

# *Especificación de requerimientos*

- 1. Requerimientos funcionales y no funcionales**
- 2. Especificación de requerimientos en lenguaje natural**
- 3. Herramientas de especificación**

## **Modelado de datos**

- Diagramas entidad/relación
- Diagramas de clases en UML
- Diccionarios de datos

## **Modelado de procesos**

- Diagramas de flujo de datos
- Casos de uso

- 4. Documento de especificación del sistema**

# 1. *Requerimientos funcionales y no funcionales*

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**

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 no funcionales**

Restricciones en 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...

NOTA: La distinción entre requerimientos funcionales y no funcionales no siempre resulta evidente (p.ej. la seguridad puede interpretarse inicialmente como un requerimiento no funcional al principio pero, tras elaborarlo, conduce a la aparición de requerimientos funcionales como la necesidad de autenticar a los usuarios del sistema).

## 2. 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),
- ✚ han de ser claros y concretos (evitando imprecisiones y ambigüedades) p.ej. Uso de puntos suspensivos, etcétera...
- ✚ deben ser concisos (sin rodeos ni figuras retóricas),
- ✚ deben ser completos y consistentes,
- ✚ 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?).

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).

Los **requerimientos no funcionales**...

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

## 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.
- Así mismo, 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...

### Gestión de docencia

- El secretario será el encargado de introducir qué profesores corresponden a cada asignatura (si no, no podrían introducir las actas los profesores).
- Los profesores de cada asignatura tendrán acceso a las listas de los alumnos que estén matriculados en sus asignaturas y la aplicación les debe permitir rellenar las actas.

### Estadísticas

- En secretaría se podrán obtener estadísticas que clasifiquen a los alumnos por su lugar de residencia, sexo, edad, cursos o asignaturas.

...

## 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 software. La aplicación deberá funcionar sobre Oracle

...

## Observaciones

La existencia de un requerimiento debe estar debidamente justificada.

---

MAL (amalgama de varios requisitos)

- ✗ 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.

BIEN (conciso y justificado)

- ✓ 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).

Otro problema habitual es que los requerimientos de un sistema son, a veces, difíciles de verificar (especialmente los requerimientos no funcionales).

---

MAL (objetivos generales, vagos y abiertos a distintas interpretaciones):

- ✗ 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.

BIEN (requisitos verificables):

- ✓ 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.
- ✓ 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.

# 3. Herramientas de especificación

Durante el desarrollo del sistema, se han de modelar tanto los datos empleados por el sistema como los procesos que realizan tareas sobre esos datos:

## **Modelado de datos**

- Diagramas entidad/relación
- Diagramas CASE\*Method
- Diagramas de clases en UML
- Diccionarios de datos
- ...

## **Modelado de procesos**

- Diagramas de flujo de datos
- Diagramas de estados (autómatas finitos)
- Casos de uso
- ...

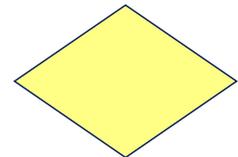
# Herramientas de especificación: Modelado de datos

## Diagramas entidad/relación (E/R)

- ✓ Entidades (conceptos de interés)



- ✓ Relaciones (asociaciones entre entidades)  
Cardinalidad: uno-a-uno, uno-a-muchos, muchos-a-muchos



- ✓ Atributos (características de las entidades)

## Preparación de un diagrama E/R

1. Identificación de las entidades.
2. Identificación de las relaciones.
3. Identificación de los atributos.
4. Normalización.

Intuitivamente:

1NF Todos los atributos de una relación toman valores atómicos.

2NF Todos los atributos de una relación dependen funcionalmente de la clave primaria de la relación.

3NF El valor de un atributo no puede determinarse a partir de los valores de los demás atributos.

## Restricciones semánticas

---

- Cada profesor pertenece a un solo departamento.
- Todo profesor pertenece a algún departamento.
- Todo departamento debe tener un director, que es un profesor.
- Un profesor puede impartir varios grupos de la misma o diferentes asignaturas.
- Un grupo de una asignatura ha de estar impartido por, al menos, un profesor.
- Las asignaturas se imparten en clases en días, horas y aulas determinadas.
- Los alumnos se matriculan de varias asignaturas (al menos una).
- Una asignatura puede tener varios alumnos matriculados.
- Los atributos de cada entidad son “los habituales”.

## Identificación de entidades, atributos y relaciones

---

<b>Entidades</b>	Asignatura	Alumno	Profesor	Depart.	Aula	Grupo
<b>Atributos</b>	<u>ID</u> Nombre Créditos Carácter Curso ...	<u>DNI</u> Nombre Dirección Beca ...	<u>NRP</u> Nombre Categoría Área ...	<u>ID</u> Nombre ...	<u>ID</u> Capacidad ...	<u>ID</u> Tipo ...

## Relaciones

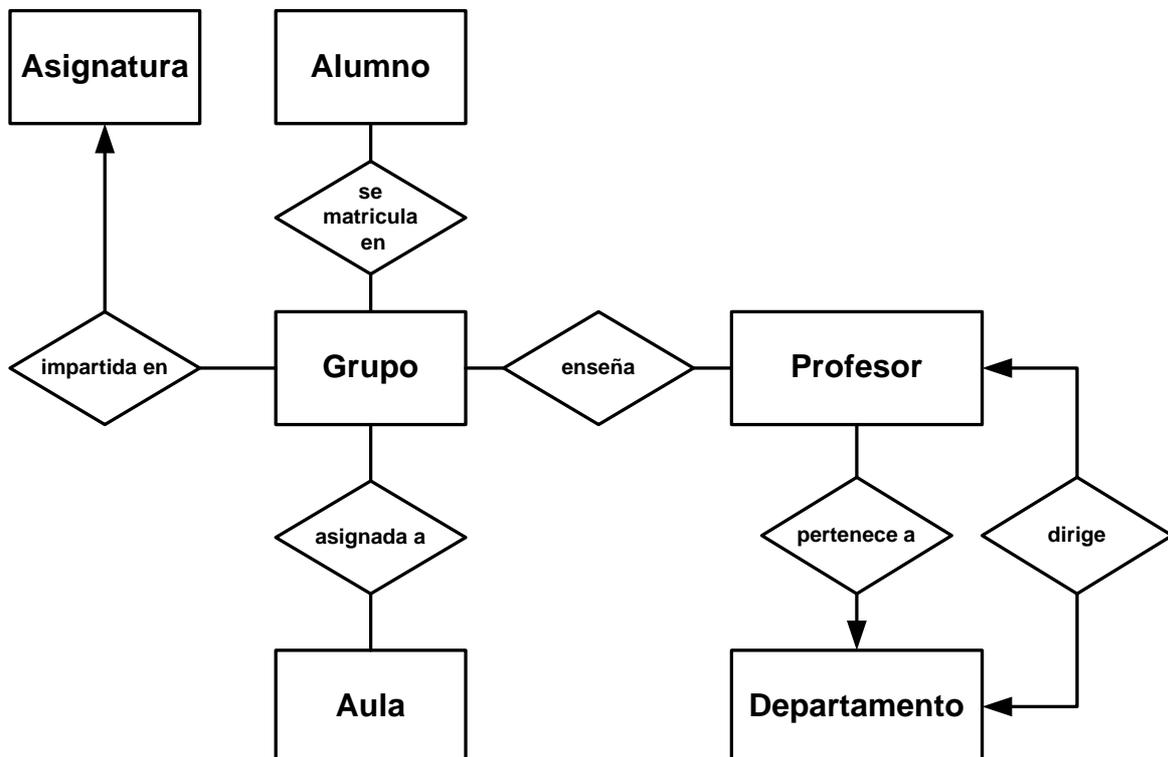
---

Relación	Entidades participantes	Cardinalidad	Atributos
se matricula en	Alumno – Grupo	N:M	Calificación
enseña	Profesor – Grupo	N:M	
impartida en	Asignatura – Grupo	1:N	
asignada a	Aula – Grupo	N:M	Día, hora
pertenece a	Profesor – Departamento	N:1	
dirige	Profesor – Departamento	1:1	

## Diagrama entidad/relación

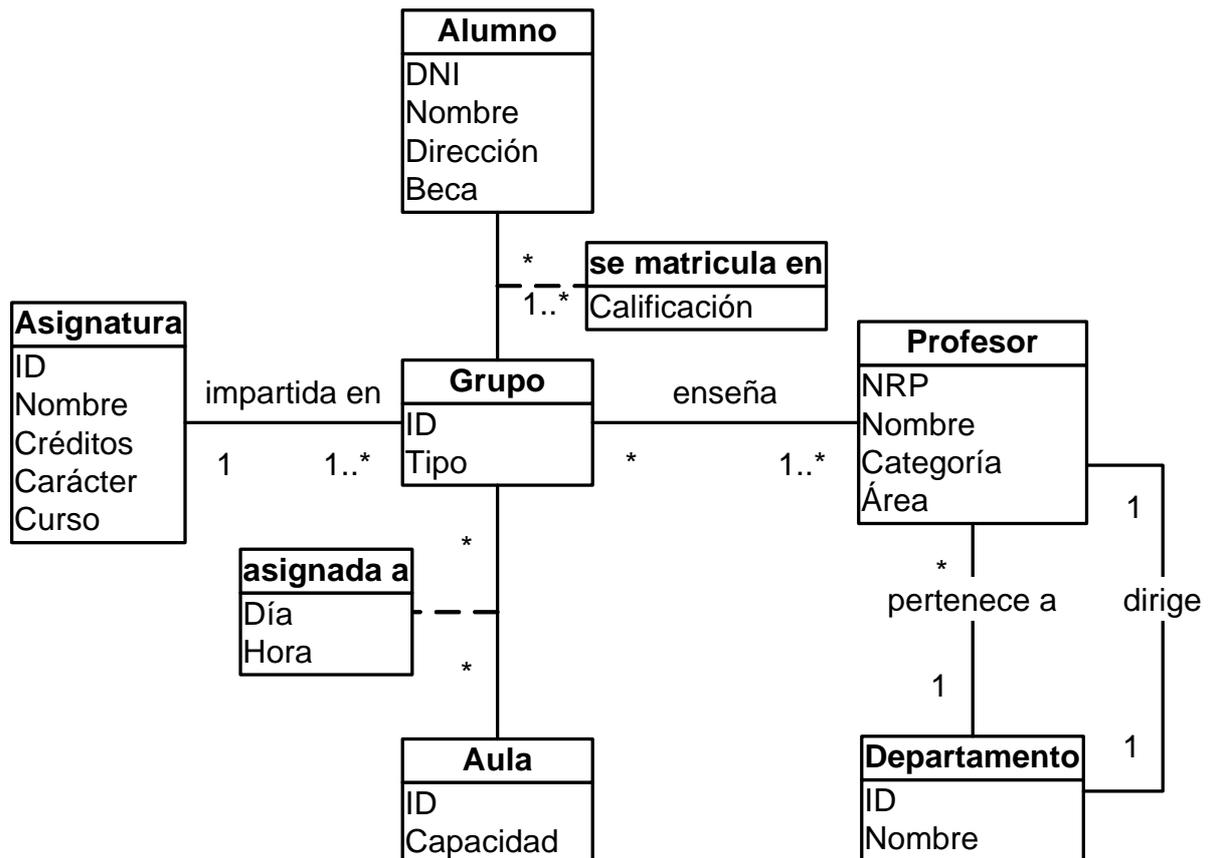
---

(omitiendo los atributos para favorecer la legibilidad del diagrama)



*Diagramas de clases en UML*  
*UML = Unified Modeling Language*

- ✓ Clases (conceptos de interés)
- ✓ Asociaciones (relaciones entre clases)
- ✓ Atributos (características de los objetos de una clase)



*Diccionario de datos*

Descripción escrita de los datos almacenados en la base de datos

Entidad ASIGNATURA	
Atributo	Descripción
<b>Cod-asig</b>	Código oficial de la asignatura
<b>Nombre</b>	Nombre completo de la asignatura
<b>Créditos</b>	Número de créditos de la asignatura (teoría y prácticas)
<b>Carácter</b>	Tipo de asignatura (troncal, obligatoria, optativa...)
<b>Curso</b>	Curso del plan de estudios en la que está incluida la asignatura
...	

# *Herramientas de especificación: Modelado de procesos*

## *Diagramas de flujo de datos (DFD)*

Representación gráfica de un sistema que ilustra cómo fluyen los datos a través de distintos procesos.

Los DFDs se realizan a distintos niveles de abstracción, detallando procesos concretos que aparecen como elementos simples en DFDs de nivel superior.

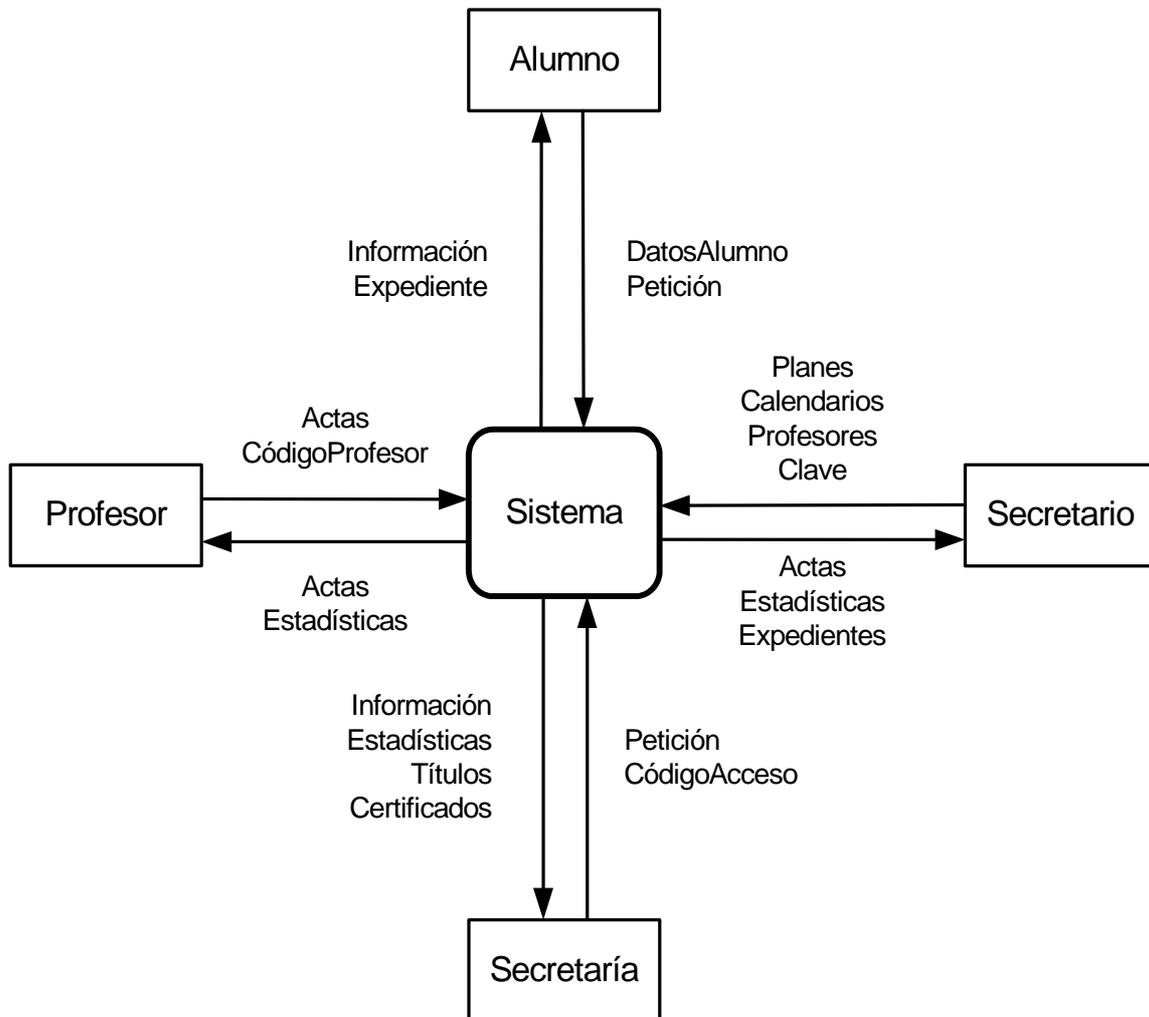
### **Elementos de un DFD**

- **Entidades externas, terminadores o elementos del entorno:**  
Fuentes o sumideros de información. Emiten o reciben la información que fluye a través de las interfaces externas del sistema (vg: usuarios).
- **Flujos de datos**  
Indican el flujo de información a través del sistema
- **Procesos o actividades**  
Transforman la información que les llega a través de los flujos de datos de entrada en la información que sale a través de los flujos de datos de salida.
- **Almacenes de datos y ficheros**  
Lugares donde se guardan los datos para su procesamiento posterior.

NOTAS:

- Cada elemento tiene asociado un nombre unívoco a modo de etiqueta.
- Los flujos de datos pueden converger o divergir.
- Procesos y ficheros no pueden poseer sólo flujos de entrada (ni sólo de salida)
- Los flujos no pueden incluir información de control
- Las entradas y salidas netas de un DFD deben coincidir con los flujos de entrada y salida del proceso a que corresponde en el nivel superior.

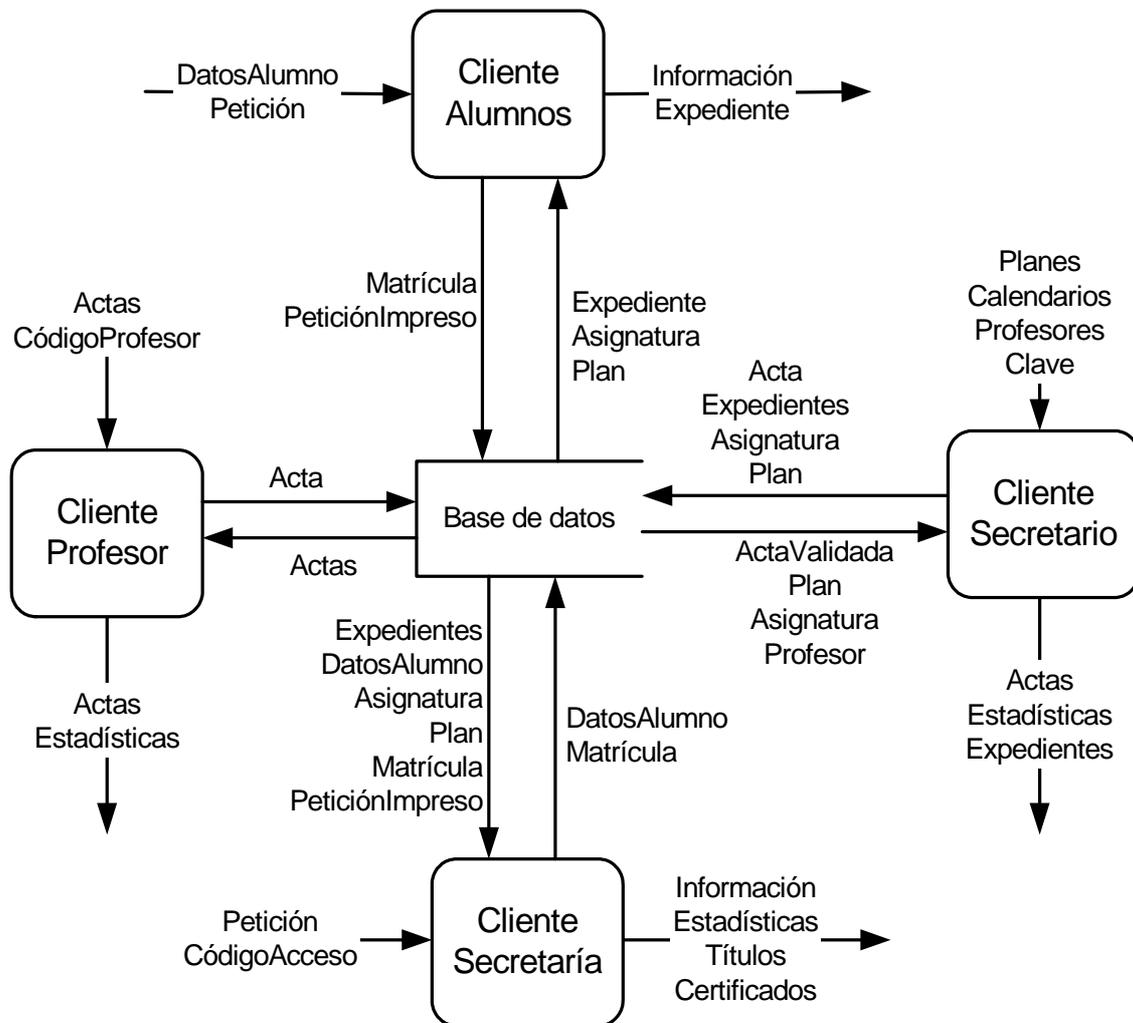
## DFD de contexto



### Notación de Gane&Sarson



## DFD de nivel 1

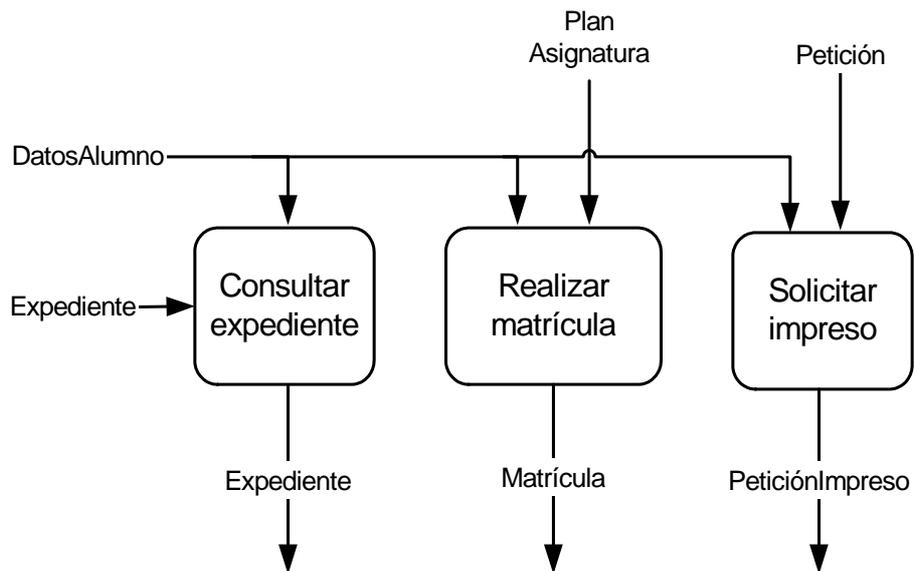


### Acerca del diagrama de contexto (DFD de nivel superior)

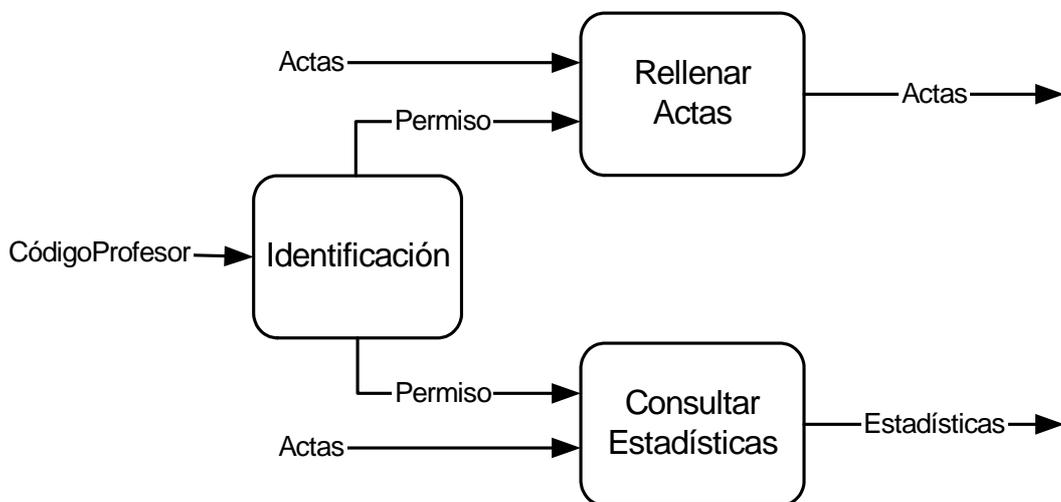
Contiene un único proceso que representa al sistema completo, todos los terminadores (entidades externas que interactúan con el sistema) y todos los flujos de datos que conectan un sistema con su entorno.

## DFDs de nivel 2

### Ciente Alumnos



### Ciente Profesor



## 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.
- Pueden expresar tanto requerimientos funcionales como no funcionales.
- 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.

### Caso de uso

<b>Actor</b>	<b>Profesor</b>
<b>Rol</b>	<b>Consultar estadísticas</b>

- 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.
- Si la identificación es positiva, se presenta una lista de estadísticas:
  - N° de alumnos y porcentaje de repetidores de sus asignaturas.
  - Clasificación de alumnos por nota en cada asignatura
- Una vez que el profesor ha seleccionado la estadística, el programa presenta los datos correspondientes a la misma, agrupando la información por asignaturas, y al final para todas sus asignaturas en conjunto.
- El profesor podrá 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, sale del programa terminando la ejecución del mismo.

## 4. 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)