

## DIAGRAMAS DE CLASES

1. Estudia el siguiente diagrama. Fíjate en los siguientes aspectos:

a. La visibilidad de los atributos y funciones

Público = +

Privado = -

Protegido = #

Funciones = nombreDeLaFuncion();

b. La multiplicidad de las asociaciones

Hospital - Reception = 1 ---- 1

Reception - Doctor = 1 ---- \*

Reception - Patient = 1 ---- \*

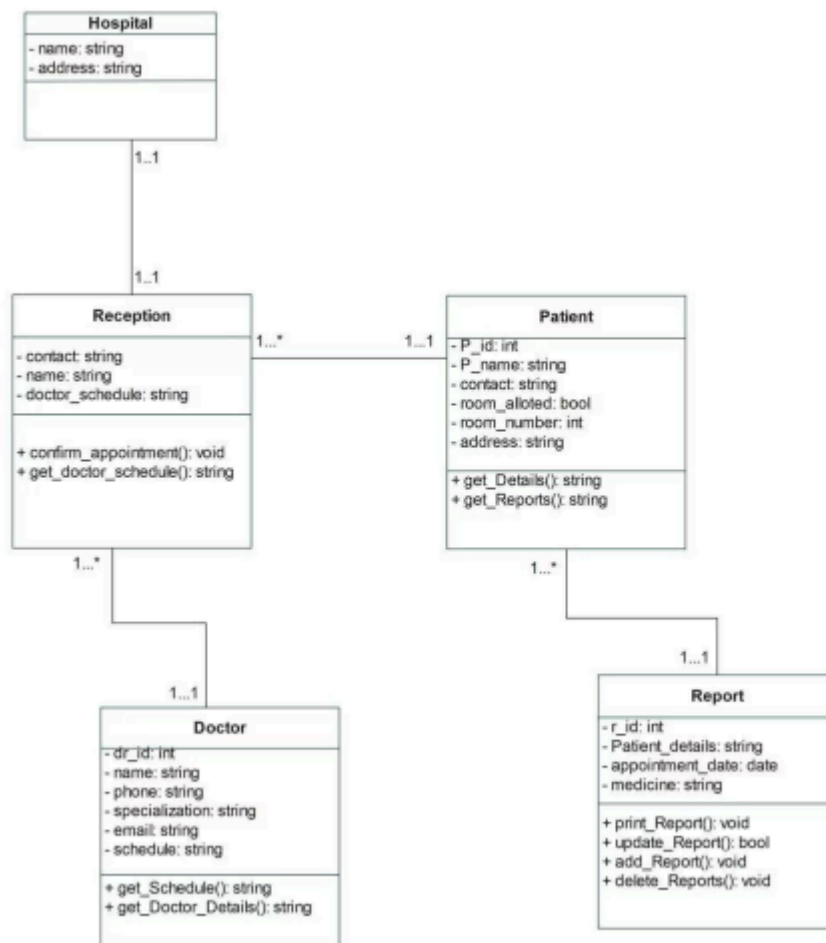
Patient - Report = 1 ---- \*

c. Qué tipo de relaciones hay entre las clases ¿Hay alguna jerarquía?

Son todas asociaciones.

d. ¿Cómo crees que se representarán esas relaciones al convertir el diagrama en código?

Atributo de tipo a otra clase.



2. Ahora vamos con otro diagrama. Fíjate en los siguientes aspectos:

a. La visibilidad de los atributos y funciones

Público = +

Privado = -

Protegido = #

Funciones = nombreDeLaFuncion();

b. La multiplicidad de las asociaciones

Persona - Animal = 1 ---- \*

Animal - Histórico = 1 ---- 1

Histórico - ElementoHistórico = 1 ---- \*

Animal - Diagnóstico = 1 ---- \*

Diagnóstico - Personal = \* ---- 1

Diagnóstico - Factura = 1 ---- 1

Factura - Elemento Factura = 1 ---- \*

c. Qué tipo de relaciones hay entre las clases ¿hay alguna jerarquía?

Hay dos jerarquías:

- Persona ( padre ) // ( hijos ) Física, Jurídica
- Personal ( padre ) // ( hijos ) Veterinario, Auxiliar
- 

d. ¿Cómo crees que se representarán esas relaciones al convertir el diagrama en código?

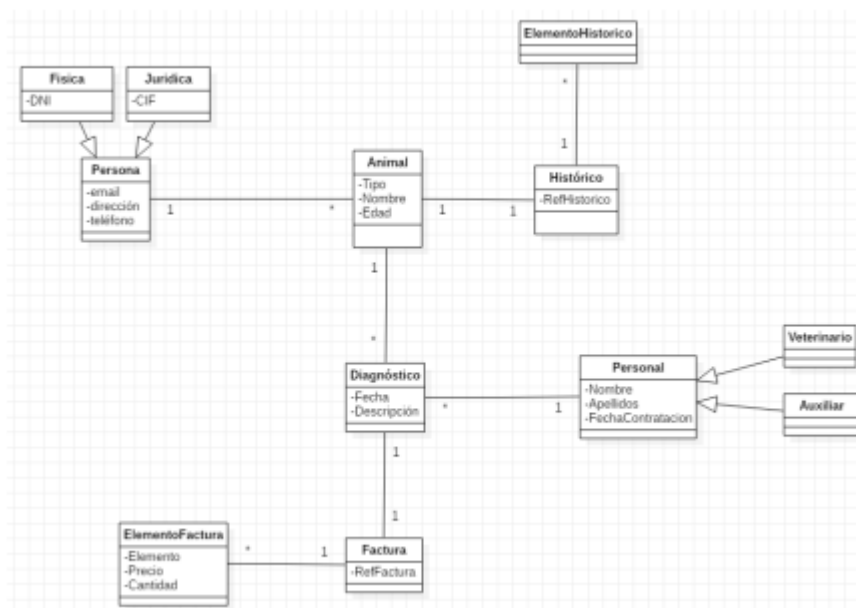
Las jerarquías con 'extends' en java.

Las asociaciones como atributo de tipo a la otra clase.

e. No aparecen los métodos explicitados en las clases. Imagina cuales sería interesante incluir

Pongamos como ejemplo la clase animal:

- + medicarAnimal();
- + intervenirAnimal();
- + examinarAnimal();



3. Dado el siguiente diagrama de clases:

a. Retoca las relaciones entre clases ¿Debería ser alguna una composición o agregación?

Podría haber una composición entre LíneaDeProducto - carritoCompra.

Podría haber una agregación entre pedido - cliente.

b. Reflexiona sobre cuáles serían las multiplicidades en las relaciones en las que tenga sentido ponerlas

LíneaProducto - CarritoCompra = 1 ---- \*

LíneaProducto - Producto = \* ---- \*

LíneaProducto - Pedido = 1 ---- 1

Pedido - Cliente = 1 ---- \*

c. Qué tipo de relaciones hay entre las clases ¿hay alguna jerarquía?

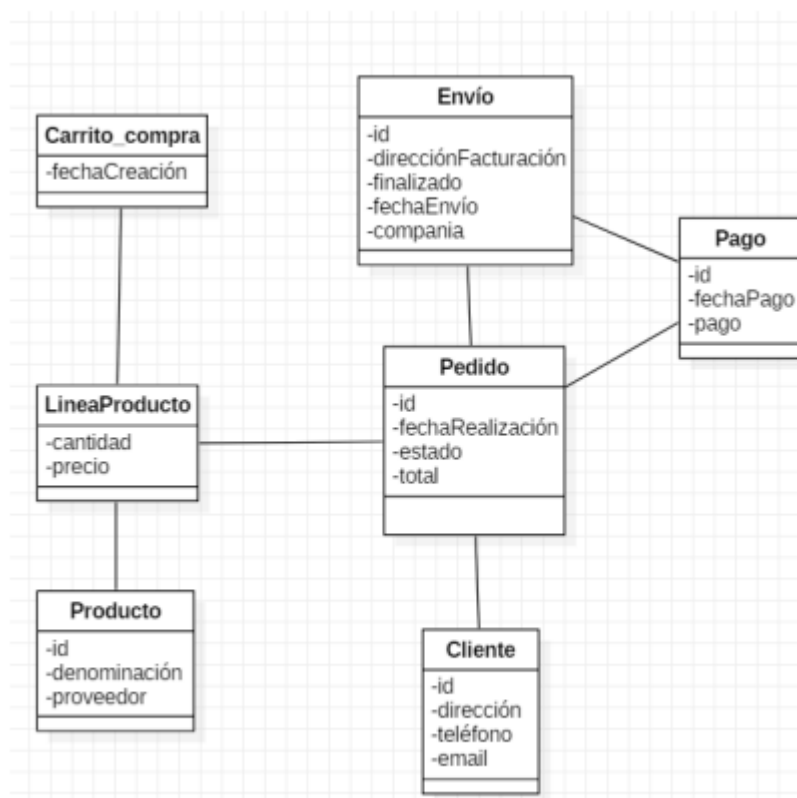
Según se representa en la gráfica son todo asociaciones, con las modificaciones que nosotros hemos aplicado, también podemos encontrar composición y agregación.

d. ¿Cómo crees que se representarían esas relaciones al convertir el diagrama en código?

Las asociaciones como atributo de tipo a la otra clase.

La agregación como objeto pasado desde fuera.

La composición como objeto creado dentro de la clase.



4. Dado el siguiente diagrama de clases:

a. Observa las relaciones de agregación y composición e indica en qué influye en el código posterior que sean de un tipo o de otro

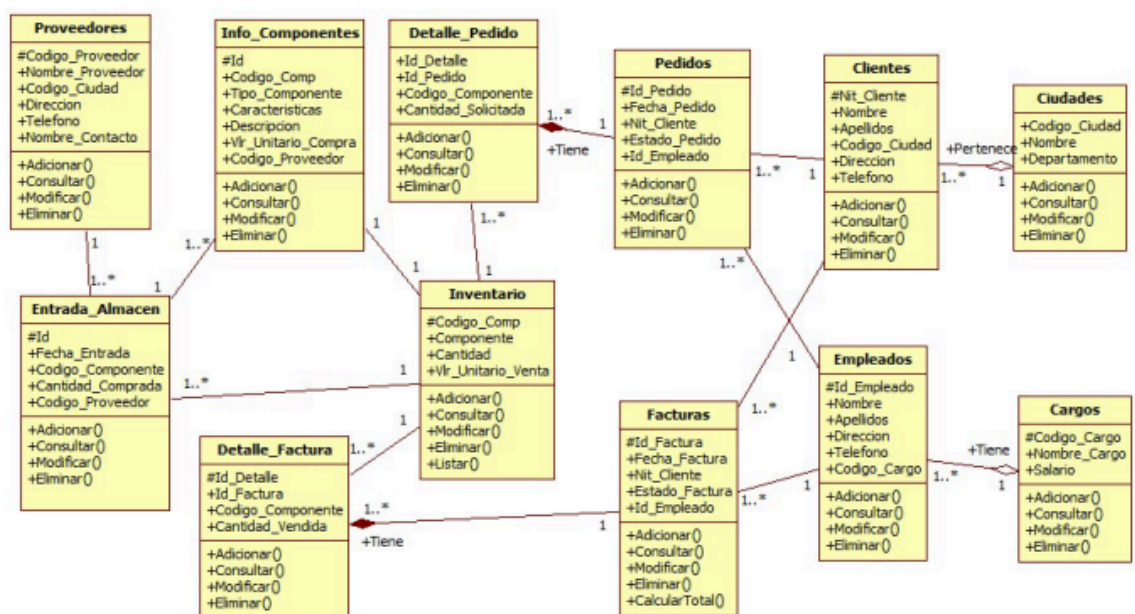
La diferencia principal, es que en la agregación el objeto "parte" puede existir independientemente del objeto "todo", en caso contrario no, la "parte" no puede existir sin el todo.

b. Revisa las multiplicidades y anota en qué clases aparecerán "colecciones" de objetos de otra clase al codificar

Aparecerán las ciudades en clientes y los cargos en empleados.

c. Observa la visibilidad de atributos y métodos ¿Cambiarías algo?

Cambiaría todos los atributos como privados de la propia clase.



- Que docentes y estudiantes optimicen su estructura

Haciendo una jerarquía, ya que hay muchos atributos repetidos entre las clases.

- Que docentes y estudiantes optimicen su estructura

Haciendo una jerarquía, ya que hay muchos atributos repetidos entre las clases.

- Que un docente sea tutor de otro docente en prácticas

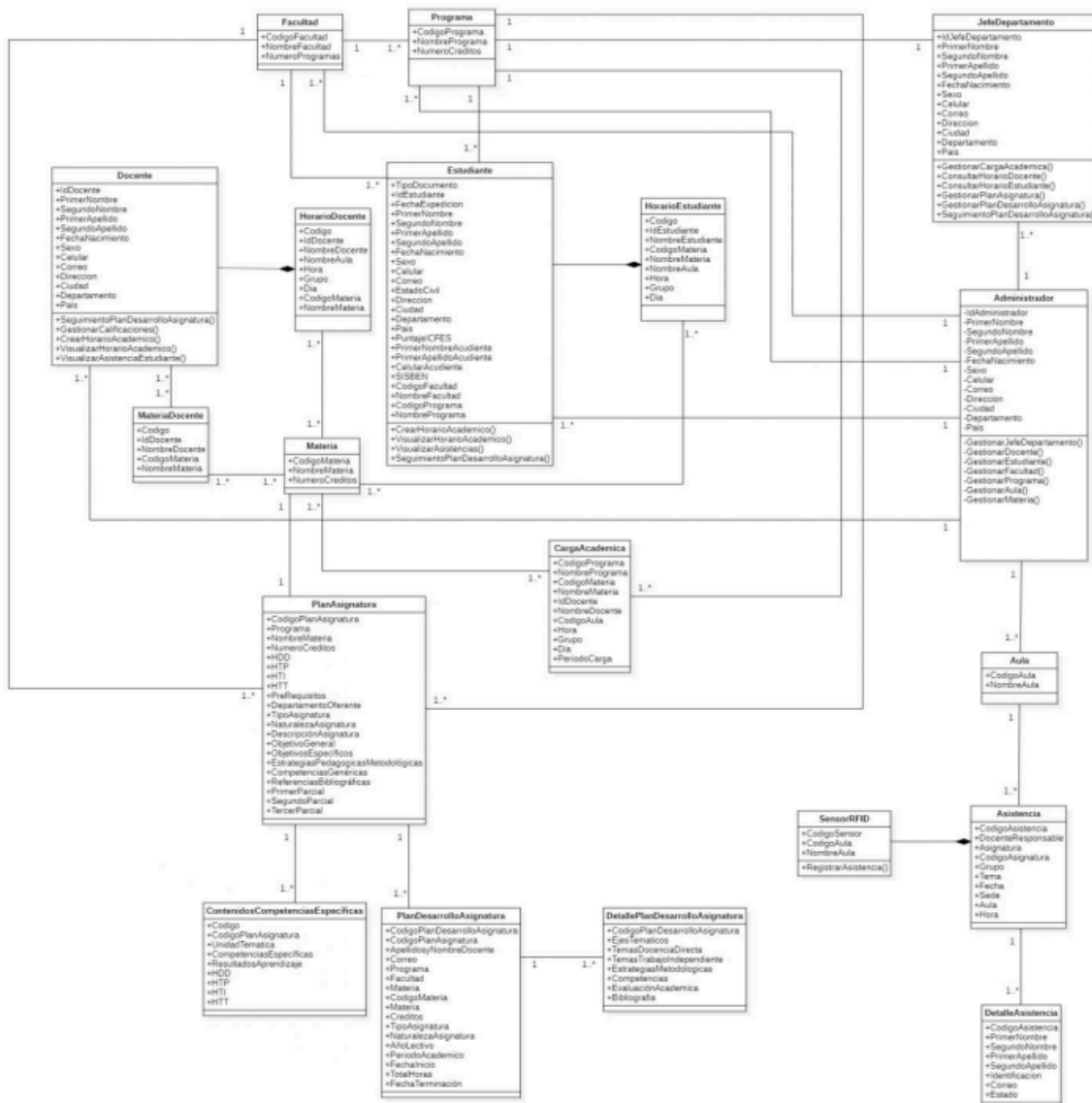
Lo pondremos de forma reflexiva entre los docentes.

- Que una misma asignatura pudiera estar en más de un programa

Cambiando la cardinalidad \* ---- \*.

- Que una materia docente la pueda impartir más de un profesor

Cambiando la cardinalidad \* ---- \*.



6. Representa mediante un diagrama de clases realizado con la siguiente especificación de los datos persistentes de una aplicación:
- Una empresa, se divide en departamentos, cada uno de los cuales está formado por un número de empleados.
  - Los datos a considerar de un empleado son código, nombre, categoría, salario y año de contratación.
  - De cada departamento, nos interesa manejar código, denominación y localidad. También se podrá realizar un recuento de los empleados que forman parte de un departamento
  - Un empleado puede ser jefe de varios empleados y empleado sólo tiene un jefe directo
  - Por otra parte, el empleado puede ser contratado, despedido y se podrá calcular su sueldo cada mes
  - ¿Qué otros métodos deberían añadirse a cada clase, como mínimo?

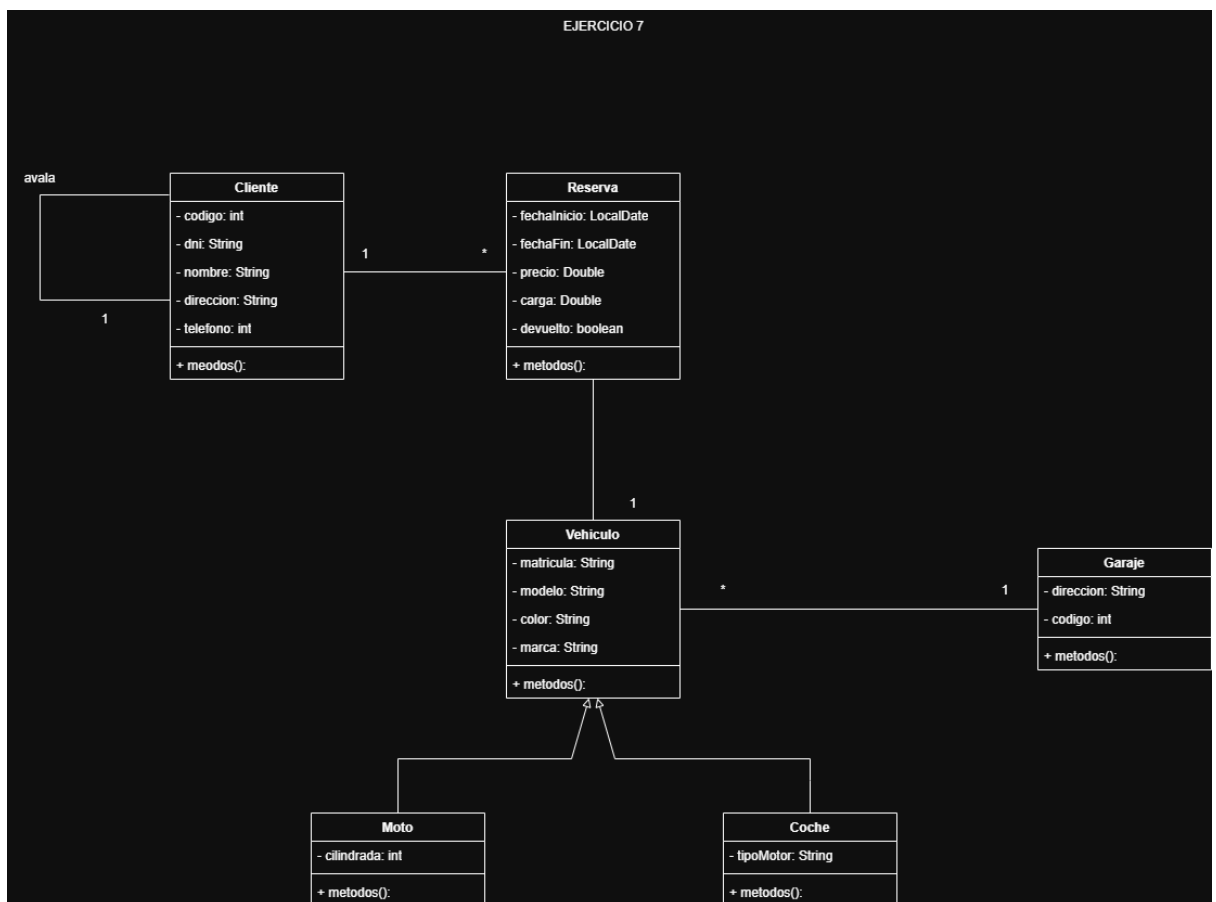




```
1  public class Departamento {
2      private int codigo;
3      private String denominacion;
4      private String localidad;
5      private empleados = new Empleado();
6
7      // .. METODOS .. //
8  }
9
10 public class Empleado {
11     private int codigo;
12     private String nombre;
13     private String categoria;
14     private Double salario;
15     private int anioContratacion;
16     private Departamento departamento;
17     private Empleado jefeEmpleado;
18     private subordinados = new Empleado();
19
20     // .. METODOS .. //
21 }
22
23
```

7. Se desea diseñar un diagrama de clases sobre los datos persistentes de las reservas de una empresa dedicada al alquiler de vehículos, teniendo en cuenta que:

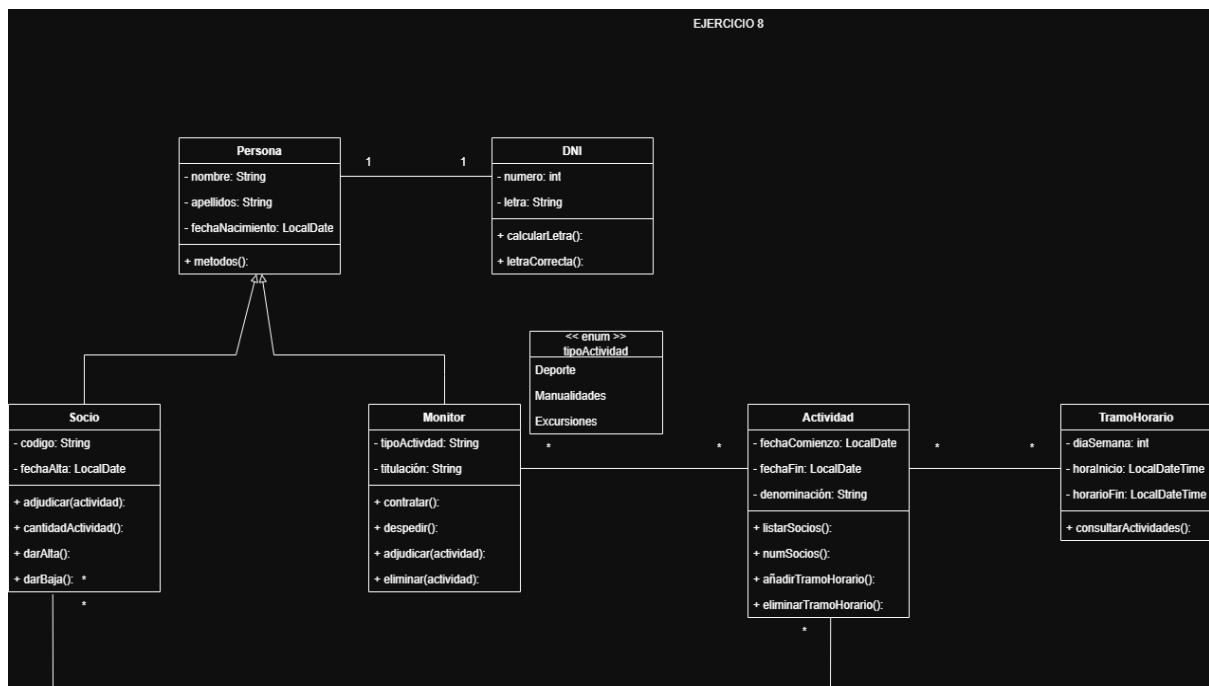
- Un determinado cliente puede tener en un momento dado hechas ninguna, una o varias reservas.
- De cada cliente se desean almacenar su DNI, nombre, dirección y teléfono. Además, dos clientes se diferencian por un código único.
- Cada cliente puede ser avalado por otro cliente de la empresa. De ser así, nos interesa saber quién avala a quien
- Una reserva la realiza un único cliente y siempre involucra a un sólo vehículo.
- Es importante registrar la fecha de inicio y final de la reserva, el precio del alquiler del vehículo (que puede ser distinto según fechas y otros criterios: fidelidad, ofertas...), los litros de gasolina en el depósito en el momento de realizar la reserva, y un indicador de si ha sido devuelto.
- Todo vehículo tiene siempre asignado un determinado garaje del que se saca y al que se devuelve. Del garaje necesitamos conocer dirección (calle, número, población y código postal). En un garaje puede estar estacionado más de un vehículo.
- De cada vehículo, se requiere registrar su matrícula, modelo, color y marca. Si es una moto, además la cilindrada, y si es un coche, si es de gasolina, gasoil, eléctrico o híbrido
- De momento, no detallaremos los métodos relevantes en cada clase, y sólo tendremos en cuenta sus atributos.





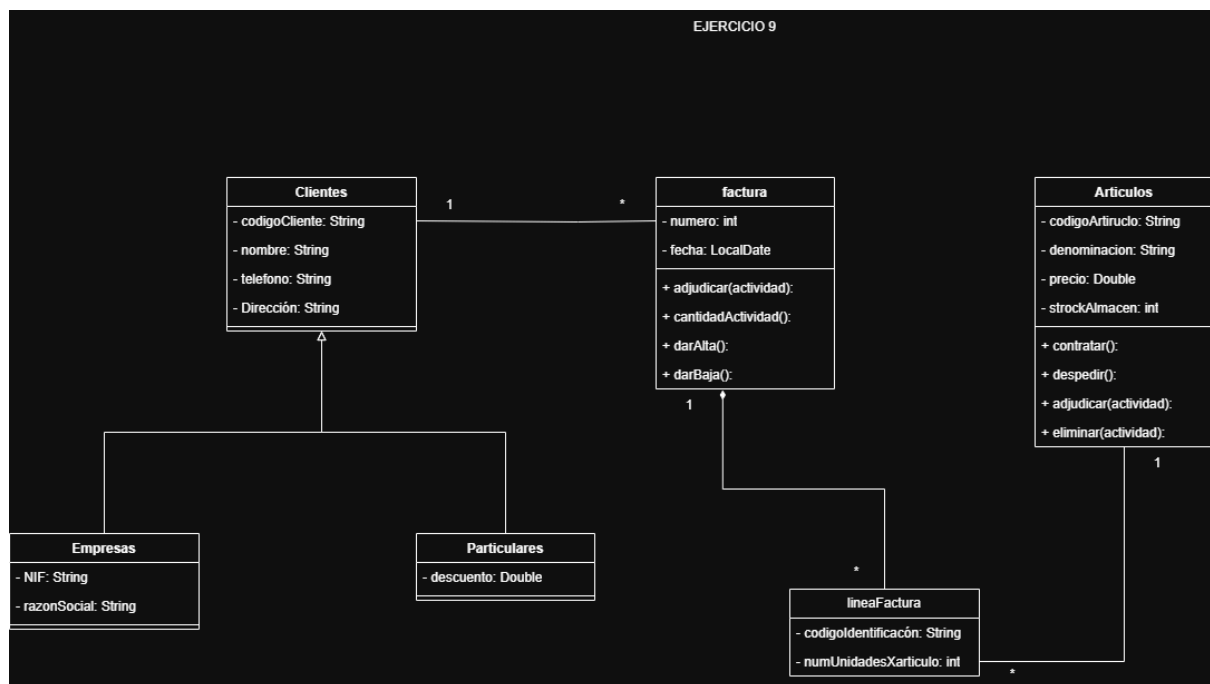
8. Diseña un diagrama de clases que modele solo las clases persistentes relativas al proceso de gestión de las personas que se apuntan a una asociación, en la cual podrán realizar distintos tipos de actividades lúdicas y deportivas. Utiliza el entorno que desees:

- El sistema deberá tener en cuenta a todas las personas relacionadas con la asociación, ya sean socios o monitores
- De cada persona interesa conocer sus datos básicos: NIF, nombre completo y fecha de nacimiento
- Un socio puede darse de alta o de baja y apuntarse a una o más actividades. Cuando cada nuevo socio se da de alta, se le asigna un código de asociado, alfanumérico y se anota la fecha de alta
- Queremos también poder computar a cuantas actividades está apuntado un socio, y listarlas, y así mismo, dada una actividad, saber cuántos socios están apuntados y listarlos
- De un monitor, interesa conocer el tipo genérico de actividad que imparte (deporte, manualidades, excursiones) y la titulación que tiene para hacerlo. Un monitor puede ser contratado, despedido, adjudicado a una actividad, o eliminado de ésta
- La fecha tiene tres campos (día, mes y año) de tipo entero. Queremos poder calcular los meses transcurridos entre dos fechas, y en que cae una fecha a partir del momento actual y un número de meses
- El NIF se modela como una clase con un campo de tipo entero llamado DNI y un campo de tipo carácter llamado letra. Queremos poder calcular la letra a partir del DNI y comprobar si una combinación letra-DNI es correcta
- Una actividad tiene una denominación, una fecha de comienzo, una de fin, y uno o varios tramos horarios. Queremos poder añadir y eliminar tramos a una actividad
- El tramo horario incluye el día de la semana (de 1 a 7) y la hora de comienzo y de fin. Es interesante saber qué actividades se ofertan en un tramo dado
- Una actividad además es impartida por uno o varios monitores y también un monitor podrá impartir uno o más talleres o actividades diferentes



9. Se desea realizar el análisis de un sistema de gestión informática de un pequeño almacén dedicado a vender artículos a clientes. Para ello se necesita manejar los datos de clientes, artículos, facturas y detalle de las facturas:

- Los datos de los clientes son: código de cliente, nombre, teléfono y dirección
- Los datos de los artículos son: código de artículo, denominación, pvp y stock de almacén
- Los datos de las facturas son: número de factura y fecha de la factura
  - Hay que tener en cuenta que un cliente puede tener muchas facturas o ninguna, y que cada factura está compuesta de una o varias líneas de factura
  - Cada línea de factura tiene un código que las identifica, así como un código del artículo que se factura y el número de unidades de este
- Existen dos tipos de clientes: empresas y particulares
- De los clientes empresa se necesita saber el CIF y la razón social
- De los particulares, el porcentaje de descuento



11. Se desea realizar el análisis de un sistema de gestión informática de una pequeña agencia que oferta viajes a sus clientes

- o El sistema debe proporcionar una ventana inicial de login que dará paso a otra ventana con una serie de menús que conectará con el resto de ventanas de la aplicación, permitiendo realizar las siguientes acciones:
- o La gestión de reservas de viajes para realizar reservas, modificar reservas, consultar reservas, borrar reservas y generar e imprimir facturas
- o El mantenimiento de datos de clientes, para mantener actualizados los datos se realizarán operaciones de consultas, altas, bajas y modificaciones de datos de clientes, y además debe permitir generar listados de clientes
- o Mantenimiento de datos de viajes, para mantener actualizados los datos se realizarán operaciones de consultas, altas, bajas, modificaciones e informes de viajes. Disponemos de una base de datos donde están almacenados los datos de los clientes, los viajes, las reservas, las fechas de viaje. Los datos son los siguientes:
  - Datos de clientes: código de cliente, nombre, teléfono y dirección
  - Datos de viajes: código, nombre, plazas y precio
  - Datos de las reservas: número de reserva y estado de la reserva
  - Un cliente puede realizar muchas reservas y una reserva es de un solo cliente
  - Igualmente, de un viaje se pueden realizar muchas reservas, y una reserva pertenecerá a un viaje
  - Los viajes se ofertan en varias fechas de viaje. De estas se necesita saber la fecha de comienzo y de fin. Varios viajes pueden compartir las mismas fechas.
  - También se cuenta con la información de un catálogo de viajes, los datos del catálogo son; código, destino, procedencia, temporada y precio.
  - Los viajes se crean a partir del catálogo

En relación con la lógica de la aplicación, necesitaremos diseñar clases de control que realicen las operaciones CRUD directas de cada una de las clases correspondientes a cada entidad

## Ejercicio 11

