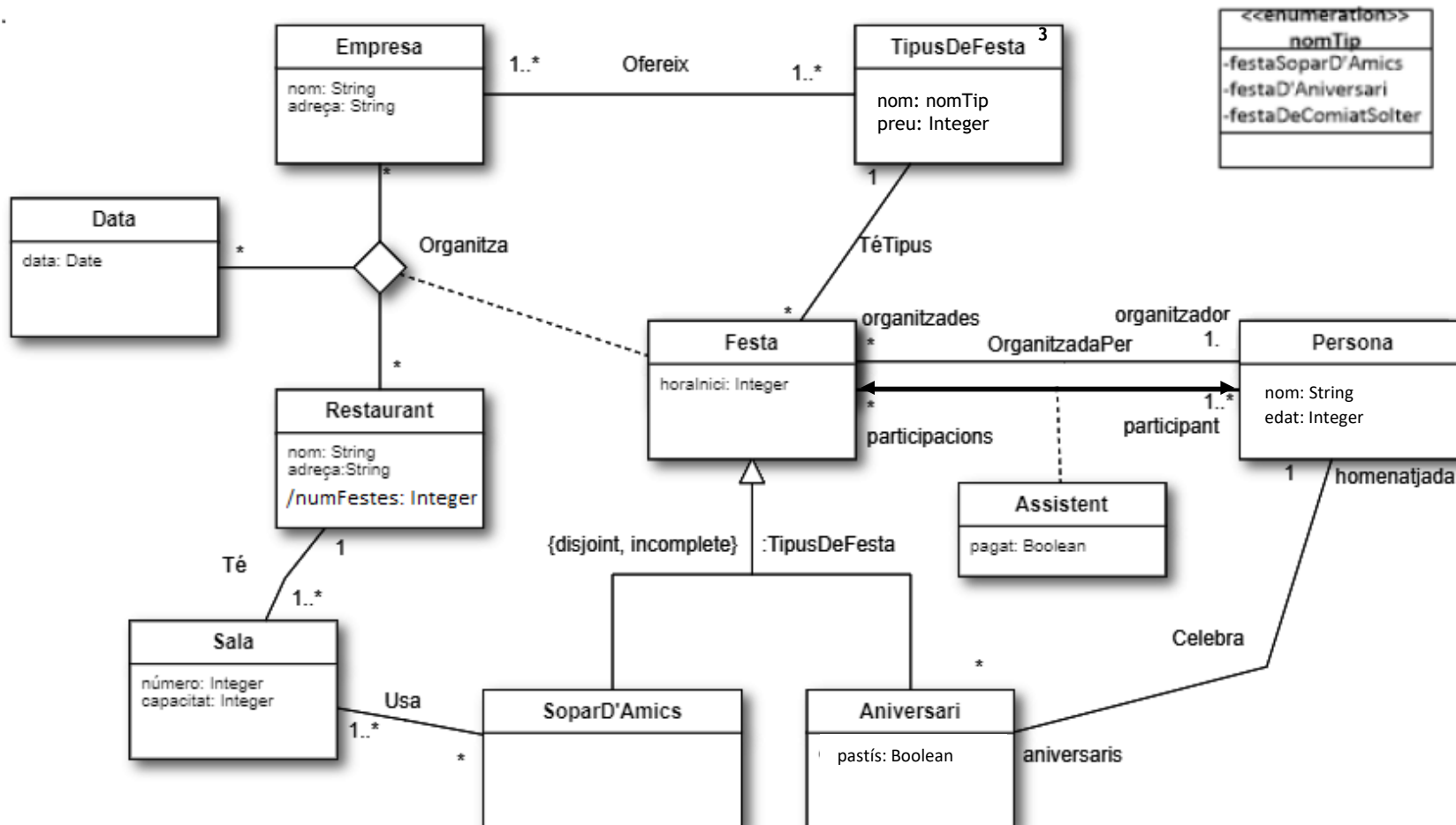


# Problema de disseny en UML: Festes d'aniversari

# Enunciat: esquema conceptual d'especificació



A aquesta solució suposarem que nom de TipusDeFesta és una enumeration

## Enunciat: restriccions textuais

1. Claus externes: (Empresa, nom), (TipusDeFesta,nom), (Restaurant, nom), (Persona, nom)
2. El tipus de festa d'una festa ha de ser ofert per l'empresa que organitza la festa
3. Una persona no pot assistir a dues o més festes que se celebren en la mateixa data i amb idèntica hora d'inici
4. Un restaurant no pot tenir dues o més sales amb el mateix número
5. La capacitat total de les sales usades per una festa sopar d'amics ha de ser superior al nombre de participants a la festa
6. Les sales usades per una festa sopar d'amics han de ser del restaurant on se celebra la festa
7. La persona homenatjada a una festa d'aniversari ha de ser assistent a aquesta festa
8. L'hora de inici de una festa és al rang [1..24]
9. Una empresa no pot organitzar més de 100 festes.

## Enunciat: informació derivada

1. L'atribut númFestes de Restaurant és el nombre de festes organitzades pel restaurant

## Operació 1: *LlistaRestaurants*

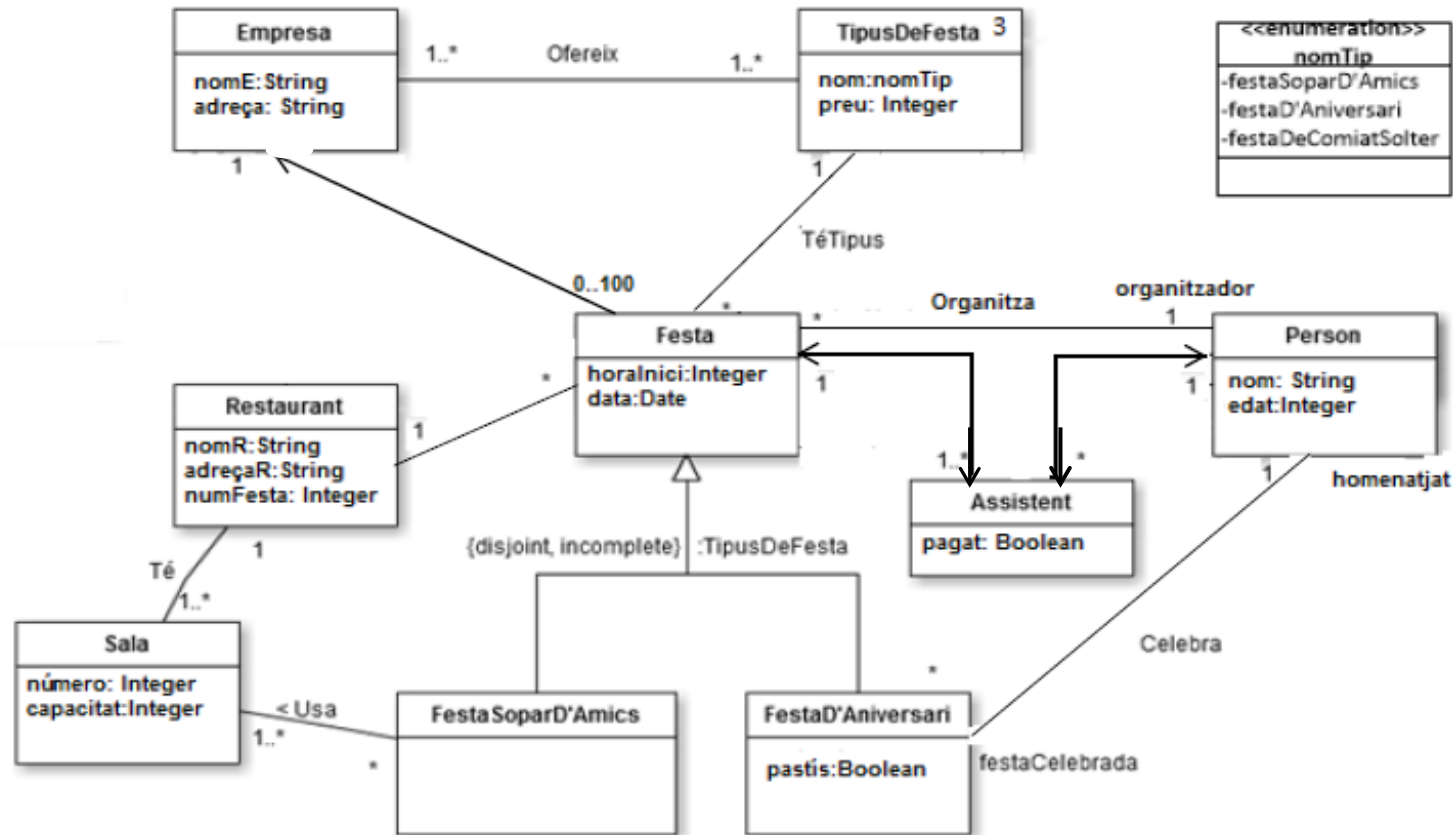
- **Operació:** `llistaRestaurants(nomE: String): Set (String)`
- **Pre:**
  - L'empresa identificada per `nomE` existeix.
- **Post:**
  - Es retornen el noms dels restaurants on l'empresa `nomE` organitza festes d'aniversari amb més de 30 assistents, amb pastís d'aniversari, i on l'homenatjat fa 50 anys i es també l'organitzador de la festa.

## Operació 2: *AltaAniversari*

- **Operació:** `altaAniversari(nomR:String, nomE:String, dt:Date, hl:Integer, pastís:Bool, nomHom:String, nomOrg:String)`
- **Pre:**
  - El restaurant identificat per `nomR` existeix
  - L'empresa identificada per `nomE` existeix
  - La persona identificada per `nomHom` existeix
  - La persona identificada per `nomOrg` existeix
- **Post:**
  - Es dona d'alta una instància de `FestaD'Aniversari` amb els atributs i associacions corresponents.
  - La persona `nomHom` es registra com homenatjat i assistent a la festa (`pagat=false`).
  - Es dona d'alta una instància de l'associació entre festa i la persona organitzadora `nomOrg`.

# Diagrama de classes de disseny i contractes de les operacions de disseny

# Solució: diagrama de classes de disseny



## Restriccions textuais afegides

- No hi pot haver dues festes amb els mateixos Empresa, Restaurant i Data
- ~~No hi pot haver dos assistents amb els mateixos Festa i Persona~~  
(Redundant amb RT3 “Una persona no pot assistir a dues o més festes que se celebren ...”)
- S’elimina la RT9 de l’enunciat

## Solució: contracte de disseny de *llistaRestaurants*

**Context** CapaDomini::llistaRest(nomE: String): Set (String)

**excepcions:**

[*EmpresaNoExisteix*]: L'empresa identificada per *nomE* no existeix

**post:** Es retornen el noms dels restaurants on l'empresa *nomE* organitza festes d'aniversari amb més de 30 assistents, amb pastís d'aniversari, i on l'homenatjat fa 50 anys i es també l'organitzador de la festa

## Solució: contracte de disseny d'*altaAniversari*

**Context** **CapaDomini::***altaAniversari*(*nomR*:String, *nomE*:String, *dt*:Date, *hI*:Integer, *pastís*:Bool, *nomHom*:String, *nomOrg*:String)

**excepcions:**

[*RestaurantNoExisteix*]: El restaurant identificat per *nomR* no existeix

[*EmpresaNoExisteix*]: L'empresa identificada per *nomE* no existeix

[*HomenatjatNoExisteix*]: La persona identificada per *nomHom* no existeix

[*OrganitzadorNoExisteix*]: La persona identificada per *nomOrg* no existeix

[*FestaJaExisteix*]: La festa identificada per *nomR*, *nomE* i *dt* ja existeix (clau externa de Festa)

[*HoraIniciIncorrecta*]: L'hora inici *hI* no és al rang [1..24] (RT8)

[*EmpresaJaTé100Festes*]: L'empresa *nomE* ja té 100 festes (RT9 - multiplicitat a disseny)

[*TipusFestaNoEsOfertPerEmpresa*]: l'empresa *nomE* no ofereix festaD'Aniversari (RT2)

[*HomenatjatTeAltraFesta*]: La persona homenatjada *nomHom* té una altra festa en la mateixa data *dt* i hora inici *hI* (RT3)

**post:**

[*altaFestaAniversari*] Es dona d'alta una instància de FestaD'Aniversari amb els atributs i les associacions corresponents.

[*registraHomenatjatAssistent*] La persona *nomHom* es registra com homenatjat i assistent a la festa (amb atribut pagat=False).

[*registraOrganitzador*] Es dona d'alta una instància de l'associació entre festa i la persona organitzadora *nomOrg*.

[*incrementaNumFesta*] S'incrementa en una unitat l'atribut *numFesta* del restaurant *nomR*.



# Diagrama de seqüència: *LlistaRestaurants*

## Solució: contracte de disseny de *llistaRestaurants*

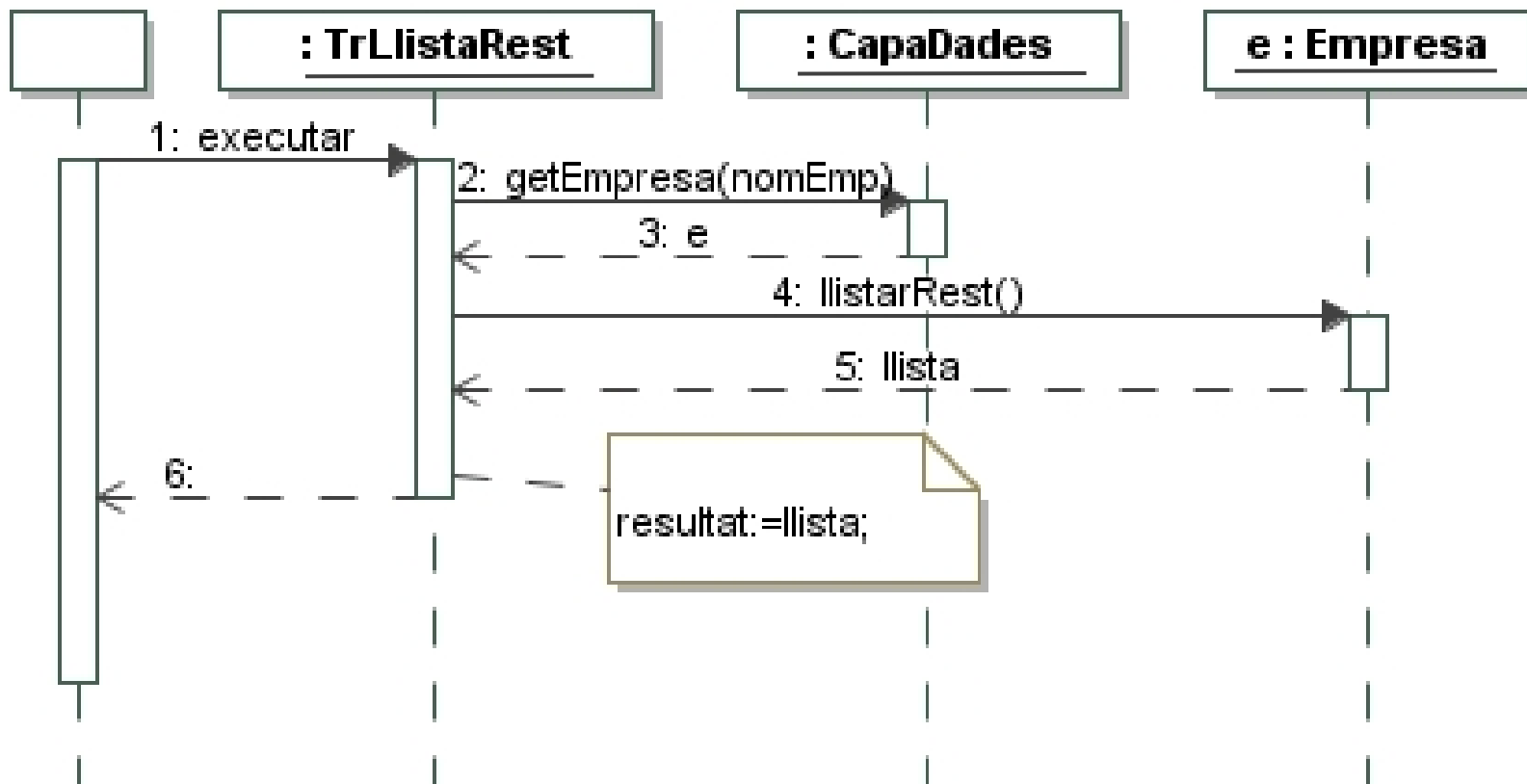
**Context** CapaDomini::llistaRest(nomE: String): Set (String)

**excepcions:**

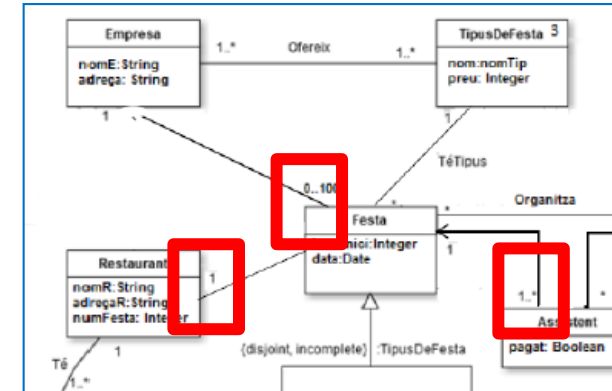
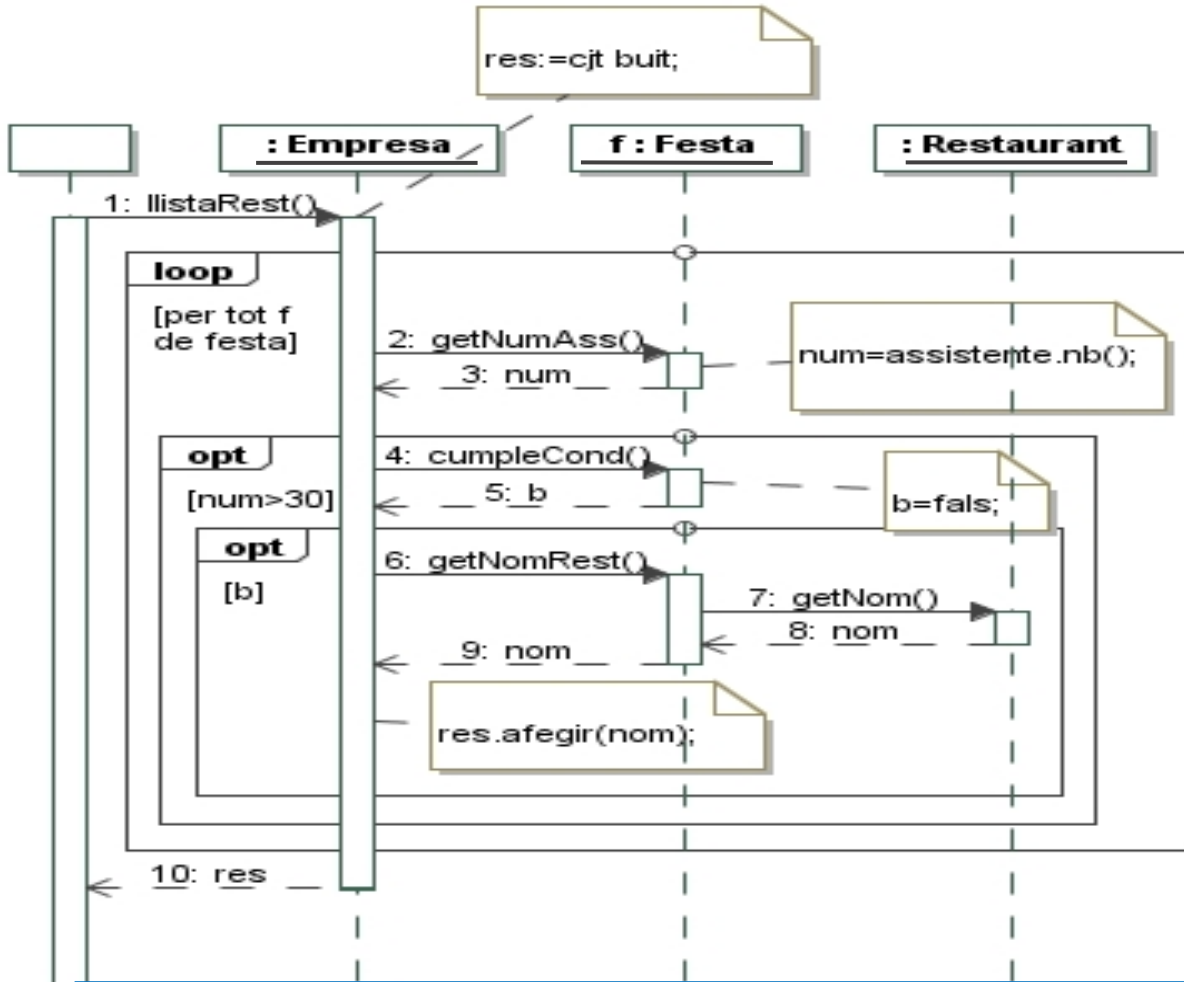
[*EmpresaNoExisteix*]: L'empresa identificada per *nomE* no existeix

**post:** Es retornen el noms dels restaurants on l'empresa *nomE* organitza festes d'aniversari amb més de 30 assistents, amb pastís d'aniversari, i on l'homenatjat fa 50 anys i es també l'organitzador de la festa

# Executar

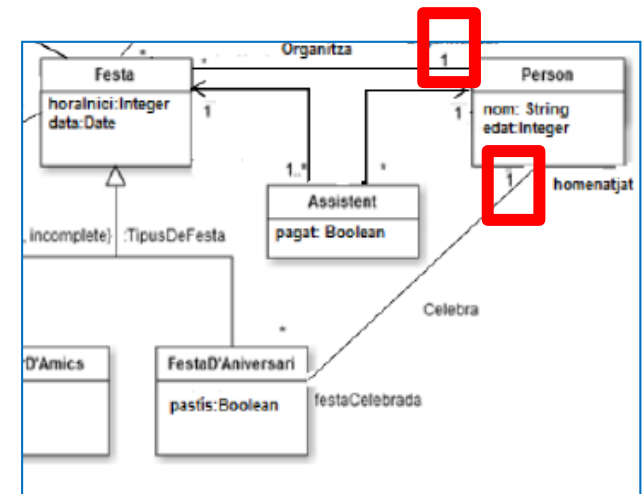


- Obtenim l'empresa de la Capa de Dades.
  - Això pot llençar l'excepció [EmpresaNoExisteix]
- Deleguem la resta de la operació a la classe Empresa
- Ens guardem el resultat a l'atribut resultat del controlador transacció



- Per cada Festa, obtenim el seu número d'assistents
- Si en té més de 30, mirem si satisfà la resta de condicions (*operació ganxo*)
- Si les satisfà, obtenim el nom del restaurant i l'afegim a la llista

## Facultat d'Informàtica de Barcelona



- 13



# Diagrama de seqüència: altaAniversari

# Aplicació del *patró Expert*

**Context** **CapaDomini::**altaAniversari(nomR:String, nomE:String, dt:Date, hI:Integer, pastís:Bool, nomHom:String, nomOrg:String)

**excepcions:**

[*RestaurantNoExisteix*]: El restaurant identificat per *nomR* no existeix

[*EmpresaNoExisteix*]: L'empresa identificada per *nomE* no existeix

[*HomenatjatNoExisteix*]: La persona identificada per *nomHom* no existeix

[*OrganitzadorNoExisteix*]: La persona identificada per *nomOrg* no existeix

[*FestaJaExisteix*]: La festa identificada per *nomR*, *nomE* i *dt* ja existeix

[*HoraIniciIncorrecta*]: L'hora inici *hI* no és al rang [1..24]

[*EmpresaJaTé100Festes*]: L'empresa *nomE* ja té 100 festes

[*TipusFestaNoEsOfertPerEmpresa*]: l'empresa *nomE* no ofereix festa d'Aniversari

[*HomenatjatTeAltraFesta*]: La persona homenatjada *nomHom* té una altra festa en la mateixa data *dt* i hora inici *hI*

**post:**

[*altaFestaAniversari*] Es dona d'alta una instància de Festa d'Aniversari amb els atributs i les associacions corresponents.

[*registraHomenatjatAssistent*] La persona *nomHom* es registra com homenatjat i assistent a la festa (amb atribut pagat=False).

[*registraOrganitzador*] Es dona d'alta una instància de l'associació entre festa i la persona organitzadora *nomOrg*.

[*incrementaNumFesta*] S'incrementa en una unitat l'atribut *numFesta* del restaurant *nomR*.

Capa de Dades

Capa de Dades

Capa de Dades

Capa de Dades

Capa de Dades

Paràmetres (transacció)

Expert

Empresa

Empresa

Persona

Empresa o Restaurant o Persona

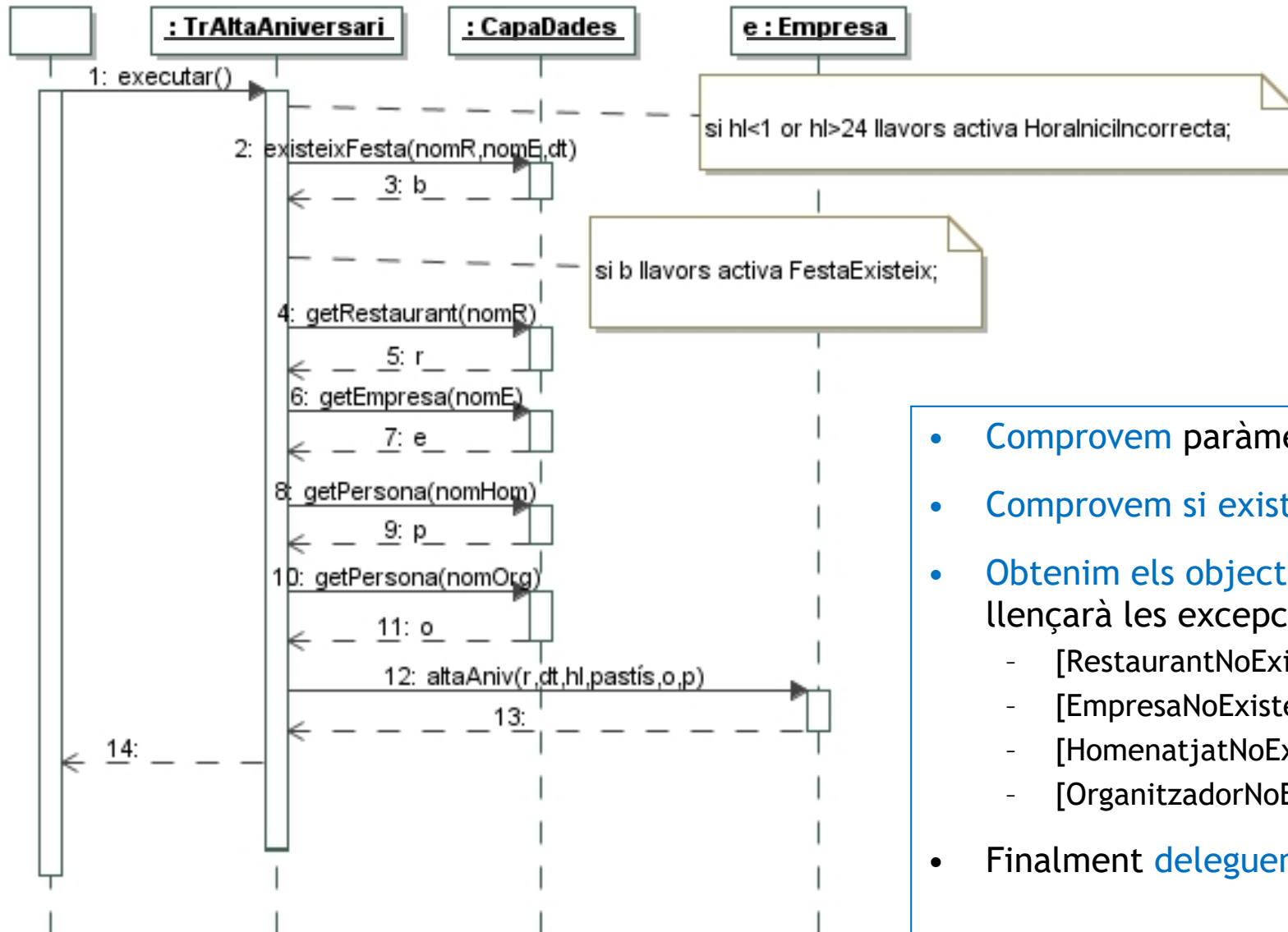
Festa

Festa o Persona

Restaurant

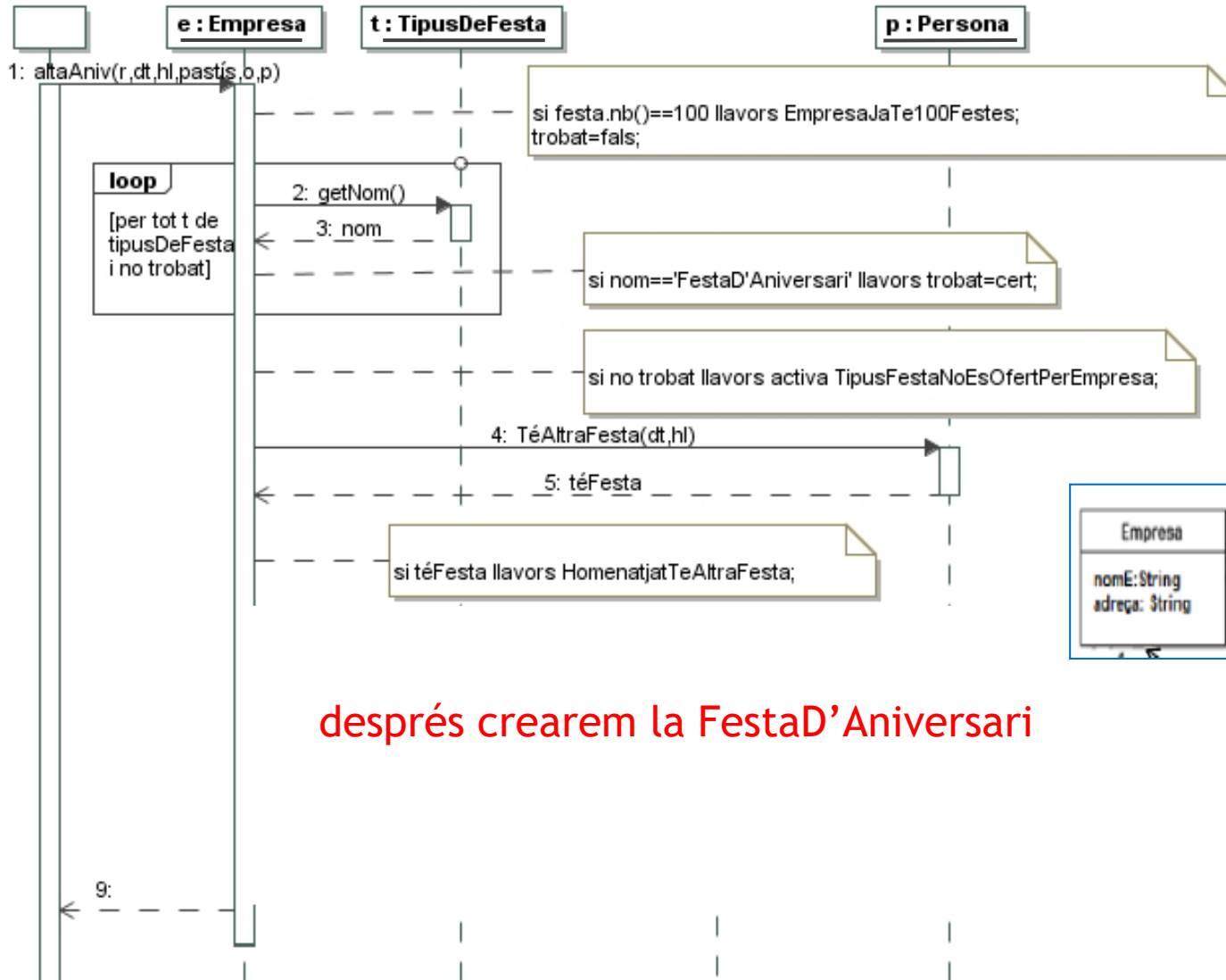


# Executar



- Comprovem paràmetre **hl**
- Comprovem si existeix Festa
- Obtenim els objectes, cosa que llençarà les excepcions:
  - [RestaurantNoExisteix]
  - [EmpresaNoExisteix]
  - [HomenatjatNoExisteix]
  - [OrganitzadorNoExisteix]
- Finalment **deleguem a Empresa**

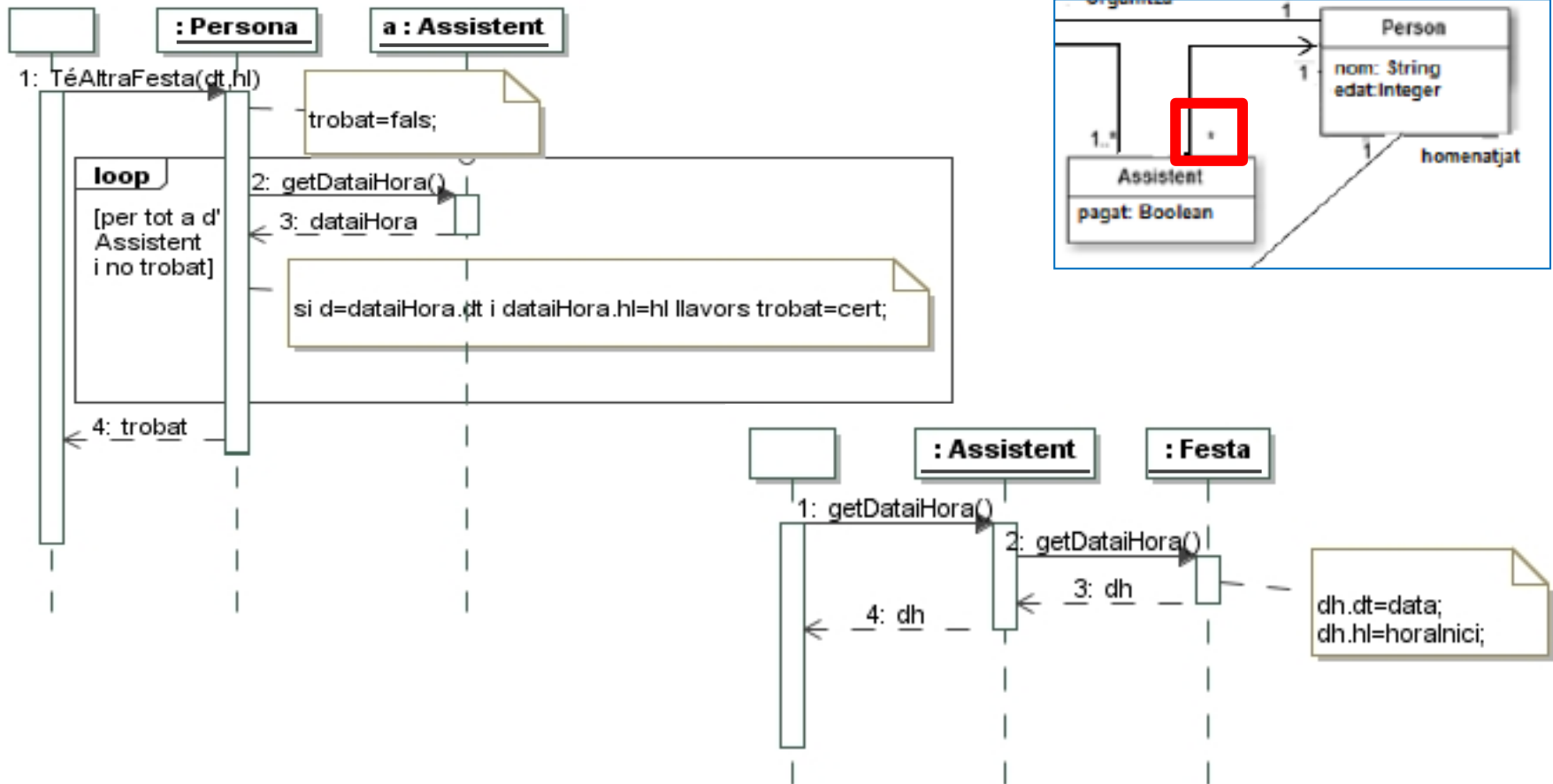
# Empresa::altaAniv(r,dt,hl,pastís,o,p)



després crearem la FestaD'Aniversari

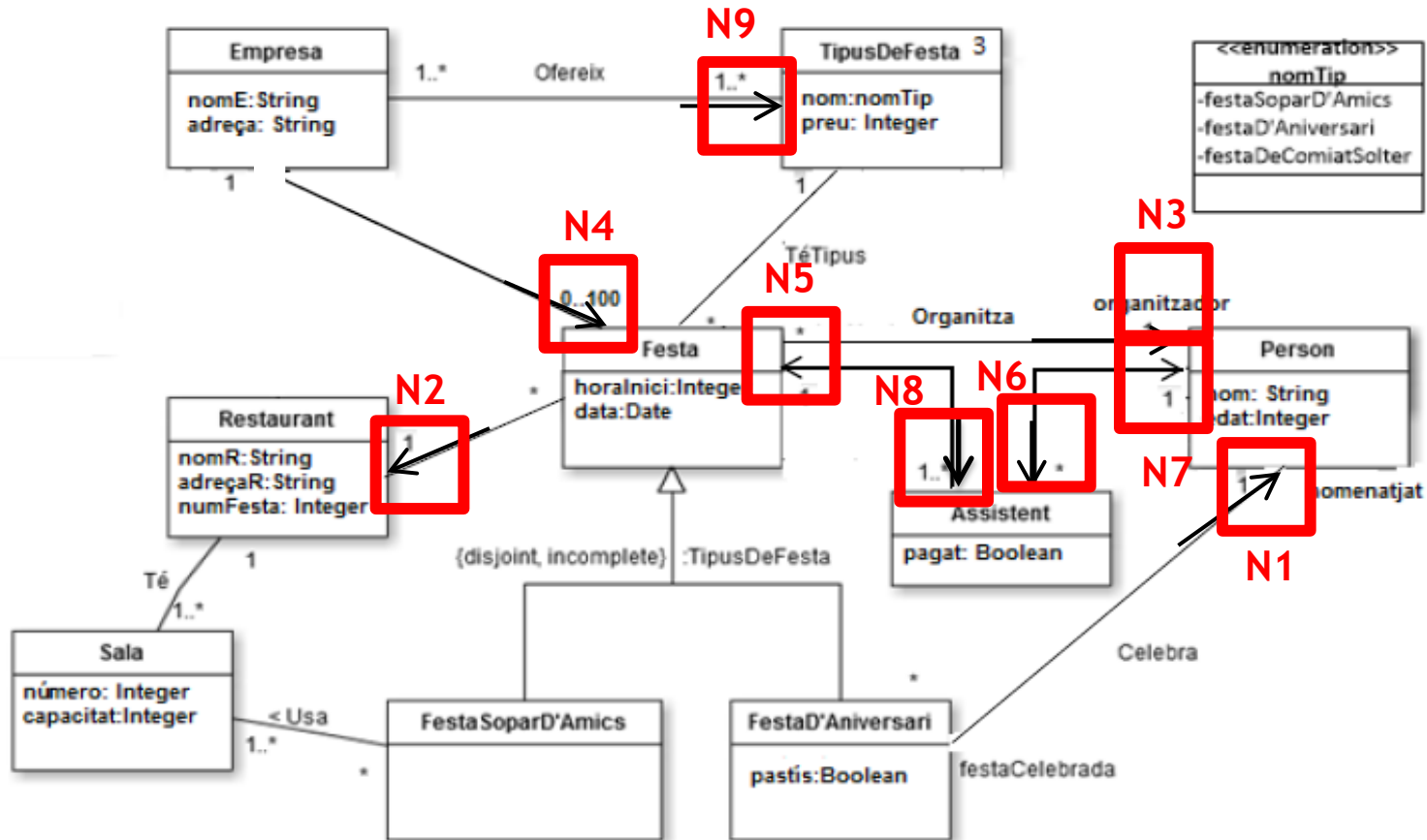
- Comprovem les excepcions: `[EmpJaTé100Festes]`, `[TipusNoOfert]`, `[HomTéAltraFesta]`

## Persona::téAltraFesta(dt,hl)

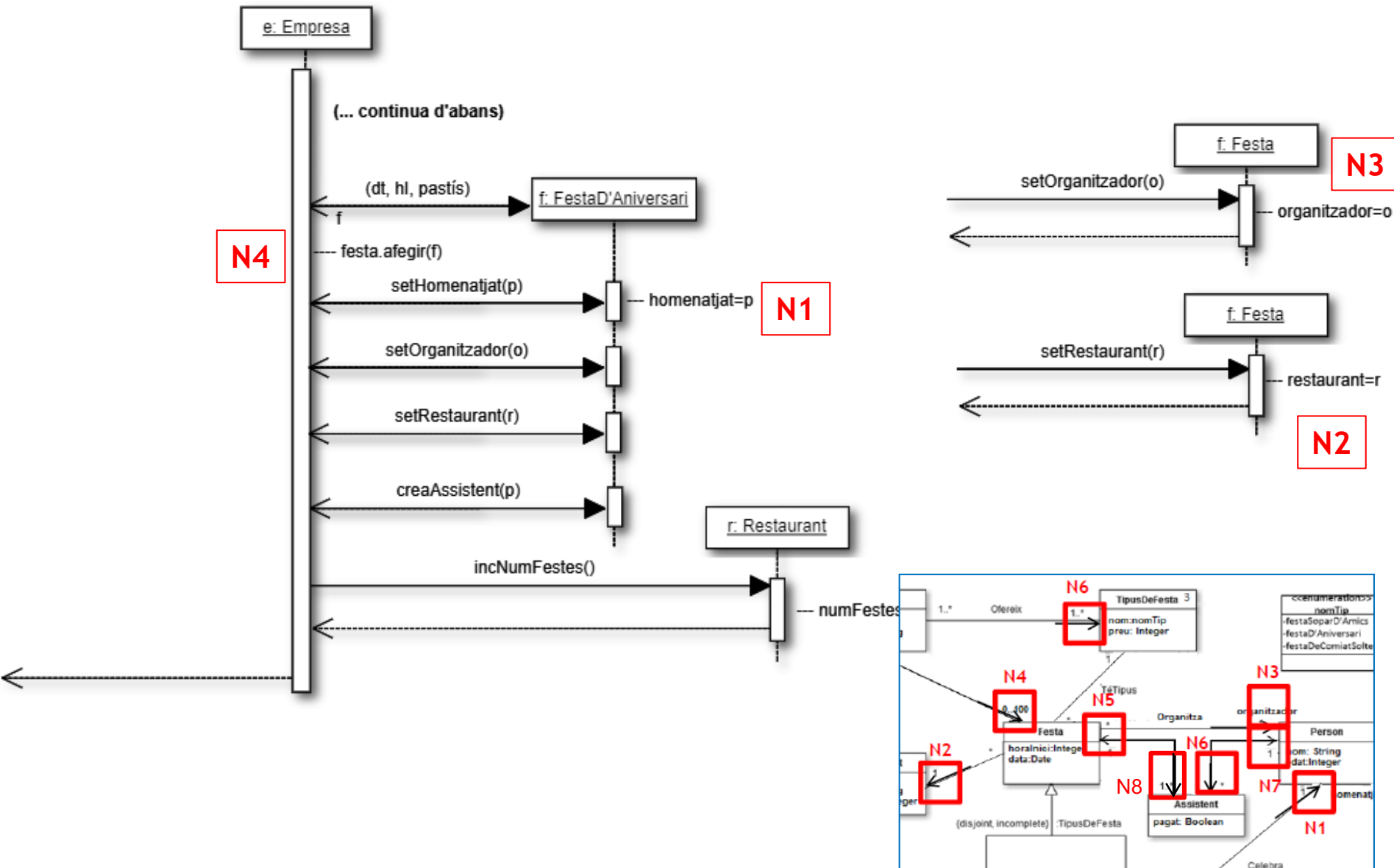


- Mirem si l'homenatjat/da assisteix a una altra festa en els mateixos data i hora d'inici

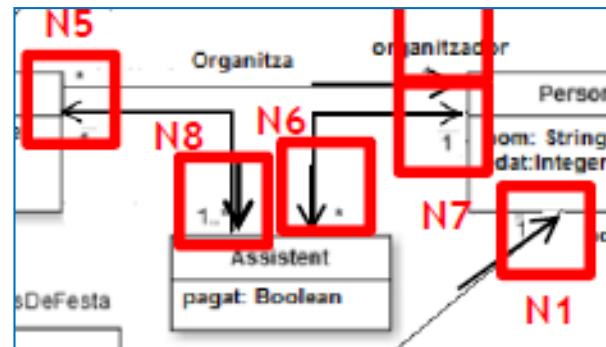
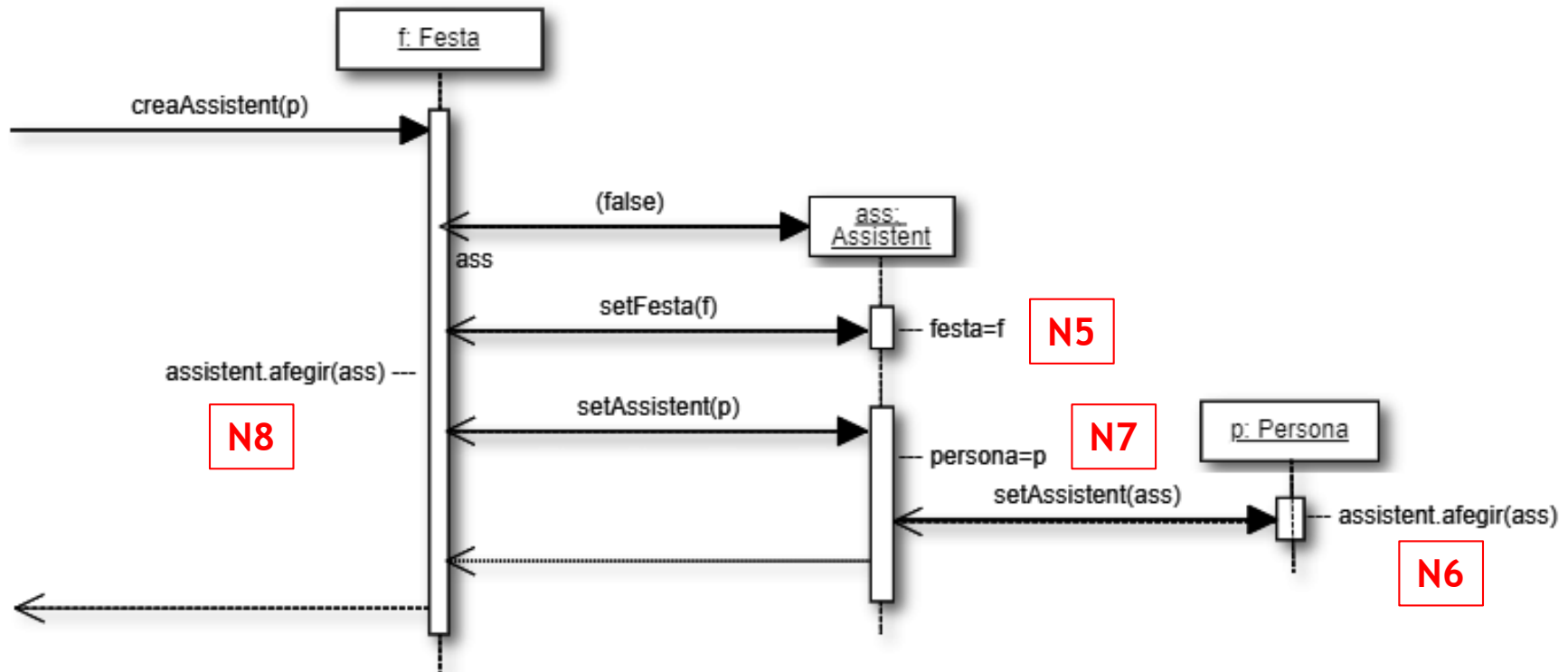
# Noves navegabilitats



# Creació de FestaDAniversari



# Creació d'Assistent



# Problema de disseny en UML: Festes d'aniversari