Curso: Informática Industrial

Mohamed Abderrahim

Álvaro Castro González

José Carlos Castillo Montoya

Ejercicios modelado de software

Ejercicio 1

Se quiere hacer el diseño de un robot modular. El robot estará compuesto por varios módulos entre los que se encuentran: rotación, extensión, helicoidal, cámara. Los módulos podrán ser dinámicos (capaces de moverse: rotación, extensión, helicoidal) o estáticos (no se pueden mover: cámara).

Los módulos tendrán un identificador (1-255) y unas dimensiones (largo, ancho y alto, entre 1 y 200mm). Los módulos estarán compuestos de un sistema de control y un sistema de comunicación. Los módulos dinámicos tendrán:

• motores (1 ó 2).

• un parámetro que es el tipo de movimiento que pueden realizar.

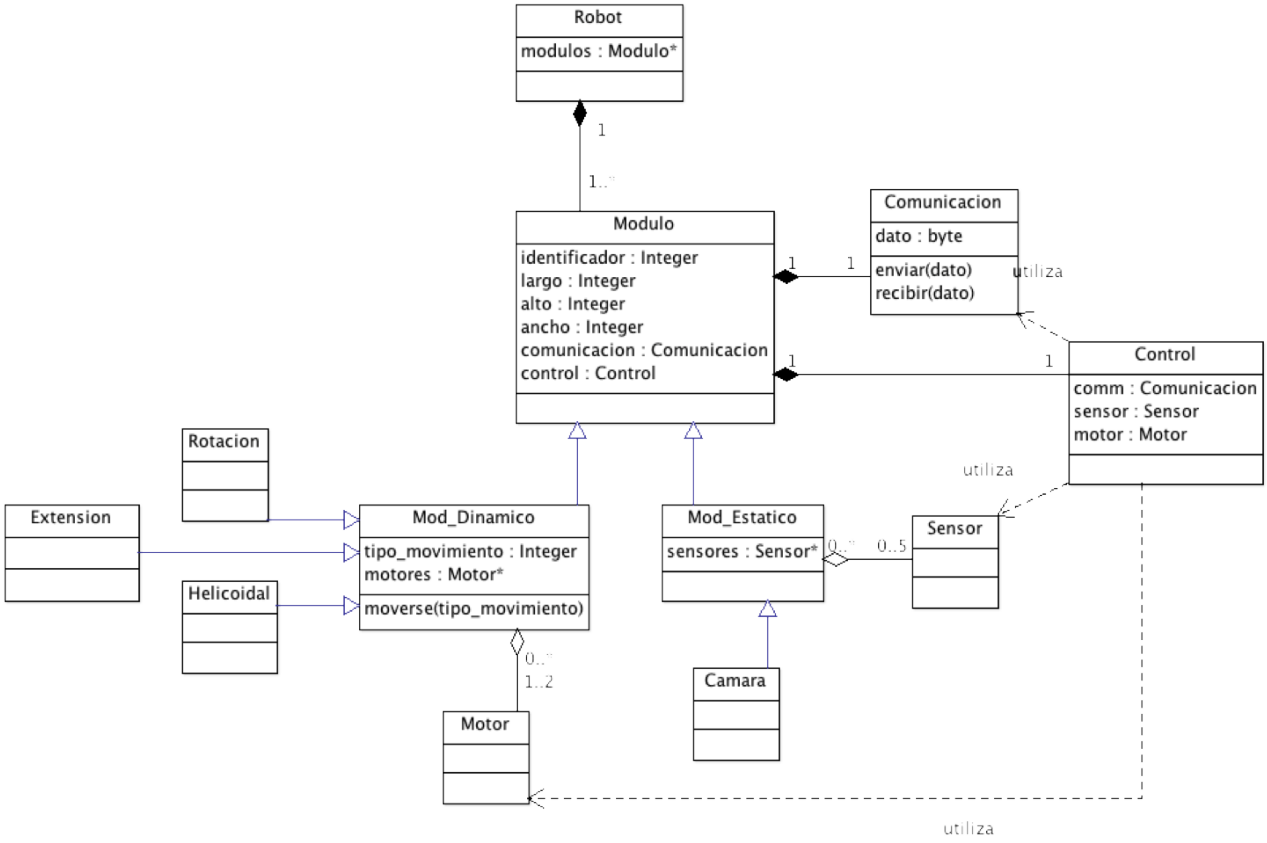
• una función que es moverse (con parámetro el tipo de movimiento).

Los módulos estáticos podrán tener sensores (de 0 a 5).

El sistema de control utiliza el sistema de mensajes para comunicarse. Los módulos pueden enviar y recibir mensajes de/hacia el usuario y otros módulos, con un parámetro que es un array de datos a mandar o recibir. También utiliza los motores para moverse y los sensores para captar información del medio.

Se pide que utilizando herencia siempre que se pueda, se realice un diseño UML de las clases necesarias para representar todas las entidades del sistema, indicando atributos y métodos, así como las relaciones existentes entre las clases.

Solución propuesta:



Ejercicio 2

Especificar un diagrama de clases que describa redes de ordenadores. Los elementos que se pueden incluir en la red son:

- Servidor, PC, Impresora.
- Hub, Cable de red.

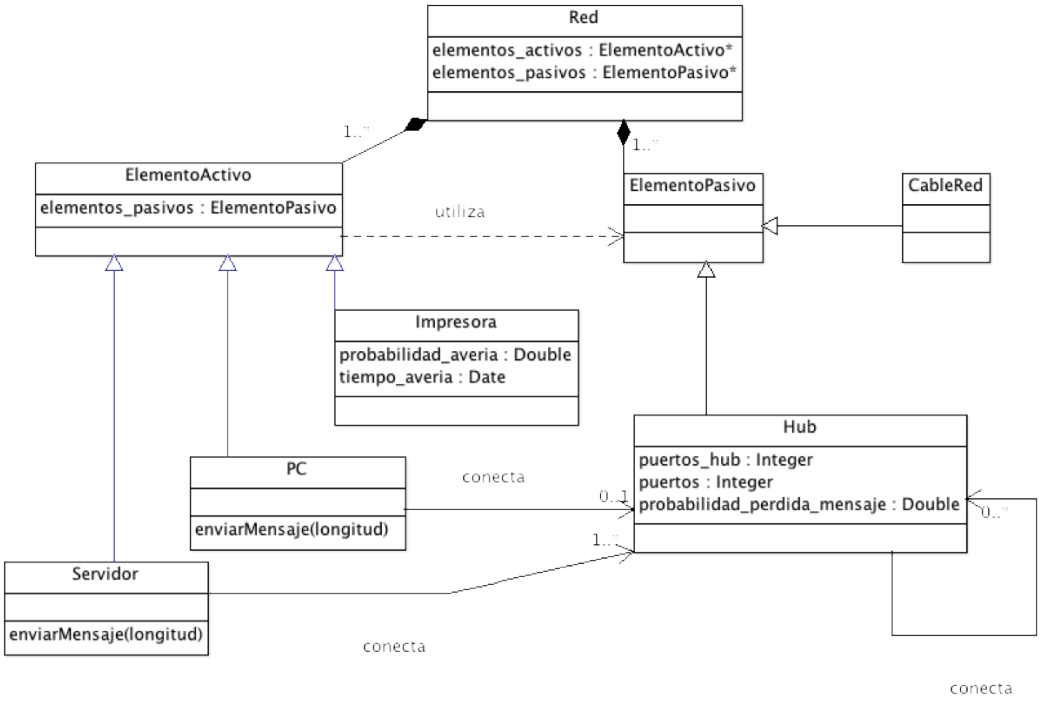
Los PCs pueden conectarse con un único Hub, los servidores con uno o varios.

Los Servidores y PCs pueden generar mensajes, con una cierta longitud.

Los Hubs tienen un número de puertos, algunos de los cuales puede usarse para conectar con otros Hubs. Tienen cierta probabilidad de “perder” mensajes.

Las impresoras pueden averiarse, con cierta probabilidad, durante cierto tiempo.

Solución propuesta:



Ejercicio 3

Una biblioteca tiene copias de libros. Estos últimos se caracterizan por su nombre, tipo (ingeniería, literatura, informática, historia ...), editorial, año y autor.

Los autores se caracterizan por su nombre, nacionalidad y fecha de nacimiento.

Cada copia tiene un identificador, y puede estar en la biblioteca, prestada, con retraso o en reparación.

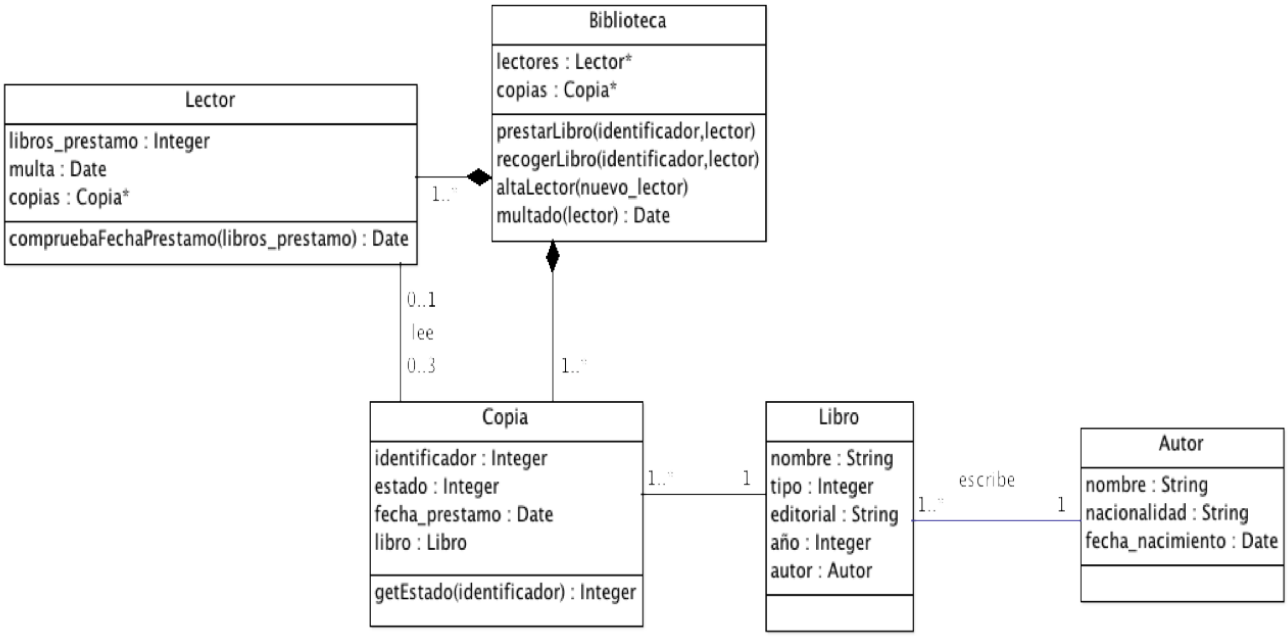
Los lectores pueden tener un máximo de 3 libros en préstamo.

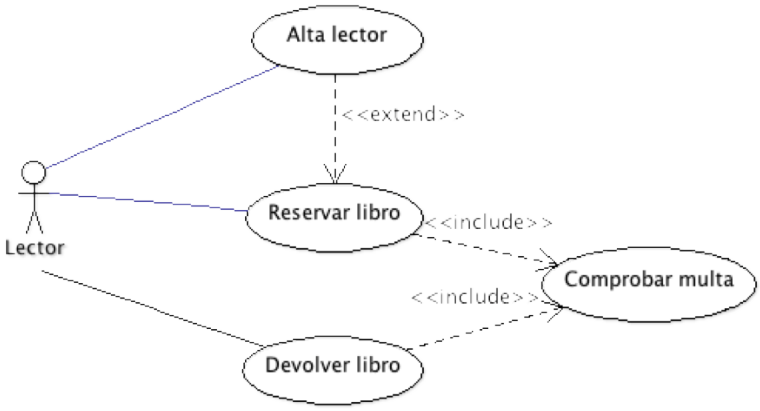
Cada libro se presta un máximo de 30 días, y por cada día de retraso, se impone una "multa" de dos días sin posibilidad de coger un nuevo libro.

Realiza un diagrama de clases y añade los métodos necesarios para realizar el préstamo y devolución de libros.

Realiza un diagrama de casos de usos.

Solución propuesta:





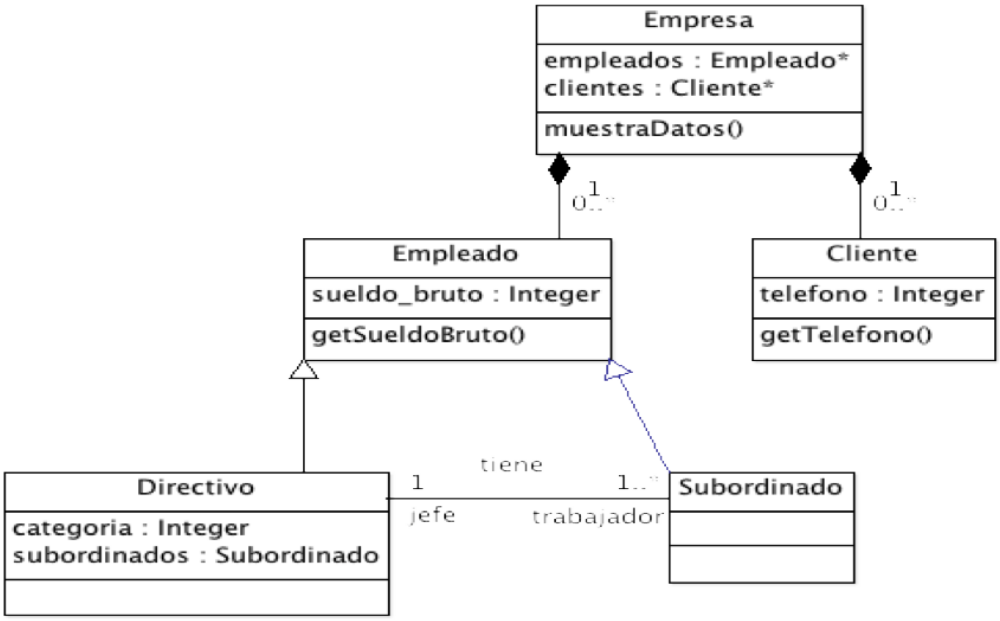
Ejercicio 4

Representa mediante un diagrama de clases la siguiente especificación.

Una aplicación necesita almacenar información sobre empresas, sus empleados y sus clientes. Ambos (empleados y sus clientes) 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.

Solución propuesta:



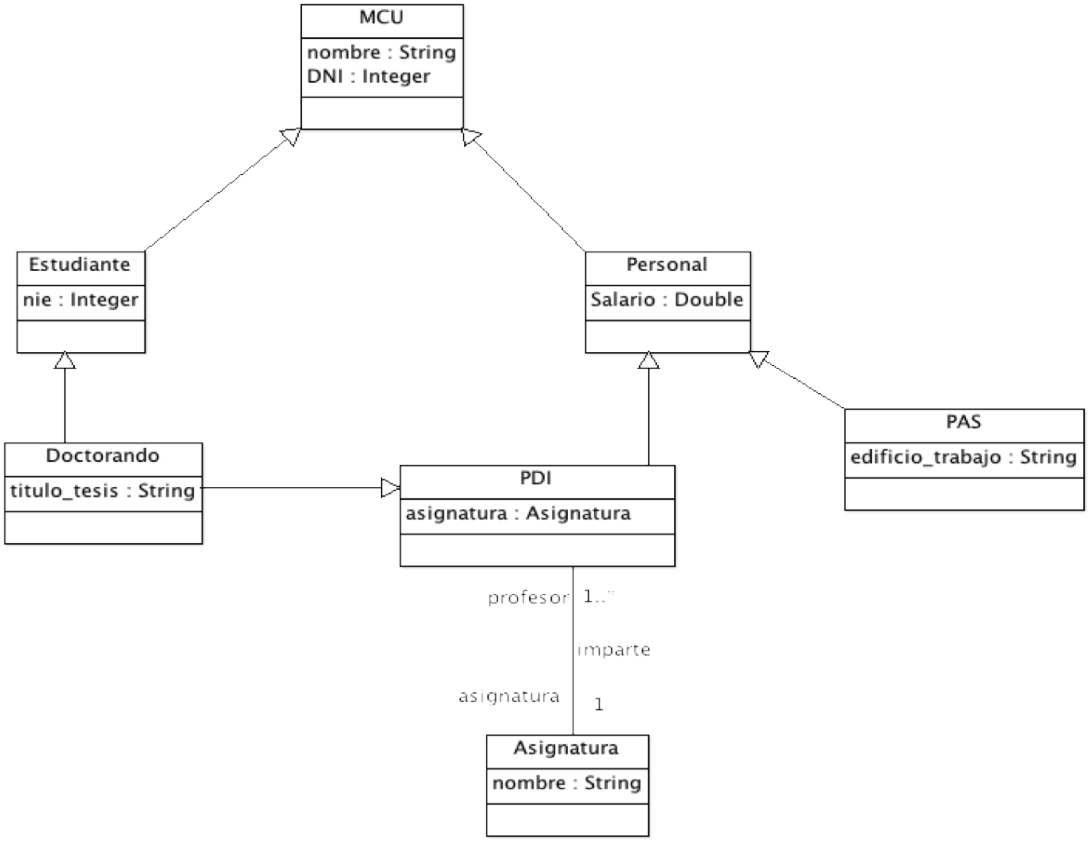
Ejercicio 5

Se propone realizar un modelo simplificado de los distintos miembros de la comunidad universitaria. Todos los miembros de la comunidad universitaria se caracterizan por un nombre y un D.N.I. Los miembros se dividen en estudiantes o personal de la universidad. Todos los estudiantes tienen un número de identificación asociado: el nie.

En cuanto al personal, todos tienen un salario asignado y a su vez estos pueden ser personal docente investigador (pdi) ó personal de administración y servicios (pas). Los pdi tienen asignada una asignatura que impartir (se identificará por el título) y los pas un edificio donde trabajan (se identificará por el nombre del edificio). Además de los anteriores, existen los doctorandos que son a la vez pdi y estudiantes. Los doctorandos se caracterizan por el título de la tesis doctoral sobre la que investigan.

Se pide que, utilizando herencia siempre que se pueda, se realice un diseño UML de las clases Miembro, Personal, Estudiante, Pdi, Pas y Doctorando y se implementen en código C++ ajustándose a la descripción dada anteriormente.

Solución propuesta:



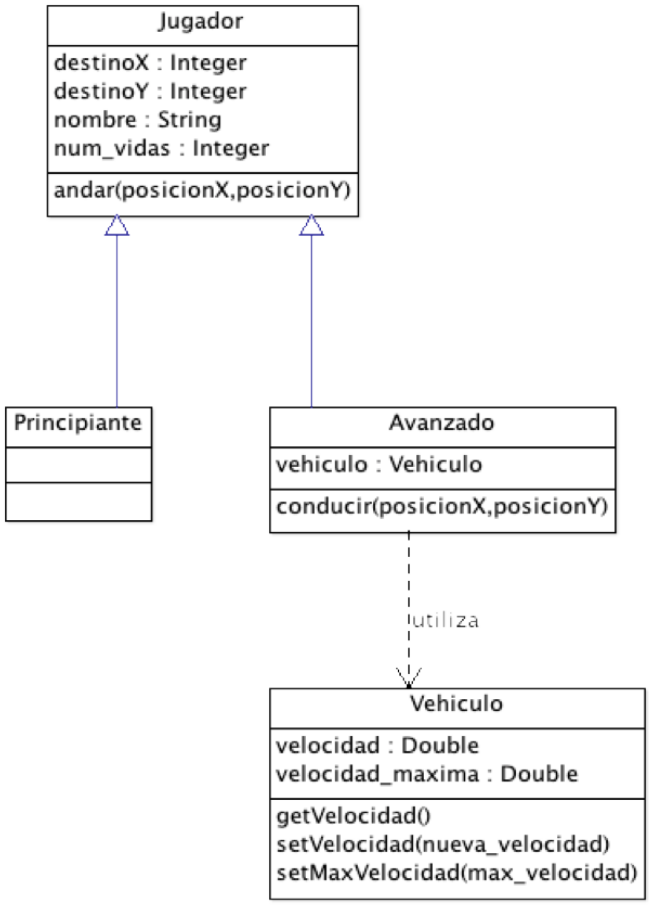
Ejercicio 6

En un juego de ordenador existen 2 tipos de jugadores: los principiantes y los avanzados. Todos ellos deben tener un nombre y un número de vidas. Los principiantes se desplazan andando a unas coordenadas (x,y). Los jugadores avanzados además de andar también pueden conducir un vehículo para desplazarse más rápido a unas coordenadas. Cada vehículo tienen asociada una velocidad que puede ser leída y ajustada a un valor dado pero no puede superar una velocidad máxima dada. La velocidad máxima sólo se podrá asignar una vez y no podrá ser modificada.

Todos los atributos de las clases serán privados y tendrán métodos públicos para acceder a ellos (get/set) salvo que los requisitos indiquen lo contrario. Debe existir un método que se llame andar y otro conducir.

Crear el diagrama UML de clases para el problema propuesto y que incluya todas las funcionalidades descritas.

Solución propuesta:



Ejercicio 7

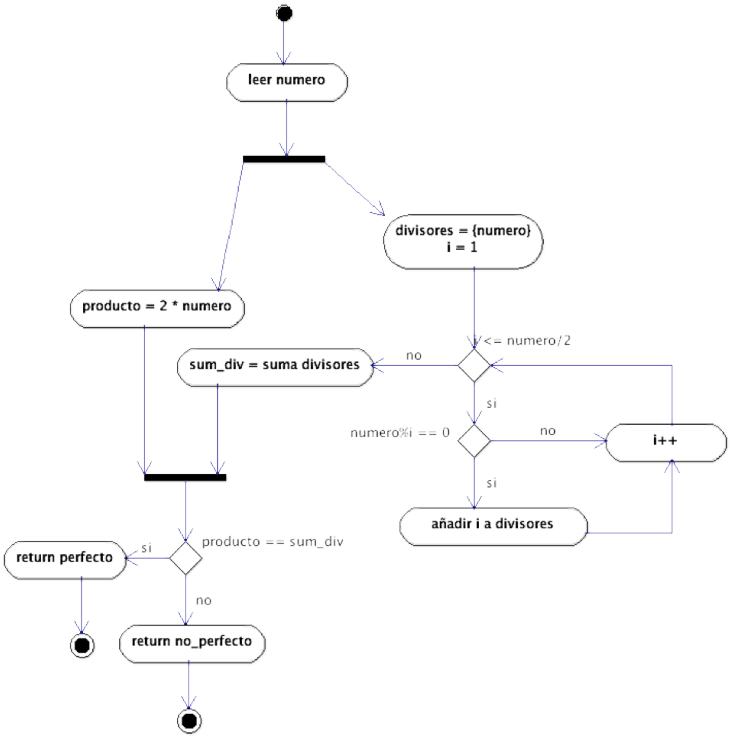
Sea n ∈ Z+. Decimos que n es un entero perfecto si 2n es igual a la suma de todos sus divisores positivos incluido el mismo. Por ejemplo, el número 6 es un número perfecto ya que

2 * 6 = 1 + 2 + 3 + 6 = 12

Al igual que 28 y 496.

Realice el diagrama de flujo y la implementación en código C, para determinar si un número entero es perfecto o no.

Solución propuesta:

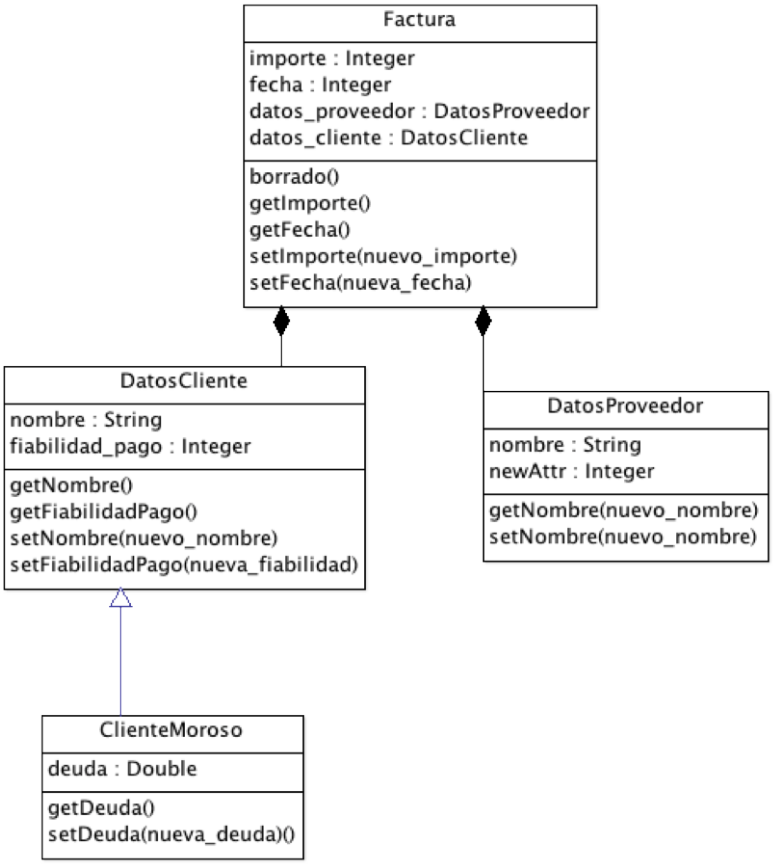


Ejercicio 8

En un programa de ordenador, las facturas tienen necesariamente un conjunto de datos del proveedor, un conjunto de datos del cliente, un importe (valor decimal) y una fecha (vector de enteros). Los datos del cliente son la cadena de caracteres nombre y el entero fiabilidad de pago, mientras que los datos del proveedor son sólo su nombre. Dentro de la categoría cliente está el subtipo “cliente moroso”, que lleva también asociado el número decimal deuda. Cada clase tendrá los métodos para leer y fijar (“set” y “get”) todos sus atributos (no hace falta incluir en la clase Factura los métodos para fijar datos del cliente o del proveedor). También se debe incluir en la clase Factura un método de “Borrado”, que no devuelve ni recibe ningún parámetro. Los atributos serán privados y tendrán métodos públicos para acceder a ellos.

Se pide dibujar el diagrama UML de las clases Factura, Datos_del_cliente y Datos_del_proveedor.

Solución propuesta:



Ejercicio 9

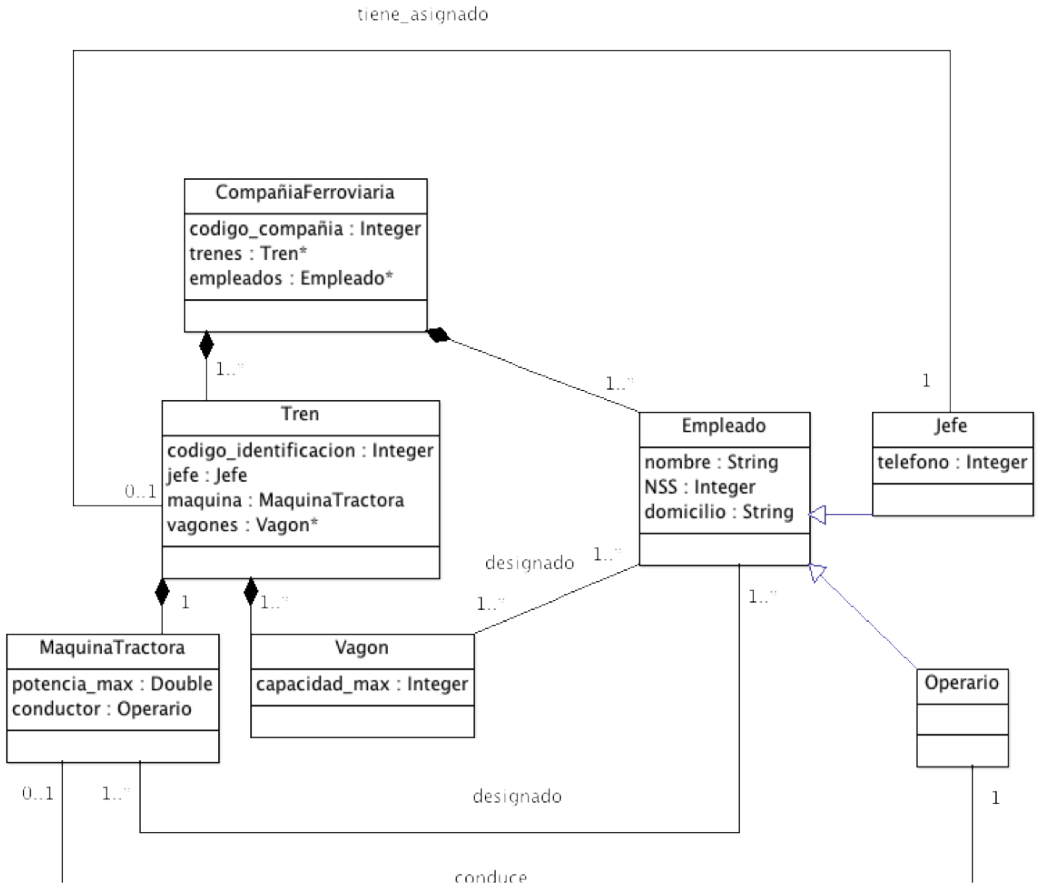
Se ha de modelar parte de la funcionalidad requerida para un subsistema de gestión de trenes de compañías ferroviarias. Los requisitos de almacenamiento de información que se necesitan están definidos (con lenguaje natural) como sigue.

En primer lugar, toda compañía (con su denominación) a considerar posee al menos un tren. Cada tren está compuesto por una máquina tractora y al menos un vagón. Pueden existir vagones y máquinas no asignados a tren alguno. Cada tren tiene un código identificador propio de su compañía, los vagones una capacidad máxima, y las máquinas tractoras una potencia máxima.

Una compañía tiene al menos un empleado, del que se almacenan sus principales datos, como son el nombre, el número de la seguridad social y el domicilio. Según su trabajo, estos pueden ser jefes u operarios. Si es jefe, se almacena su número de teléfono. Cada empleado puede tener designados un conjunto de máquinas tractoras y/o vagones. A su vez, cada máquina tractora o vagón podrá estar asignado a un conjunto de empleados. Eso sí, cada tren tiene siempre asignado su jefe, y cada máquina tiene un operario que la conduce.

Realizar en UML el diagrama de clases que recoge la relación existente entre trenes, empleados, teniendo en cuanta que todos los atributos son privados. Las clases deben contemplar el acceso (get) y el cambio (set) para todos los atributos menos en el caso de la potencia de la máquina. Especificar relaciones y multiplicidades.

Solución propuesta:



Ejercicio 10

Se desea realizar el modelo de un sistema de gestión bibliográfica. Para ello se consideran los siguientes conceptos:

• Hay "Bases_de_datos" que tienen un atributo que es su "url", y un método "anadir_publicación" (que devuelve un puntero a "Publicación" y tiene un parámetro que es un string).

• Las bases de datos se componen de "Publicaciones".

• Una publicación tiene dos atributos: "nombre" (de tipo string) y "fecha" (de tipo Date – suponerla ya implementada).

• Hay varios tipos de publicaciones: "Articulo_de_conferencia", "Capitulo_de_libro" y "Libro". A su vez hay un tipo de libro que son los "Proceedings", que son libros compuestos por varios artículos de una misma conferencia. Los artículos de conferencia tienen un atributo: "num_paginas" (int). Los capítulos de libro tienen un atributo: "num_capitulo" (int). Los libros tienen un atributos: "ISBN" (array de 4 enteros). Los proceedings tienen un atributo: "lugar_conferencia" (string).

• Los libros pueden contener capítulos de libro.

• Los proceedings están compuestos por artículos de conferencia.

• También hay "Autores", que tienen tres atributos: "nombre" (string), "correo_electronico" (string) y "entidad" (string), y un método: "añadir_entidad" (no devuelve nada, y tiene un parámetro de tipo string).

• Los autores pueden escribir artículos y consultar bases de datos.

Utilizando herencia siempre que se pueda, realizar un diseño UML de clases del sistema indicando:

Las clases que se consideren necesarias con sus atributos y métodos. El tipo de privacidad más idóneo. Las relaciones entre clases: asociación, dependencia, herencia, agregación, composición, etc. La multiplicidad en las relaciones entre clases. Al menos una de las clases debe ser abstracta (indicarlo claramente).

Solución propuesta: