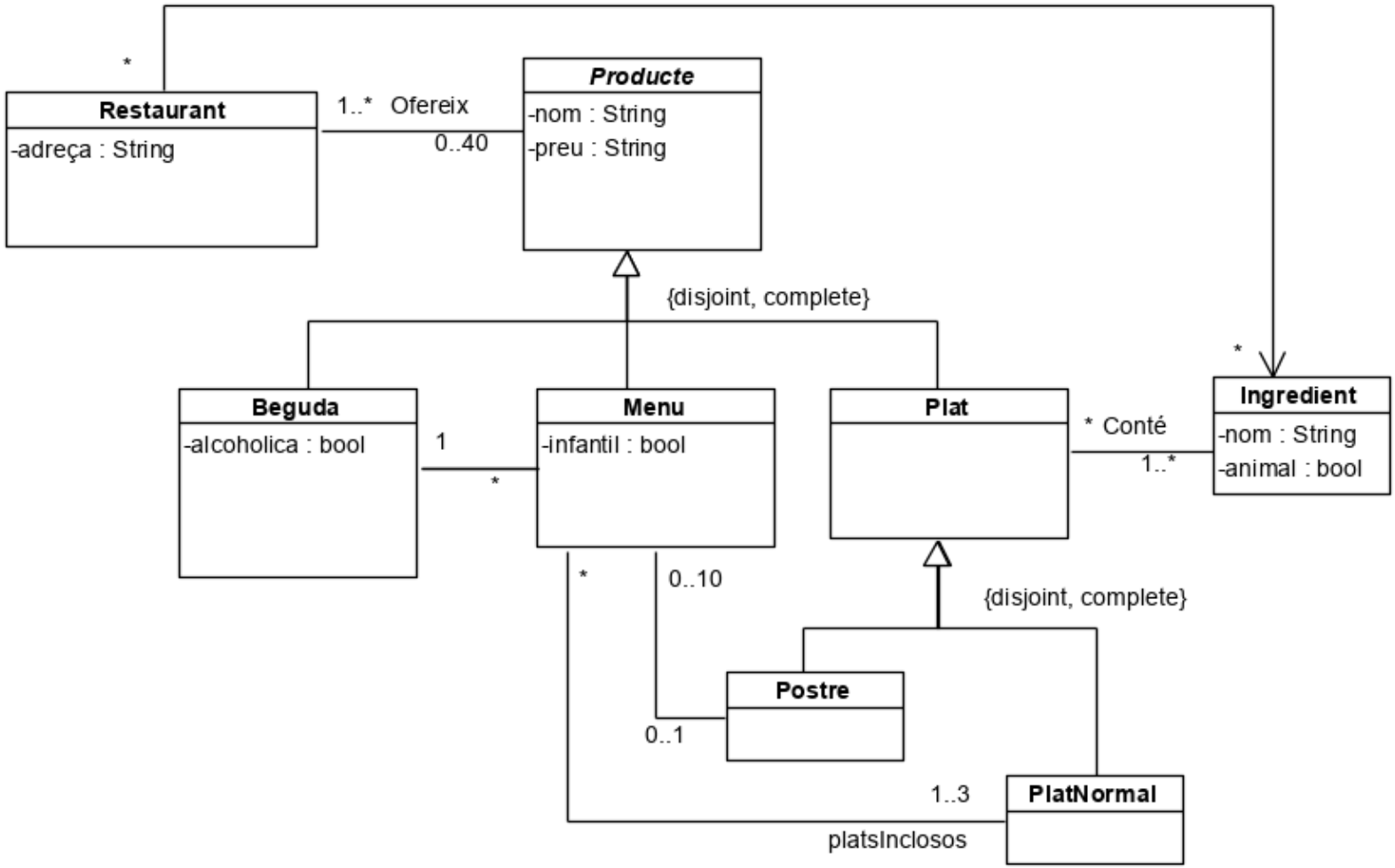
Problema de disseny en UML: Cadena de Restaurants

Enunciat: Presentació

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.

Enunciat: Diagrama de Classes de Disseny



Enunciat: Restriccions Textuals

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

Enunciat: Informació Derivada

- *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 r*estaurant*, conjuntament amb els *ingredients* de les *postres* i els *platsInclosos* en els *menús* oferts pel *restaurant*).

Operació 1: 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 dóna 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*.

Operació 2: LlistaMenusVegetarians

- **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 platsInclosos 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 platsInclosos, de la beguda i del postre.

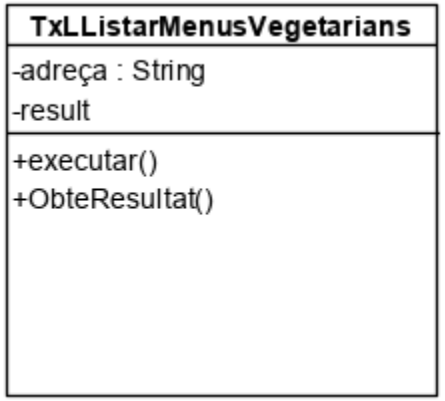
Enunciat: Què es demana?

- 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.

- 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.

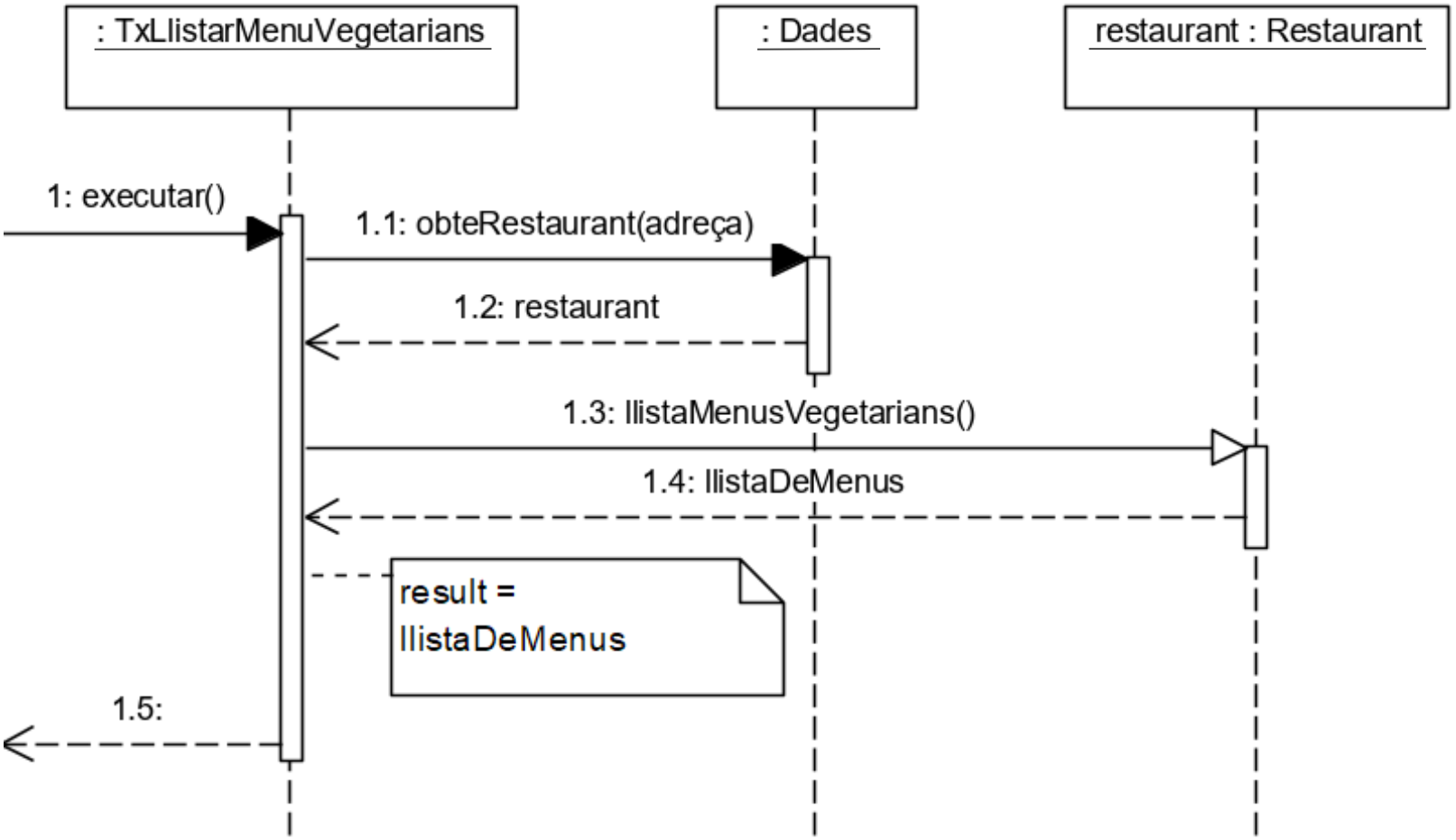
Solució LListarMenusVegetarians

LlistaMenusVegetarians



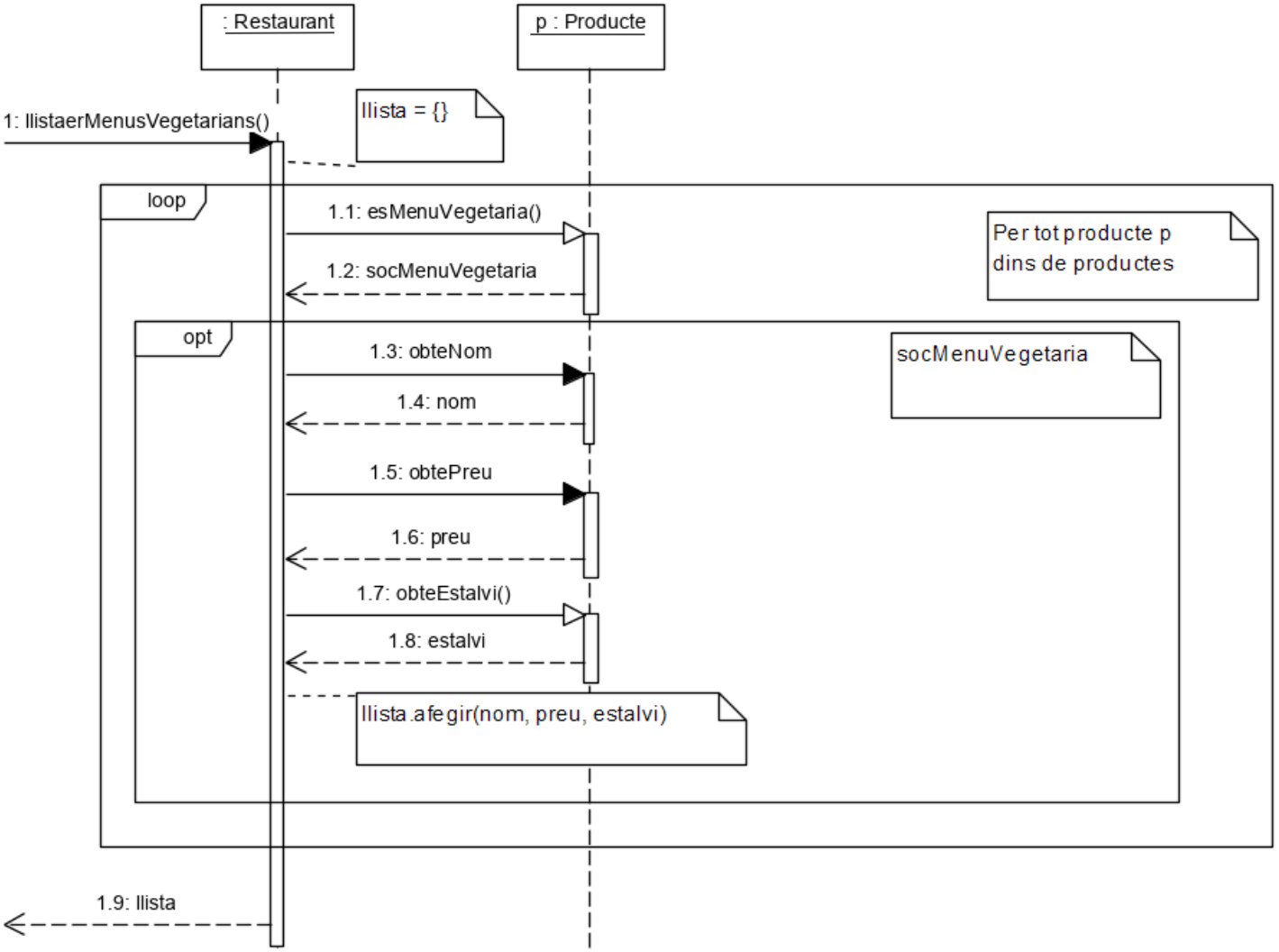
- Creem una clase que segueix el *Controlador Transacció* que conté
  - Un atribut per cada paràmetre de la operación
  - Un atribut resultat del tipus retornat per la operación
  - Una operación *Executar*()
  - Una operación de *ObteResultat*()

Executar



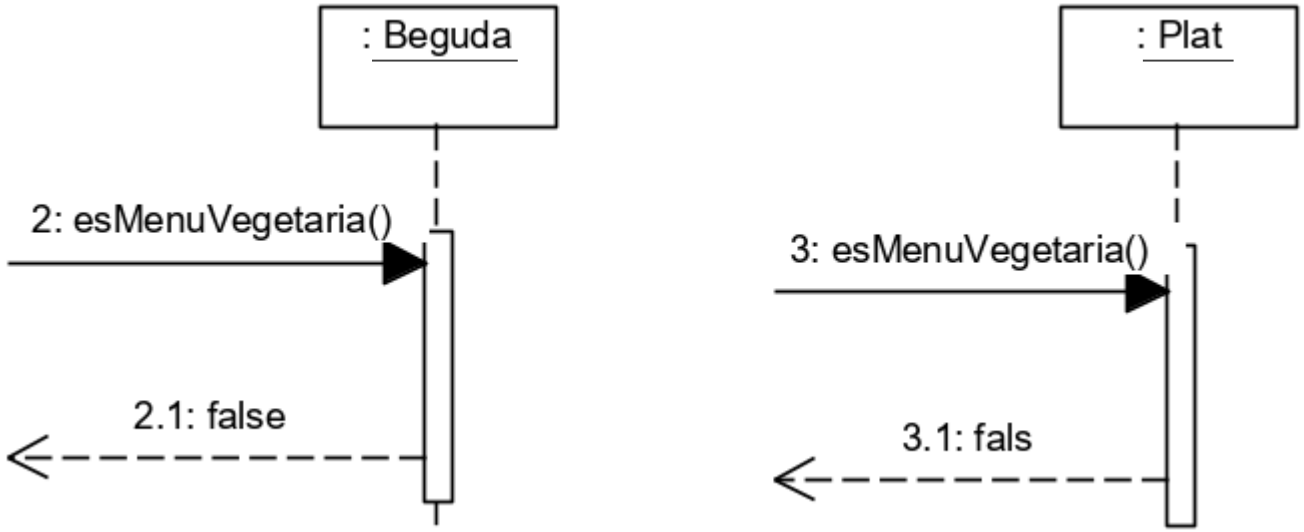
- Obtenim el restaurant de la Capa de Dades.
  - Això pot llençar la Excepció [NoExisteixRestaurant]
- Deleguem la resta de la operació a la classe Restaurant
- Ens guardem el resultat

Restaurant::LlistaMenusVegetarians



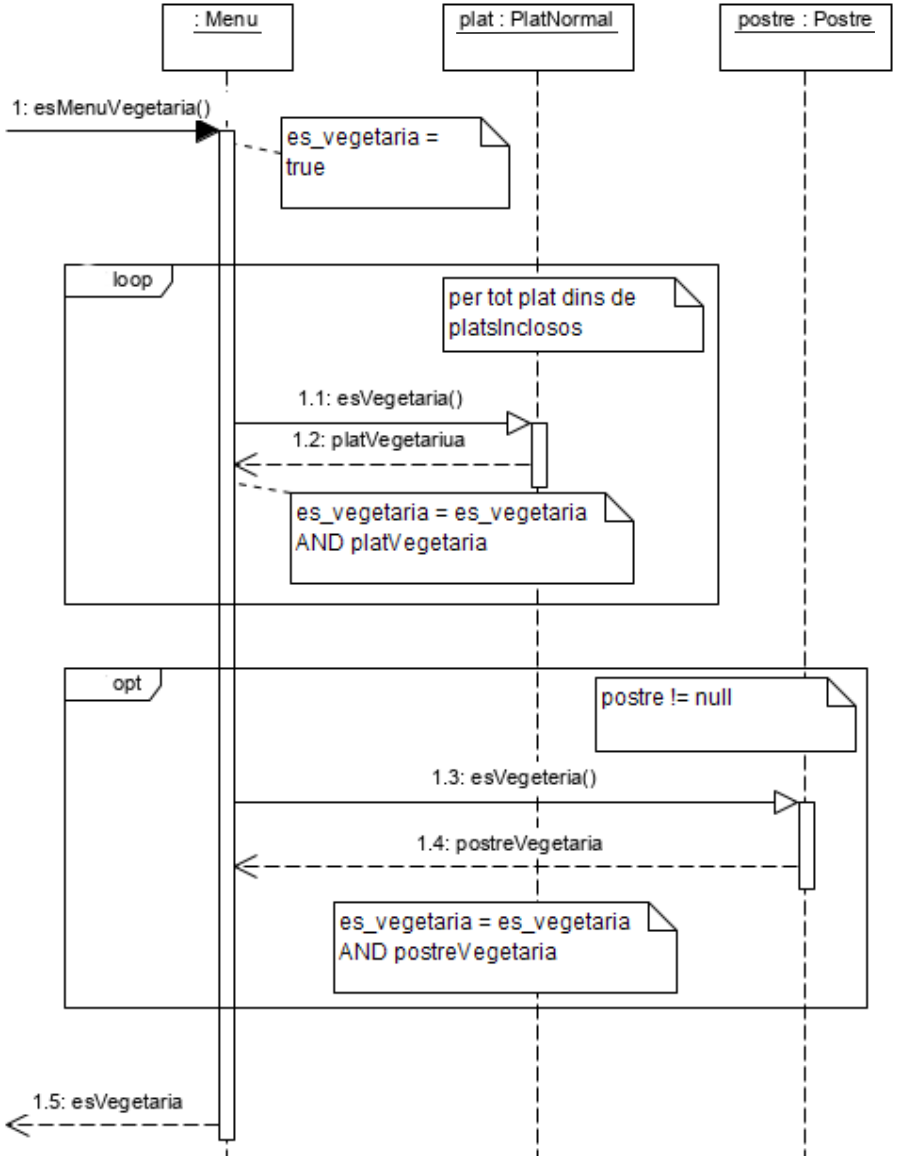
- Per cada Prodcute Ofert, mirem si es tracta d'un Menu Vegetarià amb una operació Abstracta
- Si és un Menu Vegetarià, guardem la seva informació a "llista", però necessitarem una nova operació abstracta: ObtéEstalvi()

Beguda/Plat::EsMenuVegetaria()



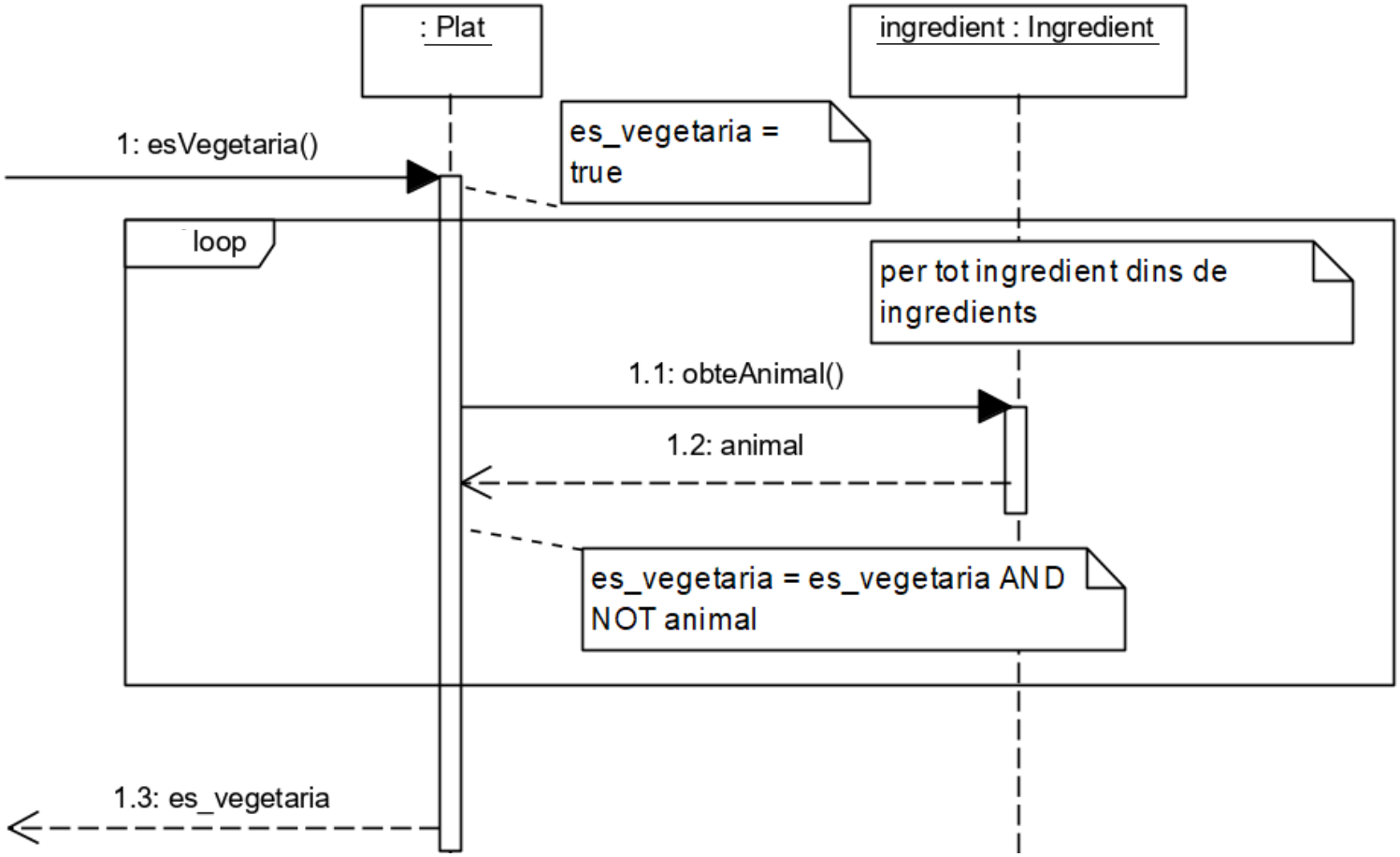
- Beguda i Plat implementen EsMenuVegetarià retornant “fals”, ja que cap dels dos és un Menú

Menu::EsMenuVegetarià



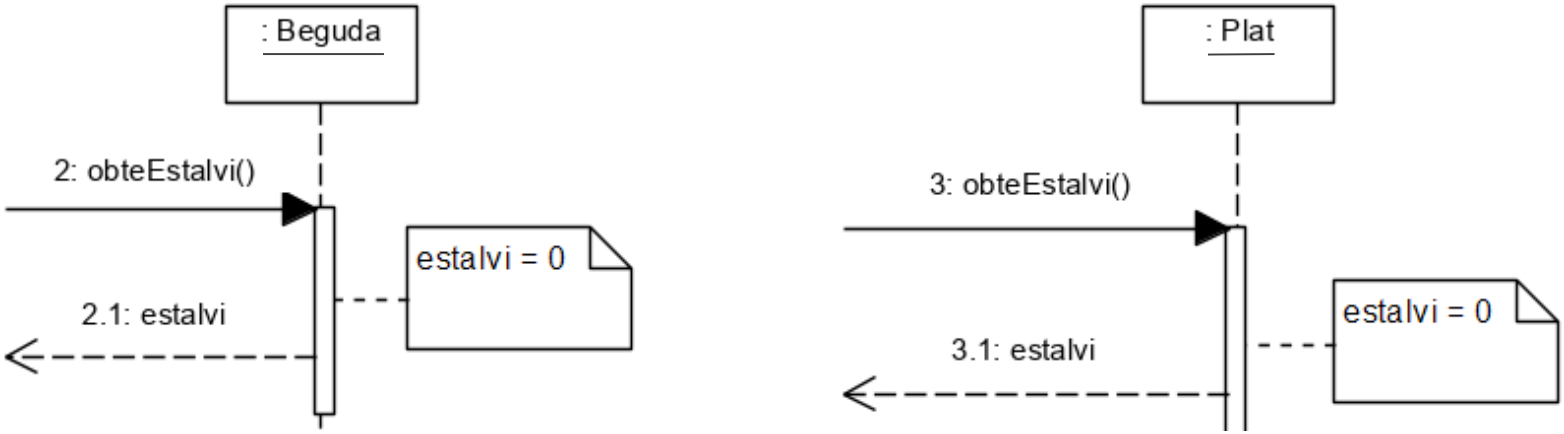
- Iterem per tots els platsInclosos i mirem si tots són vegetarians
- Mirem si el postre és vegetarià també
- L'operació nova EsVegetarià és comuna a *Postre* i *PlatNormal*, així que la posarem a *Plat*, per tal de reaprofitar-la en els dos casos

Plat::EsVegetaria()



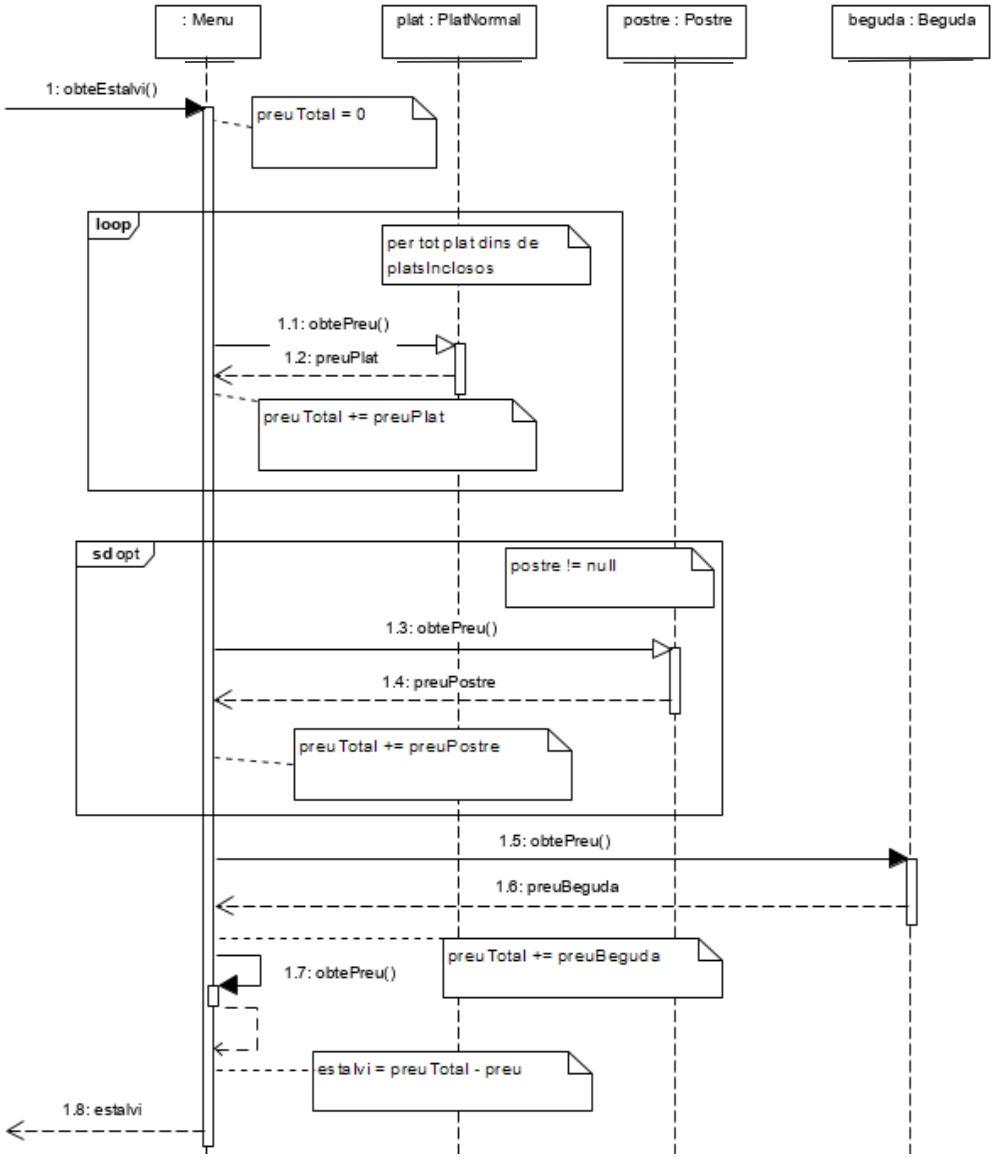
- Mirem per tots els ingredients i garantim que cap no és d'origen animal

Beguda/Plat::ObteEstalvi()



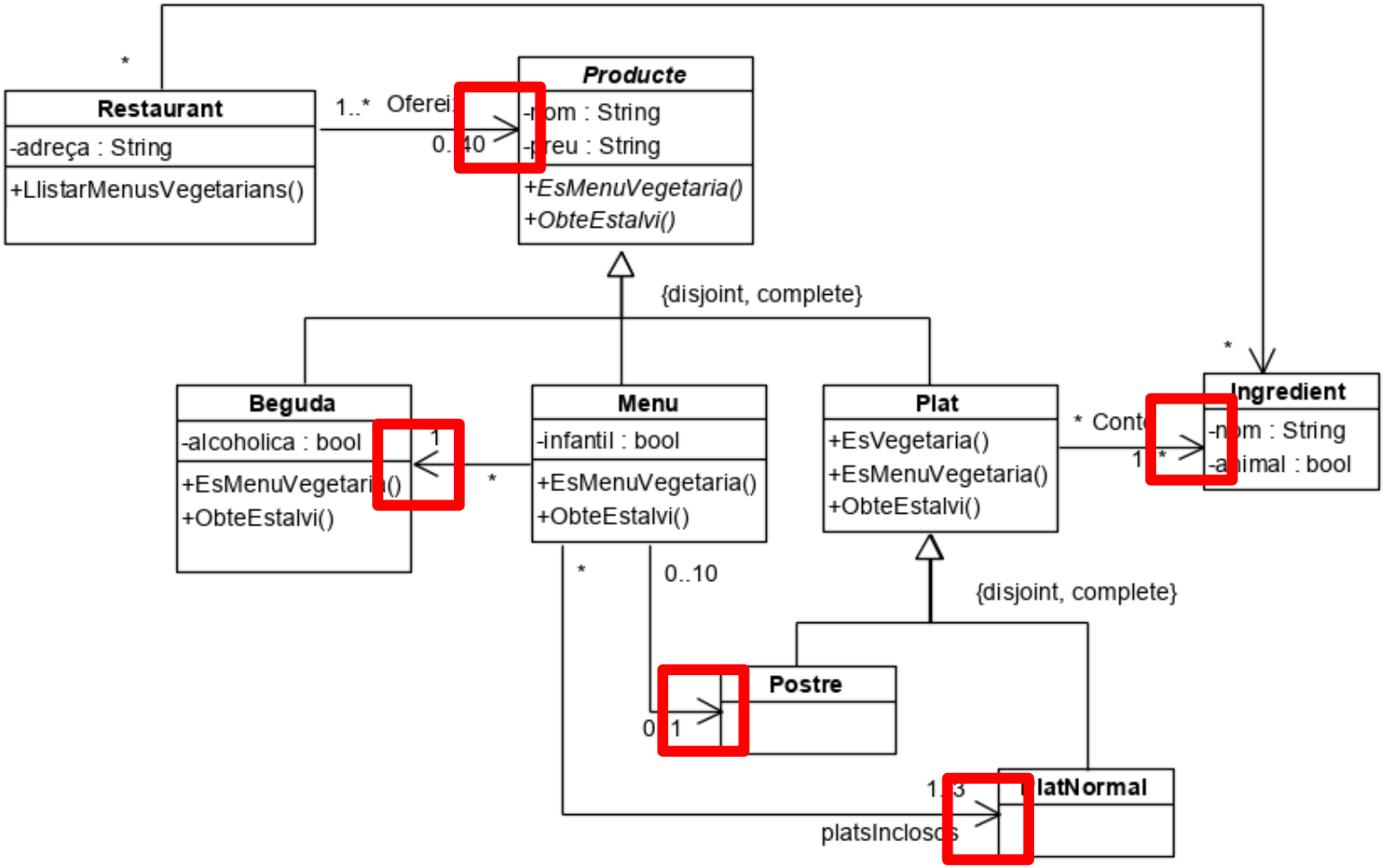
- L'Estalvi és un concepte que només aplica als Menus, així que Plats i Begudes tenen un estalvi de 0

Menu::ObteEstalvi



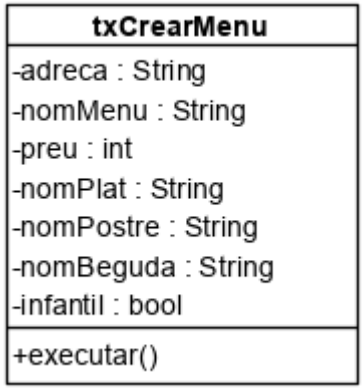
- Obtenim el Preu de tots els platsInclosos i el sumem
- Sumem també el preu del postre
- Sumem també el preu de la beguda
- L'Estalvi serà el preu sumat fins ara menys el preu del menú

Noves Navegabilitats i Operacions



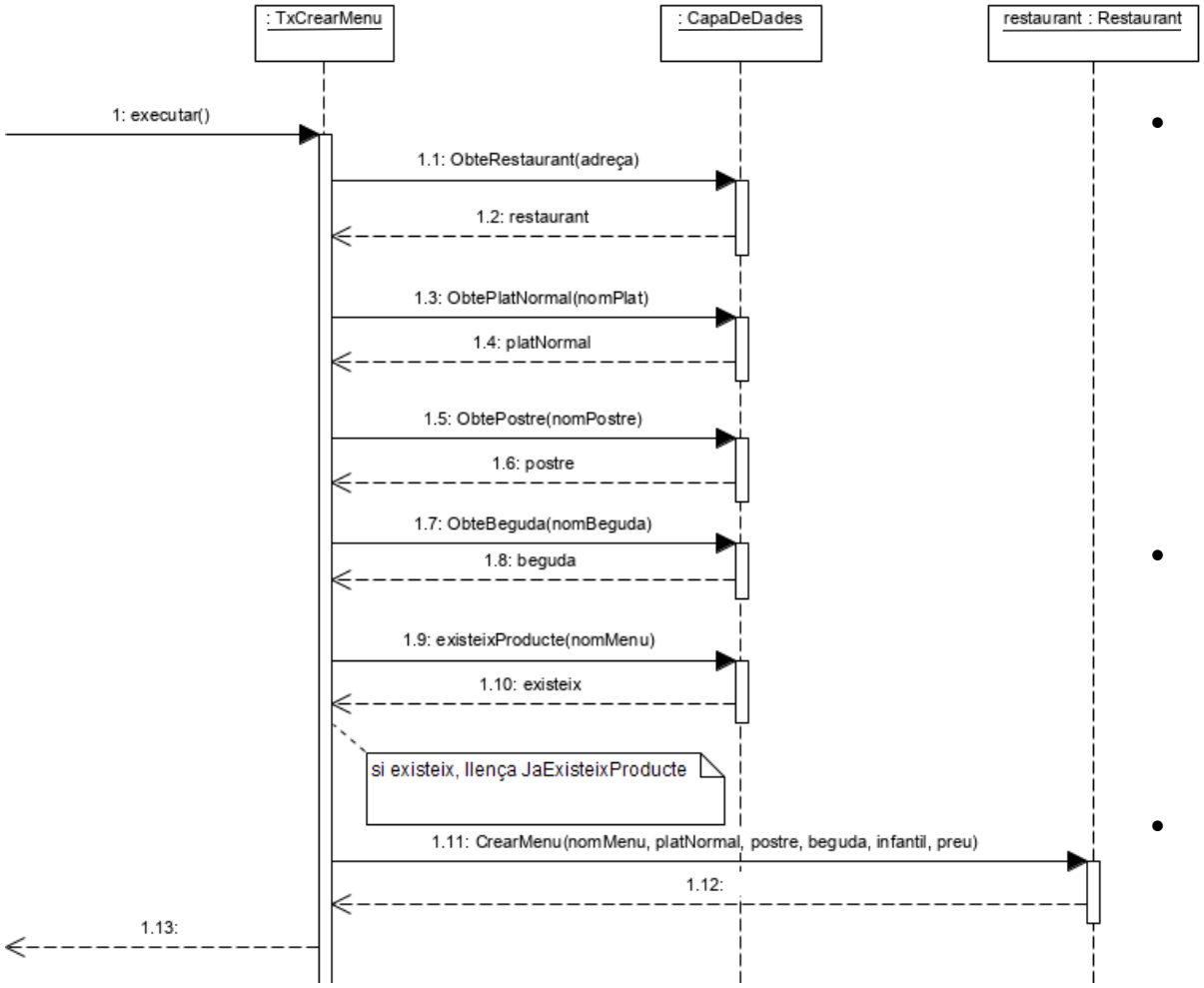
Solució CrearMenu

CrearMenu



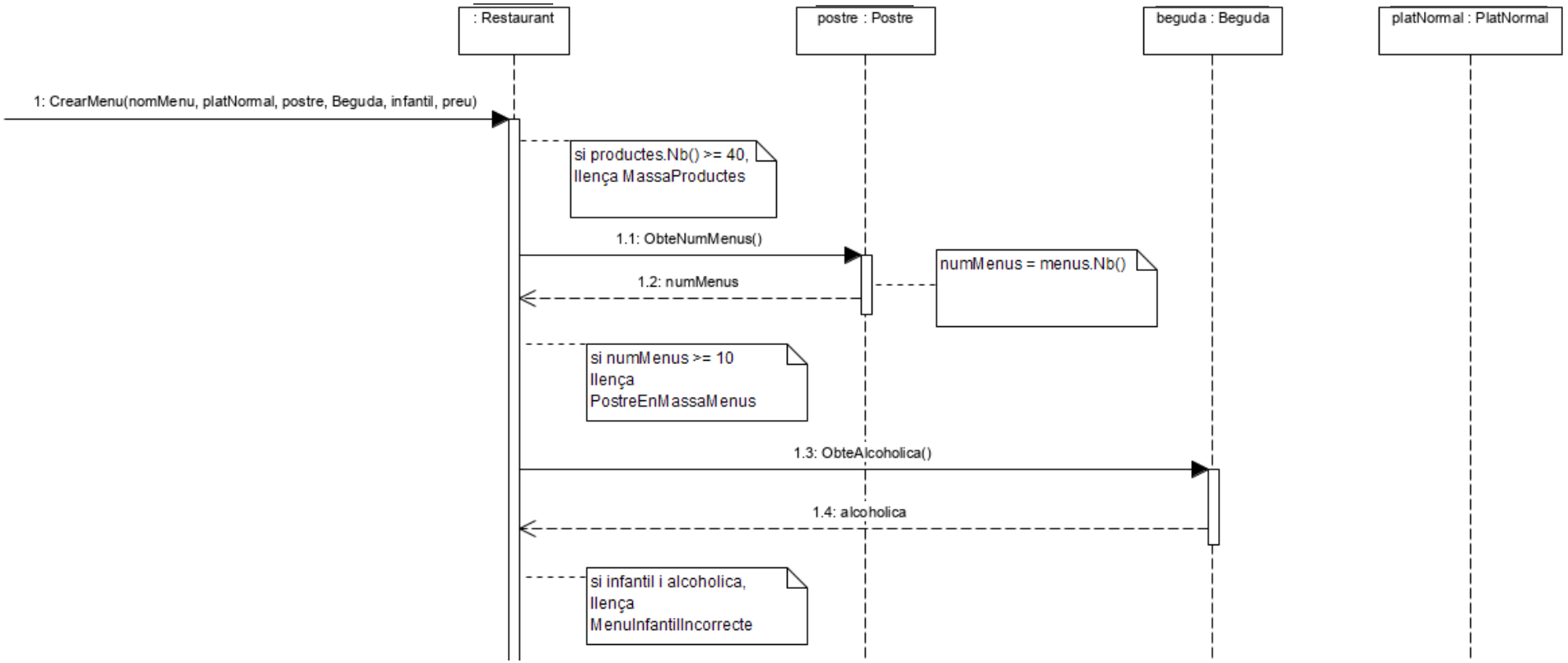
- Creem una clase que segueix el *Controlador Transacció* que conté
  - Un atribut per cada paràmetre de la operación
  - Una operación *Executar*()
- En aquest cas no cal resultat perquè l'operació no retorna res

Executar



- Primer obtenim la informació, cosa que llençarà les excepcions:
  - [NoExisteixRestaurant]
  - [NoExisteixPlatNormal]
  - [NoExisteixPostre]
  - [NoExisteixBeguda]
- Després comprovem si el producte existeix, si ho fa, llencem
  - [JaExisteixProducte]
- Finalment deleguem a Restaurant

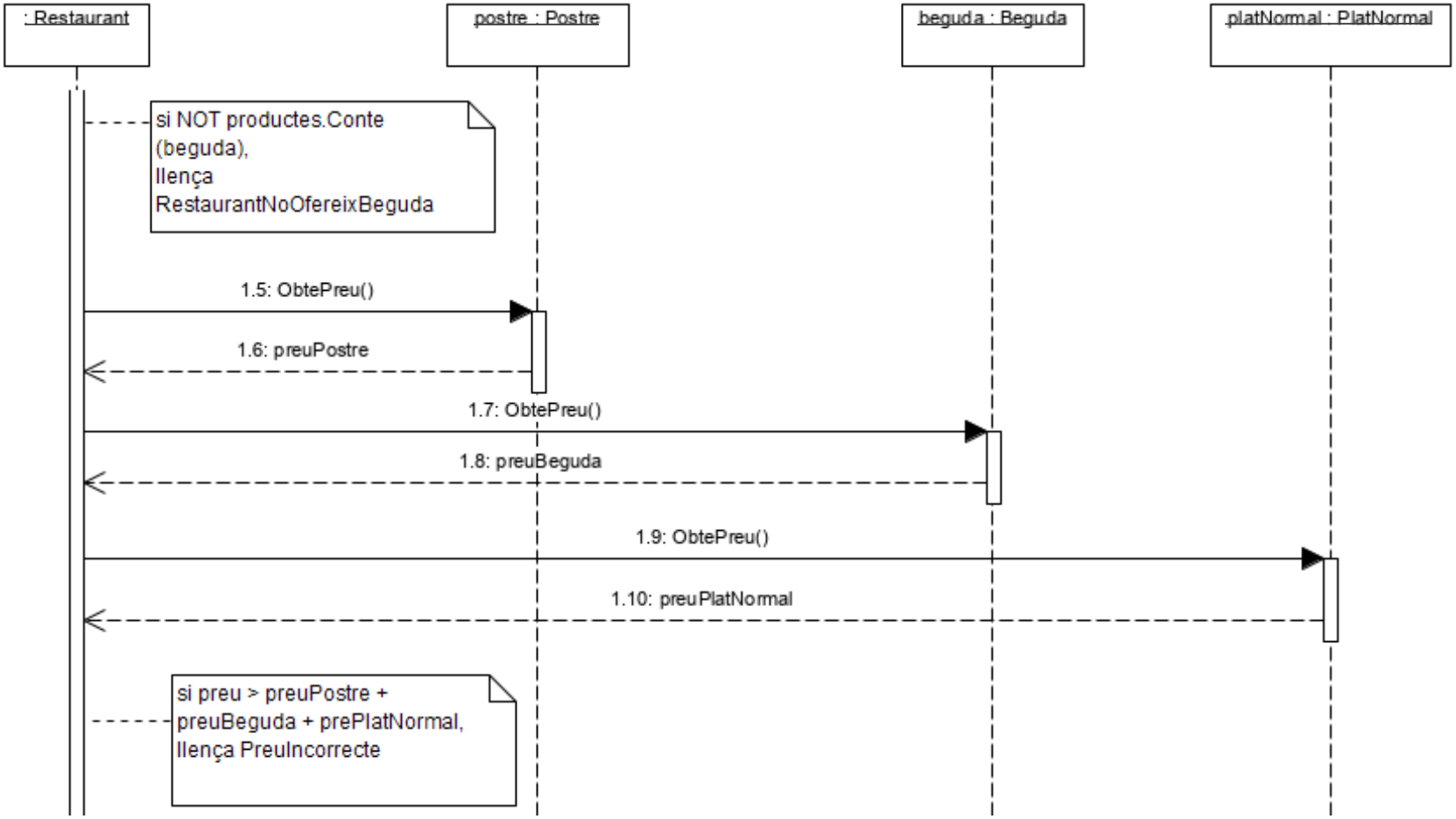
Menu::CrearMenu (Part 1)



- Fem les comprovacions de
  - [MassaProductes]
  - [PostreEnMassaMenús]
  - [MenúInfantilIncorrecte]

- Necessitarem una nova operació a Postre: ObteNumMenus

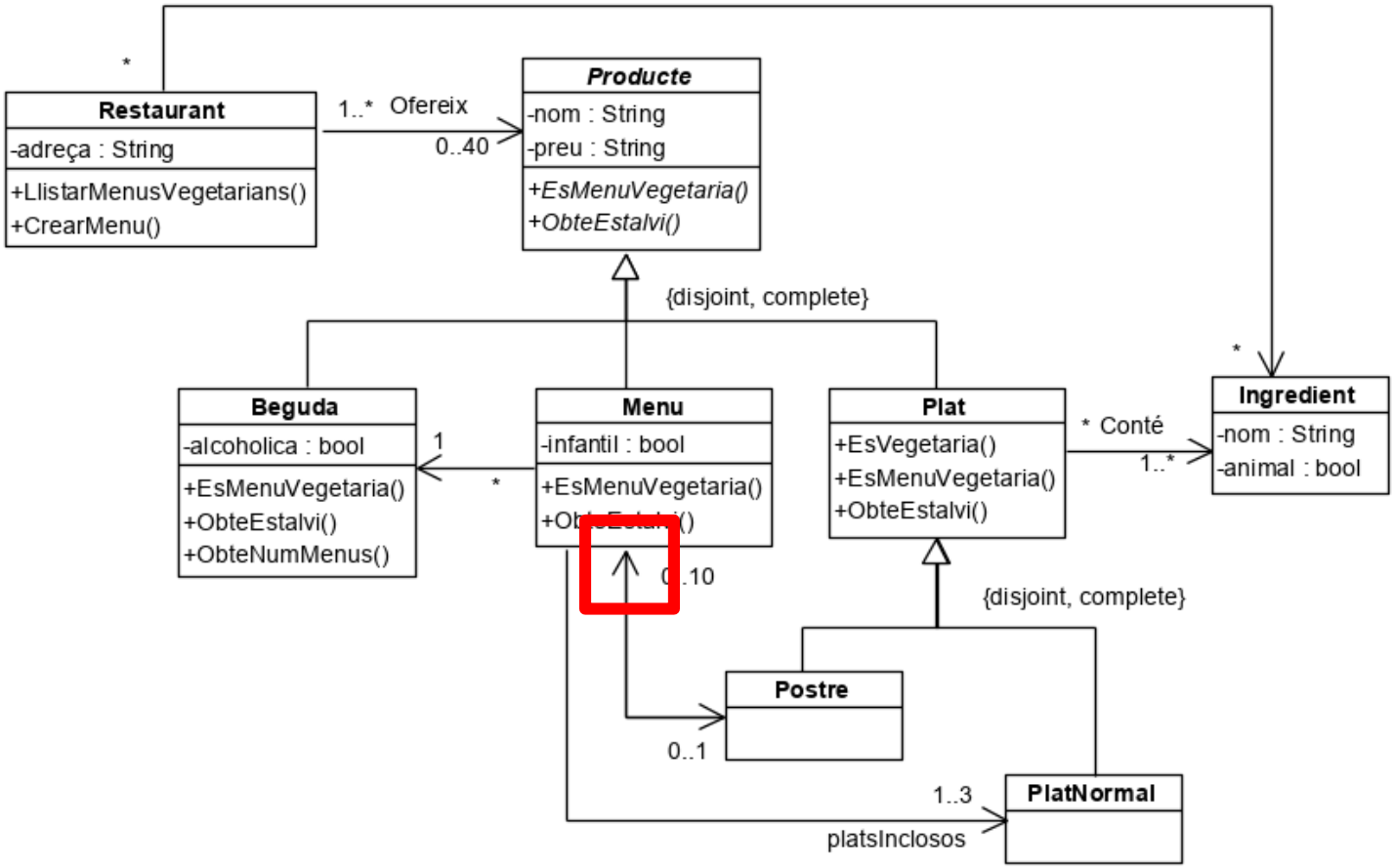
Menu::CrearMenu (Part 2)



- Comprovem les excepcions:
  - [RestaurantNoOfereixBeguda]
  - [PreuIncorrecte]

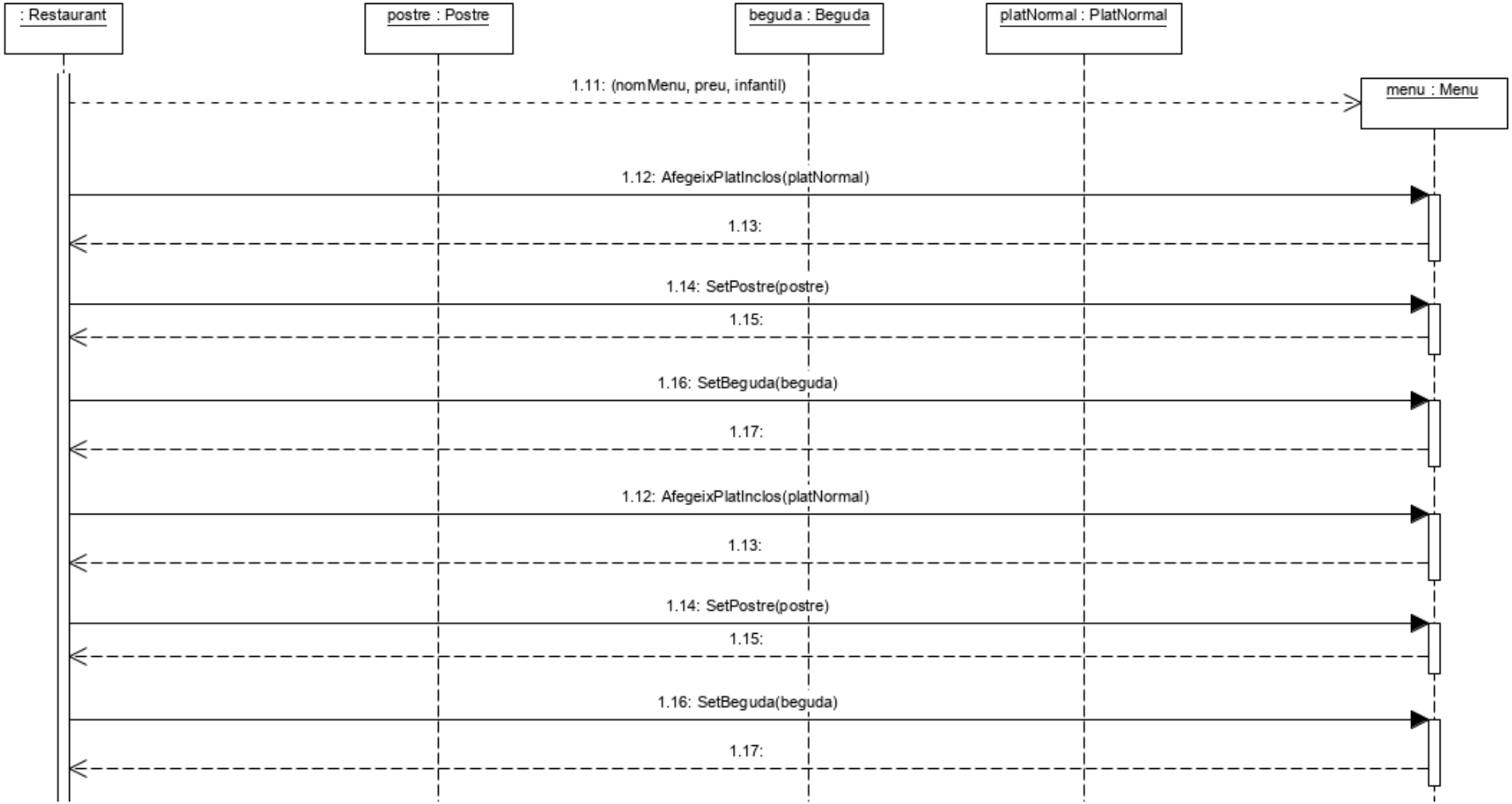
- Ara ja podem crear l'objecte nou

Alerta! Nova navegabilitat!



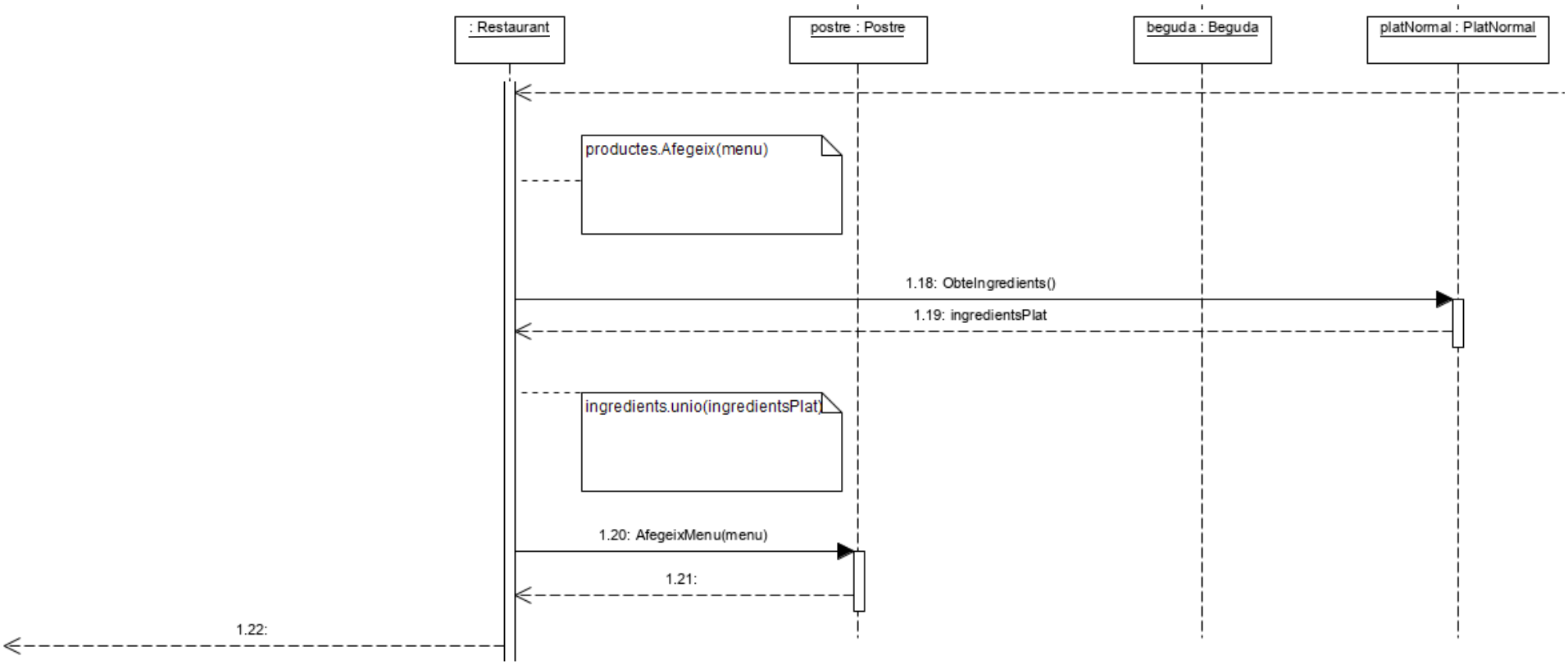
- Abans de Crear la instància nova, cal fer una repassada a possibles navegabilitats noves afegides.
- Hem afegit una nova navegabilitat entre Postre i Menu que s'ha de tenir en compte.

Menu::CrearMenu (Part 3)



- Creem el nou Menu i el relacionem amb totes les instàncies que ens indiquen les navegabilitats

Menu::CrearMenu (Part 4)



- Relacionem el Menú amb aquelles instàncies que tenen navegabilitat cap a Menú
  - Restaurant (via producte)
  - Postre
- Actalitzem la llista d'Ingredients
- Retornem

Problema de disseny en UML: Cadena de Restaurants