Relaciones

Curso: Programación Orientada a Objetos
Ciclo 2020-01-A

LOGRO DE LA UNIDAD 2

Al finalizar la unidad 2 el alumno logrará implementar programas para resolver casos prácticos aplicando los conceptos de relaciones entre clases.

AGENDA

1. Relación
2. Tipos de Relaciones
3. Multiplicidad
4. Composición/Agregación
5. Conclusiones

1. Relación

Una relación es una conexión semántica entre elementos de un modelo.

2. Tipos de Relaciónes

Las relaciones entre clases que existen son:

– Asociación

– Composición

- Agregación

– Generalización.

2.1 Asociación

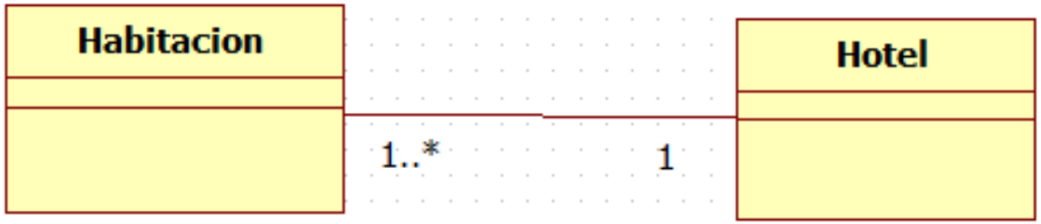
- Una asociación representa la relación entre dos o más clases.
- Una asociación binaria representa una relación entre dos clases.
- Existe una asociación binaria si un objeto de una clase requiere un objeto de otra clase para hacer su trabajo. “Para cada X hay un “Y”.
- Se representa por medio de una línea continua entre dos clases.

2.2 Asociación - UML

Aquí, cada cliente vive en una dirección y dirección es utilizada por un solo cliente (es decir, un objeto Cliente está asociado sólo a un objeto Dirección).

3. Multiplicidad (cardinalidad)

- La multiplicidad es el número de instancias que tiene una clase en relación con otra clase.



Asociación uno a muchos entre clases que representa un hotel y sus habitaciones y entre una habitación del Hotel.

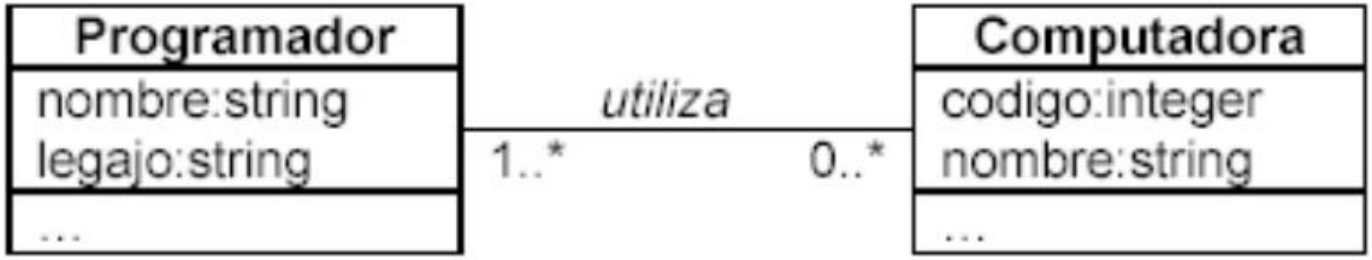
Multiplicidad

La multiplicidad puede especificarse con un solo entero o con un rango **n..m** donde n es el limite inferior y m es el limite superior. Se puede utilizar un asterisco para denotar que no existe un límite superior.

0..1	Cero o una instancia
*0..** ó *	Cero o más instancias
1	Exactamente una instancia
1..*	Una o más instancias

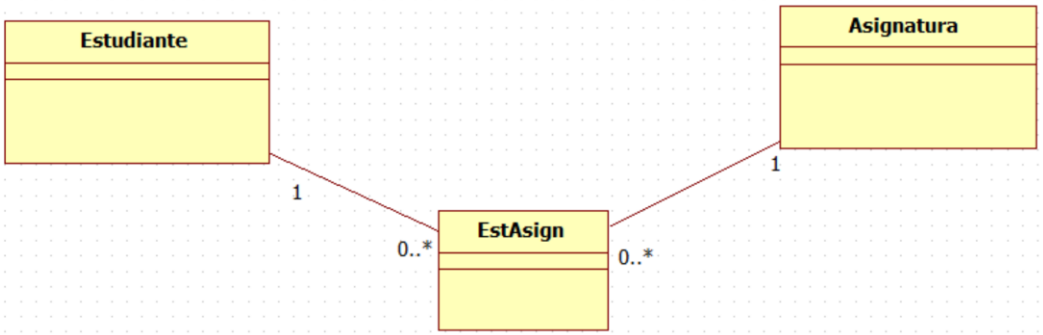
Las asociaciones pueden clasificarse de acuerdo a su multiplicidad, uno a uno, uno a muchos y muchos a muchos.

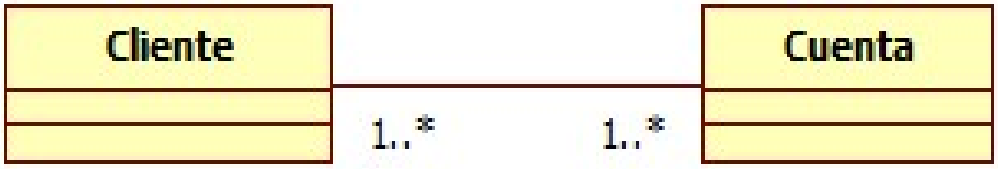
Multiplicidad

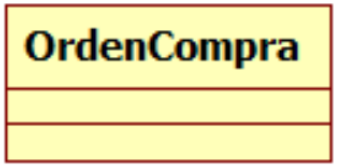


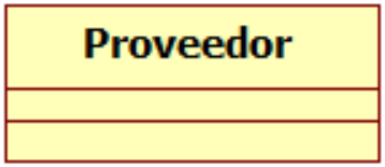
Este diagrama indica que cada programador tendrá varias computadoras (posiblemente ninguna), y que cada computadora será usada por al menos un programador.

Una relación muchos a muchos se puede ver como dos relaciones de uno a muchos









Ustedes……

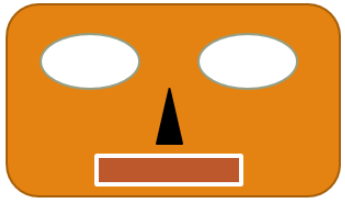
1) **Muchos a muchos**
2) **Uno a muchos**
3) **Uno a uno**

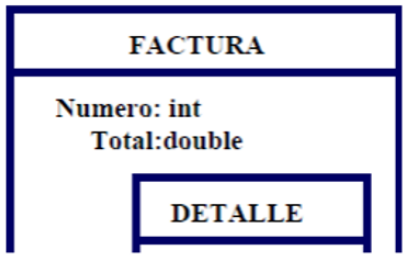
4. Relaciones de Composición / Agregación

Son formas especiales de relación donde una clase está compuesta de otra clase.

En tal forma que un atributo de una clase es un objeto de otra clase.

Se les conoce como relación TODO-PARTE.

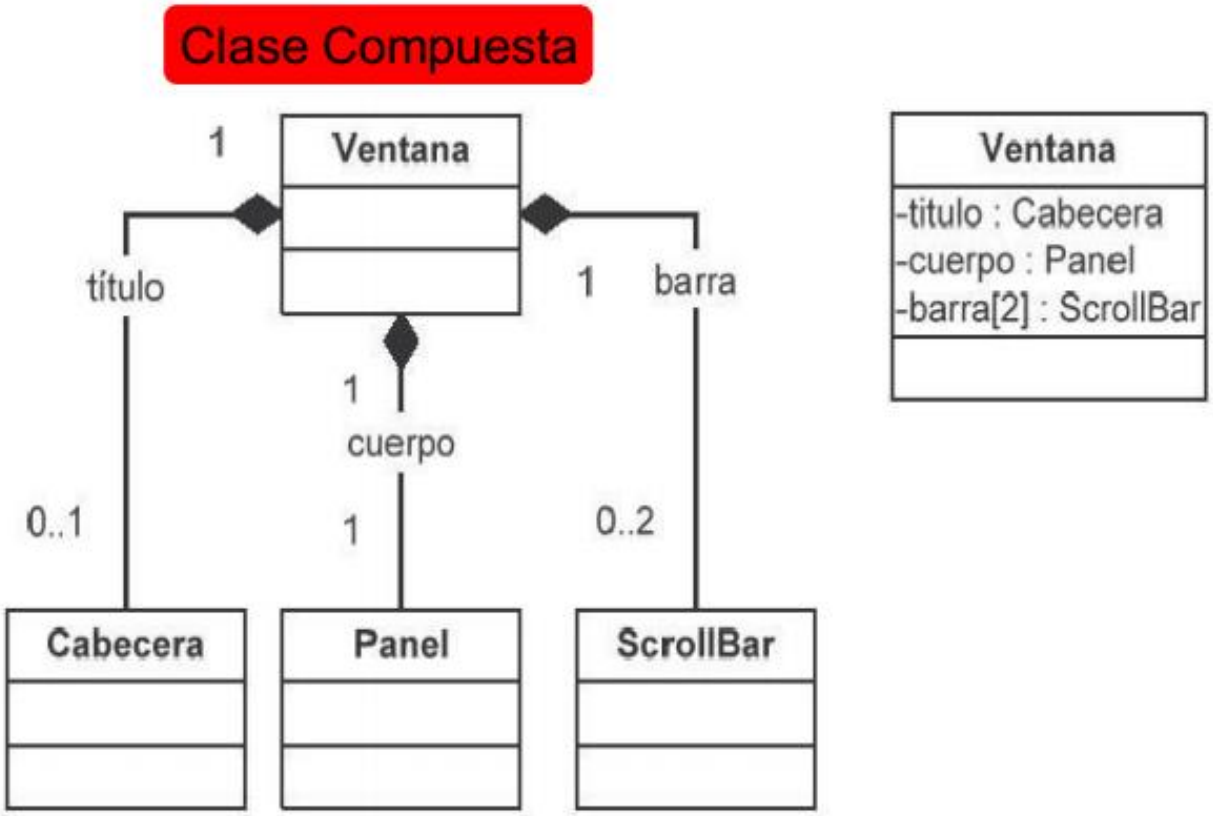




4.1 Composición

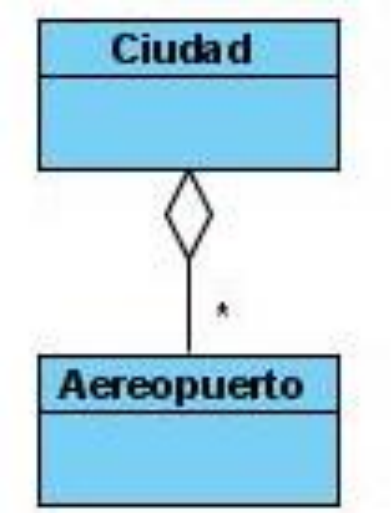
Si la relación es fuerte, tal que objetos de la clase PARTE son dependiente de la existencia de la clase TODO entonces la relación es **composición**. La clase **TODO**, tiene la responsabilidad de la creación y destrucción de objetos de sus componentes

4.1 Composición



4.2 Agregación

- Al contrario, si la existencia de objetos de la clase **PARTE** es **independiente** de la existencia de objetos de la clase **TODO**, entonces la relación es **agregación**.



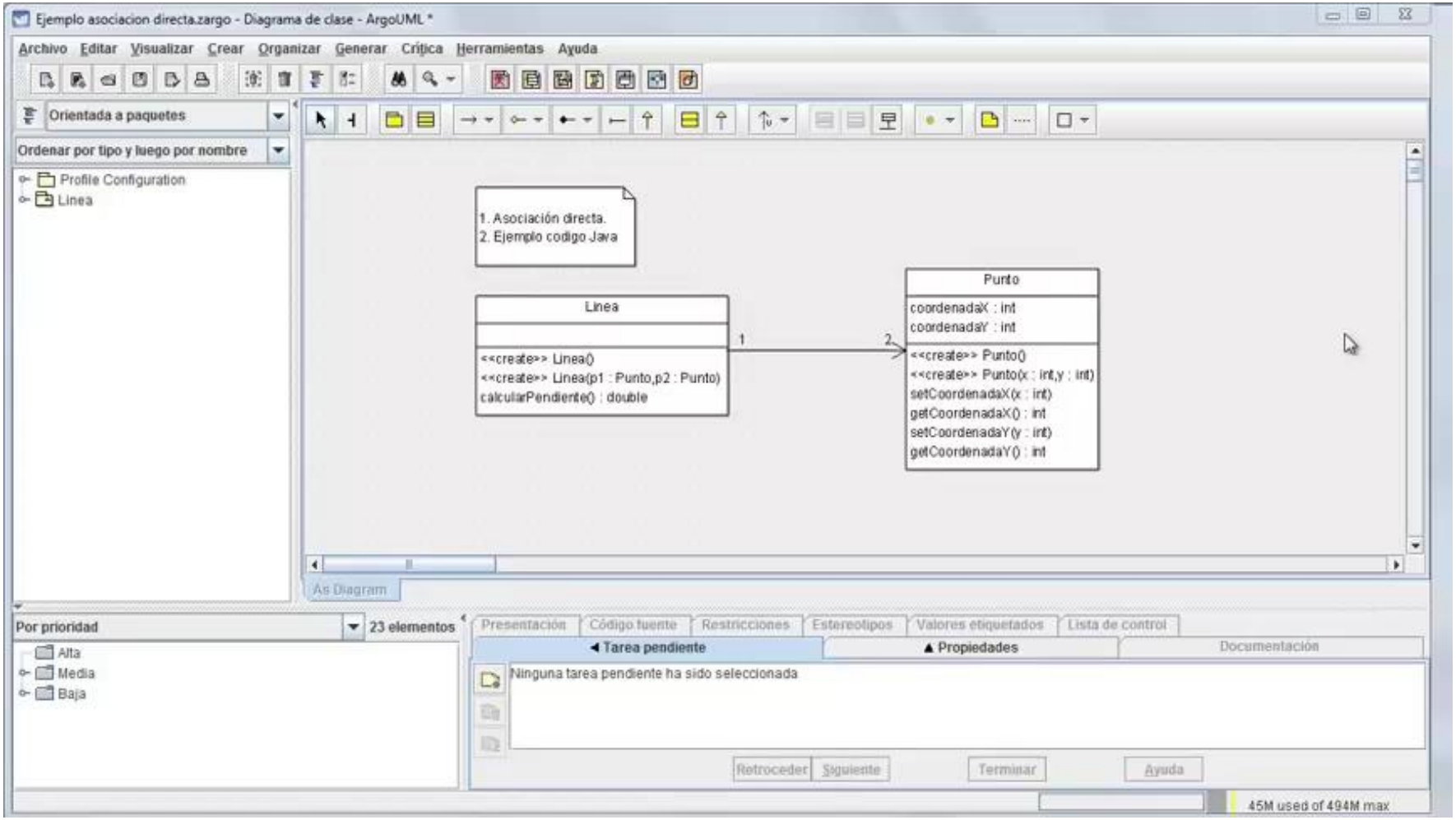
Representación

Agregación / Composición

- Su representación gráfica es la siguiente:

5. Conclusiones

- Las relaciones entre clases nos permiten entender mejor como es que ciertas clases utilizan o colaboran con otras para lograr un objetivo del Sistema.
- Muchas veces el significado de la relación entre clases dependen exclusivamente del negocio, no hay que inventarlo ni aplicar sólo nuestro sentido común para identificarlos.
- La representación de las relaciones entre clases se realizan utilizando la especificación UML.


Ejemplo asociacion directa.zargo - Diagrama de clase - ArgoUML *
Archivo Editar Visualizar Crear Organizar Generar Crítica Herramientas Ayuda
Orientada a paquetes
Ordenar por tipo y luego por nombre
Profile Configuration
Linea
1. Asociación directa.
2. Ejemplo codigo Java
Linea
<<create>> Linea()
<<create>> Linea(p1 : Punto,p2 : Punto)
calcularPendiente() : double
1
2
Punto
coordenadaX : int
coordenadaY : int
<<create>> Punto()
<<create>> Punto(x : int,y : int)
setCoordenadaX(x : int)
getCoordenadaX() : int
setCoordenadaY(y : int)
getCoordenadaY() : int
As Diagram
Por prioridad
23 elementos
Alta
Media
Baja
Presentación
Código fuente
Restricciones
Estereotipos
Valores etiquetados
Lista de control
Tarea pendiente
Propiedades
Documentación
Ninguna tarea pendiente ha sido seleccionada
Retroceder
Siguiente
Terminar
Ayuda
45M used of 494M max


Gracias!