UCA

Programación Orientada a Objetos I

UML – CLASES

Clases

- Una clase es una abstracción de las cosas que forman parte del vocabulario
- Una clase no es un objeto individual, representa a un conjunto de objetos
- Una clase es una descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, relaciones y semántica
- Son los bloques de construcción más importantes de cualquier sistema OO
- Se pueden utilizar para representar cosas de hardware, software o conceptos
- Tienen responsabilidades
- Se representan con un rectángulo

Clases

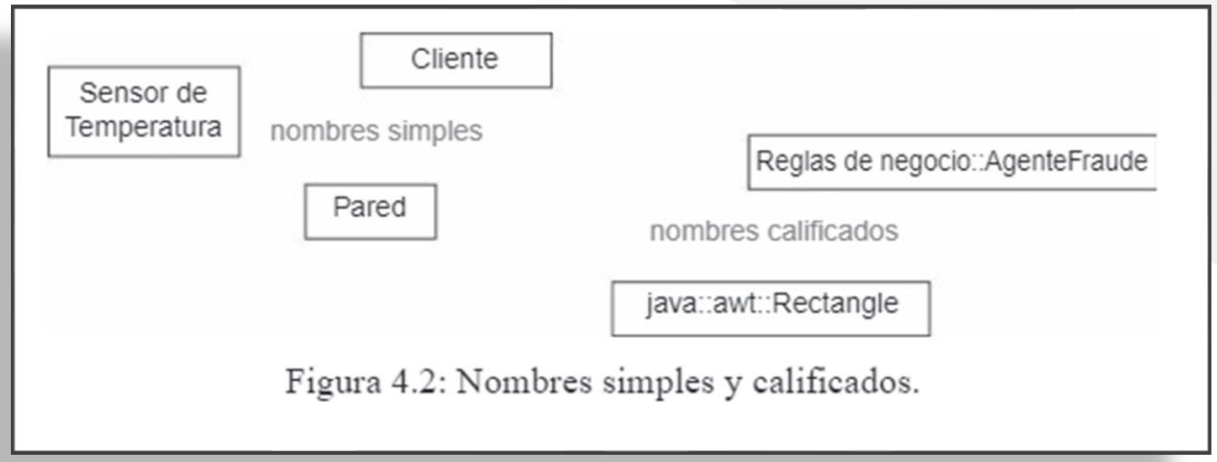


Figura 4.2: Nombres simples y calificados.

Nombre:
La distingue de otras clases

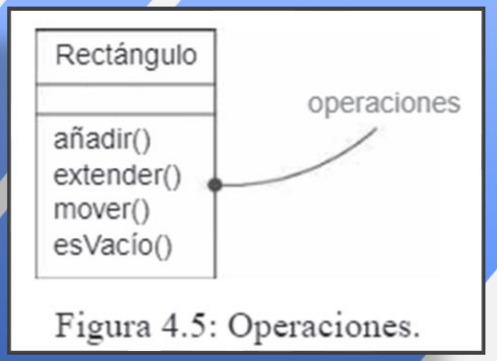


Figura 4.5: Operaciones.

Atributos:
Propiedades que tendrán los objetos de dicha clase

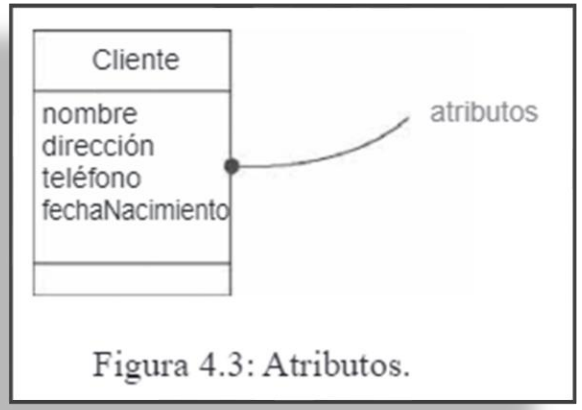


Figura 4.3: Atributos.

Operaciones:
Servicio que puede prestar el objeto

Nombres

- **Nombres Sustantivos**: No usar verbos o frases del estilo "ProcesarDatos“.
- **Singular**: Representan una entidad específica. "**Cliente**" en lugar de "Clientes"
- **Especificidad**: Ser específico. En lugar de "Cosa" u "Objeto", usar "**Producto**" o "**Persona**"
- **Evita abreviaciones excesivas**: Si es necesario, utiliza abreviaciones comunes y conocidas
- **Descriptivo**: Que el nombre de la clase describa claramente su función y propósito

Ejemplos de nombres de clases bien nombradas:

- "Pedido", "Empleado", "Factura", "Automóvil", etc.

Atributos

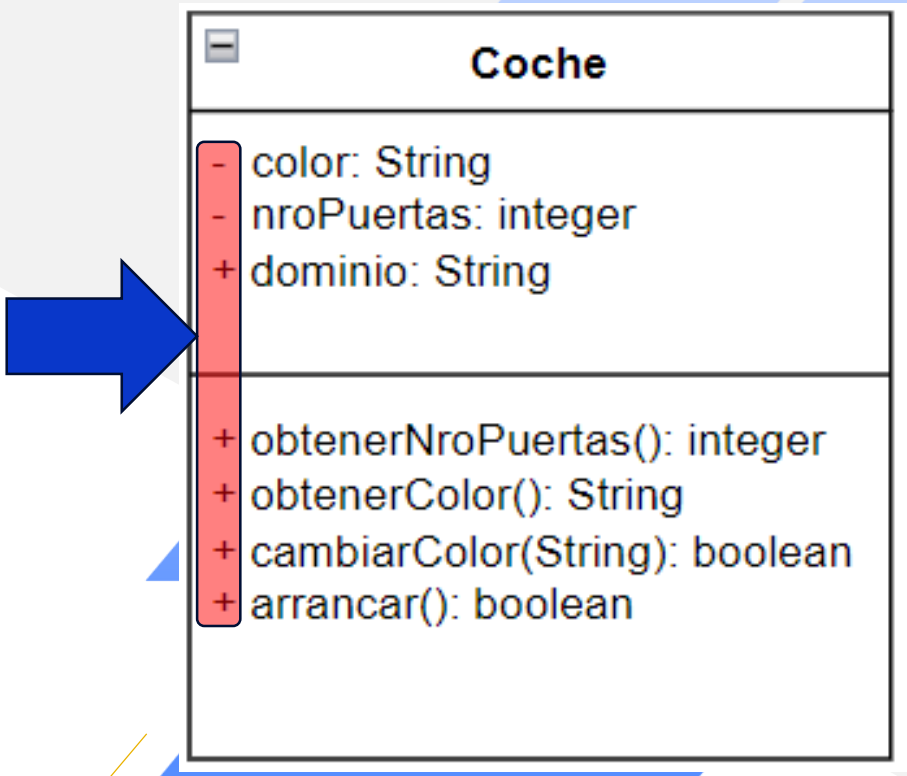
- Una clase puede tener cualquier cantidad de atributos o ninguno
- Representa una propiedad del elemento modelado
- Se listan en un compartimento debajo del nombre
- Se pueden representar:
  - Solo con su nombre
  - Con su nombre y tipo o clase
  - Con su nombre, tipo y valor inicial por defecto
- No necesariamente se deben mostrar todos los atributos (seguramente no alcance el lugar). Pueden mostrarse los más importantes. También pueden ponerse puntos suspensivos para dar a entender que hay más

Operaciones

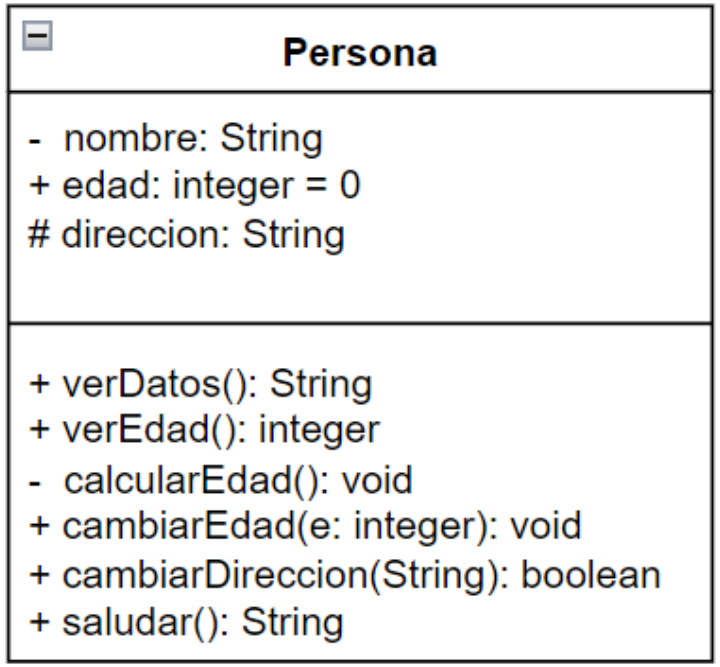
- Es la implementación de un servicio que puede ser requerido a cualquier objeto
- Es una abstracción de algo que puede hacer un objeto
- Puede tener cualquier número de operaciones o ninguna
- Se listan en un compartimento debajo de los atributos
- Se pueden representar:
  - Solo con su nombre y los paréntesis
  - Con su nombre, paréntesis y el tipo de retorno
  - Con su nombre, parámetros y su tipo entre paréntesis
  - Con su nombre, parámetros y su tipo entre paréntesis y, y tipo de retorno
- No necesariamente se deben mostrar todos las operaciones, algunas se suponen.

Clases: Visibilidad

- Determina el nivel de encapsulamiento de los elementos de una clase.
  - (-) Privado : Los atributos/operaciones son visibles solo desde la propia clase.
  - (+) Los atributos/operaciones públicos son visibles a otras clases (cuando se trata de atributos se está transgrediendo el principio de encapsulación)
  - (#) Los atributos/operaciones protegidos están visibles para la propia clase y para las clases derivadas de la original



Ejemplo



Sugerencias y consejos:

- Mostrar solo aquellos atributos (o propiedades) y operaciones (o métodos) que sean importantes o diferenciales
- Organizar las listas de atributos y operaciones de acuerdo a su categoría

Al modelar una clase hay que recordar que se debe corresponder con una abstracción tangible o conceptual del dominio

Responsabilidad

- Es un contrato u obligación de la clase
- Los atributos y operaciones son características y medios por el cual las clases cumplen con su misión (su responsabilidad)
- Las responsabilidades pueden expresarse como conjuntos de atributos y/o métodos, o como un compartimento separado (muchas veces son implícitas y no se expresan).
- La responsabilidad de una clase es la visión de más alto nivel que podemos tener de ella.
- Está más cerca de su razón de ser que de al implementación para poder serlo

Herramientas

Hay muchas herramientas para dibujar clases:

- Lápiz y papel: Si… muchas veces es mejor que el software
- Lucidchart
- GetMind
- Microsoft Visio
- Enterprise Architect
- MagicDraw
- Draw.io (instalado en el laboratorio):
  - Se puede utilizar online (sin descargar nada)
  - Se puede utilizar de manera portable (descargado pero sin instalar)
  - Se puede instalar
  - Es de uso libre

Técnicas de modelado de clases

- Identificar aquellas cosas que se utilizan o se necesitan utilizar
- Identificar y separar las responsabilidades de cada clase
- Es mejor un conjunto reducido de responsabilidades pero efectivas
- Proporcionar los atributos y operaciones de cada una necesarios
- Mostrar solo lo que hace falta para comprender la clase en su contexto y en su nivel de abstracción

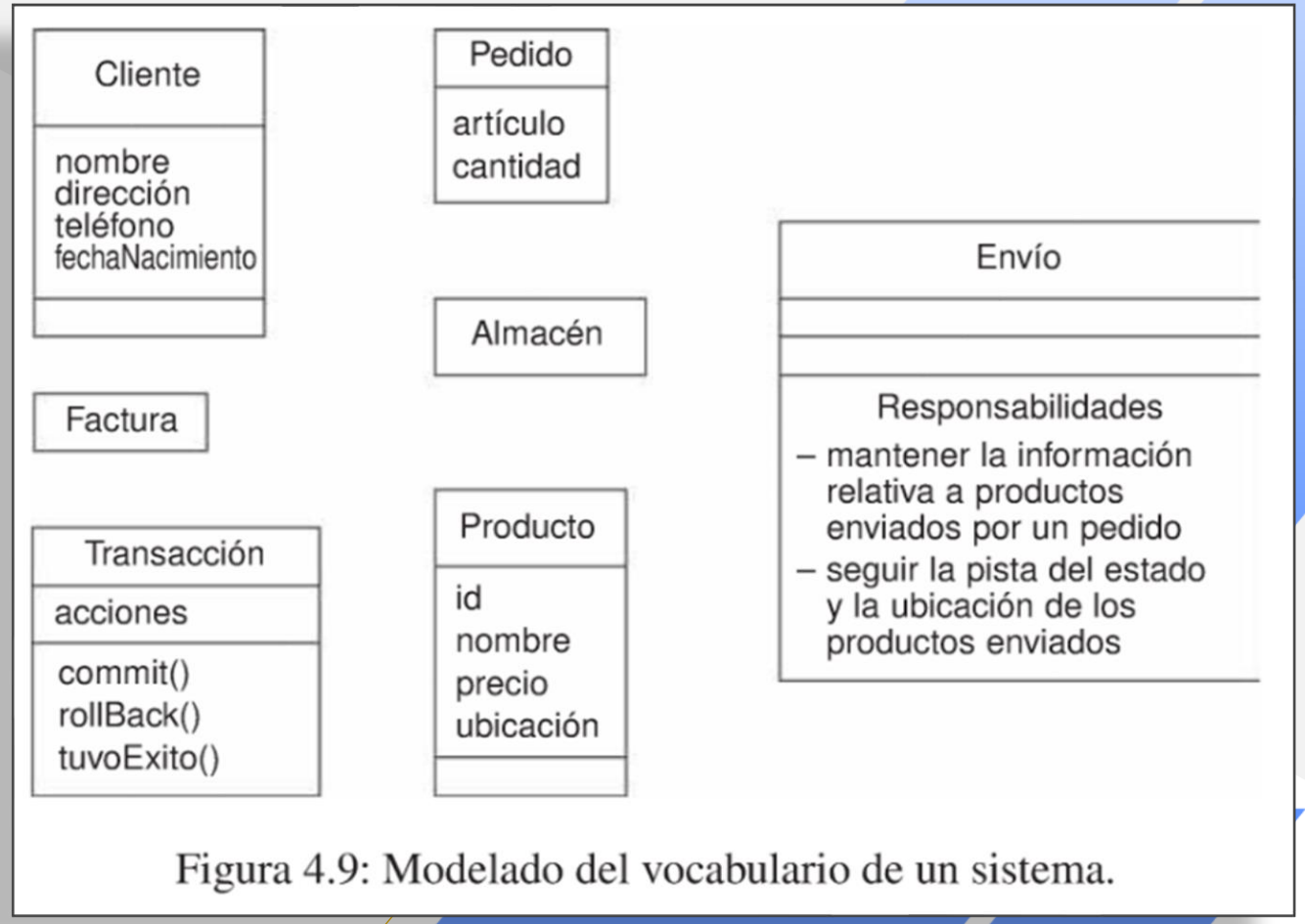


Figura 4.9: Modelado del vocabulario de un sistema.

Ejercicio 1: Biblioteca

Imagina que estás trabajando en el diseño de un sistema de gestión para una biblioteca.

El sistema debe realizar un seguimiento de los libros disponibles en la biblioteca. Cada libro tiene un título, un autor, un número de identificación único (ISBN), una fecha de adquisición y una ubicación específica en el estante.

Los libros se prestan a los socios de la biblioteca, quienes tienen algunos datos personales como nombre, un número de socio, una dirección, un número de teléfono y una dirección de correo electrónico.

Los socios pueden tomar prestados libros de la colección. Lógicamente necesitamos llevar un registro de los préstamos, que incluye la fecha de préstamo, la fecha de vencimiento y la información sobre el libro prestado.

Para mantener la colección actualizada, la biblioteca necesita gestionar las devoluciones de libros y, en caso necesario, imponer multas por retrasos en la devolución.

A partir de esto, hay que identificar y nombrar las clases que serían necesarias para implementar cada aspecto del sistema, junto con sus atributos y métodos principales.