

# REQUISITOS

## INGENIERÍA DE SOFTWARE I

2º DE GRADO EN INGENIERÍA INFORMÁTICA  
CURSO 2020/2021

Francisco José García Peñalvo / fgarcia@usal.es

Alicia García Holgado / aliciagh@usal.es

Andrea Vázquez Ingelmo / andreavazquez@usal.es

Departamento de Informática y Automática

Universidad de Salamanca





# 1. INGENIERÍA DE REQUISITOS

# PROBLEMÁTICA

*"Nunca sopla viento favorable para el que no sabe a dónde va"*

Seneca (55a.C - 39d.C)

*"La correcta obtención de los requisitos es uno de los aspectos más críticos de un proyecto software, independientemente del tipo de proyecto que se trate, dado que una mala captura de los mismos es la causa de la mayor parte de los problemas que surgen a lo largo del ciclo de vida"*

(Johnson, 1995)

Problemas en la  
obtención de requisitos



Crisis del software

*"La parte más difícil de construir de un sistema software es decidir qué construir. [...] Ninguna otra parte del trabajo afecta más negativamente al sistema final si se realiza de manera incorrecta. Ninguna otra parte es más difícil de rectificar después"*

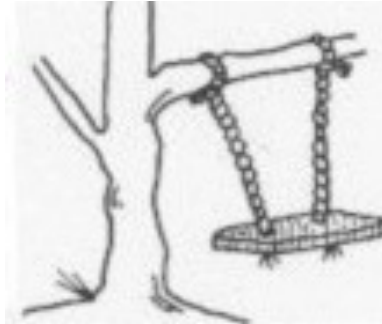
(Brooks, 1995)

*"El coste de un cambio en los requisitos, una vez entregado el producto, es entre 60 y 100 veces superior al coste que hubiera representado el mismo cambio durante las fases iniciales de desarrollo"*

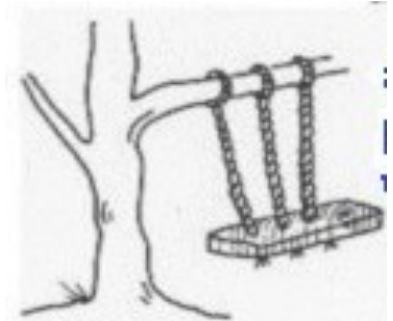
(Pressman, 2000)

# PROBLEMÁTICA

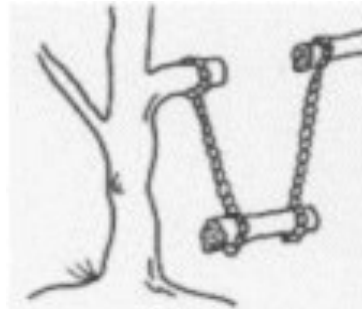
**Esto es lo que pidió el usuario**



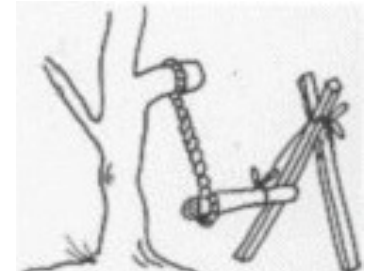
**El programador lo escribió así**



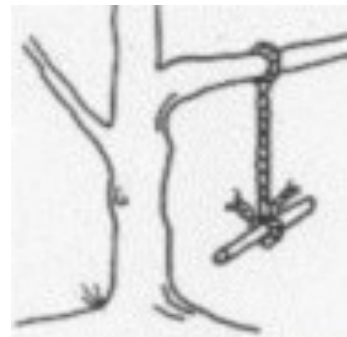
**El analista lo vio de esta forma**



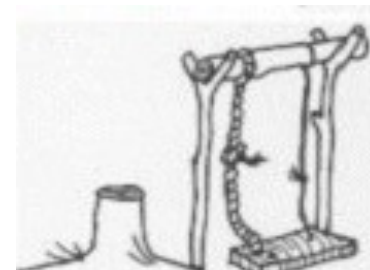
**Esto es lo que quería el usuario**



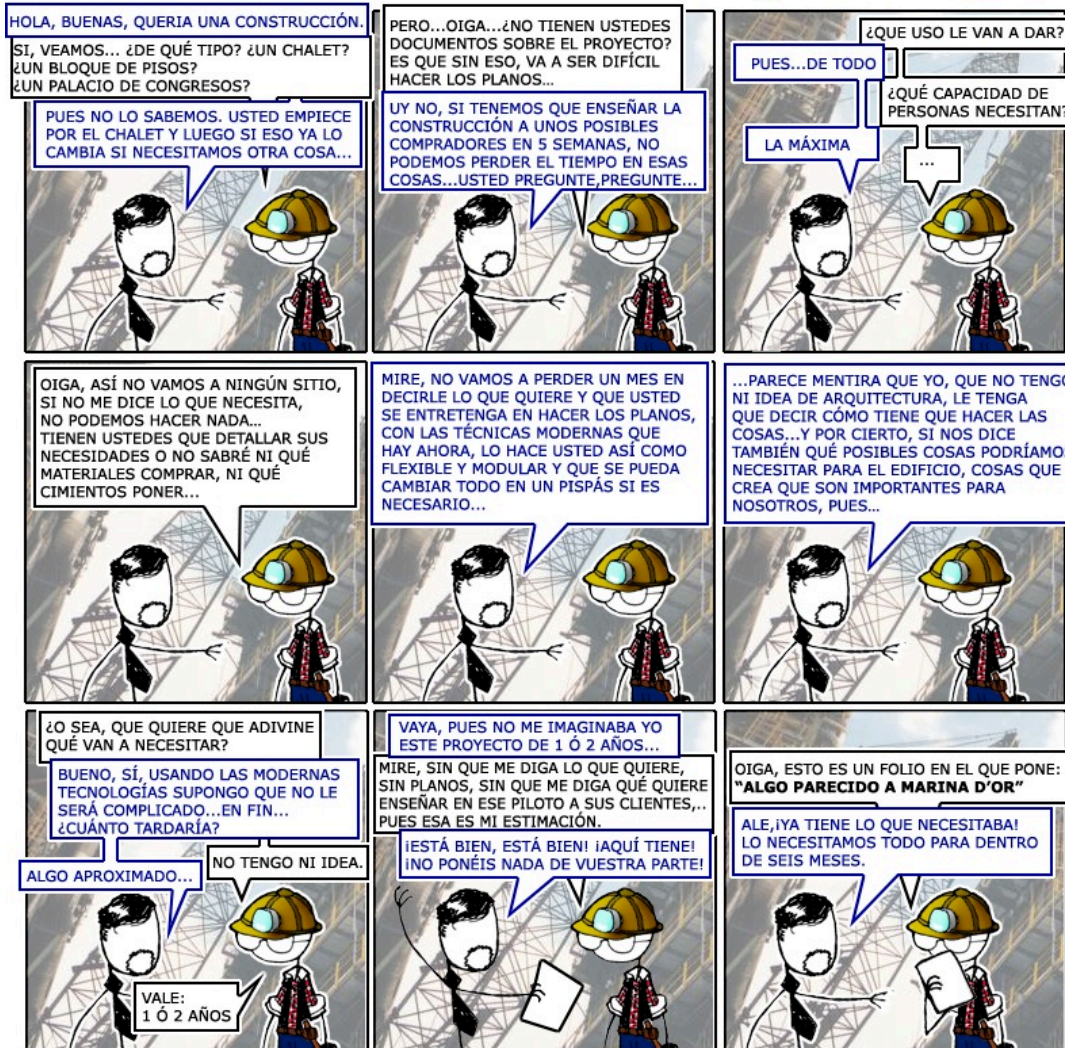
**Así se diseñó el sistema**



**Así funciona el sistema en la actualidad**



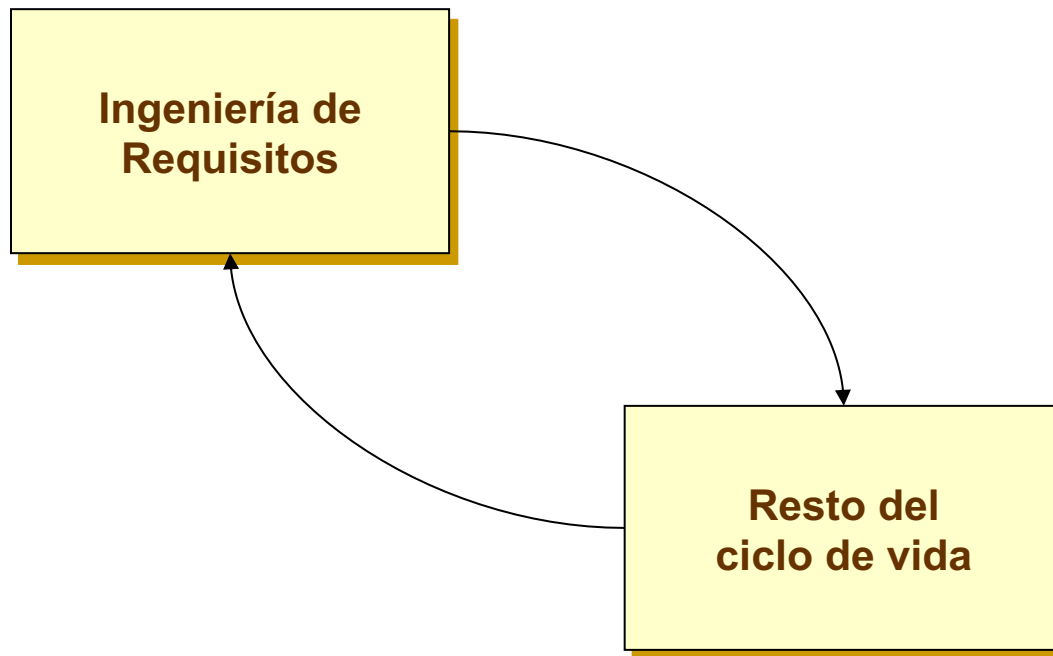
# SI EL SECTOR DE LA CONSTRUCCIÓN TUVIESE QUE TRABAJAR COMO LOS INFORMÁTICOS...



(BASADO EN HECHOS REALES LITERALES)

# VISIÓN GLOBAL

Primera fase del ciclo de vida del *software* en la que se produce una especificación a partir de ideas informales



# VISIÓN GLOBAL

- Deben obtenerse y documentarse
  - Los requisitos de información
  - Los requisitos funcionales
  - Los requisitos no funcionales
  - Los criterios para medir el grado de su consecución
- El proceso de desarrollo de dicha especificación de requisitos es lo que se conoce como **ingeniería de requisitos**
- Importancia creciente de
  - El correcto entendimiento (obtención), documentación (especificación) y validación de las necesidades de los usuarios y clientes
  - La medida de la calidad de los sistemas en función del grado de satisfacción de los usuarios

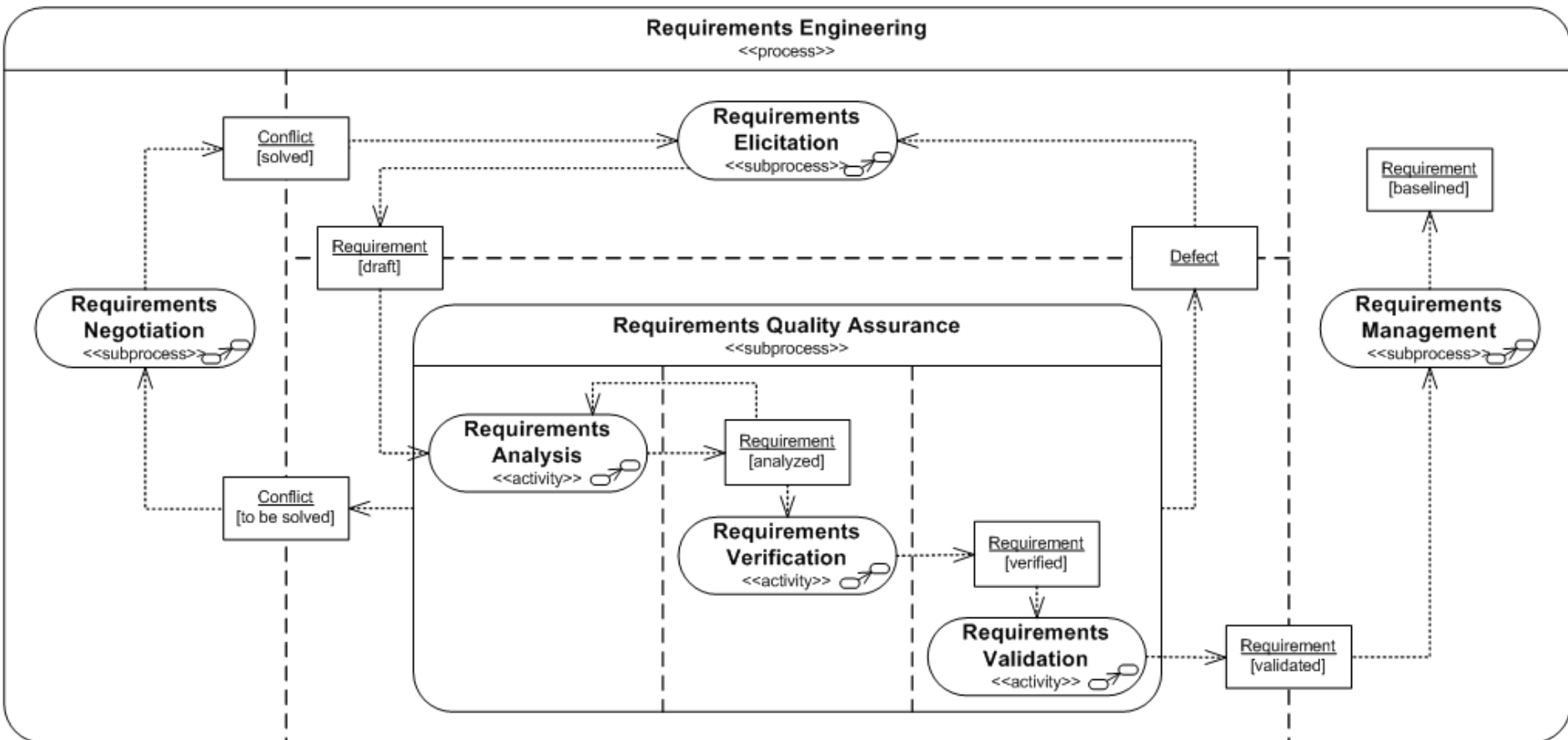
# DEFINICIÓN DE INGENIERÍA DE REQUISITOS

Un proceso sistemático de desarrollo de requisitos mediante un **proceso cooperativo** consistente en **analizar** el problema, **documentar** las observaciones resultantes en una variedad de formatos de representación y **comprobar la exactitud** de la comprensión conseguida

(Loucopoulus y Karakostas, 1995)



# PROCESO DE INGENIERÍA DE REQUISITOS



Proceso de Ingeniería de Requisitos (solo se muestran algunos productos)

# FACTOR HUMANO

- La comunicación es uno de los aspectos más destacables en la ingeniería de requisitos



- Esta característica hace de la ingeniería de requisitos una disciplina especialmente compleja al intervenir el factor humano
- Este factor es el responsable de que la Ingeniería de Requisitos tenga aspectos sociales y culturales y no solo técnicos (Goguen, 1994)





## 2. REQUISITOS

# ¿QUÉ DESCRIBE UN REQUISITO?

Una utilidad para el usuario

- *“El tratamiento de textos ha de incluir la comprobación y corrección gramatical”*

Una propiedad general del sistema

- *“El sistema ha de garantizar que la información personal solamente será accesible mediante autorización explícita”*

Una restricción general del sistema

- *“El sensor ha de muestrearse 10 veces por segundo”*

Cómo llevar a cabo cierto cálculo

- *“Calificación final = nota examen + 2\*nota trabajo + 2/3 nota ejercicios”*

Una restricción sobre el desarrollo del sistema

- *“El sistema ha de implementarse en C#”*

# CONCEPTO DE REQUISITO

**(a)** Una condición o capacidad que un usuario necesita para resolver un problema o lograr un objetivo. **(b)** Una condición o capacidad que debe tener un sistema o un componente de un sistema para satisfacer un contrato, una norma, una especificación u otro documento formal. **(c)** Una representación en forma de documento de una condición o capacidad como las expresadas en **(a)** o en **(b)** (IEEE, 1999a)

Una propiedad que debe exhibirse para solucionar algún problema del mundo real (Sawyer y Kontoya, 2001)

# CONCEPTO DE REQUISITO

Aparente simplicidad del concepto

Es frecuente encontrar el término **requisito** calificado con adjetivos que pueden resultar confusos en un primer momento

- de sistema
- *hardware*
- *software*
- de usuario
- de cliente
- funcional
- no funcional
- ...

# DIMENSIONES DE LOS REQUISITOS

- La gran cantidad de calificativos que se aplican al término requisito muestra distintos aspectos ortogonales que a menudo se consideran aisladamente
- Para clarificar la situación es frecuente identificar dimensiones para clasificar los requisitos
  - Ámbito
  - Característica que define
  - Audiencia
  - Representación

# DIMENSIONES DE LOS REQUISITOS

## Ámbito

- Indica en qué contexto se debe entender el requisito
  - *Sistema, Software, Hardware*

## Característica que define

- Los requisitos se clasifican en función de la naturaleza de la característica del sistema que se especifica
  - *Requisitos funcionales, Requisitos no funcionales, de Información*

## Audiencia

- Indica a quién está dirigido el requisito, es decir, las personas que deben ser capaces de entenderlo. Implica un nivel de descripción determinado
  - *Requisitos-C, Requisitos-D* (Rombach, 1990; Brackett, 1990)
  - *Requisitos de usuario o Requisitos de cliente, Requisitos software* (Mazza et al., 1994)
  - *Requisitos de usuario, Requisitos de sistema, Especificaciones de diseño* (Sommerville, 2002)

## Representación

- Establece la forma cómo se definen los requisitos
  - *Formal, Semiformal, No formal*



# DIMENSIONES DE LOS REQUISITOS

## Definición de requisitos de usuario

1. El *software* debe proporcionar un medio para la representación y acceso a ficheros externos creados por otras herramientas

## Especificación de requisitos del sistema

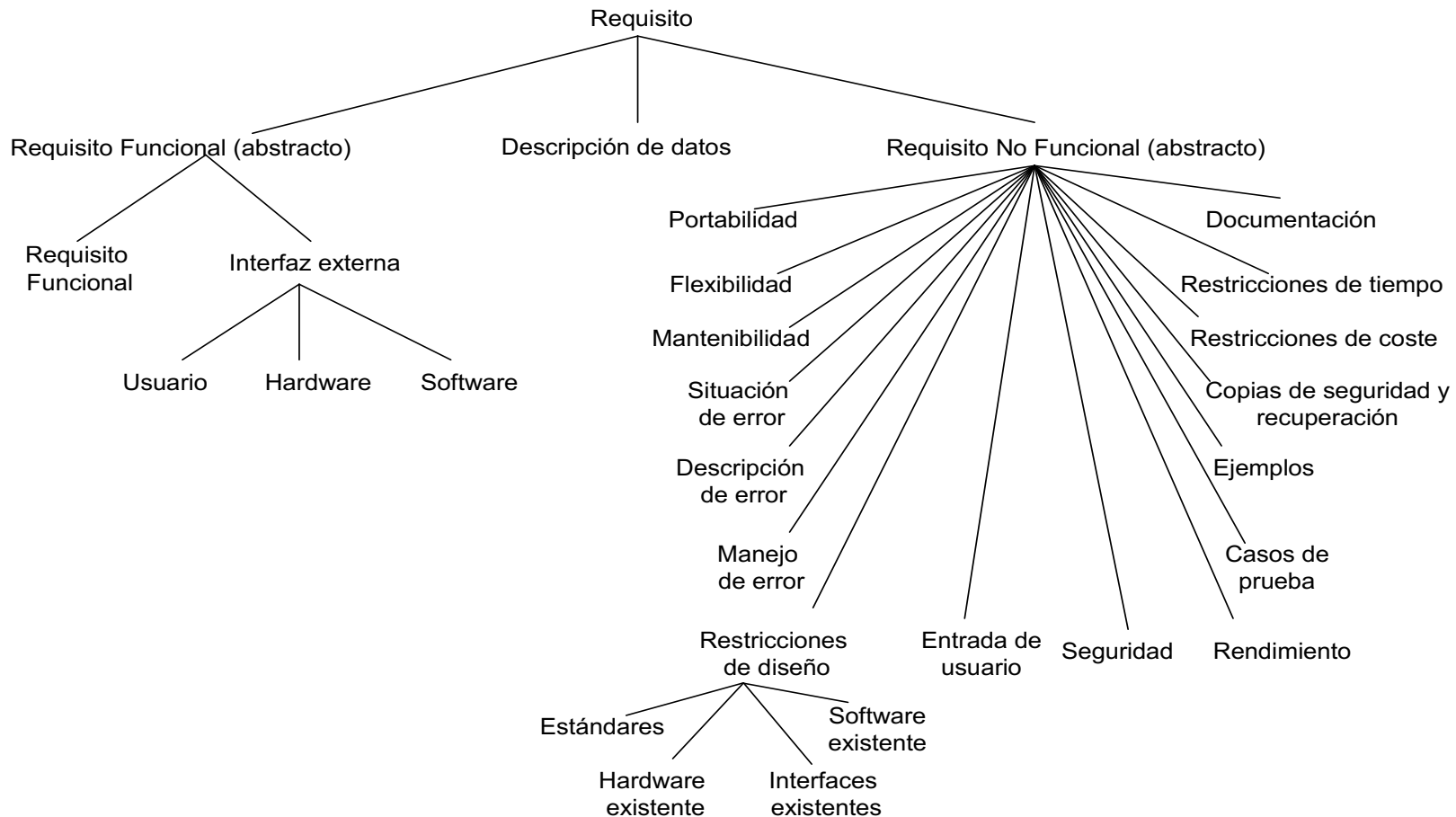
- 1.1 Hay que proporcionar al usuario utilidades para definir el tipo de los ficheros externos
- 1.2 Cada tipo de fichero externo tendrá asociado una herramienta que podrá ser aplicada al fichero
- 1.3 Cada tipo de fichero externo podrá estar representado mediante un icono específico en la interfaz del usuario
- 1.4 Se proporcionarán utilidades para que el usuario pueda definirse el tipo de icono que asociará a cada tipo de fichero
- 1.5 Cuando un usuario seleccione un icono que representa un tipo de fichero externo, el efecto de la selección será la aplicación de la herramienta asociada con el tipo de fichero externo al fichero representado por el icono seleccionado

# DIMENSIONES DE LOS REQUISITOS

K. Pohl (1997) establece una completa clasificación de requisitos, RSM (*Requirements Specification Model*)

- Las principales categorías son
  - Requisitos funcionales
    - Describen la funcionalidad o los servicios que se espera que este proveerá
      - De usuario: descripción general
      - De sistema: descripción detallada (función, entradas, salidas, etc.)
  - Requisitos de datos
  - Requisitos no funcionales

# DIMENSIONES DE LOS REQUISITOS



Jerarquía de especialización de RSM – adaptado de (Pohl, 1997)

# DIMENSIONES DE LOS REQUISITOS

*“El sistema ha de mantener el registro de todo el material de la biblioteca incluyendo libros, revistas, vídeos, informes, CD-Roms...”*

- **Requisito general** expresado en términos generales. Qué tiene que hacer el sistema

*“El sistema debe permitir a los usuarios buscar un ejemplar por título, autor o ISBN”*

- **Requisito funcional** que define una parte de funcionalidad del sistema

*“La interfaz del usuario ha de implementarse mediante un navegador web”*

- **Requisito de implementación**, establece cómo ha de implementarse el sistema

*“El sistema ha de soportar al menos 20 transacciones por segundo”*

- **Requisito de rendimiento** que especifica el rendimiento mínimo aceptable para ese sistema

# PROBLEMAS CON LOS REQUISITOS

- Los requisitos no reflejan las necesidades reales del cliente
- Requisitos inconsistentes o incompletos
- El cambio de requisitos, una vez acordados, es muy costoso
- Problemas de comunicación
  - Incomprensiones entre los clientes, los que desarrollan los requisitos y los ingenieros de *software* que desarrollan o mantienen el sistema

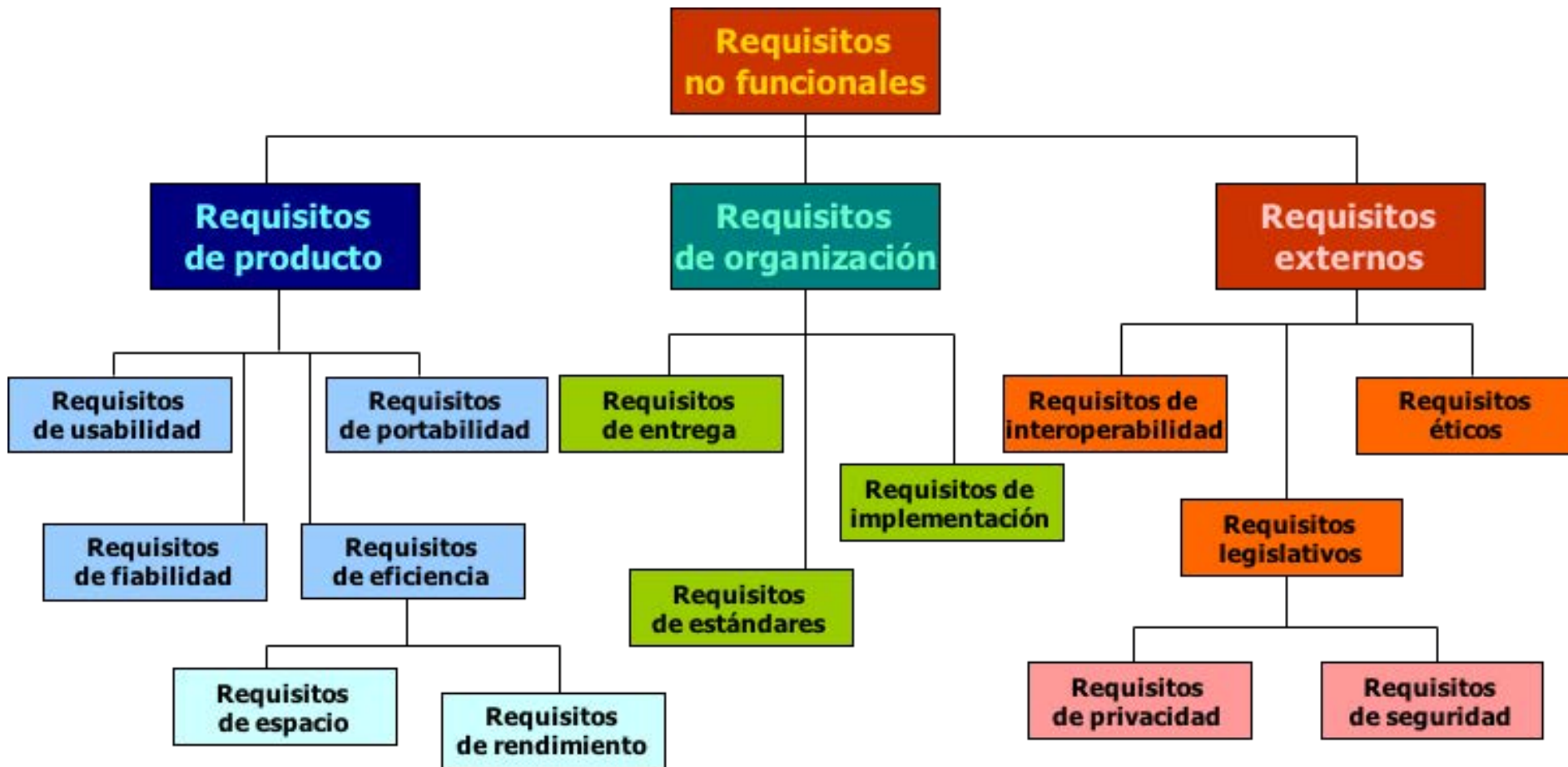
# REQUISITOS NO FUNCIONALES

- Requisitos no relacionados directamente con la funcionalidad del sistema
- Pueden estar relacionados con propiedades emergentes del sistema
- Pueden describir restricciones al producto a desarrollar
- Pueden describir restricciones externas del sistema
- Definen las cualidades globales que el sistema ha de exhibir
- Suelen hacer referencia al sistema considerado de forma global
- Suelen ser requisitos más críticos que los requisitos funcionales
- Suelen ser difíciles de verificar

# REQUISITOS NO FUNCIONALES

- Clasificación de los requisitos no funcionales (Sommerville, 2002)
  - **Requisitos de producto**
    - Especifican el comportamiento del producto
      - Tiempo de respuesta, memoria requerida, fiabilidad, portabilidad, usabilidad, etc.
  - **Requisitos de organización**
    - Se derivan de las políticas y procedimientos existentes en la organización del cliente y en la del desarrollador
      - Estándares de proceso, lenguajes de programación, métodos de diseño, estándares de documentación, etc.
  - **Requisitos externos**
    - Factores externos al sistema y de su proceso de desarrollo
      - Interoperabilidad, éticos, legislativos, privacidad, seguridad, etc.

# REQUISITOS NO FUNCIONALES





# REQUISITOS NO FUNCIONALES

*“El máximo espacio de almacenamiento ocupado por el sistema debe ser de 8 MB porque el sistema debe alojarse completamente en una memoria de solo lectura e instalarse en el coche”*

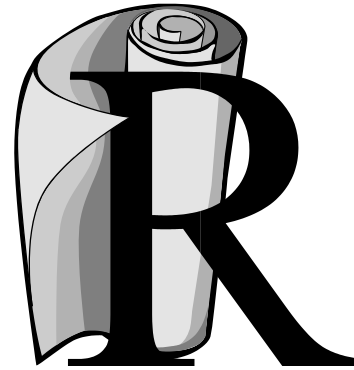
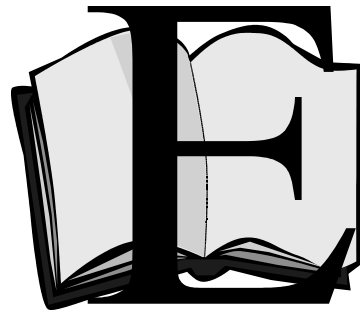
- **Requisito de producto** que define una restricción en el tamaño del producto

*“El proceso software y los documentos a realizar deben conformar el proceso y los estándares de documentación recogidos en la norma TELMo-ES-2003”*

- **Requisito de organización** que especifica que el sistema debe desarrollarse de acuerdo a un proceso estándar dentro de la compañía

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

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



### **3. ESPECIFICACIÓN DE REQUISITOS DEL SOFTWARE**

# ESPECIFICACIÓN DE REQUISITOS

- Los requisitos se recogen en documentos técnicos que reciben el nombre genérico de ERS (Especificación de Requisitos del *Software*)
- Este documento debe contemplar tanto los requisitos–C como los requisitos–D
- Hay metodologías que abogan por la separación de estas representaciones en dos documentos diferentes
  - DRS (Documento de Requisitos del Sistema), también denominado catálogo de requisitos, donde se recogen los requisitos–C
  - ERS propiamente dicha, donde se recogen los requisitos–D
- En la realización de una ERS participan
  - Ingenieros de *software* (analistas)
  - Clientes y usuarios

# PROPIEDADES DE LAS ERS

## Comprensible por clientes y usuarios

- Una especificación debe servir como canal de comunicación entre los participantes en el proceso de ingeniería de requisitos
- La mejor forma de lograr esta comunicación es pensar en la audiencia a la que van dirigidos los requisitos

## No ambigua

- Cada requisito solo tiene una interpretación

## Completa

- Incluye todos los requisitos significativos
- Define la respuesta a todo tipos de entradas
- Es conforme con el estándar de especificación a cumplir
- Están etiquetadas y referenciadas todas las figuras, tablas, etc.

# PROPIEDADES DE LAS ERS

## Consistente

- No hay conflictos ni contradicciones
- Es consistente externamente si y solo si todo requisito contenido en ella no está en conflicto con otros documentos de nivel superior
- Es consistente internamente si y solo si no existen conflictos entre los requisitos que contiene
- Tipos de conflictos entre requisitos (Davis, 1993)
  - **Conflictos de conducta.** Dos o más requisitos especifican conductas distintas del sistema para las mismas condiciones y el mismo estímulo externo
  - **Conflictos de términos.** Se utilizan términos distintos para referirse al mismo concepto
  - **Conflictos de característica.** Dos o más requisitos especifican aspectos contradictorios para la misma característica del sistema
  - **Conflictos temporales.** Dos o más requisitos exigen características temporales contradictorias al sistema

# PROPIEDADES DE LAS ERS

## Verificable

- Todo requisito contenido en ella es verificable. Existe un proceso finito y de coste razonable por el que una persona o una máquina pueda comprobar que el sistema final cumple el requisito
- Una condición absolutamente necesaria para que un requisito sea verificable es que no sea ambiguo y que se defina de forma mensurable
- Los procedimientos de observación para comprobar que el sistema cumple los requisitos son la base para las pruebas de aceptación por parte del cliente (Wieringa, 1996)

## Modificable

- Su estructura y estilo de redacción permiten que los cambios se puedan realizar fácil, completa y consistentemente
- Debe tener una organización coherente
- No debe ser redundante
- Los requisitos deben expresarse individualmente y no de forma conjunta

# PROPIEDADES DE LAS ERS

## Trazable

- Para cada requisito contenido en ella se conoce su origen y puede referenciarse como origen en posteriores documentos
- Cada requisito puede seguirse hacia atrás y hacia delante

## Anotada con importancia y estabilidad

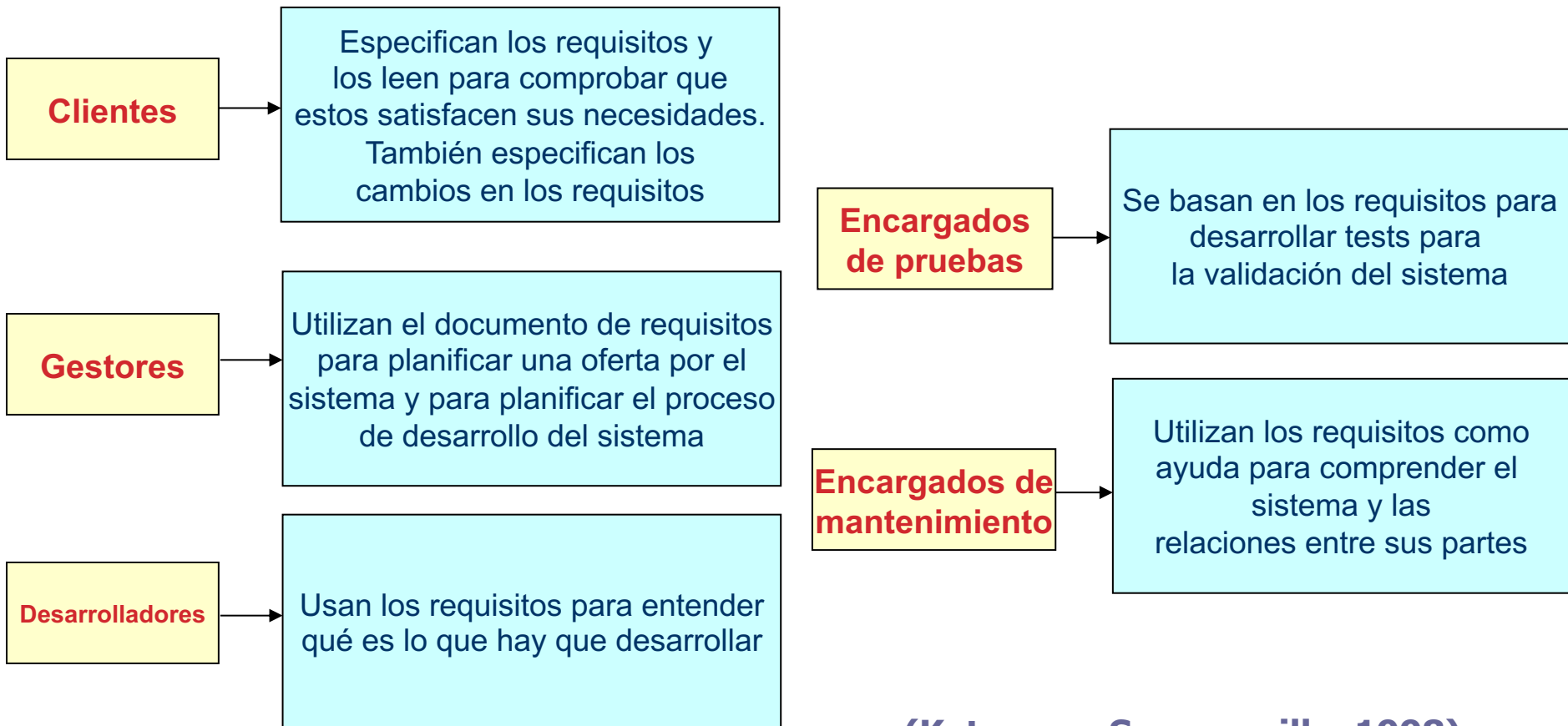
- Cada requisito contenido en ella está anotado con la importancia que tiene su cumplimiento para clientes y usuarios y la estabilidad que se espera del requisito, es decir, la probabilidad de que cambie durante el desarrollo

## Independiente del diseño y de la implementación

- No especifica una determinada descomposición del sistema (arquitectura) ni ningún aspecto de su posible implementación
- Solo deben admitirse requisitos que limiten la libertad de los diseñadores y programadores en el caso de que el cliente lo solicite explícitamente

# USUARIOS DE UNA ERS

El documento de requisitos tiene un conjunto de diversos usuarios que requieren un uso diferente del mismo



(Kotonya y Sommerville, 1998)



<https://unsplash.com/collections/173229/work-online?photo=B0n4-1PeF60Q>



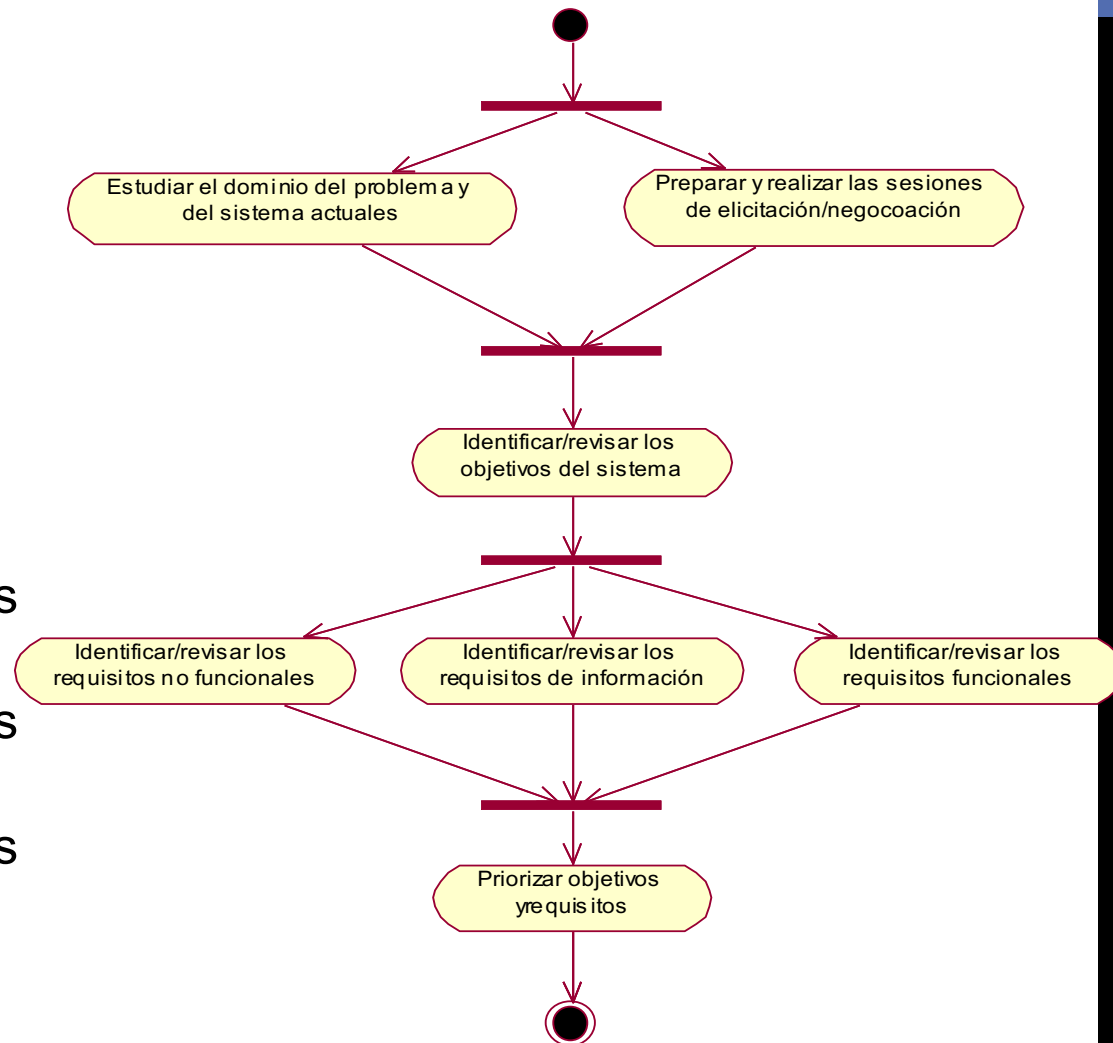
# 4. MDB: UNA METODOLOGÍA DE ELICITACIÓN DE REQUISITOS

# INTRODUCCIÓN

- MDB – Método de Durán y Bernárdez
- Desarrollada en el Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Sevilla
- Metodología para la obtención y documentación de los requisitos de sistemas de información
- Principales referencias (Durán, 2000; Durán y Bernárdez, 2002)
- Soportada por herramienta CASE
  - REM (*REquirements Manager*) (<https://goo.gl/Dtk59e>)

# TAREAS

- Obtener información sobre el dominio del problema y el sistema actual
- Preparar y realizar las reuniones de obtención/negociación
- Identificar/revisar los objetivos del sistema
- Identificar/revisar los requisitos de información
- Identificar/revisar los requisitos funcionales
- Identificar/revisar los requisitos no funcionales
- Priorizar objetivos y requisitos



(Durán y Bernárdez, 2002)

# TÉCNICAS

- Se recomiendan diversas técnicas para llevar a cabo las tareas anteriores
- Las más importantes son los casos de uso, las plantillas y los patrones lingüísticos
- Para la correcta descripción de los casos de uso (que son una forma de expresar requisitos funcionales), los requisitos no funcionales, los requisitos de información, así como de otros requisitos del sistema, se recurre a la utilización de diversos tipos de plantillas
  - Objetivos del sistema
  - Requisitos de información
  - Requisitos de restricción (reglas de negocio)
  - Actores
  - Casos de uso
  - Requisitos funcionales (expresados de forma tradicional, como texto libre)
  - Requisitos no funcionales
- Cada plantilla presenta los campos de información necesarios para especificar el concepto que está representando

# ESTRUCTURA DEL DOCUMENTO DE REQUISITOS DEL SISTEMA

Portada
Lista de cambios
Índice
Lista de figuras
Lista de tablas
1. Introducción
2. Participantes en el proyecto
3. Descripción del sistema actual
4. Objetivos del sistema
5. Catálogo de requisitos del sistema
5.1 Requisitos de información
5.2 Requisitos funcionales
5.2.1 Diagramas de casos de uso
5.2.2 Definición de actores
5.2.3 Casos de uso del sistema
5.3 Requisitos no funcionales
6. Matriz de rastreabilidad objetivos/requisitos
7. Glosario de términos
8. Conflictos pendientes de resolución <i>[opcional, pueden ir en un documento aparte]</i>
Apéndices <i>[opcionales]</i>

(Durán y Bernárdez, 2002)

# PLANTILLA PARA LOS OBJETIVOS DEL SISTEMA

Los objetivos del sistema pueden considerarse como **requisitos de alto nivel** (Sawyer y Kontoya, 2001), de forma que los requisitos propiamente dichos serían la forma de alcanzar los objetivos

OBJ-<id>	<nombre descriptivo>
Versión	<nº de la versión actual> (<fecha de la versión actual>)
Autores	<ul style="list-style-type: none"> <li>• &lt;autor de la versión actual&gt; (&lt;organización del autor&gt;)</li> <li>...</li> </ul>
Fuentes	<ul style="list-style-type: none"> <li>• &lt;fuente de la versión actual&gt; (&lt;organización de la fuente&gt;)</li> <li>...</li> </ul>
Descripción	El sistema deberá <objetivo a cumplir por el sistema>
Subobjetivos	<ul style="list-style-type: none"> <li>• OBJ-x &lt;nombre del subobjetivo&gt;</li> <li>• ...</li> </ul>
Importancia	<importancia del objetivo>
Urgencia	<urgencia del objetivo>
Estado	<estado del objetivo>
Estabilidad	<estabilidad del objetivo>
Comentarios	<comentarios adicionales sobre el objetivo>

(Durán y Bernárdez, 2002)

# PLANTILLAS PARA REQUISITOS DE INFORMACIÓN

IRQ- <i>&lt;id&gt;</i>	<i>&lt;nombre descriptivo&gt;</i>	
Versión	<i>&lt;nº de la versión actual&gt;</i> ( <i>&lt;fecha de la versión actual&gt;</i> )	
Autores	• <i>&lt;autor de la versión actual&gt;</i> ( <i>&lt;organización del autor&gt;</i> ) ...	
Fuentes	• <i>&lt;fuente de la versión actual&gt;</i> ( <i>&lt;organización de la fuente&gt;</i> ) ...	
Objetivos asociados	• OBJ-x <i>&lt;nombre del objetivo&gt;</i> ...	
Requisitos asociados	• Rx-y <i>&lt;nombre del requisito&gt;</i> ...	
Descripción	El sistema deberá almacenar la información correspondiente a <i>&lt;concepto relevante&gt;</i> . En concreto:	
Datos específicos	• <i>&lt;datos específicos sobre el concepto relevante&gt;</i> • ...	
Tiempo de vida	Medio	Máximo
	<i>&lt;tiempo medio de vida&gt;</i>	<i>&lt;tiempo máximo de vida&gt;</i>
Ocurrencias simult.	Medio	Máximo
	<i>&lt;nº medio de ocurr. simult.&gt;</i>	<i>&lt;nº máximo de ocurr. simult.&gt;</i>
Importancia	<i>&lt;importancia del requisito&gt;</i>	
Urgencia	<i>&lt;urgencia del requisito&gt;</i>	
Estado	<i>&lt;estado del requisito&gt;</i>	
Estabilidad	<i>&lt;estabilidad del requisito&gt;</i>	
Comentarios	<i>&lt;comentarios adicionales sobre el requisito&gt;</i>	

(Durán y Bernárdez, 2002)

# PLANTILLAS PARA REQUISITOS DE INFORMACIÓN

CRQ-<id>	<nombre descriptivo>
Versión	<nº de la versión actual> (<fecha de la versión actual>)
Autores	• <autor de la versión actual> (<organización del autor>) ...
Fuentes	• <fuente de la versión actual> (<organización de la fuente>) ...
Objetivos asociados	• OBJ-x <nombre del objetivo> ...
Requisitos asociados	• Rx-y <nombre del requisito> ...
Descripción	La información almacenada por el sistema deberá satisfacer la siguiente restricción: <restricción o regla de negocio>.
Importancia	<importancia del requisito>
Urgencia	<urgencia del requisito>
Estado	<estado del requisito>
Estabilidad	<estabilidad del requisito>
Comentarios	<comentarios adicionales sobre el requisito>

(Durán y Bernárdez, 2002)



# PLANTILLA PARA ACTORES

Aunque los actores de los casos de uso no son requisitos, por homogeneidad con el estilo de definición del resto de los elementos que componen el catálogo de requisitos se ha descrito la plantilla para definirlos

ACT-<id>	<nombre descriptivo>
Versión	<nº de la versión actual> (<fecha de la versión actual>)
Autores	✦ <autor de la versión actual> (<organización del autor>) ...
Fuentes	✦ <fuente de la versión actual> (<organización de la fuente>) ...
Descripción	Este actor representa a <rol que representa el actor>
Comentarios	<comentarios adicionales sobre el actor>

(Durán y Bernárdez, 2002)

# PLANTILLA PARA REQUISITOS FUNCIONALES

<b>CU-<i>id</i></b>	<b>&lt;nombre descriptivo&gt;</b>	
<b>Versión</b>	<b>&lt;nº de la versión actual&gt; (&lt;fecha de la versión actual&gt;)</b>	
<b>Autores</b>	<ul style="list-style-type: none"> <li>• <b>&lt;autor de la versión actual&gt; (&lt;organización del autor&gt;)</b></li> <li>...</li> </ul>	
<b>Fuentes</b>	<ul style="list-style-type: none"> <li>• <b>&lt;fuente de la versión actual&gt; (&lt;organización de la fuente&gt;)</b></li> <li>...</li> </ul>	
<b>Objetivos asociados</b>	<ul style="list-style-type: none"> <li>• <b>OBJ-x &lt;nombre del objetivo&gt;</b></li> <li>...</li> </ul>	
<b>Requisitos asociados</b>	<ul style="list-style-type: none"> <li>• <b>Rx-y &lt;nombre del requisito&gt;</b></li> <li>...</li> </ul>	
<b>Descripción</b>	El sistema debe comportarse tal como se describe en el siguiente caso de uso {abstracto durante la realización de los siguientes casos de uso: <lista de casos de uso>, cuando <evento de activación> [o durante la realización de los siguientes casos de uso: <lista de casos de uso>]}	
<b>Precondición</b>	<b>&lt;precondición del caso de uso&gt;</b>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	<i>p</i> <sub>1</sub>	{El actor <actor>, El sistema} <acción/es realizada/s por actor/sistema>
	<i>p</i> <sub>2</sub>	Se realiza el caso de uso <caso de uso (RF-x)>
	<i>p</i> <sub>3</sub>	Si <condición>, {el actor <actor>, el sistema} <acción/es realizada/s por actor/sistema>
	<i>p</i> <sub>4</sub>	Si <condición>, se realiza el caso de uso <caso de uso (RF-x)>
	...	...
<b>Poscondición</b>	<b>&lt;poscondición del caso de uso&gt;</b>	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	<i>p</i> <sub>i</sub>	Si <condición excepción>, {el actor <actor>, el sistema} <acción/es realizada/s por actor/sistema>, a continuación este caso de uso {continúa, queda sin efecto}
	<i>p</i> <sub>j</sub>	Si <condición excepción>, se realiza el caso de uso <caso de uso (RF-x)>, a continuación este caso de uso {continúa, queda sin efecto}
	...	...
<b>Rendimiento</b>	<b>Paso</b>	<b>Acción</b>
	<i>q</i>	<i>m</i> <unidad de tiempo>
	...	...
<b>Frecuencia</b>	<b>&lt;nº de veces&gt; veces / &lt;unidad de tiempo&gt;</b>	
<b>Importancia</b>	<b>&lt;importancia del requisito&gt;</b>	
<b>Urgencia</b>	<b>&lt;urgencia del requisito&gt;</b>	
<b>Estado</b>	<b>&lt;estado del requisito&gt;</b>	
<b>Estabilidad</b>	<b>&lt;estabilidad del requisito&gt;</b>	
<b>Comentarios</b>	<b>&lt;comentarios adicionales sobre el requisito&gt;</b>	

(Durán y Bernárdez, 2002)

# PLANTILLA PARA REQUISITOS NO FUNCIONALES

NFR-<id>	<nombre descriptivo>
Versión	<nº de la versión actual> (<fecha de la versión actual>)
Autores	• <autor de la versión actual> (<organización del autor>) ...
Fuentes	• <fuente de la versión actual> (<organización de la fuente>) ...
Objetivos asociados	• OBJ-x <nombre del objetivo> ...
Requisitos asociados	• Rx-y <nombre del requisito> ...
Descripción	El sistema deberá <capacidad del sistema>
Importancia	<importancia del requisito>
Urgencia	<urgencia del requisito>
Estado	<estado del requisito>
Estabilidad	<estabilidad del requisito>
Comentarios	<comentarios adicionales sobre el requisito>

(Durán y Bernárdez, 2002)

# CAUSAS DE LOS PROBLEMAS EN LA OBTENCIÓN DE LOS REQUISITOS

1. Análisis no adecuado de los *stakeholders*
2. Lenguaje no adecuado en la especificación de requisitos
  1. SMART-CC (Specific, Measureable, Attainable, Realizable and Time-bound - Complete and Concise - Específicos, medibles, alcanzables, realizables y limitados en el tiempo – Completos y Concisos)
3. Ir al diseño antes de haber obtenido los requisitos
4. No guiar la conversación durante la obtención de requisitos con un grupo de *stakeholders*
5. Conseguir la aprobación de los requisitos sin una comprensión compartida de los mismos
6. Creer que los requisitos deben estar completos al 100% antes de compartílos con los *stakeholders*
7. No construir una relación de confianza con los *stakeholders*
8. Utilizar a ciegas una plantilla

# TIPOS DE FALLOS EN LOS REQUISITOS

- Efecto *Big Bang*
- Suposiciones y expectativas no documentadas
- Desajustes con la realidad – Cuento de los 6 ciegos y el elefante
- Confundir el dominio del problema con el dominio de la solución
- Falta de cohesión de los requisitos
- Ignorar los requisitos no funcionales
- No introducir el aseguramiento de la calidad en los procesos de ingeniería de requisitos
- No tener en cuenta los escenarios de riesgo

# BIBLIOGRAFÍA

- A. Durán Toro, "Un Entorno Metodológico de Ingeniería de Requisitos para Sistemas de Información," PhD, Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla, 2000.
- A. Durán y B. Bernárdez, "Metodología para la elicitación de requisitos de sistemas software (versión 2.3)," Universidad de Sevilla, Universidad de Sevilla, España, Informe Técnico LSI-2000-10, 2002. Disponible en: <https://goo.gl/rhV8eV>.
- A. M. Davis, *Software Requirements: Objects, Functions and States*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice–Hall, 1993.
- C. Mazza, J. Fairclough, B. Melton, D. Pablo, A. Scheffer y R. Stevens, *Software Engineering Standards*. Hemel Hempstead: Prentice–Hall, 1994.
- F. J. García-Peñalvo, A. García-Holgado y A. Vázquez-Ingelmo, "Ingeniería de requisitos," Recursos docentes de la asignatura Ingeniería de Software I. Grado en Ingeniería Informática. Curso 2020-2021, F. J. García-Peñalvo, A. García-Holgado y A. Vázquez-Ingelmo, Eds., Salamanca, España: Grupo GRIAL, Universidad de Salamanca, 2021. [Online]. Disponible en: <https://bit.ly/392nEWi>. doi: 10.5281/zenodo.4420139. (pp. 4-55)
- F. P. Brooks, *The mythical man-month. Essays on software engineering. Anniversary Edition*. Boston, MA, USA: Addison Wesley, 1995.
- G. Kotonya y I. Sommerville, *Requirements Engineering: Processes and Techniques*. Chichester: John Wiley and Sons, 1998.
- H. D. Rombach, "Software Specifications: A Framework," Software Engineering Institute, Carnegie Mellon University, USA, Curriculum Module SEI–CM–11–2.1, 1990.
- I. Sommerville, *Software Engineering*, 6th ed. Boston, MA, USA: Addison-Wesley, 2001.
- IEEE. *IEEE Software Engineering Standards Collection 1999 Edition. Volume 1: Customer and Terminology Standards*. USA: IEEE Computer Society Press, 1999.
- IEEE. *IEEE Software Engineering Standards Collection 1999 Edition. Volume 4: Resource and Technique Standards*. USA: IEEE Computer Society Press, 1999.
- J. Goguen, "Requirements engineering as the reconciliation of social and technical issues," en *Requirements engineering: Social and technical issues*, M. Jirotko y J. Goguen, Eds. pp. 165-199, San Diego, CA, USA: Academic Press, 1994.

# BIBLIOGRAFÍA

- J. Johnson, "Chaos: The Dollar Drain of IT Project Failures," *Application Development Trends*, vol. 2, no. 1, pp. 41-47, 1995.
- J. W. Brackett, "Software Requirements," Software Engineering Institute. Carnegie Mellon University, Pittsburgh, PA , USA, SEI Curriculum Module SEI-CM-19-1.2, 1990.
- K. Pohl, "Requirements Engineering: An Overview," en *Encyclopedia of Computer Science and Technology*, vol. 36, A. Kent y J. Williams, Eds., New York, USA: Marcel Dekker, 1997.
- P. Loucopoulus y V. Karakostas, *System Requirements Engineering* (McGraw-Hill international series in software engineering). London, UK: McGraw-Hill, 1995.
- P. Sawyer y G. Kotonya, "Software Requirements," en *Guide to the Software Engineering Body of Knowledge SWEBOK*, A. Abran, J. W. Moore, P. Bourque, R. Dupuis y L. L. Tripp, Eds., USA: IEEE-CS Press, 2001.
- R. J. Wieringa, *Requirements Engineering: Frameworks for Understanding*. New York, NY, USA: John Wiley & Sons, 1996.
- R. S. Pressman, *Software Engineering: A Practitioner's Approach - European Adaptation*, 5th ed. London, England: McGraw-Hill, 2000.

# REQUISITOS

## INGENIERÍA DE SOFTWARE I

2º DE GRADO EN INGENIERÍA INFORMÁTICA  
CURSO 2020/2021

Francisco José García Peñalvo / fgarcia@usal.es

Alicia García Holgado / aliciagh@usal.es

Andrea Vázquez Ingelmo / andreavazquez@usal.es

Departamento de Informática y Automática

Universidad de Salamanca

