

REQUISITOS

INDICE

1	INTRODUCCIÓN	2
2	DEFINICIÓN DE REQUISITO	3
3	REQUISITOS FUNCIONALES.....	4
4	REQUISITOS NO FUNCIONALES	4
5	REQUISITOS [F]URPS+	7
6	ATRIBUTOS DE CALIDAD	8
7	REQUISITOS FUNCIONALES VS NO FUNCIONALES	9
8	REGLAS DEL DOMINIO O DEL NEGOCIO	9
9	REQUISITOS DE INFORMACIÓN.....	10
10	DESCRIPCIÓN DE REQUISITOS	12
11	EJEMPLOS DEL ENUNCIADO DE REQUISITOS.....	15
12	CONSEJOS PARA LA REDACCIÓN DE REQUISITOS	16
13	REQUISITOS VS METAS	17
14	REQUISITOS DE USUARIO Y REQUISITOS DE SISTEMA.....	17
15	PROCESOS DE LA INGENIERÍA DE REQUISITOS	18
15.1	ELICITACIÓN DE REQUISITOS.....	19
15.2	VALIDACIÓN DE REQUISITOS	21
15.3	GESTIÓN DE REQUISITOS.....	22
16	CARACTERÍSTICAS DE LOS REQUISITOS	23
17	DIFICULTADES Y BENEFICIOS DE LA DEFINICIÓN DE REQUISITOS	26
18	EL DOCUMENTO DE ESPECIFICACIÓN DE REQUISITOS DEL SOFTWARE ...	27

1 INTRODUCCIÓN

En este documento se aborda lo relativo al trabajo con los Requisitos del sistema software. En concreto, y aunque no exclusivamente, se aborda lo relativo a su definición, que

será una de las primeras tareas¹ a realizar en relación al sistema software sobre el que se vaya a trabajar.

La información recopilada podrá expresarse, bien de manera explícita en un Catálogo de Requisitos o, de manera más formal, en un Documento de Especificación de Requisitos, bien de manera implícita en la descripción textual de los Casos de Uso² para el caso de los Requisitos Funcionales.³

2 DEFINICIÓN DE REQUISITO

La elicitación, captura, especificación o determinación de requisitos es el proceso del estudio de las necesidades de los usuarios para llegar a una definición de los requisitos del sistema.

Según el estándar IEEE⁴ Std 24765-2017 IEEE Standard Glossary of Software Engineering Terminology⁵, un requisito se define como “A condition or capability needed by a user to solve a problem or achieve an objective”, es decir, como una condición o capacidad que necesita el usuario para resolver un problema o conseguir un objetivo determinado.

También se aplica a las condiciones que debe cumplir o poseer un sistema o uno de sus componentes para satisfacer un contrato, una norma o una especificación. Por tanto, un requisito puede verse como una declaración abstracta de alto nivel de un servicio que el sistema debe proporcionar o como una definición detallada y formal de una función del sistema. Los requisitos cumplen una doble función ya que inicialmente constituyen una oferta de contrato y están abiertos a la interpretación, y posteriormente se convierten en el contrato en sí mismo tras su definición detallada.

¹ Típicamente se realizará durante la Fase de Arranque del Proceso Unificado de Desarrollo de Software como parte de la Actividad de Requisitos.

² Los Casos de Uso son una técnica del Lenguaje de Modelado Unificado que permite capturar información acerca de cómo trabaja un sistema actualmente y de cómo se desea que trabaje. Además se trata de una técnica que pone énfasis en la relación del sistema con el exterior. Permite la captura de requisitos (funcionales) y sirve igual para un enfoque procedimental que para una filosofía orientada a objetos. Un Caso de Uso puede definirse como una interacción típica entre el usuario y el sistema que captura una función visible al usuario y logra un objetivo del usuario que puede ser grande o pequeño. Los Casos de Uso se pueden representar visualmente a través de los Diagramas de Casos de Uso, en los que además de los Casos de Uso como tal, representados por un óvalo con su nombre, aparecen los Actores de dichos Casos de Uso representados por un monigote con su nombre. A través del Diagrama de Casos de Uso del sistema se puede representar quiénes usan el sistema: roles de personas, máquinas u otros sistemas software, y saber qué quieren del sistema, además de determinar qué está dentro del sistema y qué queda fuera. Véase más información en el documento dedicado a este tema de forma específica.

³ Un Requisito Funcional es una característica requerida del sistema que expresa una capacidad de acción del mismo, es decir, una funcionalidad, generalmente expresada en una declaración en forma verbal. Véase más información sobre el concepto de Requisito Funcional en la sección del presente documento dedicada a este tema de forma específica.

⁴ Las siglas IEEE corresponden a Institute of Electrical and Electronics Engineers.

⁵ Véase <https://standards.ieee.org/findstds/standard/24765-2017.html> para más información.

3 REQUISITOS FUNCIONALES

Cuando se habla de una característica requerida de la cual se sabe que puede ser satisfecha por medio de la adición de un subsistema o bloque de código en el software, entonces se dice que se está ante un **Requisito Funcional** (RF)⁶, por cuanto es un requisito que denota una funcionalidad del sistema. Por tanto, este tipo de requisito podría definirse de la siguiente manera:

Requisito Funcional: Característica requerida del sistema que expresa una capacidad de acción del mismo, es decir, una funcionalidad, generalmente enunciada en una declaración en forma verbal.

Los requisitos funcionales constituyen una definición de los servicios que el sistema debe proporcionar, cómo debe reaccionar a una entrada particular y cómo se debe comportar ante situaciones particulares. Además, los requisitos funcionales pueden referirse a diferentes aspectos, por ejemplo:

- **Requisitos sobre la estructura** de la información del sistema.⁷
- **Requisitos sobre la actualización de la información del sistema**, es decir, características de las funciones que realizan cambios en la información del sistema para introducir nuevos datos o modificar o eliminar los ya existentes.
- **Requisitos sobre las consultas** que se podrán realizar sobre la información del sistema y sobre los informes que el sistema generará con dicha información.
- **Requisitos sobre la interacción con otros sistemas**.
- [...]

Según el tipo de sistema que se esté desarrollando, y su complejidad y tamaño, los requisitos funcionales se representarán de forma más adecuada en un documento específico de forma explícita o mediante Casos de Uso.

4 REQUISITOS NO FUNCIONALES

Por otra parte, debe tenerse en cuenta que no todo lo que los clientes vayan a solicitar es funcionalidad pura, por el contrario, en ocasiones, los **clientes solicitarán otras características, que si bien en principio no serán objeto de codificación, sí es cierto que pueden llegar a afectar a ésta**. En este caso, se hablará de **Requisitos no Funcionales** (RNF)⁸, y **serán todas las restricciones o exigencias de cualidades que se imponen al proyecto, como por ejemplo en**

⁶ En los textos ingleses los Requisitos Funcionales se denominan *Functional Requisites* o *Requirements* (FRQ).

⁷ Aunque hay expertos que consideran este tipo de requisitos como una forma especial de requisitos denominados Requisitos de Información. Véase más información sobre este tipo de requisitos en la sección del presente documento dedicada a ese tipo de Requisito de forma específica.

⁸ En los textos ingleses los Requisitos no Funcionales se denominan *Non Functional Requisites* o *Requirements* (NFR).

términos de rendimiento o de seguridad. Un requisito no funcional es una característica ya sea del sistema, del proyecto o del servicio prestado, que es requerida por el cliente junto con la especificación del propio sistema pero que, en principio, no se satisface añadiendo código, sino cumpliendo con ésta como si de una restricción se tratara. Por tanto, este tipo de requisito podría definirse de la siguiente manera:

Requisito no Funcional: Característica requerida del sistema, del proceso de desarrollo o del servicio prestado, que señala una restricción del mismo.

Los requisitos no funcionales suponen restricciones que afectan a los servicios o funciones del sistema, tales como restricciones de estándares, internacionalización, accesibilidad, etc. Los requisitos no funcionales definen propiedades emergentes del sistema⁹, tales como el tiempo de respuesta, las necesidades de almacenamiento, la fiabilidad, etc. También pueden especificar, por ejemplo, que el sistema software debe utilizar un determinado Sistema Gestor de Bases de Datos (DBMS, DataBase Management System) como Oracle, o bien, que el sistema software debe funcionar bajo un determinado Sistema Operativo como Linux, o incluso la utilización de una herramienta de Ingeniería de Software Asistida por Ordenador (CASE, Computer Aided Software Engineering) en particular, de un modelo de proceso de desarrollo específico o de un determinado lenguaje de programación.

Ejemplos típicos de aspectos que habitualmente se consideran mediante los requisitos no funcionales son los siguientes:

- Accesibilidad.
- Aspectos legales (por ejemplo, protección de datos, etc.).
- Aspectos éticos y morales.
- Aspectos del entorno físico (calor, vibración, etc.).
- Aspectos de internacionalización (unidades de medida, unidades monetarias, idiomas, etc.).
- Calidad.
- Capacidad (actual y previsiones).
- Certificación.
- Compatibilidad.
- Copias de seguridad.
- Estabilidad.
- Estandarización (estándares técnicos, seguridad, calidad).
- Extensibilidad.
- Fiabilidad (por ejemplo, el Tiempo Medio entre Fallos (MTBF, Mean Time Between Failures)).
- Gestión de errores.
- Licencias de uso.
- Mantenimiento.
- Plazos de entrega.
- Portabilidad.
- Restricciones de software y hardware (por ejemplo, en

⁹ Un sistema complejo está compuesto por varias partes interconectadas cuyos vínculos crean información adicional no visible anteriormente para el observador. Como resultado de las interacciones entre elementos, surgen propiedades nuevas que no pueden explicarse a partir de las propiedades de los elementos aislados. Dichas propiedades se denominan propiedades emergentes.

- Coste. relación al Sistema Operativo).
- Disponibilidad.
- Documentación (manuales de usuario, instalación, administración, etc.) y ayuda.
- Eficiencia.
- Escalabilidad.
- Restricciones de desarrollo (herramientas a utilizar, etc.).
- Rendimiento.
- Seguridad.
- Soporte.
- Velocidad de ejecución.
- [...]

Debe tenerse en cuenta que los requisitos no funcionales pueden ser más críticos que los requisitos funcionales ya que si un requisito funcional no se cumple, el sistema se degrada, mientras que si un requisito no funcional no se cumple, el sistema puede inutilizarse.

Tal y como ya se ha comentado, los requisitos no funcionales serán restricciones que impondrán limitaciones. Además, es importante señalar que, más que las restricciones relativas al diseño, implementación, etc., en la captura de requisitos, es importante considerar las restricciones que se imponen por causas inevitables, como por ejemplo podrían ser las restricciones legales por licencias de software o la estructura de una interfaz ya definida de un sistema externo existente al que se debe invocar.

También debe tenerse en cuenta que los requisitos no funcionales pueden presentar conflictos entre ellos, y así, algunas veces cumplir un requisito no funcional supone un impacto sobre otro de los requisitos no funcionales definidos debido a la existencia de interdependencias inevitables o creadas entre ellos, por ejemplo, Portabilidad vs Rendimiento o Coste vs Plazos de entrega.

Finalmente, señalar que, en algunos textos, los requisitos no funcionales se clasifican de la siguiente forma:

- **De producto:** Especifican que el producto o sistema software debe comportarse de una forma en particular, por ejemplo, en cuanto a velocidad de ejecución, memoria requerida o escalabilidad.
- **Organizacionales:** Son consecuencia de políticas y procedimientos de la organización, por ejemplo, coste, plazos de entrega, procesos estándar utilizados o documentación específica requerida.
- **Externos:** Emergen de factores que son externos a los sistemas y a sus procesos de desarrollo, por ejemplo, requisitos legales o requisitos éticos y morales.

5 REQUISITOS [F]URPS+

En ciertos textos se habla de los requisitos no funcionales como requisitos [F]URPS+¹⁰, del acrónimo en inglés de las siguientes categorías [ver Figura 1] :

- Usability – Usabilidad o Facilidad de uso –: Incluye todos aquellos atributos que facilitan la interacción de un usuario con el sistema.
- Reliability – Fiabilidad –: Agrupa los requisitos que tienen que ver con la solidez y robustez de un sistema durante su ejecución.
- Performance – Rendimiento –: Se refiere a la velocidad del sistema y a su eficiencia en lo relativo a la utilización de recursos.
- Soportability – Soporte –: Incluyen requisitos de instalación y configuración, así como facilidades para mantener y administrar la operación del sistema.
- El signo + se refiere a requisitos adicionales que, como ya es sabido, habitualmente serán restricciones:
 - Restricciones de diseño: Limitan las posibilidades en lo relativo al diseño del sistema.
 - Restricciones de implementación: Se refieren a reglas relativas a la codificación, como la utilización de un lenguaje de programación específico o el cumplimiento de ciertos estándares.
 - Restricciones físicas: Se refieren a indicaciones relativas al hardware.
 - [...]

A continuación se presentan ejemplos de especificación de requisitos URPS:

- Facilidad de uso:
 - El sistema deberá mostrar el texto de forma que se pueda leer fácilmente a una distancia de 1 metro. El cajero estará mirando a menudo al cliente o a los artículos, no a la pantalla del terminal, motivo por el que se deben comunicar las señales y avisos con sonidos, en lugar de solo mediante gráficos.
- Fiabilidad:
 - Capacidad de Recuperación: El sistema deberá solucionar localmente el hecho de que se produzca algún fallo al usar un servicio externo como una autorización de pago.
- Rendimiento:
 - El sistema deberá conseguir la autorización de pago en menos de 1 minuto al menos el 90% de las veces.

¹⁰ En realidad, FURPS+ incluiría también *Functionality*, es decir, lo que un usuario debe ser capaz de hacer a través del sistema, por tanto, el acrónimo FURPS+ se referiría tanto a requisitos funcionales como a requisitos no funcionales.

- Soporte:
 - El sistema debe ser fácilmente instalable por los propios usuarios.¹¹

Funcionality	Requisitos funcionales
U sability	Human factors aesthetics, consistency, documentation
R eliability (Fiabilidad)	Frequency/severity of failure, recoverability, predictability, accuracy, MTBF
P erformance (Rendimiento)	Speed efficiency, resource usage, throughput, response time
S upportability (Soporte)	<div>Testability</div> <div>Adaptability</div> <div>Compatibility</div> <div>Serviceability</div> <div>Localizability</div> <div>Extensibility</div> <div>Maintainability</div> <div>Configurability</div> <div>Installability</div> <div>Robustness</div>

Figura 1: Requisitos URPS (Usability, Reliability, Performance y Supportability).

6 ATRIBUTOS DE CALIDAD

En algunos textos, ciertos requisitos, típicamente no funcionales, se denominan **Atributos de Calidad**¹² del sistema. Estos incluyen **facilidad de uso, fiabilidad, rendimiento**, etc., e influyen en la arquitectura del sistema, así por ejemplo un requisito de alto rendimiento y alta fiabilidad influirá en la elección de componentes de software y hardware y en sus configuraciones. Estos atributos se refieren a las cualidades del sistema, no necesariamente implicando que dichas cualidades sean de alta calidad, así, por ejemplo, la cualidad de soporte podría elegirse de baja calidad, si el sistema a desarrollar se fuera a utilizar durante un plazo relativamente breve. Estos atributos de calidad en ocasiones se clasifican en:

- **Observables en la ejecución** (facilidad de uso, fiabilidad, rendimiento, etc.).
- **No observables en la ejecución** (soporte, documentación, etc.).

¹¹ La redacción de este Requisito no funcional resulta poco específica en comparación con, por ejemplo, el ejemplo de Requisito no funcional de Rendimiento, motivo por el cual no sería fácil evaluar el grado de cumplimiento de dicho requisito. Así, sería deseable concretar qué porcentaje de usuarios deberían ser capaces de instalar el sistema software de forma autónoma. No obstante, para una primera versión del requisito, la redacción sería válida.

¹² En los textos ingleses los Atributos de Calidad se denominan *Quality Attributes*.

7 REQUISITOS FUNCIONALES VS NO FUNCIONALES

Es importante tener en cuenta que el objetivo de la Ingeniería de Software es el desarrollo de sistemas apegados a las necesidades del cliente, pero también ajustados a otros criterios como los recursos disponibles o los plazos de entrega. En base a esto es importante que la aplicación de técnicas de Ingeniería de Software sirva, no solo para implementar la funcionalidad demandada, es decir para escribir código ajustado a los requisitos funcionales, sino también para cumplir con las cualidades suplementarias, es decir, con los requisitos no funcionales, o de lo contrario no cumplirá con su misión que es desarrollar el software que se necesita para el momento y condiciones de contorno existentes.

Comparando los requisitos funcionales frente a los requisitos no funcionales podría decirse que la lista detallada de requisitos debe incluir la siguiente información:

- Qué debe hacer el sistema o cómo funcionará el sistema: Requisitos funcionales.
- Con qué restricciones sobre su operación e implementación: Requisitos no funcionales, es decir, capacidad, coste, disponibilidad, estandarización, rendimiento, seguridad, etc.

8 REGLAS DEL DOMINIO O DEL NEGOCIO

En lo relativo a la especificación de requisitos también es importante prestar atención a las denominadas **Reglas del Dominio** que son una forma especializada de requisitos que establecen el modo en el que opera un determinado dominio. No son, por tanto, requisitos de la aplicación solicitados por el cliente o usuario, aunque, con frecuencia, los requisitos de una aplicación se ven afectados por dichas reglas que se originan en el dominio en el que se enmarca el sistema software. En ocasiones se denominan **Reglas del Negocio**, aunque este término resulta más limitado, ya que existen sistemas que no se crean para organizaciones, como por ejemplo juegos, predicción climática, logística militar, etc., así un simulador de clima incluirá reglas del dominio, relacionadas con ciertas leyes y relaciones físicas, que afectarán a los requisitos de la aplicación.

Las reglas del dominio o del negocio describen por tanto las características del dominio en el que se encuadra la organización y pueden, por ejemplo, constituir restricciones de los requisitos funcionales existentes, definir casos particulares, etc. Si las reglas del dominio no se satisfacen, el sistema puede no trabajar de forma adecuada. Ejemplos de reglas del dominio de diferentes tipos son las siguientes:

- El sistema deberá calcular la desaceleración del tren como $D_{tren} = D_{control} + D_{gradiente}$ ¹³.

¹³ Aunque podría haberse contemplado ya como un requisito funcional. O bien el requisito funcional podría haberse enunciado como El sistema deberá calcular la desaceleración del tren y la regla de dominio como La desaceleración del tren se calculará como $D_{tren} = D_{control} + D_{gradiente}$. Nótese además que son bastante habituales los Reglas del Dominio que se refieren a cálculos específicos, por ejemplo, vinculados a determinadas leyes físicas como en este ejemplo.

- El sistema deberá, por las restricciones en los derechos de autor, suprimir algunos documentos justo inmediatamente después de su llegada¹⁴.

Al intentar recopilar las reglas del dominio relacionadas con un determinado sistema software, podría obtenerse como conclusión que la información proporcionada por las mismas ya se hubiera contemplado en los requisitos funcionales. No obstante, siempre es interesante pensar de forma explícita sobre cuáles son las reglas del dominio relacionadas con un determinado sistema software para evitar que alguna de ellas, no considerada al enunciar los requisitos funcionales, pudiera quedar olvidada.

9 REQUISITOS DE INFORMACIÓN

También es importante prestar atención a los **Requisitos de Información** que, al igual que las reglas del dominio, son una forma especializada de requisitos, como por ejemplo el hecho de que el sistema deba almacenar determinada información sobre los socios del videoclub¹⁵, así por ejemplo un requisito de información podría ser el siguiente:

- El sistema deberá almacenar determinada información sobre los socios del videoclub: Número de socio
 - Número del DNI
 - Nombre y apellidos
 - Fecha de nacimiento
 - Sexo
 - Fecha de alta como socio
 - Dirección
 - Teléfonos
 - Películas alquiladas en un momento dado

El objetivo de los requisitos de información es el de **recopilar todos los datos con los que trabaja la organización, y por lo tanto el sistema**, y que **contienen información**, distinguiendo los documentos con los que se trabaja que son el soporte de la información, de los datos en sí que es la información que deberá manejar el sistema. [ver Figura 2]

¹⁴ Aunque podría haberse contemplado ya como un requisito no funcional.

¹⁵ Que estará asociado a un requisito funcional que hará referencia al alta del socio en el videoclub.

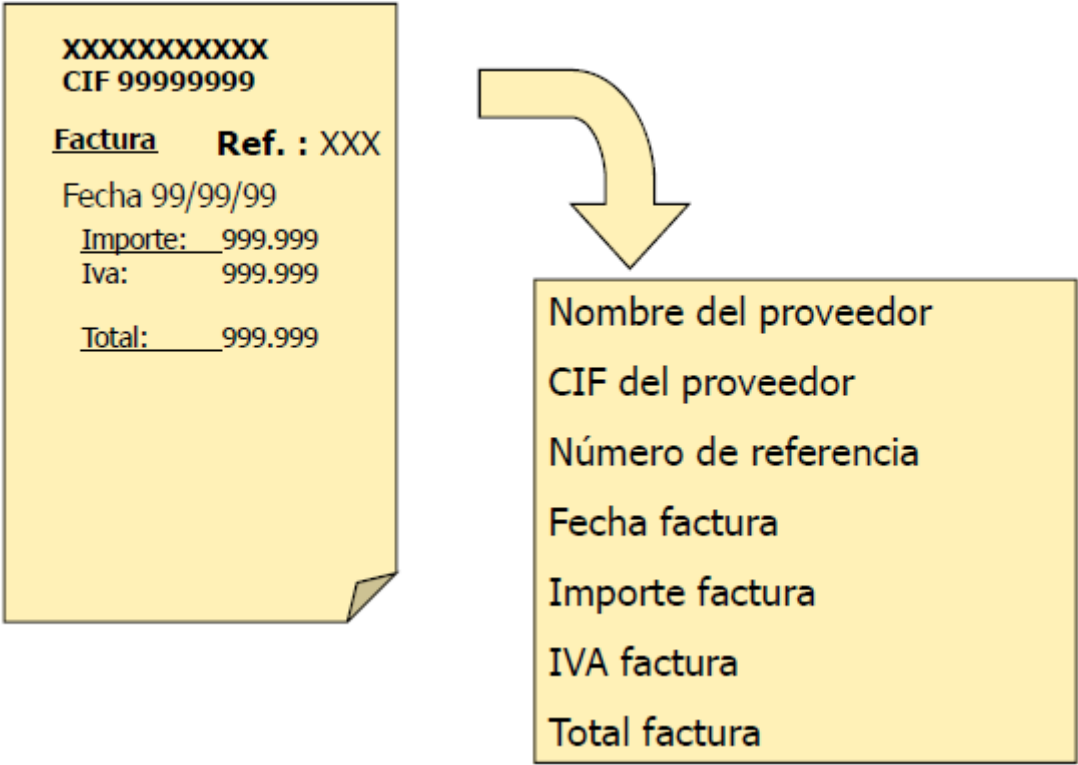


Figura 2: Documento vs Datos.

Además, para cada dato se podría recopilar la información necesaria como si de un término contenido en un diccionario se tratara para configurar un Diccionario de Datos del sistema software. [ver Tabla 1]

Nombre	Nombre Proveedor
Definición	Es el nombre del proveedor que suministra los productos.
Estructura	Cadena de 40 caracteres alfanuméricos.
Cuantificación	~ 100
Ejemplos	Coca-Cola, Carrefour...
Comentarios	Problemas de duplicación. Restricciones. Lista acotada de valores posibles. Reglas de cálculo (si el dato es calculado). Sinónimos, polisemias ¹⁶ , etc.

¹⁶ La Polisemia hace referencia a la pluralidad de significados que puede tener una palabra.

	[...]
--	-------

Tabla 1: Ejemplo de especificación de un término para un Diccionario de Datos.

Al intentar recopilar los requisitos de información relacionados con un determinado sistema software, podría obtenerse como conclusión que la información proporcionada por los mismos ya se hubiera contemplado en los requisitos funcionales. No obstante, siempre es interesante pensar de forma explícita sobre cuáles son los requisitos de información relacionados con un determinado sistema software para evitar que algún aspecto importante pudiera quedar olvidado. Además, la realización del Diccionario de Datos pudiera resultar aconsejable si los datos o información con la que va a trabajar el sistema es compleja.

10 DESCRIPCIÓN DE REQUISITOS

La descripción de los diferentes requisitos establecidos se especificará típicamente con el siguiente formato:

<id>: El <sistema> deberá <función/restricción>¹⁷

Además, para cada requisito establecido podrán identificarse y registrarse una serie de características, como por ejemplo las que a continuación se proponen [ver ejemplos en Tabla 2 y Tabla 3]:

- Identificador.
- Autor / Revisor.
- Tipo de requisito.
- Descripción.
- Fechas de creación / revisión.
- Prioridad o Importancia (escala numérica o enumerada: alta / media / baja... o PD – pendiente de determinar –). Este campo indica la importancia del cumplimiento del requisito para los clientes y usuarios. Se puede asignar un valor numérico o alguna expresión enumerada como alta, media, baja, etc. En el caso de que no se haya establecido aún la importancia, se puede indicar que aún está por determinar utilizando las siglas PD, que son equivalentes al TBD (To Be Determined) empleado en las especificaciones escritas en lengua inglesa.
- Urgencia (escala numérica o enumerada: hay presión / puede esperar... o PD – pendiente de determinar –). Este campo indica la urgencia del cumplimiento del requisito para los clientes y usuarios en el marco de un desarrollo incremental. Se puede asignar un valor numérico o una expresión enumerada como inmediatamente,

¹⁷ Esto podrá no ser cierto en algún caso, típicamente en los requisitos de tipo Reglas del Dominio y en los Requisitos de Sistema. Véase más información sobre estos tipos de requisitos en la sección del presente documento dedicada a dichos tipos de Requisitos de forma específica.

hay presión o puede esperar, o PD en el caso de que aún no se haya determinado la urgencia del requisito.

- **Estado** (escala enumerada: propuesto / aprobado o bien pendiente de negociación / pendiente de verificación / pendiente de validación / validado). Este campo indica el estado del requisito desde el punto de vista de su desarrollo.

Se podría utilizar alguna expresión enumerada de las que aparecen a continuación:

- **Pendiente de negociación** si tiene algún conflicto asociado pendiente de solución.
- **Pendiente de verificación** si no tiene ningún conflicto pendiente y está a la espera de verificación por parte de los ingenieros de software.
- **Pendiente de validación** si ya ha sido verificado y está a la espera de validación por parte de los clientes y usuarios.
- **Validado si** ya ha sido validado por clientes y usuarios.

O una clasificación más sencilla en la que solo se diferencie si el requisito se ha Propuesto o si ya ha sido Aprobado.

O bien asignar un valor numérico, o PD en el caso de que aún no se haya determinado el estado del requisito.

- **Estabilidad** (escala numérica o enumerada: alta / media / baja... o PD – pendiente de determinar –). Este campo indica la estabilidad del requisito, es decir, una estimación de la probabilidad de que pueda sufrir cambios en el futuro. Se puede asignar un valor numérico o una expresión enumerada como alta, media, baja, o PD en el caso de que aún no se haya determinado la estabilidad del requisito.
- **Datos Específicos: Para los Requisitos de Información**, este campo contiene una lista de los datos específicos asociados al concepto relevante, de los que pueden indicarse todos aquellos aspectos que se considere oportunos (descripción, restricciones, ejemplos, etc.).
- **Tiempo de Vida** (medio y máximo): **Para los Requisitos de Información**, este campo indica el tiempo de vida medio y máximo que se espera para cada ocurrencia del concepto relevante.
- **Ocurrencias simultáneas: Para los Requisitos de Información**, este campo indica el número medio y máximo de ocurrencias simultáneas del concepto relevante. Tanto este campo como el anterior denominado Tiempo de Vida permiten a los diseñadores prever determinadas necesidades del sistema a desarrollar en lo relativo al almacenamiento de información.
- [...]

A continuación, se presentan dos ejemplos:

CRQ ¹⁸ -0001	pagos firmados ¹⁹
Versión	1.0 (18/01/2012)
Dependencias	Ninguna
Descripción ²⁰	El sistema deberá requerir la firma del cliente para pagos a crédito.
Importancia	PD
Urgencia	PD
Estado	Validado
Estabilidad	PD

Tabla 2: Especificación detallada del requisito tipo Regla de Dominio CRQ 0001 – Pagos firmados.

IRQ ²¹ -0001	Producto ²²	
Descripción ²³	El sistema deberá almacenar la información correspondiente a producto. En concreto:	
Datos Específicos	Código universal de producto	
	Nombre del producto	
	Precio unitario del producto	
Tiempo de Vida	Medio	Máximo
	8 meses	5 años
Ocurrencias	Medio	Máximo

¹⁸ Siguiendo las recomendaciones, entre otros, de la IEEE, cada requisito se debe identificar por un código único y un nombre descriptivo. Con objeto de conseguir una rápida identificación, los identificadores de los requisitos de restricciones relativas a reglas del dominio se identifican con las siglas CRQ (Customer Requirements).

¹⁹ Restricción impuesta por la política de las compañías de autorización de crédito. Ejemplo de requisito del tipo Regla del Dominio.

²⁰ Para los requisitos de restricciones relativas a reglas del dominio o del negocio, este campo usa un patrón que se debe completar con la restricción o regla de negocio que se debe cumplir.

²¹ Siguiendo las recomendaciones, entre otros, de la IEEE, cada requisito se debe identificar por un código único y un nombre descriptivo. Con objeto de conseguir una rápida identificación, los identificadores de los requisitos relativos a la información que debe almacenarse o a sus restricciones comienzan con las siglas IRQ (Information Requirements).

²² Ejemplo de requisito del tipo Requisito de Información.

²³ Para los requisitos relativos a la información que debe almacenarse este campo usa un patrón que se debe completar con el concepto relevante sobre el que se debe almacenar información.

simultáneas	400	1000
Importancia	Vital	
Urgencia	PD	
Estado	Propuesto	
Estabilidad	PD	

Tabla 3: Especificación detallada del Requisito de Información IRQ 0001 – Producto.

11 EJEMPLOS DEL ENUNCIADO DE REQUISITOS

A continuación, se presentan ejemplos de la especificación de requisitos funcionales y no funcionales tal y como se enunciarían en una primera etapa del trabajo en las reuniones con los clientes y/o usuarios. [ver Texto 1]

Tal y como puede observarse, se utilizará el lenguaje natural para plantear una descripción lo más breve y sencilla posible, y siempre asequible para el cliente y/o usuario. Una vez que el trabajo esté más avanzado se podría pasar para cada requisito, de esta descripción inicial simple y breve a una descripción más detallada como la propuesta en el apartado 10 Descripción de Requisitos. [Ver ejemplos de la Tabla 2 y de la Tabla 3].²⁴

R1: El sistema deberá permitir a un funcionario crear una Reserva de Crédito asociada a una Partida Presupuestaria. – funcional –

R2: El sistema deberá almacenar todos los datos asociados a la ejecución de un presupuesto. – funcional –

R3: El sistema deberá mostrar los datos relativos a los horarios de las materias a los profesores y, si se solicita, la información de los estudiantes inscritos en una materia específica. – funcional –

R4: El sistema deberá permitir realizar búsquedas en base a diferentes criterios. – funcional –

R5: El sistema deberá presentar inicialmente la interfaz al usuario en el idioma español, aunque además dicha interfaz ha de estar disponible en inglés y en francés. – no funcional (de producto) –

²⁴ Típicamente la descripción inicial breve y sencilla corresponderá a los denominados Requisitos de Usuario. Véase más información sobre este tipo de requisito en la sección del presente documento dedicada a dicho tipo de Requisito de forma específica, en contraposición a los Requisitos de Sistema.

R6: El sistema deberá visualizarse y funcionar correctamente en la última versión de los navegadores Microsoft Internet Explorer y Mozilla Firefox. – no funcional (de producto) –

R7: El sistema no deberá tardar más de cinco segundos en mostrar los resultados de una búsqueda. Si se supera este plazo, el sistema detendrá la búsqueda y mostrará los resultados encontrados. – no funcional (de producto) –

R8: El sistema deberá contar con un manual de usuario conforme al formato definido en el manual de estilo del departamento. – no funcional (organizacional) –

R9: El sistema deberá desarrollarse de acuerdo con el proceso estándar XYZCo-SP-STAN-95. – no funcional (organizacional) –

R10: El sistema deberá cumplir las disposiciones recogidas en la Ley Orgánica de Protección de Datos de Carácter Personal. – no funcional (externo) –

R11: El sistema no divulgará a los operadores ninguna información personal sobre los clientes aparte de su nombre y su número de referencia. – no funcional (externo) –

Texto 1: Ejemplos del enunciado inicial de Requisitos.

12 CONSEJOS PARA LA REDACCIÓN DE REQUISITOS

Algunos consejos para la redacción de requisitos, tanto en su versión inicial breve y simple, como en su versión más detallada, son los siguientes:

- Utilizar el lenguaje de forma consistente.
- Evitar el uso de lenguaje excesivamente técnico inaccesible para el usuario, sobre todo en la especificación inicial de requisitos.
- Definir un formato estándar y utilizarlo para especificar todos los requisitos.
- Resaltar el texto para identificar las partes claves del requisito.
- Categorizar los requisitos, distinguiendo, por ejemplo, entre los requisitos obligatorios y los deseables, o entre los más urgentes y los menos urgentes.

13 REQUISITOS VS METAS

Debe tenerse en cuenta que los requisitos, especialmente los no funcionales, pueden ser muy difíciles de expresar con exactitud y que los requisitos imprecisos pueden ser difíciles de verificar. Así, un deseo general del usuario puede ser, por ejemplo, la facilidad de uso, mientras que, al expresar dicho deseo del usuario como un requisito no funcional verificable, debería utilizarse una frase que incluyera alguna medida que pudiera ser objetivamente evaluada. Podría decirse que, a la hora de redactar los requisitos no funcionales, es importante distinguirlos de las **Metas** del sistema, así:

- Una Meta expresa una intención general de usuario con respecto al sistema.
- Mientras que un requisito no funcional debe poder ser verificable utilizando alguna medida que pueda ser objetivamente evaluada.

Así, por ejemplo, una meta del sistema podría ser la siguiente:

- El sistema deberá ser fácil de usar por controladores experimentados y deberá estar organizado de tal manera que los errores de este tipo de usuarios se minimicen.

Y al transformarla en un requisito no funcional verificable podría quedar redactada de la siguiente manera:

- El sistema ofrecerá una serie de funciones de forma que los controladores experimentados deberán ser capaces de usar todas ellas después de un total de dos horas de entrenamiento. Tras este entrenamiento, el número promedio de errores cometidos por estos usuarios experimentados al usar el sistema no deberá exceder de dos por día.

Nótese que, si se cumple el requisito no funcional enunciado, se cumplirá con la meta, y, por otra parte, se cuenta con una base para negociar con el cliente en caso de que dicha negociación resulte necesaria.

14 REQUISITOS DE USUARIO Y REQUISITOS DE SISTEMA

Es importante señalar que algunos autores distinguen entre **Requisitos de Usuario y Requisitos de Sistema**. Así los requisitos de usuario serían declaraciones generales en lenguaje natural de los servicios del sistema y de las restricciones bajo las que el mismo debe operar. Mientras que los requisitos de sistema serían una descripción detallada de los servicios del sistema que constituye un contrato entre el cliente y el desarrollador y que contiene una especificación completa y consistente del sistema que sirve de base a los desarrolladores para progresar en su trabajo. Así, un requisito de usuario podría expresarse como:

- El sistema deberá permitir representar y acceder a archivos externos creados por otras herramientas.

El cual tendría los siguientes requisitos de sistema asociados:

- El usuario deberá poder definir el tipo de un nuevo archivo externo.

- Cada tipo de archivo tendrá una herramienta asociada, que se aplicará al archivo.
- Cada tipo de archivo se representará con un icono específico.
- El usuario deberá poder definir el icono que representa un tipo de archivo externo.
- Cuando el usuario selecciona un icono que representa un archivo externo, el efecto es aplicar la herramienta asociada con este tipo de archivo al archivo representado por el icono seleccionado.

De lo anterior se deduce que los requisitos funcionales de usuario describen el funcionamiento del sistema, típicamente mediante frases muy generales sobre lo que el sistema debería hacer, expresadas como objetivos del usuario con respecto al sistema. Mientras que los requisitos funcionales de sistema deben describir los servicios que hay que proporcionar con todo detalle, para lo cual se podrán utilizar además otras herramientas como los Casos de Uso.

15 PROCESOS DE LA INGENIERÍA DE REQUISITOS

Los procesos utilizados en la Ingeniería de Requisitos varían dependiendo del dominio de aplicación, del equipo humano implicado y de la organización que desarrolla el proyecto. Sin embargo, hay un número de actividades genéricas comunes a todos los procesos que comprenden no solo el estudio de la viabilidad del proyecto software y la **Elicitación o Extracción o Captura o Determinación de Requisitos**, sino también su **Análisis, Validación y Gestión**, es decir, que **no solo es necesario capturar los requisitos, sino que también es igualmente importante analizarlos, validarlos y gestionarlos adecuadamente**. Esto supone que, **tras la identificación del conjunto de requisitos, estos deben ser incluidos en un catálogo [ver ejemplo en Texto 2], analizados, con el objetivo de detectar inconsistencias, ambigüedades, duplicidad, escasez de información, etc., validados por los clientes/usuarios y gestionados para poder acomodar los cambios que pudieran sufrir.**

Requisitos Funcionales:

- Funciones principales del sistema:
 - Mantenimiento de datos de socios.
 - Generación de facturas con periodicidad variable (1, 2, 3, 6, 12 meses) a partir de cualquier mes.
 - Facturación con el formato exigido por la Caja de Ahorros.
 - Facturación mensual para recibos corrientes, y en cualquier momento para los no corrientes.
- Funciones de consultas:
 - Socios, facturas e impagados.
 - Lista detallada de facturas impagadas para poder proceder a su reclamación.
- Funciones de información:
 - Socios (datos personales, bancarios, cuota y periodicidad).

- Facturas (todas las facturas emitidas, sean cobradas o pendientes de pago).
- Funciones de interacción con otros sistemas:
 - Programa de contabilidad, para realizar los asientos correspondientes a cada mes.

Requisitos No Funcionales:

- De rendimiento:
 - No se especifican detalles.
 - Volumen de 500 socios.
- De frecuencia de tratamiento:
 - Facturación mensual típica de 250 socios, con picos de hasta 5.000 socios.
 - Los impagados suelen ser el 2% del volumen total facturado al mes.
- De seguridad:
 - Control de accesos: Una palabra clave para el usuario (secretaria).
 - Copias de respaldo: No especificado.
 - Integridad de la información: No especificado.
- De comunicaciones:
 - Ninguno. Todas las aplicaciones funcionan en el mismo ordenador.

Texto 2: Ejemplo sencillo y reducido de un Catálogo de Requisitos en su redacción inicial.

La **Elicitación** de Requisitos es el proceso mediante el cual los usuarios descubren, revelan, organizan y comprenden los requisitos que desean. Para este objetivo se utilizan técnicas como la observación o las entrevistas. El **Análisis** de Requisitos es el proceso de razonamiento sobre los requisitos obtenidos en la etapa anterior, detectando y resolviendo posibles inconsistencias o conflictos, coordinando los requisitos relacionados entre sí, etc. En esta etapa se utilizan técnicas de revisión como las listas de comprobación. La **Validación** de Requisitos es el proceso de confirmación, por parte de los usuarios, de que los requisitos especificados son válidos, consistentes, completos, etc. En esta etapa se utilizan igualmente técnicas de revisión. Finalmente, la **Gestión** de Requisitos es el proceso de manejar los requisitos que cambian durante el desarrollo del sistema. En esta etapa se utiliza sobre todo la ayuda de herramientas CASE.

15.1 ELICITACIÓN DE REQUISITOS

Durante la etapa de Elicitación de Requisitos el objetivo principal es descubrir los requisitos. Para lograr este objetivo, el personal técnico trabaja con los clientes y usuarios para descubrir el dominio de la aplicación, los servicios que se deben proporcionar y las restricciones

existentes. El número y tipo de agentes implicados²⁵ puede incluir a usuarios finales, expertos en el dominio al que pertenece la aplicación, ingenieros implicados en el desarrollo del sistema software y en su futuro mantenimiento, etc. **Problemas** que típicamente surgen durante esta etapa son los que a continuación se enumeran:

- Los clientes y/o usuarios finales de la aplicación no siempre tienen claro lo que realmente quieren.
- Diferentes agentes implicados pueden expresar los requisitos con sus propios términos que pueden resultar ambiguos para otros agentes implicados.
- Diferentes agentes implicados pueden plantear requisitos conflictivos.
- Factores políticos, organizativos u otros, pueden tener influencia en los requisitos.
- [...]

Las **actividades** que podrían realizarse durante la etapa de Elicitación de Requisitos serían las siguientes:

- Obtener información sobre el dominio del problema y sobre el sistema actual si ya existiera.
- Preparar y realizar las sesiones de trabajo: reuniones, entrevistas, etc., de elicitación y negociación de requisitos.
- Identificar y revisar los objetivos o metas del sistema.
- Identificar y revisar los requisitos funcionales.
- Identificar y revisar los requisitos no funcionales.
- Identificar y revisar los requisitos de información.
- Identificar y revisar las reglas del dominio que afecten al sistema.
- Priorizar metas y requisitos.

Las diferentes técnicas usadas para la captura de requisitos podrían ser las siguientes:

- **Talleres de Discusión²⁶ de Requisitos.** Su objetivo es lograr un acuerdo general sobre el alcance, riesgos y características más importantes del sistema software. Típicamente estarán dirigidos por un facilitador.
- **Entrevistas.** Las técnicas de recogida inicial de información pueden ser la observación directa, el estudio de documentos, etc., pero sin olvidarse de las entrevistas a los agentes implicados. Las entrevistas deberán ir orientadas a lograr un censo de los objetivos deseados, determinar las interfaces con otros proyectos o

²⁵ En los textos ingleses a los agentes implicados en un asunto o causa se les denomina stakeholders.

²⁶ En los textos ingleses a los talleres de discusión se les denomina workshops.

delimitar en lo posible el campo de estudio. Se entrevistará a varios de los implicados ya que cada persona del departamento tendrá su propia visión del sistema, la dirección, global pero difusa, mientras que los trabajadores tendrán una visión parcial pero más concreta y detallada. Se tendrán en cuenta las restricciones de medios, calendario, etc. El contenido de una ficha de entrevista a un puesto de trabajo podría ser el siguiente:

- Identificación:
 - Persona
 - Departamento
 - Puesto
- Tareas que realiza y descripción.
- Participación concreta en el proyecto de desarrollo de software.
- [...]

15.2 VALIDACIÓN DE REQUISITOS

Además, tal y como ya se ha comentado, a lo largo del desarrollo del proyecto software, se deben realizar actividades que confirmen que los requisitos definidos se cumplen. Así, la validación de los requisitos se centrará en demostrar que los requisitos definen el sistema que el cliente realmente desea. Aspectos que pueden controlarse en lo relativo a los requisitos son los siguientes:

- **Validez.** ¿El sistema proporciona las funciones que dan cobertura a las necesidades de los clientes?
- **Completo.** ¿Están recogidas todas las funciones solicitadas y las restricciones expresadas?
- **Consistencia.** ¿Hay conflictos y/o contradicciones, en los requisitos?
- **Verificabilidad.** ¿Pueden comprobarse los requisitos?
- **Comprensibilidad.** ¿Se han comprendido adecuadamente los requisitos?
- **Adaptabilidad.** ¿Se puede cambiar el requisito sin un gran impacto en otros requisitos?
- **Realismo.** ¿Pueden implementarse los requisitos con la tecnología y conocimientos actuales o con los recursos disponibles?
- **Trazabilidad.** ¿El origen del requisito está claramente establecido?

La **Trazabilidad** de requisitos se define como la habilidad para describir y seguir la vida de un requisito en ambos sentidos, hacia sus orígenes o hacia su implementación, a través de todas las especificaciones generadas durante el proceso de desarrollo del software. Es decir, que la trazabilidad es un proceso que debe permitir identificar los requisitos que originan cada

producto intermedio de un proyecto software, así como las características del producto final debidas a cada requisito. [ver Figura 3]

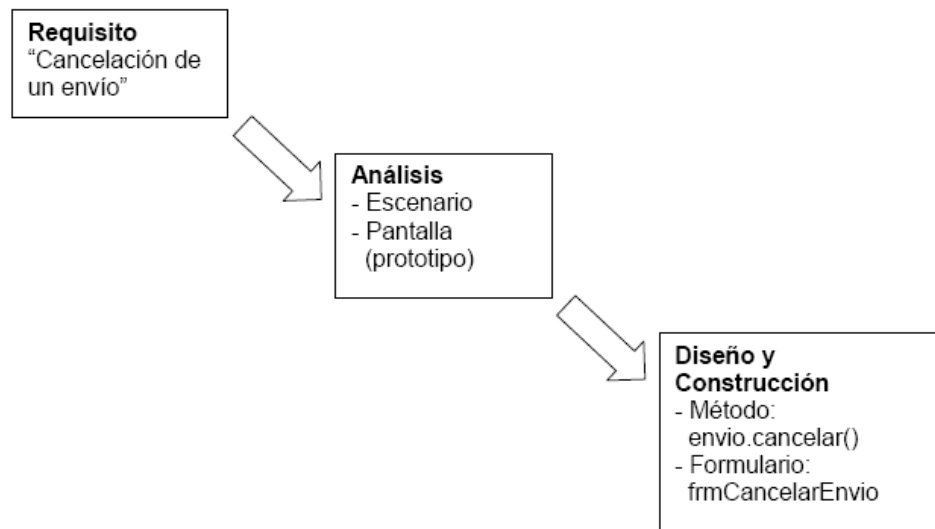


Figura 3: Trazabilidad de requisitos.

La trazabilidad se refiere por tanto a las relaciones entre los requisitos, sus fuentes y el diseño del sistema, así es posible hablar de:

- **Trazabilidad de las fuentes:** Enlace desde los requisitos a los participantes que los propusieron.
- **Trazabilidad entre requisitos:** Enlace entre requisitos dependientes.
- **Trazabilidad del diseño:** Enlace desde los requisitos al diseño.²⁷

15.3 GESTIÓN DE REQUISITOS

Además de la validación de los requisitos, será necesaria su gestión entendida como el **proceso de manejar los requisitos que cambian durante el desarrollo del sistema software**. Esto es necesario e importante, puesto que durante el proceso pueden emerger nuevos requisitos si hay variaciones en determinados elementos del dominio o, simplemente, si se llega a una mejor comprensión del sistema, o bien pueden aflorar diversos puntos de vista por lo que algunos requisitos pudieran resultar contradictorios. Así, durante el proceso de la ingeniería de requisitos es necesario, además de la identificación de los requisitos, o de definir políticas de trazabilidad que determinen la cantidad de información sobre las relaciones entre los requisitos que se mantiene, el planificar un proceso de gestión de los cambios en los requisitos, así como el soporte que darán las herramientas CASE, no solo en lo relativo al almacenamiento de los requisitos, o a la gestión de la trazabilidad, permitiéndose por ejemplo la recuperación automatizada de los enlaces entre los requisitos y los diferentes artefactos UML generados, sino también en lo relativo a la gestión de los requisitos cambiantes.

²⁷ Tal y como se observa en la Figura 3, se podría ir más allá y hablar de la Trazabilidad de la codificación en el caso de los requisitos funcionales que, como es sabido, se satisfacen añadiendo código.

16 CARACTERÍSTICAS DE LOS REQUISITOS

Cualquier conjunto de requisitos definido para un sistema debe poseer una serie de propiedades principales tanto individualmente como de forma conjunta. A continuación, se presentan las más importantes:

- **Necesario:** Un requisito es necesario si su omisión provoca una deficiencia en el sistema a construir, y además no puede ser reemplazado en forma alguna.
- **Conciso:** Un requisito es conciso si es fácil de leer y entender. Su redacción debe ser simple y clara para aquellos que vayan a tener que consultarlo en un futuro.
- **Completo:** Un requisito está completo si no necesita ampliar detalles en su redacción, es decir, si proporciona toda la información suficiente para su comprensión.
- **Consistente:** Un requisito es consistente si no es contradictorio con otro requisito.
- **No ambiguo:** Un requisito no es ambiguo cuando tiene una sola interpretación. Es decir que el lenguaje usado en su definición no deberá causar confusiones al lector.
- **Verificable:** Un requisito es verificable cuando puede ser comprobado mediante algún método de verificación propuesto: inspección, prueba, etc.

También es importante tener en cuenta las características que no deben poseer el conjunto de requisitos definido y/o ninguno de ellos individualmente:

- **Omisión:** No describir algo que sí debe contemplarse.
- **Ruido:** Datos que sin aportar nueva información, solo causan confusión.
- **Contradicción:** El tener dos o más requisitos en conflicto entre ellos.
- **Ambigüedad:** Presencia de texto que se presta a dos o más interpretaciones.
- **Que no sea comprobable:** Un requisito que no se puede verificar.

Si el conjunto de requisitos no está expresado con exactitud, aparecerán problemas puesto que un requisito expresado de forma ambigua puede ser interpretado de manera diferente por los ingenieros de software y por los clientes/usuarios. Además, es un objetivo importante el que la especificación de requisitos sea completa y consistente, completa en el sentido de que todos los servicios solicitados por el usuario están definidos y consistente en el sentido de que no existan definiciones contradictorias en los requisitos. En definitiva, cada requisito debería tener una única interpretación [ver Figura 4].

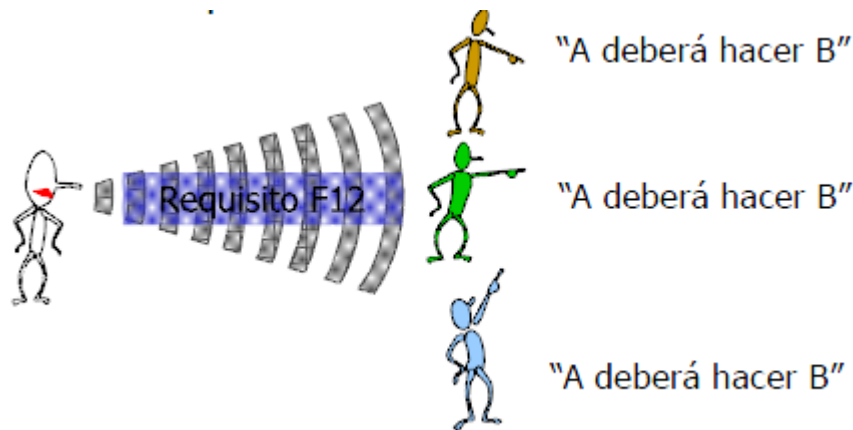


Figura 4: Los requisitos deben expresarse de forma no ambigua.

En este sentido señalar que en ocasiones el lenguaje natural dificulta el expresarse con claridad y precisión [ver Figura 5] a la hora de especificar los requisitos. Otra dificultad añadida puede ser el que se confundieran requisitos, mezclando por ejemplo los requisitos funcionales y los no funcionales o los de usuario con los de sistema, o bien que existiera conjunción de requisitos, es decir, que varios requisitos se expresen juntos, como un único requisito, cuando resultaría mucho más claro expresarlos por separado. A continuación, se presenta un ejemplo de mezcla de requisitos:

- De requisitos funcionales y no funcionales:
 - Para ayudar en la ubicación de una entidad en un diagrama, el sistema permitirá al usuario activar una cuadrícula en centímetros o en pulgadas, mediante una opción en el panel de control. Inicialmente, la cuadrícula estará desactivada. La cuadrícula se podrá activar o desactivar en cualquier momento y ponerse en centímetros o en pulgadas.
- El hecho de que el sistema deba soportar distintos sistemas de unidades es habitualmente considerado un requisito no funcional. Así, el requisito anterior podría haberse enunciado de la siguiente forma dividido en dos requisitos:
- El sistema deberá permitir al usuario ubicar una entidad en un diagrama, sobre una cuadrícula, mediante una opción en el panel de control. La cuadrícula estará inicialmente desactivada y se podrá activar o desactivar en cualquier momento – funcional –
 - El sistema deberá permitir al usuario visualizar la cuadrícula, bien en centímetros, bien en pulgadas, así como cambiar de una a otra opción en cualquier momento. – no funcional –
- De requisitos de usuario con requisitos del sistema:
 - La base de datos debe soportar el control de la configuración de los elementos y agrupaciones de elementos almacenados. Este control de la configuración debería permitir al usuario acceder a los elementos de una determinada versión sin especificar su nombre completo.

El funcionamiento de la base de datos debería considerarse un asunto de funcionamiento interno del sistema, transparente al usuario. Así, el requisito anterior podría haberse enunciado de la siguiente forma dividido en dos requisitos:

- El sistema deberá permitir al usuario acceder a los elementos de una determinada versión sin especificar su nombre completo. – de usuario –
- La base de datos del sistema deberá soportar el control de la configuración de los elementos y agrupaciones de elementos almacenados de forma que sea posible acceder a los elementos de una determinada versión sin especificar su nombre completo. – de sistema –

María tenía un cordero...

Tenía, del verbo Tener

1. tr. Asir o mantener asido algo.
 2. tr. **poseer** (|| **tener** en su poder).
 3. tr. **mantener** (|| sostener). U. t. c. prml.
 4. tr. Contener o comprender en sí.
 5. tr. **dominar** (|| sujetar).
 6. tr. **guardar** (|| cumplir). *Tener la palabra, la promesa*
 7. tr. **hospedar** (|| recibir huéspedes).
 8. tr. Estar en precisión de hacer algo u ocuparse en ello. *Tener clase Tener junta*
 9. tr. Juzgar, reputar, considerar. *Tener a alguien POR rico. Tener A gala, A honra algo. U. t. c. prml. Tenerse POR sabio*
 10. tr. Estimar, apreciar. *Tener EN POCO, EN MUCHO. U. t. c. prml.*
 11. tr. Emplear, pasar algún espacio de tiempo en un lugar o sitio, o de cierta manera. *Tener las vacaciones en Barcelona Tener un día aburrido*
 12. tr. **experimentar**. *Tener cuidado, vergüenza, miedo, hambre, calor, nervios*
-

cordero.

(Del lat. vulg. **cordarius*, der. de *cordus*, tardío).

1. m. Hijo de la oveja, que no pasa de un año.
 2. m. Piel de este animal adobada.
 3. m. Hombre manso, dócil y humilde.
 4. m. por antonom. Jesucristo, Hijo de Dios.
- ...

Figura 5: El lenguaje natural a veces es ambiguo.

Dada su, en ocasiones, falta de precisión y ambigüedad, podría plantearse la utilización de alternativas al lenguaje natural como, por ejemplo, la utilización de un lenguaje natural estructurado que mantiene la expresividad y comprensión del lenguaje natural, pero delimitando la terminología utilizada y haciendo uso de plantillas para describir las funciones que ejecuta el sistema. Otra alternativa para la captura de requisitos, no obstante, no muy utilizada, serían las notaciones gráficas, que consistirían en utilizar un lenguaje gráfico complementado con anotaciones en lenguaje natural estructurado.

17 DIFICULTADES Y BENEFICIOS DE LA DEFINICIÓN DE REQUISITOS

En definitiva, el definir el conjunto de requisitos de un sistema hará que nos enfrentemos a ciertas dificultades puesto que:

- Los requisitos no son obvios y provienen de muchas fuentes.
- Los requisitos son, en ocasiones, difíciles de expresar en palabras de forma no ambigua.
- Existen diferentes tipos de requisitos y diferentes niveles de detalle.
- La cantidad de requisitos definida para un sistema puede resultar difícil de manejar.
- Los requisitos son difícilmente comparables, ya que existirán requisitos, por ejemplo, más difíciles de cumplir, más importantes o estables que otros.
- Existen interrelaciones e interdependencias entre los requisitos.
- Ciertos requisitos pueden cambiar a lo largo del ciclo de desarrollo del sistema software.
- El conjunto de requisitos debería ser único y diferente para cada sistema.

Como contrapartida, el contar con un conjunto de requisitos adecuadamente definido para un determinado sistema proporcionará los siguientes beneficios:

- Permitirá realizar una estimación de los costes, los plazos y los recursos necesarios, más fiable que si no se hubieran definido los requisitos de forma adecuada.
- Permitirá disminuir costes y minimizar los potenciales retrasos y riesgos del proyecto de desarrollo software, ya que una especificación de requisitos no ajustada a las necesidades del cliente supondrá probablemente la necesidad de hacer rectificaciones en fases avanzadas del citado proyecto.
- Mejorará la comunicación entre los implicados en el desarrollo del sistema software, puesto que representa una forma de consenso entre clientes y desarrolladores. Sin este consenso, el proyecto software difícilmente tendrá éxito.
- Evitará rechazos de los usuarios finales, puesto que un conjunto de requisitos adecuadamente definidos y ajustados a las necesidades del cliente es la mejor garantía de que el sistema software resultante satisfacerá sus necesidades. Además, es una forma de involucrar al cliente en el proyecto de desarrollo del sistema software desde su comienzo puesto que se le obliga a considerar los requisitos de este cuidadosamente.

18 EL DOCUMENTO DE ESPECIFICACIÓN DE REQUISITOS DEL SOFTWARE

El documento de requisitos generado será la declaración oficial de lo que se necesita construir y se denominará Documento de Especificación de Requisitos del Software (ERS)²⁸. Este documento debería incluir tanto los requisitos del usuario como la especificación detallada de los requisitos del sistema. No obstante, debe tenerse en cuenta que no es un documento de diseño ya que debe indicar qué es lo que el sistema debe hacer, y con qué restricciones, pero no cómo va a hacerlo. La especificación de requisitos del software consensuada debe caracterizarse por ser no ambigua, completa, consistente, fácil verificar y modificar, así como fácil de identificar el origen y las consecuencias de cada requisito y fácil de utilizar durante la fase de explotación y mantenimiento en la medida en que resulte necesario.

El estándar IEEE 29148-2011 para la Especificación de Requisitos del Software (ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes -- Requirements engineering)²⁹ dicta la siguiente estructura:

- Introducción.
- Descripción general.
- Requisitos específicos.
 - Cubriría los requisitos funcionales y no funcionales.
- Apéndices.
- Índice.

²⁸ En los textos ingleses, la Especificación de Requisitos del Software (ERS) se denomina Software Requirements Specification (SRS).

Nótese, que también podría generarse, simplemente, un compendio o catálogo de requisitos con una estructura más informal y flexible.

²⁹ Véase <http://standards.ieee.org/findstds/standard/29148-2011.html> para más información. Nótese que este documento en el formato propuesto iría más allá de un simple catálogo de requisitos e incluiría el contenido propuesto para el Documento de Visión del Sistema y para el Glosario del proyecto.

Table of Contents	
1. Introduction	
1.1 Purpose	
1.2 Scope	
1.3 Definitions, acronyms, and abbreviations	
1.4 References	
1.5 Overview	
2. Overall description	
2.1 Product perspective	
2.2 Product functions	
2.3 User characteristics	
2.4 Constraints	
2.5 Assumptions and dependencies	
3. Specific requirements (See 5.3.1 through 5.3.8 for explanations of possible specific requirements. See also Annex A for several different ways of organizing this section of the SRS.)	
Appendixes	
Index	

Figura 6: Propuesta de esquema para el Documento de Especificación de Requisitos del Software extraída del estándar IEEE 29148-2011 para la Especificación de Requisitos del Software (ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes -- Requirements engineering).

No obstante, cada empresa, equipo, etc., podría usar para el Documento de Especificación de Requisitos del Software la estructura que considere más oportuna, o se adapte mejor a las características específicas de cada proyecto. Así, un ejemplo de estructura para el Documento de Especificación de Requisitos del Software podría ser la siguiente [ver ejemplo parcial en Texto 3].

Estructura:

- **Visión:**
 - **Introducción:** Ámbito y alcance del Proyecto. Describiría la necesidad de crear el sistema software, sus funciones y cómo trabajará con otros sistemas.
 - **Participantes en el proyecto** software. Tanto los desarrolladores de software como los clientes y usuarios.
 - **Objetivos del sistema.** Tanto los usuarios, como quizás otros sistemas externos, etc., que utilizarán el sistema para satisfacer sus objetivos.
 - **Visión general del producto:** Visión global del sistema y de sus características.
 - **Glosario de términos.** Definiría los términos técnicos utilizados en el documento que, típicamente, estarían relacionados con el dominio de la aplicación.
- **Resumen de entrevistas a los participantes en el proyecto.**

- Catálogo de requisitos del sistema. Servicios que se proveen al usuario junto con los requisitos no funcionales del sistema.
 - **Requisitos Funcionales**³⁰:
 - Diagrama inicial de Casos de Uso.
 - Definición de Actores.
 - Definición de Casos de Uso.
 - **Requisitos no Funcionales**.
 - **Requisitos de Información**:
 - Diccionario de Datos.
 - **Reglas del Dominio**.
- **Matrices de Rastreabilidad**. **Matriz** objetivo–requisito, de forma que para cada objetivo se pueda conocer con qué requisitos está asociado. El formato de una matriz de rastreabilidad podría ser el siguiente:

	OBJ-01	OBJ-02	...	OBJ-n
IRQ-01	•	•		
IRQ-02		•		
...				
FR-01	•			
FR-02	•	•		
...				
NFR-01				•
NFR-02		•		
...				

Ejemplo:

- Visión:
 - Introducción:
 - Se prevé una aplicación de Punto de Venta (PDV) de próxima generación, PDV NuevaEra, con flexibilidad para poder soportar variación en las reglas del negocio, múltiples mecanismos de

³⁰ Los Casos de Uso se utilizan en ocasiones como técnica para la extracción de requisitos funcionales. Véase más información en el documento dedicado a este tema de forma específica.

terminal e interfaz de usuario, y la integración con múltiples sistemas de terceras partes.

- Oportunidad del negocio: Los productos PDV existentes no son adaptables al negocio del cliente, etc. Además, no permiten su extensión de manera adecuada cuando se incrementan los terminales y crece el negocio. Y ninguno permite trabajar en línea o desconectado, adaptándose de forma dinámica dependiendo de los fallos, etc.
- [...]
- Participantes en el proyecto:
 - Descripción del personal involucrado y usuarios.
- Objetivos del Sistema:
 - Objetivo de alto nivel: El sistema deberá procesar las ventas de modo rápido, robusto e integrado.
 - Objetivos secundarios: Los usuarios y los sistemas externos, necesitan un sistema para satisfacer sus objetivos:
 - Cajero: Procesar las ventas, gestionar las devoluciones, abrir y cerrar caja.
 - Administrador del sistema: Gestionar los usuarios, gestionar la seguridad, etc.
 - Sistema de actividad de ventas: Analizar los datos de las ventas, etc.
 - [...]
- Visión general del producto:
 - El PDV NuevaEra residirá, normalmente, en tiendas, si se utilizan terminales móviles se encontrarán muy próximos a la red de la tienda, en el interior o en el exterior.
 - Proporcionará servicios al usuario y colaborará con otros sistemas.
 - Principales características del sistema:
 - Entrada de ventas.
 - Autorización de pagos (crédito, débito, cheque).
 - Administración de la seguridad del sistema, de sus usuarios, de los códigos empleados, etc.
 - Procesamiento automático de ventas sin conexión cuando resulte necesario.
 - [...]

- [...]

Nota de los profesores: Si detectas algún error, piensas que alguna información está presentada de manera confusa, o que sobra o falta algún contenido, por favor, envía un mensaje a mperez@tel.uva.es. Gracias por anticipado. Vuestras sugerencias se utilizarán para mejorar esta documentación.