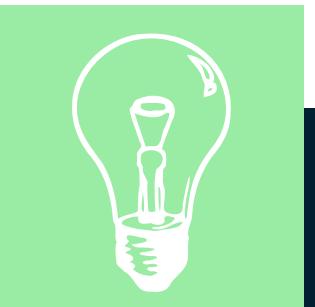




Programación orientada a objetos

Introducción a Python

¿Qué es la programación orientada a objetos?



Abreviatura: POO

La Programación Orientada a Objetos (POO) es un **paradigma de programación**, es decir, 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**.



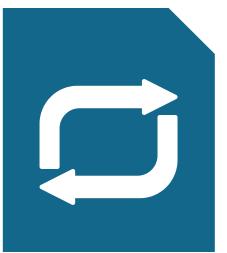
¿En que consiste?

Con el paradigma de Programación Orientado a Objetos lo que buscamos es dejar de centrarnos en la lógica pura de los programas, para empezar a pensar en objetos, lo que constituye la base de este paradigma. La Programación Orientada a objetos permite que el código sea reutilizable, organizado y fácil de mantener.

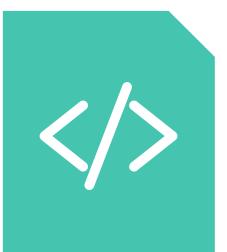
| **En POO todo es considerado como un objeto**



Don't



Repeat



Yourself

Clases, objetos e instancias

¿Cómo se crean los programas orientados a objetos?

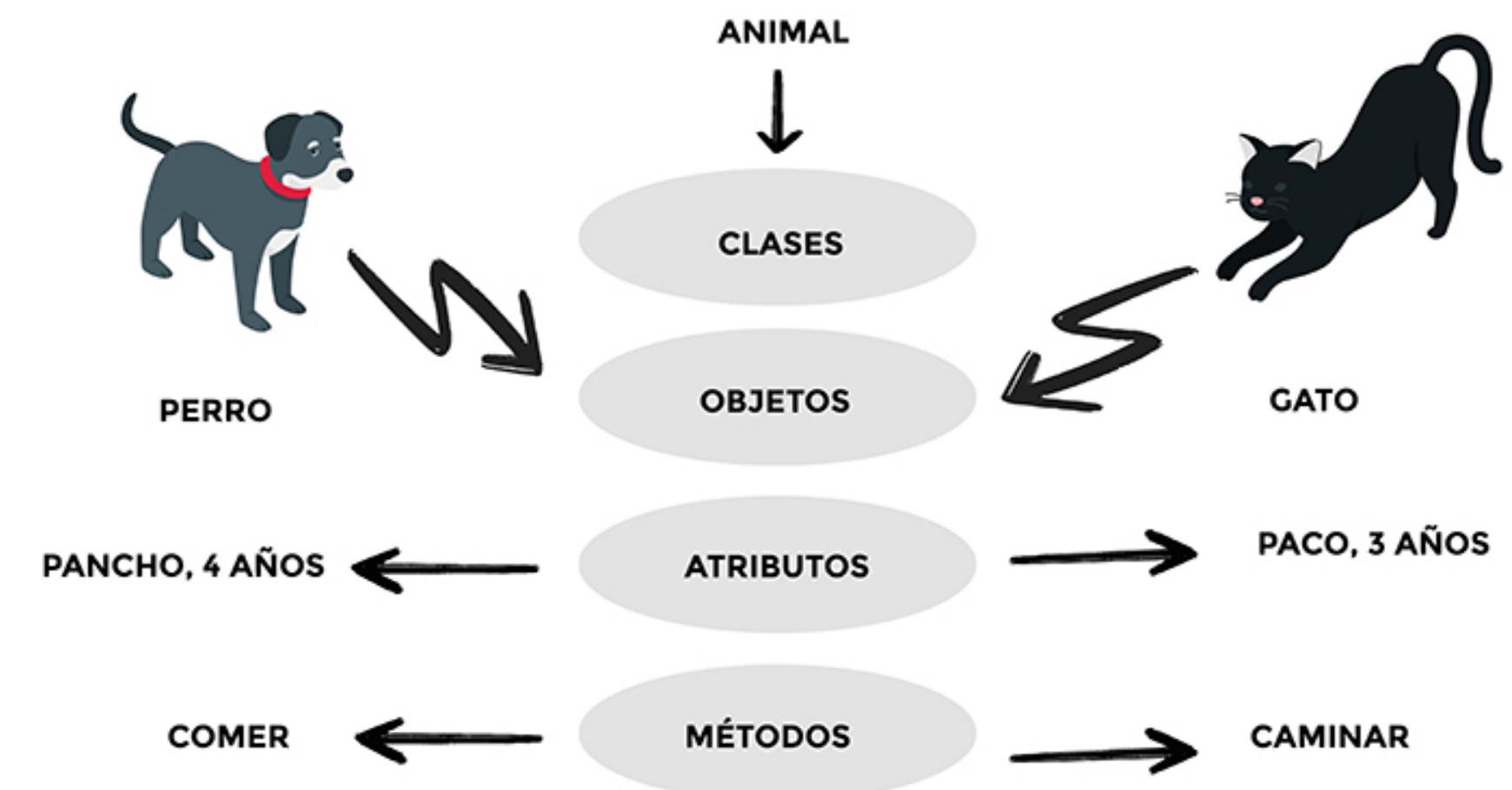
esumiendo mucho, consistiría en hacer clases y crear objetos a partir de estas clases. Las clases forman el modelo a partir del que se estructuran los datos y los comportamientos.



El primer y más importante concepto de la POO es la distinción entre clase y objeto

Clase vs Objeto

Una clase es una plantilla. Define de manera genérica cómo van a ser los objetos de un determinado tipo. Por ejemplo, una clase para representar a animales puede llamarse 'animal' y tener una serie de atributos, como 'nombre' o 'edad' (que normalmente son propiedades), y una serie con los comportamientos que estos pueden tener, como caminar o comer, y que a su vez se implementan como métodos de la clase (funciones). Un ejemplo sencillo de un objeto, como decíamos antes, podría ser un animal.



Principios de la POO

Encapsulamiento

La encapsulación contiene toda la información importante de un objeto dentro del mismo y solo expone la información seleccionada al mundo exterior.

Abstracción

La abstracción es cuando el usuario interactúa solo con los atributos y métodos seleccionados de un objeto, utilizando herramientas simplificadas de alto nivel para acceder a un objeto complejo.

Herencia

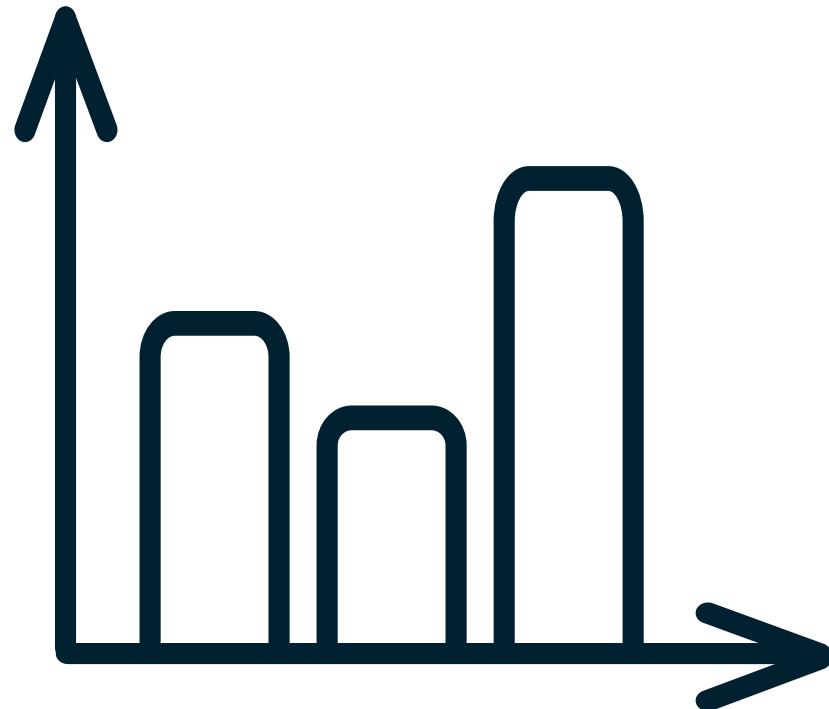
La herencia define relaciones jerárquicas entre clases, de forma que atributos y métodos comunes puedan ser reutilizados. Las clases principales extienden atributos y comportamientos a las clases secundarias.

Polimorfismo

El polimorfismo consiste en **diseñar objetos para compartir comportamientos**, lo que nos permite procesar objetos de diferentes maneras. Es la capacidad de presentar la misma interfaz para diferentes formas subyacentes o tipos de datos.

¿Por qué es uno de los paradigmas más utilizados?

Beneficios



- Reutilización del código.
- Convierte cosas complejas en estructuras simples reproducibles.
- Evita la duplicación de código.
- Permite trabajar en equipo gracias al encapsulamiento ya que minimiza la posibilidad de duplicar funciones cuando varias personas trabajan sobre un mismo objeto al mismo tiempo.
- Al estar la clase bien estructurada permite la corrección de errores en varios lugares del código.
- Protege la información a través de la encapsulación, ya que solo se puede acceder a los datos del objeto a través de propiedades y métodos privados.
- La abstracción nos permite construir sistemas más complejos y de una forma más sencilla y organizada.

POO un paradigma a seguir

Conclusiones

La Programación Orientada a Objetos es actualmente el paradigma que más se utiliza para diseñar aplicaciones y programas informáticos. Son muchas sus ventajas, principalmente cuando necesitas resolver desafíos de programación complejos. Permite una mejor estructura de datos y reutilización del código.





Gracias