

# 13019 – Diseño de bases de datos

## *Diseño de requisitos mediante casos de USO*

Wladimiro Díaz

Wladimiro.Diaz@uv.es

Universitat de València

# Introducción

- En literatura, un *escenario* es un esquema que describe la situación en términos que el lector puede entender fácilmente.
- En UML, los *casos de uso* cumplen una función similar en el desarrollo de los sistemas de *software*.
- Como los escenarios, los casos de uso proporcionan, un tenue reflejo del funcionamiento real del sistema.
- Son fundamentales: describen a alto nivel qué es lo que el sistema debe hacer.

# Efectos en el diseño de BD

1. El caso de uso representa una transacción atómica del sistema.
  - El conocimiento de las transacciones del sistema permite diseñar la base de datos lógica y física y la arquitectura de modo que se acomode a estos requisitos.

# Efectos en el diseño de BD

2. El caso de uso muestra cómo el sistema utiliza los datos, tanto internamente (inserción, actualización y borrado) como externamente (consulta).
  - Los casos de uso constituyen la base de las reglas de negocio que definen las restricciones del diseño de la base de datos.

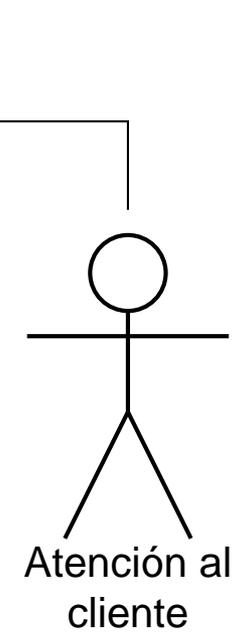
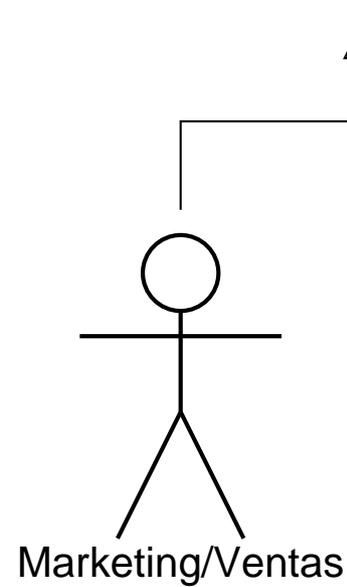
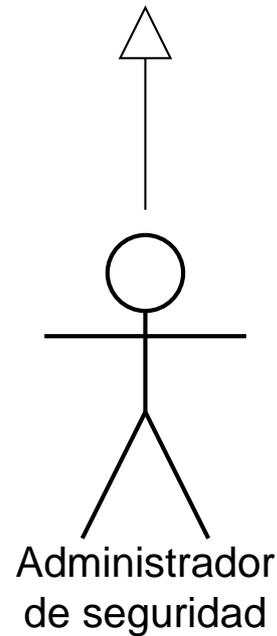
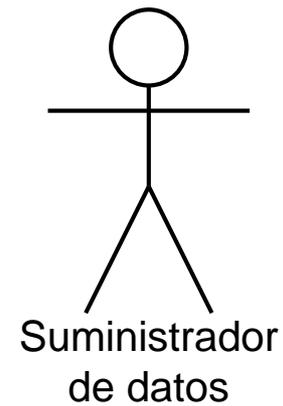
# Efectos en el diseño de BD

3. ...

# Actores

- El **actor** se define como el papel que desempeña un objeto u objetos externos al sistema que interaccionan directamente con él como parte de una unidad de trabajo coherente (el caso de uso).
- Los actores representan el contexto del caso de uso.
- El actor puede ser un usuario, pero también puede ser otro subsistema de *software* o incluso una pieza de *hardware*.

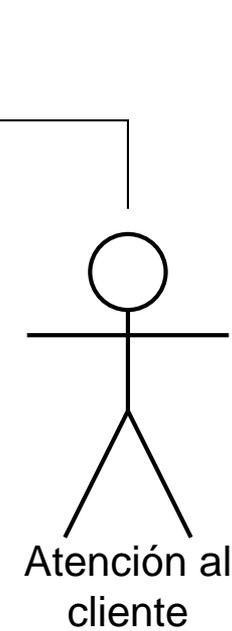
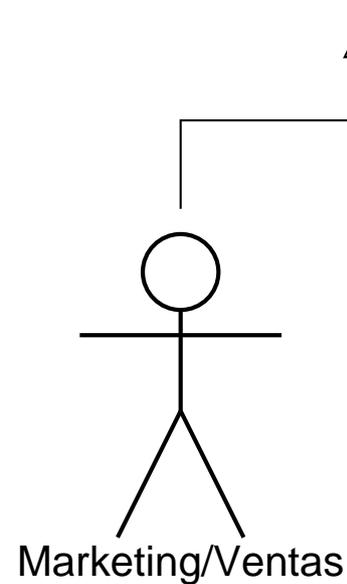
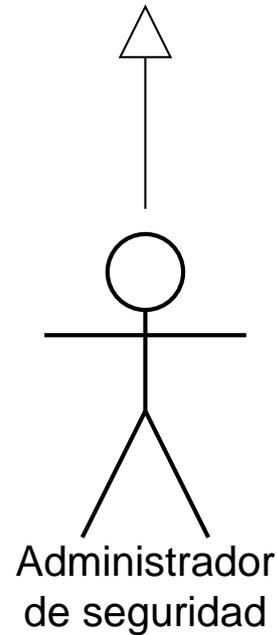
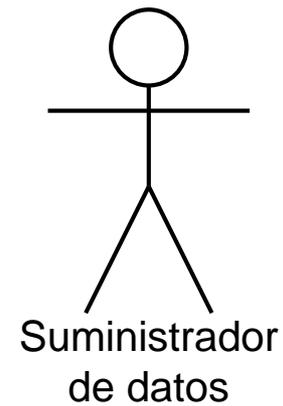
# Actores



# Jerarquías de actores

- Puede suceder que algunos actores se relacionen a través de una relación de generalización. Por ejemplo:
  - ◆ Un *Administrador de seguridad* puede ser un tipo de *Administrador de la base de datos*.
  - ◆ Los empleados de los departamentos de *Atención al cliente* y de *Marketing/Ventas* pueden considerarse un tipo de *Suministrador de datos*.

# Jerarquías de actores



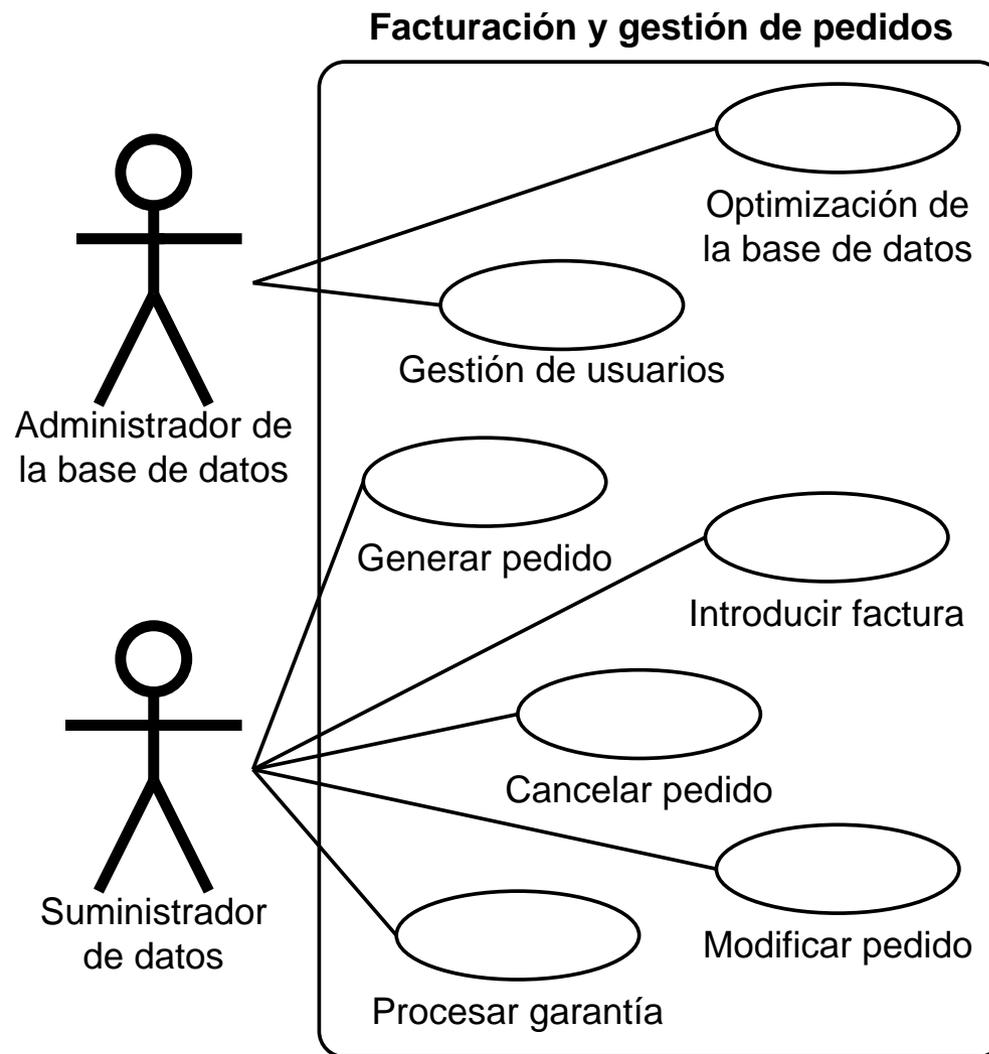
# Jerarquías de actores

- El diseño jerárquico de actores permite simplificar la descripción de los casos de uso.
- La herencia significa que un *subactor* hereda el papel desempeñado por su *superactor*.
  - ◆ Todo lo que puede hacer el *Administrador de la base de datos* también lo puede hacer el *Administrador de seguridad*, pero no al revés.
- La definición de la jerarquía de actores es el primer paso para describir el *escenario* de nuestro sistema: el modelo de casos de uso.

# Diagramas de casos de uso

- El diagrama de caso de uso es un diagrama de contexto que muestra las relaciones de alto nivel entre los actores y los casos de uso.
- También es posible incluir en estos diagramas relaciones entre casos de uso (relaciones *use* y *extends*).
- La idea de este diagrama es la de modelar las transacciones del sistema al nivel más alto y con muy poco detalle.

# Diagramas de casos de uso



# Diagramas de casos de uso

- El rectángulo representa las fronteras del sistema de **facturación y gestión de pedidos**.
- Los actores aparecen fuera de los límites del sistema mientras que los casos de uso aparecen dentro del sistema, ya que constituyen transacciones del propio sistema.
- Los casos de uso se representan mediante óvalos etiquetados y en los que no aparece ningún tipo de estructura.

# Construcción del diagrama

- Para cada actor identificado en el diagrama de actores, imaginar cada una de las transacciones que éste inicia.
- Crear y etiquetar un caso de uso en el sistema para cada una de estas transacciones y conectala con el actor mediante una *asociación*.
- Normalmente se comienza con los actores de mayor nivel en la jerarquía de actores y después se desciende añadiendo casos de uso especializados para los subactores.

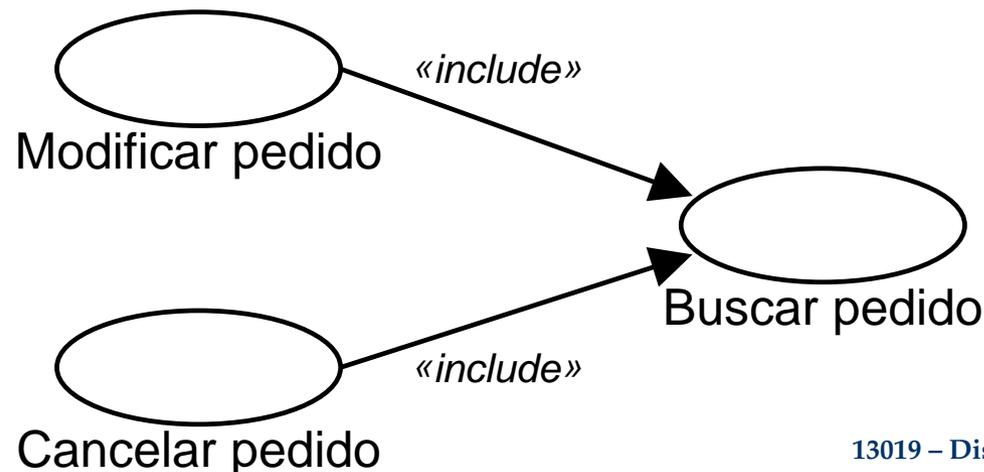
# Relaciones entre casos de uso

- La relación «*extends*»: El caso de uso *extiende* condicionalmente el comportamiento de otro caso de uso.
  - ◆ La extensión no es una transacción completa, sino una parte condicional de la transacción.



# Relaciones entre casos de uso

- La relación «*include*»: El caso de uso contiene la funcionalidad de otro caso de uso como parte de su procesamiento normal.
  - ◆ Reducen la complejidad y la redundancia.
  - ◆ Los casos de uso incluidos se invocan cada vez que se da el principal.



# Diagramas de actividad

- El diagrama de actividad representa gráficamente la lógica de control del caso de uso.
- Recuerda a un *diagrama de flujo*, pero difiere en la notación y la semántica.

# Elementos del diagrama

- Reflejan dos estructuras de control básicas: la *secuencia* y la *condición*.
- La *secuencia* es la conexión directa entre los elementos.
- La *condición* es el equivalente a una sentencia *if*.
- El *diagrama de actividad* es un caso especial de diagrama de estado UML.

# Diagramas de actividad

