

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?

El paradigma orientado a objetos define clases como abstracciones de los objetos del mundo real. Estas clases tienen atributos y métodos que definen su comportamiento y estado. A partir de estas clases, se crean instancias llamadas objetos que interactúan entre sí para resolver problemas.

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

Análisis Orientado a Objetos

Proceso de examinar los requisitos y especificaciones de un sistema para identificar los objetos y sus relaciones, que formarán la base del diseño orientado a objetos.

Diseño Orientado a Objetos

Enfoque de diseño de software que utiliza objetos y sus interacciones para modelar y resolver problemas, centrándose en la definición de clases y sus atributos, métodos y relaciones.

Programación Orientada a Objetos

Paradigma de programación basado en el uso de objetos, que son instancias de clases que encapsulan datos y comportamiento.

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

- Componentes independientes y altamente reutilizables.
- Comunicación mediante colaboraciones.
- Construye a través de objetos y clases.

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

Ayuda a combatir la complejidad.

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

El paradigma orientado a objetos resuelve la problemática de la complejidad del software mediante el uso del encapsulamiento, la abstracción, la herencia, el polimorfismo y la modularidad. Esto permite que el software sea más fácil de entender, de mantener y de reutilizar.

6. Definir clase y objeto. Dar ejemplos

Clase

Descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, relaciones y semántica. Pueden representar software, hardware o cosas puramente conceptuales.

Una clase es una abstracción de las cosas que forman parte del vocabulario.

Una clase no es un objeto individual, sino que representa un conjunto de objetos.

Auto, Profesor, Comida, etc.

Objetos

Instancia de una clase.

Entidad discreta con identidad, estado y comportamiento que se puede invocar.

- 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.

Fiat 600, Valeria Ortiz, Pizza, 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 un taller de pintura de vehículos, tenemos en cuenta el color del vehículo y no el estado de su motor.

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.

```
$$ Animal $$  
$$ \downarrow $$  
$$ Mamífero - \ Anfibios - \ Peces - \ Reptiles $$
```

Elementos secundarios

Tipificación:

Comprobación de que los valores y las variables en un programa se utilicen de acuerdo a sus tipos definidos. Esto previene errores y asegura la integridad del comportamiento del programa al restringir las operaciones permitidas según el tipo de dato.

Auto sus atributos serían: marca, modelo, color

Concurrencia:

Capacidad de un sistema para ejecutar múltiples procesos o hilos de manera simultánea, mejorando la eficiencia y capacidad de respuesta del sistema.

En un banco, la **concurrencia** permite la posibilidad de que varias personas puedan transferir, retirar y consultar su saldo al mismo tiempo.

Persistencia:

Capacidad de un objeto para conservar su estado más allá de la ejecución del programa, generalmente mediante almacenamiento en bases de datos o archivos.

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?

El modelo de objetos del dominio del problema permite modelar el vocabulario y los conceptos claves del dominio del problema. Dentro de ellos, se identifican todas las entidades que forman parte del sistema y se definen sus atributos y operaciones.

Para desarrollarlo, se utiliza el **diagrama de clases**.

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

Las clases son todo aquello que se necesita que el sistema administre dentro del dominio de análisis.

- Cosas tangibles (Producto, Medicamento, Casa, Avión)
- Roles de personas (Proveedor, Cliente, Empleado)
- Lugares (Barrio, Provincia, País, Zona)
- Transacciones u operaciones (Venta, Préstamo, Pedido)
- Hechos o eventos (Vuelo, Accidente, Incidente)
- Otras organizaciones o áreas (Institución, Banco, Departamento de ventas)

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 y las relaciones estáticas entre las clases y objetos del sistema. No describe cómo interactúan los objetos, sino cómo están organizados y cómo se relacionan entre sí.

En relación con la vista del sistema, el diagrama de clases ofrece una perspectiva **estática**, como se mencionó anteriormente. Por otro lado, en los *diagramas de componentes*, que constituyen una variante de los diagramas de clases, se presenta la vista **interna** del sistema. Esto incluye la encapsulación de una clase, interfaces, puertos, etc.

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

Una clase está conformada por:

- **Nombre:** Cadena de texto. Puede ser un nombre simple o calificado **calificado: nombre de la clase precedido por el nombre del paquete en el que se encuentra**
- **Atributos:** Propiedades identificadas con un nombre que describen un rango de valores que pueden tomar las instancias de la propiedad.
- **Operaciones:** Implementación de un servicio que puede ser requerido a cualquier objeto de la clase para que muestre un comportamiento. Abstracción de algo que se puede hacer a un objeto y que es compartido por todos los objetos de la clase.
- **Responsabilidad:** Contrato u obligación de una clase. Las cosas que debe hacer.

13. ¿Cuáles son las vistas de una clase?

- **Vista de diseño/interfaz:** Como se ve la clase desde afuera.
- **Vista de implementación:** Como se ve la clase desde adentro.

14. Envío de mensajes

El envío de mensajes es la forma en que los objetos interactúan entre sí. Un objeto envía un mensaje a otro objeto para solicitarle que realice una operación. El objeto que recibe el mensaje es el que decide cómo responder a ese mensaje. Esto es llevado a cabo a través de la **invocación de métodos**.