

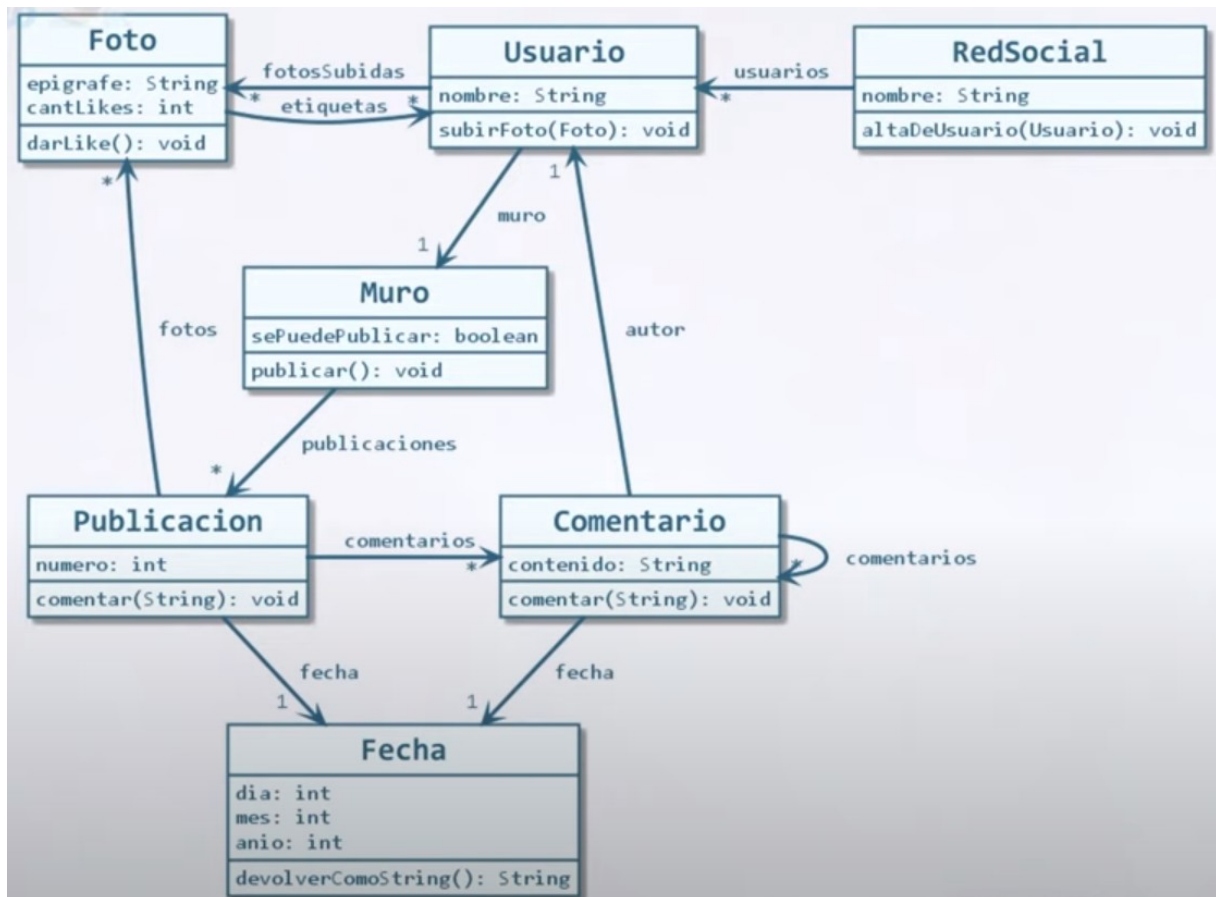
DAM – Entornos de Desarrollo		
PRÁCTICA UT6 - 1		CURSO : 2022-23

Diagramas de Clases UT6 – TAREA1

EJERCICIO 1

Dado el diagrama de clases que representa una posible red social se pide:

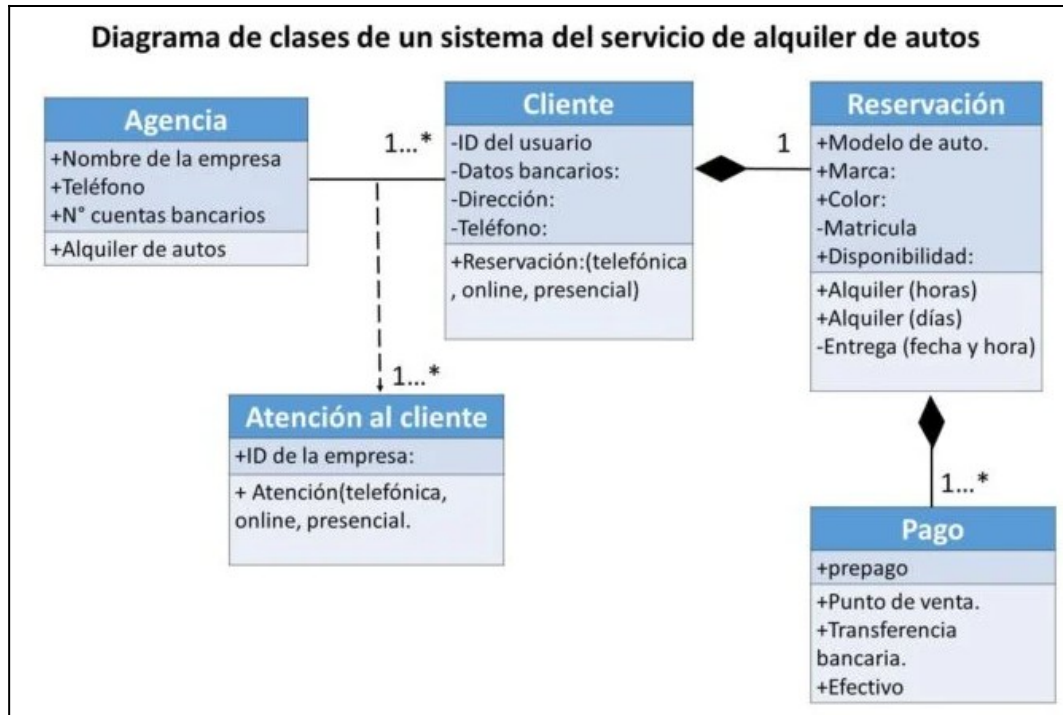
- Explicar con tus propias palabras las distintas relaciones entre clases teniendo en cuenta la multiplicidad entre ellas.
- Escribir en código java las distintas clases, de manera que las relaciones del diagrama queden correctamente implementadas.



EJERCICIO 2

Dado el diagrama de clases que representa una posible red social se pide:

- Explicar con tus propias palabras las distintas relaciones entre clases teniendo en cuenta la multiplicidad entre ellas.
- Escribir en código java las distintas clases, de manera que las relaciones del diagrama queden correctamente implementadas.



EJERCICIO 3

Especificar un diagrama de clases que describa los vuelos que oferta una compañía de viajes según la siguiente especificación:

- La compañía oferta una serie de vuelos para unas fechas concretas y con un número de plazas.
- La compañía dispone de una flota de aviones con una capacidad que da soporte a los vuelos ofertados.
- Las personas compran billetes para los vuelos que le interesan. Para emitir el billete es necesario conocer el nombre, apellidos y edad del pasajero.
- Los billetes identifican el número de asiento que ocupan.

EJERCICIO 4

Representa mediante un diagrama de clases la siguiente especificación sobre una empresa:

- Una aplicación necesita almacenar información sobre empresas, sus empleados y sus clientes.
- Ambos se caracterizan por su nombre y edad.
- Los empleados tienen un sueldo bruto, los empleados que son directivos tienen una categoría, así como un conjunto de empleados subordinados.
- De los clientes además se necesita conocer su teléfono de contacto.
- La aplicación necesita mostrar los datos de empleados y clientes.

EJERCICIO 5

Se desea implementar la lógica de un dispositivo TPV que procesa pagos con tarjetas de crédito.

Las tarjetas de crédito guardan el nombre de la entidad financiera a la que pertenecen (únicamente Visa, MasterCard, American Express), el nombre de la entidad bancaria, el número de tarjeta, el saldo disponible y los datos del titular (DNI, nombre, apellido, teléfono y mail). Cada vez que se cree una nueva tarjeta, deberán indicarse todos estos datos.

A la hora de pagar, el TPV recibiría la tarjeta con la que desea hacerse el pago, junto con el importe que se desea abonar y la cantidad de cuotas (de 1 a 6).

Si el pago es en 1 cuota, no se genera ningún recargo, de lo contrario, el importe se incrementará en un 3% por cada cuota superior a 1. (Ejemplo: Pagar en 4 cuotas representará un 9% de incremento).

El TPV debe chequear que la tarjeta tenga saldo suficiente para poder efectuar el pago junto con el recargo, si hubiese. En caso de éxito, debe generar y retornar (no mostrar) un ticket donde consten los siguientes datos:

- Nombre y apellido del cliente.
- Monto total a pagar.
- Monto de cada cuota.

Si la operación no tuvo éxito, se retornará null.

Puntos a desarrollar:

1) Desarrollar el diagrama de clases UML que modele el enunciado y donde consten las clases con sus atributos, métodos y relaciones (los constructores pueden omitirse).

2) Crear un proyecto en Java que resuelva:

- Desarrollar, en la clase TPV, el método **efectuarPago()**, cuyos parámetros, lógica y valor de retorno deben deducirse según lo enunciado. Desarrollar también los métodos derivados que puedan surgir de él para conseguir el objetivo.
- Al efectuar el pago debe validarse que los datos de entrada sean correctos (importe a pagar > 0, saldo en tarjeta suficiente para realizar el pago....)
- Desarrollar el método **main** del proyecto y generar las instancias necesarias para poder efectuar un pago de 10000 euros en 5 cuotas, usando una tarjeta de crédito con saldo disponible de 15000 euros (el resto de los datos, pueden inventarse a gusto).