

# INTRODUCCIÓN A POO



SOMOS  
EVOLUCIÓN



# ¿QUÉ ES POO?

La Programación Orientada a Objetos (POO) es un modelo o un estilo de programación que nos da unas guías sobre cómo trabajar con él. Se basa en el concepto de clases y objetos.

En otras palabras, se utiliza para estructurar un programa en piezas simples y reutilizables de planos de código (clases) para crear instancias individuales de objetos.

## CONCEPTOS

- POO lo que buscamos es dejar de centrarnos en la lógica pura de los programas, para empezar a pensar en objetos.
- La POO se organiza organizando piezas de información y comportamientos relacionados en una plantilla llamada clase.



## ¿POR QUÉ USAR POO?

- Sigue el principio de desarrollo de software.
- POO permite que el código sea reutilizable, organizado y fácil de mantener.
- Evita la duplicidad de código.
- Evita el acceso no deseado a los datos o la exposición de código propietario mediante la encapsulación y la abstracción



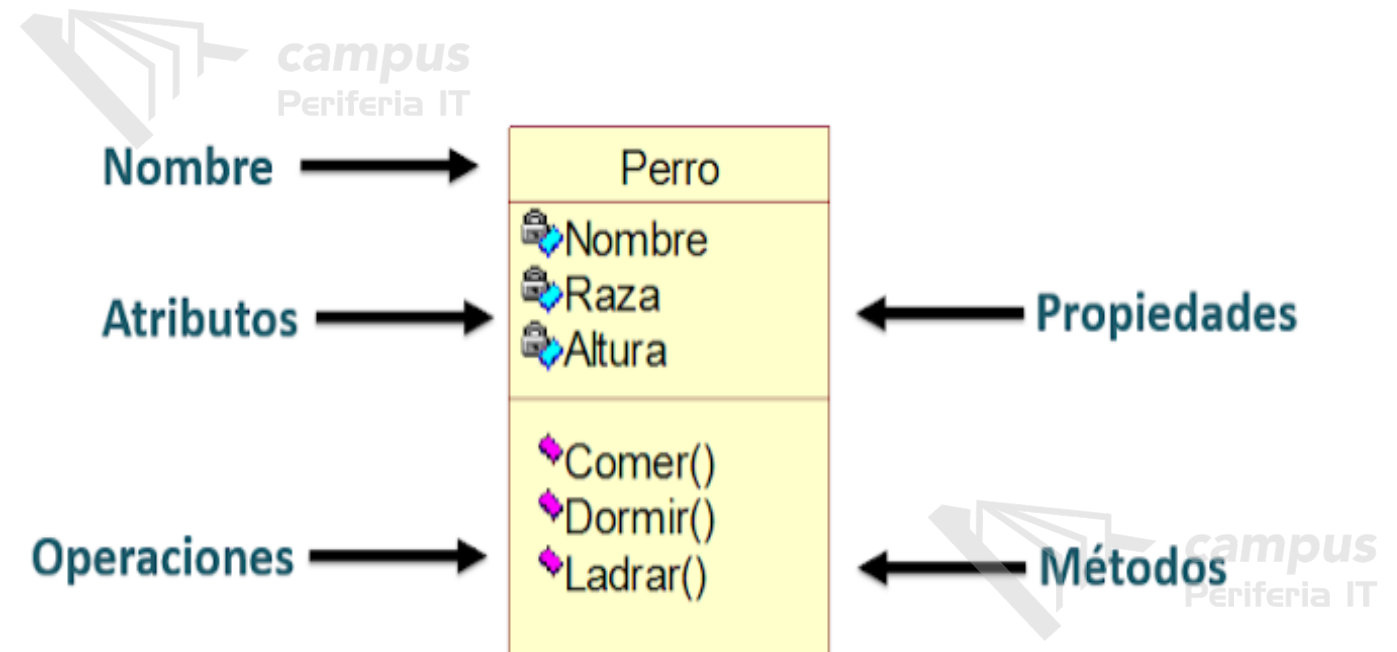


# CLASE

Es una plantilla que se define de manera genérica cómo van a ser los objetos de un determinado tipo.

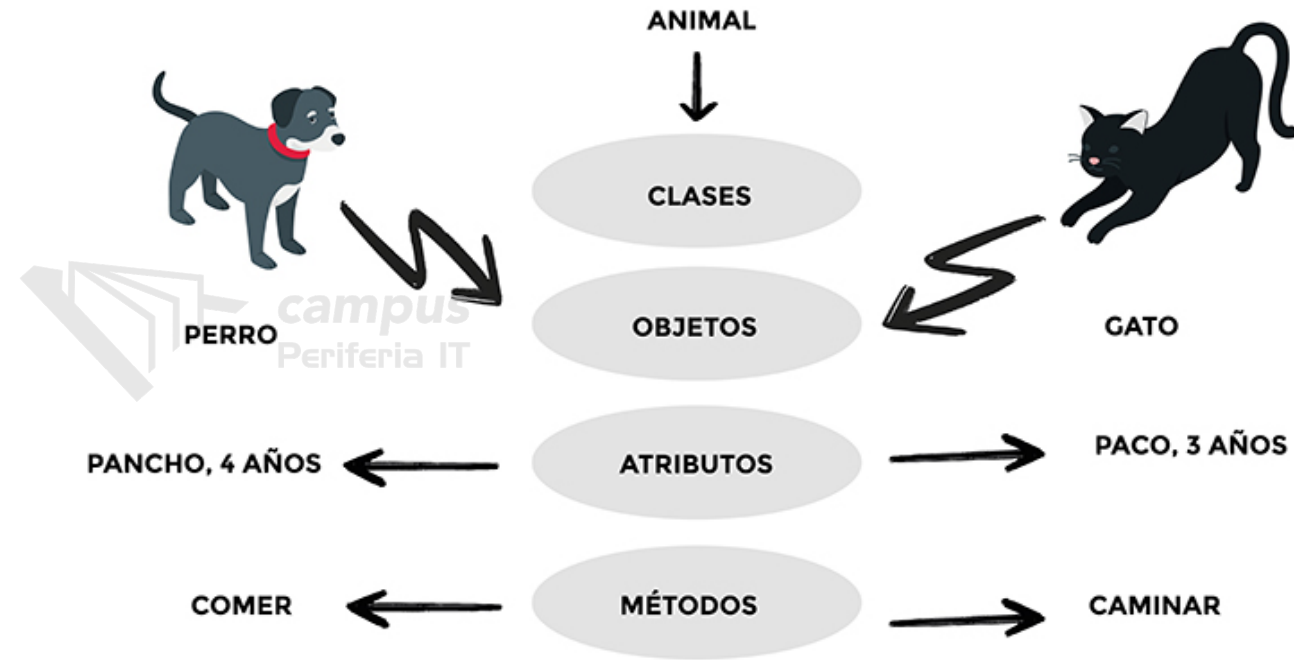
Las clases pueden ser definidas como un molde que contendrá todas las características y acciones con las cuales podemos construir N cantidad de objetos.

Clase
-atributoPrivado: Tipo +atributoPublico: Tipo #atributoProtegido: Tipo
-operacionPrivada() +operacionPublica() #operacionProtegida()



# OBJETO

- Son aquellos que tienen propiedades únicas.
- Son instancias de una clase.
- Sus atributos definidos de forma independiente.





# MÉTODO

- Los métodos son las acciones que una clase puede realizar.
- Definen el comportamiento de los objetos de una clase.

```
package Clases;
```

```
public class Perro {
```

```
    //Propiedades
```

```
    public String nombre;
```

```
    public String raza;
```

```
    public String altura;
```

```
    //Métodos
```

```
    public String comer(String carne)
```

```
    {
```

```
        return nombre + " mide " + altura + " y comerá " + carne;
```

```
    }
```

```
    public void dormir()
```

```
    {
```

```
    }
```

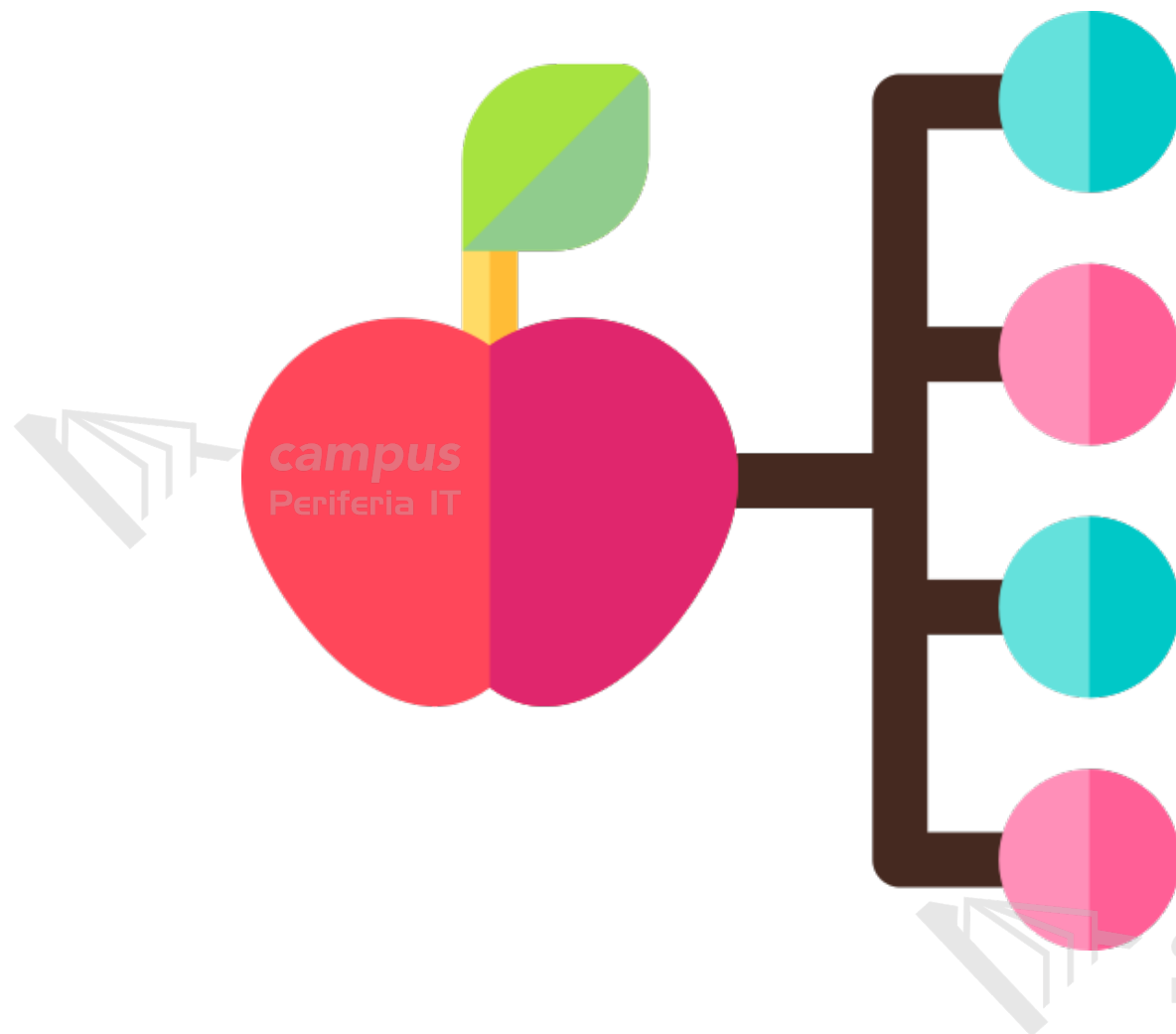
```
    public void ladrar()
```

```
    {
```



# PROPIEDAD

- Las propiedades son las características de una clase.
- Una propiedad es un identificador con un determinado tipo de dato que accede normalmente a un campo en forma directa o a través de un método.
- Otra manera de definirla son las mismas variables en programación estructurada, solo que a diferencia de la POO estas pueden heredarse



## PILARES DE LA POO

### ABSTRACCIÓN

Es cuando separamos los datos de un objeto para luego generar una clase; se define también como las características específicas de un objeto.

### ENCAPSULAMIENTO

Lo puedes utilizar cuando deseas que ciertos métodos o propiedades sean inviolables o inalterables; se define como las precondiciones o métodos previamente validados para q una acción se ejecute.

### HERENCIA

Nos permite crear nuevas clases a partir de otras; Es el mecanismo por el cual una clase permite heredar las características (atributos y métodos) de otra clase.

### POLIMORFISMO

Proviene de Poli = muchas, morfismo = formas. Se utiliza para crear métodos con el mismo nombre, pero con diferente comportamiento.







# CONSTRUCTORES

- Sirve para inicializar un objeto al crearlo.
- Existe sobrecarga (distintos parámetros) (para cualquier método).
- Coincide con el nombre de la clase y no devuelve nada por definición.

```
public Perro()
```

```
{
```

```
}
```

```
public Perro(String nombre,String raza,String altura)
```

```
{
```

```
    this.nombre=nombre;
```

```
    this.raza=raza;
```

```
    this.altura=altura;
```

```
}
```



**campus**  
Periferia IT



G R A C I A S

