



# Guía de estudio

## 1. ¿Qué es un paradigma?

Conjunto de teorías, estándares, métodos que, unificados, representan una forma de organizar el conocimiento.

Forma de “ver” y “entender” el mundo, es una forma de abstraerlo dentro de nuestra cabeza.

## 2. ¿Qué define el paradigma OO?

• Componentes independientes y altamente reutilizables. • Comunicación mediante colaboraciones. • Bloques de construcción → objetos y clases.

Los objetos administran sus propios datos y son responsables de sus propias funciones.

Ayuda a combatir la complejidad.

## 3. Defina: Análisis OO, Diseño OO y programación OO

- Análisis OO: Es el proceso de examinar un problema y su entorno para identificar las clases y objetos que formarán parte del sistema, así como sus relaciones y responsabilidades.
- Diseño OO: Involucra la definición de la arquitectura del sistema mediante la especificación de las clases, sus atributos y métodos, y cómo interactúan entre sí para cumplir con los requisitos del sistema.
- Programación OO: Es la implementación del diseño en un lenguaje de programación orientado a objetos, donde se crean las clases y objetos definidos en el diseño y se implementan sus métodos.

## 4. ¿Cuáles son las ventajas de la utilización del paradigma de objetos para el desarrollo de software?

- Reutilización: Las clases y objetos pueden ser reutilizados en diferentes partes del programa o en otros proyectos.
- Modularidad: Facilita la división del sistema en módulos independientes, lo que simplifica el mantenimiento y la comprensión del código.
- Escalabilidad: Permite el desarrollo de sistemas grandes y complejos de manera más manejable.
- Facilidad de Mantenimiento: La encapsulación ayuda a aislar cambios en el sistema, reduciendo el impacto en otras partes del código.

## 5. ¿Cómo resuelve el paradigma de objetos la problemática de la complejidad del software?

El paradigma de objetos maneja la complejidad mediante la abstracción, encapsulamiento, modularidad y jerarquía. La abstracción permite centrarse en aspectos relevantes del problema; el encapsulamiento oculta detalles internos de los objetos; la modularidad divide el sistema en partes manejables y la jerarquía de clases organiza el sistema en diferentes niveles.

## 6. Definir clase y objeto. Dar ejemplos

### Clase

Descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, relaciones y semántica.

- Está compuesta por características (atributos) y comportamientos (métodos). Ambos están determinados por el contexto del problema.

Alumno, Bebida, Vehículo, etc.

### Objetos

Es una instancia de una clase, un caso particular, una entidad concreta.

“Los objetos son entidades del mundo real que combinan estado, comportamiento e identidad”

- El comportamiento está definido por los métodos con que puede operar dicho objeto, es decir, qué operaciones se pueden realizar con él.
- El estado representa uno o varios atributos a los que se habrán asignado unos valores concretos (datos).
- La identidad es una propiedad de un objeto que lo diferencia del resto, dicho con otras palabras, es su identificador

Mateo Letona, Coca Cola, Ciclomotor, etc.

## 7. ¿Cuáles son las características de los objetos según su naturaleza?

Las características de un objeto son sus atributos. Estos son dependientes del dominio del problema.

En la clase Alumno no nos interesa saber su color de pelo o no es redundante para el sistema.

## 8. ¿Cuáles son los elementos esenciales y los secundarios del modelo orientado a objetos? Dé un ejemplo de cada uno (que no sean los del libro)

### Elementos esenciales

#### Abstracción

Extraer las propiedades esenciales de un objeto que lo distinguen de los demás tipos de objetos y proporciona fronteras conceptuales definidas respecto al punto de vista del observador, en un determinado

dominio del problema.

Clase Vehículo  $\rightarrow$  Ruedas, Motor, Asientos

### Encapsulamiento

- Ocultación de detalles
- Implementación
- Concepto de Interfaz
- Independencia

En una clase Tarjeta, sería conveniente ocultar atributos como el número, el CVV.

### Jerarquía

Orden de abstracción organizado por niveles.

La clase animal se crean las subclases de perro, gato, etc. También podríamos tener más subclases diferenciando los animales por reinos y otras características.

Elementos secundarios

### Tipificación

Características precisas que comparten una serie de objetos. Puesta en vigencia de la clase de los objetos.

Auto sus atributos serían: marca, modelo, color

### Concurrencia

Permite a diferentes objetos actuar al mismo tiempo. Distingue objetos activos de inactivos.

En un sistema operativo, la concurrencia es necesaria para el funcionamiento de múltiples programas al mismo tiempo

### Persistencia

Cantidad de espacio que ocupa y tiempo que dura un objeto. Conservación del estado del objeto en el espacio y en el tiempo.

En una biblioteca, la **persistencia** permite guardar Libros y Préstamos en un almacenamiento para luego poder ser consultado.

9. ¿Qué permite modelar el Modelo de Objetos del Dominio del Problema? ¿Qué herramienta se utiliza para desarrollarlo?

DIAGRAMA DE CLASE

UML

## 10. ¿Cómo se identifican las clases que participan de un dominio de problema?

Las clases se identifican a través del análisis de requisitos y del entendimiento del dominio del problema y son todo aquello que se necesita que el sistema administre dentro del dominio de análisis.

Un ayuda memoria podría ser buscar dentro del dominio del problema

Cosas tangibles, Personas, Lugares, Transacciones u operaciones, Hechos o eventos, Otras organizaciones o áreas

## 11. El diagrama de clases qué modela del sistema? ¿Qué vista del sistema muestra, dinámica o estática, interna o externa? Explique

- El diagrama de clases modela la estructura estática del sistema, mostrando las clases, sus atributos, métodos, y las relaciones entre ellas. Ofrece una vista estática y externa del sistema, enfocándose en cómo están organizados los componentes en lugar de cómo se comportan durante la ejecución.

## 12. ¿Cuáles son los elementos que conforman una clase?

- Atributos: Datos que representan el estado de la clase.
- Métodos: Funciones que definen el comportamiento de la clase.
- Constructores: Métodos especiales utilizados para crear instancias de la clase.
- Relaciones: Asociaciones con otras clases.

## 13. ¿Cuáles son las vistas de una clase?

- Vista Interna: Detalles de implementación, incluyendo atributos y métodos privados.
- Vista Externa: Interfaz pública, mostrando los métodos y atributos accesibles a otros objetos.
- Vista Conceptual: El propósito y el rol de la clase dentro del sistema, sin entrar en detalles de implementación.

## 14. Envío de mensajes

- En la programación orientada a objetos, el envío de mensajes es la manera en que los objetos se comunican entre sí. Un objeto envía un mensaje a otro objeto invocando uno de sus métodos, lo que puede implicar pasar información (argumentos) y recibir una respuesta (valor de retorno)