



# Tema 6: Validación de Requisitos

## Ingeniería de Requisitos

Raquel Martínez España

Grado en Ingeniería Informática



# Índice

---

1. Validación de requisitos
2. Técnicas de validación
  1. Revisiones
  2. Prototipado
  3. Generación de casos de prueba
  4. Otras técnicas



# Objetivos

---

- Entender la necesidad de la validación de requisitos para obtener un documento de especificación que define lo que el usuario desea.
- Comprender por qué las técnicas de validación de requisitos evitan los altos costos que significaría el tener que corregir una vez avanzado el desarrollo.
- Conocer el uso de diferentes técnicas de validación.



# Índice

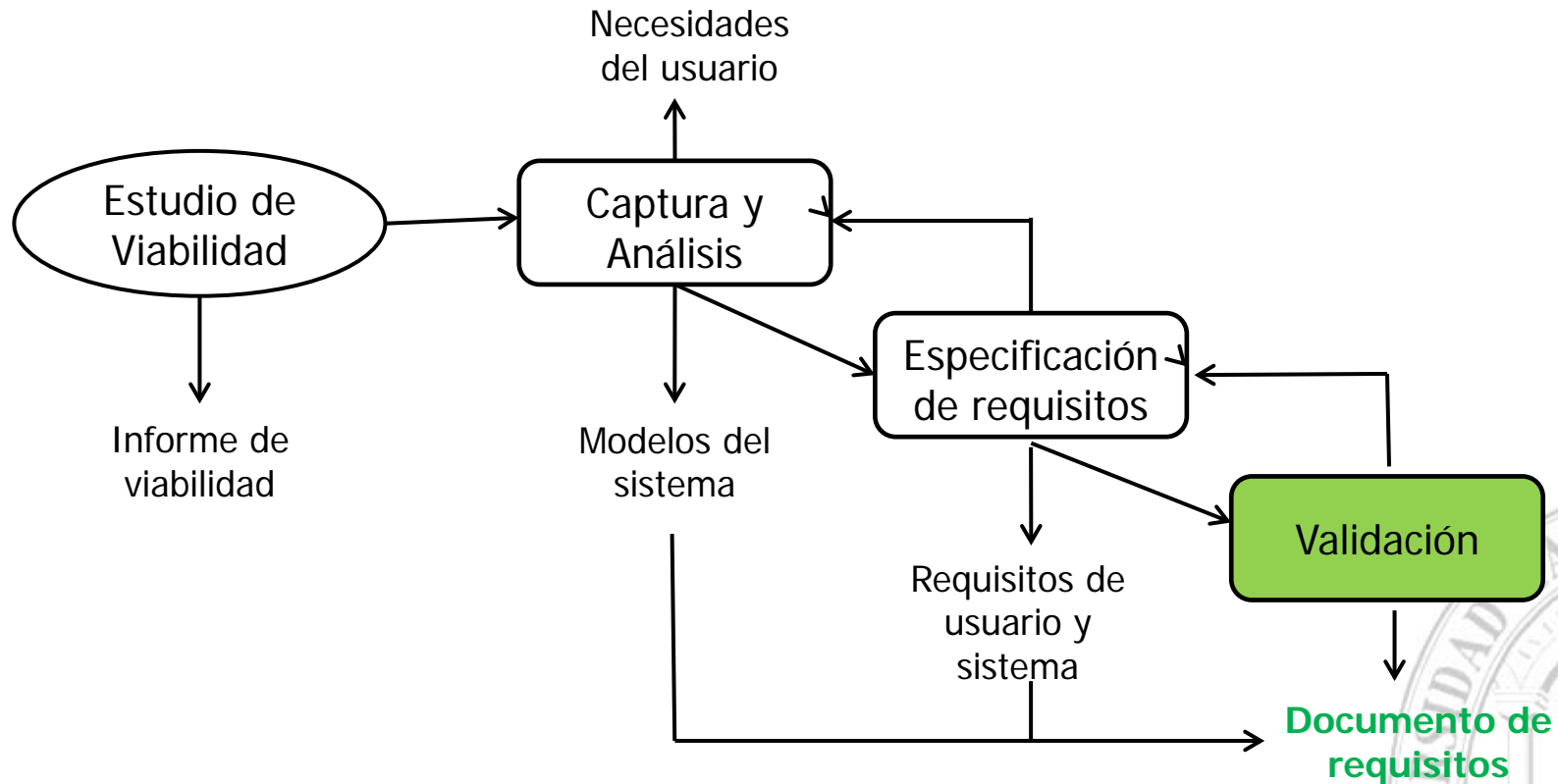
---

1. Validación de requisitos
2. Técnicas de validación
  1. Revisiones
  2. Prototipado
  3. Generación de casos de prueba
  4. Otras técnicas



# Validación de Requisitos

- Se trata de asegurar que el documento ERS define el software adecuado, es decir, el que espera el usuario.



# Validación de Requisitos

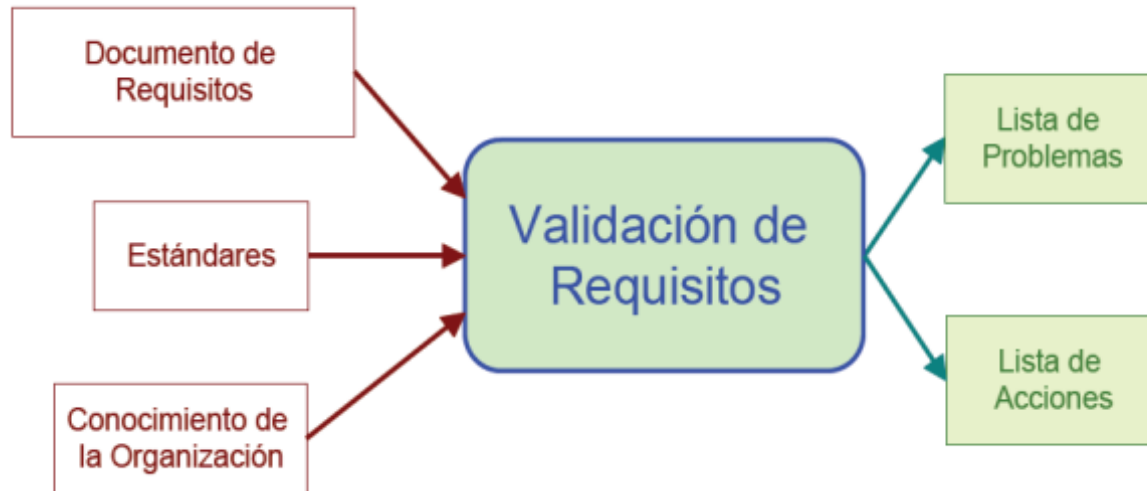
---

- Finalidad → comprobar que los requisitos poseen todos los atributos de calidad vistos:
  - **Validez**. El sistema provee las funciones que soportan mejor las necesidades del cliente.
  - **Consistencia**. No existen conflictos entre los requisitos.
  - **Compleitud**. Están incluidas todas las funciones requeridas por el usuario.
  - **Realismo**. Los requisitos pueden ser implementados con el presupuesto y tecnologías disponibles.
  - **Verificable**. Los requisitos pueden ser verificados.



# Validación de Requisitos

---



- El proceso de validación de requisitos comprende actividades que generalmente se realizan una vez obtenida una **primera versión de la documentación de requisitos**



# Índice

---

1. Validación de requisitos
2. Técnicas de validación
  1. Revisiones
  2. Prototipado
  3. Generación de casos de prueba
  4. Otras técnicas





# Técnicas de Validación

---

- Más habituales:
  - **Revisiones:** reuniones donde un equipo de analistas intenta localizar errores en el documento de especificación
  - **Prototipado:** construir una maqueta del futuro sistema software a partir de los requisitos recogidos en la especificación. Esta maqueta será evaluada por el cliente y usuarios para comprobar su corrección y completitud.
  - **Generación de casos de prueba (test de requisitos):** consiste en la definición de casos de prueba que permitan verificar el cumplimiento de los requisitos funcionales.
- Otras:
  - **Animación y validación de modelos o especificaciones formales**
  - **Manuales de usuario**



# Revisiones

---

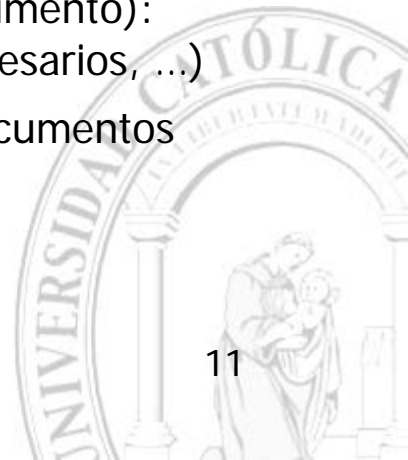
- Las revisiones de requisitos permiten:
  - Descubrir una gran cantidad de defectos en los requisitos.
  - Reducir los costos de desarrollo entre un 25% y un 30%.
  - Reducir el tiempo de pruebas entre un 50% y un 90%
- Consisten en:
  - Una o varias **reuniones planificadas**, donde se intenta confirmar que los requisitos poseen los atributos de calidad deseados.
  - **Analista encargado del proyecto y un conjunto de colegas** que, preferiblemente, *no están relacionados con el proyecto* y, además, son competentes en la actividad de requisitos.
  - El **resultado**: documento que contiene la lista de defectos localizados y *una lista de acciones recomendadas*.



# Revisiones

---

- Las reuniones de revisión:
  - 1. Preparar el plan de la revisión.**
    - Planificación temporal y participantes de la revisión.
  - 2. Distribuir los documentos a revisar.**
    - Documento de especificación.
    - Aquellos documentos que ayuden a comprender adecuadamente el documento de especificación (documentos referenciados y los anexos a la especificación)
  - 3. Preparar la reunión.**
    - Para el analista promotor de la reunión (encargado del documento): logística (reservar sala, solicitar los materiales que sean necesarios, ...)
    - Para los analistas participantes: leer cuidadosamente los documentos recibidos y anotar defectos.



# Revisiones

---

## 4. Realizar la reunión de revisión.

- El formato de esta reunión puede ser muy diverso.

## 5. Identificar los defectos y acciones a realizar.

- La lista de defectos y acciones recomendadas es el documento final obtenido en las revisiones de requisitos → *Formato visto anteriormente*

## 6. Realizar las correcciones que sean precisas a los documentos revisados.

- El analista promotor de la reunión debe evaluar y, si lo estima conveniente, llevar a cabo, las acciones recomendadas que han surgido de la reunión de revisión.

## 7. Informar de las modificaciones realizadas a los participantes en la reunión.

- Enviar un breve informe de las tareas realizadas, y una copia corregida de los documentos de especificación, a los participantes en la reunión para su visto bueno



# Revisiones

---

- Preparación y reunión:
  - Uso de **checklists de validación** (similar al análisis)
  - Es aconsejable que cada organización desarrolle y mantenga checklists propios y adaptados a los sistemas que desarrolla.
- **Pre-revisiones:** revisión rápida e informal de los documentos de especificación.
  - Identificar errores sencillos (desviaciones de los estándares, omisiones, etc.) que pueden detectarse fácilmente sin necesitar de un proceso tan largo como son las revisiones.
  - Permitir que los participantes se concentren en los defectos difíciles de identificar.
  - Disminuir el número de errores y acciones a realizar



# Prototipado

---

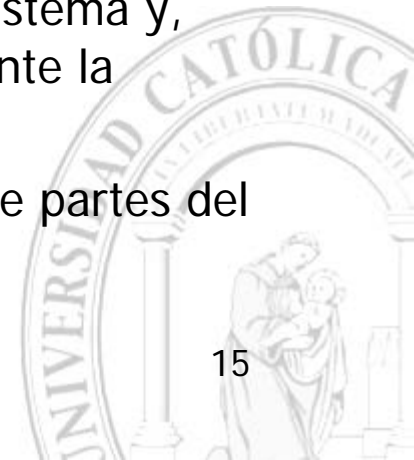
- Usado en diversas disciplinas.
- Consiste en la creación de una maqueta o versión del producto final.
- Actividad de requisitos:
  - Los prototipos se utilizan, fundamentalmente, para comprobar la corrección y completitud de la especificación de requisitos.



# Prototipado

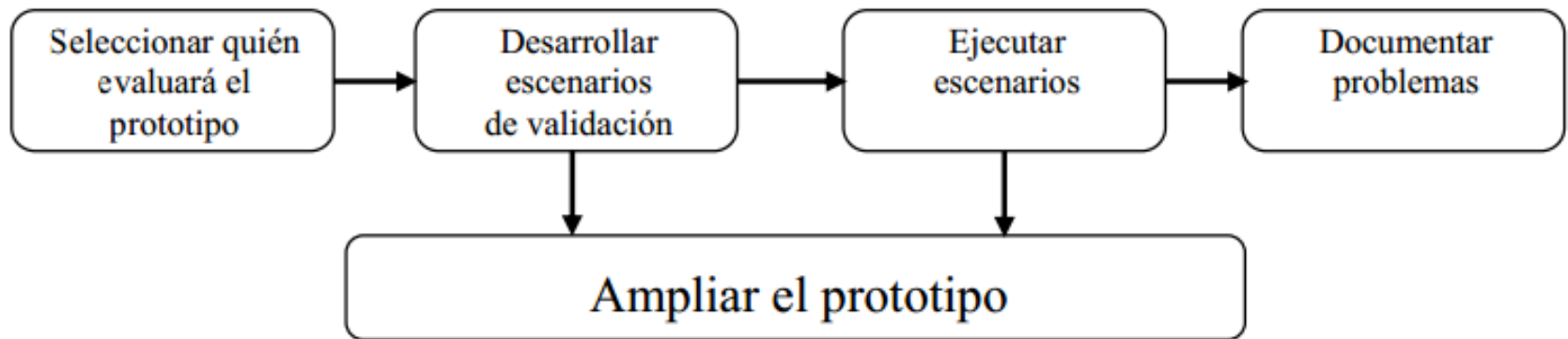
---

- En IR, los prototipos más comunes son los siguientes:
  - **Mock-ups:** pantallas, típicamente dibujadas a mano, que representan un aspecto concreto del sistema:  
Ej: <http://www.youtube.com/watch?v=-QVKgcmBwF0>
  - **Storyboards:** son una evolución de los mock-ups, ya que además del interfaz, se muestra la secuencia de acciones, o escenarios, que se deben realizar con el programa.
  - **Maquetas:**
    - Versión simplificada del sistema software deseado.
    - Típicamente, representa únicamente el interfaz del sistema y, opcionalmente, las conexiones entre pantallas mediante la utilización de elementos activos como los botones.
    - Si fuera necesaria mayor fidelidad, podrían codificarse partes del sistema → “prototipo funcional”.



# Prototipado

---



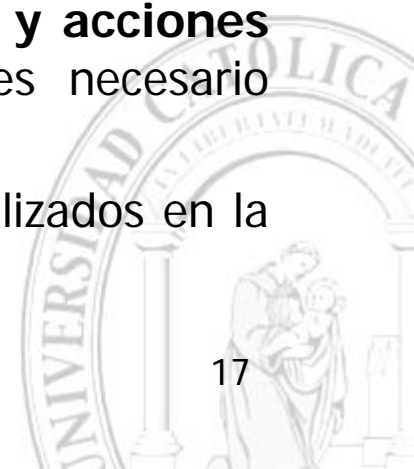
1. **Seleccionar quién evaluará el prototipo:** conjunto de usuarios representativos de los distintos perfiles
2. **Desarrollar escenarios de validación:** identificar escenarios o tareas que deben realizar los usuarios (a partir del detalle de la especificación o manual de usuario)
3. **Ejecutar escenarios:** el usuario realiza o ejecuta los escenarios previstos. El analista debe suplir la funcionalidad no presente en el prototipo.
4. **Documentar problemas:** confeccionar la lista de problemas encontrados.



# Generación de casos de prueba

---

- El único método existente para comprobar la **verificabilidad** de un requisito es que sea posible definir uno o varios casos de prueba para dicho requisito.
- **Caso de prueba:** descripción de una acción bien definida que se debe realizar con el software → perfectamente descritos los *datos de entrada*, las *tareas* a realizar y los *resultados* esperados.
- Los casos de prueba produce dos resultados:
  - **Lista de errores** (que el requisito X no es verificable) **y acciones recomendadas** (los aspectos del requisito X que es necesario completar: datos de entrada, datos de salida, etc.).
  - Un **conjunto de casos de prueba** que pueden ser utilizados en la posterior prueba de sistema y aceptación.

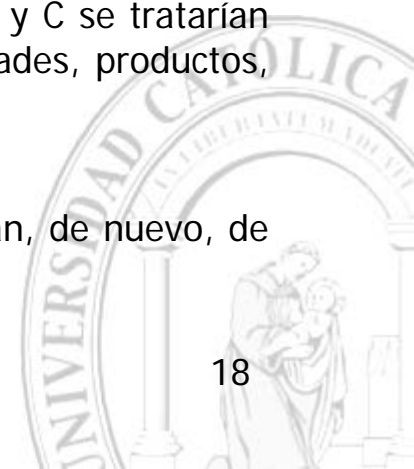


# Generación de casos de prueba

---

## Ejemplo:

- **REQ1. El sistema deberá generar facturas** → Incompleto:
  - No se indica cómo se deben generar las facturas (¿una única factura, una factura por cliente, una factura por pedido, etc.?)
  - No se indica qué se factura (¿un pedido específico, los pedidos entre dos fechas, etc.?)
- REQ1. El sistema deberá generar facturas
  - REQ1.1. Se generará una factura por cliente. Esta factura incluirá todos los pedidos no facturados del cliente en cuestión.
  - REQ1.2. El formato de la factura será el especificado en el anexo 1
- **Caso de prueba:**
  - *Se introducirán en el sistema los pedidos A, B, C, ...* Estos pedidos A, B y C se tratarían de pedidos bien definidos, indicando todos los datos necesarios (cantidades, productos, etc.)
  - *Se invocará la función de generar factura*
  - *Se deberán obtener las facturas X, Y, ...* Estas facturas X e Y se tratarían, de nuevo, de facturas bien definidas



# Otras técnicas de validación

---

- **Manuales de usuario:**
  - Verificar si la especificación de requisitos contiene el suficiente detalle como para preparar el manual de usuario.
  - De no ser así, podría ocurrir que la especificación fuera incompleta.
  - El detalle con el que puede confeccionarse el manual de usuario es variable:
    - Depende del grado de detalle del documento de especificación → si el documento de especificación es muy detallado, también lo será el manual, y viceversa
  - Puede usarse conjuntamente con el prototipado.



# Otras técnicas de validación

---

- Animación y validación de modelos o especificaciones formales:
  - Cuando se han utilizado **modelos** durante el análisis o métodos de **especificación formal** durante la documentación.
  - **Animación de modelos/especificaciones formales**, con el objetivo de verificar que el funcionamiento del sistema, tal y como está especificado, es el deseado por cliente y usuarios.
  - Algunos modelos y todos los tipos de especificaciones formales admiten la posibilidad de validación de los requisitos mediante **demostraciones formales**.



# Puntos clave

---

- El proceso de validación de requisitos permite obtener un documento de especificación de requisitos consistente, completo, preciso, realista, verificable y que define lo que el usuario desea del producto final.
- Las técnicas de validación más habituales son las revisiones, prototipado y generación de casos de prueba.
- Las revisiones de requisitos consisten en reuniones donde un equipo de analistas intenta localizar errores en el documento de especificación.
- El método del prototipado consiste en construir una maqueta del futuro sistema software a partir de los requisitos recogidos en la especificación. Esta maqueta será evaluada por el cliente y usuarios para comprobar su corrección y completitud.
- La generación de casos de prueba (test de requisitos) tiene como objetivo comprobar la verificabilidad de los requisitos

