

Objetivos:

- Clases, instancias, mensajes y métodos
- Atributos y Modificadores de acceso
- Practicar el concepto polimorfismo del paradigma de objetos.
- UML

Ejercicio 1.

Se desea modelar un **Contador**. Un contador no es más que un objeto simple que lleva la cuenta de algo.

- Diseñe la clase Contador donde los mensajes que se le pueden enviar al Contador son:
 - iniciar()** -> pone en 0 al contador.
 - iniciar(n)** -> inicia al contador en un número N
 - sumar()** -> suma de a uno
 - restar()** -> resta de a uno
 - sumar(n)** -> suma N
 - restar(n)** -> resta N
- Implemente en JAVA-Like cómo deberían ser los métodos del contador.
- Preste atención a los mensajes que llevan argumento (n) y a los que no. ¿Que nos podría decir sobre el re-uso de los mismos?

Ejercicio 2.

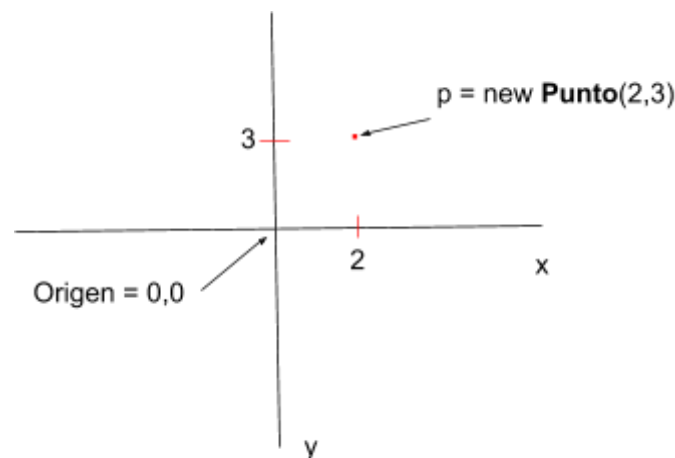
Modele la clase **AireAcondicionado** (la misma del ejercicio 7 del TP1) en UML.

- Enumere sus atributos y mensajes de un AireAcondicionado
- Implemente en JAVA-Like
- Realice el diagrama de secuencias del AireAcondicionado

Ejercicio 3.

Defina y diseñe la clase **Punto** (que posee un valor x e y) e implemente los siguientes métodos:

- i. `sumarValor(unValor)` -> 'Suma el mismo valor a x e y'
- ii. `sumarPunto(unPunto)` -> 'Suma las coordenadas x y las coordenadas y'
- iii. `distanciaDe(unPunto)` -> 'Calcula la distancia por Pitágoras'
- iv. `distanciaAlOrigen` -> 'Calcula la distancia por Pitágoras al origen'



Pitágoras:

$$d(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Ejercicio 4. (SUBIR a <https://plataformaed.unnoba.edu.ar>)

Defina la clase **Círculo**, que posee un centro (expresado como un **Punto**) y un radio, e implemente los siguientes métodos:

- i. `Circulo>>> ampliar(unValor)` -> 'Amplia el valor del radio según el parámetro'
- ii. `Circulo>>> área()` -> 'Retorna la superficie del círculo'

$$A = \pi \cdot r^2$$

iii. Circulo>> trasladar(x, y) -> “Traslada al círculo según los valores”

Ejercicio 5.

Se desea modelar un **sistema de información sobre películas y series**. Las mismas cuentan con actores y directores. Además de personajes y en el caso de las series se tienen además los capítulos.

- a) Realice el diagrama de clases en UML
- b) Especifique todos los atributos de cada clase.

Ejercicio 6. (SUBIR a <https://plataformaed.unnoba.edu.ar>)

Defina la clase **Persona** la clase **Cita** y **Lugar**. Una cita está compuesta por una fecha (vea en la **Guía de JAVA-Like y BlueJ** cómo utilizar fechas), un contacto (Persona), un lugar (Lugar) y una importancia (un número de 1 a 5 que marca la prioridad, siendo 5 la más alta).

- a) Realice el diagrama de clases en UML
- b) Implemente en JAVA-Like las clases necesarias.
- c) Cree objetos **Cita** e inicialícelos con los datos necesarios.

Ejercicio 7.

Retome el ejercicio 3 del trabajo práctico 1. Diseñe la clase **Guitarra**, pero intente abstraer características de las guitarras que son comunes a todas. Ya sabemos que el precio, el número de serie y la fecha de fabricación son parte de la guitarra. Pero el resto de los atributos se podrían modelar como una clase diferente.

- a) ¿Cómo hace esto?
- b) Modele en UML la solución.

Ejercicio 8. (SUBIR a <https://plataformaed.unnoba.edu.ar>)

Enumere las partes de un auto. Realice la abstracción del mismo en Clases y relacione las mismas.

- a) Modele en UML la solución.

Ejercicio 9.

En <http://www.lesser-software.com/en/flavours/frames/default.htm> hay algunas implementaciones libres de Smalltalk y otras comerciales.

En base a lo que encuentre en la bibliografía y en Internet responda:

- a) ¿Que es Smalltalk?
- b) ¿Que es VisualWorks?
- c) ¿Por qué se dice que el Smalltalk/V es un ambiente?
- d) ¿Por qué se dice que Smalltalk está programado en Smalltalk? Brinde algunos ejemplos.
- e) ¿En Smalltalk es necesario compilar el código? ¿Por qué?