

Clases y Objetos

- La codificación de clases con sus respectivos métodos y atributos suele requerir de un análisis del problema.
- Encontrar las clases que componen un sistema y sus comportamientos como las relaciones que existen entre las mismas es esencial para el adecuado diseño del sistema y su posterior implementación (codificación).
- Desde el enfoque OO además se persiguen otros objetivos: modularización (división de la funcionalidad), reutilización, facilidad de mantenimiento, por ejemplo.

Clases y Objetos

- Proceso de análisis (modelado), puede no ser necesario o pasado por alto en problemas muy simples, pequeños y acotados. Pero resulta necesario en problemas de dimensión mediana - grande y es complejo.
- Además, la mayoría de las veces, el desarrollador no está familiarizado con el problema, por lo que se enfrenta a un nuevo mundo, con su propio vocabulario, reglas y procedimientos.
- Existen técnicas de modelado (asignaturas Requerimientos – Análisis y Diseño), el proceso de identificar clases y su comportamiento, es un proceso iterativo e incremental, en el cual los pasos no son estrictos o taxativos.
- El procedimiento implica hacer preguntas, buscar relaciones, acudir a ejemplos (escenarios), revisar lo hecho para hacer correcciones y mejoras.
- Aunque no existe una fórmula (del estilo matemático) para encontrar las clases de un sistema, se pueden establecer algunas pautas que ayudarán a esta tarea.

Pautas para analizar clases, atributo y métodos.

A partir de un enunciado (descripción del problema), generalmente llamado especificación de requisitos o requerimientos:

- **Identificar la función de los vocablos como: sustantivos, adjetivos y verbos.**
 - a) Los sustantivos son posibles clases o atributos (no ambos)
 - b) Los adjetivos son posibles atributos.
 - c) Los verbos o acciones indican posibles operaciones o métodos.

Pautas para analizar clases, atributo y métodos.

- **Comenzar el análisis por los vocablos sustantivos.**
 - a) Algunos sustantivos tienen mayor relevancia que otros, porque se reiteran en el enunciado y/o agrupan otros en torno asimismo. Estos suelen ser clases candidatas.
 - b) Algunos sustantivos y adjetivos, describen propiedades, características, cualidades y/o estados de las clases candidatas. Estos son atributos candidatos. A veces se usan dos o más adjetivos para referir al mismo atributo, son posibles valores.
 - c) Analizar instancias, si NO hay más de una instancia para la clase candidata, descartar. Cada instancia tiene valores diferentes para cada atributo candidatos.
 - d) Algunos sustantivos refieren a los mismos objetos (sinónimos), elegir uno solo, el más representativo.

Pautas para analizar clases, atributo y métodos.

- **Métodos básicos.**
 - a) No siempre aparecen en el enunciado pero los atributos por lo general involucran dos métodos (set/get).
 - b) Algunas veces solo uno. Los datos productos del cálculo de los atributos no se configuran como atributos sino como métodos.
- **Una clase candidata, sin atributos y métodos debe ser descartada.**

Ejemplo 1

- Una ruta es un camino que tiene 2 o más carriles para que los vehículos circulen. Una ruta tiene una longitud que se mide en kilómetros. Una ruta comienza y finaliza en alguna localidad. Una ruta se identifica por un número. Una ruta puede estar asfaltada o no. Una ruta es nacional o provincial.
- Sustantivos: ruta – camino - carriles – vehículos – longitud – kilómetros – localidad – número –
- Adjetivos: nacional – provincial - asfaltada

Paso 1: Sustantivos	Paso 2: analizar relaciones e importancia	Paso 3: definir	Paso 4: analizar tipos
Ruta	Se repite 6 veces	Clase candidata	
Carriles	Describe a ruta	Atributo candidato	Entero
Longitud	Describe a ruta	Atributo candidato	Entero
Localidad	Describe a ruta	Atributo candidato	Cadena
Numero	Describe a ruta	Atributo candidato	Entero
Kilómetros	Unidad de medida de longitud	Descartar	-
Camino	Sinónimo de ruta	Descartar	

Ejemplo 1

- Podemos decir que numero describe a localidad? NO, la relación es con ruta, tanto de número como de localidad. Idem con el resto de los atributos candidatos.

Paso 1: Adjetivo	Paso 2 : analizar relaciones	Paso 3	Paso 4: analizar tipo
Asfaltada	Describe a ruta	Atributo candidato	Booleano
Nacional	Describe a ruta	Atributo candidato	Son excluyentes y valores en sí mismos. Puede usarse cadena o carácter (‘N’o ‘P’)
Provincial	Describe a ruta	Atributo candidato	

- Analizar instancias

N°	Carriles	Kilómetros	Inicio	Fin	Asfaltada	Jurisdicción
3	2	3060	CABA	Paso Fronterizo Integración Austral	si	N
6	2	180	Angel Etcheverri	Zarate	si	P

Ejemplo 1

```
class Ruta {  
    private int numero;  
    private int carriles;  
    private int longitud;  
    private String inicio,  
    private Boolean asfaltada;  
    private char juridiccion,  
  
    Ruta() {  
    ...}  
    int getNumero()  
        { return numero;}  
    void setNumero(int n)  
        { numero = n; }  
  
}
```

```
-----  
Ruta r2;  
r2 = new Ruta();  
r2.setNumero(2);
```


REUSO EN POO

Sandra Casas

2017

¡Qué es y para se hace?

- Reusar código es escribirlo una vez y usarlos muchas veces (en forma total o parcial).
- Busca reducir el esfuerzo y los costos del desarrollo de software.
- Técnicas: API – Librerías – Componentes... varían según el paradigma y lenguaje.
- MODULARIZACION.

Reuso en POO

- La POO promueve la reutilización de clases.
- Se trata de usar o adaptar clases ya hechas.
- Hay tres formas fundamentales de reutilizar clases:
 - Mediante composición/agregación
 - Mediante uso/dependencia
 - Mediante herencia

Composición/agregación

- Consiste en formar objetos a partir de otros.
- Cuando entre dos objetos de distintas clases se produce la relación:

..... ***“se compone de”***

..... ***“es parte de”***

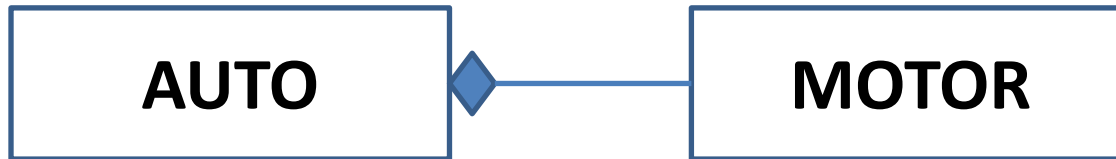
.... ***“tiene un”***

- estamos en presencia de una relación de agregación.

Ejemplo

- un auto tiene un precio, (real, simple). Pero un auto tiene un motor, y un motor tiene a su vez una serie de particularidades, es más no todos los autos tienen el mismo motor. Además los motores no solo se usan en los autos, también en barcos, aviones, electrodomésticos, etc. Es decir, motor es una clase. La relación se puede describir:
 - ✓ *un auto “se compone de” un motor*
 - ✓ *un auto “tiene” un motor*
 - ✓ *un motor “es parte” un auto*

Modelado e Implementación



(2)

```
class Auto
{
    String marca;
    Motor mot;
    -----

    Auto()
    { //código }

    ---
}
```

(1)

```
class Motor {
    string nro_serie;
    int valvulas;
    int cilindrada;

    Motor (...) {}
    // set-get
}
```

- La agregación implica que en la clase Auto hay al menos un atributo de tipo Motor.
- En algún método de Auto mot debe ser creado, ya que esta declarado como atributo.
- Significa que desde cualquier método de Auto se pueden enviar mensajes al objeto mot.
- Otras clases pueden usar Motor.

Uso/dependencia

- Cuando entre dos objetos de distintas clases se produce la relación:

.... ***“usa”***

.... ***“depende”***

- estamos en presencia de una relación de dependencia.

Ejemplo

- un auto tiene 4 ruedas, y si es necesario cambiar una rueda, Auto deberá tener un método cambiarRueda.
- Para poder cambiar una rueda es necesario usar un cricket, que no es un componente de un auto, pero se usa para cambiar la rueda.
- un objeto de la clase Auto, usará un objeto de la clase Cricket, cada vez que se ejecute el método “cambiarRueda”.
- La relación se puede describir: *un auto “usa” un cricket*

Modelado e implementación



(1)
`class Cricket {`

`}`

(2)
`class Auto {`
`---`
`void cambiarRueda(Cricket c) {`
`//código`
`}`

`void cambiarRueda(Cricket c) {`
`Cricket c = new Cricket()`
`//código`
`}`
`}`

una clase A tiene la posibilidad de enviar mensajes a un objeto de una clase B, que recibe por parámetro o crea como variable local.