

Sistemes Distribuïts - Supermercat

(a) Es vol dissenyar una aplicació de gestió de productes d'un supermercat. En concret, ens interessa una operació de sistema:

context modificaPreu(codProd: String, nouPreu: Float) pre 1.1: existeix el producte de codi codProd pre 1.2: nouPreu > 0 post 2.1: enregistra que el preu del producte de codi codProd és nouPreu
--

Aquesta operació s'executa dins del cas d'ús ModificaPreu. La classe Producte té com a atributs *codi* (que actua com a clau), *preu* i *stock*.

Ara es vol efectuar el disseny de l'operació. A tal efecte, es decideix un disseny extern amb un desplegable de productes existents al sistema i un camp numèric per entrar el nou preu. Es demana que dissenyeu la capa de domini en el supòsit de disseny centralitzat amb patró de domini Domain Model amb consultes arbitràries. Mostreu també com interaccionen la capa de presentació i la de domini.

(b) L'arquitecte del software decideix convertir l'aplicació centralitzada en orientada a serveis i preparada per actuar en un entorn distribuït amb clients eventualment desconeguts actualment. Es vol que els diversos serveis resultants siguin robustos i reaccionin correctament sota condicions anòmales. Es demana que reformuleu la capa de domini en aquest nou context. Mostreu els canvis en la interacció de la capa de presentació i la capa de domini de l'aplicació. Mostreu també la interacció des del mètode següent de la capa de domini d'una altra aplicació desenvolupada posteriorment:

context ClasseClient::pujaIPC(cProd: String, ipc: Float) exc 1.1 producte-no-existeix: el producte cProd no existeix post 2.1: modifica el preu del producte cProd incrementant-lo amb l'ipc
