



Entender las diferencias entre la programación procedural y la programación orientada a objetos (POO).  
Empezar a entender la terminología utilizada para este paradigma de programación y hacer los primeros experimentos.

## Aprender a pensar como un programador: Introducción a la POO

Al comenzar a programar por primera vez habitualmente se sigue un estilo procedimental al escribir código. Es decir, los programas consisten en una serie de instrucciones, una detrás de otra, que se ejecutarán paso a paso. Para que sea funcional y poder reutilizarla se definen las funciones. El problema es que nos terminan quedando programas muy extensos con muchas variables globales y pocas locales. Suele ser muy complicado aislar los datos específicos de cada función (esto es un problema de seguridad) además de estar todo desorganizado.

Sin embargo, los lenguajes modernos como C#, Java o... en realidad casi cualquiera, utilizan otros paradigmas para definir los programas. Entre éstos, el paradigma más popular es el que se refiere a la Programación Orientada a Objetos o POO.

**¿Por qué la Programación Orientada a Objetos (POO)?** Detectado el problema necesitamos una solución:

**1.- El problema:** En la programación procedimental nos terminan quedando programas muy extensos además de estar todo desorganizado. Nos queda código espagueti.

**2.- La solución:** La Programación Orientada a Objetos (POO). Este tipo de programación se utiliza para estructurar un programa de software en piezas simples y reutilizables. Nos permite que el código sea reutilizable, esté organizado y sea fácil de mantener. Sigue el principio de desarrollo de software utilizado por muchos programadores *DRY (Don't Repeat Yourself)*, para evitar duplicar el código y crear de esta manera programas eficientes.

Con el paradigma de la Programación Orientado a Objetos lo que buscamos es dejar de centrarnos en la lógica pura de los programas para empezar a pensar en objetos, la base de este paradigma. Esto nos ayuda muchísimo en grandes proyectos, donde intervienen muchas personas, ya que en vez de pensar en funciones, pensamos en las relaciones o interacciones de los diferentes componentes del sistema. Es un puzzle donde todas las fichas son necesarias.

Los programadores diseñan un software organizando piezas de información y comportamientos relacionados en una plantilla. Esta plantilla es la **CLASE**. Luego, se crean **OBJETOS** individuales a partir de la plantilla de clase. Todo el software se ejecuta haciendo que varios objetos interactúen entre sí para crear un programa que resuelva problemas más complejos.

En el caso de Python, lenguaje interpretado que se utiliza mucho para construir soluciones de IA, es un lenguaje orientado a objetos. Los objetos contienen una serie de **MÉTODOS** (en la programación procedimental los llamamos funciones) y **PROPIEDADES** o **ATRIBUTOS** (en la programación procedimental los llamamos parámetros) que lo definen. Por ejemplo:

1.- Si pensamos en un humano, tenemos como propiedades/atributos el color de ojos, color de piel, altura... y los métodos serían las acciones como el baile, leer, saltar, correr, ...

2.- Si pensamos en una pelota, tienen propiedades/atributos como el color, la textura, el peso, ... y los métodos serían las acciones como botar, pinchar, rebotar, ...

3.- Si pensamos en autobuses, tienen propiedades/atributos como el color, la matrícula, la capacidad, ... y los métodos serían las acciones como arrancar, detenerse, recoger pasajeros, ...

Esta metodología de programación nos permite reutilizar el código (crear autobuses o los objetos que sean de diferentes colores, matrículas y capacidades) las veces que queramos sin tener que estar reescribiendo código una y otra vez. Además tendremos todo el proyecto bien organizado y estructurado.

**Ejercicio:** Se recomienda leer con detenimiento estos dos enlaces ([uno](#), [dos](#)).

**A.-** Intenta comprender que significan los siguientes conceptos:

- . Clase
- . Objeto
- . Método
- . Propiedad/Atributo
- . Instancia
- . Constructor
- . Instancia
- . Encapsulamiento
- . Herencia

**B.-** Intenta profundizar y comprender qué significa esta *nueva* metodología para la programación y recapacita en cómo afecta en tu manera de pensar a la hora de programar. ¿Supone mucho cambio? No te desanimes, es más sencillo de lo que parece.

**C.-** Posteriormente debes realizar estos ejercicios:

1.- Crear una clase llamada *Persona*. Sus atributos son: *nombre*, *edad* y *DNI*. Construye los siguientes métodos para la clase:

- . Un *constructor*, donde los datos pueden estar vacíos.
- . Los *setters* y *getters* (métodos *set* y *get*) para cada uno de los atributos. Hay que validar las entradas de datos.
- . *mostrar()*: muestra los datos de la persona.
- . *esMayorDeEdad()*: devuelve un valor lógico indicando si es mayor de edad.

2.- Crea una clase llamada *Cuenta* que tendrá los siguientes atributos:

- . *titular* (que es una persona)
- . *cantidad* (puede tener decimales).

El titular será obligatorio y la cantidad es opcional. Construye los siguientes métodos para la clase:

- . Un *constructor*, donde los datos pueden estar vacíos.
- . Los *setters* y *getters* para cada uno de los atributos. El atributo 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.

Enlace de los ejercicios [AQUÍ](#)

Resolución de los ejercicios [AQUÍ](#)

**#HASHTAGS** (*etiquetas de ayuda para búsqueda de información relevante*)

#programación-orientada-a-objetos #POO #Python #clase #objeto #método #atributo

## LINKS DE INTERÉS

<https://echemosunbitstazo.es/blog/curso-de-python-introducci%C3%B3n-programacion-orientada-a-objetos>

<https://docplayer.es/12657913-Introduccion-a-la-programacion-orientada-a-objetos-con-python.html>

<https://www.youtube.com/watch?v=Nka4JSBgf7I>

<https://profile.es/blog/que-es-la-programacion-orientada-a-objetos/>

## DICCIONARIO

Clase | Objeto | Método | Atributo | Instancia | Constructor | Instancia | Encapsulamiento | Herencia

## PUNTUACIÓN

Programación: 6

Redes: 1

Seguridad: 4

Algoritmia: 1