TEMA 4. Análisis de requisitos

Ingeniería del Software

Raquel Martínez España

Grado en Ingeniería Informática

Raquel Martínez España- Tlf: (+34) 968 27 88 21
Universidad Católica San Antonio de Murcia - Tlf: (+34) 968 27 88 00
info@ucam.edu - www.ucam.edu

Análisis de requisitos

- **Especificación de requisitos: Generación ERS**
- Modelado de requisitos del sistema: Escenarios/casos de uso
- Elaboración del modelo de requisitos
- Validación de los requisitos
- Análisis de requisitos de la práctica

Análisis de requisitos - Pasos

- Concepción
- Indagación
- Elaboración
- Negociación

Tema 3 (Análisis previo)

- Especificación
- Validación

Tema 4

- Gestión de los requisitos

Análisis de requisitos

Análisis de requisitos - Concepto

- La ingeniería de requisitos ofrece los mecanismos para:
  - ✓Entender lo que desea el cliente
  - ✓Analizar las necesidades
  - ✓Evaluar la factibilidad
  - ✓Negociar una solución razonable
  - ✓Especificar la solución sin ambigüedades
  - ✓Validar la especificación
  - ✓Gestionar los requisitos a medida que surgen, desaparecen o se modifican.

Análisis de requisitos - Concepto

- **Objetivo:** producir un documento de especificación de requisitos que describa qué tiene que hacer el sistema pero no cómo.
  - No sólo análisis, también síntesis.
- **Análisis de requisitos (IEEE):** El proceso de estudio de las necesidades de los usuarios para llegar a una definición de los requisitos del sistema, de hardware o de software, así como el proceso de estudio y refinamiento.
- **Requisito (IEEE):** Condición o capacidad que necesita el usuario para resolver un problema o conseguir un objetivo determinado.
- Definición de requisitos (Especificación de Requisitos Software, ERS) debe ser fruto del trabajo conjunto: clientes y desarrolladores.

Análisis de requisitos – Tres grandes actividades (1)

- **Definir los requisitos software:**
  - Tarea iterativa para crear especificación preliminar de requisitos que debe cumplir el software.
  - Utilizar técnicas de extracción de requisitos.
- **Definir los requisitos de las interfaces:**
  - Definir la interacción con otros elementos del sistema como el usuario, el hardware u otras aplicaciones.
  - Interacción con el usuario es crítica para la facilidad de uso.
- **Integrar los requisitos en un documento y asignarles prioridades:**
  - Contrato entre el usuario y el cliente.
  - Prioridades para saber relevancia de los mismos.
  - **Resultado:** Documento de Especificación de Requisitos software**.**

Análisis de requisitos – cuatro grandes actividades (2)

1. **Extracción o determinación de requisitos:** proceso mediante el cual los clientes del software descubren, revelan, articulan y comprenden los requisitos que desean.
2. **Análisis de requisitos:** Proceso de razonamiento sobre (1), detectando y resolviendo posibles inconsistencias o conflictos.
3. **Especificación de requisitos:** proceso de redacción o registro.
4. **Validación de requisitos:** proceso de confirmación y verificación.

Tipos de requisitos

1. **Requisitos funcionales:** expresan algún aspecto relativo al comportamiento del sistema -> Tarea a realizar
2. **Requisitos no funcionales:** expresan propiedades del sistema, entre ellos:
   - **2.1 Restricciones:** Describen los límites del sistema.
   - **2.2 De funcionamiento:** recogen especificaciones de diseño, a nivel del sistema, de hardware o de software.
   - **2.3 Manejo de excepciones:** Describen los posibles comportamientos anormales del sistema y sus tratamientos correspondientes.

Ejemplos de tipos de requisitos

1. El día 26 de cada mes deben generarse las nóminas de los empleados de la empresa.
2. Cuando el número de existencias en stock sea menor de 5 se debe realizar un pedido a la empresa proveedora.
3. La edad de un empleado debe ser siempre mayor de 16 años.
4. La aplicación se ejecutará sobre un i7, utilizando el sistema operativo linux.
5. Como herramienta para desarrollar el análisis se va a utilizar Easy Case.
6. Cuando se produzca un error al introducir un dato, se debe mostrar un mensaje en la pantalla que proporcione información, lo más detallada posible, respecto a la causa que lo origino.

Especificación de Requisitos del software (ERS)

- Documentación de los requisitos esenciales (funcionales, rendimiento, diseño, restricciones y atributos) del software y sus interfaces externas -> Objetivo final de la etapa.
- 2 Características fundamentales:
  - Debe contener información con las necesidades reales del cliente.
  - Debe comunicar dicha información de forma eficaz.
- Qué hay que desarrollar, no cómo ni cuándo:
  - Sólo requisitos del software necesarios (Dejar a un lado futuribles).
  - No hay detalles del diseño, de su verificación o dirección del proyecto, excepto las restricciones (pej. Limitación hardware disponible) .

Características de una buena ERS

- ✓ **No ambigua:** Un requisito solo puede tener una interpretación (OJO: Lenguaje natural).
- ✓ **Completa:** Es completa si

1. Incluye todos los requisitos significativos del software:
   - Funcionalidad
   - Ejecución
   - Restricciones
   - Atributos de calidad
   - Interfaces externa
2. Define la respuesta ante cualquier entrada (correcta o no)
3. Cumple con los estándares de especificación a aplicar (pej IEEE 830).
4. Están etiquetadas y referenciadas todas las figuras, tablas y diagramas.
5. Están definidos todos los términos y unidades de medida -> ACRÓNIMOS

- ❑ Si no es completa indicar TBD (To Be Determined) y por qué

Características de una buena ERS

- ✓ **Fácil de verificar:**
  - ❑ Requisitos difícil de verificar: **"El programa no debe entrar nunca en un bucle infinito"**
  - ❑ Tenemos que dar métodos para verificar requisitos, en caso contrario, eliminar del ERS.
- ✓ **Consistente:** No hay conflictos entre los requisitos. Tres posibles de conflictos:
  - ❑ Describir el mismo requisitos varias veces.
  - ❑ Contradictorios.
  - ❑ Lógico o temporal: Un requisito sumar y otro multiplicar.
- ✓ **Fácil de modificar** -> Debe estar bien estructurado (tabla de contenidos).
  - ❑ Si hay requisitos relacionados, enlazarlos.

Características de una buena ERS

- ✓Facilidad para identificar el origen y las consecuencias de cada requisito (Trazabilidad).
  - ❑Relación entre dos o más productos del ciclo de vida.
  - ❑Establecer el origen claro y referencias en desarrollos futuros.
  1. Referencia hacía atrás: A documentos previos al desarrollo.
  2. Referencia hacía adelante: A documentos posteriores del desarrollo.
- ✓Facilidad de utilización durante la fase de explotación y de mantenimiento.
  - ❑Personal que no ha desarrollado el software se encarga del mantenimiento y puede provocar cambios en la documentación.

Evolución de la ERS

- ✓Imposible especificar todos los detalles -> por ejemplo detalles de todas las pantallas.
- ✓Cambios por deficiencias encontradas.
- ✓**Objetivo:** Intentar especificar los requisitos de la manera mas detallada.
- ✓Desarrollar un proceso formal de cambio para identificar, controlar, seguir e informar de cambios.
  - ❑Generar nuevas versiones de la ERS.

Estructura para la ERS (IEEE 830)

1. Introducción
   - 1.1. Objetivo
   - 1.2. Ámbito
   - 1.3. Definiciones, siglas y abreviaturas
   - 1.4. Referencias
   - 1.5. Visión global
2. Descripción general
   - 1.1. Perspectiva del producto
   - 1.2. Funciones del producto
   - 1.3. Características del usuario
   - 1.4. Limitaciones generales
   - 1.5. Supuestos y dependencias

3. Requisitos específicos

Apéndices

Índice

3. Requisitos específicos
    3.1 Requisitos funcionales
        3.1.1 Requisitos funcional 1
            3.1.1.1 Introducción
            3.1.1.2 Entradas
            3.1.1.3 Procesamiento
            3.1.1.4 Salidas
        3.1.2 Requisito funcional 2
            ……..
        3.1.n Requisito funcional n
    3.2 Requisitos de interfaz externa
        3.2.1 Interfaces de usuario
        3.2.2 Interfaces hardware
        3.2.3 Interfaces software
        3.2.4 Interfaces de comunicación
    3.3 Requisitos de ejecución
    3.4 Restricciones de diseño
        3.4.1 Acatamiento de estándares
        3.4.2 Limitaciones de hardware
        …..
    3.5 Atributos de calidad
        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

Análisis de requisitos

- Especificación de requisitos: Generación ERS
- **Modelado de requisitos del sistema: Escenarios/casos de uso**
- Elaboración del modelo de requisitos
- Validación de los requisitos
- Análisis de requisitos de la práctica

Escenarios/casos de uso

"*un caso de uso capta un contrato que describe el comportamiento del sistema en distintas condiciones en las que el sistema responde a una petición de alguno de sus participantes*" (Alistair Cockburn, 2001)

¿Qué se necesita saber a fin de desarrollar un caso de uso eficaz?

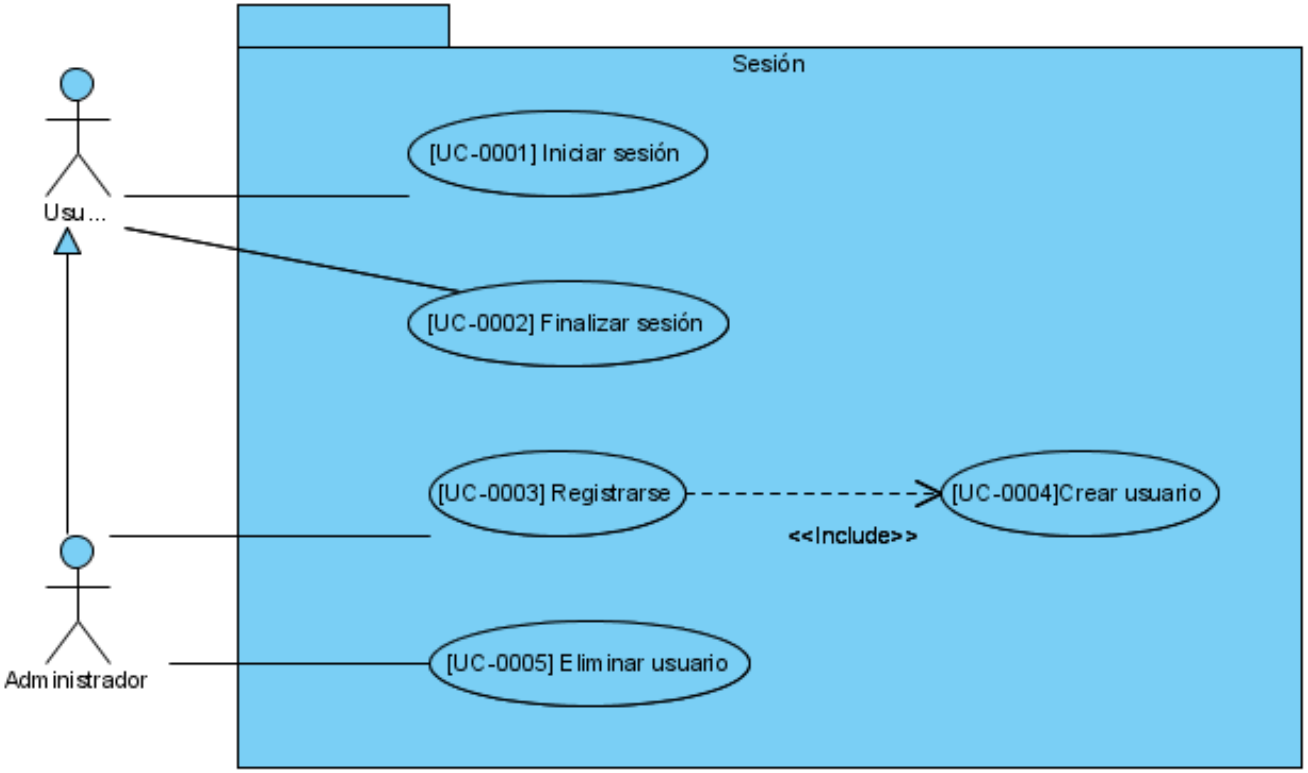
- ¿Quién es el actor principal y quién son los secundarios?
- ¿Cuáles son los objetivos de los actores?
- ¿Qué precondiciones deben existir antes de empezar la historia?
- ¿Qué tareas principales son realizadas por el actor?
- ¿Qué excepciones deben ser consideradas al describir la historia?
- ¿Qué variaciones son posibles en la interacción del actor?
- ¿Qué información del sistema adquiere, produce o cambia el actor?
- ¿Tendrá que informar el actor al sistema acerca de cambios en el ambiente externo?
- ¿Qué información desea obtener el actor del sistema?
- ¿Quiere el actor ser informado sobre cambios inesperados?

Escenarios/casos de uso

Define:

- Actores (actor ≠ usuario) – distintas personas o dispositivos que usan el sistema
- Objetivo del caso de uso
- Precondiciones
- Disparador
- Escenario – qué acción realiza cada actor en el sistema
- Excepciones
- Prioridad
- Cuándo estará disponible
- Frecuencia de uso
- Canal (interfaz) para el actor
- Aspectos pendientes

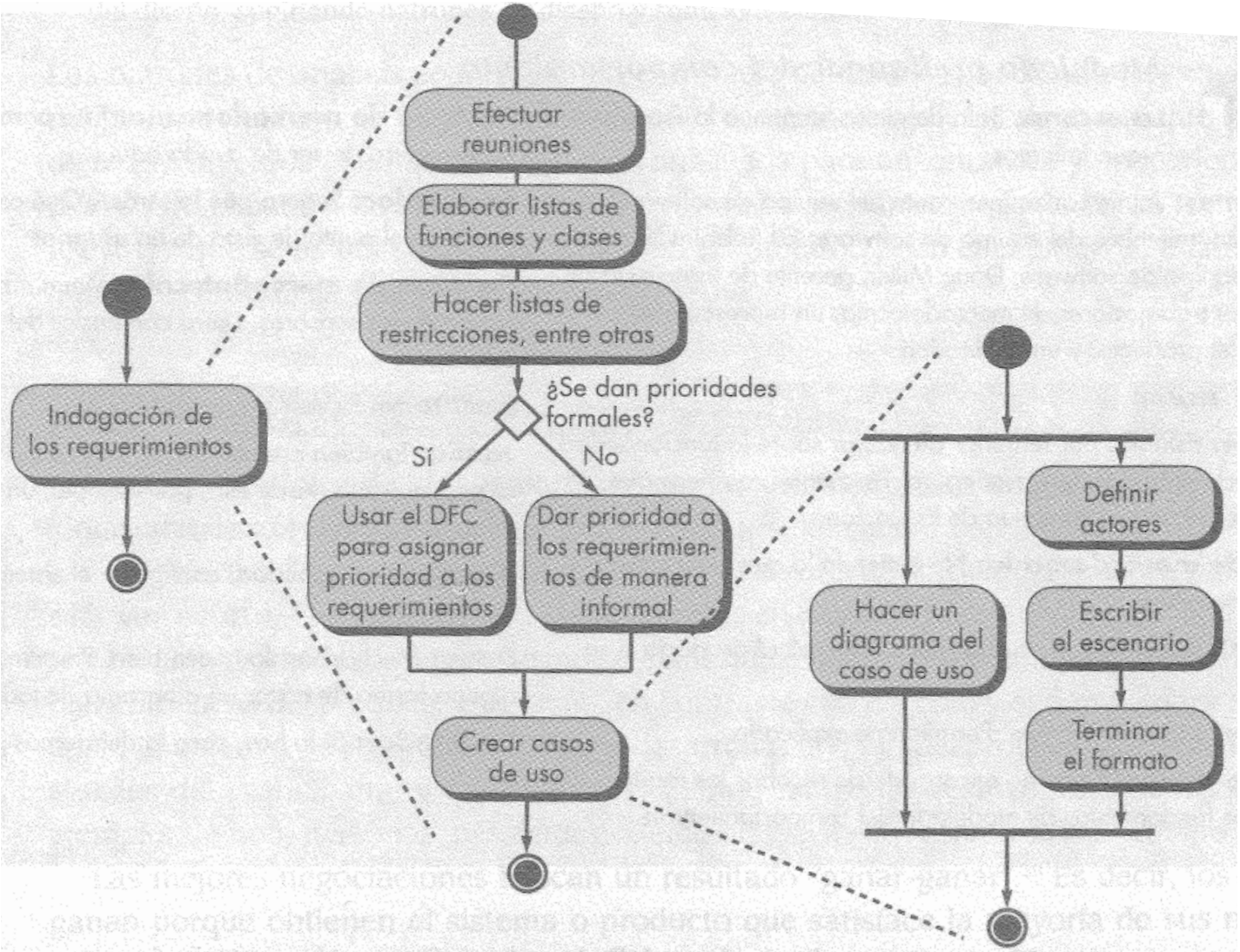
Escenarios/casos de uso



Análisis de requisitos

- Especificación de requisitos: Generación ERS
- Modelado de requisitos del sistema: Escenarios/casos de uso
- **Elaboración del modelo de requisitos**
- Validación de los requisitos
- Análisis de requisitos de la práctica

Proceso de definición de los requisitos



Elaboración del modelo de requisitos

- A medida que se avanza, los modelos se hacen más estables.
- ¿Utilizar un modo de representación o varios?

Diagrama de casos de uso

Diagrama de clases

- **Tipos de elementos** presentes en todos los modelos:
  - ✓ Basados en el escenario – desde el punto de vista del usuario, definición de casos de uso (actores, escenario, detalles).
  - ✓ Basados en clases – un caso de uso es un conjunto de objetos que manipula un actor cuando interactúa con el sistema (clase de objetos). Especifican herencia, colaboraciones, interacciones…
  - ✓ De comportamiento – representación de los estados del sistema y eventos que se producen
  - ✓ Orientados al flujo – estudia las relaciones de transformación de las entradas y salidas a lo largo de los procesos en el sistema.

Diagrama de estados

Diagrama de flujo de datos

Elaboración del modelo de requisitos

Diagrama de casos de uso

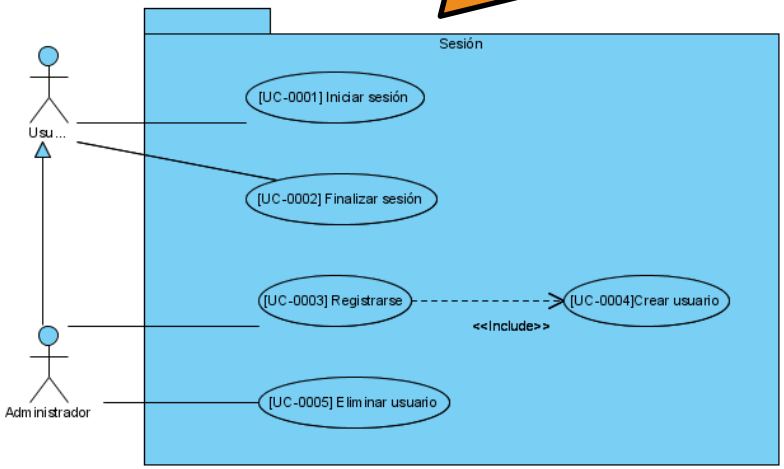


Diagrama de clases

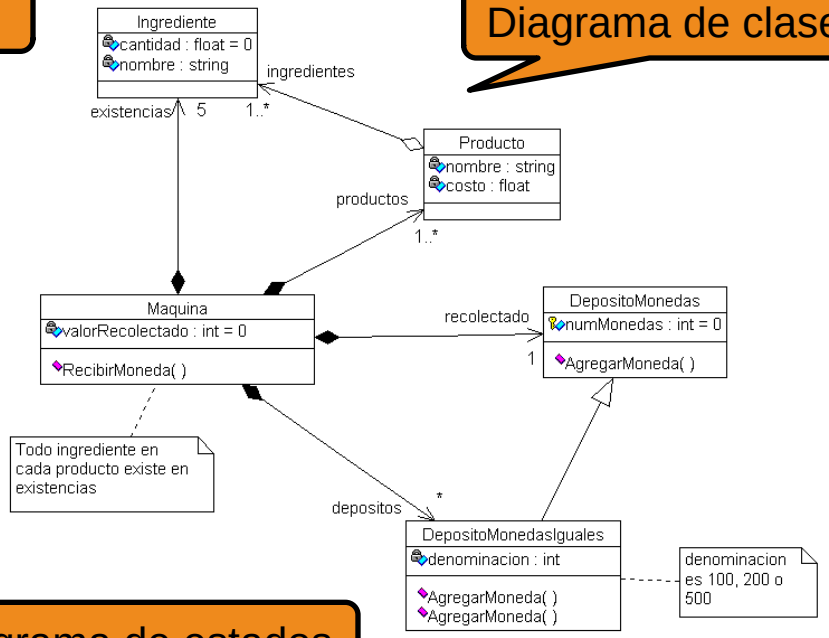
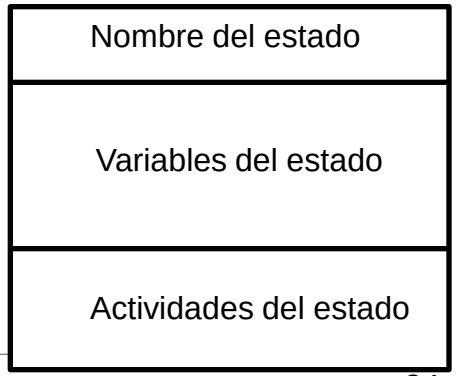


Diagrama de flujo de datos

Diagrama de estados



Elaboración del modelo de requisitos

- **Patrones de análisis**:
  - ✓ Responden a comportamientos recurrentes en un mismo dominio de aplicación.
  - ✓ Modelos de análisis reutilizables (descripción, ventajas, limitaciones, …)
  - ✓ Aceleran el desarrollo de los modelos de análisis abstractos que capturan los principales requerimientos del problema.
  - ✓ Deben estar bien identificados en la documentación.

Análisis de requisitos

- Especificación de requisitos: Generación ERS
- Escenarios/casos de uso
- Elaboración del modelo de requisitos
- **Validación de los requisitos**
- Análisis de requisitos de la práctica

Validación de los requisitos

- ¿Es coherente cada requerimiento con los objetivos generales del sistema o producto?
- ¿Se han especificado todos los requerimiento en el nivel apropiado de abstracción? ¿Algunos de ellos están detallados a un nivel que no corresponde con la etapa?
- ¿Cada requerimiento es realmente necesario o representa una característica agregada que tal vez no sea esencial para el objetivo del sistema?
- ¿Cada requerimiento esta acotado y no es ambiguo?
- ¿Existen requerimientos en conflicto?
- ¿Cada requerimiento es asequible en el ambiente técnico del sistema?
- ¿Puede probarse una vez implementada su solución?
- ¿Se refleja de manera adecuada la información, función y funcionamiento del sistema?
- ¿Se ha utilizado algún patrón de requerimientos para simplificar el modelado?

Bibliografía básica

- Pressman, R. "*Ingeniería del Software: Un enfoque práctico*". 7ª edición. Madrid: McGraw Hill, 2010.
  - ✓ Capítulo 5