



**DECSAI**

**Departamento de Ciencias de la Computación e I.A.**

Universidad de Granada



# Especificación de requerimientos

Diseño de bases de datos

## Documento de especificación del sistema

1. Definición del problema
2. **Descripción funcional**
3. **Restricciones**
4. Diagramas de flujo de datos
5. Modelo de datos
6. Diccionario de datos
7. **Casos de uso**
8. **Documentos adicionales**



# Especificación de requerimientos

- **Requerimientos**
  - Definición
  - Requerimientos funcionales y no funcionales
- **Especificación de requisitos en lenguaje natural**
- **Casos de uso**
- **Documento de especificación del sistema**



# Requerimientos



Los requerimientos/requisitos de un sistema describen los servicios que ha de ofrecer el sistema y las restricciones asociadas a su funcionamiento.

## **Requerimientos:**

Propiedades o restricciones determinadas de forma precisa que deben satisfacerse.



# Requerimientos funcionales y no funcionales



## **Requerimientos funcionales:**

Expresan la naturaleza del funcionamiento del sistema (cómo interacciona el sistema con su entorno y cuáles van a ser su estado y funcionamiento).

NOTA: A veces, también es conveniente indicar lo que **no** hará el sistema.



# Requerimientos funcionales y no funcionales



## **Requerimientos no funcionales:**

Restricciones sobre el espacio de posibles soluciones.

- **Rendimiento del sistema:**  
Fiabilidad, tiempo de respuesta, disponibilidad...
- **Interfaces:**  
Dispositivos de E/S, usabilidad, interoperabilidad...
- **Proceso de desarrollo:**  
Estándares, herramientas, plazo de entrega...



# Requerimientos funcionales y no funcionales



Los requisitos funcionales definen  
**qué debe** hacer un sistema.

Los requisitos no funcionales definen  
**cómo debe** ser el sistema.



# Requerimientos funcionales y no funcionales



A los requisitos no funcionales se les suele llamar coloquialmente “calidades” del sistema [“-ilities” en inglés] y pueden dividirse en dos categorías:

- **Calidades de ejecución**, como la seguridad o la usabilidad, observables en tiempo de ejecución.
- **Calidades de evolución**, como la “testabilidad”, mantenibilidad, extensibilidad o escalabilidad, determinadas por la estructura estática del software.



# Requerimientos funcionales y no funcionales



La distinción entre requerimientos funcionales y no funcionales no siempre resulta evidente.

Ejemplo: La seguridad puede interpretarse inicialmente como un requerimiento no funcional al principio. No obstante, su elaboración puede conducir a nuevos requerimientos funcionales, como la necesidad de autenticar a los usuarios del sistema.

Más allá de si decidimos incluir este tipo de requisitos en una sección u otra, lo importante es identificarlos correctamente.



# Especificación de requerimientos en lenguaje natural



## Los requerimientos...

- se suelen especificar en lenguaje natural,
- se expresan de forma individual (p.ej. esquemáticamente),
- se organizan de forma jerárquica (a distintos niveles de detalle),
- a menudo, se numeran (para facilitar su gestión),



# Especificación de requerimientos en lenguaje natural



## Los requerimientos han de ser...

- **claros y concretos**  
(evitando imprecisiones y ambigüedades)  
p.ej. Uso de puntos suspensivos, etcétera...
- **concisos**  
(sin rodeos ni figuras retóricas),
- **completos y consistentes,**



# Especificación de requerimientos en lenguaje natural



## Los requerimientos han de indicar...

- lo que se espera que haga el sistema (¿qué?),
- su **justificación** (¿por qué ha de ser así? ¿quién lo propuso?) y,
- en su caso, los **criterios de aceptación** que sean aplicables (¿cómo se verifica su cumplimiento?).



# Especificación de requerimientos en lenguaje natural



## Los **requerimientos funcionales**...

- deben estar redactados de tal forma que sean comprensibles para usuarios sin conocimientos técnicos avanzados (de Informática, se entiende),
- deben especificar el comportamiento externo del sistema y evitar, en la medida de lo posible, establecer características de su diseño,
- deben priorizarse (al menos, se ha de distinguir entre requisitos obligatorios y requisitos deseables).



# Especificación de requerimientos en lenguaje natural



## Los **requerimientos no funcionales**...

- han de especificarse cuantitativamente, siempre que sea posible (para que se pueda verificar su cumplimiento).



# Especificación de requerimientos en lenguaje natural



## **MAL**

Para facilitar el uso del editor gráfico, se podrá activar y desactivar una rejilla que permitirá alinear las figuras del diagrama. Cuando se ajuste la figura al tamaño de la pantalla, se reducirá el número de líneas de la rejilla para que no se dificulte la visualización del diagrama.

¿Por qué?

Amalgama de varios requisitos.



# Especificación de requerimientos en lenguaje natural



## BIEN

El editor permitirá el uso de una rejilla de líneas horizontales y verticales que aparecerán dibujadas tras el diagrama.

**Justificación:** La rejilla facilita la creación de diagramas cuidados en los que las figuras se puedan alinear con facilidad (*Manual Práctico de Usabilidad*, sección 15.3).

¿Por qué?

Preciso, conciso y justificado correctamente.



# Especificación de requerimientos en lenguaje natural



## **MAL**

- ✘ El sistema será lo más fácil de utilizar posible.
- ✘ El sistema proporcionará una respuesta rápida al usuario.
- ✘ El sistema se recuperará automáticamente tras producirse un fallo.

¿Por qué?

Objetivos generales, vagos  
y abiertos a distintas interpretaciones.



# Especificación de requerimientos en lenguaje natural



## BIEN

- ✓ Un usuario experimentado debe ser capaz de utilizar todas las funciones del sistema tras un entrenamiento de 2 horas, tras el cual no cometerá más de 3 errores diarios en media.
- ✓ Cuando haya hasta 100 usuarios accediendo simultáneamente al sistema, su tiempo de respuesta no será en ningún momento superior a 2 segundos.



# Especificación de requerimientos en lenguaje natural



## BIEN

- ✓ Ante un fallo en el software del sistema, no se tardará más de 5 minutos en restaurar los datos del sistema (en un estado válido) y volver a poner en marcha el sistema.

¿Por qué?

Requisitos verificables.



# Especificación de requerimientos en lenguaje natural



## PROBLEMAS HABITUALES:

- La existencia de un requerimiento ha de estar debidamente justificada (debemos saber por qué es un requisito del sistema).
- Un requerimiento es, a veces, difícil de verificar (especialmente, si es un requisito no funcional). Además, si somos incapaces de especificarlo, ¿cómo sabemos que realmente es un requisito?



# Especificación de requerimientos en lenguaje natural



## EJEMPLO: REQUERIMIENTOS FUNCIONALES

### Matriculación

- La matrícula será realizada de forma interactiva. Se le preguntará al alumno cuál es el plan de estudios en que desea matricularse (pueden ser varios).
- Se podrá generar una copia impresa de la matrícula (sin valor oficial) en el ordenador desde donde se realice el proceso de matriculación.
- Se podrá generar el impreso de pago debidamente cumplimentado.
- Para la matriculación se consultarán los datos del expediente y se realizarán las validaciones necesarias, descritas a continuación...
- **Pago de matrícula:**
  - La aplicación generará un impreso para que el alumno realice el pago correspondiente a la matrícula en 1 ó 2 plazos (según las fechas establecidas).
  - Si el alumno tiene matrículas de honor de cursos anteriores o disfruta de algún tipo de beca, la aplicación deberá calcular automáticamente los descuentos correspondientes...

Organizados jerárquicamente  
y desglosados en requisitos individuales



# Especificación de requerimientos en lenguaje natural



## EJEMPLO: REQUERIMIENTOS NO FUNCIONALES

### Interfaces

- **Hardware:** El sistema se debe implementar sobre la infraestructura existente en las aulas de prácticas de la E.T.S. Ingeniería Informática.
- **Software:**
  - No existe posibilidad de adquirir licencias de software.
  - La aplicación deberá funcionar sobre Oracle.

Requisitos claros, concretos, concisos  
y verificables



# Casos de uso



## Los casos de uso...

- Describen el modo en que un actor interactúa con el sistema (descripción de un rol en lenguaje natural).
- Narran el comportamiento dinámico del sistema desde un punto de vista concreto (el del actor).
- Pueden expresar tanto requerimientos funcionales como no funcionales.



# Casos de uso



## Los casos de uso...

- Son muy útiles para explicar el funcionamiento del sistema, priorizar requerimientos cuando el sistema se desarrolla de forma incremental, elaborar manuales de usuario y especificar pruebas de aceptación.
- Mejoran la trazabilidad de los requerimientos durante el proceso de desarrollo de software.
- Se pueden desarrollar en paralelo con los requerimientos del sistema de forma iterativa.



# Casos de uso



Dependiendo de la situación, los casos de uso se pueden especificar con distinto grado de detalle:

- **Especificación textual** de un caso de uso (enumeración de pasos del caso de uso).
- **Especificación “esencial”** de un caso de uso (eliminando todos los detalles no estrictamente necesarios).
- **Especificación detallada** de un caso de uso (utilizando una plantilla para no olvidarnos de nada).



# Casos de uso



## Especificación textual de un caso de uso (1/2)

Actor	Profesor
Rol	Consultar estadísticas

- El profesor ejecuta el programa de consulta de estadísticas.
- Se le pide su identificativo (*login*) y palabra clave de acceso (*password*).
- El sistema verifica la identificación del usuario.
- Si la identificación es positiva, se presenta una lista con las estadísticas disponibles:
  - N° de alumnos y porcentaje de repetidores de sus asignaturas.
  - Clasificación de alumnos por nota en cada asignatura.



# Casos de uso



## Especificación textual de un caso de uso (2/2)

Actor	Profesor
Rol	Consultar estadísticas

...

- Una vez que el profesor ha seleccionado una de las estadísticas, el programa presenta los datos correspondientes a la misma, agrupando la información por asignaturas y, al final, para todas sus asignaturas en conjunto.
- Al profesor se le da la opción de imprimir la estadística.
- Cuando el profesor termina de ver la estadística, se presenta de nuevo la lista de estadísticas disponibles.
- Si no desea ver otra estadística, termina la ejecución de la aplicación.



# Casos de uso



## Especificación esencial de un caso de uso

### Consulta de estadísticas

Profesor	Sistema
El profesor se identifica.	
	El sistema autentifica al profesor y le ofrece una lista de estadísticas disponibles.
El profesor selecciona una de las opciones disponibles.	
	El sistema presenta un informe con los datos solicitados.
Si así lo desea, el profesor imprime el informe.	

# Casos de uso



## Especificación detallada de un caso de uso (1/3)

Nombre	Consulta de estadísticas
Descripción	Se permite a los profesores consultar las estadísticas correspondientes a sus asignaturas
Dependencias	Autenticación de usuarios
Actores	Profesor (principal e iniciador)
Precondiciones	-
Postcondiciones	-

# Casos de uso



## Especificación detallada de un caso de uso (2/3)

Escenario principal	Profesor	Sistema
	1. El profesor se identifica.	
		2. El sistema autentifica al profesor y le ofrece una lista de estadísticas disponibles.
	3. El profesor selecciona una de las opciones.	
		4. El sistema presenta un informe con los datos solicitados.
	5. Si así lo desea, el profesor imprime el informe.	



# Casos de uso



## Especificación detallada de un caso de uso (3/3)

Alternativas		2. Si, tras un tercer intento, la autenticación no se realiza con éxito, se guarda la incidencia en un registro y se impide volver a acceder a la aplicación desde la misma IP durante 15 minutos.
Observaciones	-	
Requisitos no funcionales	El sistema debe estar preparado para aceptar 100 sesiones simultáneas de profesores consultando sus estadísticas sin degradar su rendimiento más de un 50% con respecto a un usuario único.	

# Apartados del documento de especificación del sistema



1. Definición del problema.
2. Descripción funcional  
(lista de requerimientos funcionales)
3. Restricciones  
(requerimientos no funcionales)
4. Diagramas de flujo de datos
5. Modelo de datos  
(diagrama E/R, CASE\*Method o diagrama de clases UML)
6. Diccionario de datos
7. Casos de uso
8. Documentos adicionales  
(p.ej. modelos de informes y formularios)

