



# PRÀCTICA 3: GESTOR DE ALQUILERES

PROGRAMACIÓ II GEI



**Treball fet per:**

Pau Esquerda Masip - 48056677V

25/05/2025 PRIMER CURS GEI - UDL

## La Classe Product

Aquesta classe pretén representar un producte el qual pot ser llogat per un client. Usem un constructor per a definir els paràmetres d'aquest:

### Constructor Product

El constructor Product consta de diferents paràmetres que permeten representar les distintes característiques d'un producte dins del sistema com poden ser: id (identificador), description, price i stock (unitats disponibles).

### Getters

Disposem de mètodes getters per a obtenir el valor de cada un dels paràmetres de Product. Aquests ens retornen l'id, la descripció el preu o l'estoc del producte.

### Incrementar i decrementar estoc

També disposem de dos mètodes per a incrementar o decrementar l'estoc en una unitat. Aquests mètodes seran de gran ajuda a l'hora de llogar o retornar un producte ja que l'estoc es veurà afectat per aquests moviments.

### Mètodes fromBytes i toBytes

Aquests mètodes s'utilitzen per a emmagatzemar la informació d'un producte en forma de bytes. En concret el mètode toBytes empaqueta els diferents paràmetres d'un producte i els va afegint en un array de Bytes en la posició que indica l'offset, el qual es va incrementant en funció dels bytes necessaris per a emmagatzemar cada tipus de dada, (int - 4 bytes, long - 8 bytes, String - 2\*String\_length bytes).

## La Classe Client

Aquesta classe representa un client dins del sistema amb capacitat per a llogar o retornar productes. Definim els paràmetres del client usant un constructor:

### Constructor Client

El constructor Client consta de diferents paràmetres que permeten representar les distintes característiques d'un client dins del sistema com poden ser: id (identificador), name, price i balance (capacitat econòmica del client). A part, s'han definit dues variables d'instància més per a representar els productes que el client té en la seva possessió.

### Variables d'instància afegides

Hem afegit dos arrays, un long[] per a representar els diferents id dels productes que té el client llogats i un int[] per a representar les unitats de cada producte que té el client. Aquests són de gran utilitat a l'hora de comprovar si un client pot o no llogar un nou producte.

## Getters

Disposem de mètodes getters per a obtenir el valor de cada un dels paràmetres de Client. Aquests ens retornen l'id, el nom i el balanç.

## Afegir i restar balanç

També disposem de dos mètodes per a afegir o restar el balanç d'un client. Aquests mètodes junt amb getBalance() seràn de gran ajuda a l'hora tant de llogar com de comprovar si el client realment pot pagar el lloguer d'un determinat producte.

## Mètodes relacionats amb la gestió de productes

- canAddProduct()
  - Aquest mètode comprova si el client pot llogar un producte o no tenint en compte la condició de que no pot tenir més de tres productes diferents. Recorrem l'array buscant elements que no siguin 0 i els contem. Mentre aquests elements no superin la quantitat de 3, el client pot llogar el producte.
- hasProduct()
  - Aquest mètode comprova si el client té ja llogat un determinat producte. Rep un idProduct com a parametre i el busca dins l'array de id llogades del client. Si coincideix es que el client ja té aquest producte llogat.
- rentProduct()
  - Aquest mètode serveix per a llogar un producte quan ja hem fet totes les comprovacions pertinents. Si el client ja té el producte llogat incrementa en 1 les unitats de producte, si no el té llogat, busca una posició disponible en l'array d'ids i col·loca una unitat de producte en un espai buit.
- returnProduct()
  - Aquest mètode serveix per a retornar un producte llogat anteriorment quan s'ha comprovat que el client el té en possessió. Es busca el producte a retornar dins de l'array de ids i quan es troba es resta una unitat de l'array d'unitats.

## Mètodes toBytes i fromBytes

A l'igual que els mencionats anteriorment en la classe Product, aquests serveixen per emmagatzemar la informació sobre un client en un array de bytes, calculant la mida corresponent en funció del tipus de dades i col·locant aquesta en la posició adequada de l'array usant l'offset.

## Les Classes ClientFile i ProductFile

Aquestes dues classes controlen els fitxers on es registren els diferents clients i productes. Tenen una estructura molt semblant, amb mètodes per a llegir i escriure en els fitxers, per a obtenir el següent id disponible per adjudicar a un producte/client, un mètode per a comprovar si els id son vàlids. Finalment mètodes per a obrir, tancar i resetejar els fitxers.

## La Classe ProdRent

Aquesta és la classe encarregada d'integrar totes les anteriors en una sola.

### Mètode processMovement

Aquest mètode (cradat per processMovements) s'encarrega de distribuïr les peticions a les seves corresponents funcions auxiliars. Llegeix el primer token del tokenizer i aquest indica el tipus d'ordre. Per exemple, ALTA\_PRODUCTO, descripción, precio, stock es redirigit a processAltaProducto.

### Mètode processAltaProducto

Aquesta funció processa l'alta d'un producte, és a dir, quan afegim un producte disponible per a llogar. Primerament agafem el preu i l'estoc utilitzant el tokenizer, comprovem que ni el preu ni l'estoc són zero o negatius i si els resultats son favorables procedim a crear un nou producte i afegir-lo al fitxer de productes.

### Mètode processAltaCliente

A l'igual que l'anterior, agafem les dades usant el tokenizer i comprovem que el balanç del client no sigui negatiu o zero. Si aquesta condició es favorable procedim a crear un nou perfil de client amb les dades obtingudes amb el tokenizer i l'escrivim al fitxer de clients.

### Mètode processInfoProduct

En aquest mètode obtenim la id del producte del qual volem saber la informació. Comprovem si aquesta id és vàlida usant isValid(), si ho és, procedim a llegir els paràmetres del producte i ho registrem al logFile.

### Mètode processInfoClient

A l'igual que en l'anterior mètode, comprovem que l'id és correcte, si és així, llegim la informació del client al fitxer corresponent. Creem un array long[] amb els id dels productes que el client té llogats i amb aquests ids creem un altre array de productes llogats. Finalment registrem la informació al logFile.

### Mètode processAlquilar

Primerament agafem els id del producte a llogar i del client que el vol llogar. Els llegim dels fitxers corresponents i comprovem que l'estoc del producte no és zero, que els id son vàlids, que el balanç del client no es menor que el preu del producte i

finalment comprovem que el client té espai per a llogar el producte amb `canAddProduct()`. Si tot és correcte lloguem el producte, restem el preu d'aquest al balanç del client i decrementem l'estoc, tot això amb mètodes fets anteriorment. Per finalitzar registrem els moviments als fitxers corresponents.

### **Mètode processDevolver**

Utilitzant els mateixos mitjans que en l'anterior, obtenim i llegim els id del client i el producte a ser retornat. Comprovem que els id siguin vàlids i que el client tingui el producte abans de retornar-lo. Si tot és favorable, retornem el producte, incrementem l'estoc i ho registrem als fitxers pertinents.