

Paradigmas de la Programación

Informe Laboratorio nro. 3

Paradigma Orientado a Objeto

*Profesor: Gonzalo Martínez.*

*Alumna: Karina Bustamante H.*

12 de diciembre de 2023

# Introducción

# Descripción del paradigma

El paradigma orientado a objetos (POO) es un enfoque de programación que se basa en la organización y manipulación de objetos como elementos centrales. Un objeto es una entidad que combina datos (propiedades o atributos) y comportamiento (métodos o funciones) relacionados.

En el paradigma orientado a objetos, los objetos son instancias de clases, que actúan como plantillas o moldes para la creación de objetos. Una clase define las propiedades y los métodos que los objetos de esa clase tendrán. Las propiedades representan el estado o los datos que posee un objeto, mientras que los métodos encapsulan el comportamiento o las acciones que el objeto puede realizar.

La programación orientada a objetos se basa en cuatro conceptos fundamentales:

1. **Abstracción**: Permite representar las características esenciales de un objeto del mundo real en un modelo simplificado dentro del programa. La abstracción se logra mediante la identificación de las propiedades y el comportamiento relevantes para el problema en cuestión.
2. **Encapsulación**: Consiste en agrupar datos y métodos relacionados en un objeto y ocultar los detalles internos del objeto. Esto se logra definiendo la visibilidad de los miembros de la clase (públicos, privados, protegidos) para controlar el acceso desde el exterior.
3. **Herencia**: Permite la creación de nuevas clases (subclases) basadas en clases existentes (superclases). La herencia permite la reutilización de código y establece una relación "es-un" entre las clases, donde una subclase hereda las propiedades y métodos de su superclase.
4. **Polimorfismo**: Permite que objetos de diferentes clases respondan de manera diferente a un mismo mensaje o método. El polimorfismo permite tratar a objetos de distintas clases de manera uniforme, lo que facilita la flexibilidad y la extensibilidad del código.

El paradigma orientado a objetos se utiliza ampliamente en el desarrollo de software debido a su capacidad para modelar de manera eficiente problemas complejos, promover la reutilización de código y mejorar la modularidad y mantenibilidad del sistema.

# Descripción del problema

# Análisis del problema

# Diseño de la solución

# Consideraciones de implementación

# Instrucciones de uso

# Resultados y evaluación

A continuación se muestra la autoevaluación de los requerimientos funcionales requeridos por el laboratorio, y el grado de logro alcanzado para cada uno de ellos.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| RF Nº | Requerimiento funcional | Grado de alcance | Tipo o cantidad de pruebas | Éxitos | Fracasos | Razones de fallo |
| 1 |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |
| 9 |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |

**Tabla 1.-** Muestra los requerimientos funcionales y el nivel de logro.

# 

# Conclusiones

# Referencias