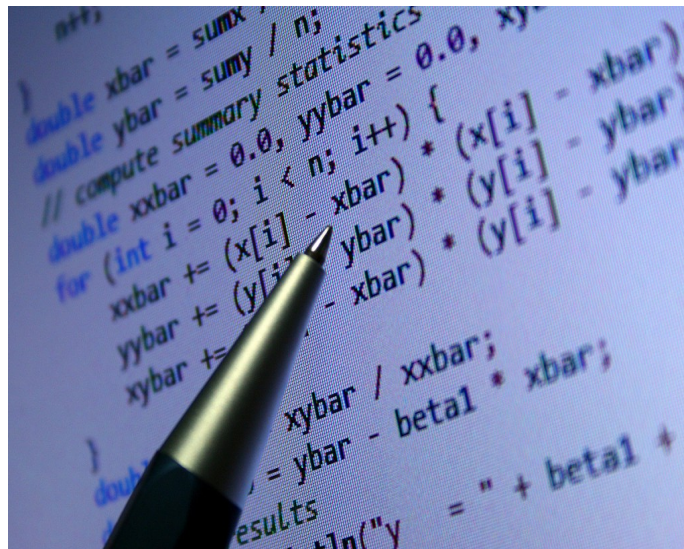


PROGRAMACIÓN ORIENTADA A OBJETOS

TEMA: EJERCICIOS DIAGRAMAS DE CLASES



DOCENTE: Ing. Mónica Bonilla, MsC.
ESTUDIANTE: Ariel Alejandro Calderón

17/11/2024

EJERCICIO 1_

Se desea diseñar un diagrama de clases sobre la información de las reservas de una empresa dedicada al alquiler de automóviles, teniendo en cuenta que:

- Un determinado cliente puede tener en un momento dado hechas varias reservas.
- De cada cliente se desean almacenar su DNI, nombre, dirección y teléfono. Además, dos clientes se diferencian por un código único.
- Cada cliente puede ser avalado por otro cliente de la empresa.
- Una reserva la realiza un único cliente, pero puede involucrar varios coches.
- Es importante registrar la fecha de inicio y final de la reserva, el precio del alquiler de cada uno de los coches, los litros de gasolina en el depósito en el momento de realizar la reserva, el precio total de la reserva y un indicador de si el coche o los coches han sido entregados.
- Todo coche tiene siempre asignado un determinado garaje que no puede cambiar. De cada coche se requiere la matricula, el modelo el color y la marca.
- Cada reserva se realiza en una determinada agencia.

DESCRIPCIÓN DE LAS RELACIONES DEL EJERCICIO

Cliente -> Reserva: Un cliente puede hacer varias reservas (*relación 1 a **), pero una reserva está hecha por un solo cliente (*relación * a 1*).

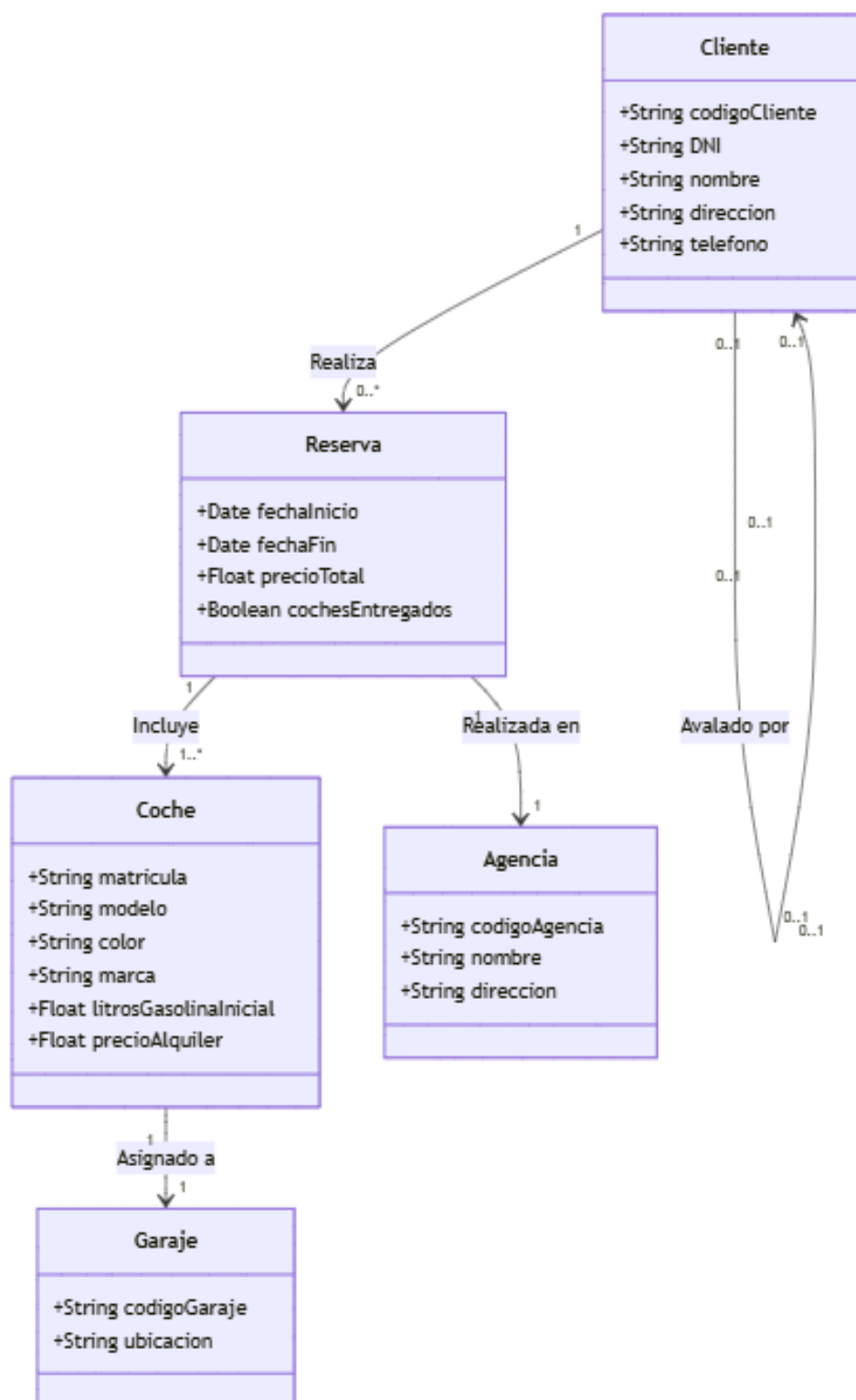
Cliente -> Cliente: Asociación reflexiva, donde un cliente puede ser avalado por otro cliente.

Reserva -> Coche: Una reserva puede incluir varios coches (*relación 1 a **), y un coche puede estar en varias reservas (*relación * a **).

Coche -> Garaje: Cada coche está asignado a un solo garaje (*relación * a 1*), pero un garaje puede contener varios coches (*relación 1 a **).

Reserva -> Agencia: Cada reserva se hace en una única agencia (*relación * a 1*), pero una agencia puede manejar varias reservas (*relación 1 a **).

DIAGRAMA DE CLASES DEL EJERCICIO



EJERCICIO 2

Una cadena de agencias de viajes desea disponer de una Base de Datos que contemple información relativa al hospedaje y vuelos de los turistas que la contratan. Los datos a tener en cuenta son:

- La cadena de agencias está compuesta por un conjunto de sucursales. Cada sucursal viene definida por el código de sucursal, dirección y teléfono.
- La cadena tiene contratados una serie de hoteles de forma exclusiva. Cada hotel estará definido por el código de hotel, nombre, dirección, ciudad, teléfono y número de plazas disponibles.
- De igual forma, la cadena tiene contratados una serie de vuelos regulares de forma exclusiva.
- Cada vuelo viene definido por el número de vuelo, fecha y hora, origen y destino, plazas totales y plazas de clase turista de las que dispone.
- La información que se desea almacenar por cada turista es el código de turista, nombre y apellidos, dirección y teléfono.
- Por otra parte, hay que tener en cuenta la siguiente información:
- A la cadena de agencias le interesa conocer que sucursal ha contratado el turista.
- A la hora de viajar el turista puede elegir cualquiera de los vuelos que ofrece la cadena, y en que clase (turista o primera) desea viajar.
- De igual manera, el turista se puede hospedar en cualquiera de los hoteles que ofrece la cadena, y elegir el régimen de hospedaje (media pensión o pensión completa). Siendo significativa la fecha de llegada y de partida.

DESCRIPCIÓN DE LAS RELACIONES DEL EJERCICIO

CadenaAgencias -> Sucursal: La cadena de agencias posee varias sucursales (relación 1 a *), pero cada sucursal pertenece solo a una cadena (relación * a 1).

CadenaAgencias -> Hotel: La cadena de agencias contrata varios hoteles de forma exclusiva (relación 1 a *), y cada hotel está contratado por una sola cadena (relación * a 1).

CadenaAgencias -> Vuelo: La cadena de agencias contrata varios vuelos de forma exclusiva (relación 1 a *), y cada vuelo está vinculado a una única cadena (relación * a 1).

Sucursal -> Turista: Cada sucursal puede atender a múltiples turistas (relación 1 a *), pero cada turista es atendido por una única sucursal (relación * a 1).

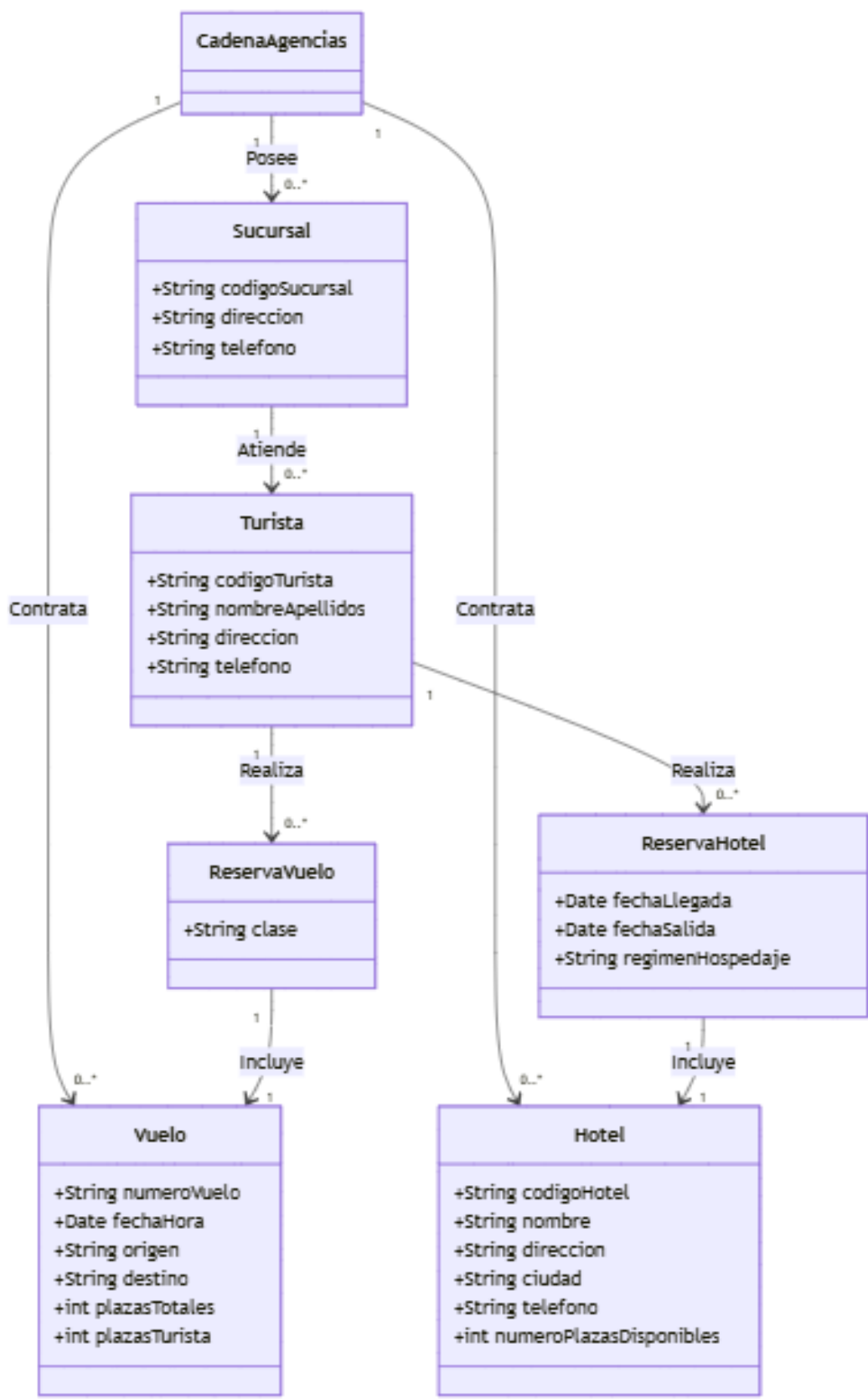
Turista -> ReservaVuelo: Un turista puede realizar varias reservas de vuelo (relación 1 a *), y cada reserva pertenece a un único turista (relación * a 1).

Turista -> ReservaHotel: Un turista puede realizar varias reservas de hotel (relación 1 a *), y cada reserva está asociada a un único turista (relación * a 1).

ReservaVuelo -> Vuelo: Cada reserva de vuelo incluye un único vuelo (relación 1 a 1), mientras que un vuelo puede estar asociado a varias reservas (relación * a 1).

ReservaHotel -> Hotel: Cada reserva de hotel corresponde a un único hotel (relación 1 a 1), mientras que un hotel puede estar relacionado con varias reservas (relación * a 1).

DIAGRAMA DE CLASES DEL EJERCICIO



Bibliografía

1. Libros:

- **Craig Larman.** (2002). *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process* (3rd ed., pp. 105-115). Prentice Hall.

Descripción: Se consultaron los capítulos que detallan el diseño de clases en UML, incluyendo el análisis de relaciones y cardinalidades entre entidades en el modelado de sistemas orientados a objetos.

- **Martin Fowler.** (2003). *UML Distilled: A Brief Guide to the Standard Object Modeling Language* (3rd ed., pp. 45-55). Addison-Wesley Professional.

Descripción: Este libro fue útil para comprender las relaciones y notaciones UML, especialmente en el contexto de sistemas de información complejos como bases de datos de agencias de viajes.

2. Recursos Web:

- **GeeksforGeeks.** (n.d.). Class Diagram in UML. *GeeksforGeeks*. Retrieved October 20, 2024, from <https://www.geeksforgeeks.org/class-diagram-in-uml/>

Descripción: El artículo proporciona una introducción detallada sobre los diagramas de clase en UML, con ejemplos que ilustran asociaciones, agregaciones y composiciones en el contexto de sistemas orientados a objetos.

- **Lucidchart.** (n.d.). Understanding Class Diagrams: A Guide with Examples. *Lucidchart Blog*. Retrieved October 20, 2024, from <https://www.lucidchart.com/blog/uml-class-diagram-tutorial>

Descripción: Recurso que ayudó a clarificar el uso de notaciones UML para representar relaciones de cardinalidad y especificar propiedades de cada clase dentro de un sistema de gestión de agencias.