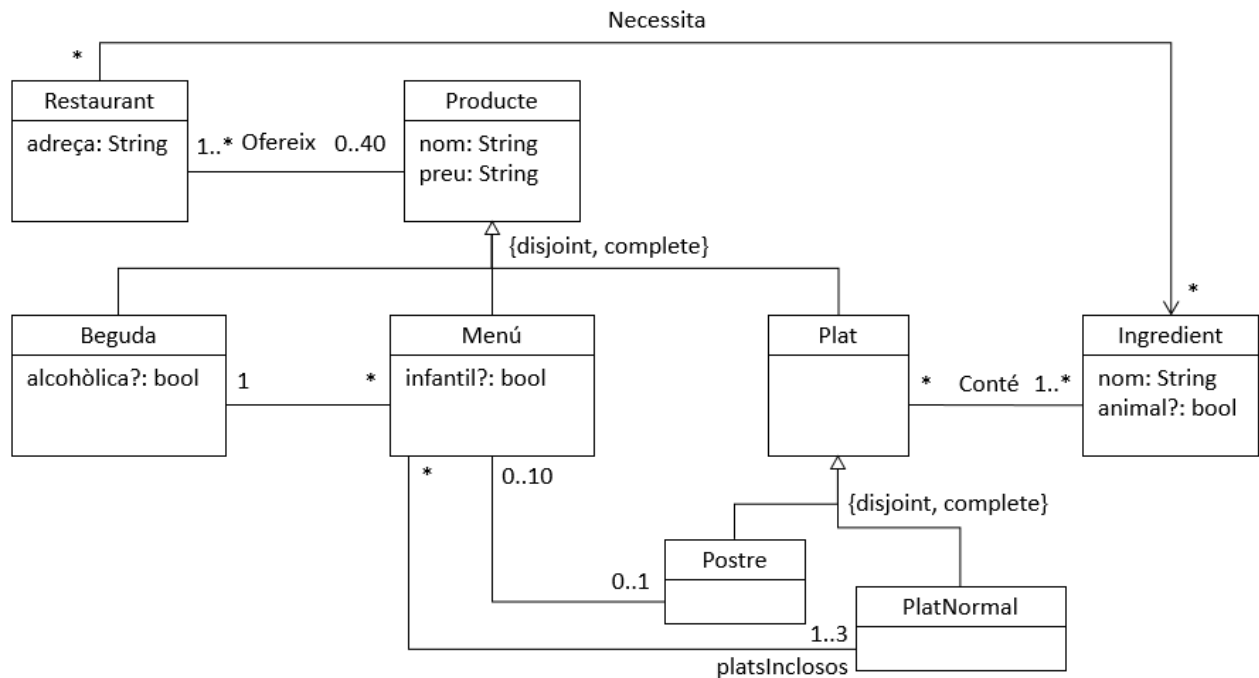


IES – 4rt problema de disseny

17 o 18 de desembre del 2018 – QT1819 (3a FHC d'IES QT1718)

Considereu un sistema per la gestió de l'oferta de productes d'una cadena de restaurants. Cada restaurant ofereix uns productes (begudes, menús o plats) compartits entre tota la cadena. El sistema també enregistra els ingredients de cada plat, i si l'ingredient té un origen animal.

Diagrama de classes de disseny



Restriccions textuais:

1. Claus externes: (Restaurant, *adreça*), (Producte, *nom*), (Ingredient, *nom*)
2. Un menú infantil no pot tenir una beguda alcohòlica.
3. Un *menú* no pot tenir un *preu* superior a la suma dels *preus* dels seus *productes* (*beguda*, *platsInclosos* i *postre*).
4. Un *restaurant* ofereix totes les *begudes* que s'inclouen en els seus *menús* oferts.

Informació derivada materialitzada

- *Necessita* és una associació derivada materialitzada (amb navegabilitat de *Restaurant* a *Ingredient*) que indica, donat un *restaurant*, quins *ingredients* contenen els *plats* elaborats en aquell restaurant (és a dir, els *ingredients* dels *plats* oferts pel *restaurant*, conjuntament amb els *ingredients* de les *postres* i els *platsInclosos* en els *menús* oferts pel *restaurant*).

Contracte de l'operació crearMenu

Operació: crearMenú(adreça: String, nomMenu: String, preu: int, nomPlat: String, nomPostre: String, nomBeguda: String, infantil: bool)

Exc:

- [NoExisteixRestaurant]: el restaurant identificat per *adreça* no existeix.
- [NoExisteixPlatNormal]: El plat normal identificat per *nomPlat* no existeix.
- [NoExisteixPostre]: El postre identificat per *nomPostre* no existeix.
- [NoExisteixBeguda]: La beguda identificada per *nomBeguda* no existeix.
- [JaExisteixProducte]: Ja existeix un producte amb nom *nomMenu*.
- [MassaProductes]: El restaurant *adreça* ja ofereix 40 productes.
- [PostreEnMassaMenús]: El Postre *nomPostre* ja s'usa en 10 Menús.
- [MenúInfantilIncorrecte]: *infantil* val cert quan la beguda *nomBeguda* és alcohòlica.
- [RestaurantNoOfereixBeguda]: El restaurant *adreça* no ofereix la beguda *nomBeguda*
- [PreuIncorrecte]: El *preu* del menú és superior a la suma del preu del *nomPlat*, *nomPostre* i *nomBeguda*.

Post:

- Es dona d'alta un nou Menú amb els valors de nom, preu i infantil passats per paràmetre, el plat *nomPlat* com a únic platInclós, la beguda *nomBeguda*, i el postre *nomPostre*.
- S'enregistra que el Restaurant *adreça* ofereix aquest nou Menú.
- S'actualitzen els ingredients necessitats pel Restaurant *adreça*.

Contracte de l'operació llistaMenusVegeterians

Operació: llistaMenusVegeterians(adreça: String): Set(TupleType(nom: String, preu: int, estalvi: int))

Exc:

- [NoExisteixRestaurant]: el restaurant identificat per *adreça* no existeix

Body:

Per cada menú ofert en el restaurant identificat per *adreça*, tal que ni cap dels seus platsInclusos ni el postre tenen un ingredient d'origen animal, es retorna la següent informació:

- el nom del menú
- el preu del menú
- L'estalvi. És a dir, la diferència entre el preu del menú, i la suma dels preus dels platsInclusos, de la beguda i del postre.

Es demana:

1. **[5 punts]** Diagrama de seqüència de l'operació *llistaMenusVegeterians* i de totes les operacions que siguin invocades en aquest diagrama de seqüència. Poseu comentaris de tot el que no hi aparegui de forma explícita. Utilitzeu el controlador transacció. Indiqueu les navegabilitats resultants d'aquesta operació (per exemple, si l'associació *Ofereix* és navegable de *Restaurant* a *Producte*, indiqueu *Restaurant -> Producte*) i les operacions que són abstractes.
2. **[5 punts]** Diagrama de seqüència de l'operació *crearMenú* i de totes les operacions que siguin invocades en aquest diagrama de seqüència. Poseu comentaris de tot el que no hi aparegui de forma explícita. Utilitzeu el controlador transacció. Indiqueu la navegabilitat resultant final (navegabilitats de les dues operacions) i les operacions que són abstractes.