



Clases y métodos virtuales



Contenido

- ¿Qué es el Polimorfismo?
- Polimorfismo con Clases Abstractas.
- Ejemplo Práctico en Python.
- Beneficios y Conclusiones.

¿Qué es el Polimorfismo? 🤔

- **Definición:** "La habilidad de un objeto de tomar múltiples formas."
- "Permite que diferentes objetos respondan al mismo mensaje de manera distinta."
- Analogía: Un mismo comando, "sonar", puede ser interpretado de forma diferente por un perro (ladrido) o un gato (maullido).

¿Cómo debería interpretarse el comando "sonar"?

Interpretación del perro

El perro interpreta el comando como una señal para ladrar.



Interpretación del gato

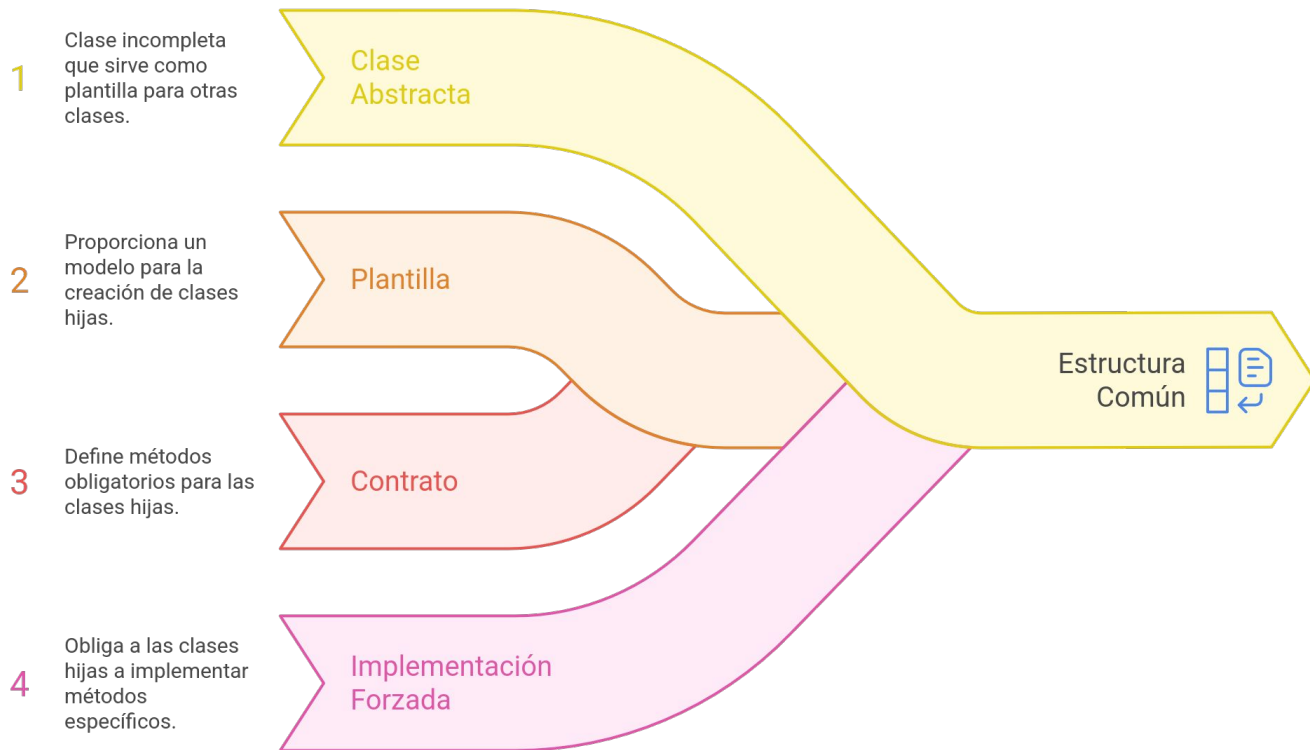
El gato interpreta el comando como una señal para maullar.

Polimorfismo de 'Duck Typing'

- "En Python, el polimorfismo se basa en el 'duck typing'."
- "Si parece un pato y camina como un pato, entonces es un pato."
- "No importa el tipo de objeto, sino lo que puede hacer."
- Ejemplo: Una función que acepta cualquier objeto con un método `vuela()` puede operar tanto en una clase Pájaro como en una clase Avión.



¿Qué es una Clase Abstracta?



¿Qué es un Método Abstracto?

- **Definición:** "Un método **declarado** en la clase abstracta, pero **sin implementación**."
- "Debe ser **sobreescrito** por todas las clases hijas que hereden de ella."
- **Uso:** Se marca con el decorador `@abstractmethod` del módulo `abc` (Abstract Base Classes).

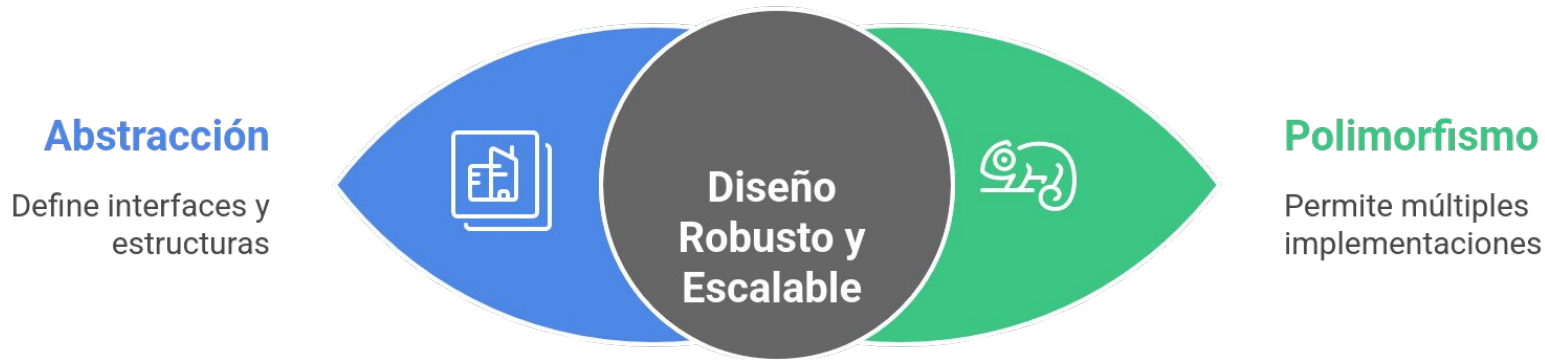
Módulo 'abc' en Python

- "Python utiliza el módulo `abc` para definir clases y métodos abstractos."
- **Pasos clave:**
 1. Importar `ABC` y `abstractmethod` de `abc`.
 2. La clase padre debe heredar de `ABC`.
 3. Marcar los métodos obligatorios con `@abstractmethod`.

```
1  from abc import ABC, abstractmethod
2
3  class Animal(ABC):
4      @abstractmethod
5      def hacer_sonido(self):
6          pass
7
8  class Perro(Animal):
9      def hacer_sonido(self):
10         print("Guau!")
11
12 class Gato(Animal):
13     def hacer_sonido(self):
14         print("Miau!")
```

Polimorfismo y Abstracción

- "La **abstracción** define el qué (la interfaz)."
- "El **polimorfismo** maneja el cómo (la implementación)."
- "Ambos conceptos trabajan juntos para crear diseños más robustos y escalables."



Caso práctico

Problema:

Debemos diseñar un sistema de diseño asistido que debe ser capaz de representar figuras geométricas.

El sistema debe poder ser extensible mediante plugins.

Cada plugin debe implementar una figura geométrica, con la siguiente funcionalidad:

- Se debe poder crear indicando el nombre y la longitud de sus lados
- Debe poder indicar su nombre, su superficie y su perímetro



¡Gracias!

Hasta la próxima

