[KoTh2017] R. Kohavi, S. Thomke: The Surprising Power of Online Experiments – Getting the most out of A/B and other controlled tests. Harvard Business Review, Sept–Oct 2017: 74–82.

[Kuma2013] V. Kumar: 101 Design Methods – A Structured Approach for Driving Innovation in Your Organization. John Wiley & Sons, Hoboken, 2013.

[Laue2002] S. Lauesen: Software Requirements. Styles and Techniques. Addison-Wesley, Harlow, 2002.

[Leff2011] D. Leffingwell: Agile Software Requirements, Lean Requirements Practices for Teams, Programs, and the Enterprise. Addison-Wesley, Boston, 2011.

[LiOg2011] J. Liedtka, T. Ogilvie: Designing for Growth: A Design Thinking Tool Kit For Managers. Columbia University Press, 2011.

[LiSZ1994] H. Lichter, M. Schneider-Hufschmidt, H. Zullighoven: Prototyping in Industrial Software Projects – Bridging the Gap Between Theory and Practice. IEEE Transactions on Software Engineering 1994, 20 (11): 825–832.

[MFeA2019] D. Méndez Fernández, X. Franch, N. Seyff, M. Felderer, M. Glinz, M. Kalinowski, A. Volgelsang, S. Wagner, S. Bühne, K. Lauenroth: Do We Preach What We Practice? Investigating the Practical Relevance of Requirements Engineering Syllabi – The IREB Case. CIbSE 2019: 476–487.

[Moor2014] C. W. Moore: The Mediation Process – Practical Strategies for Resolving Conflicts, 4th edition. John Wiley & Sons, Hoboken, 2014.

[MWHN2009] A. Mavin, P. Wilkinson, A. Harwood, and M. Novak: Easy Approach to Requirements Syntax (EARS). 17th IEEE International Requirements Engineering Conference (RE'09), Atlanta, Georgia, 2009. 317–322.

[OleA2018] K. Olsen et al.: Certified Tester, Foundation Level Syllabus – Version 2018. International Software Testing Qualifications Board, 2018.

[OMG2013] Object Management Group: Business Process Model and Notation (BPMN), version 2.0.2. OMG document formal/2013-12-09 http://www.omg.org/spec/BPMN. Visitada por última vez en enero de 2026.

[OMG2017] Object Management Group: OMG Unified Modeling Language (OMG UML), version 2.5.1. OMG document formal/2017-12-05. https://www.omg.org/spec/UML/About-UML/. Visitada por última vez en enero de 2026.

[OMG2019] Object Management Group: OMG Systems Modeling Language (OMG SysML™), Version 1.6. OMG Document formal/19-11-01. https://www.omg.org/spec/SysML/. Visitada por última vez en enero de 2026.

[Osbo1979] A. F. Osborn: Applied Imagination, 3rd revised edition. Charles Scribner´s Sons, New York, 1979.

[RoRo2012] S. Robertson and J. Robertson: Mastering the Requirements Process, 3rd edition. Addison-Wesley, Boston, 2012.

[Rupp2014] C. Rupp: Requirements-Engineering und Management, 6. Auflage. Hanser, München, 2014. (en alemán).

[Sdsc2012] Stanford d.school: An Introduction to Design Thinking. Hasso Plattner Institute of Design, Stanford, 2012. https://web.stanford.edu/~mshanks/MichaelShanks/files/509554.pdf . Visitada por última vez en enero de 2026.

[vLam2009] Axel van Lamsweerde: Requirements Engineering: From System Goals to UML Models to Software Specifications. Chichester: John Wiley & Sons, 2009.

[Vole2020] Volere: Requirements Resources. https://www.volere.org. Visitada por última vez en enero de 2026.

[WiBe2013] K. Wiegers and J. Beatty: Software Requirements, 3rd edition. Microsoft Press, Redmond, 2013.