



Entornos de Desarrollo y Programación

Entrega 4 (PR)

El objetivo de esta práctica conjunta es aprender a analizar enunciados y obtener código a partir de ellos.

En cuanto a la parte de Entornos de Desarrollo, se busca que analicéis los enunciados, extrayendo los datos que sean necesarios para poder elaborar el diagrama de clases. Los diagramas de clases deben entregarse en un pdf con orientación horizontal, en el que cada página se corresponderá a un único ejercicio.

Una vez que se tiene el diagrama de clases refinado, se busca que lo utilicéis de base para desarrollar el proyecto en Java, con todos los archivos que sean necesarios. Cada uno de los ejercicios debe entregarse como un proyecto de Java diferente, en el que internamente se tendrá como mínimo, un archivo por cada clase, más el archivo con la clase ejecutable (la que contiene el método main).

Para ambas asignaturas la entrega deberá consistir en 5 ejercicios a elegir, los cuales se distribuirán de la siguiente manera:

- 2 ejercicios a elegir entre los ejercicios del 1 al 5.
- 2 ejercicios a elegir entre los ejercicios del 6 al 8.
- 1 ejercicio a elegir entre los ejercicios 9 y 10.

En mi github esta el código de cada ejercicio, dentro de PROYECTOS_PEPULU la carpeta PR_ENº2 se encuentran todos los ejercicios con su nombre correspondiente al ejercicio.

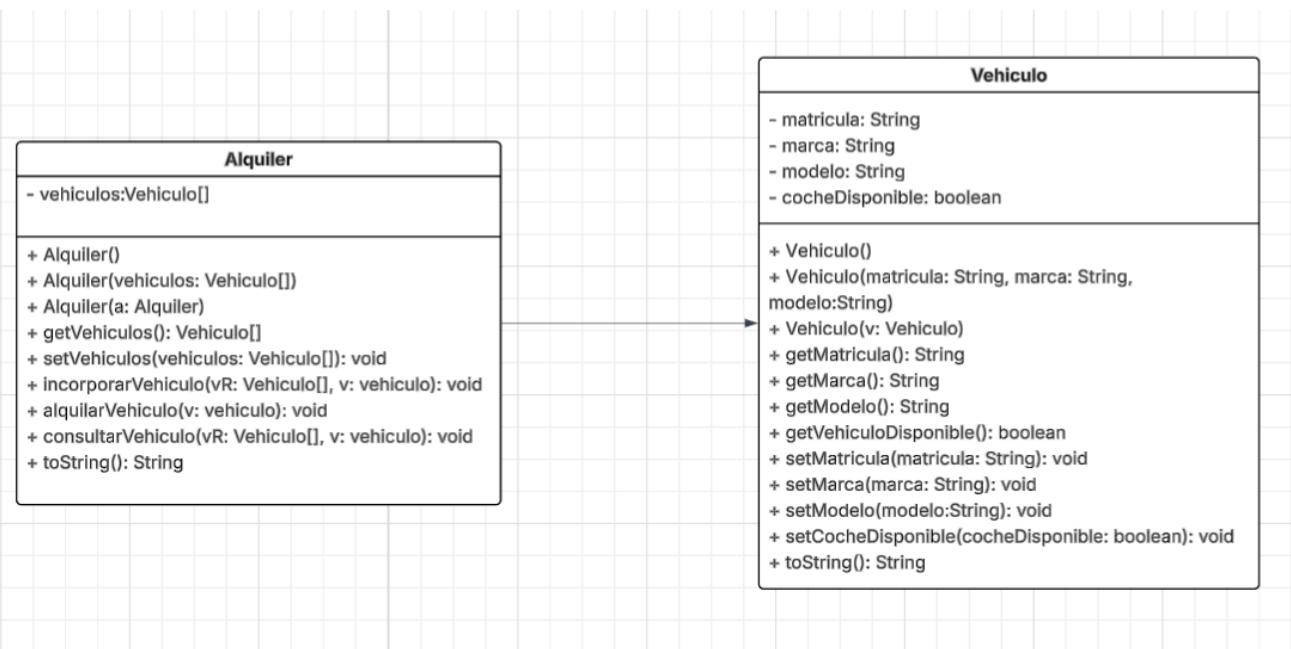
Los ejercicios con diagrama son los elegidos para la entrega, es decir, el ej 1, 2,

Ejercicio 1. Alquiler de vehículos

Una empresa dedicada al alquiler de vehículos quiere informatizar la gestión de su flota.

Para cada vehículo se necesita conocer su matrícula, la marca y el modelo. Cuando un vehículo se registra en el sistema, debe quedar marcado como disponible.

Desde la empresa se deben poder incorporar nuevos vehículos y alquilar un vehículo concreto, de manera que deje de estar disponible mientras dure el alquiler. En cualquier momento debe ser posible consultar la información completa de un vehículo.



Ejercicio 2. Cuentas bancarias

Un banco necesita un sistema para gestionar las cuentas de sus clientes.

De cada cuenta se almacena un número identificativo y el saldo disponible. Al crear una cuenta, el saldo inicial será cero.

El sistema debe permitir realizar ingresos y retiradas de dinero, siempre que haya saldo suficiente, así como consultar el saldo actual en cualquier momento.

Cliente
- account: Cuenta
+ Cliente()
+ Cliente(account: Cuenta)
+ Cliente(cli: Cliente)
+ getAccount(): Cuenta
+ setAccount(account: Cuenta): void
+ ingresarSaldo(saldoAIngresar: int): void
+ retirarSaldo(saldoARetirar: int): void
+ consultarSaldo(): int
+ toString(): String

Cuenta
- nIdent: int
- saldo: int
+ Cuenta()
+ Cuenta(nIdent: int, saldo: int)
+ Cuenta (cu: Cuenta)
+ getNIdent(): int
+ getSaldo(): int
+ setNIdent(nIdent:int): void
+ setSaldo(saldo:int):void
+ toString(): String

Ejercicio 3. Pedidos en una tienda online

Una tienda online quiere gestionar los pedidos realizados por sus clientes. De cada producto se almacena su nombre y su precio.

Un pedido puede incluir varios productos y, a partir de ellos, debe poder calcularse el importe total.

Cuando se crea un pedido, inicialmente no contiene productos y su importe es cero. El sistema debe permitir añadir productos a un pedido y consultar el total acumulado.

Ejercicio 4. Biblioteca municipal

Una biblioteca municipal desea informatizar el préstamo de libros a sus usuarios.

De cada libro se guarda el título y el autor. De cada usuario se guarda el nombre y un número que lo identifica como socio.

Los usuarios pueden llevarse libros prestados durante un periodo de tiempo. Un libro no puede prestarse si ya se encuentra prestado.

El sistema debe permitir registrar préstamos y comprobar si un libro está disponible.

Ejercicio 5. Plataforma de formación online

Una plataforma de formación online ofrece distintos cursos a los que se pueden apuntar estudiantes.

De cada curso se almacena su nombre y la duración en horas. De cada estudiante se guarda el nombre y su correo electrónico.

Los estudiantes pueden apuntarse a uno o varios cursos. El sistema debe permitir consultar en qué cursos está inscrito un estudiante.

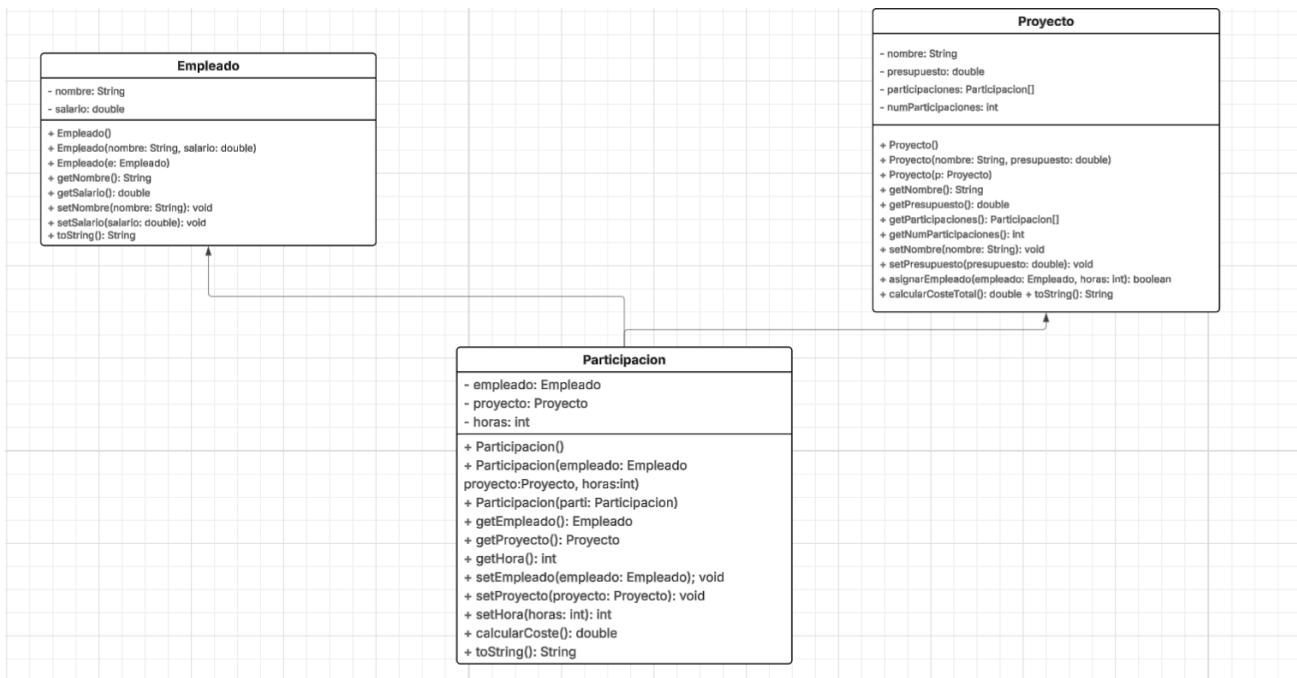
Ejercicio 6. Empresa y proyectos

Una empresa desarrolla distintos proyectos en los que participan sus empleados.

De cada empleado se guarda su nombre y su salario. De cada proyecto se conoce su nombre y el presupuesto asignado.

Un empleado puede participar en varios proyectos y un proyecto puede contar con varios empleados. Además, para cada participación se necesita conocer el número de horas dedicadas.

El sistema debe permitir asignar empleados a proyectos y calcular el coste total de un proyecto teniendo en cuenta las horas trabajadas.

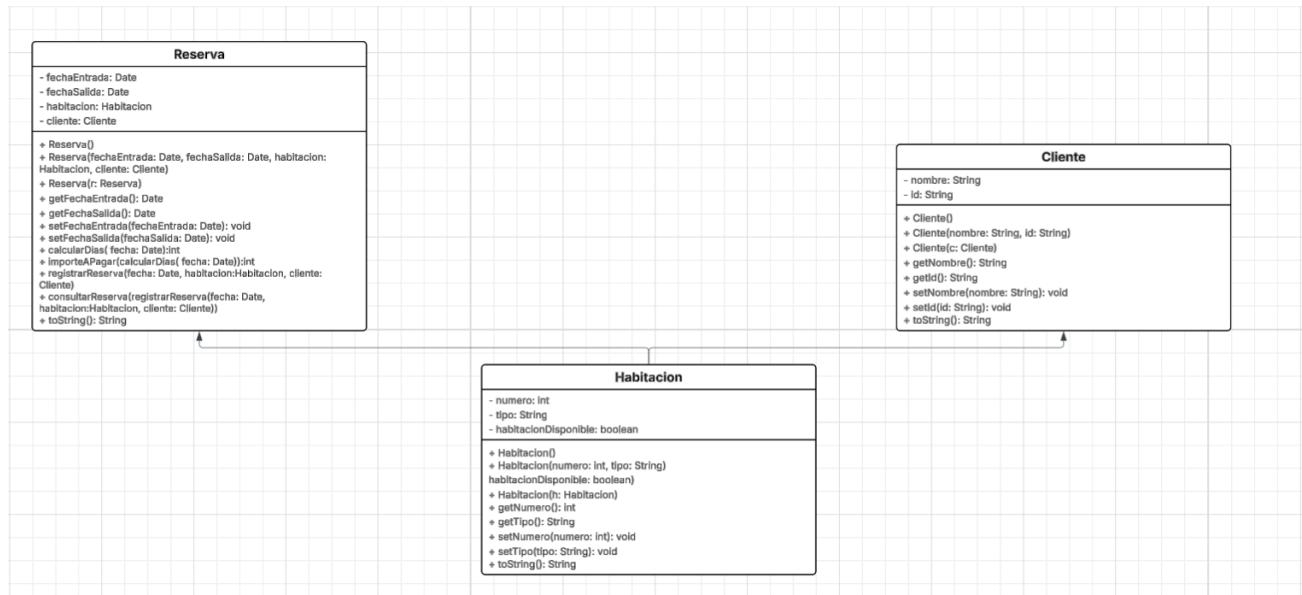


Ejercicio 7. Gestión hotelera

Un hotel quiere informatizar la gestión de las reservas realizadas por sus clientes.

De cada cliente se guarda el nombre y su documento de identidad. De cada habitación se conoce su número y el tipo de habitación. Las reservas se realizan indicando una fecha de entrada y una fecha de salida.

A partir de la información de la reserva debe poder calcularse el importe a pagar. El sistema debe permitir registrar reservas y consultar la información completa de las mismas.



Ejercicio 8. Videojuego

Un videojuego multijugador necesita gestionar a los jugadores y las partidas que se disputan.

De cada jugador se guarda su nombre y un nivel inicial que comienza en uno. Las partidas tienen una duración y en ellas participan varios jugadores.

Al finalizar una partida, el nivel de los jugadores participantes aumenta en una unidad. El sistema debe permitir crear partidas, añadir jugadores y finalizar partidas.

Ejercicio 9. Red social

Una red social quiere gestionar la actividad de sus usuarios.

De cada usuario se guarda su nombre y su dirección de correo electrónico. Cada usuario dispone de un perfil con información personal.

Los usuarios pueden publicar contenidos. Sobre estos contenidos se pueden realizar comentarios y expresar reacciones.

El sistema debe permitir crear publicaciones, añadir comentarios y contabilizar las reacciones asociadas a cada contenido.

Ejercicio 10. Centro deportivo

Un centro deportivo quiere desarrollar una aplicación para gestionar su funcionamiento diario.

De cada socio se almacena su nombre y un número que lo identifica. El centro ofrece distintas actividades, cada una con un horario y una persona responsable de impartirla. Los socios pueden apuntarse a las actividades ofertadas.

Cada vez que un socio se apunta a una actividad se genera un importe que debe quedar registrado.

El sistema debe permitir:

- inscribir socios en actividades
- consultar las actividades de un socio
- calcular los ingresos totales del centro

