



# Tema 5.1: Especificación de Requisitos

## Ingeniería de Requisitos

Raquel Martínez España

Grado en Ingeniería Informática



# Índice

---

1. Especificación de requisitos
2. El documento de requisitos
3. Especificación formal



# Objetivos

---

- Entender las diferentes formas en qué puede describirse un requisito.
- Comprender cómo los requisitos se pueden organizar en un documento de requisitos del software.
- Conocer las principales características de un buen documento de requisitos.
- Conocer el documento de requisitos según el estándar IEEE 830.



# Índice

---

1. Especificación de requisitos
2. El documento de requisitos
3. Especificación formal



# Especificación de requisitos

---

- Se refiere a la producción de un documento
  - Que puede ser revisado, evaluado y aprobado.

Documentación de los requisitos esenciales (funciones, rendimiento, restricciones de diseño y atributos) del software y de sus interfaces externas

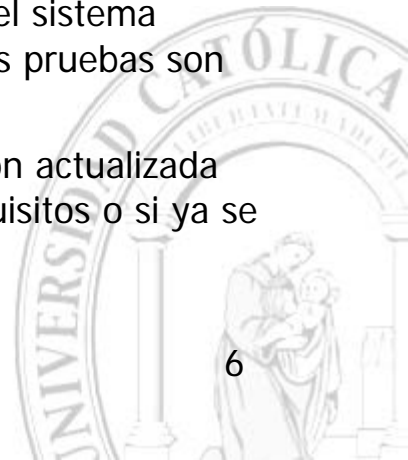
*[Glosario de términos estándar de Ingeniería del Software del IEEE (IEEE90)]*



# Especificación de requisitos

---

- Los **objetivos** de la especificación de requisitos software son diferentes según el punto de vista del usuario y desarrollador.
  - Para el **usuario** o cliente el objetivo es:
    - Definir la necesidad (lo que se quiere y lo que no se quiere).
  - Para el **desarrollador** tiene varios objetivos:
    - **Comunicación entre clientes, usuarios y desarrolladores.** Si está bien hecho el usuario estará satisfecho con el producto final. El acuerdo se realiza al principio, no en las pruebas de aceptación.
    - **Permitir iniciar la actividad de diseño.** Los requisitos determinan las características y funcionalidades que el futuro sistema software deberá poseer.
    - **Soportar las actividades de pruebas del sistema.** Si la descripción del requisito permite crear escenarios de prueba se puede mostrar que el sistema satisface los requisitos. Si es ambiguo o inconsistente o inestable las pruebas son imposibles.
    - **Controlar la evolución del sistema.** Manteniendo la especificación actualizada se puede controlar la evolución el sistema: si se añaden nuevos requisitos o si ya se han implementado los que estaban identificados.



# Especificación de requisitos

---

## Contenido de la especificación de requisitos

- Un documento de requisitos debe contener:
  - Información acerca del problema
  - Interfaz externa del sistema con su entorno (software, hardware, usuarios, puertos de comunicación)
  - Propiedades y comportamiento del sistema
  - Restricciones de diseño y fabricación del producto
  - Descripciones acerca de cómo el sistema ayudará a sus usuarios a realizar mejor sus tareas
  - Restricciones acerca de la tecnología que será utilizada en la construcción del sistema (protocolos, SSOO, etc.)
  - Restricciones acerca de las propiedades emergentes del sistema (requisitos no funcionales)
- Un documento de requisitos NO debe incluir:
  - Requisitos del proyecto: planificación, costes, fases, hitos,...
  - Diseño
  - Planes de garantía del producto

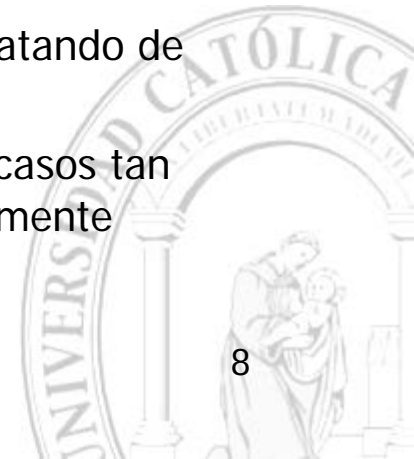


# Especificación de requisitos

---

## ¿Cómo escribir los requisitos?

- La “mejor forma” de escribir requisitos no existe.
- Lo más utilizado es el lenguaje natural utilizando expresiones del tipo: “el sistema hará X...”, “se facilitará Y...”, etc.
- En numerosas ocasiones el lenguaje natural se complementa con diagramas y notaciones formales.
  - El uso de estas notaciones formales depende de quien lee o escribe los requisitos.
  - Hay que tener en cuenta que el usuario o el cliente deben participar activamente en la elaboración de los requisitos, por tanto éstos deben estar escritos en un lenguaje que el usuario o cliente puedan entender.
- Normalmente se afronta la definición de los requisitos en positivo, tratando de recoger las necesidades que se deben satisfacer con el sistema.
  - Pero establecer lo que el sistema NO debe hacer es en algunos casos tan importante como establecer lo que debe hacer. Esto es especialmente importante para sistemas críticos.





# Especificación de requisitos

---

## Formato de las especificaciones

- Sólo texto
  - Si el texto es preciso, será más fácil de entender, pero si la especificación es grande y/o compleja, puede hacerse inmanejable
- Texto y técnicas (intuitivas, formales)
  - Combinación ideal para sistemas de tamaño complejidad media/grande
- Sólo técnicas
  - No recomendable, difícil de entender y rastrear



# Índice

---

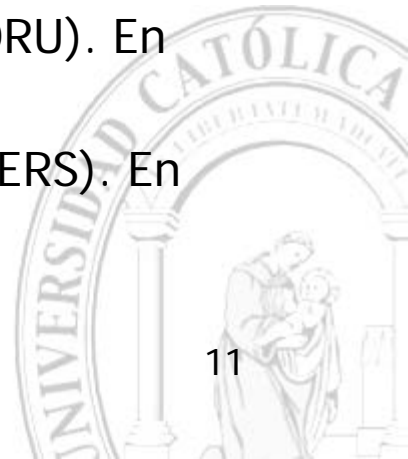
1. Especificación de requisitos
2. El documento de requisitos
3. Especificación formal



# El documento de requisitos

---

- El documento de requisitos es el modo habitual de guardar y comunicar requisitos.
- Por “Documento” se entiende cualquier medio electrónico de almacenamiento y distribución:
  - Procesador de textos
  - Base de Datos
  - Herramienta de Gestión de Requisitos
- Es buena práctica utilizar, al menos, dos documentos, a distinto nivel de detalle:
  - **Documento de Requisitos de Usuario** (en adelante DRU). En inglés, User Requirements Document, URD.
  - **Especificación de Requisitos Software** (en adelante ERS). En inglés, Software Requirements Specification, SRS.



# El documento de requisitos

---

## Documento de Requisitos de Usuario

Se trata de la fase de definición del problema y es un proceso iterativo donde el usuario es el máximo responsable.

- **Objetivo:** obtener una definición de lo que el usuario espera del sistema computacional.
- **Entradas:** resultados de entrevistas, encuestas, estudios, ejercicios de prototipado, etc.
- **Salidas:** documento de requerimientos de usuario (DRU), donde se describen todos los requerimientos con claridad y consistencia.



# El documento de requisitos

---

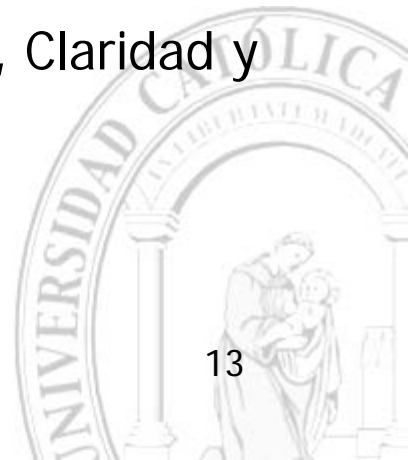
## Documento de Requisitos de Usuario

### **Actividades**

- Especificar los requisitos del usuario.
  - Describir los requisitos en términos de sus atributos.
  - Organizar los requisitos en base a la categoría a la que pertenezcan.
- Revisar y comprobar las salidas de la fase.

### **Atributos de los requisitos de usuario**

- Identificador, Necesidad, Prioridad, Estabilidad, Fuente, Claridad y Verificabilidad



# El documento de requisitos

---

## Documento de Requisitos Software

Se trata de la fase de análisis del problema, donde se debe adoptar un método de análisis y en la que el desarrollador es el máximo responsable.

- **Objetivo:** obtener una definición del software que se va a construir.
- **Entradas:** documento de requisitos del usuario (DRU).
- **Salidas:** documento de requisitos del software (DRS), donde se representa la visión que el desarrollador tiene del software y la descripción del ámbito del software.



# El documento de requisitos

---

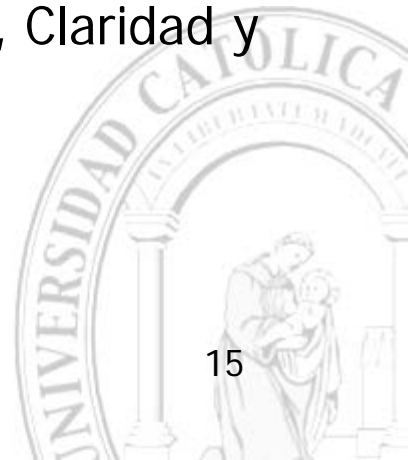
## Documento de Requisitos Software

### **Actividades**

- Especificar los requisitos del software.
  - Describir los requisitos en términos de sus atributos asegurando la completitud y consistencia, y evitando la duplicidad de los mismos.
  - Organizar los requisitos en base a la categoría a la que pertenezcan.
- Revisar y comprobar las salidas de la fase.

### **Atributos de los requisitos de software**

- Identificador, Necesidad, Prioridad, Estabilidad, Fuente, Claridad y Verificabilidad.

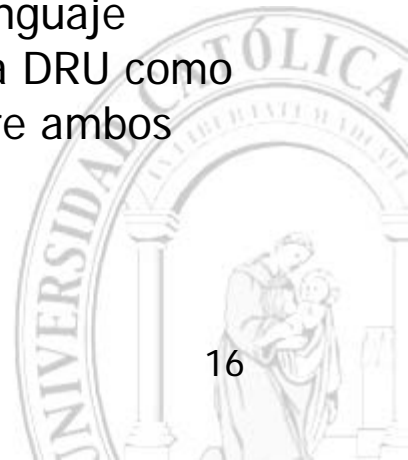


# El documento de requisitos

---

## ERS vs. DRU

- El DRU se escribe desde el **punto de vista** del usuario/cliente/interesado. Normalmente los requisitos de usuario, contenidos en la DRU, no poseen demasiado nivel de detalle. Se incluye la descripción del problema actual (razones por las que el sistema de trabajo actual es insatisfactorio) y las metas que se espera lograr con la construcción del nuevo sistema.
- La ERS desarrolla mucho más los contenidos de la DRU. Los requisitos del software contenidos en la ERS son, por tanto, **más detallados**. Contiene la respuesta a la pregunta ¿Qué características debe poseer un sistema que nos permita alcanzar los objetivos, y evitar los problemas, expuestos en la DRU?
- La diferencia entre la DRU y la ERS NO es que la DRU emplee lenguaje natural y la ERS emplee modelos o notaciones formales. Tanto la DRU como la ERS pueden utilizar todo tipo de notaciones. La diferencia entre ambos documentos es más de nivel de detalle.





# El documento de requisitos

---

## ERS vs. DRU (Ejemplo)

- Definición de requisitos
  1. El software debe acceder a ficheros externos creados por otras herramientas
- Especificación de requisitos
  - 1.1. El usuario debe poder elegir el tipo de fichero externo
  - 1.2. Cada tipo de fichero externo se puede asociar con la herramienta que produce ese tipo de fichero
  - 1.3. Cada tipo de fichero externo puede ser representado por un icono en la barra de herramientas
  - 1.4. Este icono puede ser definido por el usuario
  - 1.5. Cuando el usuario selecciona uno de estos iconos el efecto es aplicar la herramienta asociada a ese tipo de fichero



# El documento de requisitos

---

## Lectores/Usuarios de la ERS

- Clientes. Para comprobar que satisface sus necesidades y para hacer cambios a los requisitos.
- Gestores. Para preparar una oferta por el sistema y planificar el proceso de desarrollo.
- Ingenieros de Desarrollo. Para comprender el sistema a desarrollar.
- Ingenieros de Pruebas. Para elaborar pruebas de validación del sistema.
- Ingenieros de Mantenimiento. Para facilitar la comprensión del sistema y las relaciones entre sus partes.



# El documento de requisitos

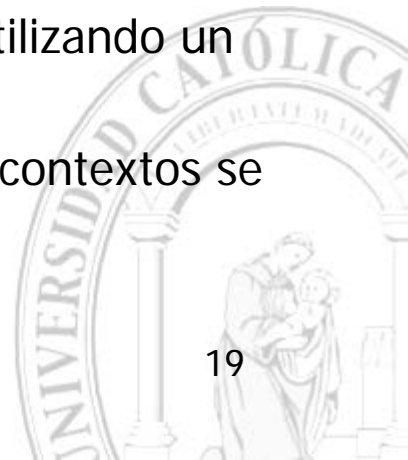
---

## Características de una buena ERS (IEEE 830):

- Calidad como objetivo. Difícil de cuantificar
- Una ERS perfecta es imposible:
  - Una ERS de calidad NO garantiza la ausencia de problemas.
  - Una ERS pésima garantiza su presencia.
- Una ERS de calidad debería poseer ¡24! características:

### 1. No ambigua

- Un requisito ambiguo se presta a distintas interpretaciones.
- Cada característica del producto final debe ser descrita utilizando un término único.
  - Si un término tiene distintos significados en distintos contextos se debe incluir un glosario



# El documento de requisitos

---

## Características de una buena ERS (IEEE 830):

### 2. Completa

- Si todo lo que se supone que el software debe hacer está incluido en la ERS.
- Se describen todas las posibles respuestas a todas las posibles entradas y en todas las situaciones posibles, para datos válidos y no válidos.
- No contendrá secciones de tipo TBD (en este caso quién, y cuándo debe completarlo)
- Las páginas deben estar numeradas; las figuras y tablas numeradas y referenciadas; las unidades de medida especificadas, el material de referencia descrito y debe contener todas las secciones del estándar utilizado.
- Es la característica más difícil de detectar.



# El documento de requisitos

---

## Características de una buena ERS (IEEE 830):

### 3. **Correcta**

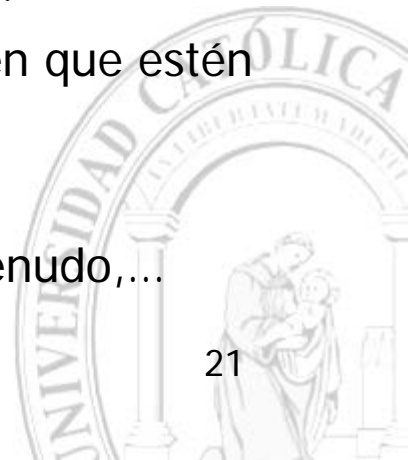
- Todo requisito de la ERS contribuye a satisfacer una necesidad real.

### 4. **Comprensible**

- Todo tipo de lectores (clientes, usuarios, desarrolladores, equipo de pruebas, gestores, etc.) entienden la ERS.

### 5. **Verificable**

- Si para cada requisito existe un procedimiento de prueba finito y no costoso para demostrar que el futuro sistema lo satisface.
- Los requisitos son verificables dependiendo de la forma en que estén escritos. Un requisito puede no ser verificable:
  - Si es ambiguo
  - Si usa medidas no cuantificables: normalmente, a menudo,...



# El documento de requisitos

---

## Características de una buena ERS (IEEE 830):

### 5. Verificable

- Algunos ejemplos para cuantificar requisitos no funcionales.

Propiedad	Medida
Rendimiento	Transacciones procesadas por segundo Tiempo de respuesta a un evento Tiempo de refresco de pantallaKbytes
Tamaño	Nº de chips de RAM
Fácil de uso	Tiempo de aprendizaje Nº de marcos de ayuda
Fiabilidad	Tiempo medio entre fallos Probabilidad de no estar disponible Tasa de ocurrencia de fallos Disponibilidad
Robustez	Tiempo de recuperación ante fallos Porcentaje de eventos causantes de fallo Probabilidad de corrupción de datos en un fallo
Portabilidad	Porcentaje de declaraciones dep. del objetivo Nº de sistemas objetivo

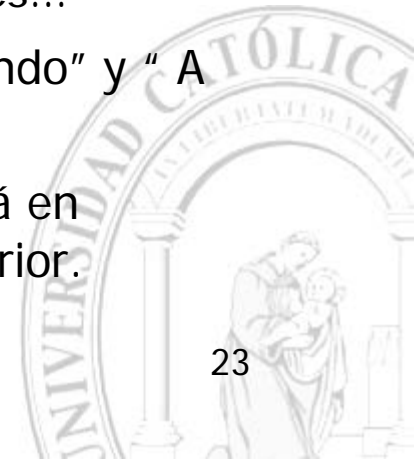


# El documento de requisitos

---

## Características de una buena ERS (IEEE 830):

6. **Consistente Internamente**: no existen subconjuntos de requisitos contradictorios.
  - Conflicto de comportamiento: "La luz se encenderá sólo si se presiona el botón" y "Cuando se suelta el botón la luz se enciende"
  - Conflicto de términos: si dos términos en contextos diferentes significan lo mismo.
  - Características en conflicto: "Todas las entradas se harán por selección de una opción en un menú desplegable" y "El lenguaje de comandos para el usuario consta de las siguientes ordenes..."
  - Inconsistencia temporal: "A ocurre cuando B está ocurriendo" y "A ocurre después de 10 segundos de que ocurra B".
7. **Consistente Externamente**: ninguno de los requisitos está en contradicción con lo expresado en documentos de nivel superior.



# El documento de requisitos

---

## Características de una buena ERS (IEEE 830):

### 8. Realizable

- Si, dados los actuales recursos, la ERS es implementable.

### 9. Concisa

- La ERS debe ser lo más breve posible, sin que esto afecte al resto de atributos de calidad.

### 10. Independiente del diseño

- Existe más de un diseño e implementación que realizan la ERS.

### 11. Trazable

- Cada requisito se puede referenciar de forma unívoca.

### 12. Modificable

- Los cambios son fáciles de introducir de forma completa y consistente. Implica la existencia de tabla de contenidos, índice y referencias cruzadas.





# El documento de requisitos

---

## Características de una buena ERS (IEEE 830):

**13. Electrónicamente almacenada**

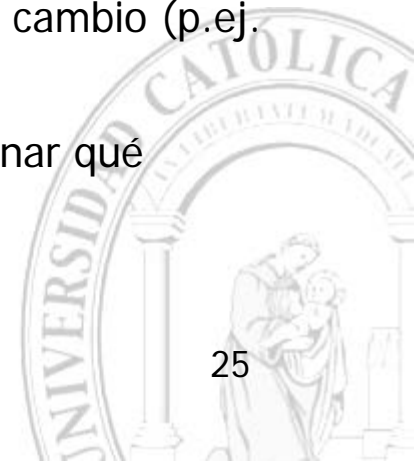
**14. Ejecutable/Interpretable/Prototipable/Animable**

- Si existe una herramienta software que, recibiendo como entrada la ERS, realice un modelo ejecutable de la misma (aplicable tan sólo a ciertas notaciones).

**15. Anotada por importancia relativa:** "Obligatorio, Deseable u Opcional".

**16. Anotada por estabilidad relativa:** una probabilidad de cambio (p.ej. "Alta, Media o Baja").

**17. Anotada por versión:** Si un lector de la ERS puede determinar qué requisitos serán satisfechos por qué versión del producto

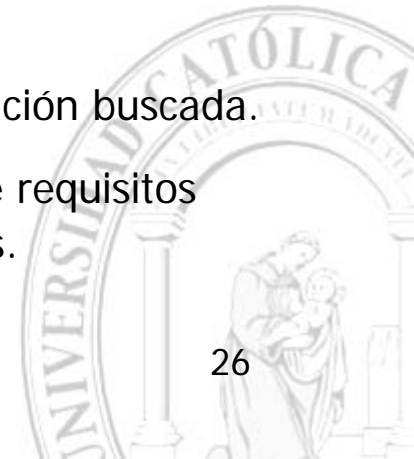


# El documento de requisitos

---

## Características de una buena ERS (IEEE 830):

- 18. **No redundante:** Cada requisito se expresa en un solo lugar de la ERS (no es del todo mala si aumenta la legibilidad).
- 19. **A un nivel de detalle adecuado**
- 20. **Precisa:** Una ERS es precisa si hace uso de valores numéricos para precisar las características del sistema. En la práctica es muy difícil.
- 21. **Reutilizable:** Si ciertas secciones de la ERS se pueden reutilizar
- 22. **Trazada:** Si está claro el origen de cada requisito (quién o qué lo pide y porqué).
- 23. **Organizada:** Si el lector puede fácilmente encontrar la información buscada.
- 24. **Con referencias cruzadas:** Si se utilizan referencias entre requisitos relacionados (trazabilidad intra-ERS) o entre secciones relacionadas.



# El documento de requisitos

---

Existen diversos estándares para documentar los requisitos. Por ejemplo:

- IEEE Std. 830
- IEEE Std. 1362 (ConOps)
- PSS-05 de la Agencia Espacial Europea (ESA)



# El documento de requisitos

---

- Software Requirements Specification (SRS/ERS)
  - Este documento es la base para el acuerdo entre clientes y desarrolladores o suministradores sobre lo que hará el producto.
  - Describir correctamente todos los requisitos del software. No describir ningún detalle del diseño del software
- **ANSI/IEEE Std 830 (1998)**
  - IEEE Recommended Practice for Software Requirements Specifications. Describe: contenido y cualidades de un buen documento SRS.
  - Propone diversas estructuras para el documento dependiendo de la forma en que se agrupan los requisitos.
  - Es revisión del 830-1993, y se ha adaptado al ISO/IEC 12207 (procesos del ciclo de vida del software)



# El documento de requisitos

---

## Estructura del ERS:

### 1. INTRODUCCIÓN

- 1.1. Propósito
- 1.2. Ámbito
- 1.3. Definiciones, acrónimos y abreviaturas
- 1.4. Referencias
- 1.5. Visión general del resto del documento

### 2. DESCRIPCIÓN GENERAL

- 2.1. Perspectiva del producto
- 2.2. Funciones del sistema
- 2.3. Características de los usuarios
- 2.4. Restricciones generales
- 2.5. Suposiciones y dependencias

### 3. REQUISITOS ESPECÍFICOS

- 3.1. Requisitos de interfaces externos
- 3.2. Requisitos funcionales
- 3.3. Requisitos de rendimiento
- 3.4. Requisitos de diseño
- 3.5. Atributos del sistema
- 3.6. Otros requisitos

### APÉNDICES

### ÍNDICE



# El documento de requisitos

---

- **Estructura del ERS:**

- La **sección 1** explica el contenido del documento.

## 1. Introducción:

### 1.1 Objetivo:

Propósito y Audiencia de la ERS

### 1.2 Ámbito

¿Qué hace y qué no hace el producto SW?

### 1.3 Definiciones, siglas, y abreviaturas

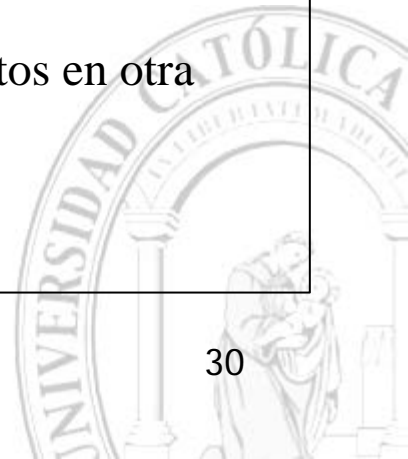
En forma de apéndices o referencias a otros documentos

### 1.4 Referencias

Lista completa de todas las referencias de los documentos en otra parte de la ERS

### 1.5 Visión Global

Cómo se organiza el resto de la ERS



# El documento de requisitos

---

- **Estructura del ERS:**

- La **sección 2**: define la relación del software con su entorno (2.1), da una visión general de las funciones del producto (2.2.), describe las características generales de los potenciales usuarios (2.3), indica limitaciones de hardware, interfaces, lenguaje de programación (2.4)

## 2. Descripción General

### 2.1 Perspectiva del Producto

Relación con otros Productos SW del Sistema: interfaces, usuario, HW, SW, comunicaciones ..

### 2.2 Funciones del Producto

¿Qué hace y qué no hace el producto SW?

### 2.3 Características del Usuario

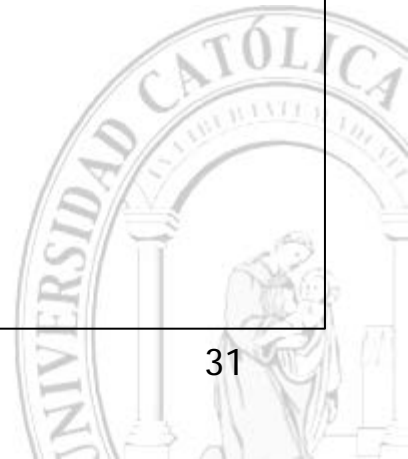
Bagaje, formación académica, experiencia, especialización técnica

### 2.4 Restricciones Generales

Regulaciones, Limitaciones HW, Interfaces, ...

### 2.5 Asunciones y Dependencias

Factores que pueden afectar a los requisitos especificados



# El documento de requisitos

---

- **Estructura del ERS:**

- La **sección 3** se puede organizar de diversas formas ya que diferentes tipos de sistemas requieren diferente énfasis en sus requisitos.
- Contiene todos los requisitos software.
- Para cada requisito, se debe incluir:
  - Identificador único
  - Descripción de cada entrada (el estímulo) en el sistema,
  - Cada salida (la contestación) del sistema, y
  - Proceso
  - Error





# El documento de requisitos

---

- **Estructura del ERS:**

- Ejemplo

Requisito: 1.1 Dar de Alta Tarjeta	
Entrada	Los datos del cliente (nombre, apellidos, dirección, teléfono y e-mail)
Proceso	Generar un nuevo identificador de tarjeta. Añadir en el almacén TARJETAS una entrada con este identificador, la fecha de alta (fecha actual), los puntos iniciales de la tarjeta (0) y los datos del cliente.
Salida	Una entrada nueva en el almacén TARJETAS y la tarjeta física que se entrega al cliente.
Error	El usuario deberá introducir obligatoriamente el nombre y los apellidos del cliente. De no hacerlo, el sistema emitirá el mensaje de error "Imposible dar alta tarjeta: no se han proporcionado el nombre y los apellidos del cliente". Asimismo, para poder ponerse en contacto con el cliente, será obligatorio introducir la dirección y el teléfono del cliente. De no hacerse, el sistema mostrará el mensaje de error "Imposible dar alta tarjeta: no se han proporcionado la dirección y el teléfono del cliente".

# El documento de requisitos

---

- **Estructura del ERS:**

- Otra forma de definir cada requisito:
  - Validación de las entradas
  - Secuencia de las operaciones
  - Mensajes de error y control de excepciones
  - Efecto de los parámetros
  - Secuencia de las salidas
  - Fórmulas para la conversión de las entradas en salidas



# El documento de requisitos

---

- **Estructura del ERS:**

- La **sección 3**:

## **3. Requisitos Específicos**

### **3.1 Interfaces Externas**

Descripción detallada de las entradas y salidas del Sistema SW

Complementa las descripciones de Interfaz de los apartados anteriores

### **3.2 Funciones (requisitos funcionales)**

### **3.3 Requisitos de Rendimiento/Ejecución**

Estáticos y Dinámicos

### **3.4 Restricciones de Diseño**

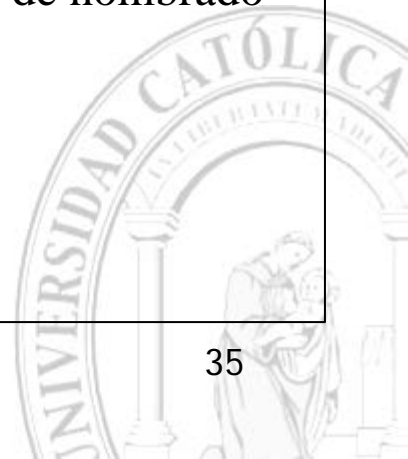
Impuestas por otros estándares (formato informes; convenciones de nombrado elementos; etc..)

Limitaciones del HW

### **3.5 Atributos de Calidad del Software**

Fiabilidad, Disponibilidad, Mantenibilidad, Seguridad,

### **3.6 Otros Requisitos**



# El documento de requisitos

---

Alternativas en la especificación de requisitos específicos, según IEEE 830-1998:

1. Modos de operación
2. Clases de usuarios
3. Objetos (en sentido OO)
4. Características, o servicios (deseados externamente, que pueden requerir una secuencia de inputs para obtener el resultado deseado)
5. Estímulo/Respuesta
6. Jerarquía funcional: por cada elemento del DFD (flujos, procesos, entidades, ficheros) cada sección describe un DFD o descripción funcional equivalente (ligado a AE)
7. Múltiples organizaciones: combinando dos o más de las anteriores según necesidades de la organización



# El documento de requisitos

- **Por clases de usuario:**

*Plantilla de SRS, sec. 3 organizada por clases de usuario (A.3)*

<p>3. Requisitos específicos</p> <p>3.1. Requisitos de interfaz externa</p> <p>3.1.1. Interfaces de usuario</p> <p>3.1.2. Interfaces hardware</p> <p>3.1.3. Interfaces software</p> <p>3.1.4. Int. de comunicaciones</p> <p>3.2. Requisitos funcionales</p> <p>3.2.1 Clase de usuario 1</p> <p>3.2.1.1. Requisito funcional 1.1</p> <p>.....</p> <p>3.2.1.n Requisito funcional 1.n</p> <p>.....</p> <p>3.2.2 Clase de usuario m</p> <p>3.2.m.1. Requisito funcional m.1</p> <p>.....</p> <p>3.2.m.n Requisito funcional m.n</p>	<p>3.3. Requisitos de rendimiento</p> <p>3.4. Restricciones de diseño</p> <p>3.4.1. Acatamiento de estándares</p> <p>3.4.2. Limitaciones hardware</p> <p>.....</p> <p>3.5. Atributos de sistema software</p> <p>3.5.1. Seguridad</p> <p>3.5.2. Mantenimiento</p> <p>.....</p> <p>3.6. Otros requisitos</p> <p>3.6.1. Base de datos</p> <p>3.6.2. Operaciones</p> <p>3.6.3. Adaptación de situación</p> <p>.....</p>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

# El documento de requisitos

- **Por estímulos:**

*Plantilla de SRS, sec. 3 organizada por estímulos (A.6)*

3. Requisitos específicos

3.1. Requisitos de interfaz externa

3.1.1. Interfaces de usuario

3.1.2. Interfaces hardware

3.1.3. Interfaces software

3.1.4. Int. de comunicaciones

3.2. Requisitos funcionales

3.2.1 Estímulo 1

3.2.1.1. Requisito funcional 1.1

.....

3.2.1.n Requisito funcional 1.n

.....

3.2.2 Estímulo m

3.2.m.1. Requisito funcional m.1

.....

3.2.m.n Requisito funcional m.n

3.3. Requisitos de rendimiento

3.4. Restricciones de diseño

3.4.1. Acatamiento de estándares

3.4.2. Limitaciones hardware

.....

3.5. Atributos de sistema software

3.5.1. Seguridad

3.5.2. Mantenimiento

.....

3.6. Otros requisitos

3.6.1. Base de datos

3.6.2. Operaciones

3.6.3. Adaptación de situación

.....

# El documento de requisitos

---

- Metainformación asociada a un requisito. Plantillas de requisitos
  - La Norma IEEE1233 recomienda los siguientes atributos para cada requisito:
    - Identificación (única)
    - Prioridad
    - “Criticidad”
    - Viabilidad
    - Riesgo
    - Fuente
    - Relaciones de traza con otros requisitos

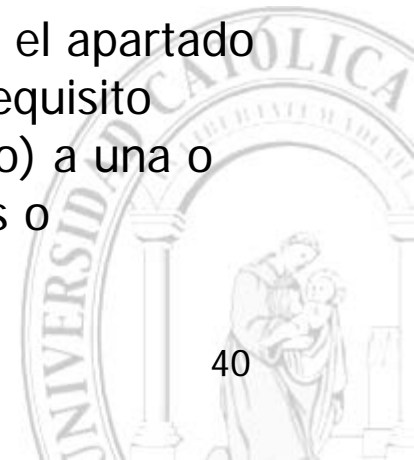


# El documento de requisitos

---

Algunos problemas prácticos con el documento de requisitos:

- Manejo de **gran cantidad** de requisitos. Solución: CARE.
- No consideración de **requisitos “óbvios”**. Solución: No se trata de especificar todo. Utilizar el sentido común
- **Nivel de detalle** en que debe expresarse un requisito (relacionado con el punto anterior, evitando detallar cuestiones que sean obvias). Dependerá del nivel de abstracción en que nos encontremos (requisitos de empresa, de usuario, de software). Solución: trazabilidad entre requisitos para ligar niveles de abstracción
- Determinación de la **correcta ubicación** de un requisito en el apartado que mejor corresponda del documento. A veces un mismo requisito pertenece a varias categorías y puede por tanto ser asignado) a una o más secciones del documento Solución: referencias cruzadas o tratamiento basado en aspectos software).

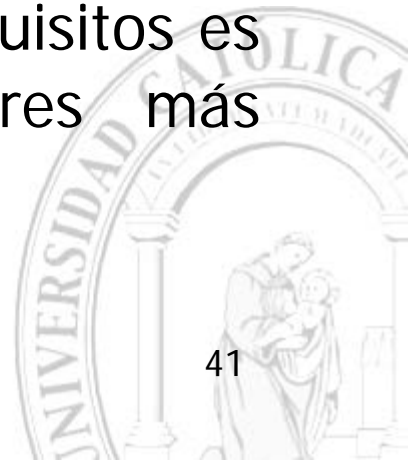




# Puntos clave

---

- La especificación de requisitos puede realizarse mediante texto, técnicas o una combinación de ambas.
- Los documentos de requisitos de software son la declaración acordada de los requisitos del sistema.
- Se deben organizar de tal forma que puedan ser utilizados tanto por los clientes del sistema como por los desarrolladores del software.
- El estándar IEEE para los documentos de requisitos es un punto de partida útil para estándares más específicos de especificación de requisitos.



# Índice

---

1. Especificación de requisitos
2. El documento de requisitos
3. Especificación formal

