

PROGRAMA OBJETOS

Módulo 1

- Discusión previa: Entes - Representación - Usuario e interacción
- Objeto
 - Definición
 - Comportamiento.
 - Ejemplos de objetos
- Mensaje
 - Elementos de un mensaje.
 - Mensajes que entiende un objeto.
 - Ejemplos de objetos con mensajes que entiende c/u.
- Método
 - *¿Dónde va el código? ¿Qué pasa cuando envío un mensaje?* = método
 - Relación entre mensaje y método.
 - Sintaxis básica de Smalltalk: objeto mensaje parámetros.
 - Tipos de mensaje en Smalltalk: unarios, binarios, de clave.
- Conocimiento - Estado interno - Variables
 - ¿Que es una variable? Es una referencia a un objeto.

OBJETO

Aquí presentamos un paradigma donde su principal abstracción son los objetos, que representan distintos tipos de entidades.

¿Qué es un objeto?

Es algo que puedo representar a través de una idea, un concepto. Tiene entidad.

¿Una silla? Seguro.

Una silla es algo del mundo real, que podemos conceptualizar fácilmente.

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?

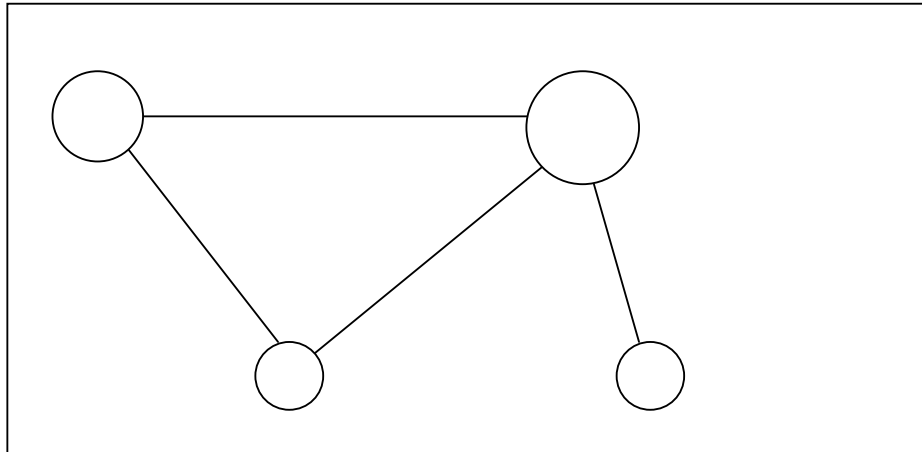
Definamos entonces qué es un sistema en el paradigma orientado a objetos:

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

¿Cuál era la definición de sistema según la Teoría General de Sistemas?

“Conjunto de partes que se relacionan para un objetivo común”

Bueno, al menos sabemos que vamos con una definición congruente...



¿Qué es lo ideal, que las partes se conozcan mucho, poco o nada?¹

Si se conocen mucho, vamos a ver que no va a tener onda, porque cada vez que una parte sufra una modificación, todas las otras partes que la conocen van a sufrir también ese impacto.

Si todas las partes quedan inconexas, dejamos de tener un sistema...

Entonces, lo ideal va a ser que las partes se conozcan sólo lo necesario...

Dejamos de decir partes, para decir: objetos.

Y un sistema no son sólo objetos, son objetos que se envían mensajes y entonces se cumple el principio holístico: el todo es más que la suma de las partes, porque sumando los objetos más sus interacciones tenemos al sistema final.

MENSAJE

¿Qué es todo eso de interacciones?

¿Mensajes?

¿Cómo envío un mensaje a otro objeto?

Bueno, vamos por partes.

Un objeto es una entidad, un concepto, un ente.

Ese ente tiene tres cosas:

- Identidad (cada objeto sabe que es él y que no es otro objeto, es lo que nos identifica a nosotros como seres)
- Atributos y valores para esos atributos
- Comportamiento

Ejemplo: un cajero de un banco (una persona que atiende como cajero),

tiene como atributos:

- Nombre: "Rúben"
- Fecha de nacimiento: 17/05/1948

Rúben es el valor que le corresponde al atributo Nombre,

17/05/1948 es el valor que le corresponde al atributo Fecha de nacimiento.

Tiene como comportamiento:

- sacame: 50 de: unaCuenta

¿Qué es 50, qué es unaCuenta?

¹ Esta es la pregunta central de toda la cursada y vamos a seguir estudiando el tema más adelante

Son dos objetos distintos:

- Uno representa el concepto matemático del número 50
- El otro representa una cuenta de un banco.

¡Eh! Una cuenta vaya y pase... pero ¿50 es un objeto?

Y... depende la implementación, pero en Smalltalk **todo** es un objeto. Y cuando decimos **todo, es todo**.

Otro mensaje que puede entender el cajero:

- depositá: 45 en: unaCuenta
- saldo (me devuelve la plata que tiene, equivaldría a un arqueo de caja)
- cerrá
- abrí
- pagá: unServicio con: 40

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

Qué atributos puede tener:

- energía
- kilometrosRecorridos (*Consejo:* cuando queremos nombrar algo, está bueno no abreviar las palabras, de manera que el nombre represente exactamente lo que tiene que representar. kmsRecorridos es difícil confundirlo con otra cosa, pero es común que al abreviar o sacar letras pts no se sabe si son pesetas, puntos, partes o qué extraña cosa)
- cantidad

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.

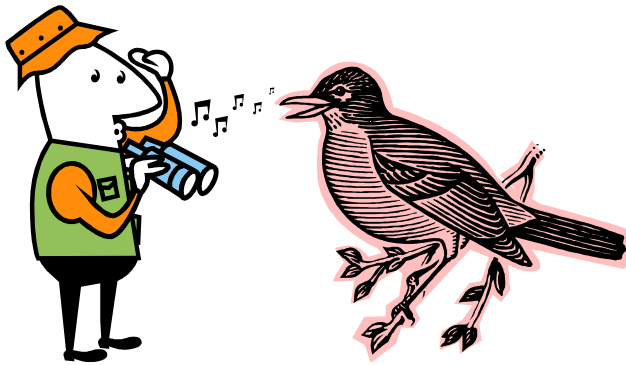
1) Si soy el que pide a pepita que vuele y que coma...

No se, supongo que al volar gastará energía, y al comer la recuperará pero eso no podemos saberlo, y tampoco nos interesa. Ese beneficio se llama: **abstracción** (concentrarnos sólo en lo que queremos resolver y dejar los detalles que no son esenciales de lado). Y esa abstracción favorece a que yo sepa menos de los objetos que manejo.

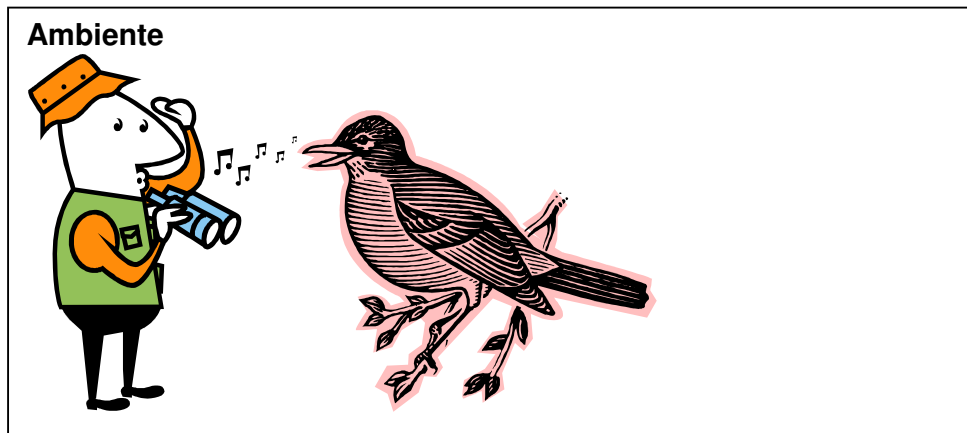
2) Si soy pepita, yo sí se cómo volar. Pero es algo mío, no tengo por qué andar contándole a todo el mundo que yo tengo energía y esas cosas. A mí me piden que vuele, vuelo. Me dicen que coma, como. Desde el lado de pepita, estoy **encapsulando**: agrupando funcionalidades que me son propias. Pero no para protegerme, no hay nada que me proteja en este mundo cruel: yo sólo le estoy facilitando las cosas a mi cliente.

REFERENCIAS. 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.

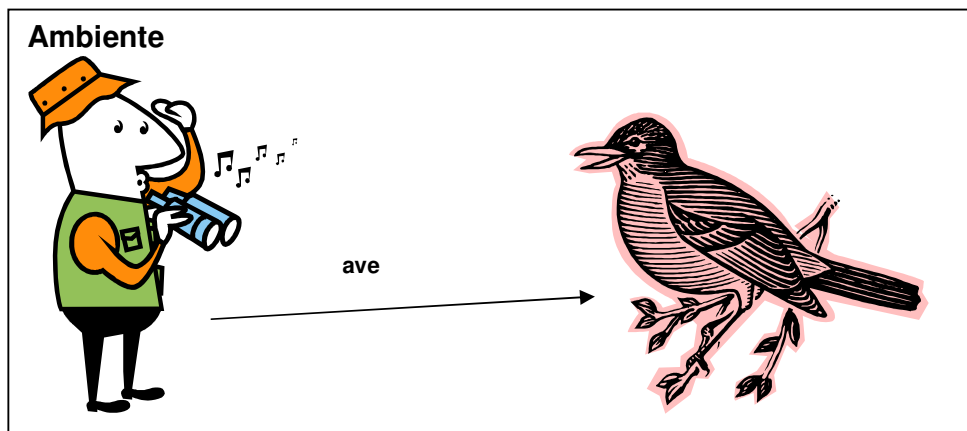


Ya está, los metí en el arenero. Pero para que uno pueda pedirle cosas a otro, necesito que se conozcan...

¿Cómo hago que se conozcan?

A través de referencias, y aquí surge el término variable. Una variable es un puntero a un objeto, o bien, **una variable referencia a un objeto**.

Una variable necesita un nombre, que es la forma que el observador tiene de dirigirse al objeto observado.

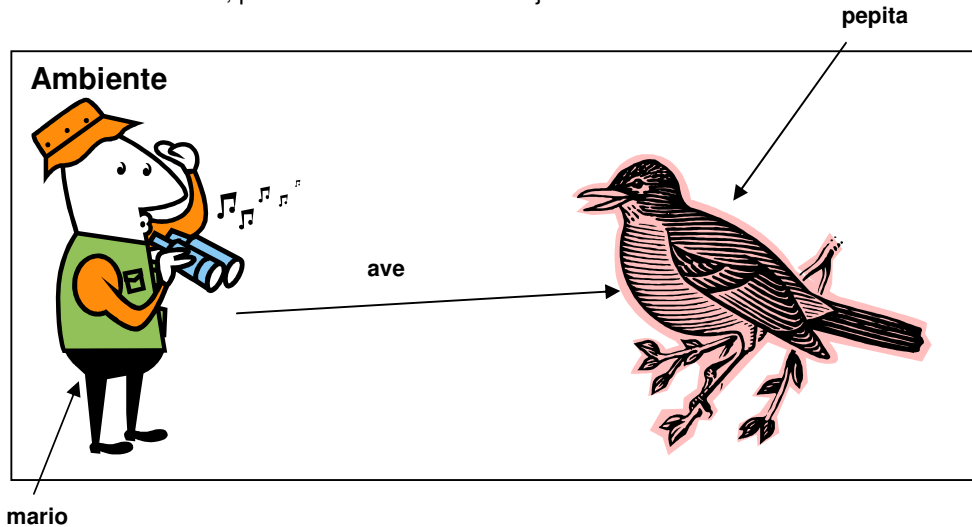


El observador le puede enviar un mensaje a pepita:

¡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:



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.

Ese 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 se entera de cómo se resuelve el mensaje, sólo lo pide (el qué y no el cómo, ¿a qué suena eso?)

El receptor recibe el mensaje y se ejecuta un método (porción de código).

Sintaxis para enviar un mensaje en Smalltalk
objeto mensaje parámetro/s.

La sintaxis es similar al que usamos para escribir en castellano: un mensaje se escribe como una oración, primero el sujeto, después el verbo, si necesitamos parámetros usamos los modificadores directos/indirectos y terminamos la oración con un punto.

La única diferencia es que una oración asertiva (está afirmando algo) se transforma en una orden hacia un objeto.

El perro muerde al cartero.

lo transformo a

perro mordéA: cartero.

Un perro y un cartero solos no pueden llevar a cabo la acción. Entonces ambos individuos (ahora los llamo objetos) **colaboran** para que esa acción se cumpla.

En Smalltalk tenemos tres tipos de mensaje:

MENSAJES UNARIOS

Tienen objeto receptor y un selector o nombre de método, pero no necesitan otro objeto como parámetro

2 negated → devuelve -2

true not → devuelve false

'hola mundo' reverse → devuelve 'odnum aloh'

2 es un objeto.

true es un objeto.

'hola mundo' es un objeto.

Los enteros, los booleanos y los strings son objetos. Todo objeto entiende ciertos mensajes.

¿Cuáles eran los mensajes unarios de Rúben, el cajero?

- dameTuSaldo
- abrí
- cerrá

Mensajes Binarios

Son operadores que tienen un objeto como parámetro

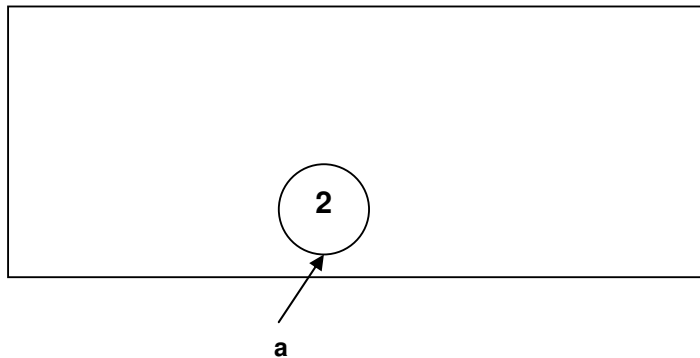
2 + 3 → estoy enviando el mensaje + 3 al objeto 2, que es entero.

Si yo hago

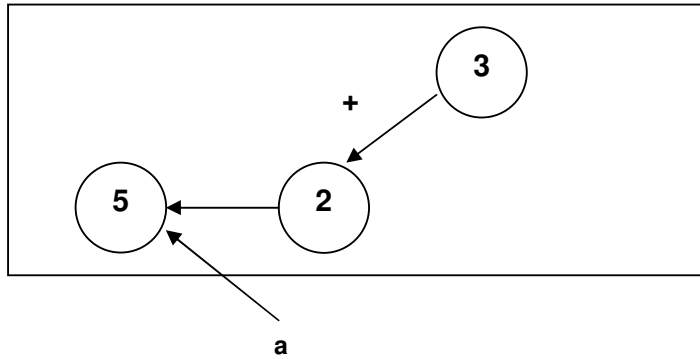
a ← 2.

a ← a + 3

Pasamos de:



al siguiente estado:



Entonces, ¿hay efecto colateral en el paradigma de objetos?

Sí, porque *a* apuntaba a un objeto 2 y pude asignarle otro valor (destruí el valor que tenía antes y lo cambié por otro).

Otros mensajes binarios:

4 @ 5 → defino un punto cuya abscisa es 4 y cuya ordenada es 5

6 * 8 → 48

'hola ' , 'mundo' → devuelve un nuevo string: 'hola mundo'

Mensajes de palabra clave

El nombre del método se parte anteponiendo un prefijo y dos puntos (:) a cada argumento a recibir.

'hola' at: 2 → devuelve el carácter \$o (toma el 2º elemento de un String)

ruben sacame: 50 de: unaCuenta → necesito dos argumentos
el nombre del método es sacame:de: (con los dos puntos)

¿Qué sentido tiene partir el nombre de un método antes de cada argumento? Que le da mucha más expresividad cuando lo uso.

Ojo que no vale pedirle:

ruben sacame: 50 unaCuenta

ni

ruben sacame: 50: unaCuenta

ni tampoco

ruben sacame: 50 De: unaCuenta (con mayúscula). Smalltalk es case sensitive, así que hay que sacarse los guantes...

unArray at: 2 put: 'pepe' → en la segunda posición del array pongo el String 'pepe' (ya veremos más cuando usemos Colecciones)