



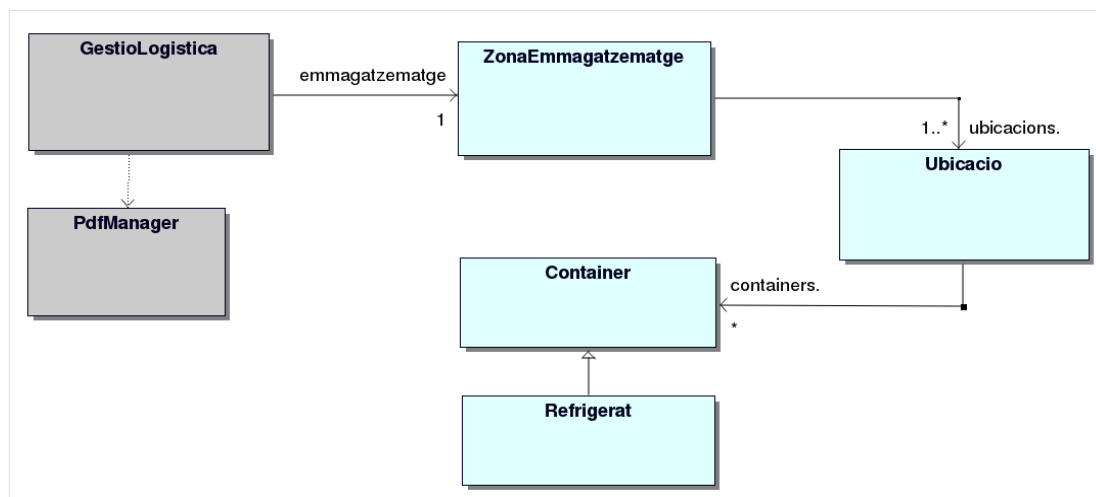
DAM2/DAW2. UF5 Pe1 2019-20

Enunciat

Cal que implementeu un sistema que simula la logística d'emmagatzematge dels contenidors de mercaderies d'un port.

A la figura 1 es mostra el diagrama de classes simplificat amb les classes del sistema i les relacions entre elles. A l'[annex A](#) trobareu el diagrama detallat amb els atributs i mètodes de cada classe.

Figura 1. Diagrama de classes simple



- «**GestioLogistica**» és l'entrada a l'aplicació de gestió i inclou la **zona d'emmagatzematge** del port.
- «**ZonaEmmagatzematge**» representa l'espai del port on es col·loquen els contenidors. Té forma rectangular i es divideix en **vàries ubicacions** formant una graella.
- «**Ubicacio**» representa una parcel·la dins la zona d'emmagatzematge **destinada a apilar els contenidors** un a sobre dels altres. S'identifica per dues coordenades: fila i columna. El nombre de contenidors que s'hi poden apilar està limitat (màxim 5), així com també el pes màxim que s'hi pot posar (30 tones).
- «**Container**» representa **un contenidor** de mercaderies; inclou un identificador únic, el port origen, el pes en tones del contenidor amb la càrrega i la data d'arribada al port. Tots tenen prioritat normal.
- «**Refrigerat**» representa un contenidor **amb sistema de refrigeració** i transporta càrrega que pot fer-se malbé, per això inclou el nombre màxim de dies que pot estar emmagatzemat el contenidor des de la seva arribada. Mentre no superi el màxim de dies la seva prioritat és urgent, en cas contrari és crítica.

Adjunt a l'enunciat trobareu el projecte per importar a Eclipse i les classes «GestioLogistica» que



conté el mètode «main», les dades de prova i els mètodes per generar un document PDF amb l'esquema de la zona d'emmagatzematge amb els contenidors apilats, podeu consultar-ne un exemple a l'[annex B](#).

Nucli 1. Estructures dinàmiques

Que cal fer?

- Definir només els **constructors indicats** a l'esquema.
- **Afegir getters/setters** per tots els atributs que no siguin col·leccions.
- Per cada col·lecció, **decidir l'estructura** de dades a utilitzar.
- **Implementar la resta de mètodes** de l'esquema segons els requeriments sense canviar el nom, ni els paràmetres ni el tipus de retorn d'aquests.
- Si et cal, pots afegir **altres mètodes** sempre que siguin **privats**.

Requeriments. Descripció de les classes i mètodes.

Classe «Container»	
<code>+Map<Integer, String> PRIORITAT</code>	Mapa estàtic i final sense ordre que relaciona els nombres i valors de les prioritats. El mapa ha d'incloure les entrades següents: 1 - "CRITICA" 2 - "URGENT" 3 - "NORMAL"
<code>+Container(identificador, origen, carrega, arribada)</code>	Nova instància d'un contenidor amb les dades indicades per paràmetre.
<code>+diesDesdeArribada(): int</code>	Nombre de dies des de la data d'arribada del contenidor. Per obtenir el nombre de dies entre dues dates «LocalDate» <code>Period.between(data1,data2).getDays()</code>
<code>+prioritatSortida(): int</code>	La prioritat de sortida dels contenidors és 3.

Classe «Refrigerat» (Hereta de «Container»)	
<code>+Refrigerat(identificador, origen, carrega, arribada, maxim)</code>	Nova instància d'un contenidor amb les dades indicades per paràmetre.
<code>+prioritatSortida(): int</code>	La prioritat de sortida dels contenidors refrigerats és 1 (Crítica) si superen el nombre màxim de dies des de l'arribada a port, o 2 (Urgent) en cas contrari.

Classe «Ubicacio»	
<code>-Collection<Ubicacio> containers</code>	Pila de contenidors allotjats a la ubicació.
<code>+Ubicacio(fila, columna)</code>	Nova ubicació amb coordenades fila i columna.
<code>+getContainers(): Collection<Ubicacio></code>	Retorna la pila de contenidors.
<code>+disponible(container): boolean</code>	Indica si la ubicació compleix els requeriments per poder



	apilar el contenidor, o sigui que una vegada apilat no supera la capacitat (5 contenidors) ni el pes màxim (30 tones en total).
<code>+afegir(container)</code>	Apila el contenidor sempre que la ubicació compleixi els requeriments (<i>veure mètode #disponible(container)</i>).
<code>+treure(): Container</code>	Si la pila de contenidors no està buida, treu i retorna el contenidor del cim.
<code>+cercar(identificador): boolean</code>	Indica si hi ha un contenidor amb aquest identificador a la pila de contenidors.
<code>+superior(): Container</code>	Retorna el contenidor del cim de la pila de contenidors o bé nul si la pila està buida.
<code>+totalContenidors(): int</code>	Retorna el nombre total de contenidors a la ubicació.

Classe «ZonaEmmagatzematge»	
<code>-Collection<Ubicacio> ubicacions</code>	Conjunt d'ubicacions sense repetits ordenada per les coordenades: fila i columna de les ubicacions.
<code>+ZonaEmmagatzematge(files, columnes)</code>	Constructor. Crea les instàncies de totes les ubicacions disponibles disposades en una graella de les files i columnes indicades per paràmetre.
<code>+getUbicacions(): Collection<Ubicacio></code>	Retorna el conjunt d'ubicacions.
<code>+ubicar(container)</code>	Apila el container a la primera ubicació disponible respectant les restriccions en relació al nombre màxim de contenidors (5) i al pes màxim de tots ells (30 tones).
<code>+retirar(identificador): Container</code>	Localitza la ubicació on es troba el contenidor amb aquest identificador i el treu de la pila. Cal tenir en compte que si hi ha altres contenidors apilats a sobre, primerament cal reubicar-los abans de poder retirar-lo.
<code>+totalContenidors(): int</code>	Retorna el nombre total de contenidors a totes les ubicacions.

Nucli 2. Excepcions

Que cal fer?

- Cal que **llenceu i propagueu excepcions** amb un missatge adient sempre que es puguin produir les situacions de la llista següent.
- Per llançar noves excepcions heu de fer servir l'excepció més genèrica que **no es propaga automàticament**. («Exception»).
- També cal que afegiu noves accions a les dades de prova de la classe principal que **provoquin les situacions d'error** indicades a la segona llista i **capturar-les individualment** per garantir que el programa finalitza correctament.
- Per cada excepció, mostreu el missatge d'error per consola.



Llista de situacions que cal evitar:

1. Contenedors amb identificador nul o de longitud diferent a 11 caràcters.
2. Contenedors amb origen nul o buit.
3. Contenedors amb càrrega inferior a 2.
4. Contenedors amb data d'arribada nul·la o posterior a la data actual.
5. Contenedors refrigerats amb un nombre màxim de dies negatiu.
6. Ubicacions amb coordenades (fila o columna) negatives.
7. Des de ZonaEmmagatzematge no poder ubicar contenidors per manca de disponibilitat.
8. Des de ZonaEmmagatzematge no trobar un contenidor alhora de retirar-lo.

Llista d'accions de prova per generar situacions d'error:

1. Afegiu un nou contenidor amb les següents dades:
`"NYKU1234", "", 1, LocalDate.of(2030, 1, 26)`
2. Afegir un nou contenidor amb les següents dades
`"NYKU0012340", "Anvers", 40, LocalDate.of(2020, 2, 3)`
3. Retireu el contenidor: `"TOLU0012340"`

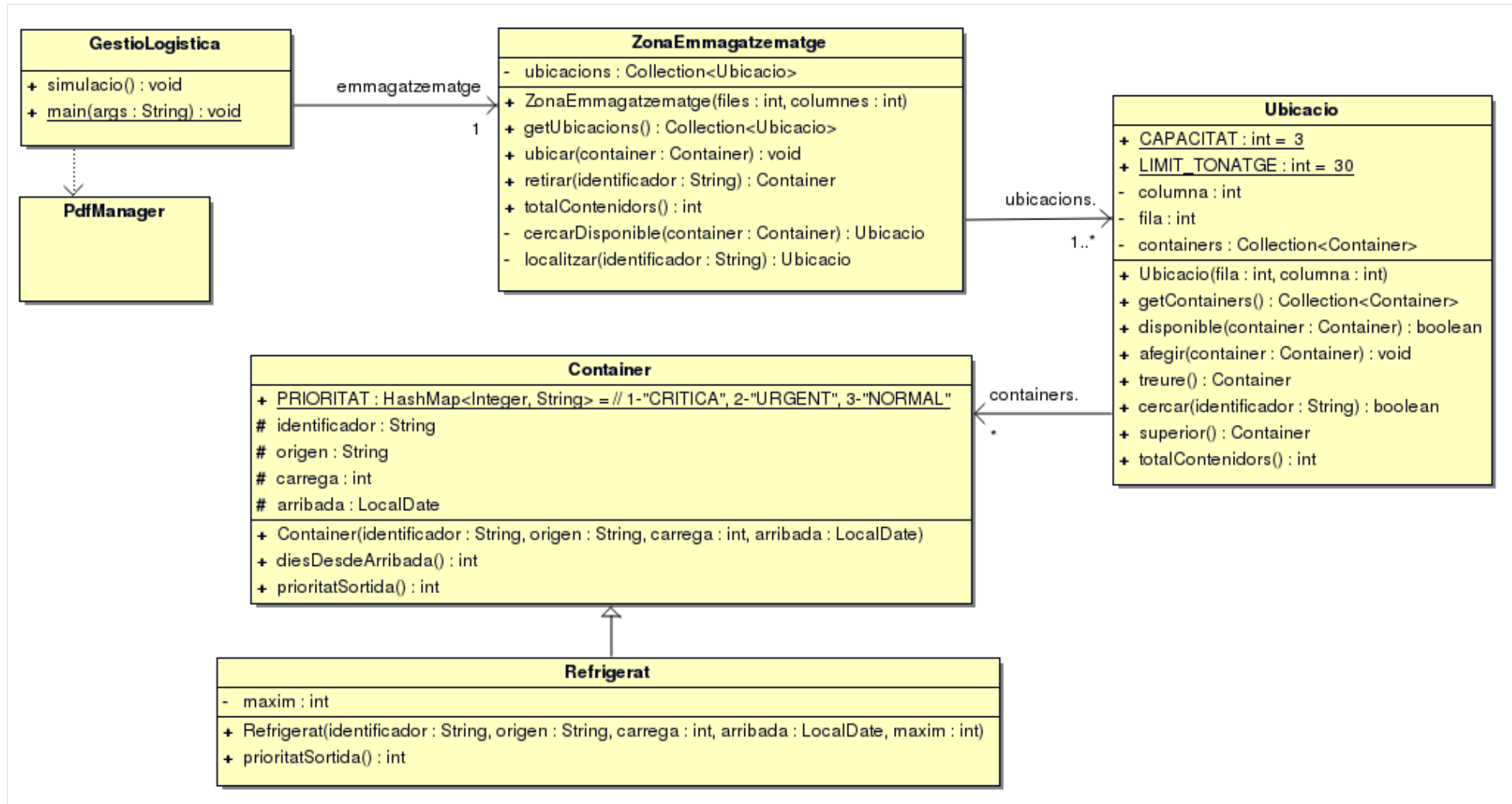
Nucli 3. Persistència

Que cal fer?

- Crear un mètode a la classe «**GestioLogistica**» per desar la zona d'emmagatzematge en un fitxer en format xml.
- Crear un mètode a la classe «**Blog**» per carregar la zona d'emmagatzematge des d'un fitxer en format xml.
- Fer els canvis necessaris a les classes per poder serialitzar-les amb les llibreries XML.
- Qualsevol error s'ha de capturar i llençar una excepció com les de l'apartat anterior amb un missatge personalitzat.

Classe Blog	
<code>+desarZona(String fitxer)</code>	Desar la zona d'emmagatzematge en un fitxer en format xml.
<code>+carregarZona(String fitxer)</code>	Carregar la zona d'emmagatzematge des d'un fitxer en format xml.

Annex A. Diagrama de classes UML





Annex B. PDF esquema zona emmagatzematge



Generalitat de Catalunya
Departament d'Ensenyament
INS Marianao

