

Trabajo Practico N:1 | Desarrollo BackEnd

Jeremias Rodriguez Guzman

1. ¿Que es una clase y atributo en OOP (Object-Oriented Programming)?

Una **clase** es una plantilla o modelo que define las propiedades (atributos) y comportamientos (métodos) comunes para un tipo particular de objeto. Un **atributo** es una variable que almacena datos específicos relacionados con una clase; son las características o propiedades que cada objeto de la clase tendrá. Por ejemplo, una clase **Coche** puede tener atributos como **color** y **marca**.

2. Modelado de clases para una compañía aérea en una base de datos relacional

Piloto:

Atributo	Tipo de Dato	Descripcion
codigo	int	ID único del piloto
nombre	String	Nombre del piloto
horasDeVuelo	int	Cantidad de horas de vuelo acumuladas
base	String	Base de operaciones del piloto

MiembroDeTripulacion:

Atributo	Tipo de Dato	Descripcion
codigo	int	ID único del miembro de tripulacion
nombre	String	Nombre del miembro de tripulación
base	String	Base de operaciones del miembro de tripulacion

Avion:

Atributo	Tipo de Dato	Descripcion
codigo	int	ID único del miembro del avion
tipo	String	Tipo de avión (por ejemplo, "BOEING-747").
baseMantenimiento	String	Base donde el avión es sometido a mantenimiento.

Vuelo:

Atributo	Tipo de Dato	Descripcion
numeroVuelo	String	Número de vuelo (por ejemplo, "IB-8830").
origen	String	Ciudad de origen del vuelo.
destino	String	Ciudad de destino del vuelo.
horaSalida	String	Hora de salida del vuelo.
avionCodigo	int	Identificador del avión asignado al vuelo (FK).
pilotoCodigo	int	Identificador del piloto asignado al vuelo (FK).
miembrosTripulacion	List<Integer>	Lista de códigos de los miembros de tripulación (FKs).

3. ¿Cuáles son las características de una clase?

Las características de una clase incluyen:

- **Encapsulamiento:** Agrupa atributos y métodos dentro de una unidad.
- **Herencia:** Permite que una clase derive de otra clase.
- **Polimorfismo:** Permite que los métodos se comporten de diferentes maneras.
- **Abstracción:** Permite centrarse en los aspectos esenciales ignorando los detalles no relevantes.

4. Qué es una interfaz? ¿Cuál es la diferencia entre una clase y una interfaz?

Una **interfaz** es un contrato que define un conjunto de métodos que una clase debe implementar, pero no proporciona la implementación de esos métodos. La diferencia principal es que una **clase** puede contener atributos y métodos con lógica implementada, mientras que una **interfaz** solo define métodos sin implementación.

5. ¿Cuáles son las relaciones que existen entre clases?

Las relaciones entre clases son:

- **Herencia:** Una clase hereda atributos y métodos de otra.
- **Asociación:** Una clase utiliza o está asociada a otra.
- **Agregación:** Una relación "débil" donde una clase contiene a otra sin poseerla.
- **Composición:** Una relación "fuerte" donde una clase contiene a otra y depende de su existencia.

6. ¿Cuáles son las diferencias entre Agregación y Composición?

Agregación: Relación donde una clase contiene referencias a objetos de otra clase, pero estos objetos pueden existir independientemente. Ejemplo: Un equipo tiene jugadores.

Composición: Relación donde una clase contiene objetos de otra clase, y estos objetos no pueden existir sin la clase contenedora. Ejemplo: Una casa tiene habitaciones.

7. ¿Qué es un diagrama de clases?

Un **diagrama de clases** es una representación gráfica de la estructura de un sistema, mostrando sus clases, atributos, métodos, y las relaciones entre ellos. Es una herramienta esencial en UML (Unified Modeling Language) para modelar la arquitectura de software.

8. Diagrama de clases para una agencia de alquiler de autos

