

Desarrollo Back End – 1º Trabajo Práctico

Quevedo Diego

1. Clases y Atributos en OPP

En la Programación Orientada a Objetos (OOP), una clase es un plano o molde que define un conjunto de atributos y métodos que los objetos creados a partir de esa clase pueden tener. Es como un modelo que describe las características y comportamientos comunes de un grupo de objetos similares.

Un atributo es una variable que pertenece a una clase y representa una característica o propiedad de los objetos de esa clase. Por ejemplo, en una **clase Persona**, los **atributos** podrían ser **nombre, edad y DNI**.

2. Modelado de clases de una compañía aérea

Entidades y Atributos

1. Piloto

- Código (clave primaria)
- Nombre
- Horas de vuelo
- Base

2. Miembro Tripulación

- Código (clave primaria)
- Nombre
- Base

3.Avión

- Código (clave primaria)
- Tipo
- Base

4.Vuelo

- Número de vuelo (clave primaria)
- Origen
- Destino
- Hora
- Fecha
- Código del avión (clave foránea)
- Código del piloto (clave foránea)

5.TripulaciónVuelo

- Número de vuelo (clave foránea)
- Código del miembro de tripulación (clave foránea)

3.Características de una Clase

Una clase en la Programación Orientada a Objetos (OOP) tiene las siguientes características:

Atributos: Variables que representan las propiedades de la clase.

Métodos: Funciones que definen el comportamiento de la clase.

Encapsulamiento: Oculta los detalles internos y expone solo lo necesario.

Herencia: Permite crear nuevas clases basadas en clases existentes.

Polimorfismo: Permite que los objetos de diferentes clases sean tratados como objetos de una clase común.

4. ¿Qué es 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. Es útil para definir capacidades que pueden ser compartidas por diferentes clases.

Diferencia entre Clase e Interfaz

Clase: Define atributos y métodos, y puede tener una implementación completa.

Interfaz: Solo define métodos sin implementación. Las clases que implementan la interfaz deben proporcionar la implementación de esos métodos.

5. ¿Cuáles son las relaciones que existen entre clases? ¿Cuál es el concepto de cada una de esas relaciones?

Asociación: Una relación entre dos clases que permite a una clase conocer y utilizar a otra. Puede ser unidireccional o bidireccional.

Agregación: Un tipo de asociación que representa una relación "tiene un" donde la vida del objeto contenido no depende del objeto contenedor. Ejemplo: una Clase puede tener múltiples Estudiantes.

Composición: Una forma más fuerte de agregación donde la vida del objeto contenido depende del objeto contenedor. Ejemplo: un Coche y sus Ruedas.

Herencia: Relación donde una clase (subclase) hereda atributos y métodos de otra clase (superclase).

Dependencia: Una relación débil donde una clase utiliza temporalmente los métodos o atributos de otra.

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

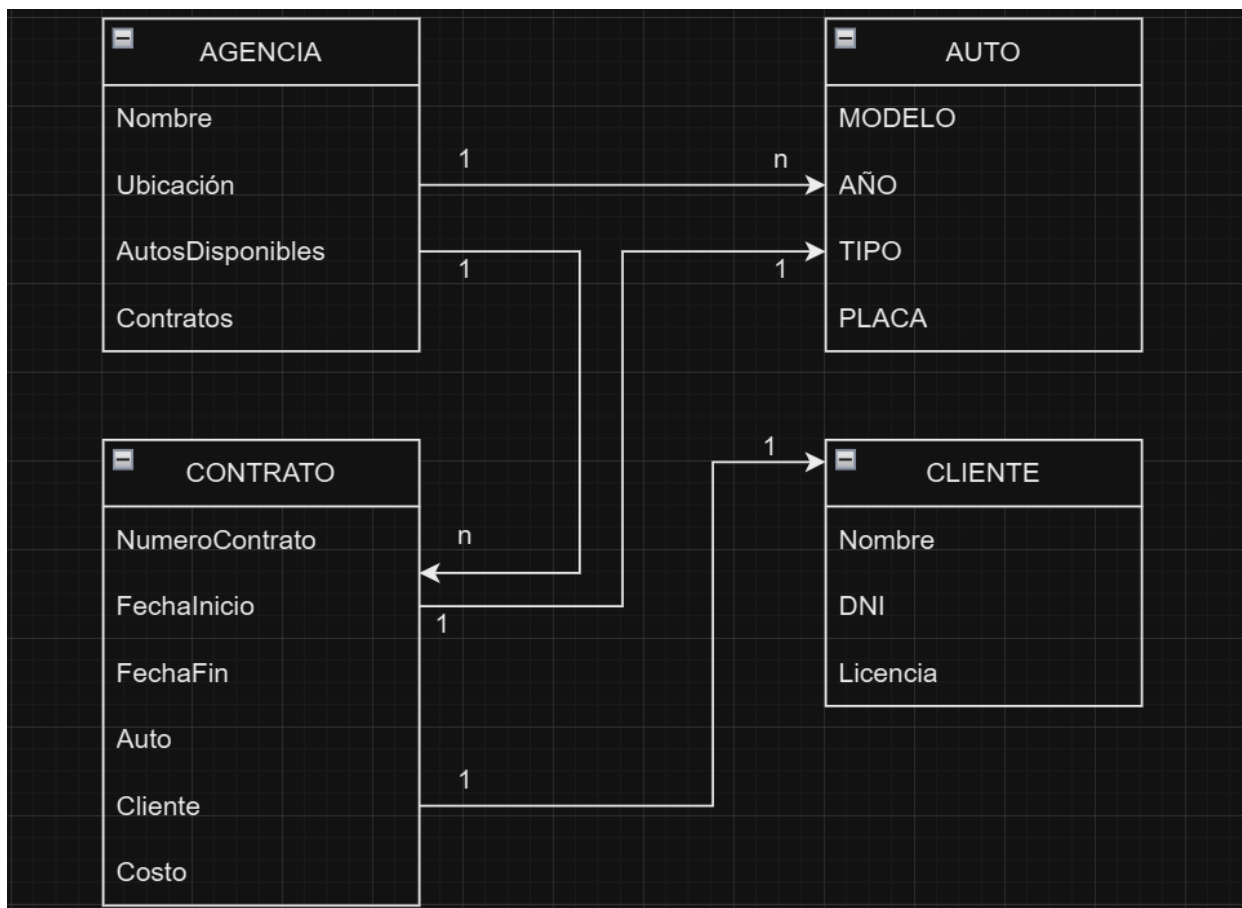
Agregación: Representa una relación "tiene un" donde las partes pueden existir independientemente del todo. Por ejemplo, una Clase puede tener Estudiantes, pero los estudiantes pueden existir sin la clase.

Composición: Representa una relación más fuerte donde las partes no pueden existir independientemente del todo. Por ejemplo, un Coche y sus Ruedas; si el coche es destruido, las ruedas también lo son.

7. ¿Qué es un diagrama de clases?

Un diagrama de clases es una representación visual de las clases dentro de un sistema y las relaciones entre ellas. Es una parte esencial del modelado en OOP y muestra cómo las clases se interrelacionan, incluyendo atributos, métodos y las relaciones de herencia, agregación, composición, y asociación entre ellas.

8. Elabore un diagrama de clases para una agencia de alquiler de autos.



Fuentes

- <https://www.gyata.ai/es/object-oriented-programming/oop-course>
- <https://www.genbeta.com/a-fondo/que-programacion-orientada-a-objetos-principales-conceptos-explicados-para-todos-publicos>
- <https://coderslink.com/talento/blog/que-es-la-programacion-orientada-a-objetos-poo-y-cuales-son-sus-principios-fundamentales/>
- <https://lovtechnology.com/que-es-la-programacion-orientada-a-objetos-poo-como-funciona-y-para-que-sirve/>