

Programación Orientada a Objetos

Ricardo Ruiz Rodríguez

Instituto de Computación
Primavera 2013

Índice I

- 1 Orientación a Objetos
 - Background
 - Paradigma
 - Una nueva forma de pensar
- 2 Conceptos
 - Fundamentos de OO
 - Java
- 3 Estructuras de Datos
 - Introducción
 - Información y significado
 - Tipos de datos abstractos ADT
 - Implementación
- 4 Pilas

Índice II

- Definición
- Operaciones primitivas
- Representación
- Aplicaciones

5 Colas

- Definición
- Operaciones primitivas
- Representación
- Colas de prioridad

6 Listas

- Definición
- Operaciones primitivas
- Aplicaciones

Índice III

- Listas circulares
- Listas doblemente enlazadas

7 Árboles

- Definición
- Operaciones primitivas
- Aplicaciones
- Recorridos
- Eliminación de un ABB
- Árboles binarios balanceados (AVL)

Background I

- Las características principales de lo que actualmente se denomina Programación Orientada a Objetos (POO) surgen en 1960.
- Por otro lado, los lenguajes Orientados a Objetos (OO) surgieron de manera más extendida en 1980.
- En Agosto de 1981 se publica en la revista *Byte* la descripción del lenguaje *Smalltalk*.
- La POO es una de las propuestas de solución para ayudar a resolver la “crisis del software”.
- Las técnicas OO *facilitan* la creación de complejos sistemas de software, pero la POO no es una panacea.

Background II

- Programar una computadora sigue siendo una de las tareas más difíciles jamás realizadas por un ser humano.
- Volverse experto en programación requiere, no sólo saber manejar herramientas, sino:
 - Talento.
 - Creatividad.
 - Inteligencia.
 - Lógica.
 - Habilidad para construir y utilizar abstracciones.
 - Experiencia.
- Hacer un uso efectivo de los principios OO requiere de una visión de mundo de una nueva manera.

Background III

- El uso de un lenguaje de POO no hace, por sí mismo, que se programe OO.
- Alan Kay es considerado el padre de la POO.

Paradigma I

Algunas definiciones:

- m. Ejemplo o ejemplar: *esa chica es el paradigma de la paciencia.*
- ling. Cada uno de los esquemas formales a los que se ajustan las palabras, según sus respectivas flexiones: *paradigma de la conjugación verbal.*
- ling. Conjunto de elementos de una misma clase gramatical que pueden aparecer en un mismo contexto: *paradigma de las preposiciones.*

Paradigma II

- El término paradigma significa *ejemplo* o *modelo*. En todo el ámbito científico, religioso u otro contexto epistemológico, el término paradigma puede indicar el concepto de esquema formal de organización, y ser utilizado como sinónimo de marco teórico o conjunto de teorías.

Entonces, ¿qué entender por paradigma en el contexto que nos compete?

- La palabra **paradigma** irrumpió en el vocabulario moderno a través del influyente libro *The Structure of Scientific Revolutions* del historiador de la ciencia Thomas Samuel Kuhn.

Paradigma III

- Thomas Khun utilizó el término en la forma de la última definición: modelo, para describir un conjunto de teorías, estándares y métodos que en conjunto representan una forma de organizar el conocimiento, esto es, *una forma de ver el mundo*.

Paradigma IV

Considero a los paradigmas como realizaciones científicas universalmente reconocidas que, durante cierto tiempo, proporcionan modelos de problemas y soluciones a una comunidad científica.

Thomas Kuhn.

Abstracción

Al pensar en objetos, deberá tener en mente los siguientes aspectos:

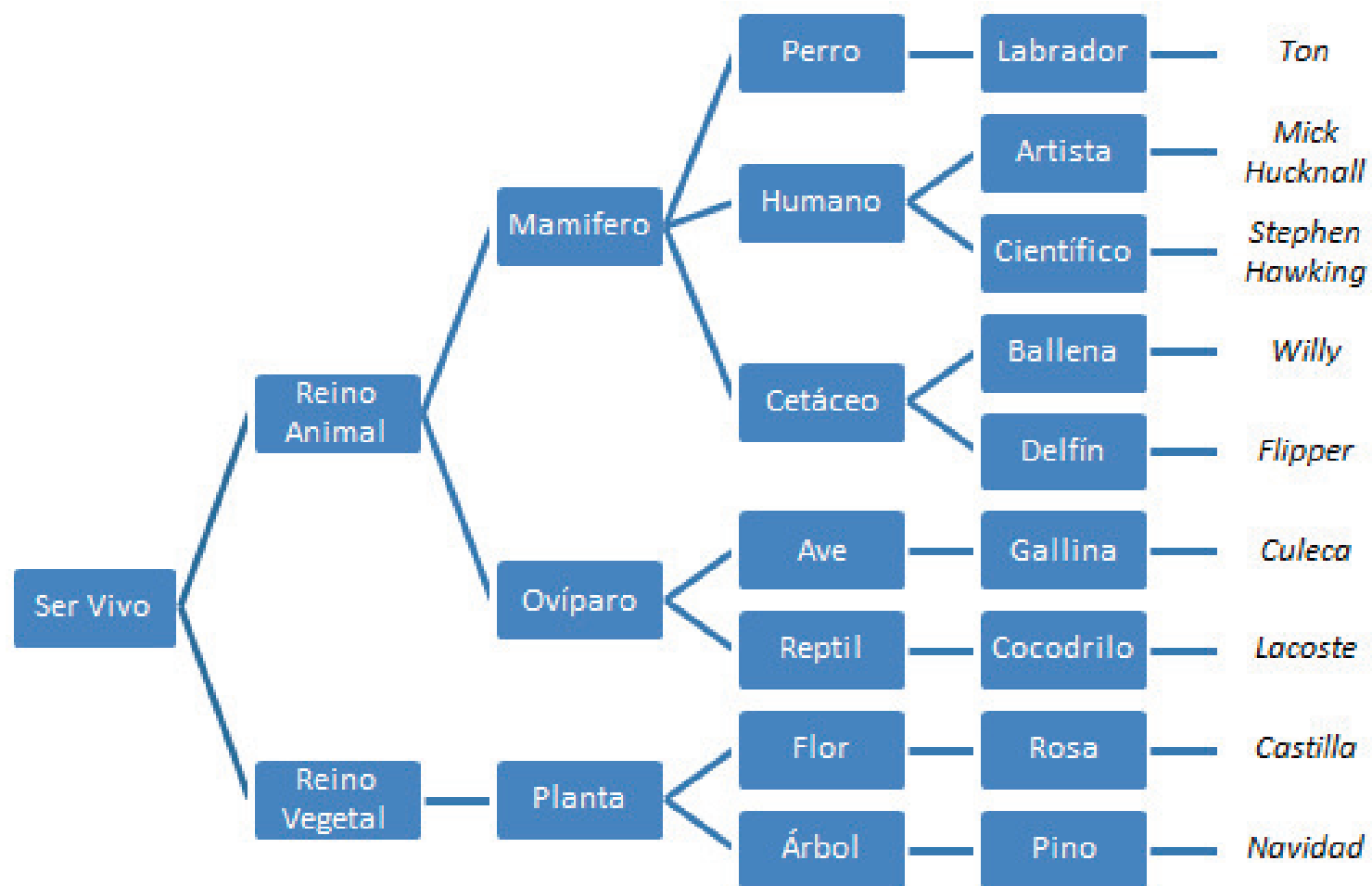
- 1 Entidades:** agentes u *objetos* en interacción, donde cada *objeto* tiene un rol.
- 2 Responsabilidades:** cada objeto proporciona un conjunto de servicios o lleva a cabo acciones que son utilizadas por otras *entidades*. Es el *comportamiento* del objeto.
- 3 Mensajes:** en la POO la *acción* es iniciada por la transmisión de un *mensaje* a un *objeto responsable* de dicha acción. En respuesta al *mensaje*, el objeto receptor llevará a cabo un *método* para satisfacer la solicitud.

Mensajes vs. procedimientos o funciones

Tanto los mensajes como los procedimientos o funciones son un conjunto de pasos bien definidos que llevan a cabo una acción. Pero se distinguen esencialmente por dos aspectos:

- 1 En un mensaje, hay un receptor designado para dicho mensaje.
- 2 La *interpretación* o método utilizado para responder al mensaje es determinado por el receptor, y puede variar en función del receptor.

Jerarquía de clases



Los objetos son abstracciones I

Ask not what you can do to your data structures, but rather ask what your data structures can do for you.

- Un concepto fundamental en la POO es la descripción del objeto en términos de *responsabilidades y características*.
- Al discutir un problema en términos de responsabilidades, se eleva el nivel de abstracción.
- Lo anterior permite una mayor *independencia* entre los objetos, lo cual es un factor crítico en la solución de problemas complejos.
- El conjunto completo de responsabilidades asociadas con un objeto es comúnmente referido como **protocolo**.

Los objetos son abstracciones II

- Un programa en un enfoque estructurado, frecuentemente opera *actuando sobre* estructuras de datos.
- Un programa en un enfoque OO *solicita* estructuras de datos (las cuales son otros objetos) para llevar a cabo un servicio.
- Todos los objetos son *instancias* de una *clase* (categoría).
- El método invocado por un objeto en respuesta a un mensaje, es determinado por la clase del receptor.
- Todos los objetos de una clase determinada, utilizan el mismo método en respuesta a mensajes similares.
- Las clases pueden ser organizadas en una estructura jerárquica de *herencia*.

Los objetos son abstracciones III

- Una clase hija o *subclase* heredará todas las características de la clase de la que deriva (clase padre).
- Una *clase abstracta* es una clase de la que no se derivan instancias directamente, sino que es utilizada únicamente para crear subclases.
- La búsqueda del método a invocar en respuesta a un mensaje dado, inicia en la clase del receptor. Si no se encuentra un método apropiado, la búsqueda se realiza en la clase padre, y así sucesivamente.
- Si se encuentran métodos con el mismo nombre dentro de la jerarquía de clases, se dice que el método ejecutado *sobre escribe* (*override*) el comportamiento heredado.

Alan Kay I

Características fundamentales de la POO:

- 1 Todo es un objeto.
- 2 El procesamiento es llevado a cabo por objetos:
 - Los objetos se comunican unos con otros solicitando que se lleven a cabo determinadas acciones.
 - Los objetos se comunican enviando y recibiendo mensajes.
 - Un mensaje es la solicitud de una acción que incluye los argumentos que sean necesarios para completar la tarea.
- 3 Cada objeto tiene su propia memoria, misma que está compuesta de otros objetos.
- 4 Cada objeto es una instancia de una clase. Una clase representa un grupo de objetos similares.

Alan Kay II

- 5 La clase es el repositorio del comportamiento asociado con un objeto.
 - Todos los objetos que son instancias de la misma clase, llevan a cabo las mismas acciones.
- 6 Las clases son organizadas en una estructura jerárquica de árbol denominada jerarquía de herencia.
 - La memoria y el comportamiento asociados con las instancias de una clase, están automáticamente disponibles para cualquier clase asociada con la descendencia dentro de la estructura jerárquica de árbol.

Abstracción

Es la consideración aislada de las cualidades esenciales de un objeto, o del mismo objeto en su pura esencia o noción.

Fundamentos de OO I

- La herramienta fundamental es la *abstracción*, la omisión deliberada de los detalles con la intención de enfatizar un conjunto pequeño de características.
- La ocultación de información describe la parte de la abstracción en donde intencionalmente se opta por ignorar algunas características para poder concentrarse en otras.
- **Ocultación de información:** es la omisión deliberada de datos en el desarrollo de una representación abstracta.
- La abstracción es frecuentemente combinada con una división en *componentes*.

Fundamentos de OO II

- Los componentes deben ser cuidadosamente seleccionados, de tal forma que *encapsulen* determinadas características clave, e interactúen con otros componentes a través de una *interfaz*.
- La división en componentes significa que se puede dividir una tarea en subproblemas que puedan ser trabajados de manera más o menos independiente.
- Las dos más importantes **formas de abstracción** utilizadas en la POO son:
 - 1 La idea de división en partes (**has-a abstraction**).
 - 2 La idea de división en especializaciones (**is-a abstraction**).
- Dos términos importantes en la POO:
 - 1 **Interfaz**: describe *qué* es lo que se debe hacer (vista externa).
 - 2 **Implementación**: especifica el *cómo* se hace (vista interna).

Fundamentos de OO III

- Un **patrón** es una descripción generalizada de una solución a un problema que se ha identificado que ocurre en muchas lugares en diversas formas.
- Un objeto es, a simple vista, la combinación de *estado* y *comportamiento*.
- La **encapsulación** es la característica o cualidad de ocultar los detalles de implementación de un objeto, ya sea en su estado o características (atributos) o en su comportamiento o responsabilidades (métodos).

Fundamentos de OO IV

- El **polimorfismo** se refiere a la posibilidad de enviar un mensaje a un grupo de objetos cuya naturaleza puede ser heterogénea. El único requisito que deben cumplir los objetos, es saber responder al mensaje que se les envía.

Java I

- Es un lenguaje de programación originalmente desarrollado por James Gosling de Sun Microsystems.
- Sun Microsystems es adquirida por Oracle Corporation en 2009.
- El lenguaje es publicado oficialmente en el 1995 como un componente fundamental de la plataforma Java de Sun Microsystems.
- El lenguaje deriva mucho de su sintaxis de C y C++, pero tiene menos facilidades de bajo nivel que cualquiera de ellos.

Java II

- Las aplicaciones de Java son generalmente compiladas a *bytecode* (clase Java) que puede correr en cualquier máquina virtual Java (JVM) sin importar la arquitectura de la computadora.
- Es un lenguaje de programación de propósito general, concurrente, basado en clases, y orientado a objetos, que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible.
- Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "*Write Once, Run Anywhere*").