



Tema 2.1: Conceptos básicos de la Ingeniería de Requisitos

Ingeniería de Requisitos

Raquel Martínez España

Grado en Ingeniería Informática



Índice

1. Concepto de requisito
2. Tipos de requisitos



Objetivos

- Presentar el concepto de requisito de un sistema software y explicar las diferentes formas de expresar requisitos.
- Establecer los conceptos de requisitos del usuario y del sistema y el por qué se deben escribir de diferentes formas.
- Diferenciar entre requisitos funcionales y no funcionales.



Índice

1. Concepto de requisito

1. Definiciones
2. Objetivos de los requisitos
3. Características de los requisitos
4. Problemas comunes

2. Tipos de requisitos



1.1. Definiciones

- “Una condición o capacidad que debe ser cumplida, o poseída, por un sistema o componente de sistema, para satisfacer un contrato, estándar, especificación u otros documentos impuestos formalmente”.
- “Una representación documentada de una condición o capacidad relativa a los puntos anteriores.”

[Glosario de términos del IEEE]



1.1. Definiciones

“Una especificación de qué se debería implementar. Son descripciones de cómo se debe comportar el sistema, o de un atributo o propiedad del sistema. Puede ser una restricción en el proceso de desarrollo de un sistema.”

[Somerville y Sawler, (1997)]



1.1. Definiciones

- **Ingeniería de Requisitos:** comprende las actividades de desarrollo de software (y SI) relacionadas con la gestión y definición de requisitos para sistemas nuevos o actuales
- **Especificación de Requisitos Software (ERS, SRS):** Documento formal de los Requisitos del Sistema



1.1. Definiciones

- **NO** es un requisito:
 - Detalles de diseño o implementación o pruebas.
 - Información relativa a la planificación del proyecto.
 - Necesidades del proyecto.
- Estos elementos se suelen llamar “Especificaciones suplementarias”



1.1. Definiciones

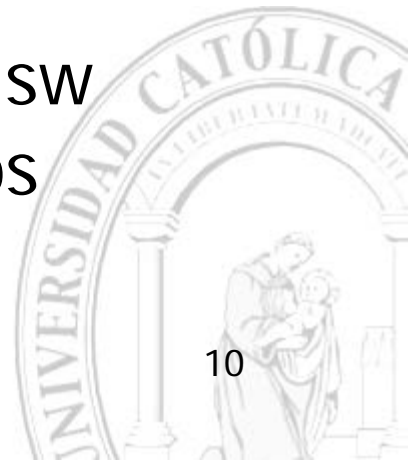
- Ejemplo: aplicación de contabilidad
 - R1) *"El sistema debe permitir al usuario ver su saldo"*
 - R2) *"El estado de la cuenta del cliente se almacenará en una tabla llamada saldoCliente en una base de datos MySQL"*

¿Son R1 y R2 requisitos de la aplicación contabilidad?



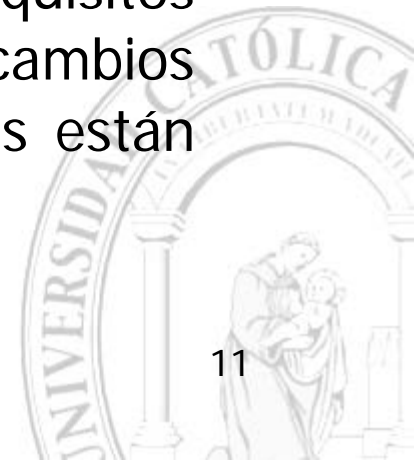
1.2. Objetivos de los requisitos

- Alcanzar un acuerdo entre clientes, productores de sw y usuarios, sobre lo que hay que producir.
- Proporcionar la base para el diseño de software.
- Servir como soporte para la verificación y validación de los productos obtenidos
- Orientar a potenciales compradores de sw (clientes) sobre la definición de nuestros productos



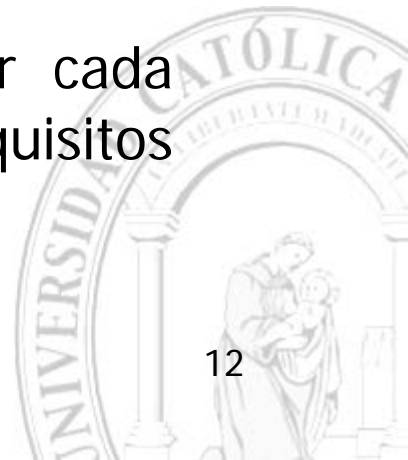
1.3. Características de los requisitos

- Según el estándar IEEE-830:
 - Deben ser **correctos**: Tanto el cliente como el desarrollador deben revisarlos para asegurar que no tienen errores.
 - Deben ser **consistentes**: Dos requisitos son inconsistentes cuando es imposible satisfacerlos simultáneamente.
 - Deben estar **completos**: El conjunto de requisitos está completo si todos los estados posibles, cambios de estado, entradas, productos y restricciones están descritos en alguno de los requisitos.



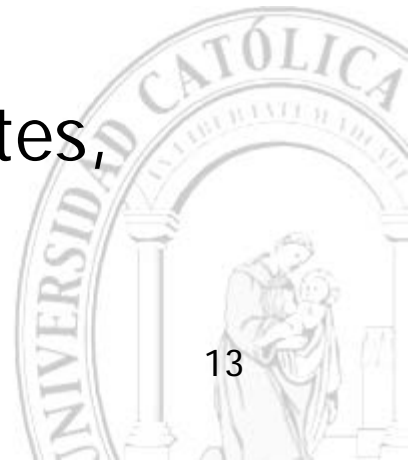
1.3. Características de los requisitos

- Deben ser **realistas**: Todos los requisitos deben ser revisados para asegurar que son posibles.
- ¿Cada requisito describe algo que es **necesario** para el cliente?: Los requisitos deben ser revisados para conservar sólo aquellos que inciden directamente en la resolución del problema del cliente.
- Deben ser **verificables**: Se deben poder preparar pruebas que demuestren que se han cumplido los requisitos.
- Deben ser **rastreables**: ¿Se puede rastrear cada función del sistema hasta el conjunto de requisitos que la establece?



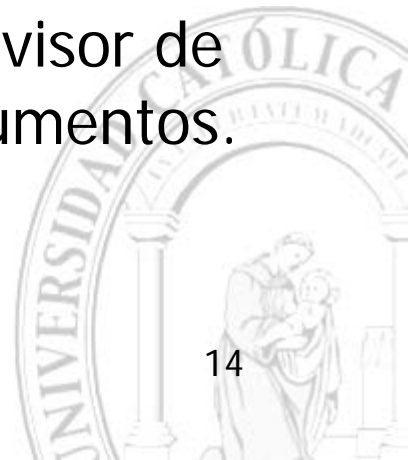
1.4. Problemas comunes

- No reflejan las necesidades reales del cliente
- Son inconsistentes y/o incompletos
- Es costoso realizar cambios sobre los requisitos una vez que han sido acordados
- Puede haber malentendidos entre clientes, analistas, ingenieros software, ..



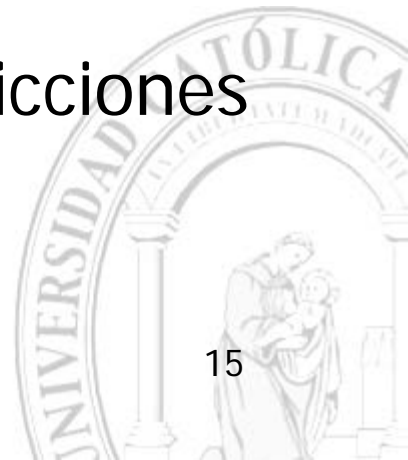
1.4. Problemas comunes

- **Imprecisión:** los requisitos ambiguos pueden ser interpretados de diferentes formas por desarrolladores y usuarios.
- Ejemplo:
 - *El software incluirá visores adecuados*
 - Intención del usuario: deberá incluir un visor especial para cada tipo de documento manejado.
 - Interpretación del desarrollador: incluir un visor de texto que muestre el contenido de los documentos.



1.4. Problemas comunes

- **Compleitud:**
 - Deben incluir descripciones de todos los servicios requeridos.
 - En la práctica este objetivo es imposible de cumplir al 100% en un documento de requisitos de complejidad media o grande.
- **Consistencia:**
 - No deben existir conflictos o contradicciones en las descripciones.



1.4. Problemas comunes

- Frontera entre Requisitos y Diseño.
 - “Los requisitos indican lo que el sistema debe hacer y el diseño describe cómo lo hace.”
 - Pero...
 - La arquitectura del sistema puede ser diseñada para estructurar los requisitos.
 - El sistema interacciona con otros sistemas que generan requisitos de diseño del software.
 - El uso de un diseño específico puede ser un requisito de dominio.



Índice

1. Concepto de requisito

2. Tipos de requisitos

1. Requisitos funcionales y no funcionales. Dominio.
2. Requisitos de usuario y sistema. Software.
3. Requisitos de cliente y requisitos de desarrollador.



2. Tipos de requisitos

- Existen muchas taxonomías.
- Según su naturaleza:
 - Funcionales y no funcionales. Dominio.
- Según el nivel de abstracción/detalle:
 - Usuario y sistema. Software.



Índice

1. Concepto de requisito

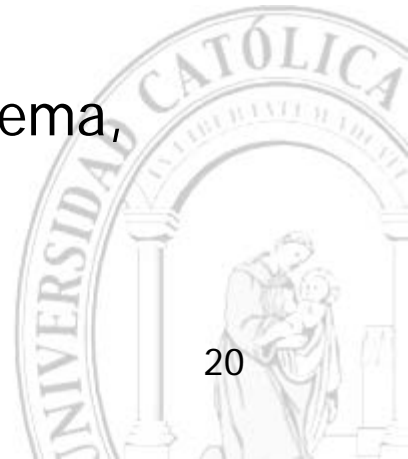
2. Tipos de requisitos

1. Requisitos funcionales y no funcionales. Dominio.
2. Requisitos de usuario y sistema. Software.
3. Requisitos de cliente y requisitos de desarrollador.



2.1. Requisitos funcionales y no funcionales. Dominio

- Este es el criterio de clasificación más habitual
 - **Funcionales**: [capacidades]
 - Describen las funciones que lleva a cabo el software, cómo debe reaccionar ante ciertas entradas y cómo debe comportarse en situaciones particulares.
 - **No Funcionales**: [restricciones, requisitos de calidad]
 - Restricciones sobre las funciones o servicios ofrecidos por el sistema.
 - **De Dominio**:
 - Proviene del dominio de aplicación del sistema, reflejando sus características.



2.1. Requisitos funcionales y no funcionales. Dominio

- **Requisitos funcionales**
 - Definen qué hace el sistema
 - Describen las funciones que lleva a cabo el software, cómo debe reaccionar ante ciertas entradas y cómo debe comportarse en situaciones particulares
 - Qué no debe hacer (opcional)
 - Dependen de:
 - El tipo de software
 - Las expectativas de los usuarios
 - EL tipo de sistema donde el software se usará.



2.1. Requisitos funcionales y no funcionales. Dominio

- **Requisitos funcionales**
 - Ejemplos de Requisitos Funcionales
 - R1: El usuario debe ser capaz de buscar entre todo el conjunto de bases de datos o de seleccionar un subconjunto de ellas.
 - R2: Cada pedido tiene un identificador único (pedido_id), que el usuario no podrá copiar al área de almacenamiento de la cuenta.
 - Diferentes niveles de abstracción.



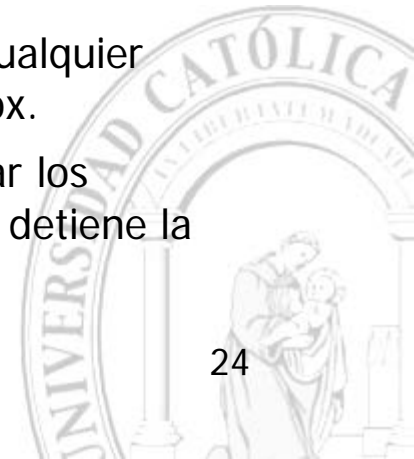
2.1. Requisitos funcionales y no funcionales. Dominio

- **Requisitos funcionales**
 - Ejemplos de Requisitos Funcionales
 - R1: La aplicación debe calcular el valor de la cartera de inversiones del usuario.
 - R2: La aplicación debe terminar el cálculo del valor de cada cartera en menos de un segundo.
 - ¿Es R2 un requisito funcional?



2.1. Requisitos funcionales y no funcionales. Dominio

- **Requisitos no funcionales**
 - Definen cómo debe funcionar el sistema
 - Establecen restricciones sobre el producto desarrollado, el proceso de desarrollo o externas
- No están generalmente relacionados con la funcionalidad del sistema
- Ejemplos:
 - R1: El sistema no debe revelar al personal que lo utilice ninguna información personal sobre los usuarios aparte de su nombre y DNI.
 - R2: El sistema debe visualizarse y funcionar correctamente en cualquier navegador, especialmente en Internet Explorer, Chrome y Firefox.
 - R3: El sistema no debe tardar más de cinco segundos en mostrar los resultados de una búsqueda. Si se supera este plazo, el sistema detiene la búsqueda y muestra los resultados encontrados.



2.1. Requisitos funcionales y no funcionales. Dominio

- **No hay una distinción clara** entre requisitos funcionales y no funcionales. Depende del nivel de detalle requerido en el documento de requisitos
 - Ejemplo:
 - R1: “El sistema asegurará que los datos son protegidos de accesos no autorizados”
 - R2: “El sistema incluirá un procedimiento de autorización de usuario en el que los usuarios se identifican mediante un nombre de usuario y una contraseña. Sólo los usuarios autorizados pueden acceder a los datos del sistema”
 - *¿R1 y R2 son funcionales o no funcionales?*



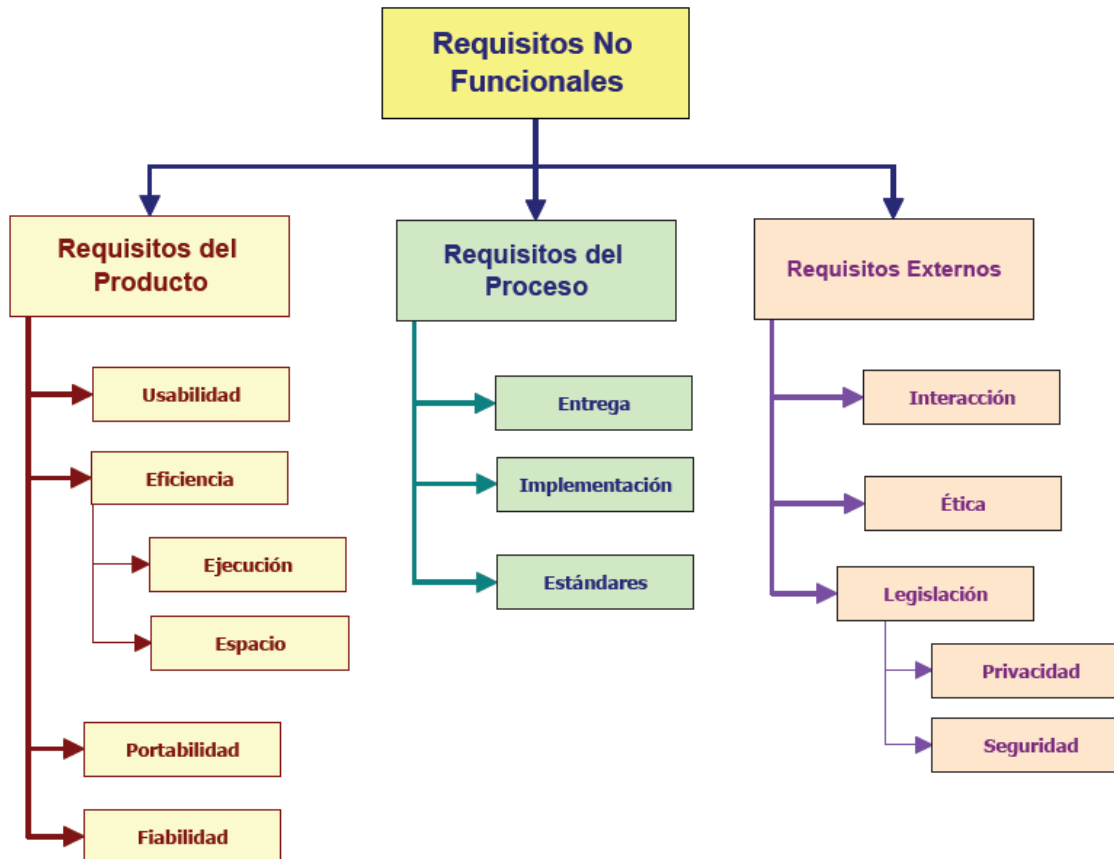
2.1. Requisitos funcionales y no funcionales. Dominio

- Tipos de **Requisitos No Funcionales** (IEEE 830-1998)
 - Interfaces Externas
 - De Rendimiento
 - Lógicos de Base de Datos
 - Restricciones de Diseño
 - Cumplimiento de Estándares
 - De Atributos del Software:
 - De Fiabilidad
 - De Disponibilidad
 - De Seguridad
 - De Mantenibilidad
 - De Portabilidad



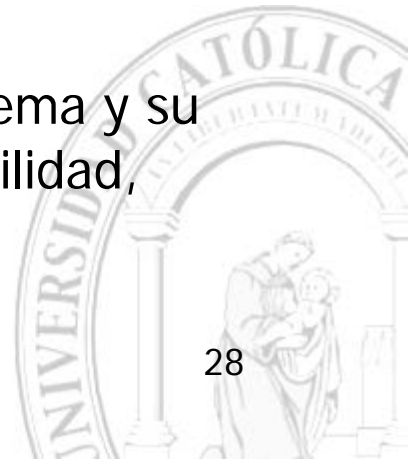
2.1. Requisitos funcionales y no funcionales. Dominio

- Tipos de **Requisitos No Funcionales** (Sommerville, 2005)



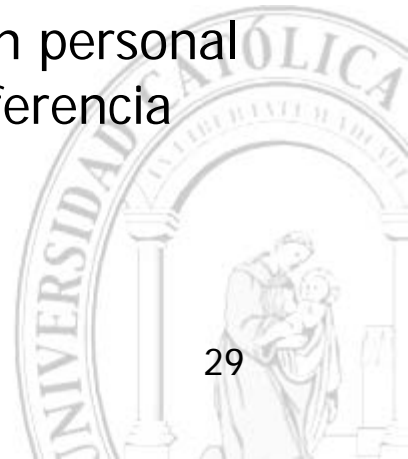
2.1. Requisitos funcionales y no funcionales. Dominio

- Tipos de Requisitos No Funcionales (Sommerville)
 - De Producto
 - Especifican comportamiento del producto (velocidad de ejecución, fiabilidad, etc.)
 - Organizacionales
 - Son una consecuencia de las políticas y procedimientos de la organización (estándares usados, requisitos de implementación, etc.).
 - Externos
 - Están relacionados con factores externos al sistema y su proceso de desarrollo (requisitos de interoperabilidad, normas legales, etc.).



2.1. Requisitos funcionales y no funcionales. Dominio

- Ejemplos de Requisitos No Funcionales
 - De Producto
 - R1: El interfaz de usuario será implementado como HTML simple sin frames ni applets Java.
 - Organizacionales
 - R2: El proceso de desarrollo y los entregables serán conformes a lo establecido en METRICA v3.
 - Externos
 - R3: El sistema no registrará ninguna información personal de los clientes salvo su nombre y número de referencia respetando lo establecido en la LOPD.



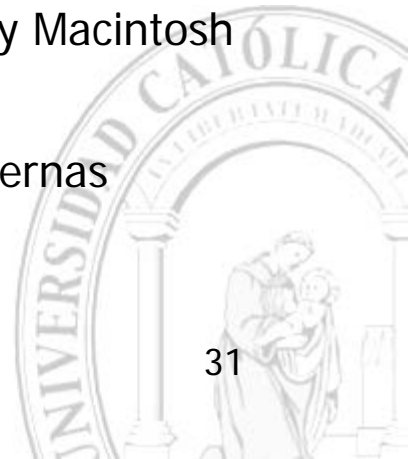
2.1. Requisitos funcionales y no funcionales. Dominio

- Requisitos No Funcionales de **Producto**
 - Especifican Restricciones en la Ejecución del Sistema
 - Parte de estos requisitos se pueden formular de forma precisa, de forma que puedan ser fácilmente cuantificables:
 - Rendimiento
 - Capacidad
 - Otros son más difíciles de cuantificar y por lo tanto se expresan generalmente de modo informal
 - Usabilidad



2.1. Requisitos funcionales y no funcionales. Dominio

- Ejemplos de Requisitos No Funcionales de **Producto**
 - Fiabilidad:
 - RF1: El servicio A del Sistema debe tener una disponibilidad de 99 %
 - Rendimiento:
 - RR1: El Sistema Y procesará un mínimo de 8 transacciones por segundo
 - Espacio:
 - RE1: El código ejecutable del Sistema Z estará limitado a 512 Kb
 - Portabilidad:
 - RP1: El Sistema se desarrollará para las plataformas PC y Macintosh
 - Seguridad:
 - RS1: El Sistema encriptará todas las comunicaciones externas usando El algoritmo RSA



2.1. Requisitos funcionales y no funcionales. Dominio

- Requisitos No Funcionales de **Producto**
 - **Conflictos**
 - Un requisito de rendimiento puede estar en conflicto con requisitos de seguridad o fiabilidad
 - Un requisito de utilización de espacio puede estar en contradicción con un requisito que exige el uso de un compilador que no genera un código muy compacto
 - Los conflictos se resuelven en función de:
 - El nivel de importancia de cada requisito
 - Consecuencias sobre el cambio en el requisito
 - Los objetivos de negocio



2.1. Requisitos funcionales y no funcionales. Dominio

- Requisitos No Funcionales de **Proceso**
 - Son **restricciones** en el **proceso** de desarrollo del Sistema
 - Estándares y Métodos de Desarrollo
 - Herramientas a usar
 - Producción de Informes



2.1. Requisitos funcionales y no funcionales. Dominio

- Ejemplos de Requisitos No Funcionales de **Proceso**
 - R1: El proceso de desarrollo debe ser conforme con ISO 9003
 - R2: El Sistema debe desarrollarse usando la Herramienta Visual Paradigm
 - R3: Los informes de Gestión sobre el esfuerzo dedicado a cada componente se deben generar cada dos semanas



2.1. Requisitos funcionales y no funcionales. Dominio

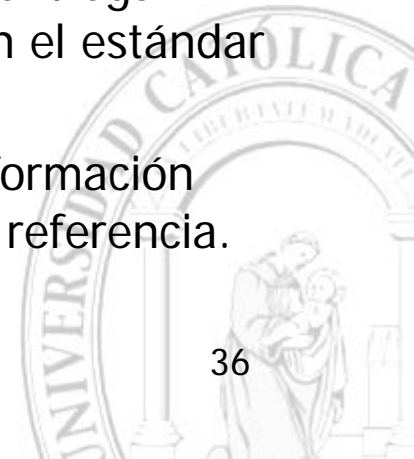
- Requisitos No Funcionales **Externos**
 - Están relacionados con el **Entorno**
 - Se pueden aplicar a Producto y Proceso
- Ejemplos:
 - R1: El responsable de la protección de datos de la organización debe certificar que todos los datos se mantienen de acuerdo a la legislación vigente



2.1. Requisitos funcionales y no funcionales. Dominio

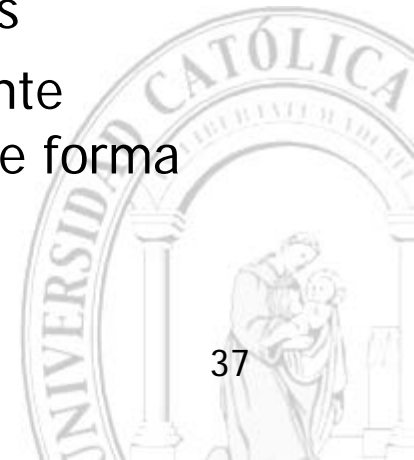
Algunos ejemplos:

1. El sistema debe permitir la impresión de informes tanto en la impresora A (láser) como en la B (matriz de punto).
2. El sistema debe ser operable con diferentes sistemas operativos (por ejemplo, Windows y Linux).
3. El sistema debe ser operable en diversos navegadores (por ej., Explorer y Firefox)
4. El formato del documento de factura debe ser compatible con el dispuesto por la DGI.
5. El proceso de desarrollo del sistema y los documentos a entregar deberán apegarse al proceso y los productos definidos en el estándar X.
6. El sistema no deberá revelar a sus operadores alguna información personal de los clientes excepto su nombre y número de referencia.



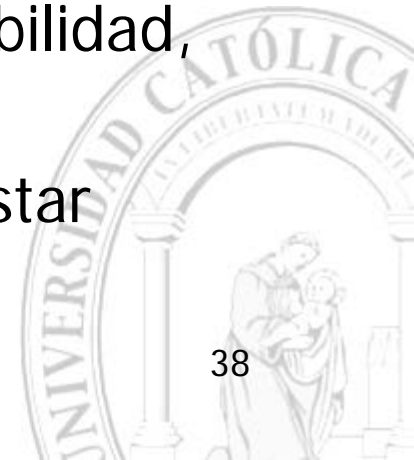
2.1. Requisitos funcionales y no funcionales. Dominio

- Los **Requisitos No Funcionales** suelen ser difíciles de expresar:
 - Hay restricciones relacionadas con soluciones de diseño que son desconocidas en la etapa de requisitos
 - Hay restricciones altamente subjetivas que sólo se pueden determinar mediante evaluaciones empíricas complejas
 - Tienden a estar relacionados con uno o más requisitos funcionales
 - Tienden al conflicto o las contradicciones entre ellos
 - No existen recomendaciones y reglas “universalmente aceptadas” para determinar cuando se satisfacen de forma óptima ciertos requisitos no funcionales



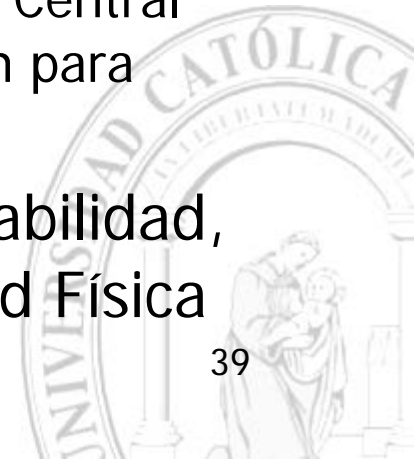
2.1. Requisitos funcionales y no funcionales. Dominio

- Los **Requisitos No Funcionales** se suelen extraer de restricciones de los interesados (**Stakeholders**).
 - Ejemplos:
 - Objetivos de Negocio Críticos
 - Características esenciales del Sistema (ej: Seguridad)
 - Seguridad Física, Rendimiento, Mantenibilidad, etc..
 - Estos intereses o preocupaciones suelen estar definidos de forma “vaga”



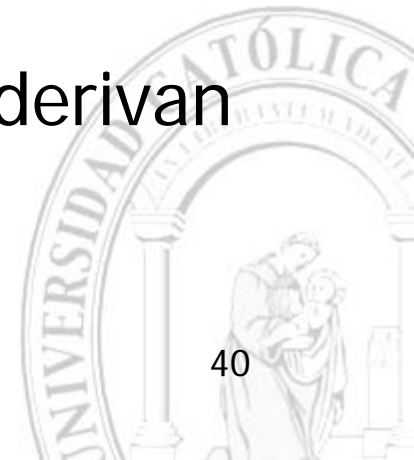
2.1. Requisitos funcionales y no funcionales. Dominio

- Los **Requisitos No Funcionales** son especialmente importantes en Sistemas Críticos
 - Es decir, aquellos cuyos fallos causan un daño significativo de tipo económico, físico o humano a las organizaciones o personas.
 - Ejemplos:
 - Negocio: Sistema reserva aerolínea
 - Misión: Sistema de control de órbita de un satélite
 - Seguridad Física o Humana: Sistema de Control Central Nuclear, Sistema médico de control de radiación para tratamiento de Cáncer
- Los principales RNF en estos sistemas son: Fiabilidad, Rendimiento, Seguridad, Usabilidad, Seguridad Física



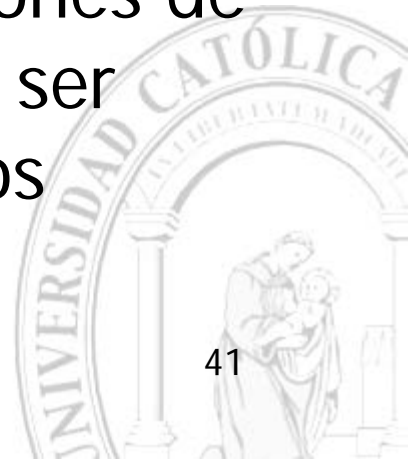
2.1. Requisitos funcionales y no funcionales. Dominio

- **Requisitos del dominio:**
 - Proviene del dominio de aplicación del sistema.
 - Describen características y necesidades propias del dominio (comunes a todas las organizaciones de ese sector).
 - *Funcionales o no funcionales*
 - Son **requisitos de usuario** que se derivan del dominio de aplicación.



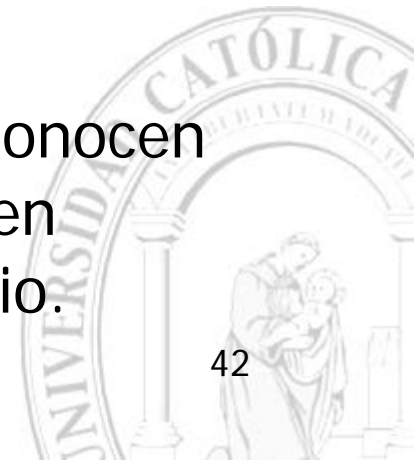
2.1. Requisitos funcionales y no funcionales. Dominio

- Ejemplos de **Requisitos del dominio**:
 - Dominio: Bibliotecas
 - RD1: El interfaz de usuario será común para acceder a todas las bases de datos electrónicas y estará basado en la norma XYZ.
 - RD2: Dependiendo de las restricciones de copyright, los documentos podrán ser impresos o sólo visualizados por los usuarios.



2.1. Requisitos funcionales y no funcionales. Dominio

- Principales problemas con los **Requisitos de Dominio** son:
 - **Comprensibilidad:**
 - Los requisitos utilizan el lenguaje y vocabulario habitual en el dominio de aplicación.
 - Los ingenieros software que desarrollan el sistema no conocen dicho lenguaje.
 - **Sobreentendidos:**
 - Los expertos en el área del dominio la conocen tan bien que inconscientemente no hacen explícitos todos los requisitos del dominio.



Índice

1. Concepto de requisito
2. Tipos de requisitos
 1. Requisitos funcionales y no funcionales. Dominio.
 2. Requisitos de usuario y sistema. Software.
 3. Requisitos de cliente y requisitos de desarrollador.

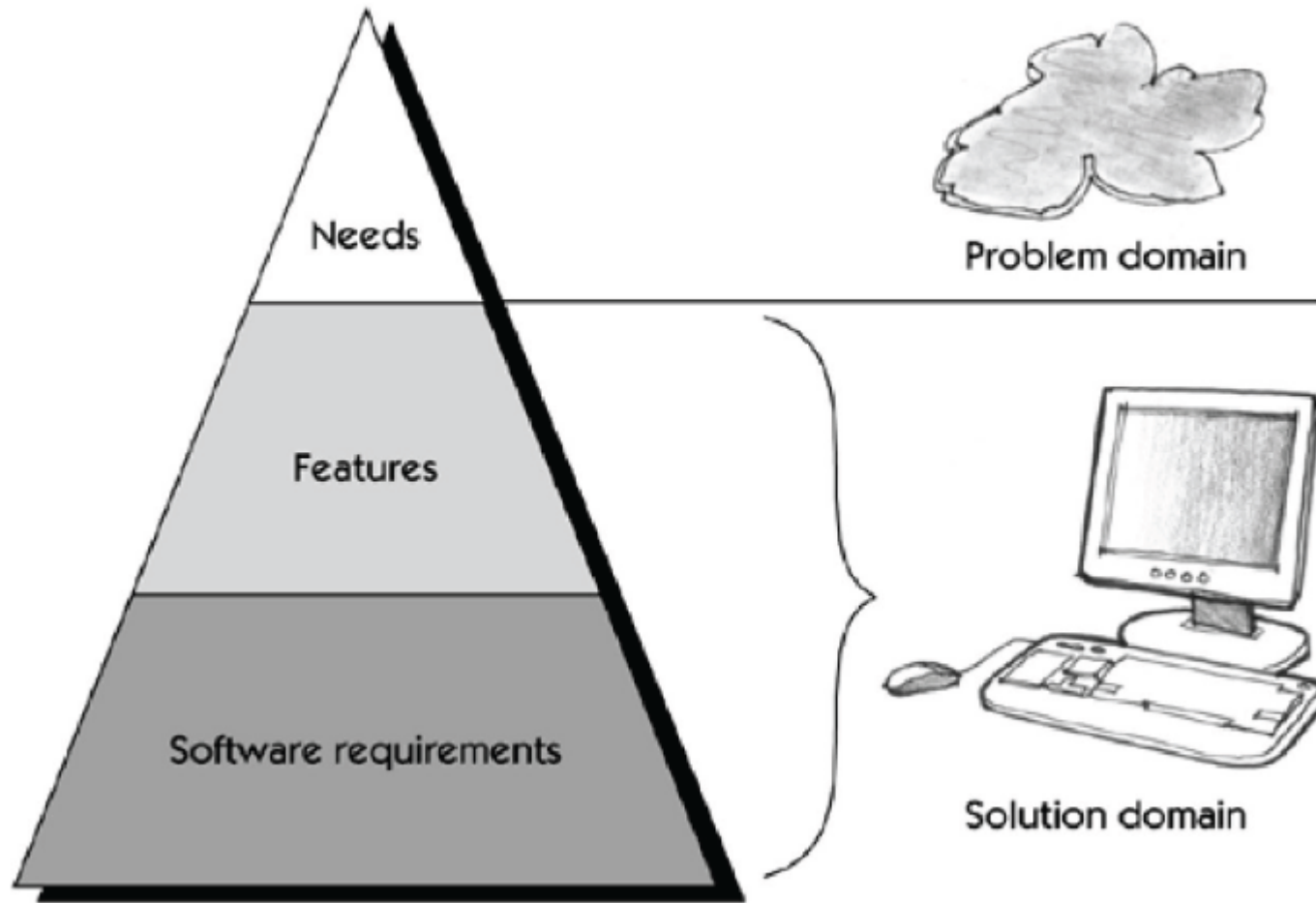


2.2. Requisitos del usuario y del sistema. Software

- Los requisitos se pueden definir a distintos niveles de abstracción o detalle:
 - **Usuario** (needs):
 - **Sistema** (features):
 - **Software** (SW requirements)



2.2. Requisitos del usuario y del sistema. Software



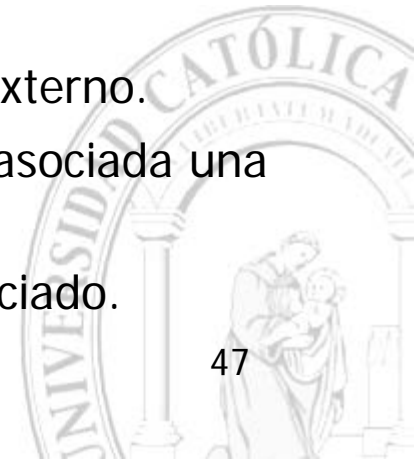
2.2. Requisitos del usuario y del sistema . Software

- **Requisitos de Usuario:**
 - Declaraciones en lenguaje natural y, quizás, tablas y diagramas, de los servicios que el sistema provee y sus restricciones operacionales.
 - Ejemplo:
 - RU1: El software debe proveer un medio de representar y acceder a los ficheros externos creados por otras herramientas.



2.2. Requisitos del usuario y del sistema. Software

- **Requisitos de Sistema:**
 - Un documento estructurado con las descripciones detalladas de las funciones, servicios y restricciones operacionales del sistema.
 - Son la base del diseño del sistema (definen lo que deberá ser implementado).
 - Puede ser parte de un contrato con el cliente.
 - Ejemplos:
 - RS1: El usuario debe poder elegir el tipo de fichero externo.
 - RS2: Cada tipo de fichero externo debe poder tener asociada una herramienta externa para editarlo y mostrarlo.
 - RS3: Cada tipo de fichero externo tiene un icono asociado.



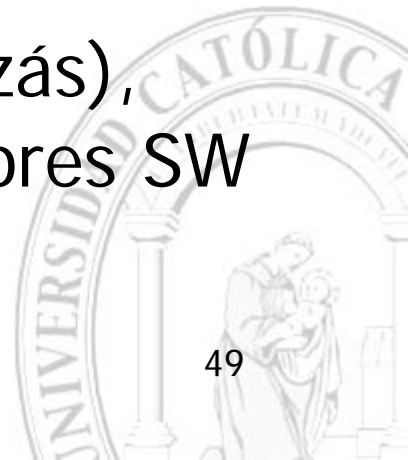
2.2. Requisitos del usuario y del sistema. Software

- **Requisitos de Software:**
 - Declaraciones detalladas de diseño e implementación del software.
 - Ejemplo:
 - RSW1: Los iconos de los tipos de fichero se guardan en archivos JPG.



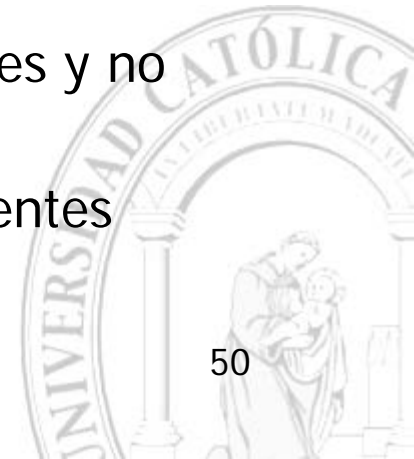
2.2. Requisitos del usuario y del sistema. Software

- Comunican información a un diferente **lector**:
 - **Usuario** → Gestores clientes, Usuarios Finales, Ingenieros Clientes, Gestores Contratistas, Arquitectos del Sistema
 - **Sistema** → Usuarios Finales, Ingenieros Clientes, Arquitectos del Sistema, Desarrolladores SW
 - **Software** → Ingenieros Clientes (quizás), Arquitectos del Sistema, Desarrolladores SW



2.2. Requisitos del usuario y del sistema. Software

- **Requisitos de Usuario:**
 - Describen requisitos de manera comprensible por los usuarios sin conocimientos técnicos detallados.
 - Se definen mediante lenguaje natural, tablas y diagramas.
 - Problemas con el lenguaje natural:
 - Falta de claridad: es difícil conjuntar precisión y facilidad de lectura.
 - Confusión entre requisitos: Requisitos funcionales y no funcionales tienden a estar mezclados.
 - Amalgama de requisitos: Varios requisitos diferentes pueden ser expresados juntos.



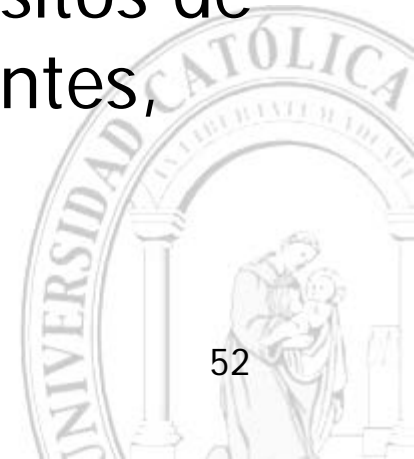
2.2. Requisitos del usuario y del sistema. Software

- Los **Requisitos de Usuario** pueden responder a varios orígenes:
 - Dominio del problema (**Requisitos de Dominio**)
 - Intereses de la organización (Requisitos de Negocio u Organizacionales)
 - Necesidades de los usuarios finales del software.



2.2. Requisitos del usuario y del sistema. Software

- Los **Requisitos de Sistema**
 - Requisitos para el sistema en su conjunto.
 - Un **Sistema** engloba, entre otros, elementos de tipo hardware, software, firmware, personas, información, técnicas, facilidades y servicios.
 - Estos requisitos engloban a los requisitos de diferentes interesados: usuarios, clientes, autoridades, etc.



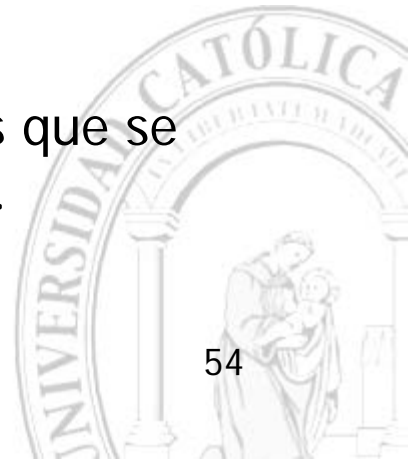
2.2. Requisitos del usuario y del sistema. Software

- Los **Requisitos de Software**
 - Solo se refieren al software, pero en un sistema completo se derivan de los requisitos del sistema.



Puntos clave

- Los requisitos → que debe hacer el sistema y restricciones en su funcionamiento e implementación.
- Los requisitos funcionales → son declaraciones de los servicios que el sistema debe proporcionar o son descripciones de cómo se deben llevar a cabo algunos cálculos.
- Los requisitos no funcionales
 - restringen el sistema en desarrollo y el proceso de desarrollo que se debe utilizar.
 - Producto, organizaciones o externo.
 - Propiedades emergentes del sistema
- Los requisitos del dominio → son requisitos funcionales que se derivan de las características del dominio de aplicación.



Puntos clave

- Los requisitos del usuario
 - Personas relacionadas con la utilización y obtención del sistema.
 - Lenguaje natural, tablas y diagramas fáciles de entender.
- Los requisitos del sistema
 - Funciones que debe proporcionar el sistema.
 - Especificación precisa.
 - Reducir ambigüedad: formulario estructurado del lenguaje natural complementado con tablas y modelos del sistema.



Índice (Continuación)

1. Concepto de requisito
2. Tipos de requisitos
 1. Requisitos funcionales y no funcionales. Dominio.
 2. Requisitos de usuario y sistema. Software.
 3. Requisitos de cliente y requisitos de desarrollador.

