

# ARQUITECTURA 4+1 VISTA

# *VISTA LOGICA*

.

# Clasificación de UML

Área	Vista	Diagramas	Conceptos Principales
Estructural	Vista Estática	Diagrama de Clases	Clase, asociación, generalización, dependencia, realización, interfaz.
	Vista de Casos de Uso	Diagramas de Casos de Uso	Caso de Uso, Actor, asociación, extensión, generalización.
	Vista de Implementación	Diagramas de Componentes	Componente, interfaz, dependencia, relaización.
	Vista de Despliegue	Diagramas de Despliegue	Nodo, componente, dependencia, localización.
Dinámica	Vista de Estados de máquina	Diagramas de Estados	Estado, evento, transición, acción.
	Vista de actividad	Diagramas de Actividad	Estado, actividad, transición, determinación, división, unión.
	Vista de interacción	Diagramas de Secuencia	Interacción, objeto, mensaje, activación.
		Diagramas de Colaboración	Colaboración, interacción, rol de colaboración, mensaje.
Administración o Gestión de modelo	Vista de Gestión de modelo	Diagramas de Clases	Paquete, subsistema, modelo.
Extensión de UML	Todas	Todos	Restricción, estereotipo, valores, etiquetados.

# EJEMPLOS DE NOTACION UML

<b>NombreClase</b>	clase		componente
<b>NombreClase</b>	clase activa		nodo
<b>nombre: Clase</b>	rol		nota
<b>nombreObjeto: Clase Rol</b>	objeto		paquete
<b>nombreObjeto: Clase</b>	objeto múltiple		interfaz
<b>NombreAsociación</b>	asociación		coleboración
	generalización		
	realización		
	dependencia		parámetro-plantilla
		{ expresión }	plantilla
			restricción

# DIAGRAMA DE CLASES

## *Diagramas de clases*

- . Son los diagramas más comunes en el modelado de sistemas orientados a objetos.
- . Muestra un conjunto de clases, interfaces, y colaboraciones y sus relaciones.
- . Se usan para modelar la vista de diseño estático de un sistema.
- . Son importantes no sólo para visualización, especificación y documentación de modelos estructurales, sino también para construir sistemas ejecutables .

# *Términos y conceptos*

- . Un diagrama de clase comparte las mismas propiedades comunes como lo hacen todos los otros diagramas – un nombre y contenido gráfico.
- . Los diagramas de clase contienen lo siguiente: Clases, interfaces, colaboraciones, y relaciones de dependencia, generalización y asociación.
- . El diseño estático de la vista de un sistema soporta principalmente los requerimientos funcionales de un sistema – los servicios que el sistema debe proporcionar a sus usuarios finales.

## *Usos comunes*

- . **Modelar el vocabulario de un sistema.** Involucra decidir sobre las abstracciones que son parte del sistema y las que están fuera de sus fronteras.
- . **Modelar colaboraciones simples.** Una colaboración es una sociedad de clases, interfaces, y otros elementos que trabajan juntos para proporcionar algún comportamiento cooperativo que es mayor que la suma de todos los elementos.
- . **Modelar un esquema de base de datos lógico.** Es como el anteproyecto para un diseño conceptual de una base de datos.

# *Representación de clase en UML*

. Una clase es una descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, relaciones y semántica

- **Nombre**
- **Atributos**
- **Operaciones**

Forma
origen
mover() tamano() desplegar()

# *Nombres*

- . Cada clase debe tener un nombre que la distinga de otras clases.
- . Un nombre es una cadena de texto.
- . Un nombre solo se conoce como nombre simple; una ruta nombre es el nombre de clase precedido del nombre del paquete en que vive la clase.

Sensor de temperatura

Reglas de negocio::Agente

Nombre simple

Ruta nombre

# *Atributos*

- . Un atributo es una propiedad de una clase que describe un rango de valores que la propiedad puede conservar.
- . Una clase puede tener cualquier número de atributos o carecer de ellos.
- . Un atributo representa alguna propiedad de la cosa que se está modelando. Por ejemplo cada pared tiene altura, ancho y espesor.
- . Al atributo se le puede especificar su clase y posiblemente un valor por omisión .

Pared
altura: Float
ancho:Float
estaPintada:Boolean=true

# TIPOS DE ATRIBUTOS

- Los atributos de una clase no deberían ser manipulables directamente por el resto de objetos. Por esta razón se crearon niveles de visibilidad para los elementos que son:
  - (-) Privado : es el más fuerte. Esta parte es totalmente invisible (excepto para clases friends en terminología C++)
  - (#) Los atributos/operaciones protegidos están visibles para las clases friends y para las clases derivadas de la original.
  - (+) Los atributos/operaciones públicos son visibles a otras clases (cuando se trata de atributos se está transgrediendo el principio de encapsulación)

# *Operaciones*

- . Una operación es la implementación de un servicio que puede ser solicitado desde cualquier objeto de la clase para afectar el comportamiento.
- . Una clase puede tener cualquier número de operaciones o carecer de ellas.
- . Un nombre de operación es un verbo o frase verbal que representa algún comportamiento.
- . En una operación se puede establecer su firma, abarcando el nombre, tipo y valor de omisión de todos los parámetros y tipo de regreso .

SensorTemperatura
reiniciar()
ponerAlarma(t:Temperatura)
valor():Temperatura

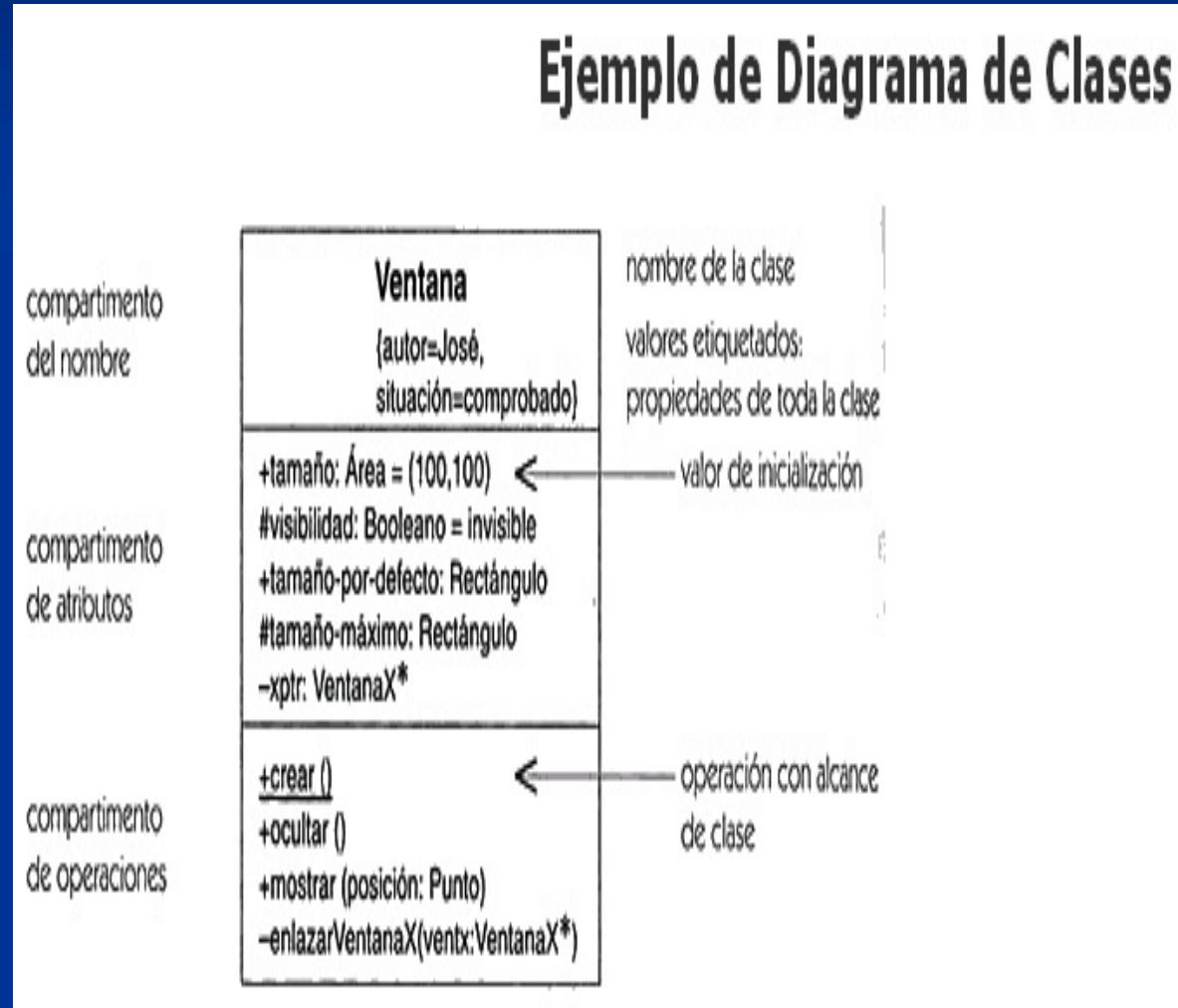
# REGLAS DE ESCRITURA

- El nombre de un atributo, el nombre de una operación se escribe con minúsculas si consta de una sola palabra. Si el nombre contiene más de una palabra, cada palabra será unida a la anterior y comenzará con una letra mayúscula, a excepción de la primera palabra que comenzará en minúscula. Por ejemplo: abrirPuerta, cerrarPuerta, buscarPuerta, etc.

# ASOCIACION

- Uno y sólo uno
- 0..1 Cero o uno
- M..N Desde M hasta N (enteros naturales)
- \* Cero o muchos
- 0..\* Cero o muchos
- 1..\* Uno o muchos (al menos uno)

## Ejemplo de Diagrama de Clases



# AGREGACION

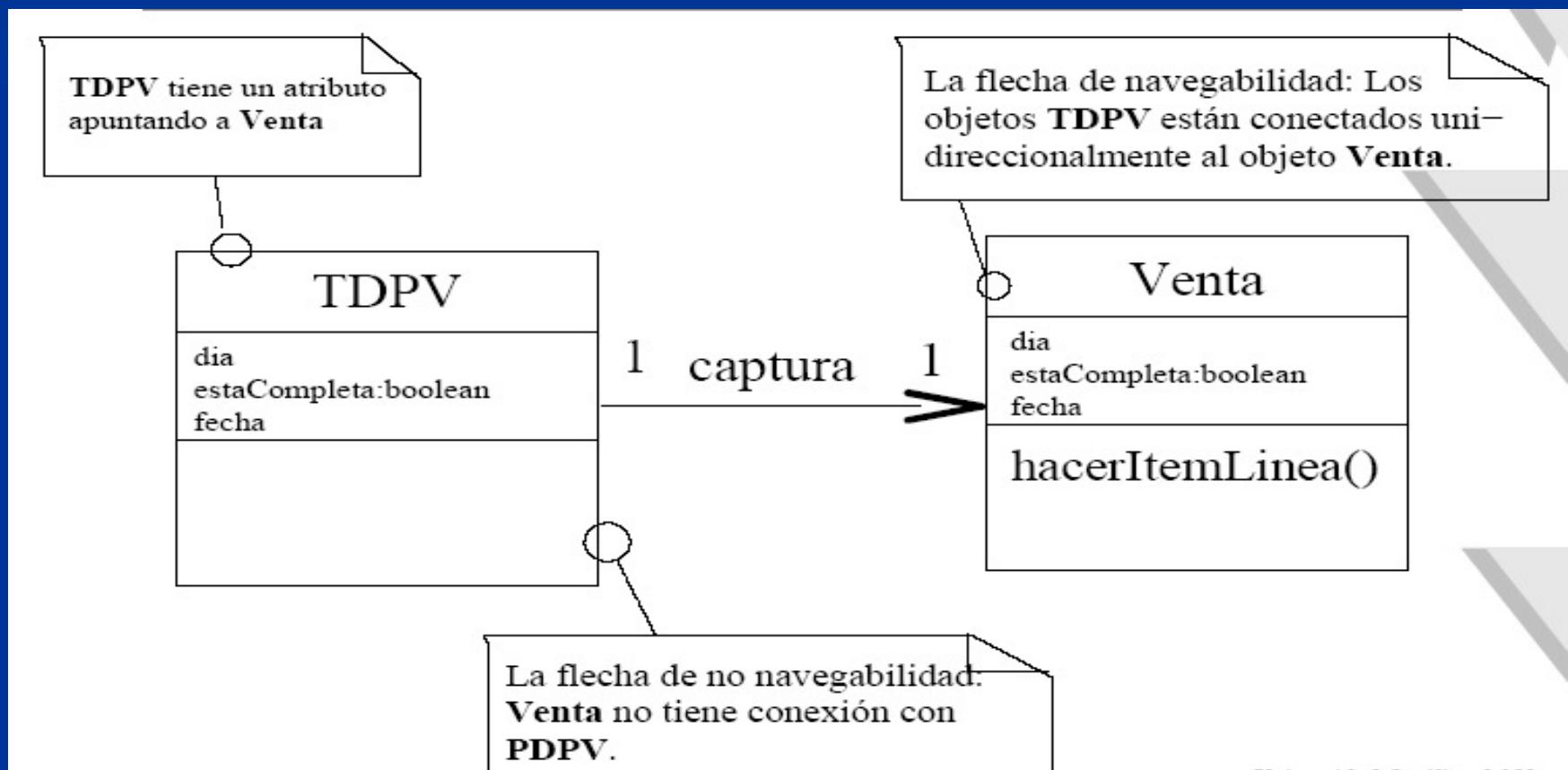
- Esta relación puede ser caracterizada con precisión determinando las relaciones de comportamiento y estructura que existen entre el objeto agregado y cada uno de sus objetos componentes
- Puede el objeto parte comunicarse directamente con objetos externos al objeto agregado?  
No => inclusiva  
Si => no inclusiva  
Puede cambiar La composición del objeto agregado?  
Si => dinámica  
No => estática

# GENERALIZACION

- La Generalización consiste en factorizar las propiedades comunes de un conjunto de clases en una clase más general.
- Los nombres usados: clase padre - clase hija. Otros nombres: superclase - subclase, clase base - clase derivada. Las subclases heredan propiedades de sus clases padre, es decir, atributos y operaciones

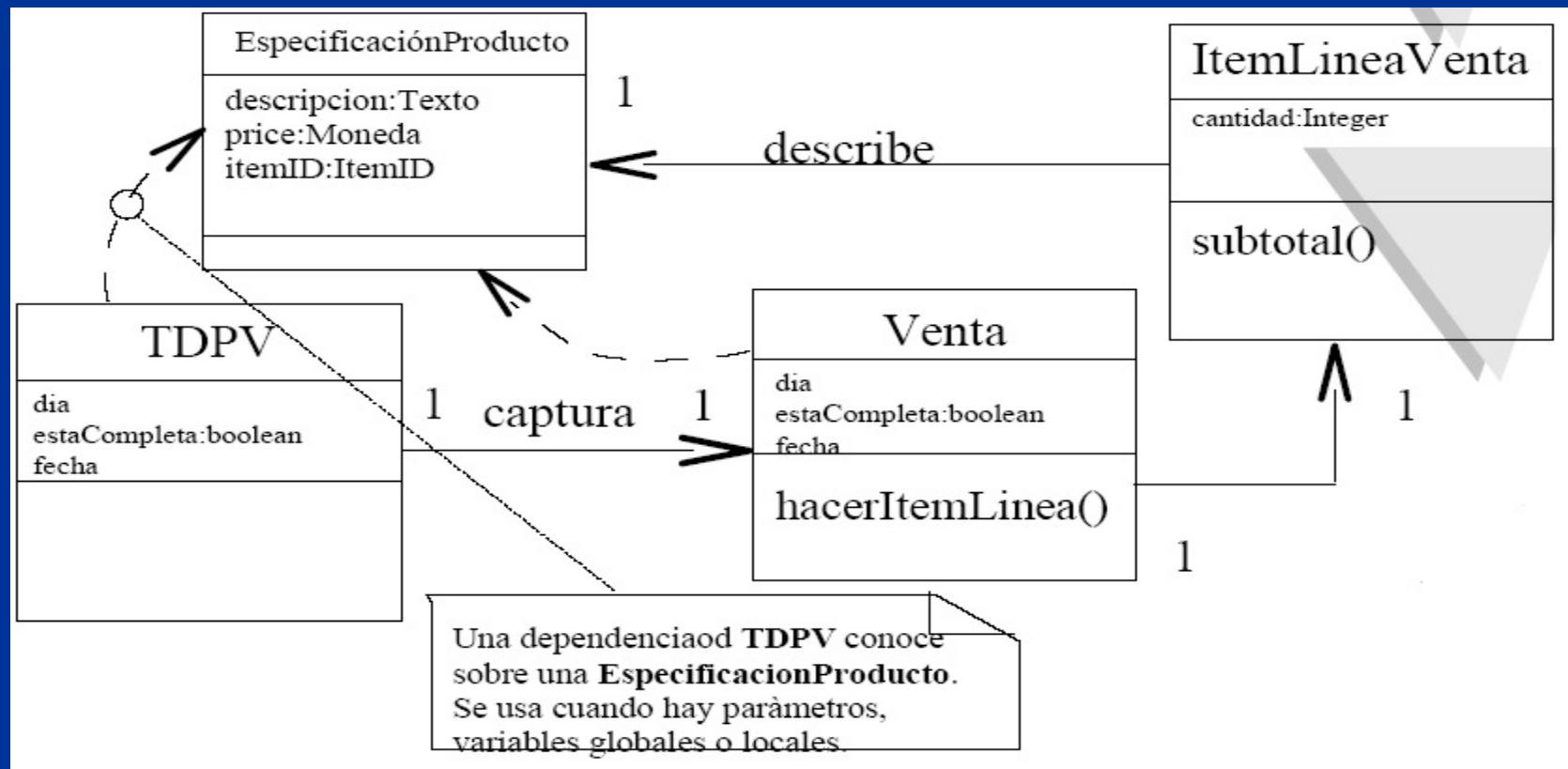
# NAVEGABILIDAD

Es recomendable que las asociaciones sean “adornadas” con las flechas de navegación.

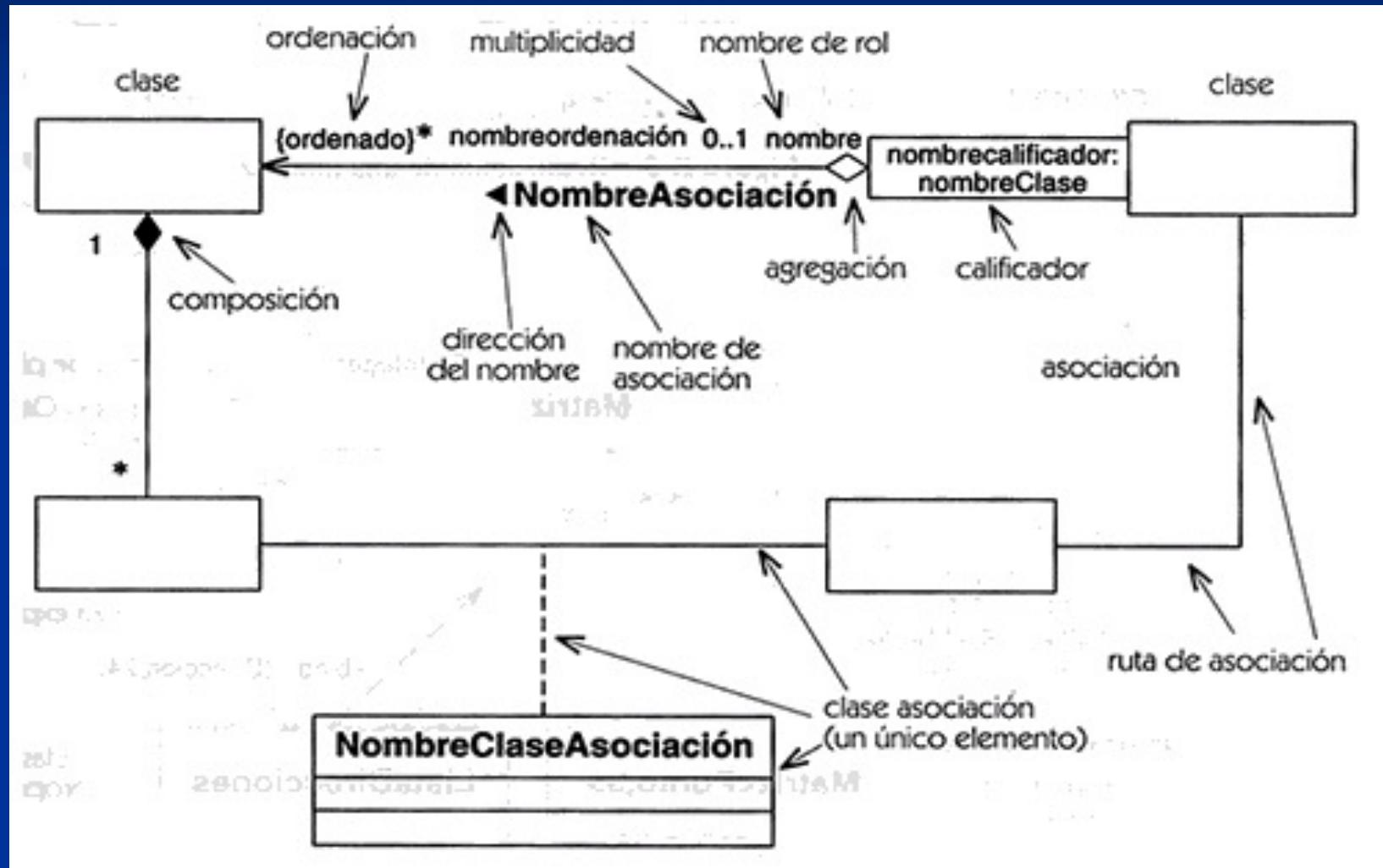


# DEPENDENCIA

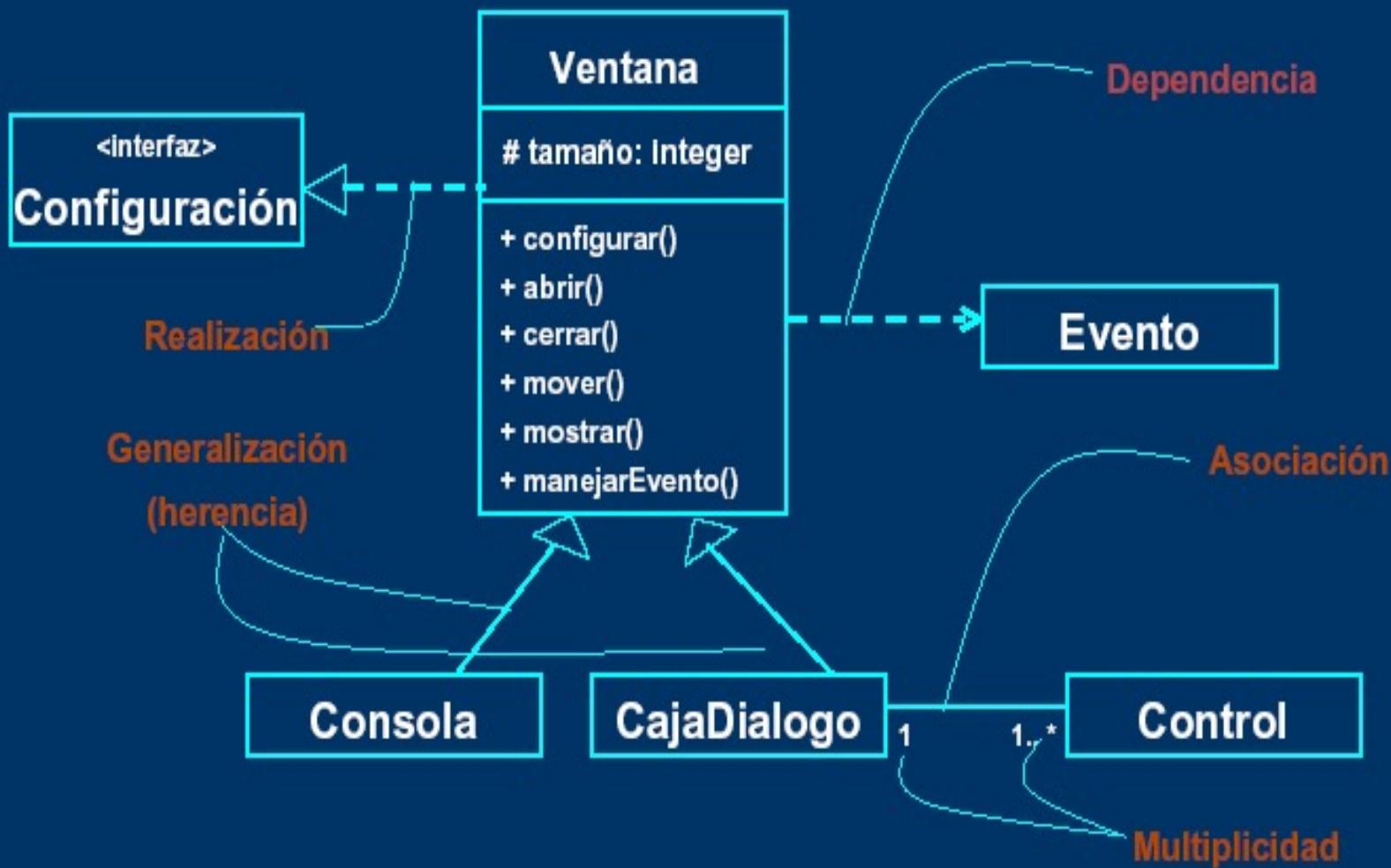
- En UML una *relación de dependencia* indica que un elemento tiene conocimiento de otro elemento.



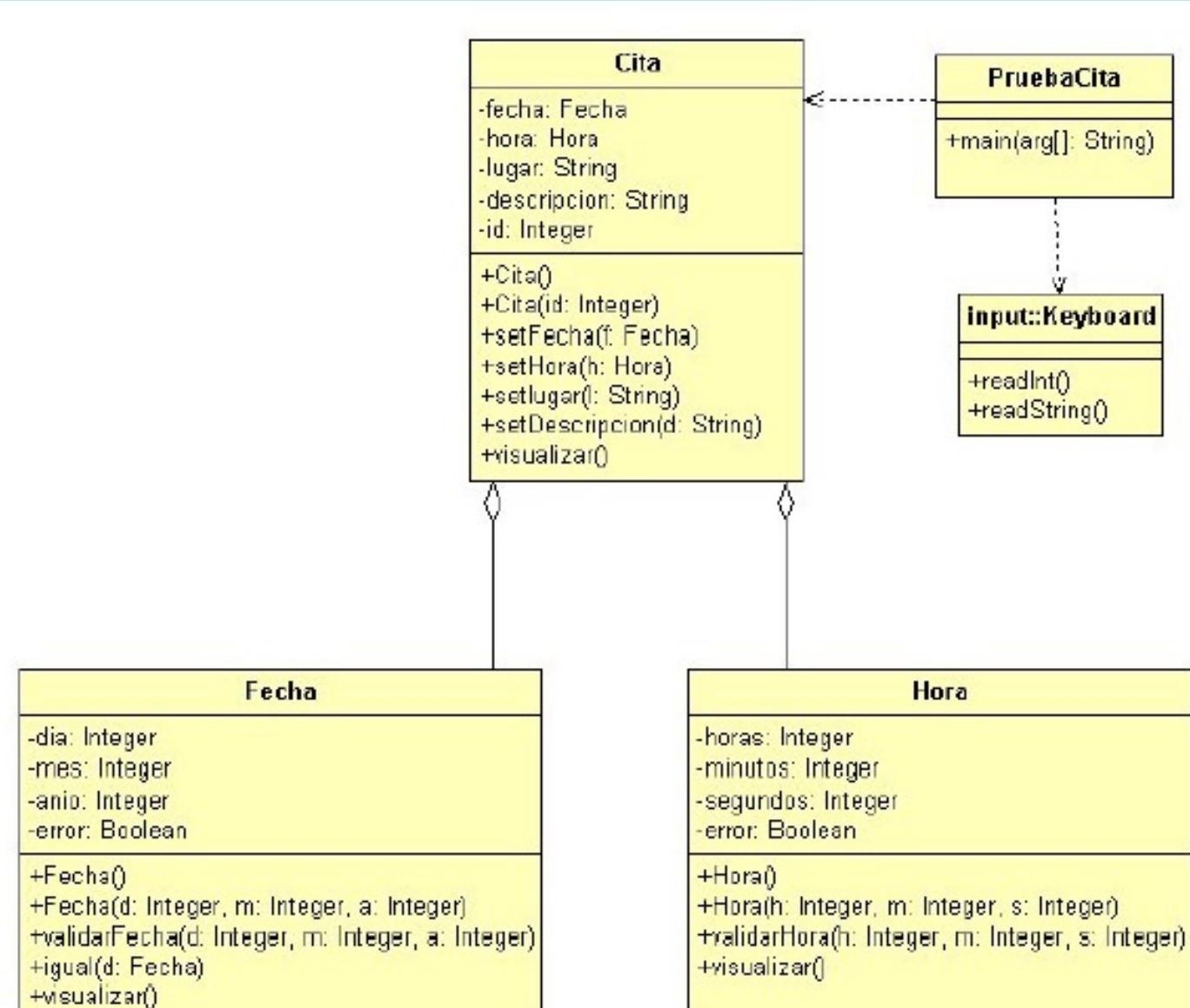
# SIMBOLOGIA



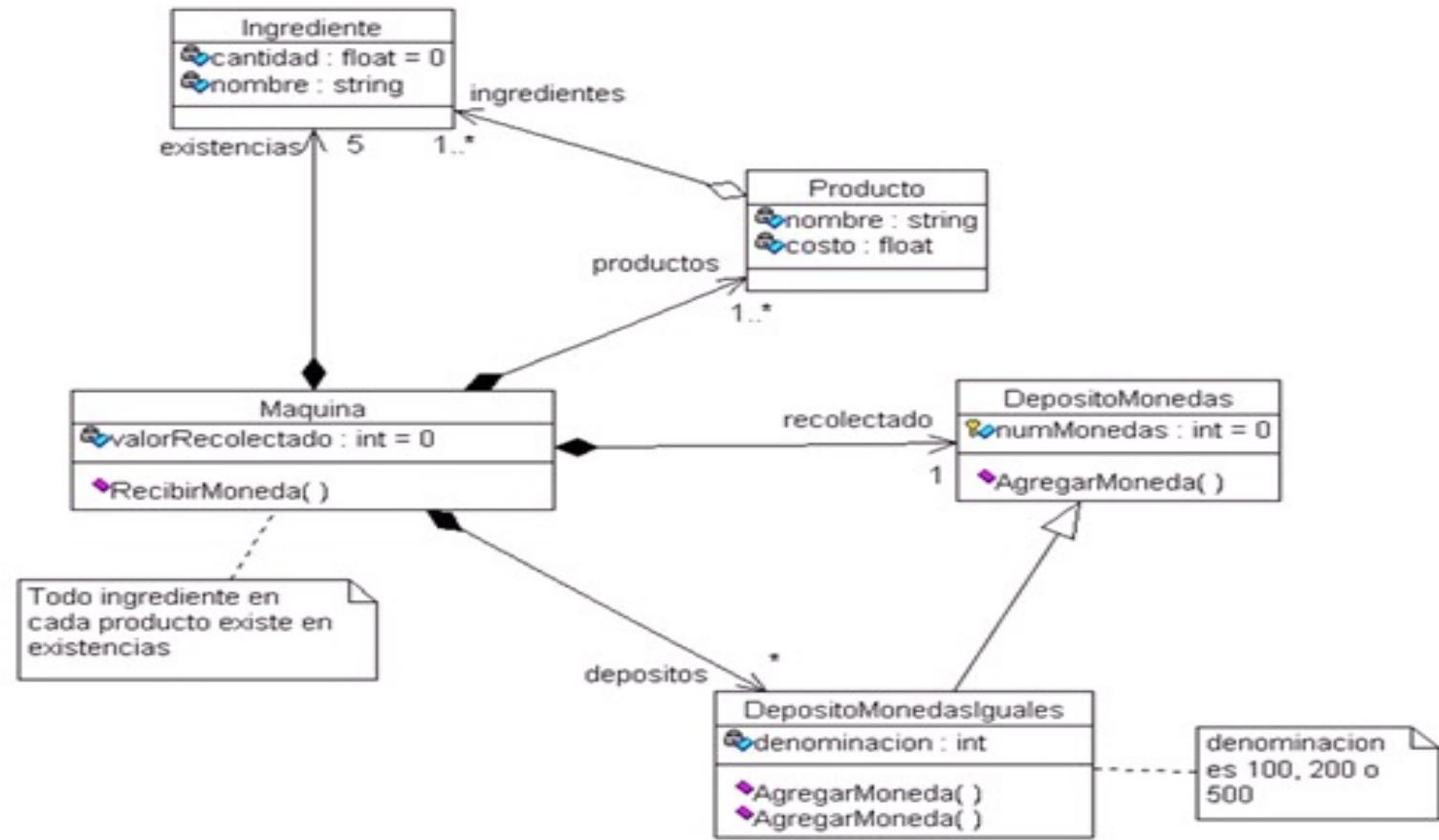
# Relaciones entre clases



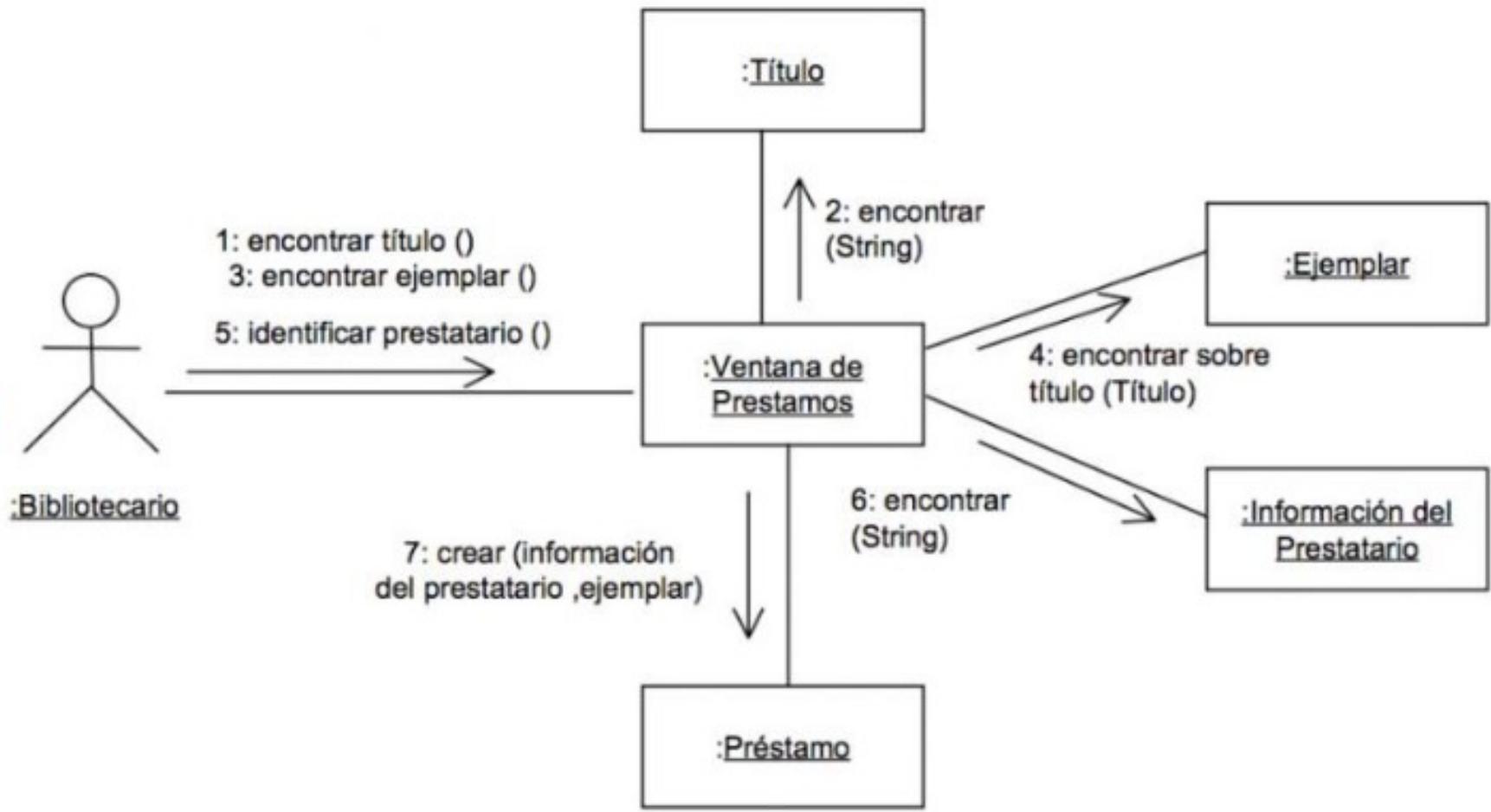
# Ejemplo: Diagrama de clases



## Ejemplo 2: Diagrama de Clases



# DIAGRAMA DE COMUNICACIÓN



# DIAGRAMA DE SECUENCIA

