



Emilio Gomez Breschi

19300100

4D1

Sonia Erika Ibáñez de la Torre

Desarrollo de Software

1/Septiembre/2021

Programación Orientada a Objetos

### ***¿Qué es programación orientada a objetos y cuál es su principal ventaja?***

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. Este tipo de programación se utiliza para estructurar un programa de software en piezas simples y reutilizables de planos de código (clases) para crear instancias individuales de objetos.

Ventajas:

- 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.

(Canelo, 2020)

	Como es	Ventajas	Desventajas
Orientada a Objetos	Este tipo de Programación se preocupa por los objetos que se forman en el programa y las operaciones entre ellos	Es más sencillo entender la secuencia de acción de los objetos que una secuencia lógica	Una de las desventajas de este tipo de programación es lo complicado de poder encontrar las clases y objetos adecuados para la elaboración del programa
Estructurada	Este tipo de programación se destaca por seguir una secuencia lógica en la elaboración de un programa	Este tipo de programación es adecuada para aquellos que se van introduciendo a la programación	Una de sus desventajas es que no se pueden hacer programas robustos ya que si se cambia una línea de código hay que cambiar varias ya que todo tiene que seguir una secuencia
Modular	La programación modular es donde se divide el programa en muchos componentes o bloques pequeños autónomos llamados módulos.	Programación independiente del proyecto, Desarrollo eficiente de programas, Uso múltiple de los módulos y Facilidad de depuración y modificación	Estructura excesivamente orientada a objetos, Módulos demasiado pequeños, Descomponer sin razón

(saldivar, 2015) (Corvo, 2020)

## **Modulado de Objetos**

### **Encapsulamiento**

Se denomina encapsulamiento al ocultamiento del estado, es decir, de los datos miembro, de un objeto de manera que sólo se puede cambiar mediante las operaciones definidas para ese objeto.

Cada objeto está aislado del exterior, es un módulo natural, y la aplicación entera se reduce a un agregado o rompecabezas de objetos.



### **Modularidad**

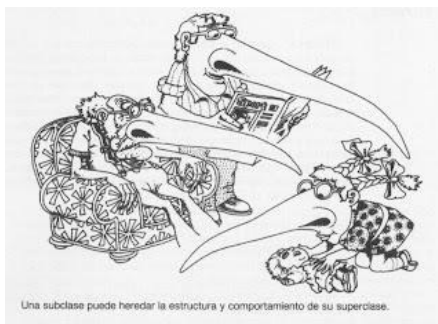
Modularidad permite subdividir una aplicación en partes más pequeñas (llamadas módulos), cada una de las cuales debe ser tan independiente como sea posible de la aplicación en sí y de las restantes partes.

Estos módulos que se puedan compilar por separado, pero que tienen conexiones con otros módulos. Al igual que la encapsulación, los lenguajes soportan la Modularidad de diversas formas.



### **Herencia**

Herencia es el mecanismo fundamental para implementar la reutilización y extensibilidad del software. A través de ella los diseñadores pueden construir nuevas clases partiendo de una **jerarquía** de clases ya existente evitando con ello el rediseño, la modificación y verificación de la parte ya implementada. La herencia facilita la creación de objetos a partir de otros ya existentes, obteniendo características similares a los ya existentes.



Es la relación entre una clase general y otra clase más específica. Por ejemplo: Si declaramos una clase párrafo derivada de una clase texto, todos los métodos y variables asociadas con la clase texto, son automáticamente heredados por la subclase párrafo.

(Lerdo, 2012)

## Referencias

Canelo, M. M. (02 de Noviembre de 2020). *profile*. Recuperado el 01 de Septiembre de 2021, de <https://profile.es/blog/que-es-la-programacion-orientada-a-objetos/>

Corvo, H. S. (08 de Marzo de 2020). *lifeder*. Recuperado el 01 de Septiembre de 2021, de <https://www.lifeder.com/programacion-modular/>

Lerdo. (19 de Mayo de 2012). *Blospot*. Recuperado el 01 de Septiembre de 2021, de <https://pooitsavlerdo.blogspot.com/2012/05/11-elementos-del-modelo-de-objetos.html>

saldivar, J. A. (12 de Abril de 2015). *Youtube*. Recuperado el 01 de Septiembre de 2021, de <https://www.youtube.com/watch?v=8j3reiK96Xo>