



Tarea 2

Clases y objetos

Sección 1: Trabajo Colaborativo

Grupo 202047919A_1704 :

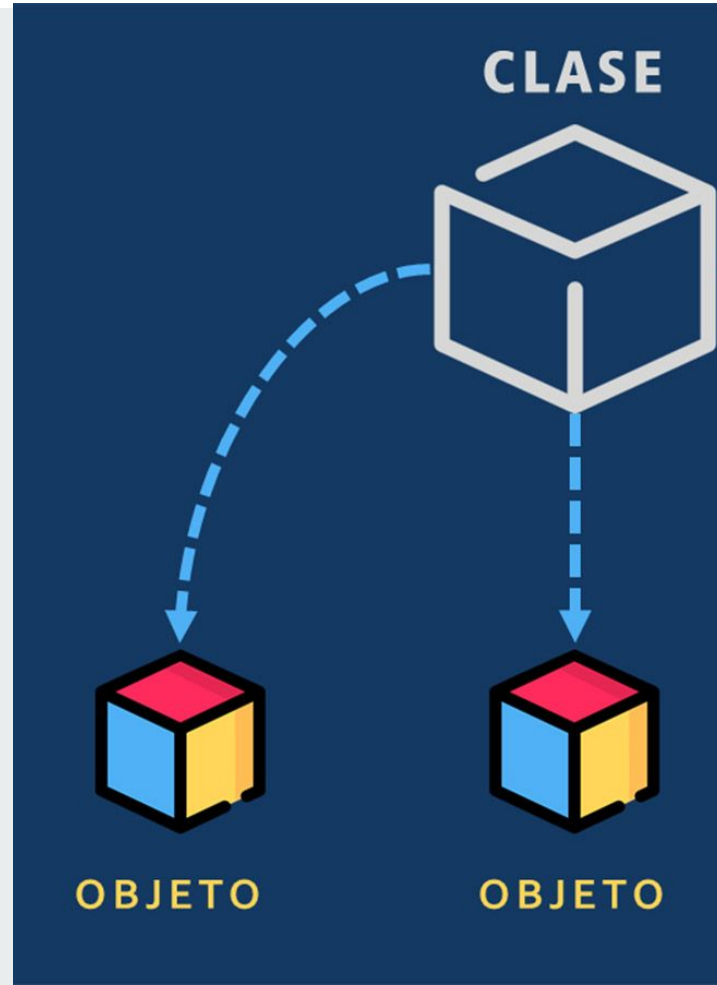
- ❖ Natalia Castaño
- ❖ Carolina Chitiva Muñoz
- ❖ John Leon
- ❖ Diego Santoyo



Universidad Nacional
Abierta y a Distancia

Programación Avanzada

2024





¿Qué es una Abstracción?

Diego Santoyo

En la programación orientada a objetos la abstracción es uno de los principios más importante, consiste en ocultar detalles complejos de un objetos para mostrar sólo información esencial para futuros colaboradores en el código.

Ejemplo:

Si continuamos con los ejemplos utilizados en la primera actividad, y en este caso tenemos el objeto Taladro. La abstracción nos permite interactuar con el taladro usando los métodos perforar() o Cambiar revolución() sin la necesidad de como saber internamente el taladro. En otras palabras, a partir de la abstracción nos enfocamos en lo que hace un objeto y no en cómo lo hace.





¿Qué es encapsulamiento?

John Leon



El encapsulamiento en la programación orientada a objetos se refiere a la práctica de ocultar los detalles internos de una clase y restringir el acceso directo a ciertos atributos o métodos, es decir que podremos clasificar ciertas características o funcionalidades de una clase como privadas, de manera que solo podremos acceder a ellas o modificarlas con algún método específico.

Ejemplo:

Creamos una clase llamada `Cuenta_de_banco` la cual representa un cuenta de banco. Debemos encargarnos de que el saldo de la cuenta no pueda ser modificado directamente desde fuera de la clase, pero si queremos que se pueda acceder o modificar mediante métodos controlados por la clase.

Para ello definimos al atributo `saldo` como privado y definimos métodos dentro de la clase, tales como `depositar`, `retirar` o `consultar_saldo` para poder acceder al saldo o modificarlo. De esta manera no se podrá modificar directamente el atributo `saldo` a menos que utilice un método definido.





¿Qué es herencia?

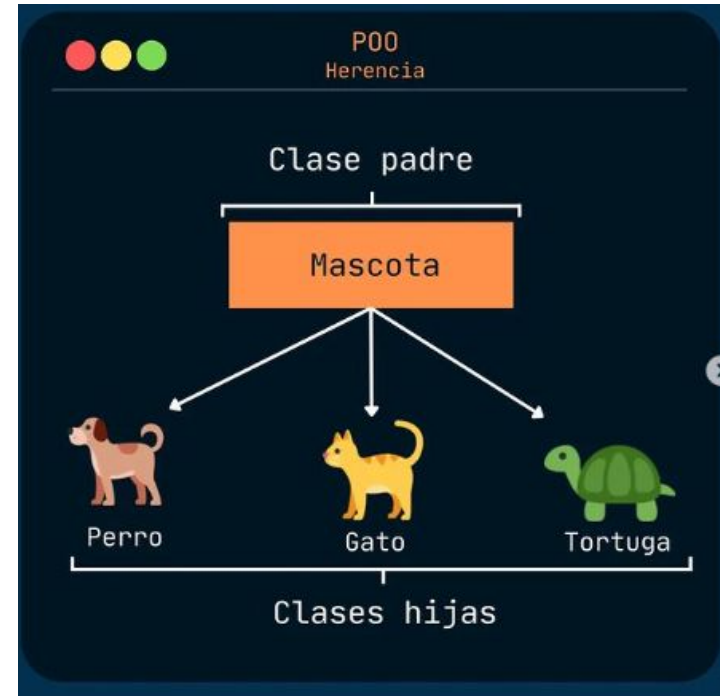
Natalia Castaño

Es un mecanismo que permite crear una nueva clase a partir de una clase existente. La nueva clase, conocida como subclase o clase derivada, herencia atributos y métodos de clase existente, llamada superclase o clase base. La herencia permite reutilizar código, extender funcionalidades y crear una jerarquía de clases más organizada.

Aspectos claves de la herencia en programación:

1. Reutilización de Código:

- Herencia de atributos y métodos:** La subclase hereda todos los atributos y métodos de la superclase, lo que significa que no es necesario reescribir el código que ya ha sido definido en la superclase. Esto fomenta la reutilización del código y reduce la redundancia.

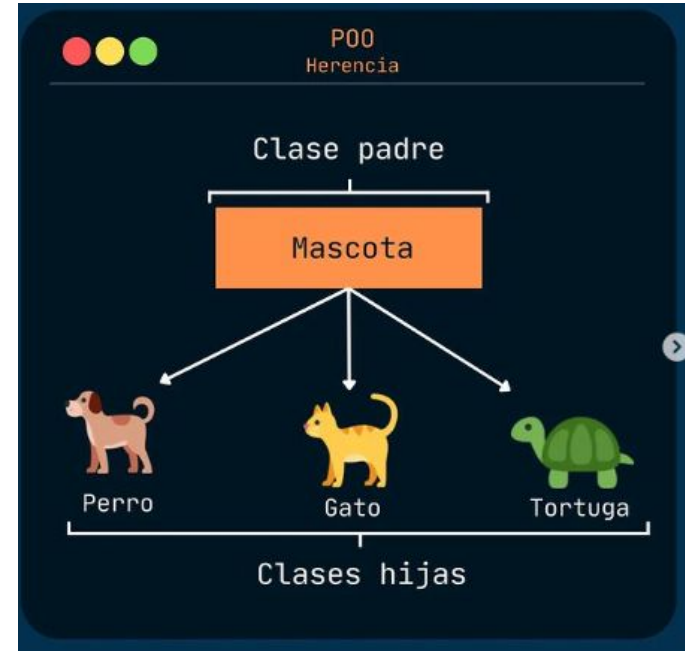




¿Qué es herencia?

Natalia Castaño

- 2. Jerarquía de Clases:** Las clases pueden organizarse en una jerarquía. Por ejemplo, si tienes una clase animal, puedes tener subclases como perro y gato que heredan de animal. De esta forma, perro y gato tienen todas las características de animal, pero también pueden tener sus propias características adicionales.
- 3. Polimorfismo:** la herencia permite el polimorfismo, que es la capacidad de usar una referencia de clase padre para referirse a objetos de una clase hija. Esto facilita la extensión y modificación de comportamientos en tiempos de ejecución.
- 4. Sobreescritura de métodos:** Las clases hijas pueden sobrescribir los métodos de la clase padre para proporcionar una implementación específica. Esto permite que una clase hija modifique o extienda el comportamiento de los métodos heredados.



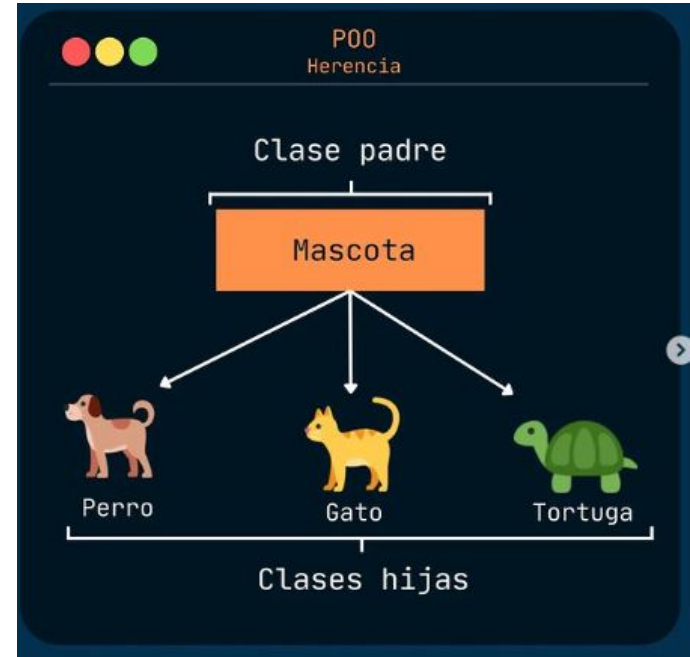


¿Qué es herencia?

Natalia Castaño

- 5. Acceso a métodos y atributos:** la herencia también define como los métodos y atributos de una clase padre son accesibles desde una clase hija. Normalmente, los métodos y atributos públicos y protegidos de la clase padre están disponibles para la clase hija.

En resumen, la herencia es una herramienta poderosa en la programación orientada a objetos que facilita la organización, reutilización y extensión del código.



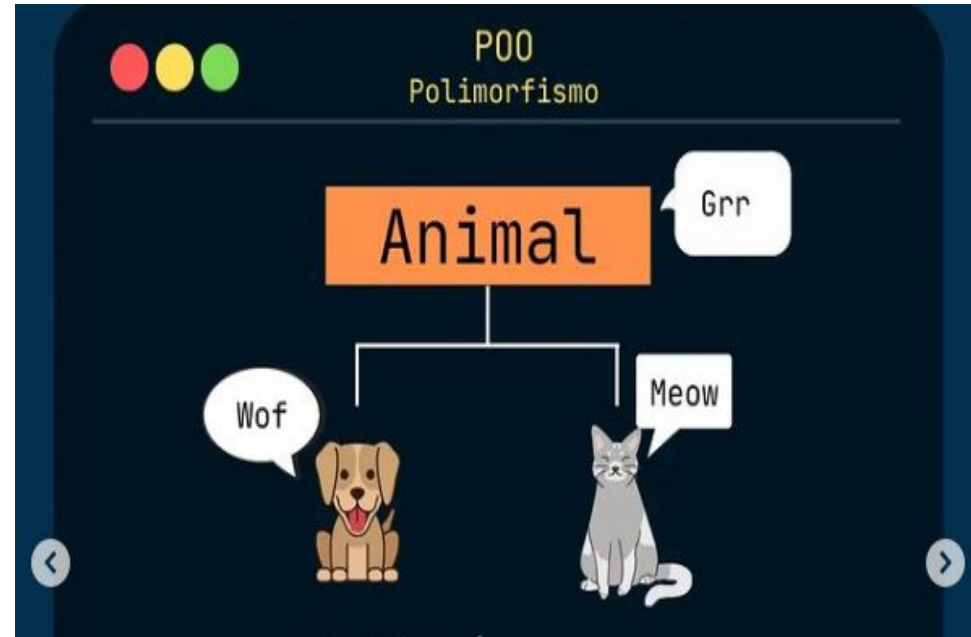


¿Qué es polimorfismo?

LEIDY GARCIA

El polimorfismo es uno de los pilares básicos en la programación orientada a objetos, por lo que para entenderlo es importante tener las bases de la POO y la herencia bien asentadas, significa que objetos de diferentes clases pueden ser accedidos utilizando el mismo interfaz, mostrando un comportamiento distinto (tomando diferentes formas) según cómo sean accedidos. El término polimorfismo visto desde el punto de vista de Python es complicado de explicar, Al ser un lenguaje con tipado dinámico y permitir duck typing, en Python no es necesario que los objetos compartan un interfaz, simplemente basta con que tengan los métodos que se quieren llamar.

Podemos recrear el ejemplo de Java de la siguiente manera. Supongamos que tenemos un clase Animal con un método hablar().



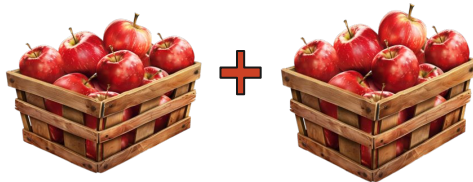


¿En qué consiste la sobrecarga de operadores?

Carolina Chitiva Muñoz

La sobrecarga de operadores es una técnica que permite cambiar el comportamiento de operadores comunes (como +, -, *, etc.) cuando los usamos con objetos creados en nuestras propias clases. Es útil porque podemos hacer que esos operadores hagan cosas personalizadas, en lugar de solo sumar o restar números.

Por ejemplo, imagina que tienes una clase que representa cajas con manzanas. Si quieres sumar las manzanas de dos cajas usando el operador +, podrías hacerlo sobrecargando el operador +.





Referencias

- ❖ *Blog IfGeekThen*. (s/f). Nttdata.com. Recuperado el 14 de septiembre de 2024, de <https://ifgeekthen.nttdata.com/s/post/herencia-en-programacion-orientada-objetos-MCPV3PCZDNBFHSROCCU3JMI7UIJQ?language=es>
- ❖ Cursa. (n.d.). Programación orientada a objetos en Python: Sobrecarga de operadores en Python. Cursa. <https://cursa.app/es/pagina/programacion-orientada-a-objetos-en-python-sobrecarga-de-operadores-en-python>
- ❖ Lutz, M. (2013). *Python 3*. O'Reilly Media. https://www.google.com.co/books/edition/Python_3/KRYyvKmZvpwC?hl=es-419&gbpv=1&dq=en+qu%C3%A9+consiste+la+sobrecarga+de+operadores+python&pg=PA283&printsec=frontcover
- ❖ Meyers, R. (2018). *Comenzar a programar con Python 3*. Anaya Multimedia. https://www.google.com.co/books/edition/Comenzar_a_programar_con_Python_3/5A7PDwAAQBAJ?hl=es-419&gbpv=1&dq=en+qu%C3%A9+consiste+la+sobrecarga+de+operadores+python&pg=PA161&printsec=frontcover
- ❖ Pressman, R. S. (2014). *Ingeniería de software: Un enfoque práctico* (8.^a ed.). McGraw-Hill.