

**Objetivos:**

- En esta práctica se verán conceptos de la POO, TADs vs Objetos
- Encapsulamiento. Envío de mensaje. Responsabilidades.
- UML y lenguaje de Modelado

**De Wikipedia** [http://es.wikipedia.org/wiki/Programación\\_orientada\\_a\\_objetos](http://es.wikipedia.org/wiki/Programación_orientada_a_objetos)

La programación Orientada a Objetos es un paradigma de programación que define los programas en términos de "clases de objetos", objetos que son entidades que combinan estado (es decir, datos), comportamiento (esto es, procedimientos o métodos) e identidad (propiedad del objeto que lo diferencia del resto). La programación orientada a objetos expresa un programa como un conjunto de estos objetos, que colaboran entre ellos para realizar tareas. Esto permite hacer los programas y módulos más fáciles de escribir, mantener y reutilizar.

Un objeto contiene toda la información, (los denominados atributos) que permite definirlo e identificarlo frente a otros objetos pertenecientes a otras clases. A su vez, dispone de mecanismos de interacción (los llamados métodos) que favorecen la comunicación entre objetos.

Una casa, una calculadora, una pieza del juego de ajedrez, son objetos de los que manipulamos día a día. Cada uno de estos objetos existe en el mundo real y tienen una identidad propia.

**Ejercicio 1.**

Supongamos que tenemos que representar una *locomotora eléctrica* como un objeto **locomotoraElectrica**. Esta locomotora eléctrica tiene ciertos atributos que lo caracterizan como así también tiene cierto comportamiento.

- Enumere los atributos que debería tener nuestro objeto **locomotoraElectrica**.
- ¿Cuáles serían los mensajes que le podríamos mandar a un **locomotoraElectrica** para que este responda con el método asociado al mismo? Realice una lista.
- Diga cuál sería el comportamiento deseado para cada uno de los ítems de la lista.

**Ejercicio 2.**

Identifique al menos 5 objetos que usted usa diariamente... como, por ejemplo: una cafetera, un auto de color rojo.

- De cada objeto indique sus atributos

- b) De cada objeto indique sus responsabilidades.

**Ejercicio 3.**

Se desea modelar en **Objetos** guitarras que estarán a la venta. Se debe tener en cuenta al menos los siguientes atributos: precio, fabricante, modelo (el modelo que le da el fabricante), tipo (p.e.: eléctrica, acústicas, etc.), la madera con la que está construida, etc.

- a) Describa todos los atributos de la guitarra
- b) Represente en UML varias guitarras

**Ejercicio 4.**

Queremos representar un horno a *micro ondas* como un objeto. Este objeto sería algo complejo, así que lo vamos a simplificar. El **microondas** solo tiene 4 botones, un botón para *abrir la puerta*, un botón para *cocinar*, un botón para *cocinar a media potencia* y un botón para *parar la cocción*. El tiempo de cocción es siempre fijo así que no nos preocuparemos por eso.

- a) Enumere los atributos que debería tener el horno a **microondas**.
- b) Suponiendo que cada botón es el disparador de un mensaje, enumere los mensajes que entiende el **microondas** y que acción realiza cada uno.
- c) ¿Qué pasaría si ahora el **microondas** que quiero modelar tiene otro botón que prende/apaga la señal sonora que indica que ya termino el proceso de cocción? ¿Qué debería modificar de dicho objeto?

**Ejercicio 5.**

Tenemos un objeto **cuentaBancaria**, la cual tiene un titular, un saldo, un máximo de extracciones y una fecha de creación.

- a) Enumere los atributos que debería tener nuestra **cuentaBancaria**.
- b) ¿Cuáles serían los mensajes que le podríamos mandar a una **cuentaBancaria** para que este responda con el método asociado al mismo? Realice una lista y diga cuales sería el comportamiento deseado para cada uno.

**Ejercicio 6.**

Investigue en internet o en bibliografía acorde los siguientes términos y características de la programación orientada a objetos:

- Clase
- Mensaje
- Método
- Atributo
- Abstracción
- Encapsulamiento
- Polimorfismo

**Ejercicio 7.**

Se desea modelar un **AireAcondicionado**. El mismo recibe “órdenes” de un supuesto control remoto. Algunas de las órdenes básicas que puede recibir son **encender, apagar, subirTemperatura, bajarTemperatura**

- Realice lista de atributos del mismo.
- Complete la lista de mensajes (órdenes) que puede recibir.
- Brinde una breve descripción sobre la acción que debe realizar el AC cuando recibe un mensaje.

**Ejercicio 8.**

Si bien los seres humanos no somos objetos, en la POO también se los representa como objetos. Teniendo esto en cuenta, supongamos que tenemos un objeto *persona* el cual tiene como atributo un nombre, edad, altura, peso, color de pelo. Todos estos atributos deberían ser accedidos por medio de mensajes y solo a través de ellos.

- ¿Cómo se llama esta característica en la POO?
- ¿Porque no sería adecuado acceder a los atributos de un objeto de forma directa?

**Ejercicio 9.**

En la actualidad hay muchos lenguajes *orientados* a objetos. Express. Smalltalk es un lenguaje creado en los años 70 por el Palo Alto Research Institute de XEROX.

- a) Busque información sobre Smalltak. Explique características del mismo y mencione los conceptos que introdujo en la informática.
- b) Además, busque información de otros lenguajes orientados a Objetos más moderno, enumere al menos 5 y compárelos entre sí.

**Ejercicio 10.**

Se desea modelar un reproductor de MP3. El mismo posee botones (que usted debe definir) para por ejemplo reproducir o pausar una canción. Indique todos los mensajes que se le pueden enviar al mismo y como reaccionaría el reproductor.

**Ejercicio 11.**

Investigue:

- a) ¿Qué es un TAD?
- b) ¿Cuál es la relación entre TADs y Objetos? ¿Qué características de los TADs tienen los Objetos?
- c) Defina el TAD pila con al menos la siguiente funcionalidad: vaciar, push(element), top, pop, isEmpty, size.
- d) Mencione y defina algún TAD de uso común que usted conozca.

**Ejercicio 12.**

A lo largo de esta materia vamos a utilizar y aprender un lenguaje de modelado para hacer los diseños de los ejercicios que veamos en cada una de las prácticas. Uno de los lenguajes más utilizados es el UML<sup>1</sup> (*Unified Modeling Language o Lenguaje Unificado de Modelado*).

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

- a) Busque información sobre cómo se representan en UML los siguientes componentes de un desarrollo orientado a objetos:

---

<sup>1</sup>

<http://www.uml.org/>

- Objetos
  - Clase
  - Mensaje
  - Método
  - Atributo
  - Secuencia
- b) Realice el **diagrama de clases** en UML del *Ejercicio 5* de la cuenta bancaria y *Ejercicio 7* del Aire acondicionado.