

# Introducción a la Programación en Python

## Módulo Nro. 6

Programación estructurada vs POO. Clases, objetos y atributos.

### Trabajo Práctico Nro. 6: Programación Orientada a Objetos

**Ejercicio 1:** Crear una clase llamada Persona con los atributos: nombre, edad y DNI. Diseñar los siguientes métodos para la clase:

- Un constructor, donde los datos pueden estar vacíos;
- Mostrar los datos de las personas;
- Determinar si es mayor de edad.

**Ejercicio 2:** Crear una clase llamada Cuenta que tendrá los atributos: titular (que es una persona) y cantidad (puede tener decimales). El titular será obligatorio y la cantidad es opcional. Construye los siguientes métodos para la clase:

- Un constructor;
- El atributo cuenta no se puede modificar directamente, sólo ingresando o retirando dinero;
- mostrar(): Muestra los datos de la cuenta;
- ingresar(cantidad): se ingresa una cantidad a la cuenta, si la cantidad introducida es negativa, no se hará nada;
- retirar(cantidad): se retira una cantidad a la cuenta. La cuenta puede estar en números rojos.

**Ejercicio 3:** Diseñar una clase Cuentas Bancarias que tiene como atributo una Lista de cuentas. Añadir las siguientes funcionalidades:

- Crear constructor;
- Saldo\_deudor: Muestra el saldo deudor total de todas las cuentas con saldo negativo;
- Mostrar todas las cuentas con saldo negativo con el valor adeudado.



# Introducción a la Programación en Python

**Ejercicio 4:** Escribir una clase en python que obtenga todas las posibles sublistas únicas de una lista de números enteros distintos. No tener en cuenta una lista vacía.

**Entrada:** [4, 5, 6]

**Salida:** [[6], [5], [5, 6], [4], [4, 6], [4, 5], [4, 5, 6]]

**Ejercicio 5:** Escribir una clase en python que encuentre un par de elementos de una matriz dada cuya suma es igual a un número determinado. Se debe indicar la posición de los números que cumplen la condición.

**Entrada:** números = [10,20,10,40,50,60,70], objetivo=50

**Salida:** 3, 4

**Ejercicio 6:** Crear una clase llamada Rectangulo con dos puntos (inicial y final) que formarán la diagonal del rectángulo (los puntos se formarán con una lista de 2 elementos).

- Añade un método constructor para crear ambos puntos fácilmente, si no se envían se crearán dos puntos en el origen por defecto.
- Añade al rectángulo un método llamado base que muestre la base.
- Añade al rectángulo un método llamado altura que muestre la altura.
- Añade al rectángulo un método llamado área que muestre el área.

**Ejercicio 7:** Crear una clase llamada Punto con 2 coordenadas (atributos) X e Y:

- Añade un método constructor para crear puntos fácilmente. Si no se recibe una coordenada, su valor será cero.
- Sobreescribe el método string, para que al imprimir por pantalla un punto aparezca en formato (X,Y)
- Añade un método llamado cuadrante que indique a qué cuadrante pertenece el punto, teniendo en cuenta que si  $X == 0$  e  $Y != 0$  se sitúa sobre el eje Y, si  $X != 0$  e  $Y == 0$  se sitúa sobre el eje X y si  $X == 0$  e  $Y == 0$  está sobre el origen.
- Añade un método llamado vector, que tome otro punto y calcule el vector (se representaría como la diferencia entre las coordenadas del segundo punto respecto al primero) resultante entre los dos puntos.



## Introducción a la Programación en Python

**Ejercicio 8:** Crear una clase Fecha con atributos para el día, el mes y el año de la fecha. Incluye, al menos, los siguientes métodos:

- Constructor predeterminado con el 1-1-1900 como fecha por defecto.
- año\_bisiesto(): indicará si el año de la fecha es bisiesto o no.
- dias\_mes(int): devolverá el número de días del mes que se le indique (para el año de la fecha).
- validar(): comprobará si la fecha es correcta (entre el 1-1-1900 y el 31-12-2050); si el día no es correcto, lo pondrá a 1; si el mes no es correcto, lo pondrá a 1; y si el año no es correcto, lo pondrá a 1900. Será un método auxiliar (privado).
- Definir métodos getter y setter.
- hoy(): devuelve la fecha actual.
- corta(): mostrará la fecha en formato corto (02-09-2003).
- larga(): mostrará la fecha en formato largo, empezando por el día de la semana (martes 2 de septiembre de 2003). los días que se indiquen desde el 1-1-1900.

