



Paradigma Orientado a Objetos

**Módulo 01: Objeto, mensaje, métodos.
Intro a method lookup.
Intro a referencias.**

**por Fernando Dodino
Versión 2.1
Agosto 2017**

Distribuido bajo licencia [Creative Commons Share-a-like](https://creativecommons.org/licenses/by-sa/4.0/)



Indice

[1 Objeto](#)

[1.1 ¿Qué es un objeto?](#)

[1.2 Objetos conceptuales](#)

[2 Composición de un objeto](#)

[3 Mensaje](#)

[4 Ambiente](#)

[5 Referencias](#)

[6 Mensaje y método](#)

[6.1 Method lookup](#)

[6.2 Acciones y preguntas](#)

[7 Resumen](#)

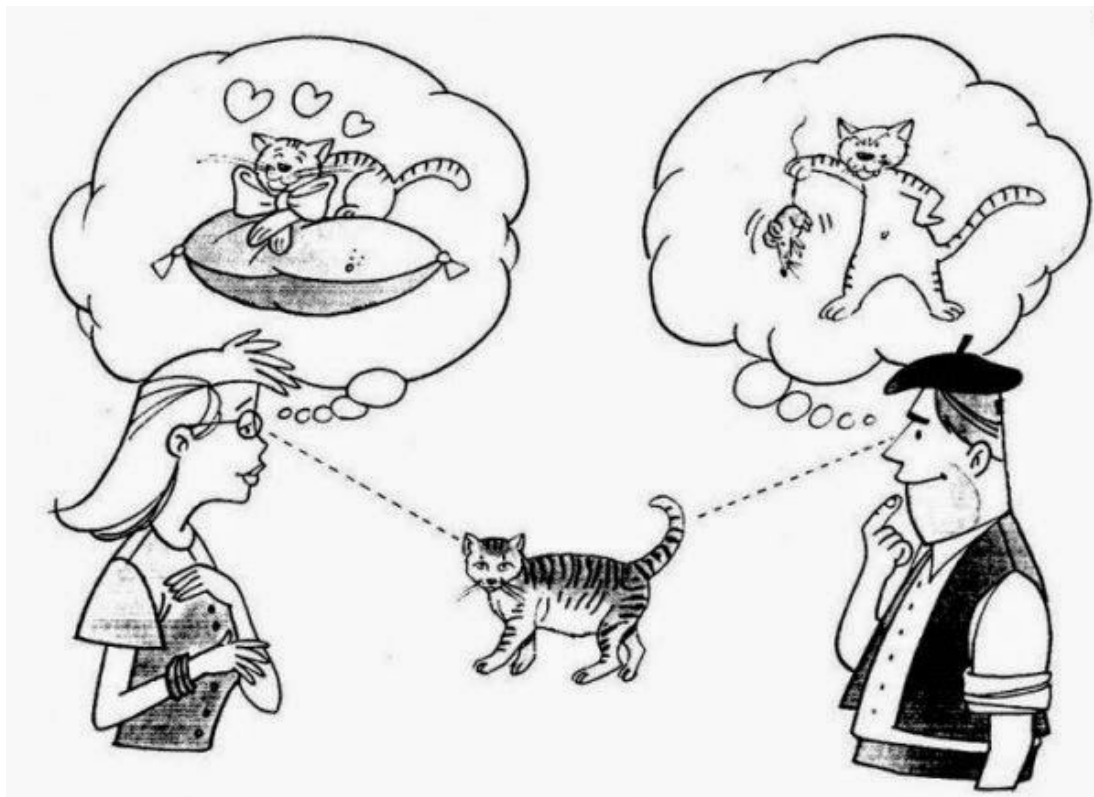


1 Objeto

La principal abstracción de nuestro nuevo paradigma son los objetos, que representan distintos tipos de entidades.

1.1 ¿Qué es un objeto?

Es algo que puedo representar a través de una idea, un concepto. Tiene entidad. Por ejemplo, un gato es algo del mundo real, que podemos conceptualizar fácilmente.



Representaciones mentales de un objeto real

Fuente:

<http://lacompuytros cuentos.blogspot.com.ar/2014/09/programacion-orientada-objetos-poo.html>

Aquí vemos que nuestro modelo mental es una simplificación del objeto real. Dos personas pueden ver la misma cosa, sin embargo están abstrayendo representaciones diferentes porque tienen objetivos distintos. Lo que importa para el paradigma no es el objeto real, sino nuestro modelo mental.



1.2 Objetos conceptuales

La deuda que mantengo con el almacenero, ¿es un objeto? Sí, también, porque es un concepto, no necesariamente tiene que ser un objeto *real*.

Un String, ¿es un objeto? “hola mundo” también es un objeto, al igual que un thread corriendo en la PC. Todo lo que existe en mi cabeza puede ser tomado como un objeto.

Esto puede resultar un tanto confuso... entonces, ¿dónde pongo el límite para definir un sistema con objetos? Definamos entonces qué es un sistema en el paradigma orientado a objetos:

“Un sistema es un conjunto de objetos
que se envían mensajes para alcanzar un determinado objetivo”

Una definición similar a la que propone la Teoría General de Sistemas: “Conjunto de partes que se relacionan para un objetivo común”.¹

2 Composición de un objeto

Un objeto es una entidad, un concepto, un ente. ¿Qué es lo que nos importa de un objeto? Qué mensajes le puedo enviar, lo que forma la **interfaz**.

Ejemplo: el gato de la imagen anterior es Félix, que sabe maullar, comer y dormir. Veamos su interfaz



¹ Entonces un sistema en este paradigma no son solo objetos, **son objetos que se envían mensajes**. Según el principio holístico: “el todo es más que la suma de las partes”, ya que el sistema no es solo sus partes sino también la interacción entre ellos



3 Mensaje

¿Cómo interactúan los objetos entre sí? A través de los mensajes. ¿Cómo envío un mensaje a otro objeto?

1. primero lo tengo que conocer (más adelante veremos cómo)
2. cuando lo conozco, sé qué mensajes le puedo mandar
3. ¿se cómo lo hace? no, no quiero saberlo, porque saberlo me ata a él (en términos de diseño decimos **"me acopla"**) y si yo asumo ciertas cosas en base a cómo lo hace, eso aumenta la posibilidad de tener que modificar cosas si el objeto al que le mando el mensaje cambia.

Otro ejemplo, nuestro caso testigo: pepita es un pájaro.

Mensajes → cosas que le puedo pedir a un objeto que haga

- energía (¿cuál es tu energía?)
- volá
- comé
- reset (volvé a tu estado inicial)

¿Qué hace volá y comé? El tema es dónde me ubico.

1. Si soy el que pide a pepita que vuele y que coma... No se exactamente lo que hace, pero se que quiero que vuele y que coma. No me interesa cómo está implementado porque yo estoy ocupado definiendo otras cosas. Ese beneficio se llama **abstracción** (concentrarnos sólo en lo que queremos resolver y dejar los detalles que no son esenciales de lado, eso nos permite mantener la complejidad lo suficientemente acotada).
2. Si soy pepita, yo sí sé cómo volar. Pero es algo mío, no tengo por qué andar contándole a todo el mundo que yo tengo energía y esas cosas. A mí me piden que vuele, vuelo. Me dicen que coma, como. Desde el lado de pepita, estoy encapsulando: agrupando funcionalidades propias. Pero no para protegerme, la implementación se puede ver en cualquier momento: yo sólo le estoy simplificando las cosas a mi cliente (donde cliente = el que me envía mensajes a mí).

4 Ambiente

Entonces tenemos: observador y objeto.



Pero ¿dónde viven esos objetos? Necesitaría un lugar (una especie de arenero donde pueda jugar con esos objetos). Entonces aparece la idea de un **ambiente**: el lugar donde viven los objetos.



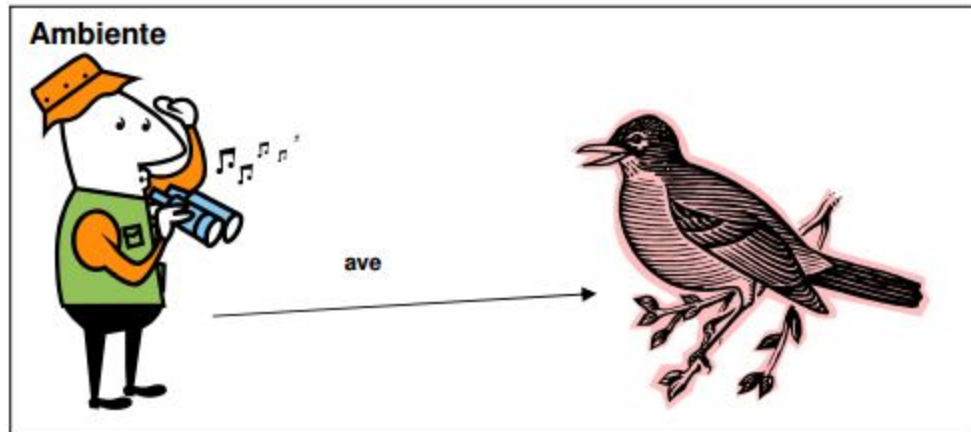
La palabra ambiente es conocida en otras tecnologías como

- **imagen** → Smalltalk
- **virtual machine** → Java
- **engine** → Javascript
- etc.

Ya está, los metí en el arenero. Pero ya sabemos que para que uno pueda pedirle cosas a otro, necesito que se conozcan...

5 Referencias

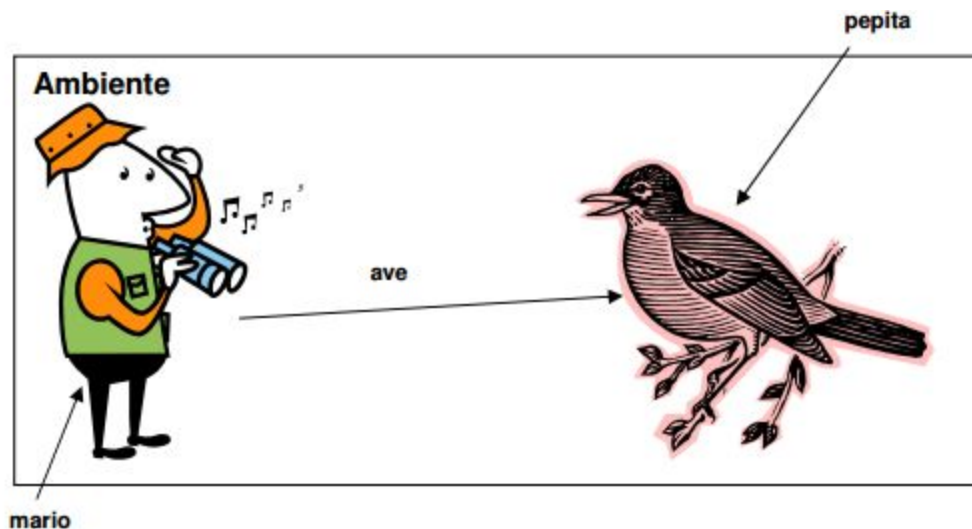
¿Cómo hago que se conozcan? A través de referencias, y aquí surge el término **variable**. Una variable apunta o referencia a un objeto. Una variable necesita un nombre, que es la forma que tiene el observador de dirigirse al objeto observado.



El observador le puede enviar un mensaje a pepita:

¡ave volá!

pero ¿no era que el ave se llamaba pepita? Lo que sucede es que puede haber varias referencias al mismo objeto. Para mí es pepita, para el observador es el ave, pero se trata del mismo objeto:



En el módulo siguiente profundizaremos esta idea.

6 Mensaje y método

¿Qué pasa cuando mario le envía el mensaje volá a pepita? Pepita sale disparada por



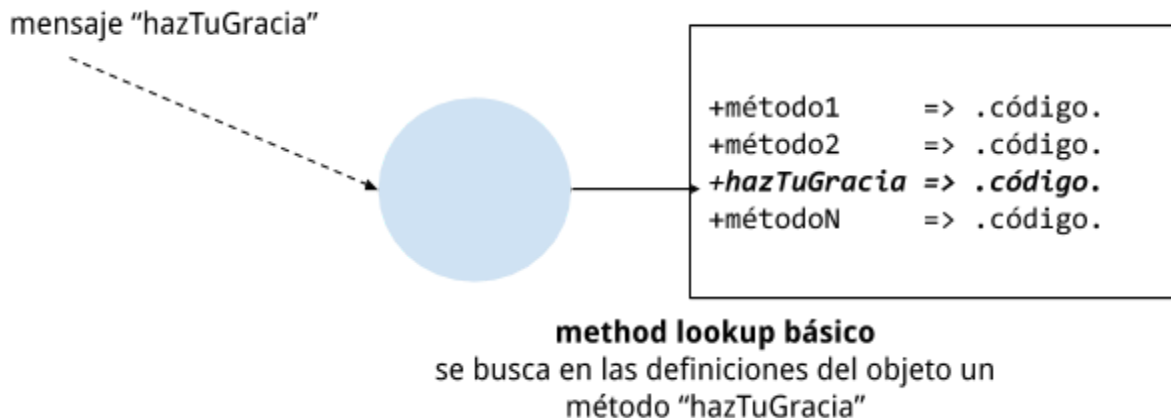
los aires. Bueno... y ¿en el software? Cuando le envío un mensaje a un objeto, se ejecuta código. ¿El código dónde se escribe? En el objeto receptor del mensaje. Este código recibe el nombre de método.

Entonces, pregunta de final: ¿qué diferencia hay entre mensaje y método?

Mensaje es lo que el objeto emisor le envía como orden al receptor. El emisor no sabe cómo se resuelve el mensaje, sólo pide que se resuelva (el qué y no el cómo, ¿a qué suena eso?). El receptor recibe el mensaje y se ejecuta un **método** (porción de código).

6.1 Method lookup

La estrategia que utilizan los lenguajes para resolver dónde está el código de un método al enviar un mensaje recibe el nombre de **method lookup**. En nuestro primer caso (el más simple) el código a ejecutar de un mensaje se busca en un método del objeto receptor de dicho mensaje.



6.2 Acciones y preguntas

Algunos mensajes responden a preguntas:

- pepita, ¿cuál es tu energía? ==> pepita.energia() ==> 10
- pepita, ¿sos feliz? ==> pepita.sosFeliz(ahora) ==> true
- pepita, ¿cuál es tu nombre? ==> pepita.nombre() ==> "pepita"

Mientras que otros mensajes simplemente disparan acciones que afectan al objeto en cuestión:



- pepita, ¡volá! ==> pepita.vola(10) ==> pepita se desplaza 10 kilómetros
- pepita, ¡comé! ==> pepita.come(2) ==> pepita come 2 gramos

En el primero de los casos nos importa el resultado, en el segundo nos importa más el **efecto colateral** que produce en el objeto.

7 Resumen

En este primer módulo conocimos los conceptos básicos del paradigma de objetos: la definición de objeto como representación de un concepto. Cada objeto define su interfaz, que son los mensajes que puedo enviar y que se traducen en métodos codificados.