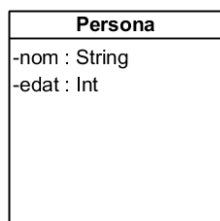


De l'Especificació al Disseny: Diagrama de Classes

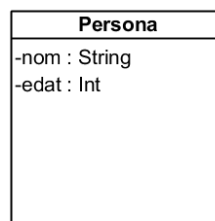
Quina diferència hi ha?

Especificació



- És la representació d'una **persona real**, de carn i ossos.

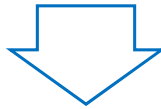
Disseny



- És la representació d'una **classe de codi**.

De l'Especificació al Disseny

- Al disseny hi tenim **components de software** i no conceptes de domini.
- Limitació tecnològica
 - No podem implementar directament tots els conceptes que hem usat a l'especificació



- Cal una **transformació** prèvia dels diagrames d'especificació
 - Obtenció del diagrama de classes de disseny:
 - Eliminar elements no compatibles amb la tecnologia
 - Obtenció dels contractes de disseny de les operacions
 - Controlar les restriccions d'integritat
 - Tractar la informació derivada.

3

Transformació del diagrama de classes

- El diagrama de classes d'especificació és més expressiu que el diagrama de classes de disseny, per tant, per caldrà:
 - Mantenir aquells elements que siguin compatibles:
 - Classes no Associatives
 - Generalitzacions/Especialitzacions Disjoint.
 - Associacions binàries
 - Atributs
 - Eliminar aquells elements que no siguin compatibles i reemplaçar-los per d'altres que sí que ho siguin:
 - Classes Associatives
 - Associacions N-àries
 - Altres casos de Generalització / Especialització
 - Classes Especials (com Data, Hora, ...)
 - Afegir aquells conceptes de disseny que no són necessaris a especificació.
 - Visibilitat
 - Àmbit

4

Elements Compatibles

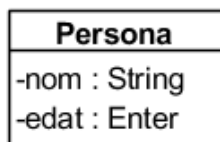
- Classes no associatives
- Generalitzacions/especialitzacions disjoint.
- Associacions binàries
- Atributs

5

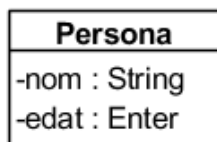
Classes no associatives

- Les classes d'especificació són compatibles, per tant, es mantenen a disseny amb la mateixa forma.

Especificació



Disseny



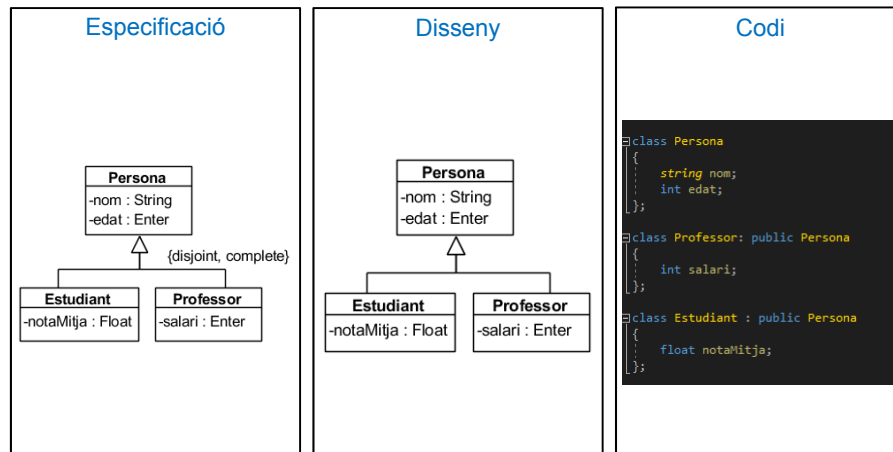
Codi (C++)

```
class Persona
{
    string nom;
    int edat;
};
```

6

Generalitzacions/Especialitzacions disjoint

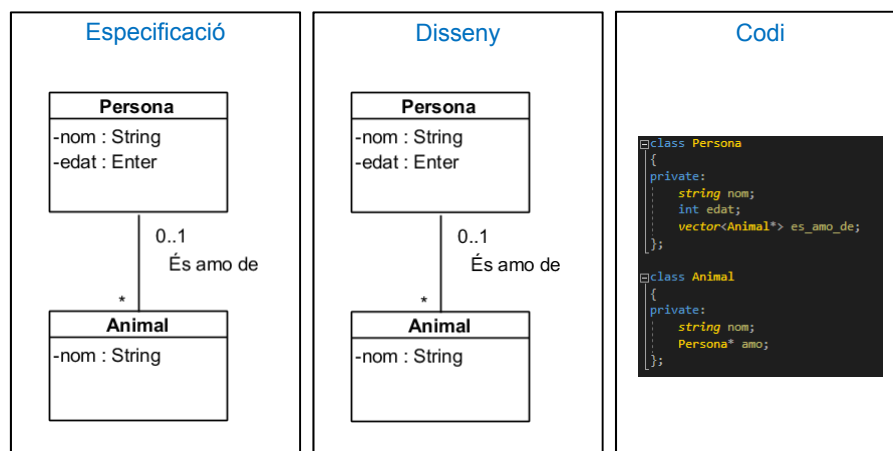
- Les herències disjoint són compatibles, per tant es mantenen igual



7

Associacions binàries

- Les associacions binàries es mantenen igual, tot i que a disseny els haurem d'afegir **navegabilitat** (al proper tema).



8

Elements No compatibles

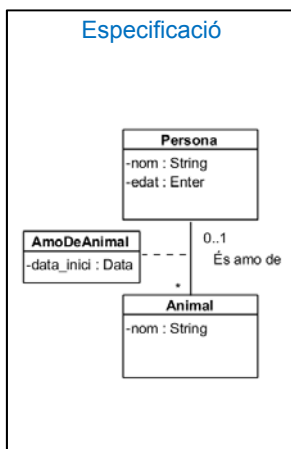
- Classes associatives
- Associacions N-àries
- Altres casos de generalització / especialització
- Classes especials (com Data, Hora, ...)

9

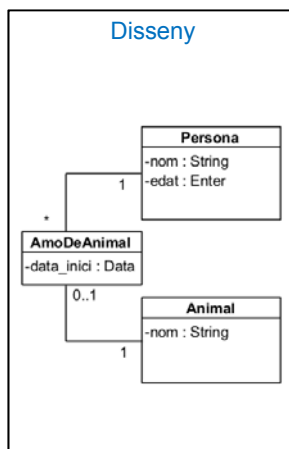
Classes Associatives (I)

- Les classes associatives es converteixen en classes no associatives i es relacionen amb les dues originals...

Especificació



Disseny



Codi*

```
class Persona
{
private:
    string nom;
    int edat;
    vector<AmoDeAnimal> es_amo_de;
};

class Animal
{
private:
    string nom;
    AmoDeAnimal* amo;
};

class AmoDeAnimal
{
private:
    Data data_inici;
    Animal& animal;
    Persona& persona;
};
```

10

Classes Associatives (II)

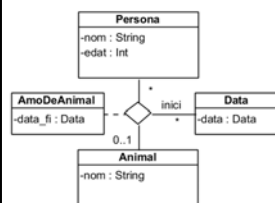
- El diagrama obtingut ha de tenir la mateixa semàntica que l'original, per tant, haurem d'afegir restriccions textuais per a compensar la pèrdua d'informació.
 - No hi pot haver dos "AmoDeAnimal" amb els mateixos Persona i Animal

11

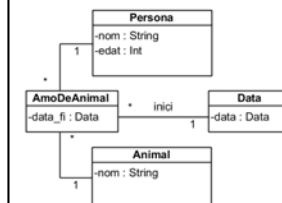
Associacions N-àries (I)

- Les associacions N-àries s'han de canviar per una classe associativa relacionada amb les N originals...

Especificació



Disseny



Codi*

```
class Persona
{
private:
    string nom;
    int edat;
    vector<AmoDeAnimal*> es_amo_de;
};

class Animal
{
private:
    string nom;
    AmoDeAnimal* amo;
};

class AmoDeAnimal
{
private:
    Data inici;
    Data data_fi;
    Animal& animal;
    Persona& persona;
};
```

12

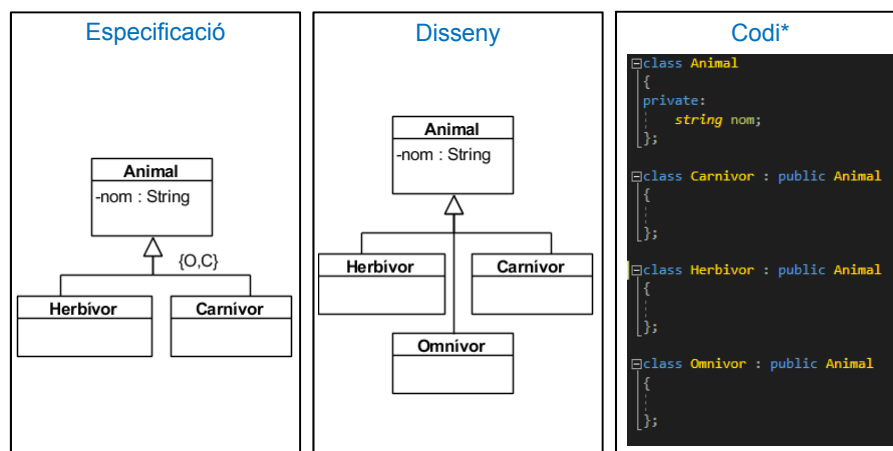
Associacions N-Àries(II)

- El diagrama obtingut ha de tenir la mateixa semàntica que l'original, per tant, haurem d'afegir restriccions textuais per a compensar la pèrdua d'informació (igual que amb les classes associatives).
 - RT1: No hi pot haver dos "AmoDeAnimal" amb els mateixos Persona, Animal i Inici
- També s'han de considerar les multiplicitats de cada un dels membres de la N-ària per a afegir noves restriccions
 - RT2: Donada una persona i una Data, màxim pot esdevenir amo d'un Animal
- I això pot provocar que algunes restriccions textuais siguin redundants.
 - RT1 és redundant amb RT2, per tant, no s'ha d'afegir RT1

13

Generalització / Especialització No Disjoint

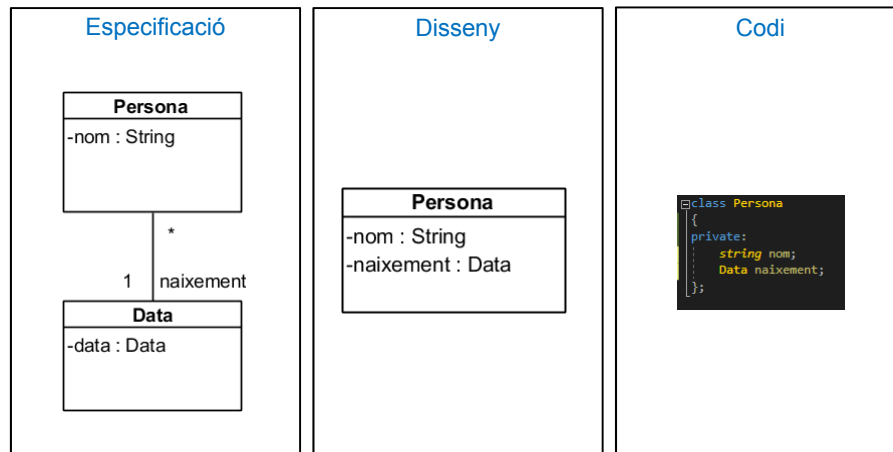
- No hi ha una forma única de fer-ho depèn de cada situació. El producte cartesà és només una de les opcions, però n'hi ha més.



14

Classes Especials (Data, Hora, ...)

- Les classes especials esdevenen atributs amb el tipus corresponent, per tant, s'ha d'eliminar la classe original i afegir com a un atribut.



15

Elements Exclusius de Disseny

- Visibilitat
- Àmbit

16

Visibilitat

- Defineix quins objectes tenen dret a consultar i modificar informació declarada en un diagrama de classes
- Pot ser de tres tipus
 - Pública (+)
 - Privada (-)
 - Protegida (#)
- Aplica a:
 - Atributs
 - Operacions
 - Rols
- A IES assumirem que, per defecte, els atributs són privats, i les operacions i rols són públiques.

17

Visibilitat - Públic

- Donat un element X d'una classe C
 - Si és públic, qualsevol que vegi C, veurà X

```
class Animal
{
public:
    string nom;
};

void main()
{
    Animal animal;
    animal.nom = "Nix";
}
```

18

Visibilitat - Privat

- Donat un element X d'una classe C
 - Si és privat, només C veurà X

```
class Animal
{
private:
    string nom;
public:
    void CanviaNom(string nouNom)
    {
        nom = nouNom;
    }
};

void main()
{
    Animal animal;
    //animal.nom = "Nix"; //No compila!!
    animal.CanviaNom("Nix");
}
```

19

Visibilitat - Protegit

- Donat un element X d'una classe C
 - Si és privat, només C o els seus descendents veuran X

```
class Animal
{
protected:
    string nom;
};

class Gos: public Animal
{
public:
    void CanviaNom(string nouNom)
    {
        nom = nouNom;
    }
};

void main()
{
    Gos gos;
    //gos.nom = "Nix"; //No compila!!
    gos.CanviaNom("Nix");
}
```

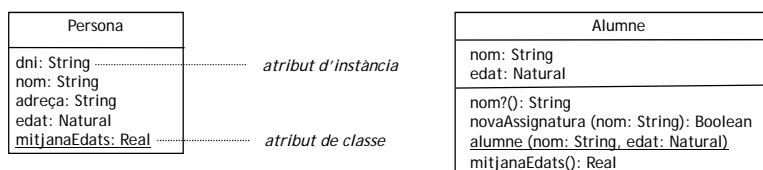
20

Àmbit

- Determina si els atributs o operacions són aplicables a objectes individuals o a la classe que defineix els elements
- Poden ser:
 - De classe (estàtic)
 - X està associat al C
 - D'instància (no estàtic)
 - X està associat als objectes de C
- Es marca subratllant el nom de la operació / atribut

21

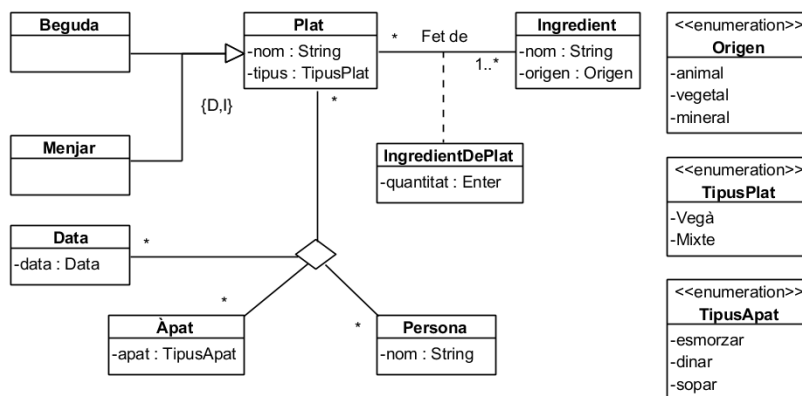
Àmbit - Exemples



22

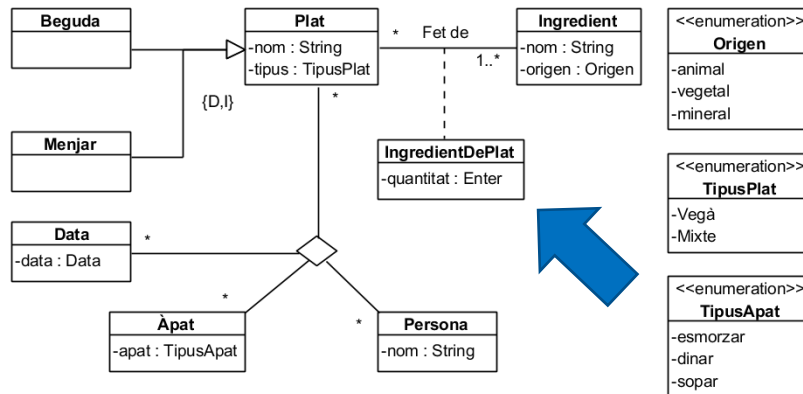
Exemple de traducció

Exemple de traducció



- RT1: Claus Externes (Plat, nom), (Ingredient, nom), (Àpat, data+tipusApat), (Persona, nom)
- RT2: Un plat vegà no pot contenir ingredients d'origen Animal

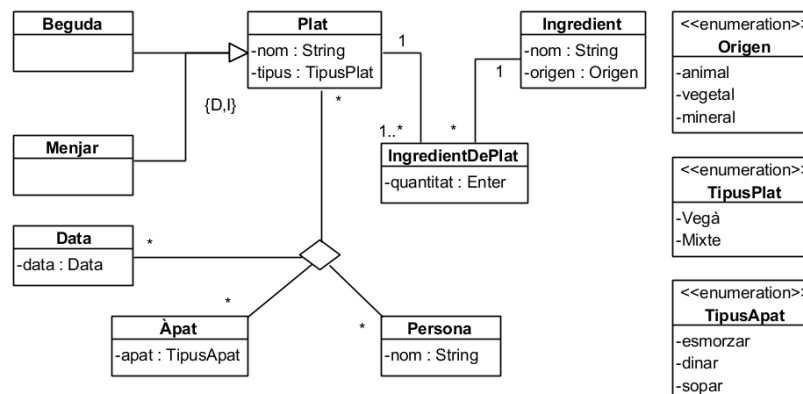
Exemple de traducció



- RT1: Claus Externes (Plat, nom), (Ingredient, nom), (Àpat, data+tipusApat), (Persona, nom)
- RT2: Un plat vegà no pot contenir ingredients d'origen Animal

25

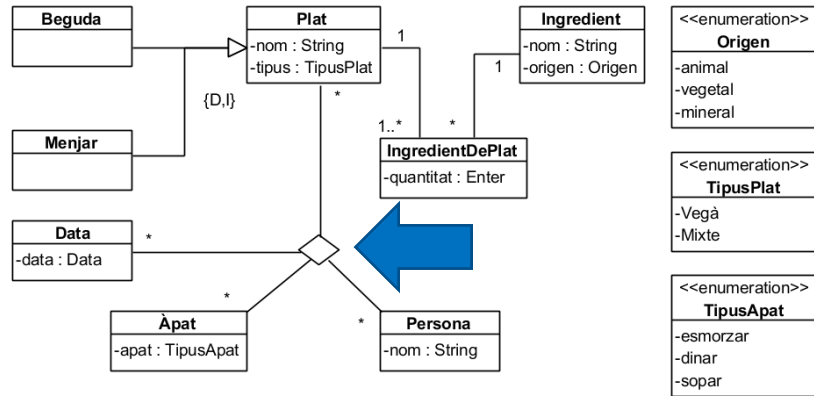
Exemple de traducció



- RT1: Claus Externes (Plat, nom), (Ingredient, nom), (Àpat, data+tipusApat), (Persona, nom)
- RT2: Un plat vegà no pot contenir ingredients d'origen Animal
- **RT3: No hi pot haver 2 IngredientDePlat amb els mateixos ingredient i plat**

26

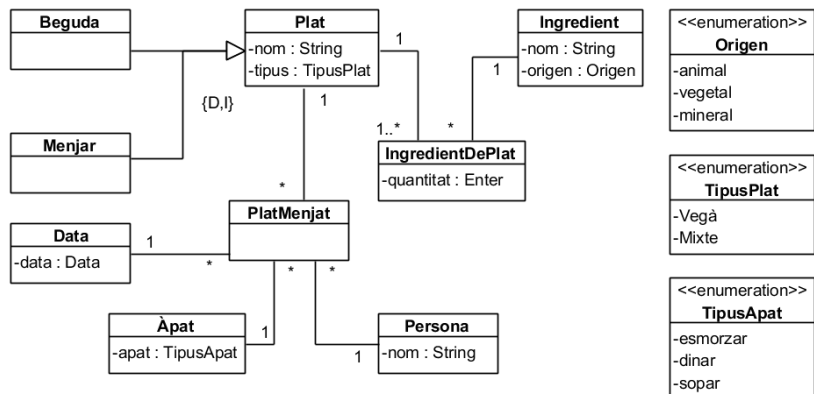
Exemple de traducció



- RT1: Claus Externes (Plat, nom), (Ingredient, nom), (Àpat, data+tipusApat), (Persona, nom)
- RT2: Un plat vegà no pot contenir ingredients d'origen Animal
- RT3: No hi pot haver 2 IngredientDePlat amb els mateixos ingredient i plat

27

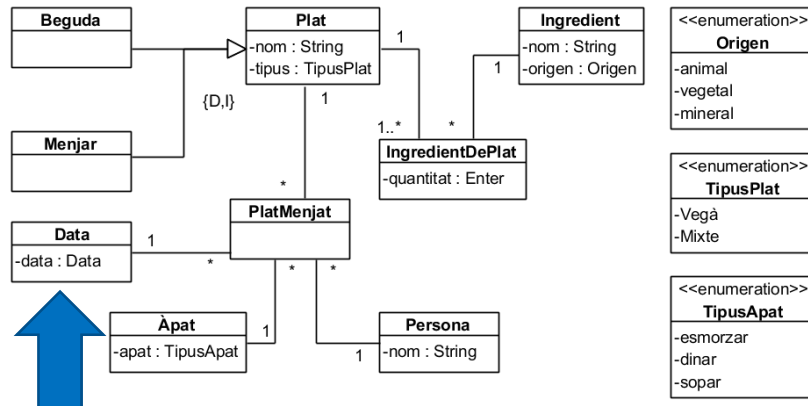
Exemple de traducció



- RT1: Claus Externes (Plat, nom), (Ingredient, nom), (Àpat, data+tipusApat), (Persona, nom)
- RT2: Un plat vegà no pot contenir ingredients d'origen Animal
- RT3: No hi pot haver 2 IngredientDePlat amb els mateixos ingredient i plat
- **RT4: No hi pot haver 2 PlatMenjat amb els mateixos Plat, Àpat, Data i Persona**

28

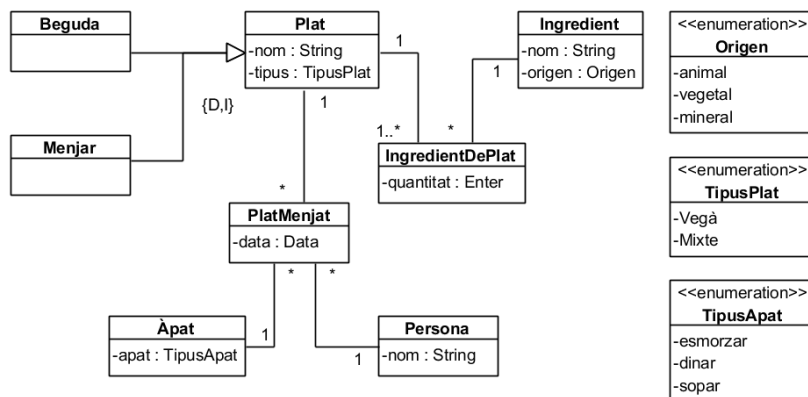
Exemple de traducció



- RT1: Claus Externes (Plat, nom), (Ingredient, nom), (Àpat, data+tipusApat), (Persona, nom)
- RT2: Un plat vegà no pot contenir ingredients d'origen Animal
- RT3: No hi pot haver 2 ingredientDePlat amb els mateixos ingredient i plat
- RT4: No hi pot haver 2 PlatMenjat amb els mateixos Plat, Àpat, Persona i Data

29

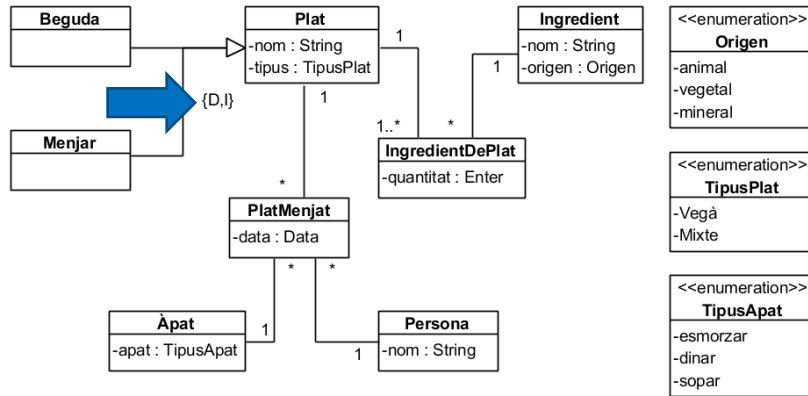
Exemple de traducció



- RT1: Claus Externes (Plat, nom), (Ingredient, nom), (Àpat, data+tipusApat), (Persona, nom)
- RT2: Un plat vegà no pot contenir ingredients d'origen Animal
- RT3: No hi pot haver 2 ingredientDePlat amb els mateixos ingredient i plat
- RT4: No hi pot haver 2 PlatMenjat amb els mateixos Plat, Àpat, Persona i Data

30

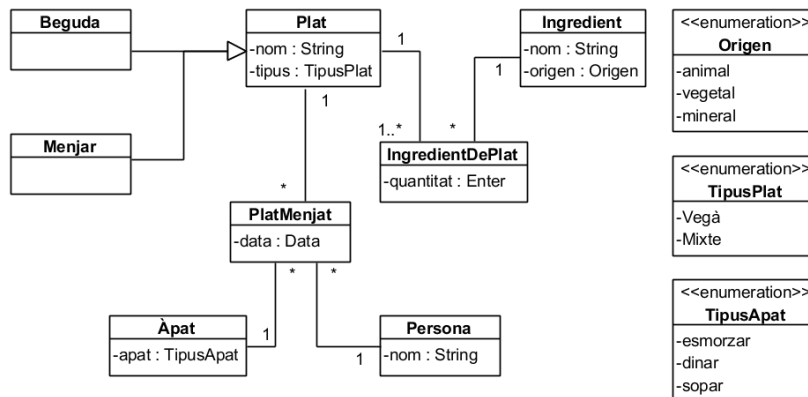
Exemple de traducció



- RT1: Claus Externes (Plat, nom), (Ingredient, nom), (Àpat, data+tipusApat), (Persona, nom)
- RT2: Un plat vegà no pot contenir ingredients d'origen Animal
- RT3: No hi pot haver 2 ingredientDePlat amb els mateixos ingredient i plat
- RT4: No hi pot haver 2 PlatMenjat amb els mateixos Plat, Àpat, Persona i Data

31

Exemple de traducció



- RT1: Claus Externes (Plat, nom), (Ingredient, nom), (Àpat, data+tipusApat), (Persona, nom)
- RT2: Un plat vegà no pot contenir ingredients d'origen Animal
- RT3: No hi pot haver 2 ingredientDePlat amb els mateixos ingredient i plat
- RT4: No hi pot haver 2 PlatMenjat amb els mateixos Plat, Àpat, Persona i Data

32

Bibliografia

- Pressman, R.G. *"Software Engineering. A Practitioner's Approach"*, Mc Graw-Hill, 2015 (8a edició).
- Larman, C. *"Applying UML and Patterns. An Introduction to Object-oriented Analysis and Design"*, Prentice Hall, 2005, (3ª edició)
- Meyer, B. *"Object-oriented Software Construction"*, Prentice Hall, 1997.