

Modulo 4

Tarea individual AE3_ABP

Emily Quintana

Parte 1: Preguntas Teóricas (Responde en tus propias palabras)

1. ¿Qué es un diagrama de clases y para qué sirve en el desarrollo de software?

R. Es una herramienta visual que permite visualizar un sistema, en él se muestra las clases, sus atributos, métodos y las relaciones entre ellas y esto nos orienta en el desarrollo del código.

2. Explica la notación básica utilizada en un diagrama de clases UML.

R. Cada clase esta representada por un rectángulo dividido en 03 secciones, superior se anota el nombre de la clase centrada y en negrita. En la sección central los atributos con letras minúsculas allí también se le determina la visibilidad que tendrá el atributo poniendo:

Para especificar la visibilidad de un miembro de la clase (es decir, cualquier atributo o método), se coloca uno de los siguientes signos delante de ese atributo:

- + Público
- Privado
- # Protegido
- / Derivado (se puede combinar con otro)
- ~ Paquete

En la sección inferior se definen los métodos, también incluyendo la visibilidad de cada uno.

3. ¿Cómo se representan los atributos y métodos en un diagrama de clases? Proporciona un ejemplo simple con una clase "Producto".

Producto	Nombre de la clase
+precio: +marca: #id:	Atributos
+ obtener_precio (): float -aplicar_descuento(): float +mostrar_stock(): int	Métodos

4. ¿Qué diferencia hay entre colaboración y composición en un diagrama de clases? Explica con ejemplos.

R. La mayor diferencia es el nivel de Dependencia, en la composición es Fuerte, La parte no puede existir sin el todo. En la colaboración es una dependencia Débil. La parte puede existir independientemente del todo.

Ejemplo colaboración: Una universidad y sus profesores. La Universidad está compuesta por muchos Profesor. Sin embargo, si la Universidad cierra, los Profesor no desaparecen. Pueden seguir existiendo y enseñar en otras universidades.

Otra diferencia es visualmente como se representan el diagrama, para composición el extremo es un rombo relleno para la colaboración es un rombo vacío.

Ejemplo de Composición: Computador y CPU

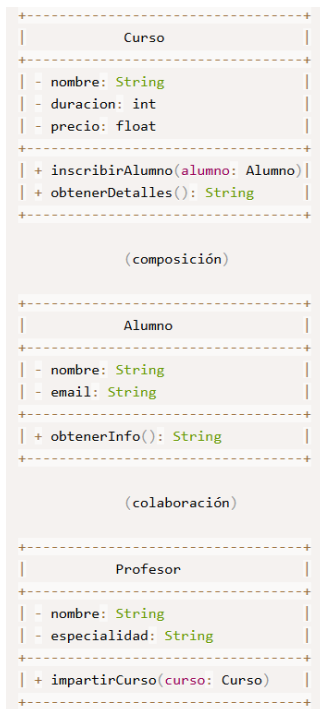
El computador es el “Todo” y el CPU es una parte vital de el. _Un computador no puede funcionar sin un CPU, y los CPU no funcionan si no están en un computador.

5. ¿Por qué es importante leer correctamente un diagrama de clases antes de implementarlo en código?

Nos permite ver la estructura de las clases, si hay herencia que implementar, las acciones (métodos) a definir... es un plano de orientación para ir haciendo el código de forma ordenada. Facilita el trabajo en equipo, pues permite que cada uno tenga una visión global del programa y las relaciones entre cada parte que les toque.

Parte 2: Interpretación de un Diagrama de Clases

A continuación, se presenta un diagrama de clases que modela un sistema de gestión de cursos en una plataforma educativa. Analiza el diagrama y responde las preguntas.



Preguntas sobre el diagrama:

¿Qué clases componen el sistema representado en el diagrama?

Clases: Curso, Alumno, profesor.

¿Cuáles son los atributos de la clase Curso y qué tipos de datos tienen?

Nombre: texto (string)

Duración: entero (int)

Precio: reales, con decimales (float)

¿Qué método de la clase Curso permite inscribir a un alumno?

InscribirAlumno

¿Cómo se representa la composición en el diagrama? ¿Qué significa?

Que un curso no puede existir no hay alumnos.

Si el objeto Curso es destruido (por ejemplo, el curso se elimina del sistema), todos los objetos Alumno que estaban inscritos en él también deberían ser eliminados.

¿Cómo se representa la colaboración en el diagrama? ¿Qué significa?

Un Profesor puede tener muchos Alumnos a lo largo del tiempo, y un Alumno puede interactuar con muchos Profesores. No hay una restricción de "pertenencia" estricta como en la composición.

Parte 3: Creación de un Diagrama de Clases

Enunciado

Se te ha asignado la tarea de diseñar un diagrama de clases UML para un sistema de gestión de bibliotecas. El sistema debe incluir las siguientes entidades:

Libro: Tiene un título, un autor y una disponibilidad (si está prestado o no).

Usuario: Representa a una persona que puede tomar libros prestados. Debe tener un nombre y un ID de usuario.

Préstamo: Relaciona un Usuario con un Libro y registra la fecha de préstamo y la fecha de devolución esperada.

