

Programación IV

Modelado de objetos –
parte 2



Atributos derivados

- Los atributos básicos son atributos independientes dentro del objeto. En contraste, los atributos derivados son atributos que dependen de otros atributos. Los atributos derivados dependen de otros atributos del objeto, los cuales pueden ser básicos o derivados. La notación es una diagonal como prefijo del atributo, como se muestra en las figuras

Nom bre de la Clase
/ Atributo

Prestamo
- Id_lector
- Id_libro
- fecha_inc
- / fecha de entrega
- vigencia
- Id_prestamo

Atributos derivados

- Ejemplo: El Área de un Rectángulo se puede calcular conociendo su Ancho y Largo, por lo cual no se define como un atributo básico de la caja, sino como un atributo derivado, como se muestra en la Figura

Rectángulo

Ancho

Largo

/ Area

{ Ancho = Largo }

{ Area = Ancho X Largo }

Operaciones

- Las operaciones son funciones o transformaciones que se aplican a todos los objetos de una clase particular. La operación puede ser una acción ejecutada por el objeto o sobre el objeto.
- Ejemplo: Arrojar, atrapar, inflar, y patear, son operaciones para la clase pelota. Abrir, cerrar, ocultar, y dibujar, son operaciones para la clase ventana.

Operaciones

- No se debe utilizar el mismo nombre para operaciones que tengan un significado totalmente diferente.
- Ejemplo: No se debe utilizar el mismo nombre invertir para la operación de invertir una figura y para la operación de invertir una matriz, ya que son operaciones totalmente diferentes. Invertir una figura es rotarla por 180 grados, mientras que invertir una matriz M es encontrar su inverso N , para que $M \times N = 1$. Se deben usar nombres diferentes, como invertir-figura e invertir-matriz.
- Las operaciones pueden tener argumentos, o sea, una lista de parámetros, cada uno con un tipo, y pueden también devolver resultados, cada uno con un tipo. Las operaciones se incorporan en la tercera sección de la clase, como se muestra en la Figura



Nombre de la Clase
Lista de Atributos
Lista de Operaciones

Operaciones

- Ejemplo: En la Figura se muestra tres clases, Persona, Universidad y Rectángulo, conteniendo atributos y operaciones. Trabajar y Votar son operaciones en Persona; Enseñar y Graduar son operaciones en Universidad; mientras que Dibujar y Borrar son operaciones en Rectángulo.

Persona
Nombre Edad
Trabajar() Votar()

Universidad
Nom bre Dirección
Enseñar() Graduar()

Rectángulo
Ancho Largo
Dibujar() Borrar()

Operación: consulta

- Las operaciones que no tienen efectos secundarios, las cuales no cambian los valores de los atributos en el objeto, se les llama consultas (query). Una consulta es una operación que no modifica al objeto. Las consultas por lo general devuelven valores de atributos, básicos o derivados, y pueden o no tener argumentos.
 - Ejemplo: Para una clase Rectángulo conteniendo los atributos Ancho y Largo pueden existir operaciones de consulta para leer los valores del Ancho y Largo sin afectarlos.
 - Una consulta se puede definir para la lectura de atributos derivados, ya que los atributos derivados no cambian los valores de los atributos del objeto.
-
- Ejemplo: En la clase Rectángulo el atributo Área puede ser leído por medio de una consulta implementada como la multiplicación de los valores de los atributos Largo y Ancho.
 - La notación para las operaciones de consulta es la misma que para las operaciones, la diferencia es sólo en su comportamiento. Por regla general estas consultas no se incluyen de forma explícita en el modelo de objetos.
 - Estas se agregan durante la etapa de diseño ya que son operaciones bastante básicas.

Operación: acceso

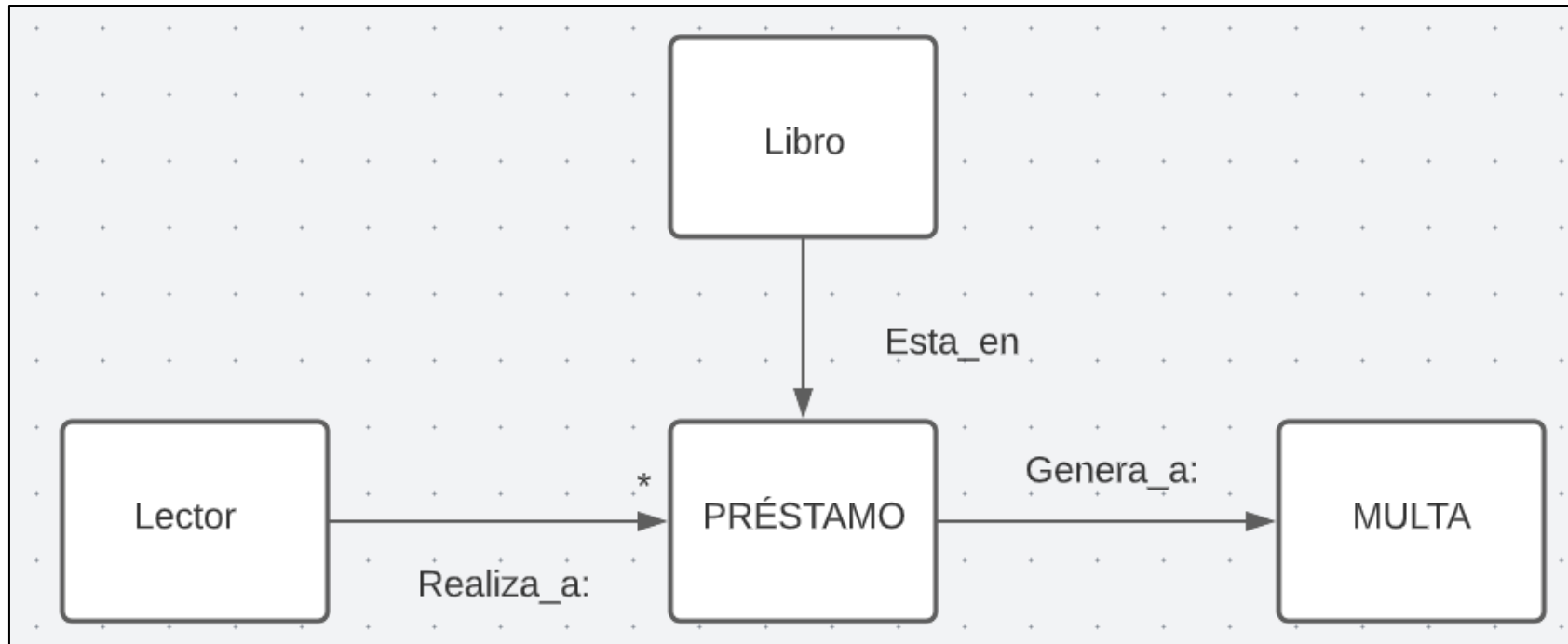
Accesos. Se puede definir operaciones de acceso para leer o escribir los atributos de un objeto. Si el acceso es hecho para leer solamente, sin afectar los valores de los atributos, entonces el acceso se considera también una operación de consulta. Se utiliza la notación de punto para indicar, dentro de la operación de acceso, el acceso a un atributo: "objeto.atributo".

Ejemplo: para acceder el nombre de una persona se utiliza:
persona.nombre

La notación para las operaciones de acceso es la misma que para las operaciones, la diferencia es sólo en su comportamiento. Por regla general estas operaciones de acceso no se incluyen de forma explícita en el modelo de objetos. Estas se agregan durante la etapa de diseño ya que son operaciones bastante básicas.

Operación: método

- Un método es una operación que implica las funciones para las cuales la clase esta diseñada en el contexto del sistema.
- Ejemplo:



IDENTIDAD

- Los objetos se distinguen por su propia existencia, su identidad, aunque internamente los valores para todos sus datos sean iguales.. Todos los objetos se consideran diferentes.
- Ejemplo: Si tenemos una biblioteca llena de libros, cada uno de esos libros, como La Iliada, Hamlet, La Casa de los Espíritus, etc., se consideran e identifican como objetos diferentes. Dos manzanas aunque sean exactamente del mismo color y forma, son diferentes objetos. Los objetos tienen un nombre que puede no ser único.
- Ejemplo: Pueden existir múltiples copias de un solo libro, lo cual requiere identificadores especiales para distinguir entre diferentes objetos con propiedades similares, como el código del libro en la biblioteca.
 - Título: el nombre del libro
 - Autor: el nombre del autor del libro
 - Año de publicación: el año en que se publicó el libro
 - Género: la categoría a la que pertenece el libro (por ejemplo, ficción, no ficción, romance, etc.)
 - ISBN: el número de identificación único del libro
 - Estado: el estado actual del libro en la biblioteca (por ejemplo, disponible, prestado, en reparación, etc.)
 - Ubicación: la ubicación física del libro en la biblioteca (por ejemplo, en la sección de ciencia ficción, en el estante B, etc.)



IDENTIDAD

- Los objetos necesitan un identificador interno único cuando son implementados en un sistema de computación para acceder y distinguir entre los objetos. Estos identificadores no deben incluirse como una propiedad del objeto, ya que solo son importantes en el momento de la implementación.
- Ejemplo: Los diferentes personas se distinguirían internamente dentro de una computadora por los identificadores



FIRMA

- La firma de una operación se define por el tipo y número de argumentos y el tipo de resultados que devuelve.
- Ejemplo.

```
int Calcular_area (int alto, int ancho)
```



POLIMORFISMO

- Una operación también se considera polimórfica si ésta se implementa en una misma clase por diferentes métodos, con diferente número y tipo de argumentos.

```
int Calcular_area (int alto, int ancho){  
    return area=alto*ancho  
}
```

```
float Calcular_area (float base, float altura){  
    return área= base*altura/2;  
}
```

```
float Calcular_area (float radio){  
    return área= PI*radio**  
}
```



HERENCIA

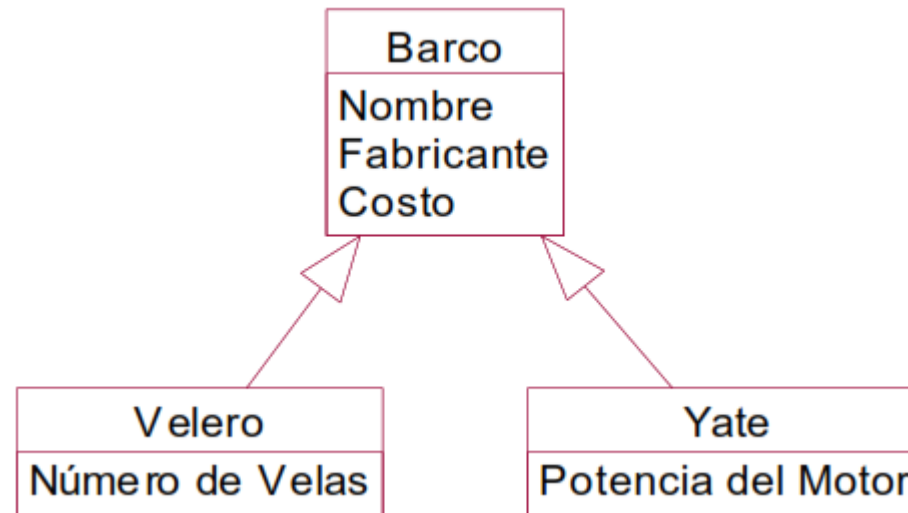
- Las clases con atributos y operaciones comunes se pueden organizar de forma jerárquica mediante la herencia. La herencia es una abstracción importante para compartir similitudes entre clases, donde todos los atributos y operaciones comunes a varias clases se pueden compartir por medio de la superclase, una clase más general. Las clases más refinadas se conocen como las subclases.
- Ejemplo: Las Impresoras Láser, de Burbuja, y de Matriz, son todas subclases de la superclase Impresora. Los atributos generales de una Impresora son el Modelo, Velocidad, y Resolución, mientras que sus operaciones son Imprimir y Alimentar.



HERENCIA

- Herencia es una relación "es-una" o "es-un-tipo-de" entre las clases las más refinadas y más generales por lo que tampoco se debe escribir el nombre de la relación.

Ejemplo: Impresora Láser es una Impresora.



MENSAJES

- En la programación orientada a objetos, los objetos descritos anteriormente se comunican a través de señales o mensajes, siendo estos mensajes los que hacen que los objetos respondan de diferentes maneras, por ejemplo un objeto en Windows como una ventana de alguna aplicación , puede cerrarse, maximizarse o restaurarse (métodos) de acuerdo al mensaje que le se enviado.
- En otras palabras, un mensaje es una comunicación dirigida a un objeto, que le ordena que ejecute uno de sus métodos pudiendo o no llevar algunos parámetros.



ENCAPSULAMIENTO

- Cada objeto está aislado del exterior, esta característica permite verlo como una caja negra, que contiene toda la información relacionada con ese objeto. Este aislamiento protege a los datos asociados a un objeto para que no se puedan modificar por quien no tenga derecho a acceder a ellos.
- Permite manejar a los objetos como unidades básicas, dejando oculta su estructura interna.
- Ejemplo: funciones de uso interno, en caso de una facturación `calcular_total` y en ese caso se debe hacer el calculo de un descuento, esa operación es interna ya que desde ninguna instancia de una clase (objeto) se debe realizar una solicitud directa a calcular un descuento, sino que esa función se llama internamente desde la operación `calcular_total`.



MÉTODOS.

- Un método en Java es una sección de código que se utiliza para realizar una tarea específica. Es una colección de declaraciones de variables, sentencias y expresiones que se agrupan juntas para realizar una tarea. Los métodos pueden tomar parámetros de entrada y devolver valores como salida.

```
public int sumar(int num1, int num2) {  
    int resultado = num1 + num2;  
    return resultado;  
}
```

- Para llamar a este método y obtener el resultado de la suma, se podría escribir algo así:

```
int resultadoSuma = sumar(5, 3);  
System.out.println(resultadoSuma); // Imprime "8"
```



OPERACIONES

- Es una acción individual que se realiza en los datos. En otras palabras, son acciones individuales que pueden realizarse en los datos.

```
# Operación de suma  
a = 5  
b = 3  
resultado = a + b  
print(resultado) # Imprime "8"
```



ASOCIACIÓN

- Una asociación describe la relación entre clases y es por naturaleza bidireccional.
- Por lo general, el nombre de la asociación implica una dirección, pero puede ser invertida para mostrar la dirección opuesta. Cualquiera de las dos direcciones es igualmente correcta, aunque se acostumbra a leer de izquierda a derecha y de arriba a abajo.
- Ejemplo: El opuesto de "estudiante estudia en la universidad" sería "universidad da estudios a estudiante". Las dos direcciones se refieren a la misma asociación.



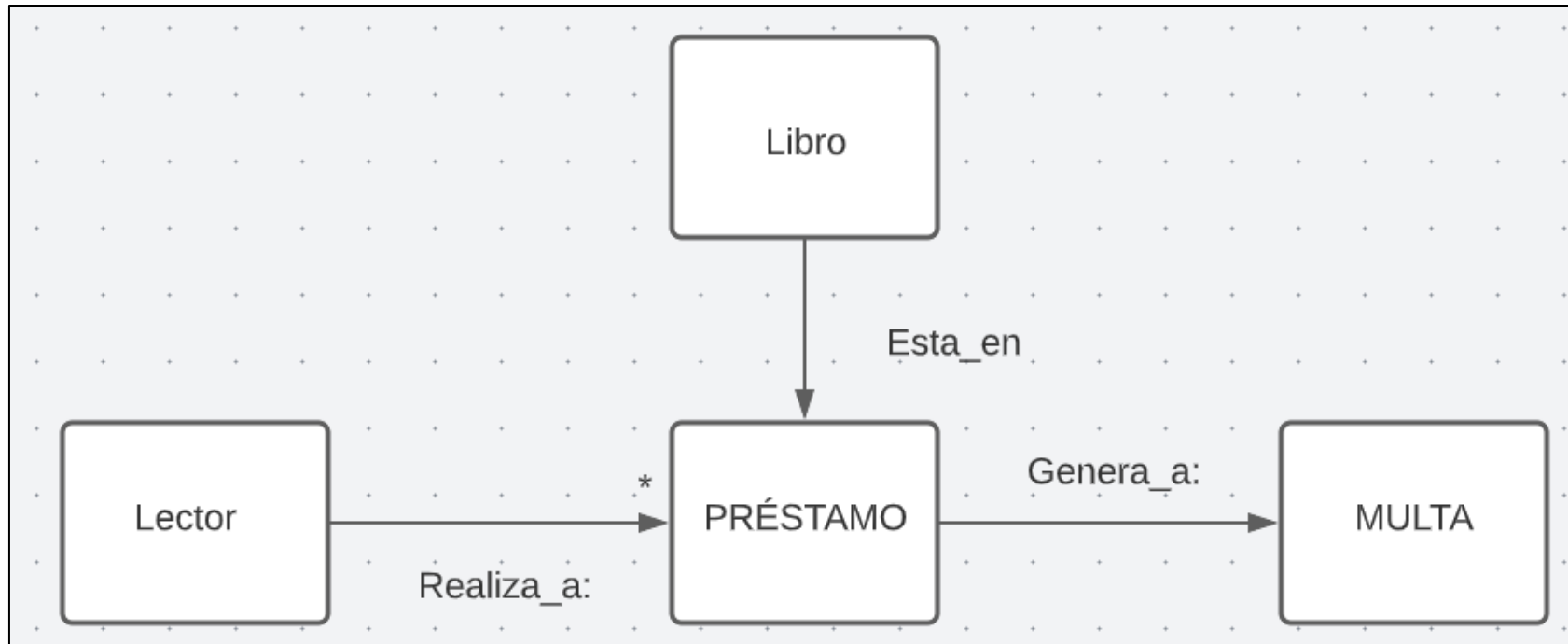
TIPOS DE ASOCIACIONES

- Asociaciones simples
- Asociaciones reflexivas
- Asociaciones ternarias
- Ensamblados: Agregación y composición
- Herencia



ASOCIACIONES SIMPLES

- Es una asociación que se da entre dos clases y que lleva un nombre para indicar cuál es la relación que hay entre ellas, son de grado 2.



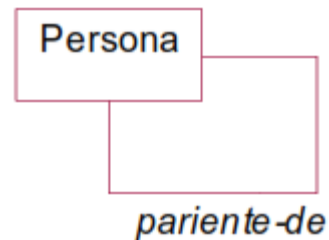
GRADO DE LA ASOCIACIÓN

- El grado de una asociación se determina por el número de clases conectadas por la misma asociación. Las asociaciones pueden ser binarias, ternarias, o de mayor grado. Las asociaciones se consideran binarias si relacionan solo dos clases.
- Ejemplo: La asociación entre Persona e Instituto es una asociación binaria.
- Las asociaciones pueden ser de mayor grado si relacionan a la misma vez más de dos clases. Aparte de relaciones binarias, lo más común son relaciones ternarias (entre tres clases), relaciones de más alto nivel son mucho menos comunes. Mientras el grado de una relación aumenta, su comprensión se dificulta, y se debe considerar partir las relaciones en varias relaciones binarias.
- Ejemplo: Puede existir una relación ternaria entre Estudiante, Profesor, y Universidad donde "un estudiante estudia con un profesor en una universidad".



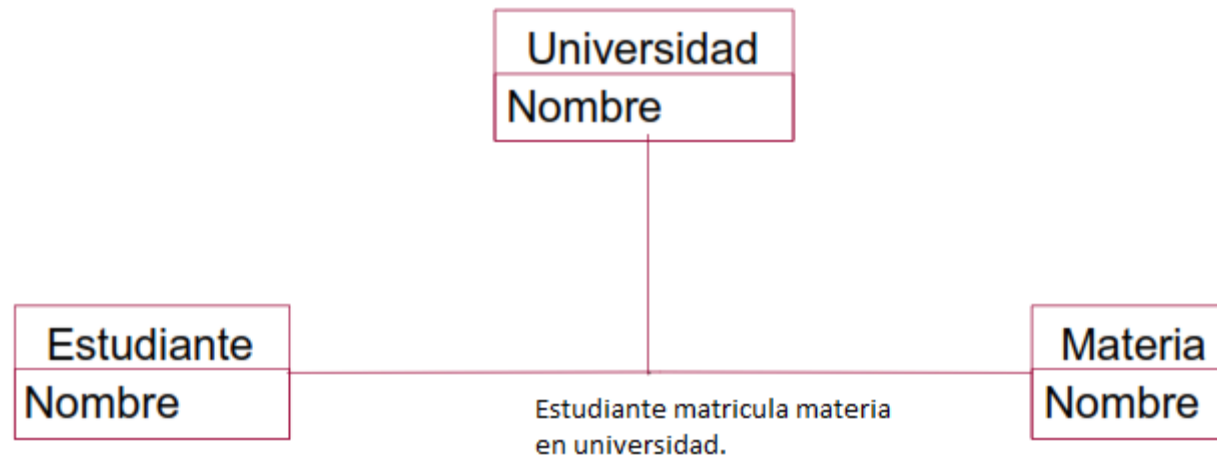
ASOCIACIONES REFLEXIVAS

- Las asociaciones pueden ser reflexivas, relacionando distintos objetos de una misma clase.
- Ejemplo: Para una clase persona puede existir una asociación pariente que describe que dos objetos de tipo persona, como Juan Pérez y Laura Pérez son parientes.
- El grado de una asociación reflexiva puede ser binario, ternario, o de mayor grado, dependiendo del número de objetos involucrados.



ASOCIACIONES TERNARIAS

- Es una asociación conformada por tres clases que son fundamentales para la composición de la relación. Debe tener un nombre que indique cual es la asociación. Es de grado 3



ROLES

- El rol describe el papel que juega cada extremo de una asociación. Una asociación binaria tiene dos roles, uno en cada extremo, los cuales pueden tener un nombre diferente cada uno.



MULTIPLICIDAD

- La multiplicidad (cardinalidad) de una asociación especifica cuantas instancias de una clase se pueden relacionar a una sola instancia de otra clase.
 - Ejemplo: En el caso de Estudiante y Universidad, la multiplicidad está dada por el número de estudiantes que puedan estudiar en una sola universidad. En otras palabras, muchos objetos de tipo Estudiante se conectan a un solo objeto de tipo Universidad.
- Es necesario decidir la multiplicidad para cada clase en una asociación, o sea dos multiplicidades por cada relación binaria, una para cada extremo de la relación.
 - Ejemplo: En la relación estudia-en es necesario definir la multiplicidad para el Estudiante y para la Universidad. La multiplicidad restringe una asociación limitando el número de objetos que pueden relacionarse a un objeto particular. Ejemplo: En la asociación estudia-en se puede restringir el número de estudiantes que pueden estudiar en una universidad.



MULTIPLICIDAD — UNO A UNO

- "uno-uno": donde dos objetos se relacionan de forma exclusiva, uno con el otro. Ejemplo:

Cada Universidad tiene un Rector, y cada Rector rige una Universidad.



MULTIPLICIDAD — UNO A MUCHOS

- "uno-muchos": donde uno de los objetos pueden estar ligado a muchos otros objetos.

Ejemplo: Muchos Estudiantes pueden estudiar en una Universidad, y una sola Universidad da estudios a cada Estudiante.



MULTIPLICIDAD – MUCHOS A MUCHOS

- "muchos-muchos": donde cada objeto de cada clase puede estar ligados a muchos otros objetos.

Ejemplo: Muchos Estudiantes pueden estudiar en varias Universidades.

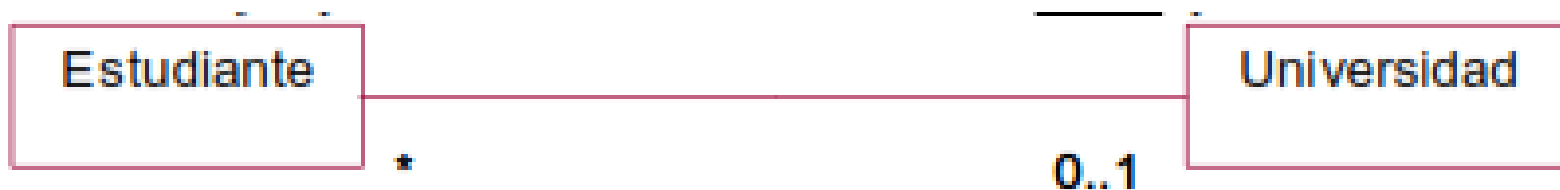


MULTIPLICIDAD – OPCIONAL

- La notación para representar una relación opcional, donde la multiplicidad es "uno" o "cero", describiendo una relación opcional, 0 o 1. Esto significa que dos objetos pueden o no estar conectados, y si lo están corresponden a una multiplicidad de 1.



- Ejemplo: El caso de muchos Estudiantes que pueden o no atender a una sola Universidad



MULTIPLICIDAD - USO

- La multiplicidad se incluye en el diagrama de clases únicamente. La multiplicidad para relaciones de mayor grado es más compleja, volviéndose esta notación un poco ambigua para relaciones de mayor orden ya que no sabría cómo leerse la relación.



EJERCICIO - SISTEMA DE PRÉSTAMO Y DEVOLUCIÓN DE LIBROS

- Una biblioteca tiene copias de libros. Estos libros se caracterizan por su nombre, tipo (ingeniería, literatura, informática, historia, ...), editorial, año y autor.
- Los autores se caracterizan por su nombre, nacionalidad y fecha de nacimiento.
- Cada copia tiene un identificador, y puede estar en la biblioteca, prestada, con retraso o en reparación.
- Los lectores pueden tener máximo de 3 libros en préstamo.
- Cada libro se presta un máximo de 30 días, y por cada día de retraso, se impone una “multa” de dos días sin posibilidad de coger un nuevo libro.



EJERCICIO — SISTEMA DE REGISTRO DE EMPLEADOS Y CLIENTES DE EMPRESAS

- Una aplicación necesita almacenar información sobre diferentes empresas, sus empleados y sus clientes. Ambos (empleados y sus clientes) se caracterizan por su nombre y edad.
- Los empleados tienen un sueldo bruto.
- Los empleados que son directivos tienen una categoría, así como un conjunto de empleados subordinados.
- De los clientes, además se necesita conocer su teléfono de contacto.
- La aplicación necesita mostrar los datos de empleados y clientes.



EJERCICIO - SISTEMA PARA UN ROBOT MODULAR

- El robot estará compuesto por varios módulos entre los que se encuentran: rotación, extensión, helicoidal, cámara.
- Los módulos podrán ser dinámicos (capaces de moverse: rotación, extensión, helicoidal) o estáticos (no se pueden mover: cámara).
- Los módulos tendrán un identificador (1-255) y unas dimensiones (largo, ancho y alto, entre 1 y 200mm)
- Los módulos estarán compuestos de un sistema de control y un sistema de comunicación.
- Los módulos tendrán:
 - ✓ Motores (1 o 2).
 - ✓ Un parámetro que es el tipo de movimiento que puede realizar.
 - ✓ Una función que es moverse (con parámetro que indica el tipo de movimiento)
- Los módulos estáticos podrán tener sensores (de 0 a 5).
- El sistema de control utiliza el sistema de mensajes para comunicarse. Los módulos pueden enviar y recibir mensajes de/hacia el usuario y otros módulos.
- También utiliza motores para moverse y los sensores y los sensores para captar información del medio.

