

**Curso 2020-2021**

# **Ingeniería del Software**

Un enfoque crítico

## **REQUISITOS**

## Contenido

- Tema I. Ingeniería de requisitos
  - Unidad 1. Introducción a la ingeniería de requisitos
  - Unidad 2. Obtención y descripción de requisitos
  - Unidad 3. Cómo escribir buenos requisitos
  - Unidad 4. Tipos de requisitos
  - Unidad 5. Propiedades, atributos y organización de los requisitos
- Tema II. Modelado conceptual
  - Unidad 6. Modelado conceptual (1)
  - Unidad 7. Modelado conceptual (2)
  - Unidad 8. Modelado conceptual (3)
  - Unidad 9. Sobre la diferencia entre análisis y diseño (artículo)
- Tema III. Modelado arquitectónico
  - Unidad 10. Arquitectura del software
  - Unidad 11. Componentes, dependencias e interfaces
  - Unidad 12. Diseño por contratos

## Introducción a la ingeniería de requisitos

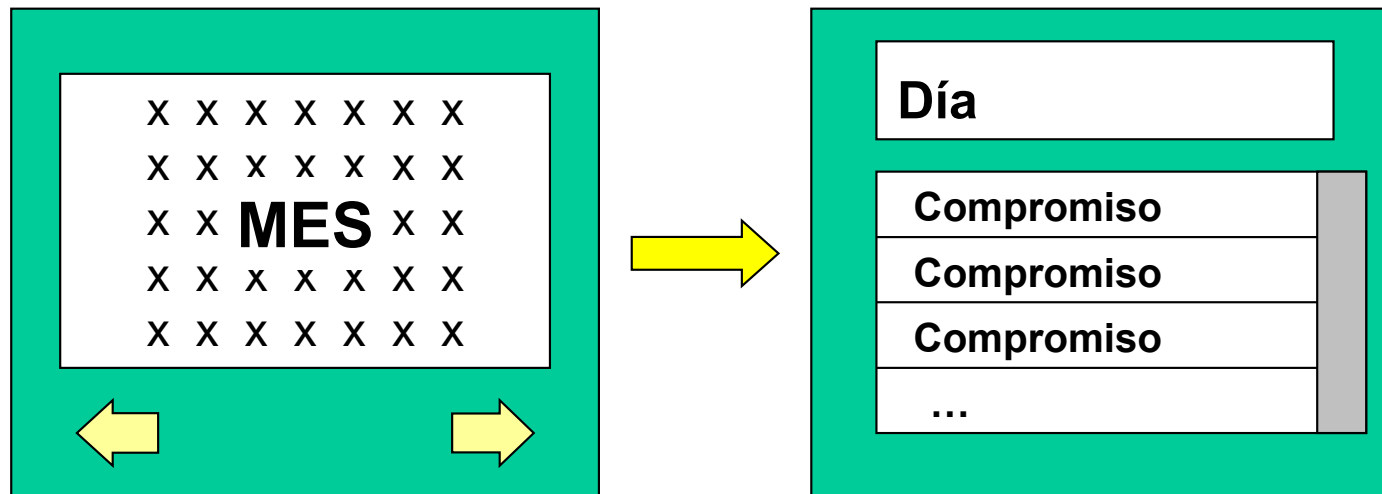
- Qué es la Ingeniería de Requisitos
  - Captura y Análisis
  - Definición de requisito
  - El documento de especificación de requisitos
- Necesidad de la Ingeniería de Requisitos
  - Para construir algo, antes hay que entender qué es ese “algo”
- Por qué es necesario escribir los requisitos
  - Sin requisitos escritos es imposible...
  - No hay ingeniería profesional sin requisitos escritos
- Dificultades de la Ingeniería de Requisitos
  - El precio pagado: es una necesidad, no un lujo
- La Ingeniería de Requisitos en el contexto del proyecto
  - Pre-proyecto: valor contractual, viabilidad, planificación, estimación...

## La ingeniería del software

- Ingeniería del software
  - “Es la aplicación sistemática de conocimientos, métodos y experiencias científicas y tecnológicas al diseño, implementación, pruebas y documentación de software” (IEEE 24765-2010 Systems and Software Engineering – Vocabulary).
- Ingeniería de requisitos:
  - “Es el desarrollo sistemático de los requisitos a través de un proceso **iterativo** y **cooperativo** en el que se analiza el problema, se **documenta** el resultado en diversos formatos de representación, y se **comprueba** la exactitud de la comprensión alcanzada” (Loucopoulos & Karakostas. Systems Requirements Engineering. McGraw-Hill, 1995).
- Peculiaridades de la ingeniería software / ingeniería informática
  - El “producto” software
  - Mucho desarrollo, poca disciplina ingenieril
  - Necesidad de describir y documentar lo que se va a producir
  - Cambios frecuentes en el producto

## Un caso práctico

- “Necesito algo para organizar mejor mis actividades, una agenda para llevar al día mi horario, mis compromisos, etc.”



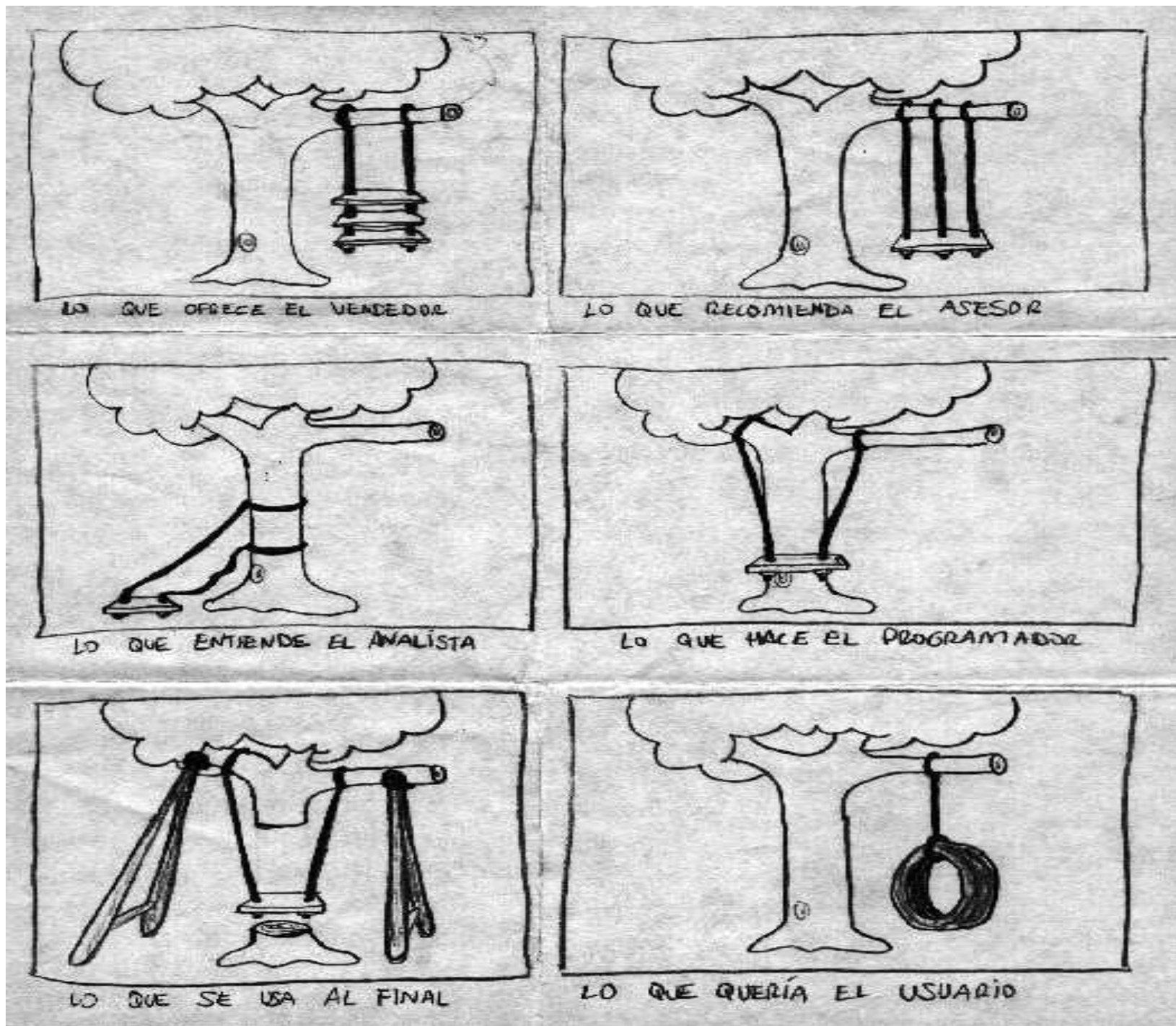
# Definición de requisito

- Según IEEE (24765-2010), un requisito es:
  - Una condición o capacidad que un usuario necesita.
  - Una condición o capacidad que debe poseer un sistema.
  - Una representación documentada de una condición o capacidad.
- Dos tipos principales de requisitos:
  - Requisitos de capacidad (funcionales): funciones y operaciones requeridas para resolver un problema o alcanzar un objetivo.
  - Requisitos de restricción (no funcionales): restricciones impuestas sobre la manera en que el problema es resuelto o el objetivo es alcanzado.

## The Chaos Report

- The Standish Group International: realiza estudios desde 1994
- 2003: datos de 13.522 proyectos de tecnologías de la información

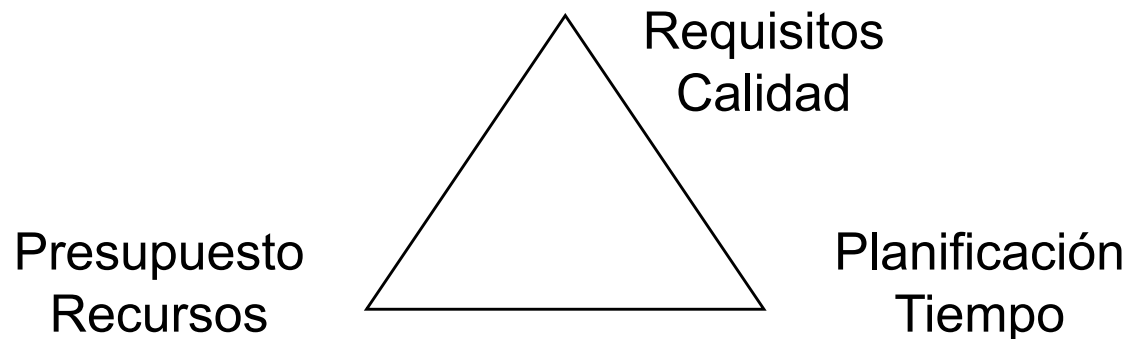
	1994	1996	1998	2000	2003
<b>Proyecto exitoso</b>	16%	27%	26%	28%	34%
<b>Proyecto completo pero deficiente</b>	53%	33%	46%	49%	51%
<b>Proyecto cancelado</b>	31%	40%	28%	23%	15%



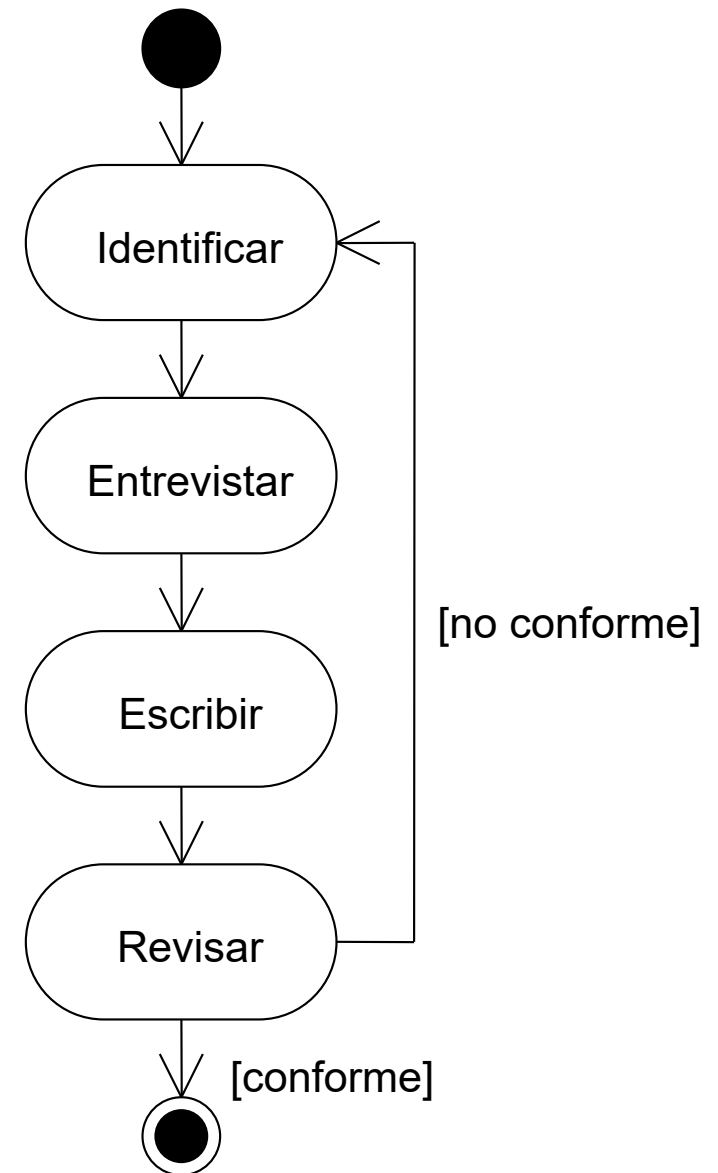


## Obtención y descripción de requisitos del usuario

- Las fuentes de los requisitos
- Plan de trabajo para obtener los requisitos
- Identificación de *stakeholders*
  - ¿Quiénes tienen interés en el producto?
  - ¿Quién es el cliente?
  - Intereses contrapuestos, requisitos contradictorios
  - Negociar el equilibrio



- Cómo llevar adelante una entrevista
  - Antes, durante, después...



## Técnicas para la obtención y descripción de requisitos

- Textuales
  - Relativamente accesibles a un cliente sin formación específica
    - Texto en prosa común y corriente
    - Texto estructurado, lenguaje técnico
    - Casos de uso
    - Tabla de roles de usuario y servicios/funciones requeridos por cada uno
- Gráficas
  - Requieren un cierto grado de formación técnica en el cliente
  - Tienen el peligro de convertir el análisis de requisitos en diseño
    - Diagramas de flujo de datos
    - Diagramas de actividad
    - Diagramas de estados
- Interfaces de usuario y prototipos
  - No confundir con diseño, valorar la inversión

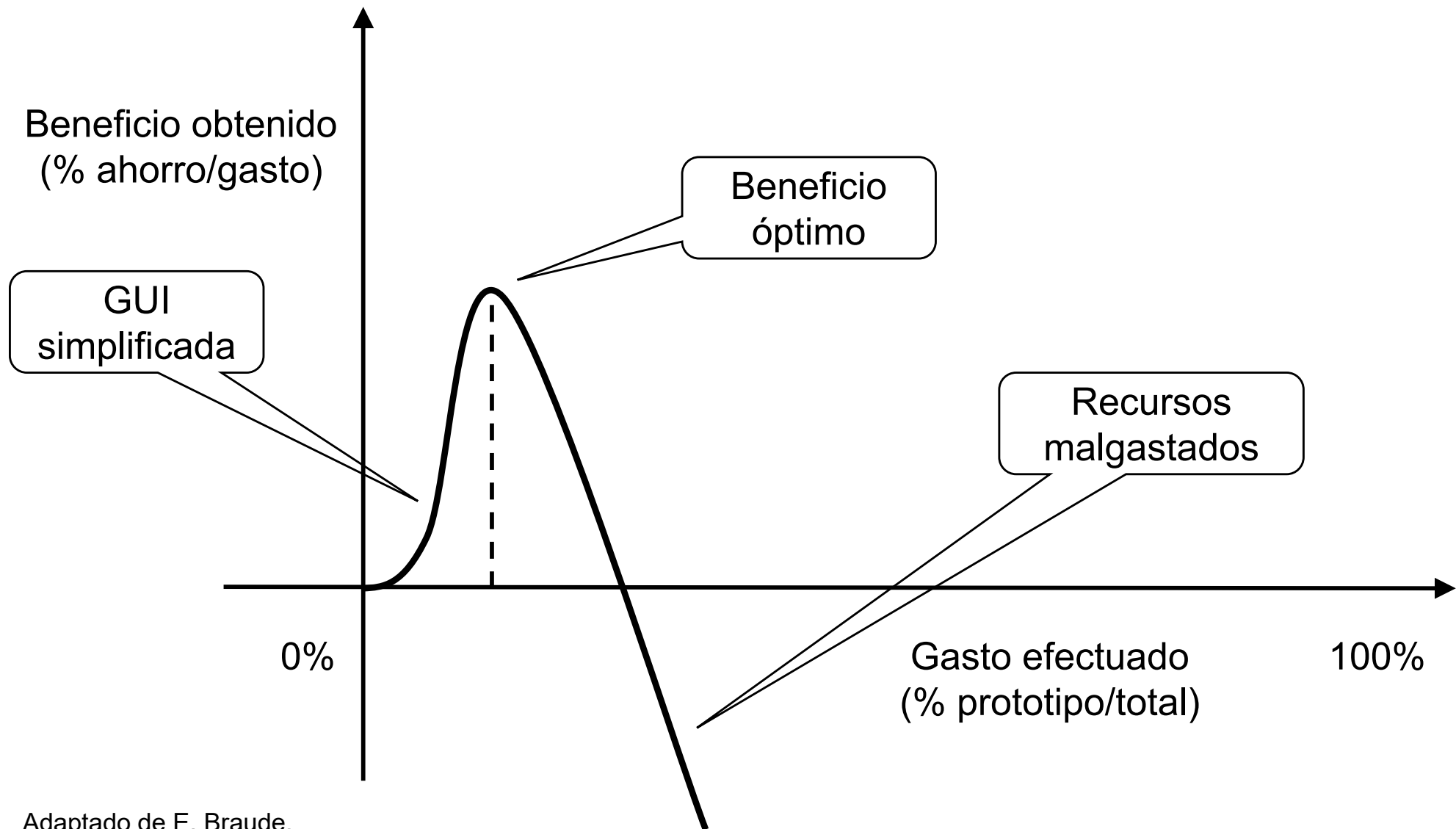
## Ejemplo: Feria de subastas

- Se desea modelar un sistema informático para gestionar las transacciones en un recinto ferial de subastas. Cualquier persona que haya logrado acceso al recinto de la feria puede conectarse al sistema a través de alguno de los muchos terminales disponibles, y participar en las subastas que tengan lugar, en alguna de las modalidades ofrecidas por el sistema, es decir, como comprador, como vendedor, o como simple observador.
- Para subastar algún artículo es necesario darse de alta como vendedor. El vendedor puede registrar artículos en la subasta, rellenando una ficha por cada artículo (foto, descripción, etc.), que sale así inmediatamente a subasta.
- Análogamente, para participar en una subasta es necesario darse de alta como comprador. El comprador puede pujar por cualquiera de los artículos subastados en la feria. Cuando no se produce ninguna nueva puja, el artículo queda definitivamente adjudicado al comprador. Si un artículo no ha recibido ninguna puja, el vendedor puede modificar alguno de sus datos. La compra-venta se formaliza posteriormente en las oficinas ante un administrador que actúa como intermediario.
- Cualquier persona puede participar como observador en una subasta, es decir, puede consultar la lista de artículos subastados y seleccionar uno de ellos para examinar la lista de pujas. Un observador puede registrarse como vendedor o comprador para participar activamente en la subasta.

## Feria de subastas: tabla de roles/servicios

Rol de Usuario	Servicio o función requerido
Anónimo	Consultar lista de artículos subastados Alta como usuario vendedor Alta como usuario comprador
Vendedor	Poner un artículo en subasta Modificar los datos del artículo no vendido
Comprador	Pujar por un artículo subastado
Vendedor Comprador Administrador	Formalizar la compra-venta en oficinas
Administrador	...

## Beneficio de la inversión en prototipos



Adaptado de E. Braude,  
Software Engineering: An Object-Oriented Perspective

## Cómo debería ser una especificación de requisitos

- **Completa:** describe todas las necesidades relevantes para los *stakeholders*.
- **Consistente:** carece de conflictos entre requisitos.
- **Correcta:** todo es pertinente y no contiene errores.
- **Modificable:** facilidad para efectuar cambios de forma sencilla, completa y consistente.
- **Verificable:** existencia de un proceso acotado que determine si el sistema final satisface el requisito.
- **Trazable:** el origen del requisito está marcado de forma clara; y se puede seguir el impacto del requisito a lo largo del desarrollo.
- **Clara y no ambigua:** una única interpretación.

*IEEE 830, 1998*

# Los siete pecados del especificador

- Ruido
  - Presencia en el texto de información irrelevante para el problema.
- Silencio
  - Existencia de aspectos del problema no cubiertos por la especificación.
- Sobre-especificación
  - Elementos que no corresponden al problema sino a una posible solución.
- Contradicción
  - Una misma característica definida de dos o más formas incompatibles.
- Ambigüedad
  - Elementos del texto que admiten varias interpretaciones distintas.
- Referencia futura
  - Referencia a una característica del problema definida más adelante.
- Pensamiento ilusorio
  - Definir el problema de tal forma que imposibilita una solución realista.

*(Bertrand Meyer, On Formalism in Specification, IEEE Software, Jan 1985, pp. 6-26)*

# Especificación inteligente de requisitos

Sigla	Concepto	Descripción
S	eSpecífico	Claros y simples: qué, por qué...
M	Medible	Se puede cuantificar y evaluar
A	Alineado	Con la estrategia o con el fin del sistema
R	Realista	Puede conseguirse con un número de recursos lógico
T	limitado en Tiempo	Establece un periodo de tiempo claro



"I believe that this nation should commit itself to achieving the goal, before this decade is out, of landing a man on the Moon and returning him safely to Earth"



## Estructura de la especificación

- Un proyecto mediano puede tener centenares de requisitos
- Escribir los requisitos para no olvidarlos. De esta forma:
  - Pueden ser firmados, con lo que son pieza clave en contratos
  - Son la fuente del diseño
  - Se verificará el software contra ellos
- La correcta organización de los mismos es vital
- Claves:
  - Utiliza estándares de estructuración de especificaciones de requisitos
  - Aclara el objetivo global a cumplir por el sistema
  - Emplea descripciones textuales y gráficas
  - Ordena y agrupa tus requisitos de forma lógica
  - Relaciona unos requisitos con otros para facilitar su entendimiento
  - Relaciona los requisitos con otros activos

## **Especificaciones completas**

- ¿Qué podemos hacer para no olvidar nada...?
  - Revisión por pares: junto con compañeros más experimentados, expertos en la materia, cliente y otros interesados
  - Emplear check-lists
  - Comparar la especificación contra taxonomías propias de la materia
  - Reutilizar requisitos de proyectos previos
    - Reutilización de grano grueso: componentes reutilizables
    - Reutilización de grano fino: buscadores avanzados

## **Especificaciones consistentes**

- El primer paso hacia la consistencia es evitar redundancia...
  - ... y las inconsistencias provocan retrabajo
- Las revisiones en grupo permiten detectar parte de estas redundancias
- Técnicas automáticas para su apoyo:
  - Comparación de grafos semánticos
  - Detección de unidades inconsistentes
    - “NASA y ESA reconocen la pérdida de una sonda marciana por haber diseñado dos módulos empleando diferentes sistemas de medida”
    - Revisiones globales de la especificación
    - Detección automática de unidades de medida inconsistentes

## Especificaciones consistentes

- Comparación de grafos semánticos:

UR001: ....

UR023: El sistema deberá enviar notificaciones semanales de nuestras ofertas a todos los clientes

URxxx: ...

UR842: La aplicación debe ser capaz de notificar periódicamente a sus clientes sobre nuestras ofertas

UR999: ...

# Una especificación de requisitos no es una novela (1)

Novela	Especificación de requisitos
Múltiples estilos narrativos	Estilo narrativo objetivo, “plano”
<p>Abiertas a todo tipo de licencias poéticas y adornos:</p> <p><i>Generalmente se viste de manera victoriana, incluyendo un traje, botas de montar, y una ostentosa corbata de moño intrincadamente atada...</i></p>	<p>Licencias poéticas no permitidas:</p> <ul style="list-style-type: none"> <li>• Textos simples y claros para facilitar su lectura y entendimiento</li> <li>• Siguiendo un conjunto de gramáticas fijas y simples</li> <li>• Usando la voz activa en lugar del pasivo</li> <li>• Evitando terminología técnica no definida, abreviaturas...</li> <li>• Utilizando un vocabulario controlado, donde los términos estén bien consensuados</li> <li>• Prescindiendo de especulaciones</li> </ul>
<p>Suelen jugar con nuestra imaginación:</p> <p><i>La verdad es que Robert parecía mayor, no aparentaba los 32 años que tenía.</i></p>	<p>No pueden jugar con nuestra imaginación ni dejar “cabos sueltos”:</p> <ul style="list-style-type: none"> <li>• Los requisitos, además de simples, deben ser fáciles de medir.</li> <li>• Ambigüedad cero, para que todos los interesados interpretemos cada requisito de una única forma.</li> </ul>

## Una especificación de requisitos no es una novela (2)

Novela	Especificación de requisitos
Puede hacer referencias a otros textos o a elementos implícitos del entorno cultural del lector.	No deben dar ningún conocimiento por sentado: <ul style="list-style-type: none"><li>• Deben ser autocontenidos.</li><li>• En el mejor de los casos, acompañados incluso de glosarios.</li></ul>
Pueden mezclar diferentes hilos argumentales, dar saltos atrás y adelante en el tiempo...	Debe adoptarse un orden lógico y bien estructurado. <ul style="list-style-type: none"><li>• Cada requisito debe ser atómico, debe expresarse una única necesidad por requisito.</li></ul>
Se redactan con maravillosos procesadores de texto.	Deben escribirse con herramientas especializadas. Con ello conseguimos: <ul style="list-style-type: none"><li>• Identificarlos unívocamente</li><li>• Atomizarlos</li><li>• Organizarlos, categorizarlos y relacionarlos</li><li>• Reutilizarlos por separado o en conjunto</li><li>• Medir su calidad individual o global</li></ul>

# Redacción del requisito: indicadores de calidad

- Tamaño del requisito:
  - El justo, ni muy breve ni muy largo
  - Tamaño medido en caracteres, palabras, párrafos
- Legibilidad:
  - La máxima posible
  - Procesadores como MS Word miden la legibilidad de los textos
- Tiempo verbal:
  - Forma “imperativa” en lugar de condicional, usar estilo uniforme en todos los requisitos
    - el usuario **puede** ver (enviar, abrir...) / el usuario **podrá** ver (enviar, abrir...): ok
    - el usuario **debe poder** ver / **deberá ser capaz de** ver: demasiado recargado
- Modo verbal:
  - Activa en lugar de pasiva
- Sentencias opcionales y especulativas:
  - Evitar sentencias del estilo “quizá...” , “posiblemente...”, “usualmente...”, “casi siempre...”
- Expresiones ambiguas:
  - Evitar expresiones del estilo: “rápido”, “amigable”...

# Redacción del requisito: indicadores de calidad

- Sentencias subjetivas:
  - Evitar sentencias del tipo: “yo creo...”, “en mi opinión...”
- Sentencias implícitas (empleo de pronombres):
  - Evitar el abuso de los pronombres
  - A la hora de la lectura, podríamos dudar en qué nombre sustituye cada nombre
- Conectores:
  - El abuso de conectores puede indicar que se está incluyendo más de una necesidad en el mismo requisito
  - También puede indicar un exceso de detalle
- Negaciones:
  - Más de una palabra negativa en la misma frase podría hacerla difícil de entender
- Sentencias incompletas:
  - Evitar sentencias del tipo “etcétera...”, “...entre otros...”, “...”
- Términos propios de diseño o de flujo:
  - Evitar términos que denotan diseño como por ejemplo “clave ajena”, “tabla hash”...



# Redacción del requisito: indicadores de calidad

- Número de términos del dominio:
  - Un exceso de términos del dominio puede indicar que se están mezclando diferentes necesidades en el mismo requisito
  - También puede indicar que se está dando un excesivo detalle
- Número de verbos principales (del dominio):
  - Un exceso de verbos del dominio puede indicar que se están mezclando diferentes necesidades en el mismo requisito
  - También puede indicar que se está dando un excesivo detalle
- Acrónimos y abreviaturas:
  - Sólo permitidos si están definidos en alguna sección del documento de requisitos
- Estructura gramatical:

Usuario	El operador del <i>Call Center</i> ...
Acción	... deberá ser capaz de ver...
Objeto	... detalles de cada compañía contactada...
Cualificador	... en menos de dos segundos.

## Errores típicos (1): modo condicional

- La información sobre los usuarios **debería** almacenarse en memoria dentro de una tabla hash, o bien en una tabla de base de datos, con una clave ajena a la tabla de Usuarios.
  - Evite el uso del modo condicional
  - Sustitúyalo por el imperativo
- La información sobre de los usuarios **deberá** almacenarse en memoria dentro de una tabla hash, o bien en una tabla de base de datos, con una clave ajena a la tabla de Usuarios.

## Errores típicos (2): detalles de diseño

- La información sobre los usuarios deberá almacenarse en memoria dentro de una **tabla hash**, o bien en una **tabla de base de datos**, con una **clave ajena** a la **tabla** de Usuarios.
  - Evite detalles propios de diseño
- La información sobre los usuarios deberá almacenarse **en el sistema**.

## Errores típicos (3): opcionalidad

- El administrador deberá ser capaz de insertar, borrar, mostrar y actualizar la información sobre los usuarios. **Opcionalmente**, deberá también ser capaz de generar un informe y enviarlo por e-mail al cliente.
  - Exprese la opcionalidad mediante un atributo del requisito, nunca como texto dentro del requisito
- El administrador deberá ser capaz de insertar, borrar, mostrar y actualizar la información sobre los usuarios. Deberá también ser capaz de generar un informe y enviarlo por e-mail al cliente.
  - Necesidad: opcional.

## Errores típicos (4): atomicidad

- El administrador deberá ser capaz de **insertar**, **borrar**, **mostrar** y **actualizar** la información sobre los usuarios. Deberá también ser capaz de **generar** un informe y enviarlo por e-mail al cliente.
  - Use un requisito individual para cada necesidad. Muchos verbos concentrados en un requisito pueden implicar la mezcla de diferentes necesidades.
- El Administrador deberá ser capaz de **añadir** usuarios.
  - Necesidad: obligatorio.
- ...
- El Administrador podrá **generar** un informe para enviarlo por e-mail al cliente.
  - Necesidad: opcional.

## Errores típicos (5): acrónimos y vagüedad

- El sistema debe ser capaz de importar ficheros **ABC**. El proceso debe ser **amigable** y **rápido** para el usuario.
  - Utilice acrónimos sólo cuando estén comúnmente aceptados por todos los interesados.
  - Términos como ‘amigable’ y ‘rápido’ son difíciles de medir y, por lo tanto, imposibles de probar de forma correcta.
  - Utilice unidades físicas para expresar el rendimiento en un requisito.
  - Utilice otros medios (p.e. WAI AA\*) claramente definidos para indicar cómo de amigable o accesible debe ser un sistema.

\* *Web Accessibility Initiative, Nivel Doble-A*

## Errores típicos (5): puntuación y legibilidad

- El administrador deberá ser capaz de crear facturas asociadas con las diferentes compañías que estén dadas de alta en el sistema y éste también deberá estar al tanto de facturas impagadas para que puedan generar un mail y enviárselos a ellos.
  - El uso apropiado de signos de puntuación hará los requisitos más fáciles de leer.
  - El número de sílabas por palabra y palabras por frase es también un buen indicador de la legibilidad del requisito.
- El administrador deberá ser capaz de crear facturas asociadas con las diferentes compañías que estén dadas de alta en el sistema. **Éste** también deberá estar al tanto de facturas impagadas para que puedan generar un mail y enviárselos a ellos.

## Errores típicos (6): pronombres

- El administrador deberá ser capaz de crear facturas asociadas con las diferentes compañías que estén dadas de alta en el sistema y **éste** también deberá estar al tanto de facturas impagadas para que puedan generar un mail y enviárse**los** a **ellos**.
  - El exceso de pronombres puede hacer un requisito difícil de entender
  - El pronombre 'éste', ¿se refiere al administrador o al sistema?
  - El pronombre 'ellos', ¿se refiere al administrador o al cliente?
  - ¿Por qué el pronombre 'los' está en plural?
- El administrador deberá ser capaz de crear facturas asociadas con las diferentes compañías que estén dadas de alta en el sistema. **El administrador** también deberá estar al tanto de facturas impagadas para que puedan generar un mail y enviar**lo** a los **clientes**.



## Errores típicos (7): pseudocódigo

- El administrador deberá ser capaz de crear facturas asociadas con las diferentes compañías que estén dadas de alta en el sistema. El administrador también deberá estar al tanto de facturas impagadas para que puedan generar un mail y enviarlo a los clientes. El proceso para localizar impagados es el siguiente:
  1. **Iterar** sobre todas las facturas
  2. **Si** *Fecha\_Factura + CondicionesPago* es mayor que la fecha actual, **entonces**:
    - **Si** la categoría del cliente es A, **entonces** se le deja un mes extra
    - **Si no**, **mientras** la factura no esté pagada no se le permite generar nuevas facturas y se le enviará un mail cada semana
- Evite el uso de pseudocódigo en sus requisitos.
- Los requisitos extensos (en caracteres o párrafos) pueden indicar baja calidad.

## Errores típicos (8): número de términos

- Los **clientes** podrán remitir **órdenes** por **Internet**. Estas **órdenes** deben incluir **fecha de envío** y **cantidad de artículos**. Una vez que se recibe la **orden**, el **equipo de empaquetado** debe recoger todos los **artículos** y enviar un **mail** al **cliente**. Deben soportarse los **protocolos *http* y *https***, así como los **navegadores Explorer y Firefox**. La **resolución** mínima será de 1024x768.
  - Un exceso de términos diferentes en el mismo requisito puede indicar:
    - Que se están mezclando diferentes necesidades en un único requisito.
    - Que se está proporcionando demasiado detalle.
  - Igualmente, muchos verbos pueden involucrar diferentes necesidades mezcladas en un único requisito.

## Errores típicos (9): subjetividad, exceso de negaciones

- En mi opinión, ningún cliente debería poder nunca enviar órdenes al equipo de empaquetado. Ya lo hicimos así en un proyecto hace tres años y el resultado fue nefasto.
  - No opine, no divague al redactar el requisito. Límitese a indicar qué es lo que debe hacer el sistema.
- En mi opinión, ningún cliente debería poder nunca enviar órdenes al equipo de empaquetado. Ya lo hicimos así en un proyecto hace tres años y el resultado fue nefasto.
  - No mezcle demasiados términos negativos, ya que a veces puede dificultar la lectura del requisito/restricción
- Un cliente no podrá enviar órdenes directamente al equipo de empaquetado.

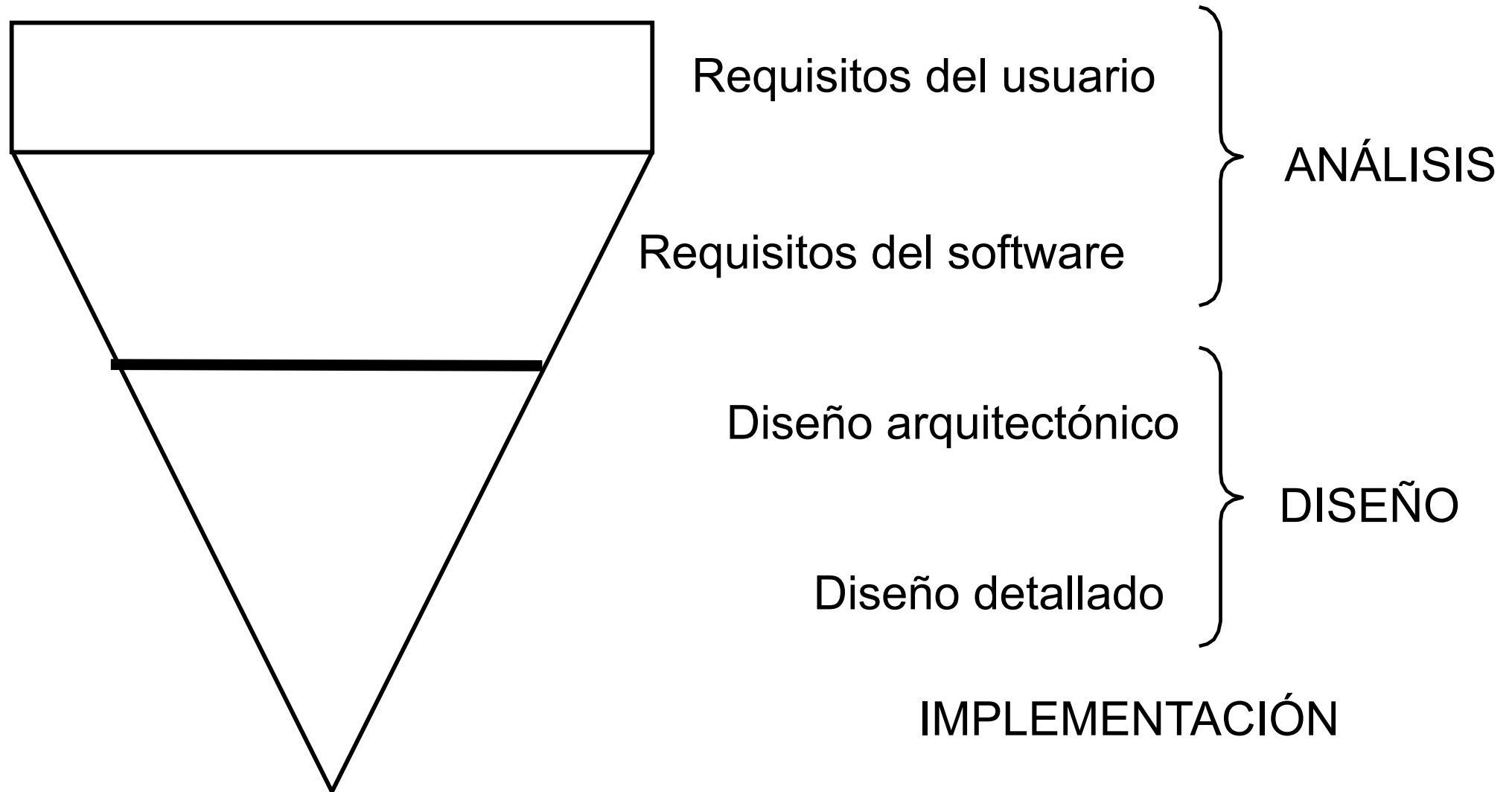
## Errores típicos (10): falta de precisión

- Generalmente, el sistema debe ser capaz de terminar el proceso de rastreo **sin sobrecargar excesivamente el servidor**.
  - Evite expresiones vagas como: ‘generalmente’, ‘comúnmente’.
  - Verifique si cada aserción puede ser medida de forma sencilla.
- El sistema debe ser capaz de terminar el proceso de rastreo **en un tiempo inferior a 2 segundos y sin que el proceso sobrepase la ocupación máxima de 250 MB de memoria**.

## Tipos de requisitos del software

- Dos niveles en los requisitos
  - Requisitos del Usuario, Requisitos del Software
- Clasificación de requisitos del software
  - **Requisitos negativos**
  - **Requisitos funcionales** – decir el qué (análisis)
    - Requisitos de información / Requisitos de operación
    - Modelo del sistema: modelo conceptual + modelo de casos de uso
  - **Requisitos no funcionales** – restringir el cómo (pre-diseño)
    - Características distintivas
    - Ejemplos
    - Análisis y documentación
- Trazabilidad: hacia atrás / hacia delante

# Requisitos del Usuario vs. Requisitos del Software



# Requisitos del Usuario-Requisitos del Software: diferencias

	Requisitos del Usuario	Requisitos del Software
<b>objetivo</b>	planteamiento del problema captura de requisitos	refinamiento del problema análisis de requisitos
<b>fuelle</b>	usuario/cliente	usuario/cliente y analista
<b>responsable</b>	usuario/cliente	analista
<b>audiencia</b>	usuario/cliente (y desarrollador)	desarrollador (y usuario/cliente)
<b>énfasis</b>	perspectiva del producto características de los usuarios entorno operacional captura mediante prototipos	conocimiento de expertos modelado, métodos formales organización, no dejar cabos sueltos consistencia y completitud

## Flujos de trabajo vs. Documentos

Flujos de trabajo		Documentos	
USDP	Clásica	IEEE	ESA
Requisitos	Análisis de requisitos	Software Requirements Specification (IEEE 830-1993)	User Requirements Document
Análisis			Software Requirements Document
Diseño	Diseño arquitectónico	Software Design Document (IEEE 1016-1987)	Architectural Design Document
	Diseño detallado		Detailed Design Document

+ Implementación, Pruebas, Mantenimiento...



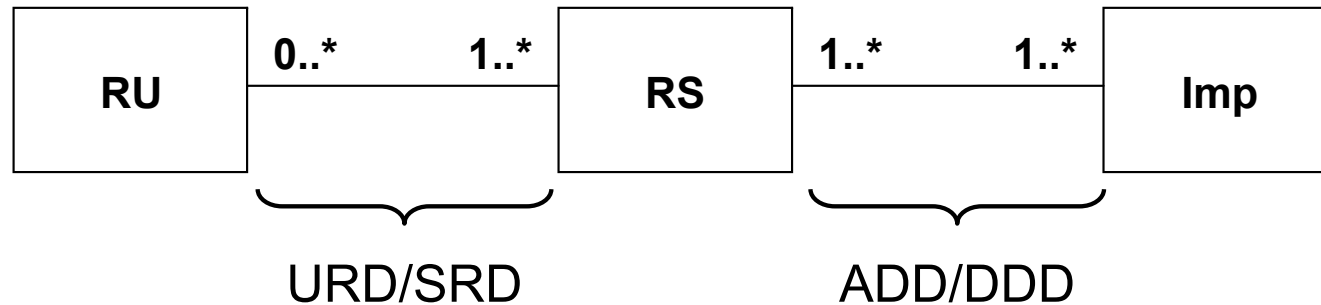
# Diferentes puntos de vista – Diferentes documentos



## Requisitos no funcionales

- Consumo de recursos
  - memoria, capacidad de tráfico...
- Rendimiento
  - velocidad, tiempo de respuesta...
- Fiabilidad y disponibilidad
  - cuantificación de fallos permitidos
- Manejo de errores
  - errores del entorno, errores internos
- Requisitos de interfaz
  - comunicación con usuarios, con otras aplicaciones
- Restricciones
  - exactitud, lenguajes y plataformas, arquitectura, estándares...
- Seguridad
  - seguridad del sistema (*security*), seguridad de las personas (*safety*)

## Matrices de trazabilidad



Matriz de doble entrada “dispersa”

	RU1	RU2	RU3
RS1	X		
RS2	X		X
RS3		X	
RS4			X
RS5			X

Matrices de tres columnas (una o las dos)

RU	RS	Descripción
1	1,2	
2	3	
3	2,4,5	

RS	RU	Título
1	1	
2	1,3	
3	2	
4	3	
5	3	

## Propiedades y atributos de los requisitos del software

- Propiedades deseables globales
  - Completitud
    - organización por tipos de requisitos
  - Consistencia
    - matriz de referencias cruzadas
- Propiedades deseables individuales
  - Tamaño adecuado
  - Claridad
  - Comprobabilidad: validación y verificación
  - Condiciones de error
- Atributos definibles
  - Automáticos: identificador, creador, fecha de creación...
  - Obligatorios: tipo, estado, descripción breve.
  - Opcionales: descripción detallada, fuente, necesidad, prioridad, estabilidad, complejidad, riesgo, coste...

## Consistencia: conflictos, acoplamientos y redundancias

	R1	R2	R3	R4	R5	R6	R7
R1			+		x	x	
R2							
R3	+			+		+	
R4			+		x		o
R5	x			x			
R6	x		+				
R7				o			

Conflicto (x)	Acoplamiento (+)	Redundancia (o)	Independencia
R1 y R5 R1 y R6 R4 y R5	R1 y R3 R3 y R4 R3 y R6	R4 y R7 ( $\rightarrow R7 \times R5, R7 + R3$ )	R2

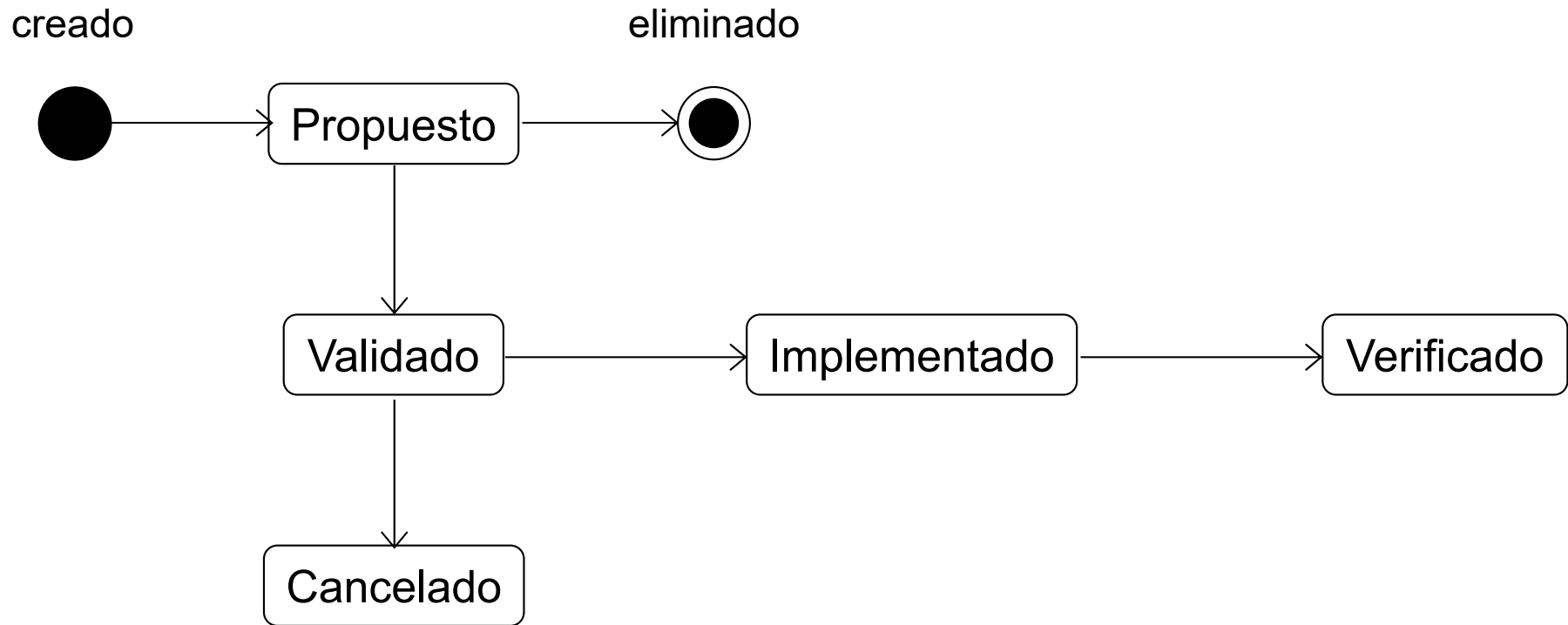
## Plantilla de requisitos (ejemplo)

ID	007
Título	Usuarios Visitantes
Creado por	Ana García (analista)
Fecha	05-09-2011
Fuente	Juan Gómez (responsable TIC)
Tipo	Funcional
Necesidad	Esencial
Estado	Propuesto
Descripción	Los usuarios Visitantes no son usuarios registrados del sistema: pueden acceder libremente a las funciones del rol Visitante sin necesidad de registrarse ni de iniciar sesión, y sin límite de sesiones simultáneas.
Pruebas	<ol style="list-style-type: none"><li>1. Comprobar que para todas las funciones del rol Visitante el sistema no requiere inicio de sesión.</li><li>2. Comprobar que el número de sesiones simultáneas de usuarios Visitantes es ilimitado.</li></ol>

## Métodos para organizar los requisitos del software

- Técnicas ya mencionadas
  - Matrices de trazabilidad
  - Matriz de consistencia: conflicto, acoplamiento, redundancia
  - Modularidad y anidamiento de requisitos: áreas temáticas
- Organizar los requisitos según el modelo del sistema
  - Según el modelo de casos de uso
    - Clases mencionadas
    - Operaciones utilizadas
  - Según el modelo conceptual (clases)
    - Atributos
    - Operaciones
  - Es esencial garantizar la coherencia entre el modelo y los requisitos
- Ciclo de vida de los requisitos
- Uso de herramientas para analizar y organizar requisitos

# Ciclo de vida de los requisitos





## Características de una herramienta de gestión de requisitos (1)

- Herramienta multiproyecto y multiusuario
  - Gestión de usuarios: analistas, jefes de proyecto, administradores.
  - Permisos de acceso por proyecto: lectura o escritura.
- Configuración de un proyecto
  - Propiedades globales de un proyecto: nombre, descripción, ubicación...
  - Atributos habilitados en función del tipo de requisito, valores despleables.
- Acceso, sesión y control de versiones
  - Registro automático de sesiones: inicio y fin, requisitos creados o modificados.
  - Control de versiones: sin control, control opcional, control obligatorio.
  - Notificación de modificaciones (opcional).
  - Seguimiento del ciclo de vida.
- Propiedades de los requisitos
  - Agrupación en paquetes y subpaquetes.
  - Atributos: automáticos, obligatorios, opcionales.
  - Relaciones entre requisitos: navegabilidad y asimetría. Requisitos “sospechosos”.
  - Otros artefactos asociados: escenarios, modelos...

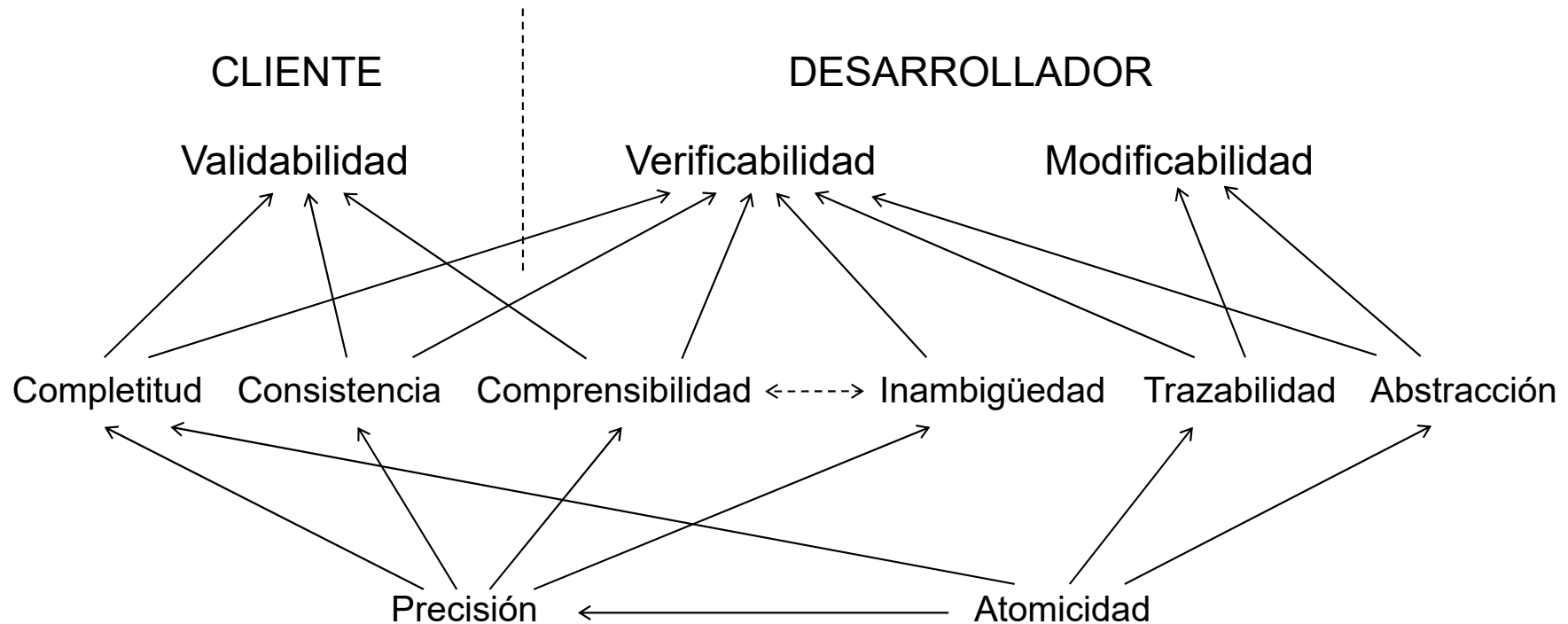
## Características de una herramienta de gestión de requisitos (2)

- Visualización, navegación y edición
  - Ficha o tabla, atributos visibles.
  - Ordenación y filtrado.
  - Búsqueda y reemplazamiento.
  - Movilidad entre paquetes.
  - Duplicación de requisitos.
- Informes
  - Listado de requisitos: ordenación, filtrado, atributos mostrados...
  - Estadísticas varias: ritmo de creación/modificación, conflictos...
  - Matrices de trazabilidad y consistencia.
- Utilidades
  - Asistencia en la creación del glosario de términos.
  - Coherencia entre los requisitos y los elementos del modelo.
  - Detección automática de solapamientos entre requisitos.
  - Métricas de calidad.

## Métricas de calidad en los requisitos del software

- Qué sentido tiene realizar medidas de calidad
  - Buscar la calidad desde el principio
  - Personal que las realiza
  - Coste añadido por la medición
- Métricas de calidad: cómo de bien están escritos los requisitos
  - ¿Qué debemos medir?
    - Propiedades deseables (cualitativas)
    - Indicadores medibles (cuantitativos)
  - ¿Cómo podemos medir?
    - Medidas manuales (inspección con ayuda de cuestionarios)
    - Medidas automáticas (características lingüísticas)
- Métricas de calidad: cómo de efectivos son los procesos
  - medidas de la efectividad de la inspección de requisitos
  - medidas de la efectividad del proceso de análisis de requisitos

# Propiedades e indicadores de calidad



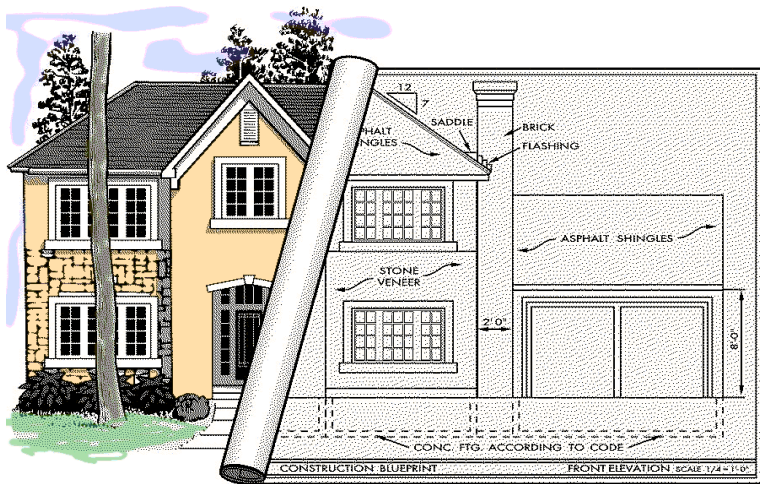
Morfológicos	Tamaño, Legibilidad, Puntuación, Acrónimos, Abreviaturas
Léxicos	Términos negativos, conectivos, de flujo, anafóricos, ambiguos, incompletos, especulativos, de diseño
Analíticos	Ortografía, gramática, verbos imperativos, verbos condicionales, voz pasiva, términos del dominio
Relacionales	Número de versiones, grado de anidamiento, dependencias, solapamientos

# Modelado conceptual con UML

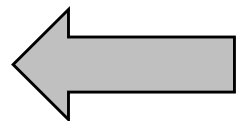
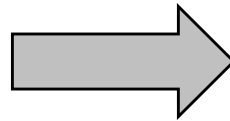
- La necesidad de modelar
- Qué significa “modelo conceptual”
  - Es una vista gráfica de la información contenida en los requisitos.
- Objetos y clases
  - Notación básica de objetos y clases
  - Tipos de clases
- Atributos
- Operaciones
- Asociaciones
  - Propiedades básicas: nombres, multiplicidad, navegabilidad
  - Relación con otros elementos: atributos, operaciones
- Generalizaciones
  - Generalización y clasificación
  - Jerarquías de clases
- Diagramas de clases y de objetos
- Asociaciones especiales

## Analogía arquitectónica

- ¿Tiene sentido poner ladrillos sin hacer antes los planos?
- El modelo -los planos- ayuda a afrontar la **complejidad del proyecto**
- ¿Cuál es el **lenguaje adecuado** para representar los planos?
- Ingeniería directa e ingeniería inversa: una casa, un coche, un virus...



Ingeniería  
directa



Ingeniería  
inversa



## Comunicación y representación del conocimiento



- Para representar el conocimiento hace falta un **lenguaje adecuado**
- El conocimiento bien representado ayuda a hacerse las preguntas oportunas: ¿qué falta aquí? ¿qué pasaría si...? ¿por qué no se puede...?