

Cuestionario Programación Orientada a Objetos

Samuel David Ospina Sánchez

23 de Octubre, 2025

Respuestas

1.1 ¿En la programación orientada a objetos qué es el constructor y qué función cumple?

El constructor es un método especial dentro de una clase que se ejecuta automáticamente al crear un objeto. Su función principal es inicializar los atributos del objeto, asignando valores según los parámetros recibidos. También puede establecer conexiones para el funcionamiento del objeto.

1.2 ¿En la programación orientada a objetos qué es un objeto y qué función cumple?

Un objeto es una instancia de una clase. Representa una entidad que combina atributos y métodos. Su función es modelar elementos del mundo real o abstracto dentro del programa, permitiendo la interacción entre componentes mediante el envío de mensajes o llamadas a métodos.

1.3 ¿En la programación orientada a objetos qué es una clase y qué función cumple?

Una clase es una plantilla que define la estructura y el comportamiento de los objetos. Especifica qué atributos y métodos tendrán sus instancias. Su función es organizar y encapsular el código para facilitar la reutilización y organización de programas complejos.

1.4 ¿Qué diferencias destacadas se aprecian entre la OOP y la programación estructurada (secuencial)?

Algunas diferencias clave son:

- La programación estructurada se basa en funciones y el flujo secuencial de instrucciones, mientras que la OOP se basa en objetos que interactúan entre sí.
- En OOP se promueve la encapsulación, herencia y polimorfismo; en la estructurada se enfatiza la división del programa en funciones o procedimientos.
- La OOP facilita la reutilización y escalabilidad del código, mientras que la estructurada puede volverse difícil de mantener en sistemas grandes.

1.5 ¿Para qué y por qué se usan modificadores de acceso?

Los modificadores de acceso (`public`, `private`, `protected`) se utilizan para controlar la visibilidad de los atributos y métodos de una clase. Su propósito es proteger la integridad de los datos, evitar el acceso no autorizado y fomentar el principio de encapsulación. Permiten definir qué partes del código pueden interactuar con ciertos elementos, esto para mejorar la seguridad y el diseño del software.