



# Programación 4

## Guía Semana 9 (20/05)

InCo, FING, Udelar

# Objetivo

Los objetivos de esta semana son:

1. analizar los criterios **GRASP** para la asignación de responsabilidades en un diseño;
2. presentar el concepto de **visibilidad** entre objetos y su impacto en el diseño.



# Resumen :: GRASP

Una **responsabilidad** es una obligación que un objeto tiene: de saber o conocer, o de hacer

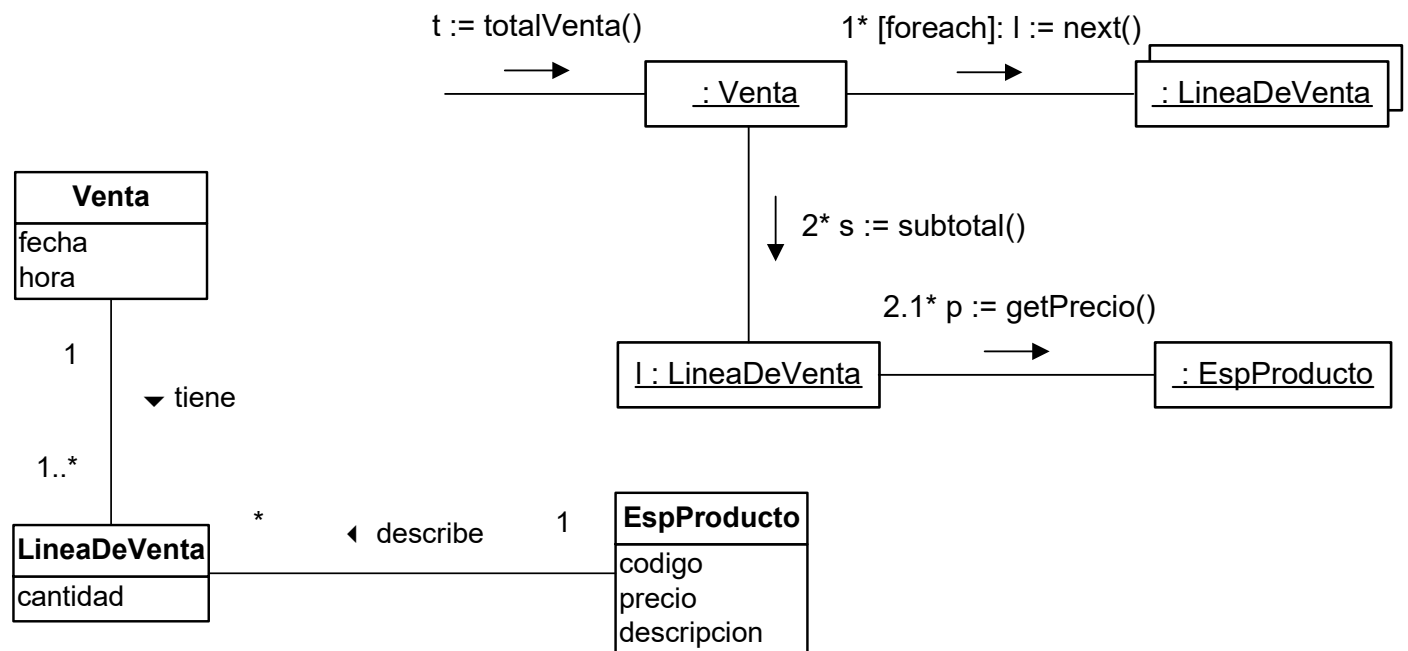
Los **criterios GRASP** guían la asignación de responsabilidades para un buen diseño de

- **Expert** - ¿Quién es el experto en la información necesaria?
- **Creator** - ¿Quién debe crear un objeto?
- **Bajo Acoplamiento** - Evitar que un objeto interactúe con demasiados objetos
- **Alta Cohesión** - Evitar que un objeto haga demasiado trabajo
- **No Hables con Extraños** - Asegurarse que un objeto realmente delega trabajo
- **Controller** - ¿Quién tiene las operaciones del sistema?



# Resumen :: GRASP

Ejemplo: ¿quién es el responsable de conocer el total de una venta?



# Resumen :: Acceso a la Lógica

**Controller** - ¿Quién tiene las operaciones del sistema?

- Llamaremos **controlador** a una clase que implemente una **operación del sistema**
- ¿Cómo definirlos de forma equilibrada?
  - Agrupar operaciones de casos de uso que estén relacionados entre sí temáticamente
  - Definir un controlador de fachada para cada uno de los grupos de casos de uso
- De esta forma existe un “**mini fachada**” por cada uno de los grupos definidos



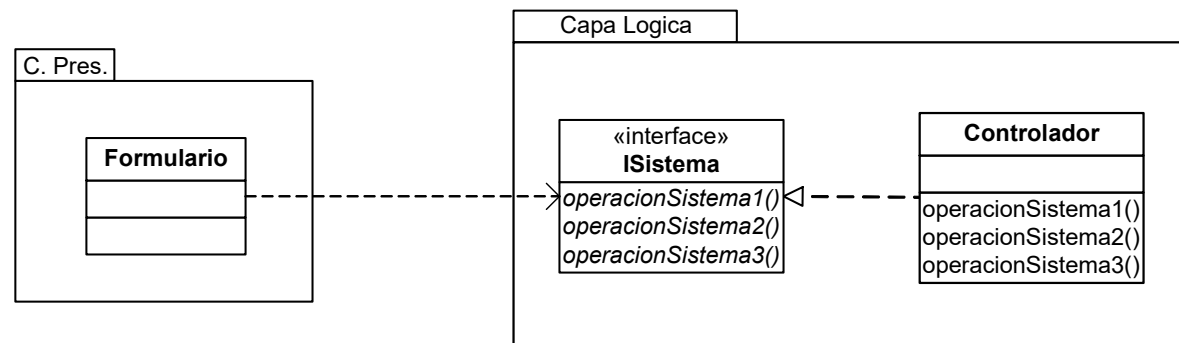
# Resumen :: Acceso a la Lógica

Las **operaciones del sistema** que implementan los controladores deben ser ofrecidas en interfaces

A estas interfaces se las llama **interfaces del sistema**

Son utilizadas por la capa de presentación para acceder a las funcionalidades del **Sistema**

Las interfaces permiten **quebrar la dependencia** entre las clases de presentación y los controladores



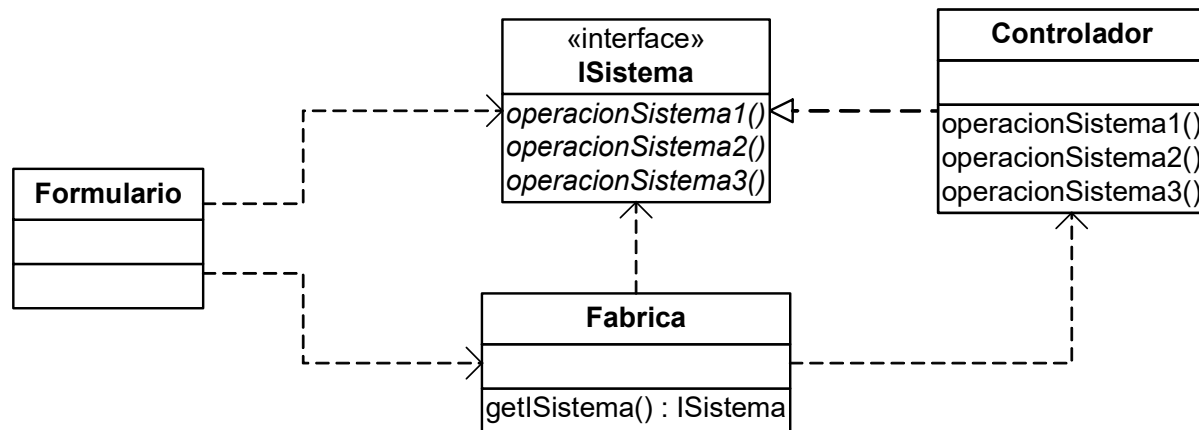


# Resumen :: Acceso a la Lógica

Una interfaz no es suficiente para quebrar la dependencia entre dos clases

Una **fábrica (factory)** es un objeto responsable de crear instancias que realicen una interfaz determinada

El invocador quedará acoplado a la fábrica pero no dependerá del realizador de la interfaz (el controlador)



# Resumen :: Acceso a la Lógica

¿Quién tiene la responsabilidad de contener las instancias de una clase determinada?

- **Opción 1:** Un **controlador**, porque las operaciones del sistema están vinculadas con esta clase
- **Opción 2:** Una **clase del dominio**, porque la clase tiene “propiedad” sobre la otra (ej.: composición)
- **Opción 3:** Una **clase ficticia**, porque las otras alternativas no son aceptables, por ejemplo, los controladores dependen mucho entre sí, o no hay una clase de dominio identificada. Suele existir una única instancia de esta clase que centraliza el acceso.





# Resumen :: Visibilidad

Los diagramas de comunicación ilustran el envío de mensajes entre un objeto y otro al que “ve” (**referencia**)  
La **visibilidad** es la capacidad de un objeto de tener una referencia a otro

Existen cuatro formas básicas de que un objeto A tenga visibilidad sobre otro B:

- **por atributo**: B es un pseudoatributo de A
- **por parámetro**: B es un parámetro de un método de A
- **local**: B es declarado localmente en un método de A
- **global**: B es visible en forma global



# ¿Qué hago esta semana?

1. Estudio los materiales de [Teórico](#) y las lecturas recomendadas. Las clases correspondientes se encuentran en [OpenFing](#).

11 - Diseño: GRASP (desde diap. 32)

12 - Diseño: Visibilidad

Caso de Estudio :: Diseño

2. Finalizo el [Práctico](#) 4 “Diseño, Diagramas de Comunicación”.
3. Continúo el [Laboratorio](#) 3 “Diseño”.  
Plazo de entrega: lunes 03/06, 15hs
4. Estudio el [material adicional](#) sobre Diseño  
Plantilla: Informe del Modelo de Diseño

