

Curso libre

POO

Juan Felipe Acevedo Pérez

Monitor Junior

Correo: uniic_bog@unal.edu.co

Tel: 3165000 **Ext:** 12301

Contenido



POO



Atributos



Métodos

1

Programación Orientada a objetos

Correo: uniic_bog@unal.edu.co

Teléfono: 3165000 ext 12301

¿Qué es?

Se trata de un paradigma de programación introducido en los años 1970s, pero que no se hizo popular hasta años más tarde.

Este modo o paradigma de programación nos permite organizar el código de una manera que se asemeja bastante a como pensamos en la vida real, utilizando las famosas clases. Estas nos permiten agrupar un conjunto de variables y funciones que veremos a continuación.



Hablemos de cotidianidades

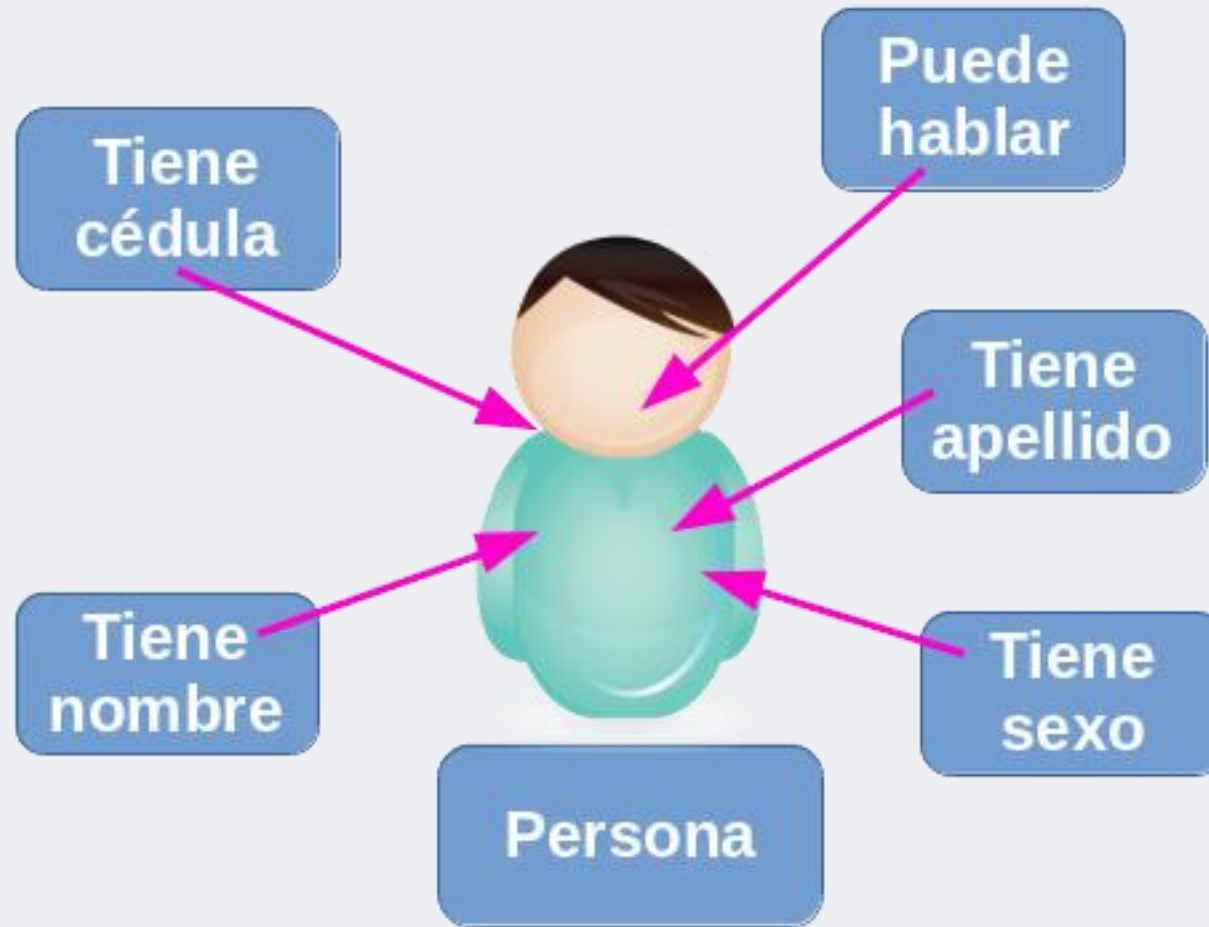
- Cosas de lo más cotidianas como un perro o un coche pueden ser representadas con clases. Estas clases tienen diferentes características, que en el caso del perro podrían ser la edad, el nombre o la raza. Llamaremos a estas características, **atributos**.
- Por otro lado, las clases tienen un conjunto de funcionalidades o cosas que pueden hacer. En el caso del perro podría ser andar o ladrar. Llamaremos a estas funcionalidades **métodos**.
- Por último, pueden existir diferentes tipos de perro. Podemos tener uno que se llama Toby o el del vecino que se llama Laika. Llamaremos a estos diferentes tipos de perro **objetos**. Es decir, el concepto abstracto de perro es la **clase**, pero Toby o cualquier otro perro particular será el **objeto**.





	Mundo Real	En OOP
Clase Generalización de características (atributos y comportamientos)	Perro Raza, Color, Edad, Corre,	Clase Define datos y métodos
Objeto Instancia de una clase distinguible por sus características específicas	Tino Pastor Alemán Marrón 7 meses Veloz	Objetos Ocupa espacio, se crea y se destruye

Ejercicio en clase



Principios

- Herencia
- Cohesión
- Abstracción
- Polimorfismo
- Acoplamiento
- Encapsulamiento





2

Atributos

Correo: uniic_bog@unal.edu.co

Teléfono: 3165000 ext 12301

Atributos

- Distinguir la existencia de dos tipos de atributos:
- **Atributos de instancia:** Pertenecen a la instancia de la clase o al objeto. Son atributos particulares de cada instancia, en nuestro caso de cada perro.
- **Atributos de clase:** Se trata de atributos que pertenecen a la clase, por lo tanto serán comunes para todos los objetos.



3

Métodos

Correo: uniic_bog@unal.edu.co

Teléfono: 3165000 ext 12301

Métodos

- Los métodos describen el comportamiento de los objetos de una clase. Estos representan las operaciones que se pueden realizar con los objetos de la clase,
- La ejecución de un método puede conducir a cambiar el estado del objeto.
- Se definen de la misma forma que las funciones normales pero deben declararse dentro de la clase y su primer argumento siempre referencia a la instancia que la llama, de esta forma se afirma que los métodos son funciones, adjuntadas a objetos.

Métodos especiales

- Las clases en Python cuentan con múltiples métodos especiales, los cuales se encuentran entre dobles guiones bajos `__<metodo>__()`.
- Los métodos especiales más utilizados son `__init__()`, `__str__()` y `__del__()`.

```
>>> class Animal :
...     def __init__(self, animal, breed) :
...         self.animal = animal
...         self.breed = breed
...
...     def __str__(self) :
...         return "You just called __str__"
...
...     def __repr__(self) :
...         return "You just called __repr__"
...
...
... a = Animal("Dog", "Pomeranian")
... print(a)
You just called __str__
>>> a
You just called __repr__
```


Métodos especiales

- Las clases en Python cuentan con múltiples métodos especiales, los cuales se encuentran entre dobles guiones bajos `__<metodo>__()`.
- Los métodos especiales más utilizados son `__init__()`, `__str__()` y `__del__()`.

```
>>> class Animal :
...     def __init__(self, animal, breed) :
...         self.animal = animal
...         self.breed = breed
...
...     def __str__(self) :
...         return "You just called __str__"
...
...     def __repr__(self) :
...         return "You just called __repr__"
...
...
... a = Animal("Dog", "Pomeranian")
... print(a)
You just called __str__
>>> a
You just called __repr__
```

__str__()

- El método `__str__()` es un método especial, el cual se ejecuta al momento en el cual un objeto se manda a mostrar, es decir es una cadena representativa de la clase, la cual puede incluir formatos personalizados de presentación del mismo.

```
def __str__(self):  
    """Devuelve una cadena representativa de Persona"""  
    return "%s: %s, %s %s, %s." % (  
        self.__doc__[25:34], str(self.cedula), self.nombre,  
        self.apellido, self.getGenero(self.sexo))
```

__del__()

- El método `__del__()` es un método especial, el cual se ejecuta al momento en el cual un objeto es descartado por el intérprete. El comportamiento de `__del__()` es muy similar a los «destructores» en otros lenguajes.

```
#Python program to illustrate Destructor
class demo:
    def __init__(self):
        print("Welcome")
    def __del__(self):
        print("Destructor executed successfully")

obj1=demo()
print(id(obj1))
del obj1
```