



UEA
UNIVERSIDAD
ESTATAL AMAZÓNICA





UEA
UNIVERSIDAD
ESTATAL AMAZÓNICA

ASIGNATURA

PROGRAMACIÓN ORIENTADA A

OBJETOS



Transformamos el mundo desde la Amazonía

Ing. Edwin Gustavo Fernández Sánchez, Mgs.

DOCENTE - PERSONAL ACADÉMICO NO TITULAR OCASIONAL

DIRECTOR DE GESTIÓN DE TECNOLOGÍAS INFORMACIÓN Y COMUNICACIÓN

UNIVERSIDAD ESTATAL AMAZÓNICA





UEA
UNIVERSIDAD
ESTATAL AMAZÓNICA

SEMANA

2

DESARROLLO DE LA SEMANA 2: DEL LUN. 02 AL DOM. 08 DE DICIEMBRE/2024

Resultado de aprendizaje: Comprender los conceptos fundamentales de la programación orientada a objetos (POO) para el desarrollo de soluciones computacionales de complejidad media.

CONTENIDOS

UNIDAD I : Fundamentos de la programación orientada a objetos - POO

- Tema 1: Fundamentos de la programación orientada a objetos - POO
 - Subtema 1.2 : Técnicas de programación



UEA
UNIVERSIDAD
ESTATAL AMAZÓNICA

Unidad 1

Fundamentos de la programación orientada a objetos

Tema 1

Fundamentos de la programación orientada a objetos - POO



UEA
UNIVERSIDAD
ESTATAL AMAZÓNICA

Subtema 1.1.2: Técnicas de programación

**PROGRAMACIÓN ORIENTADA A
OBJETOS
(UEA-L-UFB-030)**



Técnicas de Programación en el Contexto de la Programación Orientada a Objetos (POO)

Este paradigma, más que un simple enfoque, representa una filosofía de desarrollo que redefine la concepción y construcción de software.

Las Técnicas de Programación son la piedra angular de la POO. Estas técnicas no solo simplifican la complejidad intrínseca del código, sino que también desempeñan un papel fundamental en la creación de sistemas de software modular, mantenibles y reutilizables.



UEA
UNIVERSIDAD
ESTATAL AMAZÓNICA

Técnicas de Programación en el Contexto de la Programación Orientada a Objetos (POO)

Técnicas:

1. Abstracción
2. Encapsulación
3. Herencia
4. Polimorfismo



<https://youtu.be/SI7O81GMG2A?si=-fAyLyHVOPZyeYVvm>



UEA
UNIVERSIDAD
ESTATAL AMAZÓNICA



[dreamstime.com](https://www.dreamstime.com)

ID 190577014 © Michalsuszycki



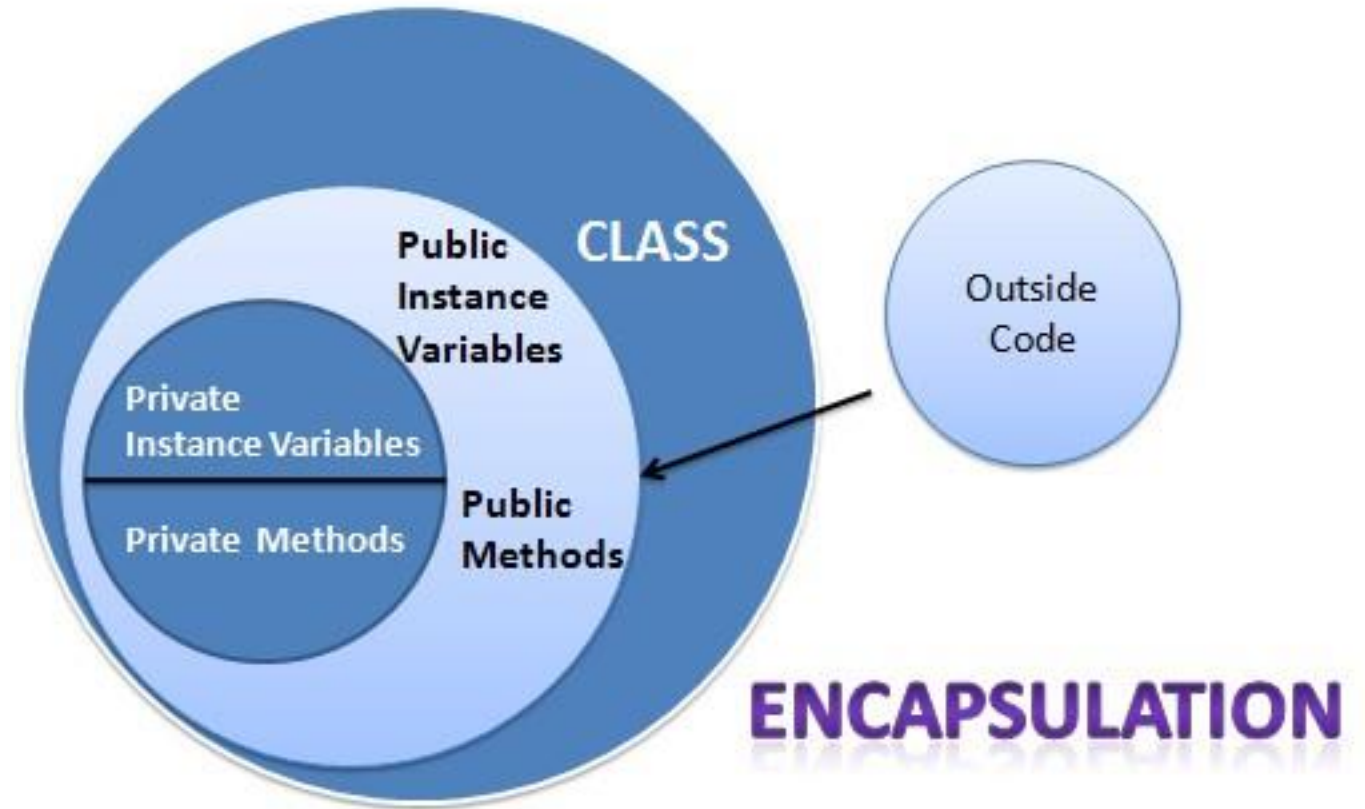
Técnica POO: Abstracción

La Técnica de Abstracción es un pilar fundamental en la Programación Orientada a Objetos (POO). **La abstracción permite a los programadores simplificar la realidad, concentrándose en los aspectos esenciales de un objeto y ocultando los detalles innecesarios.**

Al comprender esta técnica, los desarrolladores pueden crear modelos más comprensibles y manejables, desglosando sistemas complejos en componentes más digeribles. **La abstracción facilita la identificación y aislamiento de características esenciales, permitiendo una comprensión más clara y una manipulación más efectiva de los objetos en el código.**



Encapsulation



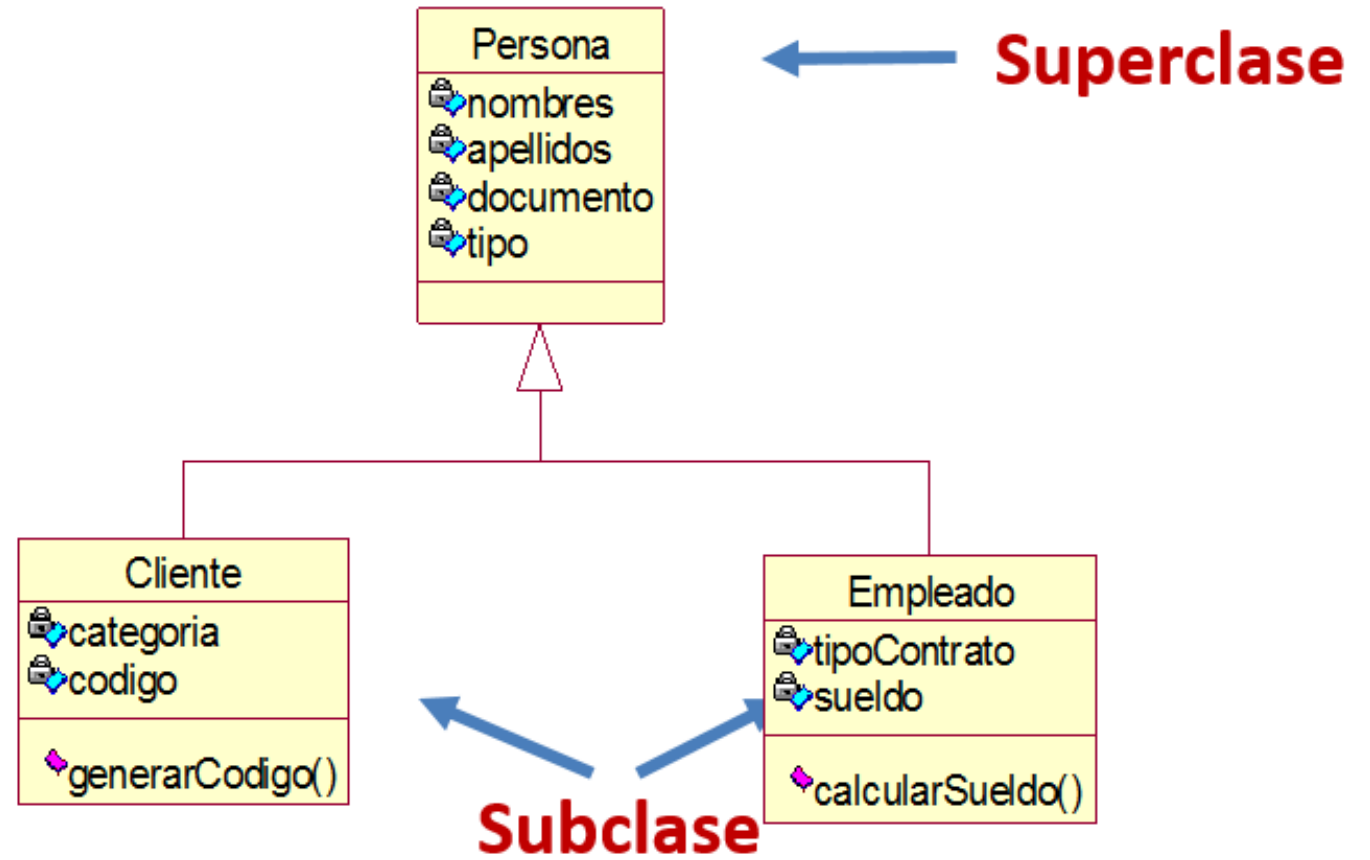
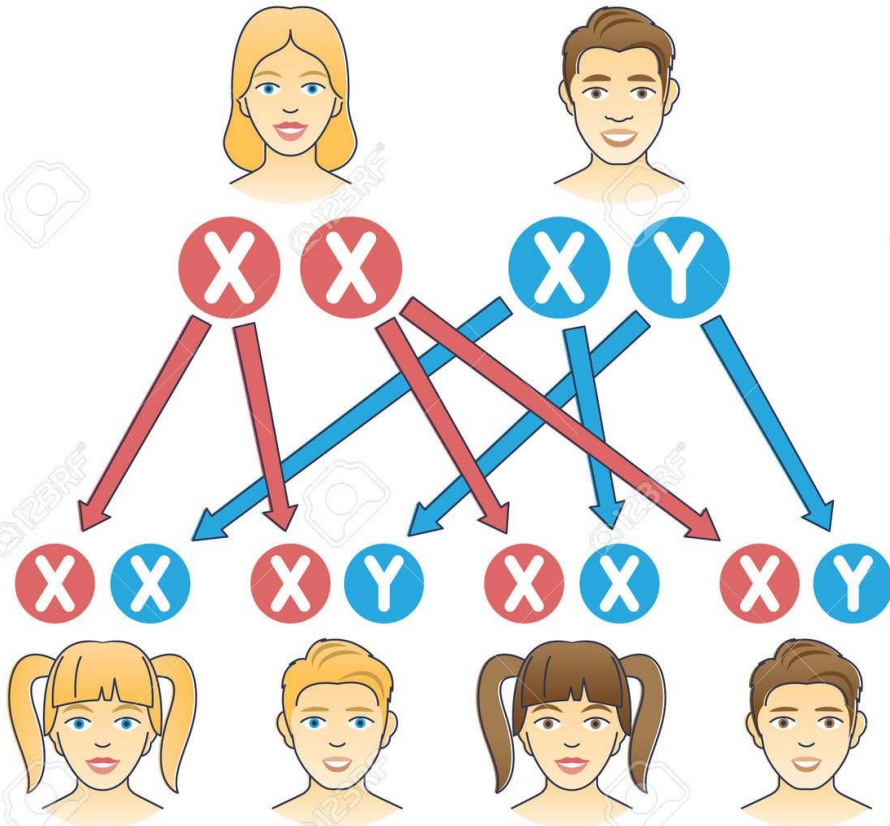


Técnica POO: Encapsulación

Su esencia reside en la capacidad de ocultar la complejidad interna de un objeto y revelar solo la información esencial para su interacción.

Al profundizar en esta técnica, descubriremos cómo los programadores pueden crear clases que actúan como cajas negras, permitiendo el acceso controlado a sus componentes internos. **La encapsulación no solo mejora la seguridad y la integridad del código, sino que también facilita la evolución y el mantenimiento del software a medida que los detalles internos pueden modificarse sin afectar el resto del sistema.**

HEREDITY





Técnica POO: Herencia

La herencia permite la creación de nuevas clases basadas en las características y comportamientos de clases existentes, facilitando la reutilización de código y la construcción progresiva de estructuras de software.

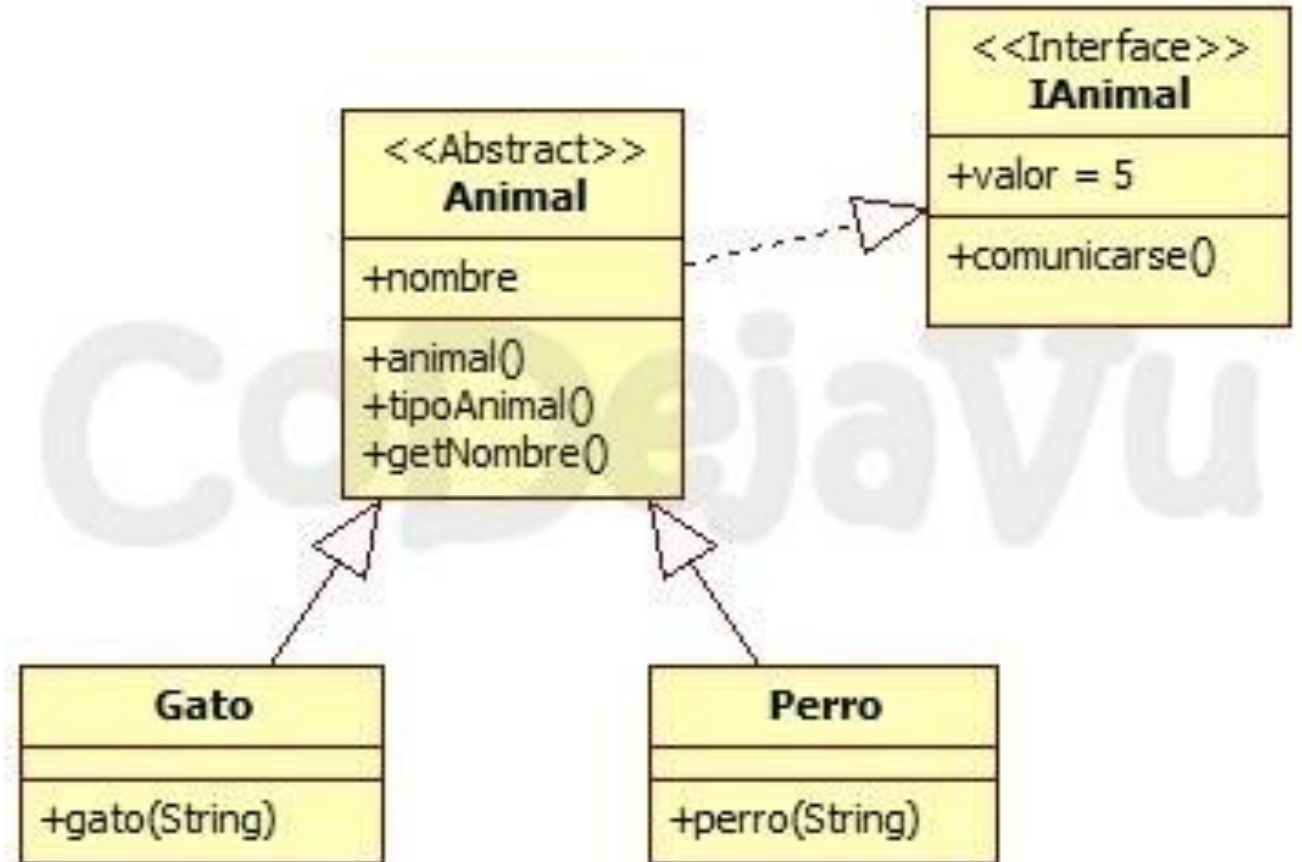
Exploraremos cómo la herencia modela las relaciones jerárquicas entre clases, permitiendo que las subclases hereden atributos y métodos de sus clases ascendentes. Esto no solo promueve la eficiencia en el desarrollo, sino que también fomenta la coherencia y la consistencia en la estructura del código.

krative

VEHÍCULO



POLIMORFISMO





Técnica POO: Polimorfismo

El polimorfismo permite que objetos de diferentes clases sean tratados de manera uniforme, brindando flexibilidad y extensibilidad al código.

Profundizaremos en cómo **el polimorfismo posibilita que múltiples clases compartan una interfaz común, permitiendo que objetos de estas clases sean utilizados de manera intercambiable**. Esto no solo simplifica el código, sino que también fomenta la adaptabilidad y la escalabilidad en el desarrollo de software.



UEA
UNIVERSIDAD
ESTATAL AMAZÓNICA

Ejemplo Python

Ejemplos concretos de cómo aplicar estas técnicas en la escritura de código en un entorno de programación orientada a objetos.



<https://youtu.be/JVNirg9qs4M?si=vGoyCeH-KsK5LNp3&t=44>



UEA
UNIVERSIDAD
ESTATAL AMAZÓNICA