# GESTIÓN DE MUELLES PARA DESCARGA DE CAMIONES

## 1. PRIMERA PARTE – Definición de modelos (2 puntos)

Crea el paquete **`models`** y dentro las siguientes clases:

1. **`Camion`** (abstracta)
    a. Atributos protegidos:
        - `String matricula`
        - `String empresaTransportista`
        - `String tipoMercancia`
        - `double pesoEstimado`
    b. Métodos:
        - Constructor con todos los campos.
        - Getters, setters y `toString()`.
2. **`CamionFrigorifico`** (hereda de `Camion`)
    a. Atributo adicional: `double temperaturaObjetivo`.
    b. Métodos:
        - Constructor con todos los campos (incluida temperatura).
        - Getters, setters y `toString()`.
3. **`CamionPeligroso`** (hereda de `Camion`)
    a. Atributo adicional: `String clasePeligrosidad`.
    b. Métodos:
        - Constructor con todos los campos (incluida peligrosidad).
        - Getters, setters y `toString()`.
4. **`Muelle`**
    a. Atributos privados:
        - `int id`
        - `String ubicacion` (p.ej. “Norte-1”, “Sur-3”)
        - `boolean ocupado`
        - `Camion camionAsignado` (null si está libre)
    b. Implementa:
        - Constructor con `id` y `ubicacion` (inicializa `ocupado = false`).
        - Getters, setters y `toString()`.
        - Método `void asignarCamion(Camion c)` que marca `ocupado = true` y guarda `camionAsignado`.
        - Método `void liberarMuelle()` que marca `ocupado = false` y limpia `camionAsignado`.

5. **`ReservaMuelle`**
    a. Atributos privados:
        - `int id`
        - `Camion camion`
        - `Muelle muelle`
        - `LocalDateTime llegada`
        - `LocalDateTime salida` (puede ser null hasta liberación)
    b. Implementa:
        - Constructor con `id`, `camion`, `muelle`, `llegada` (inicializa `salida = null`).
        - Getters, setters, equals por id y `toString()`.
        - Método `void registrarSalida(LocalDateTime hora)` que fija `salida` y llama a `muelle.liberarMuelle()`.

## 2. SEGUNDA PARTE – Lógica de negocio (3 puntos)

En el paquete **`services`**, crea la clase **`DockService`** con:

a. Atributos privados:
    - `List<Muelle> muelles`
    - `Set<ReservaMuelle> reservas`
b. Métodos que implementar:
- `void addMuelle(Muelle m)`
- `boolean removeMuelle(int id)`
- `Optional<Muelle> findFreeMuelle() -->` busca el primer muelle con `ocupado == false`.
- `ReservaMuelle createReserva(Camion c, LocalDateTime llegada) -->` utiliza `findFreeMuelle()`; si hay, crea `ReservaMuelle` con nuevo id, llama a `muelle.asignarCamion(c)` y añade la reserva. Lanza excepción si no hay muelles libres.
- `boolean liberarMuelle(int idReserva, LocalDateTime salida) -->` busca la reserva; si existe y `salida == null`, llama a `reserva.registrarSalida(salida)` y devuelve `true`, si no `false`.
- `List<ReservaMuelle> getReservasActivas() -->` filtra reservas con `salida == null`.
- `List<ReservaMuelle> getHistorial() -->` devuelve todas las reservas ordenadas por fecha de llegada.
- Map<Muelle, Long> getNumeroReservasPorMuelle(), cuenta el número de reservas por muelle.

- `Map<DayOfWeek, Long> getConteoReservasPorDia() -->` cuenta con Streams el número de reservas activas agrupadas por el día de la semana de llegada.
- `List<ReservaMuelle> getReservasFrigoríficasOrdenadasPorLlegada(boolean asc) -->` devuelve todas las reservas de camiones frigoríficos ordenadas por fecha y hora de llegada.
- Map<String, List<ReservaMuelle>> getReservasPorEmpresaTransportista(), agrupa las reservas por empresa transportista.
- List<Camion> getCamionesMasPesados(int n): devuelve una lista con los **n** camiones más pesados.

## 3. TERCERA PARTE – Entrada/Salida de ficheros (3 puntos)

En la raíz del proyecto coloca dos CSV:

- **`muelles.csv`** (sin cabecera): *`id,ubicacion`*

- **`reservas.csv`** (sin cabecera): *`id,matricula,empresaTransportista,tipoMercancia,pesoEstimado,llegada,salida,idMuelle,tipoCamion,atributoExtra`*

    - `tipoCamion` indica `FRIGORIFICO` o `PELIGROSO`.
    - `atributoExtra` es `temperaturaObjetivo` o `clasePeligrosidad`.

Crea el paquete **`io`** y la clase **DAO** con métodos estáticos:

1. `static DockService cargarDesdeCSV()`
    a. Lee `muelles.csv` y crea cada `Muelle`.
    b. Lee `reservas.csv`, por cada línea:
        - Instancia el `Camion` adecuado según `tipoCamion`.
        - Crea `ReservaMuelle` (parsea fechas y el muelle) y marca ocupado si `salida == null`.
    c. Devuelve el `DockService` poblado.
2. `static void guardarEnCSV(DockService svc)`
    a. Escribe de nuevo `muelles.csv` y `reservas.csv`, incluyendo `tipoCamion` y su `atributoExtra`.

## 4. CUARTA PARTE – Aplicación consola (2 punto)

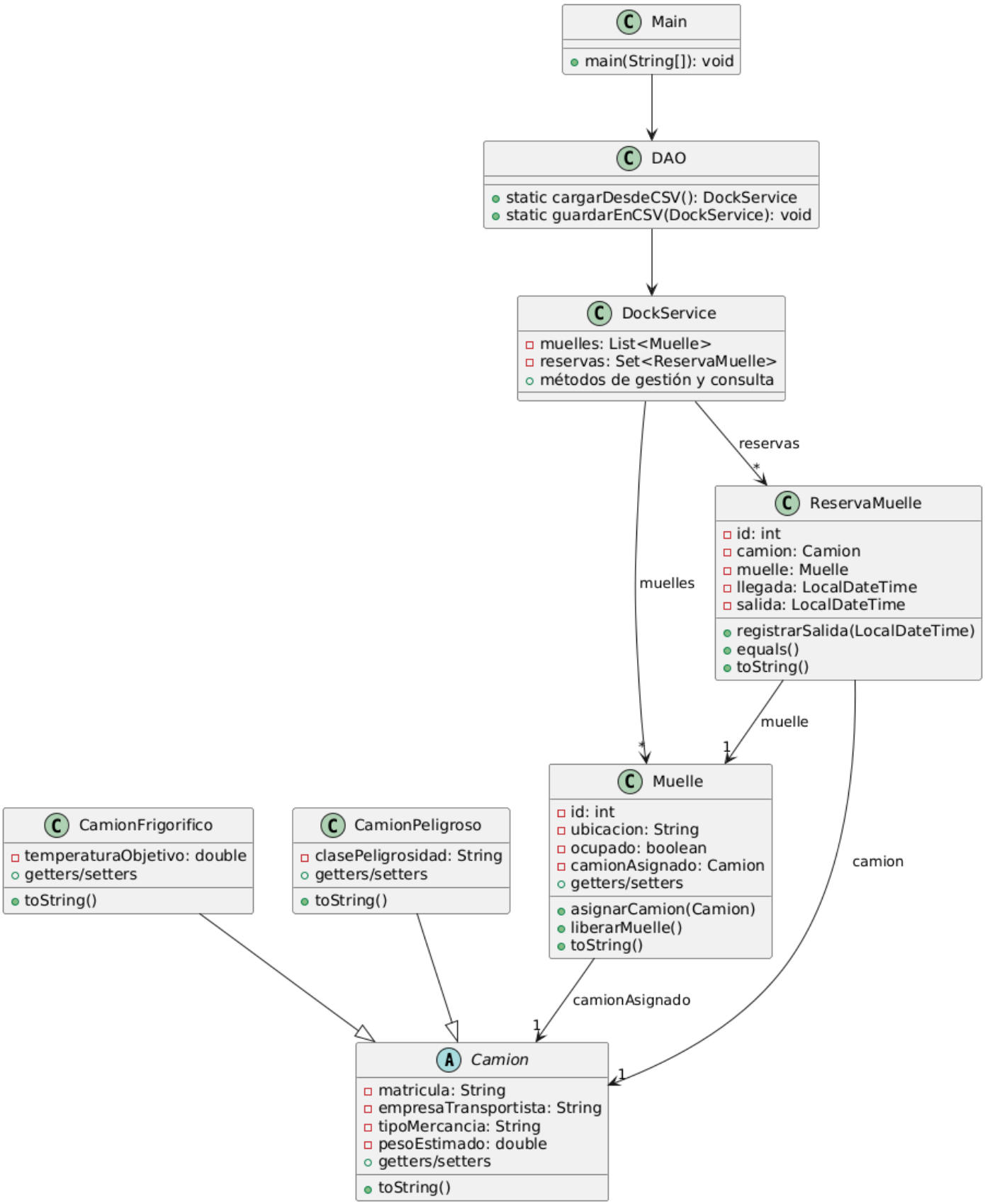
1. En el paquete **app**, crea la clase **`Main.`** `En el main crea la carga` inicial:

   ```
   DockService svc = DAO.cargarDesdeCSV();
   ```

2. Menú interactivo (do–while) con opciones:
   a. **Listar muelles**
   b. **Listar reservas activas**
   c. **Registrar llegada de camión**
      i. Pide datos: matrícula, empresa, tipo (`FRIGORIFICO`/`PELIGROSO`), atributo extra, peso y hora llegada.
      ii. Crea la subclase correspondiente y llama a `createReserva`.
   d. **Registrar salida de camión** (pide idReserva y hora salida)
   e. **Mostrar historial de reservas**
   f. **Mostrar tiempo de ocupación por muelle**
   g. **Salir** (llama a `DAO.guardarEnCSV(svc)` y termina).

## ENTREGA (1 punto extra)

- ZIP del proyecto Maven completo con:
  - Código fuente (`.java`).
  - `.jar` generado.
  - JavaDoc.
- README con instrucciones de compilación y ejecución.


Main
+ main(String[]): void
DAO
+ static cargarDesdeCSV(): DockService
+ static guardarEnCSV(DockService): void
DockService
- muelles: List<Muelle>
- reservas: Set<ReservaMuelle>
o métodos de gestión y consulta
reservas
*
muelles
*
ReservaMuelle
- id: int
- camion: Camion
- muelle: Muelle
- llegada: LocalDateTime
- salida: LocalDateTime
+ registrarSalida(LocalDateTime)
+ equals()
+ toString()
muelle
1
Muelle
- id: int
- ubicacion: String
- ocupado: boolean
- camionAsignado: Camion
o getters/setters
+ asignarCamion(Camion)
+ liberarMuelle()
+ toString()
camion
CamionFrigorifico
- temperaturaObjetivo: double
o getters/setters
+ toString()
CamionPeligroso
- clasePeligrosidad: String
o getters/setters
+ toString()
camionAsignado
1
Camion
- matricula: String
- empresaTransportista: String
- tipoMercancia: String
- pesoEstimado: double
o getters/setters
+ toString()
1