



Programador

Programación Orientada a Objetos (POO)

Qué es la Programación Orientada a Objetos (POO)



Es un paradigma de programación que organiza su diseño basado en objetos (clases).

Cada clase o entidad tiene un estado y un comportamiento:

- Su estado es expresado por el valor de sus atributos
- Su comportamiento lo definen sus métodos o responsabilidades

En la POO los objetos se comunican por mensajes

Beneficios de la POO



- Reutilización
- Escalabilidad
- Mantenimiento
- Modularidad
- Reduce la complejidad
- Evita el DRY (Don't Repeat Yourself)
- Protege la información
- Trabajo colaborativo

Principios de la POO



- Encapsulación
- Abstracción
- Herencia
- Polimorfismo

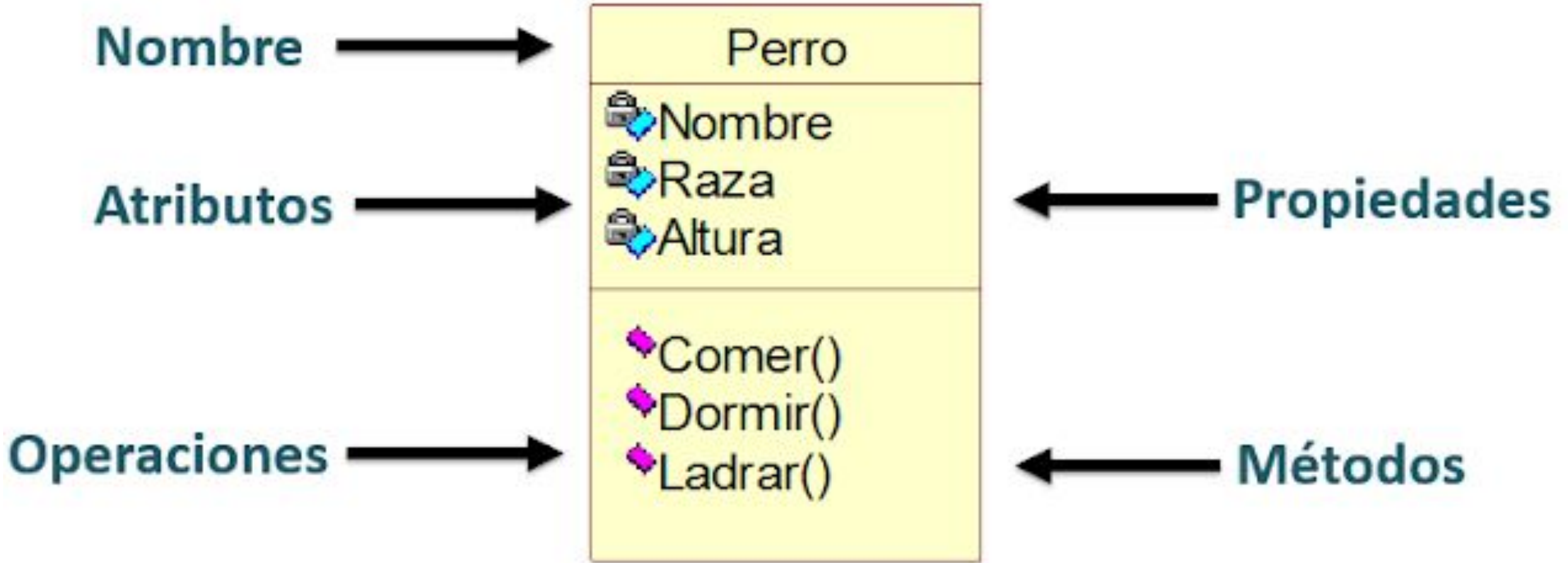
Lenguajes



- Java
- Python
- C# / C++
- PHP
- Ruby
- .NET
- JavaScript

[Lenguaje orientado a objetos - Wikipedia, la enciclopedia libre](#)

Clases | Class



Clases | Class - Definición



```
class Perro:
```

```
    pass
```

Clases - Métodos (funciones)



```
class Perro:
    def __init__(self, nombre, raza, altura):
        self.nombre = nombre
        self.raza = raza
        self.altura = altura

    def ladrar(self):
        return self.nombre + " está ladrando: ¡Guau guau!"
```


El métodos `__init__()`



```
class Perro:
    def __init__(self, nombre, raza, altura):
        self.nombre = nombre
        self.raza = raza
        self.altura = altura

    def ladrar(self):
        return self.nombre + " está ladrando: ¡Guau guau!"
```

`__init__` se ejecuta siempre que se crea una nueva instancia de la clase

Atributos de una clase



Son las características o cualidades que son importante mantener y representar. Ejemplo el nombre y apellido de una persona, su dni, género, etc.

Atributos visibilidad o modificadores



Los modificadores de atributos se usan para definir la visibilidad de los atributos de una clase:

- **Atributos Públicos:** Accesibles desde cualquier lugar.
- **Atributos Protegidos:** El atributo no debe ser accedido fuera de la clase o por sus subclases.
- **Atributos Privados:** No son accesibles desde fuera de la clase (solo con métodos).

Atributos de una clase



```
class Perro:
    def __init__(self, nombre, raza, altura):
        self.nombre = nombre    # Atributo público
        self.raza = raza        # Atributo público
        self.altura = altura    # Atributo público

    def ladrar(self):
        return self.nombre + " está ladrando: ¡Guau guau!"
```

Clases vs Objetos

Perro	
	Nombre
	Raza
	Altura
	Comer()
	Dormir()
	Ladrar()

Clase



Objetos

Crear Objeto o Instancia

```
class Perro:
    def __init__(self, nombre, raza, altura):
        self.nombre = nombre    # Atributo público
        self.raza = raza        # Atributo público
        self.altura = altura    # Atributo público

    def ladrar(self):
        return self.nombre + " está ladrando: ¡Guau guau!"

#como lo invoco
perrito = Perro("Boby", "DelBA", 45)
perrito.ladrar()
print(perrito.nombre)
print(perrito.raza)
```