

Introducció a l'Enginyeria del Software (IES)

Quadrimestre de Primavera - Curs 2015/16

Exercicis resolts de patrons de disseny



Departament d'Enginyeria de Serveis
i Sistemes d'Informació

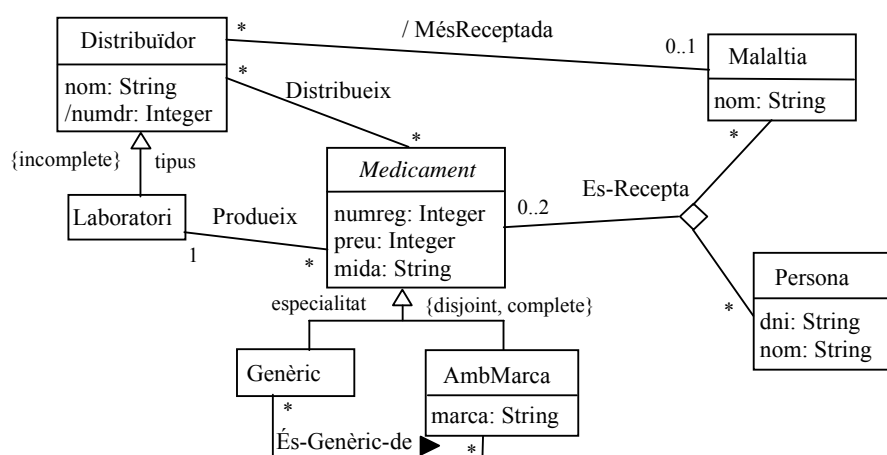
UNIVERSITAT POLITÈCNICA DE CATALUNYA



Exercicis resolts de patrons de disseny

1. Els laboratoris farmacèutics són empreses que produeixen i distribueixen medicaments. Però també existeixen altres empreses que solament els distribueixen. De tots els distribuïdors se'n coneix el nom i el nombre de medicaments distribuïts (*numdr*). Dels medicaments, se'n coneix el número de registre oficial (*numreg*), el preu i la mida. Els medicaments es recepten a persones per una malaltia concreta. Dels distribuïdors, es coneix quina ha estat la malaltia més receptada amb els medicaments que distribueixen. A més a més, es diferencien els medicaments que tenen una marca o nom comercial dels que es coneixen com a genèrics (sense marca). En qualsevol cas, es coneixen els medicament genèrics d'un medicament amb marca i a l'inrevés. A continuació disposeu de l'especificació feta per aquest sistema:

Esquema conceptual d'especificació:



R.I. Textuals:

- Les claus de classes no associatives són: (Distribuïdor, nom), (Medicament, numreg), (Malaltia, nom), (Persona, dni).
- No poden existir dos medicaments AmbMarca amb la mateixa marca i mida.
- A una persona no se li poden receptar més de 100 medicaments diferents.

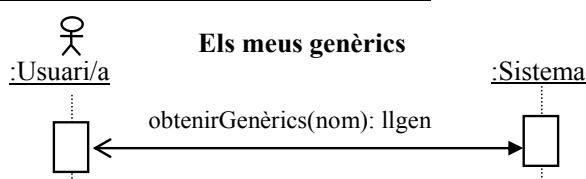
Info. derivada:

- *MésReceptada*: Relaciona un distribuïdor amb la malaltia de la que s'han fet més receptes utilitzant els medicaments distribuïts pel distribuïdor.
- *numdr*: Indica el nombre de medicaments distribuïts.

Suposicions:

- l'associació derivada *MésReceptada* és calculada i l'atribut derivat *numdr* és materialitzat

Diagrama de seqüència dels esdeveniments del sistema:



Contracte de l'operació *obtenirGenèrics*:

Operació: `obtenirGenèrics(nom:String):Set(TupleType(marca:String, mida:Integer, registres: Set(Integer)))`

Semàntica: Retorna els medicaments amb marca que són distribuïts pel laboratori amb el nom especificat i els seus genèrics.

Pre:

- Existeix el laboratori identificat per nom.

Body:

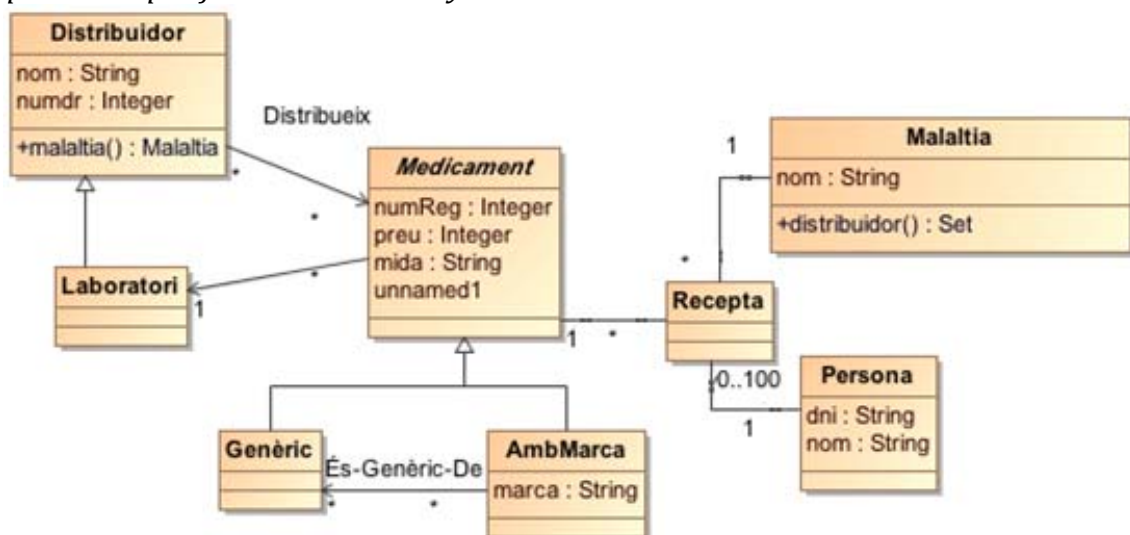
- Per tots els medicaments *AmbMarca* que distribueix el laboratori especificat, s'obté la marca i mida d'aquests, i a més, el numreg dels seus genèrics que són produïts pel propi laboratori.

Es demana:

- Diagrama de classes de disseny obtingut a partir de l'esquema conceptual de les dades, indicant explícitament les restriccions d'integritat que apareixen o desapareixen, i el contracte de l'operació *obtenirGenèrics* obtingut com a conseqüència de la traducció de l'esquema d'especificació al de disseny.
- Diagrames de seqüència de l'operació del sistema *obtenirGenèrics* i de totes les operacions que siguin invocades en aquesta operació. Cal que indiqueu als mateixos diagrames de seqüència la signatura completa de les operacions, que digueu si són abstractes/ganxo i que poseu comentaris de tot el que no hi aparegui de forma explícita. Indiqueu al diagrama de classes de l'apartat a) la navegabilitat resultant del vostre disseny.

Solució:

- Diagrama de classes de disseny obtingut a partir de l'esquema conceptual de les dades, indicant explícitament les restriccions d'integritat que apareixen o desapareixen, i el contracte de l'operació *obtenirGenèrics* obtingut com a conseqüència de la traducció de l'esquema d'especificació al de disseny.



NOTA: El tipus de retorn de `distribuidor()` hauria de ser `Set(Distribuidor)`

La restricció RI3 ara es pot expressar gràficament (0..100 a Recepta)

Afegim:

RI4: A una persona no se li pot receptar dues vegades el mateix medicament per a una malaltia concreta

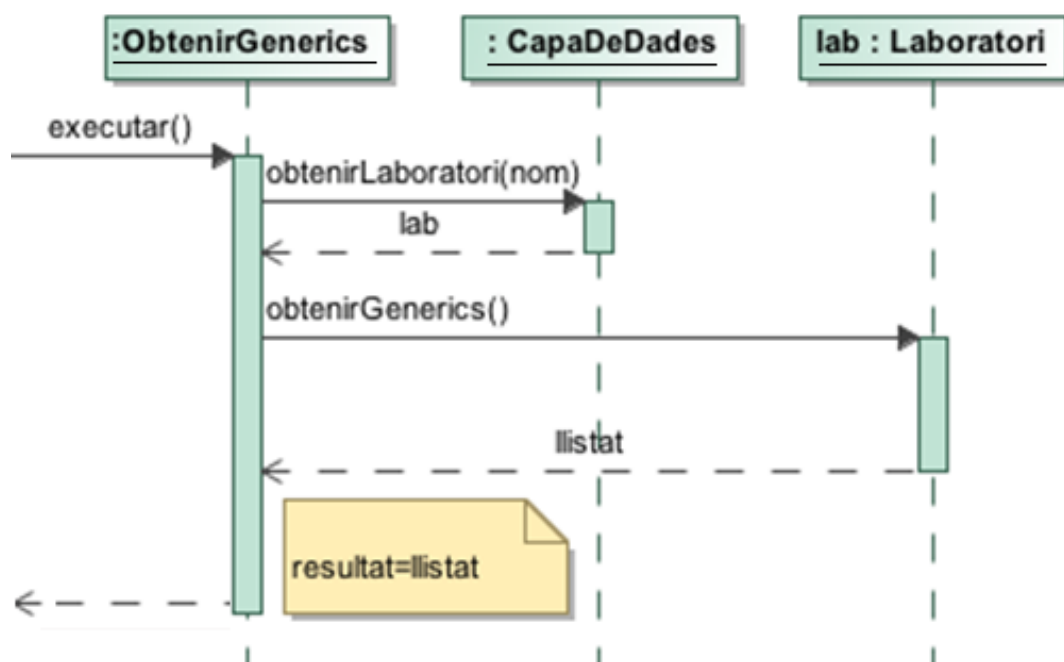
RI5: A una persona no se li poden receptar més de dos medicaments per la mateixa malaltia

Context CapaDeDomini::obtenirGenèrics(nom:String):Set(TupleType(marca:String, mida:Integer, registres:Set(Integer)))

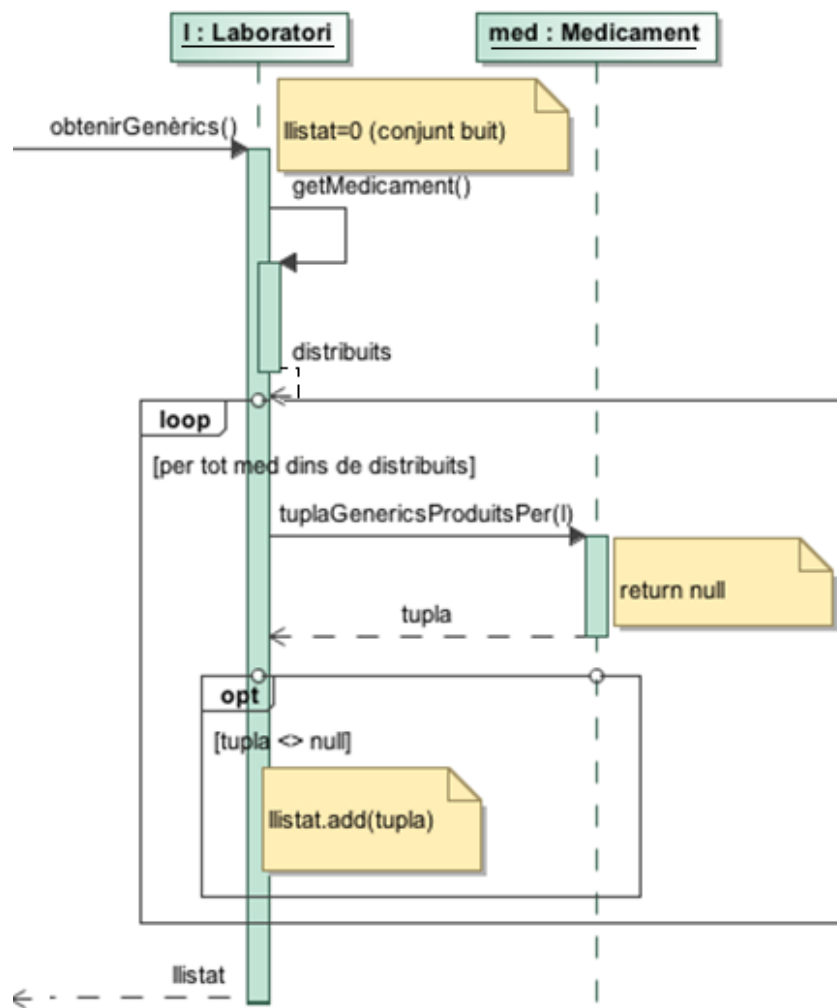
exc: laboratori-no-existeix (pre original)

post: (original) Per tots els medicaments AmbMarca que distribueix el laboratori especificat, s'obté la marca i mida d'aquests, i a més, el numreg dels seus genèrics que són produïts pel propi laboratori.

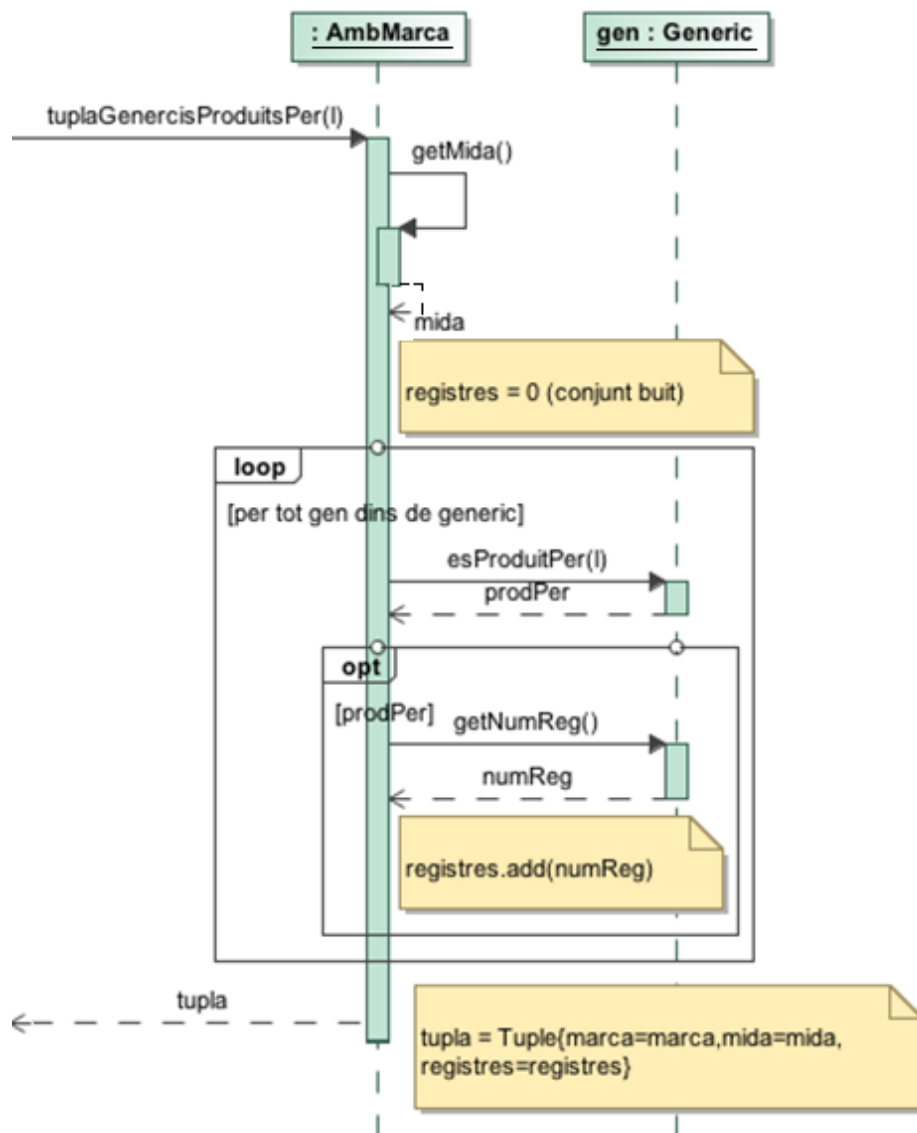
- b) *Diagrames de seqüència de l'operació del sistema acceptar i de totes les operacions que siguin invocades en aquesta operació. Cal que indiqueu als mateixos diagrames de seqüència la signatura completa de les operacions, que digueu si són abstractes/ganxo i que poseu comentaris de tot el que no hi aparegui de forma explícita. Indiqueu al diagrama de classes de l'apartat a) la navegabilitat resultant del vostre disseny*



Podem buscar el laboratori directament a la capa de dades. Com que l'excepció del nostre contracte és la mateixa que la del contracte d'**obtenirLaboratori**, si el laboratori no existeix podem propagar l'excepció.



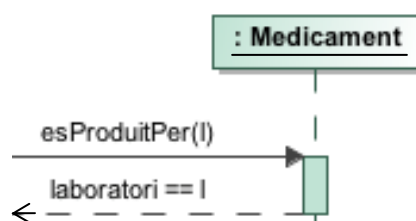
Aprofitarem el polimorfisme per tal que només els medicaments amb marca creïn la tupla que caldrà retornar al l·listat.



La mida, al contrari que la marca, és un atribut privat de Medicament i, per tant, no hi podem accedir directament. L'operació `esProduitPer` que invoquem sobre `Generic` l'hem de definir a `Medicament` que és l'expert que sap quin és el laboratori on es produeix el medicament.

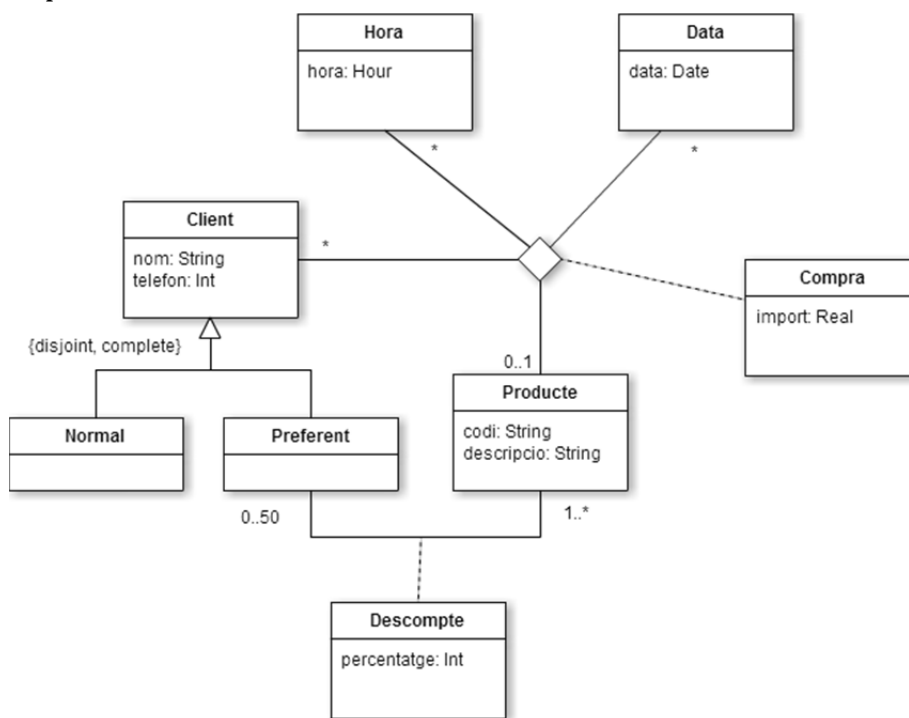
Cal tenir en compte que necessitem que ens passin el laboratori perquè ningú ens assegura que el laboratori que produeix el medicament `AmbMarca` sigui el mateix pel qual estiguem demanant el llistat (no hi ha cap restricció d'integritat que digui que un laboratori no pugui distribuir medicaments produïts per altres laboratoris).

En aquest cas, ens podrien haver passat només el nom del Laboratori i haguèssim evitat acoblar `AmbMarca` amb `Laboratori` tot i que llavors haguèssim trencat l'encapsulament de la clau de `Laboratori` (`AmbMarca` ha de saber que n'hi ha prou amb el nom per identificar un Laboratori).



2. Els clients d'una cadena de supermercats poden ser de tipus normal o preferent. El sistema enregistra les compres de productes que ha fet cada client. Els clients de tipus preferent tenen un o més descomptes a productes.

Esquema conceptual de les dades



Claus externes: (Client, nom), (Producte, codi), (Data, data), (Hora, hora)

Contracte de les operacions *clientsImportantsDelProducte* i *altaDescompte*

Operació *clientsImportantsDelProducte*(codiP: String): Set(String)

body: retorna el nom de tots els clients que han comprat alguna vegada el producte *codiP* i que són importants. Un client preferent és important si té 3 descomptes o més; un client normal és important si la suma de totes les seves compres és major que 1000.

Operació *altaDescompte*(nomC: String, codiP: String, perc: int)

pre: existeix un producte *prodC*

pre: existeix un client preferent *nomC*

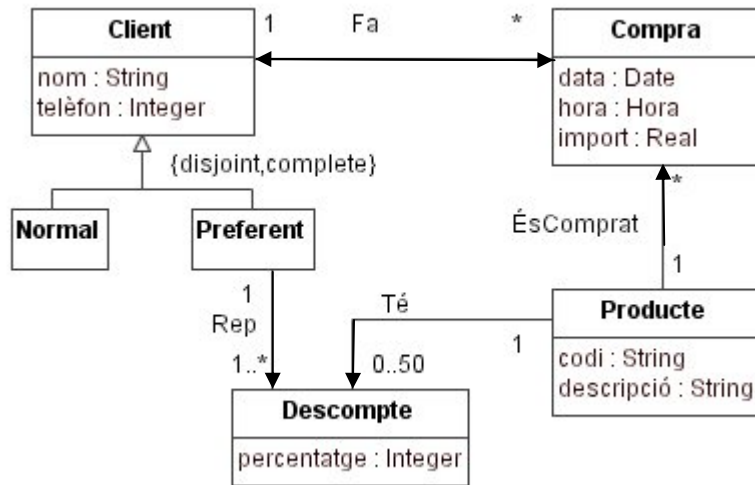
post: Es crea una instància de descompte amb percentatge *perc* associada al client preferent *nomC* i al producte amb codi *codiP*

Es demana

- Diagrama de classes de disseny obtingut a partir de l'esquema conceptual de les dades, indicant explícitament les restriccions d'integritat que apareixen o desapareixen.
- El contracte de les operacions obtingut al traduir els contractes d'especificació a contractes de disseny.
- Diagrames de seqüència de les operacions *clientsImportantsDelProducte* i *altaDescompte* i de totes les operacions invocades per aquestes. Poseu comentaris de tot el que no hi aparegui de forma explícita.
- Indiqueu al diagrama de classes de l'apartat a les navegabilitats resultants del vostre disseny.

Solució:

a) Diagrama de classes de disseny i navegabilitats:



Restriccions d'integritat textuals

1. Claus externes: (Client, nom), (Producte, codi).
2. No hi pot haver dues compres amb el mateix client, data i hora.
3. No hi pot haver dos descomptes amb el mateix client preferent i producte

b) Contracte de disseny de les operacions:

Operació `clientsImportantsDelProducte(codiP: String): Set(String)`

body: retorna el nom de tots els clients que han comprat alguna vegada el producte *codiP* i que són importants. Un client preferent és important si té 3 descomptes o més; un client normal és important si la suma de totes les seves compres és major que 1000.

Operació `altaDescomptet(nomC: String, codiP: String, perc:int)`

exc: `producte-no-existeix`: No existeix el producte amb codi *codiP*

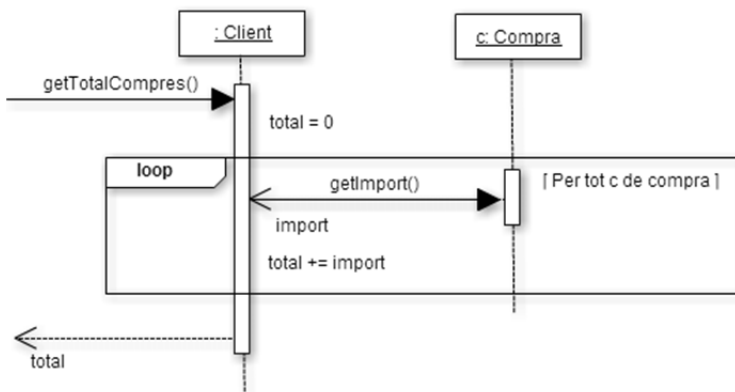
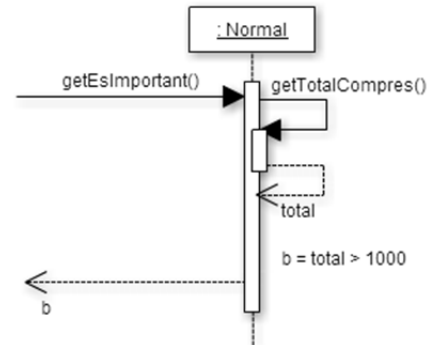
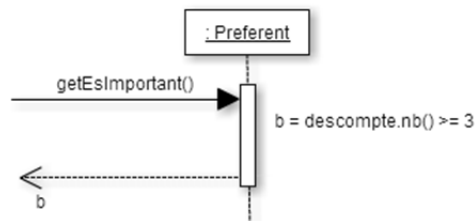
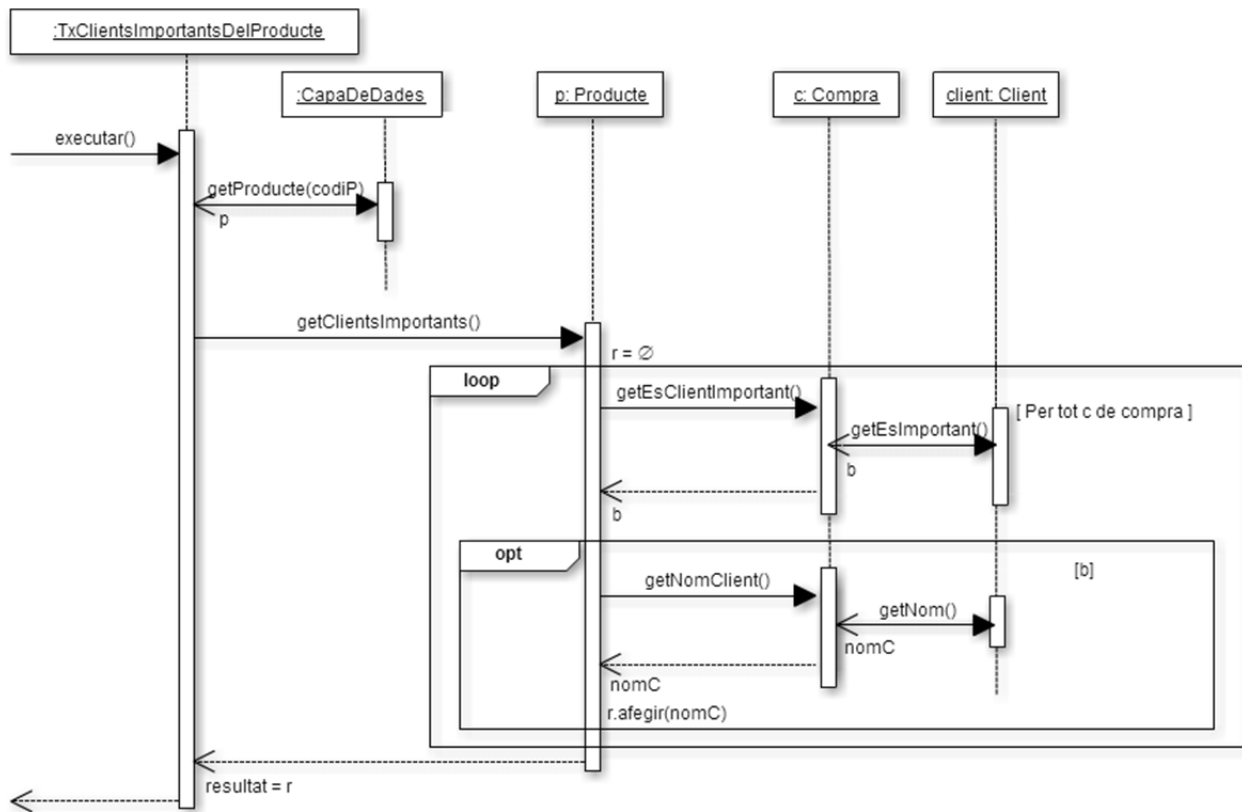
exc: `client-preferent-no-existeix`: El client preferent amb nom *nomC* no existeix

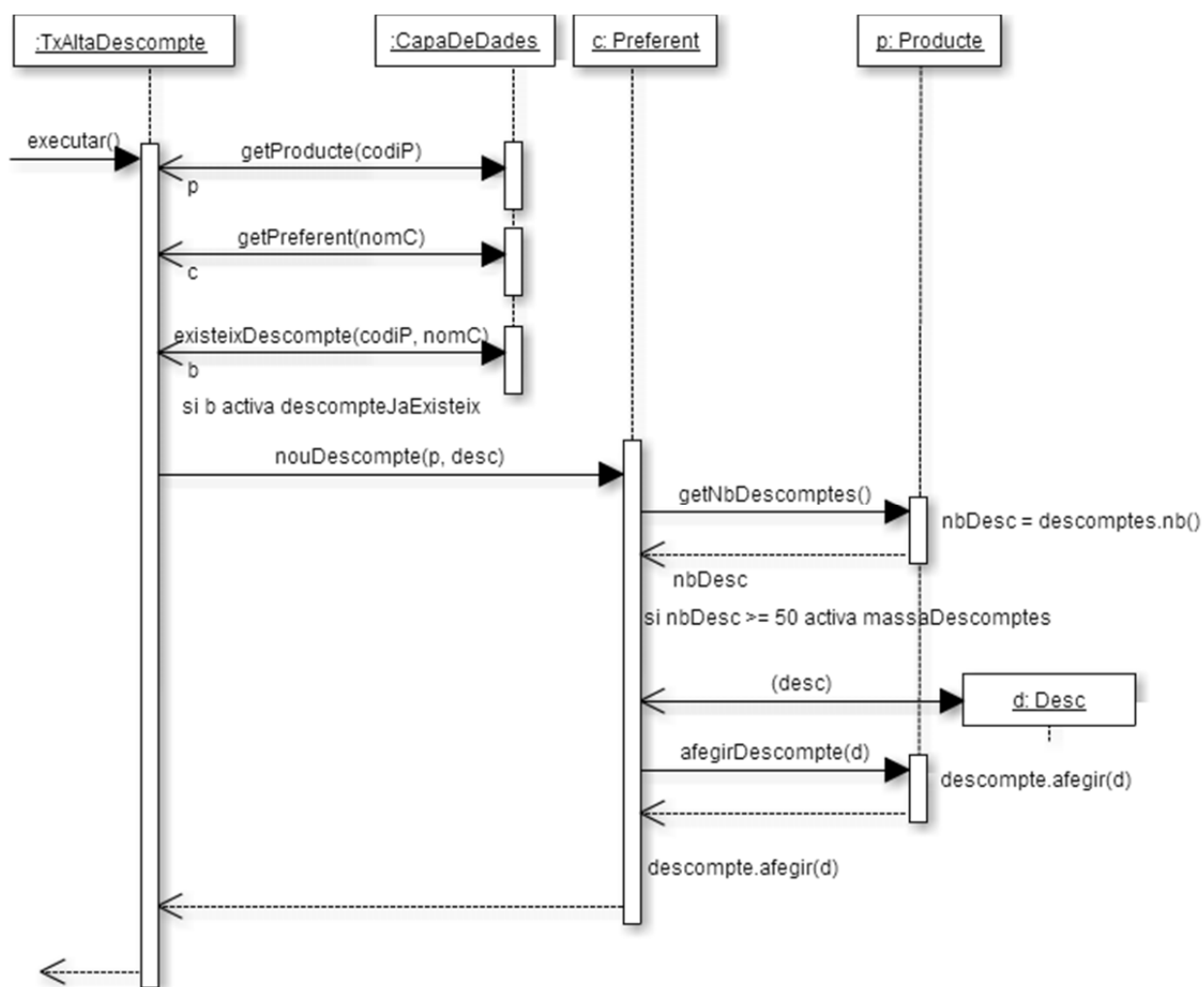
exc: `descompteJaExisteix`: El descompte entre el producte *codiP* i el client *nomC* ja existeix

exc: `massaDescomptes`: El producte amb codi *codiP* té 50 descomptes

post: `promoció`: Es crea una instància de descompte amb percentatge *perc* que s'associa amb el client *nomC* i al producte amb codi *codiP*

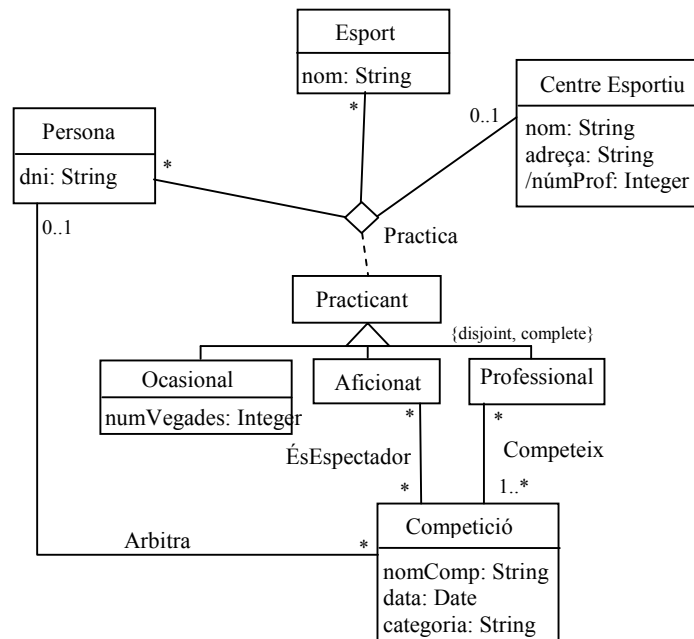
c) Diagrames de seqüència





3. Una associació de centres esportius ens ha demanat que li dissenyem un sistema software per gestionar la pràctica d'esports als seus centres. Una persona pot practicar un esport a un centre esportiu de forma ocasional, com aficionat o com a professional. De les persones que practiquen un esport ocasionalment se'n guarda el nombre de vegades que hi van a fer esport en un mes. Dels aficionats, se'n coneix les competicions en les que són espectadors. Finalment, dels professionals cal enregistrar les competicions en les que competeixen. Tota competició té una persona que l'arbitra.

Esquema conceptual d'especificació:



R.I. Textuals:

- Claus classes no associatives: (Persona, dni); (Esport, nom); (Centre Esportiu, nom); (Competició, nomComp).
- Una persona que practica un esport ocasionalment no pot anar més de deu vegades al mes al centre esportiu.
- Una persona que practica un esport de forma professional no pot competir en competicions de categoria 'Aficionat'.
- De totes les pràctiques que té una persona, només una pot ser professional.
- Una persona no pot arbitrar una competició a la que competeix com a professional

Informació derivada:

- /númProf de Centre Esportiu és igual al nombre de practicants professionals d'aquell centre.

Contracte d'especificació de l'operació *altaProfessional*:

Operació: altaProfessional (dni: String, nomEsp: String, nomCentr: String; nomComp:String)

Pre:

- existeix la persona identificada per dni
- existeix l'esport identificat per nomEsp
- existeix el centre identificat per nomCentr
- existeix la competició identificada per nomComp

Post:

- Es dóna d'alta una instància de Professional definida per dni, nomEsp i nomCentre. Es crea l'associació entre aquesta instància i la competició amb nomComp.

Contracte d'especificació de l'operació *obtenirEsports*:

Operació: obtenirEsports(dni: String, nomCentre:String): Set(String)

Pre:

- existeix la persona identificada per dni.

Body:

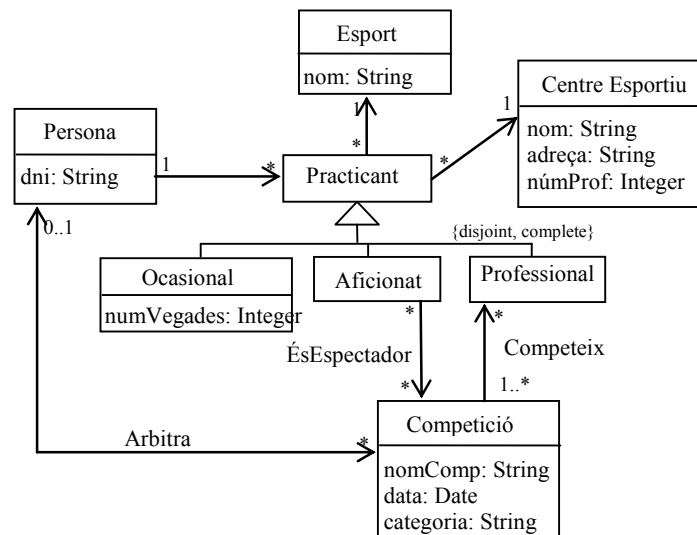
- s'obté el nom dels esports que la persona *dni* practica com a aficionat al centre *nomCentre* sempre i quan, com a practicant aficionat d'aquell esport a aquell centre, hagi sigut espectador d'alguna competició que també ha arbitrat la persona *dni*.

Suposant que l'atribut *númProf* de Centre Esportiu és materialitzat, es demana:

- Diagrames de seqüència de les operacions de disseny *altaProfessional* i *obtenirEsports* i de totes les operacions que siguin invocades a aquestes operacions. Cal que indiqueu als mateixos diagrames de seqüència **els paràmetres de les operacions**, que digueu si són **abstractes/ganxo** i que poseu **comentaris** de tot el que no hi aparegui de forma explícita.
- Diagrama de classes obtingut a partir de l'esquema conceptual d'especificació, on s'hi ha d'indicar la **navegabilitat** resultant del vostre disseny. La navegabilitat inicial és la que està explícitament indicada a l'esquema conceptual d'especificació.

• Solució:

- Diagrama de classes de disseny (amb la navegabilitat resultant del disseny):



Restriccions d'integritat afegides:

- No poden existir dues instàncies de practicant amb els mateixos Persona i Esport

b) Contracte de disseny i diagrama de seqüència d'*obtenirCompetis*

Operació: obtenirEsports(dni: String, nomCentre:String): Set(String)

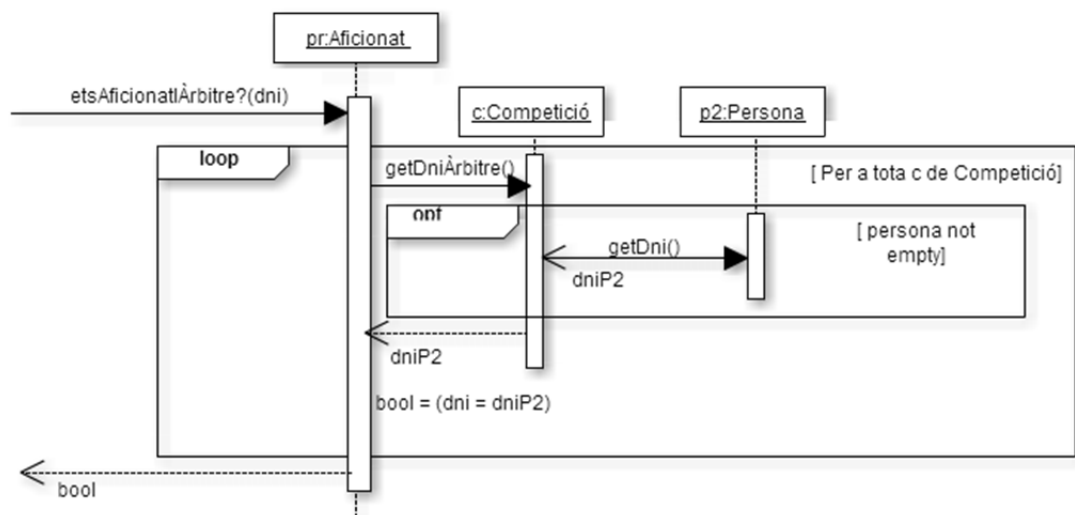
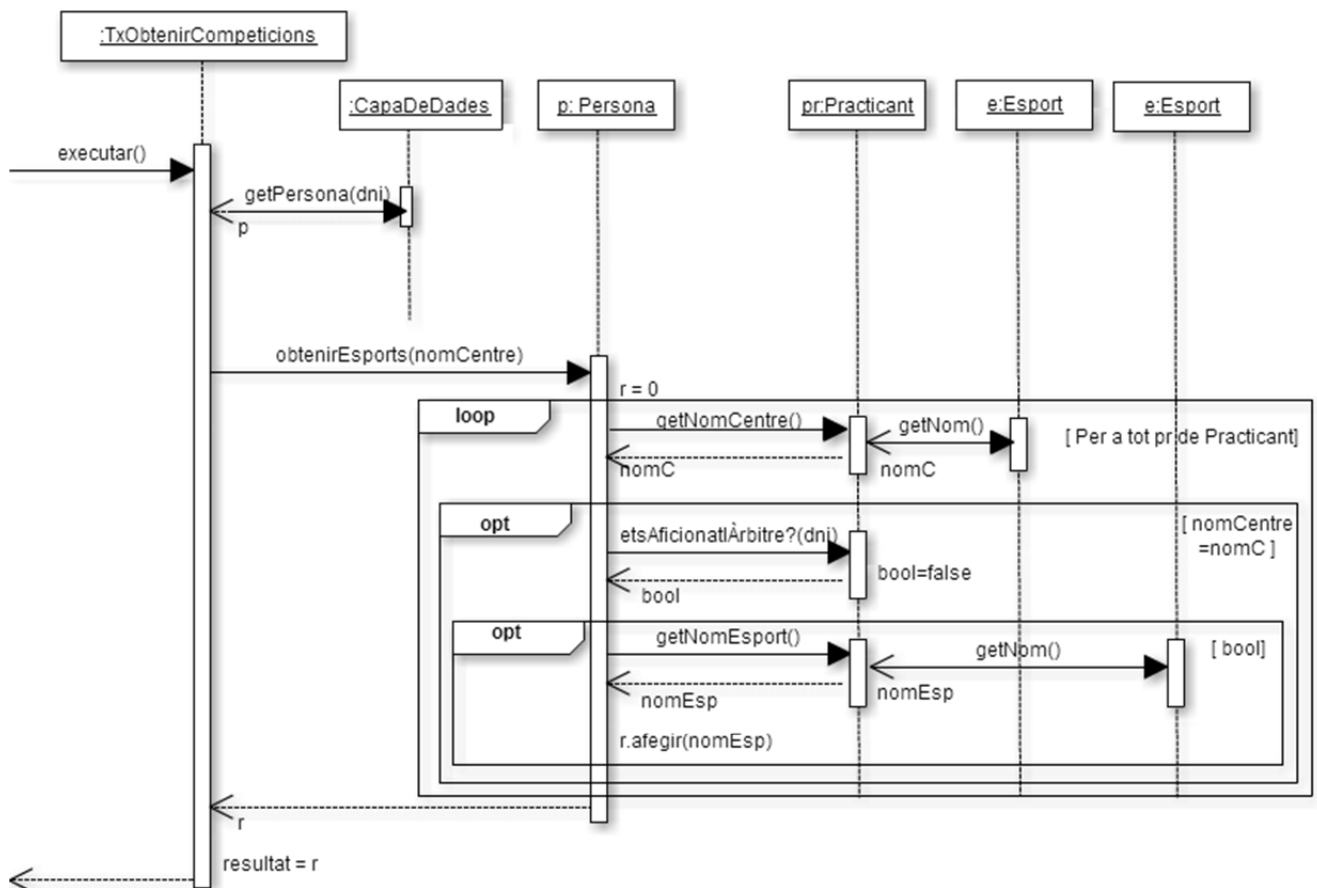
Pre:

Exc:

- personaNoExisteix: la persona identificada per dni no existeix.

Body:

- s'obté el nom dels esports que la persona *dni* practica com a aficionat al centre *nomCentre* sempre i quan, com a practicant aficionat d'aquell esport a aquell centre, hagi sigut espectador d'alguna competició que també ha arbitrat la persona *dni*.



c) Contracte de disseny i diagrama de seqüència d'*altaProfessional*

Operació: altaProfessional (dni: String, nomEsp: String, nomCentr: String;
nomComp:String)

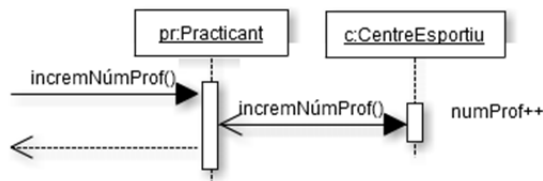
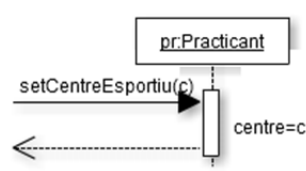
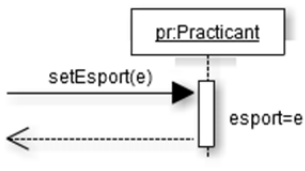
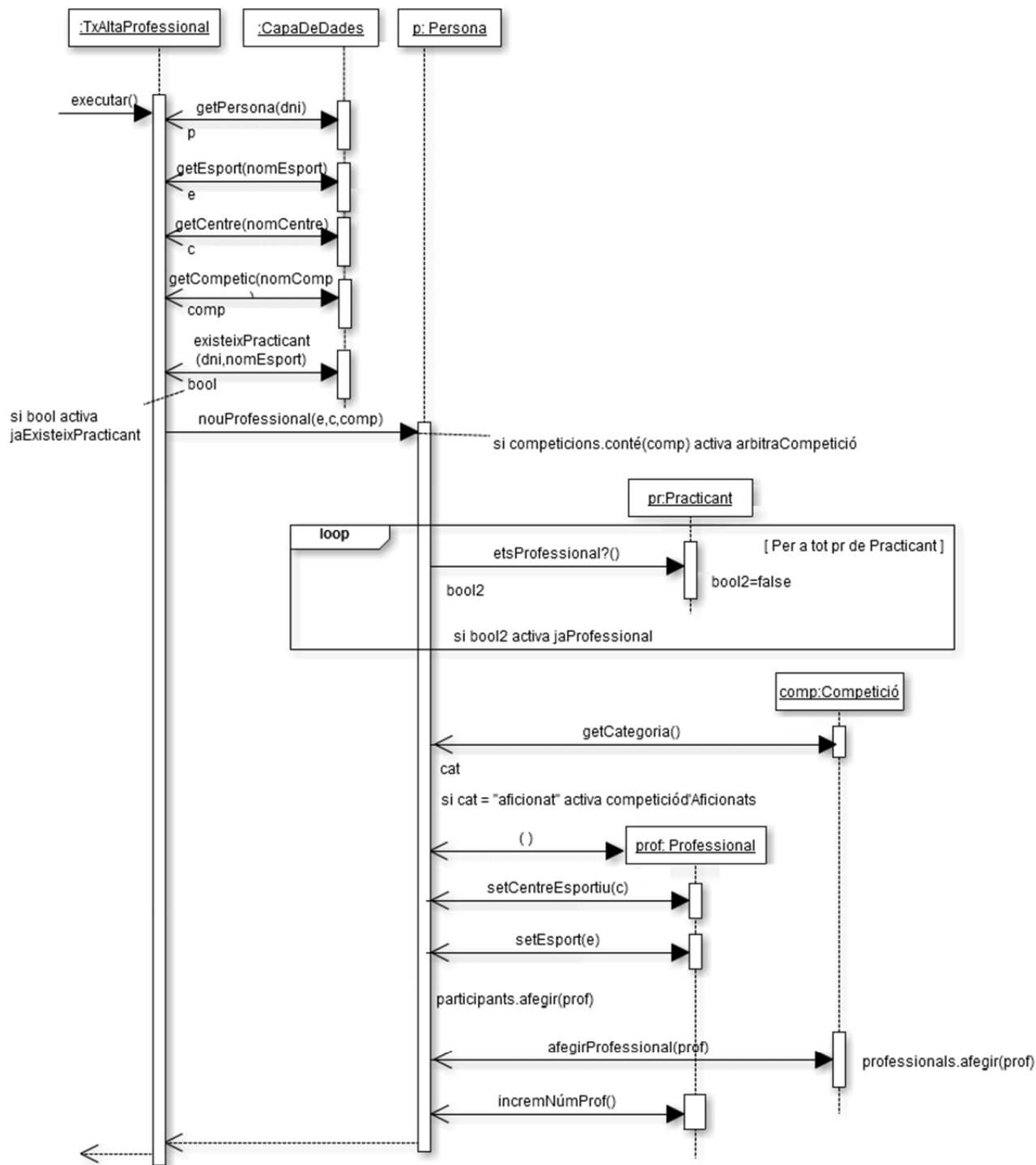
Pre:

Exc:

- personaNoExisteix: la persona dni no existeix.
- esportNoExisteix: l'esport nomEsp no existeix.
- centreNoExisteix: el centre nomCentr no existeix.
- competicióNoExisteix: la competició nomComp no existeix.
- jaÉsPracticant: la persona dni ja practica l'esport nomEsp
- arbitraCompetició: la persona dni arbitra la competició nomComp
- jaProfessional: la persona dni ja practica algun esport com a professional
- competiciód'Aficionats: la competició nomComp té la categoria "aficionat"

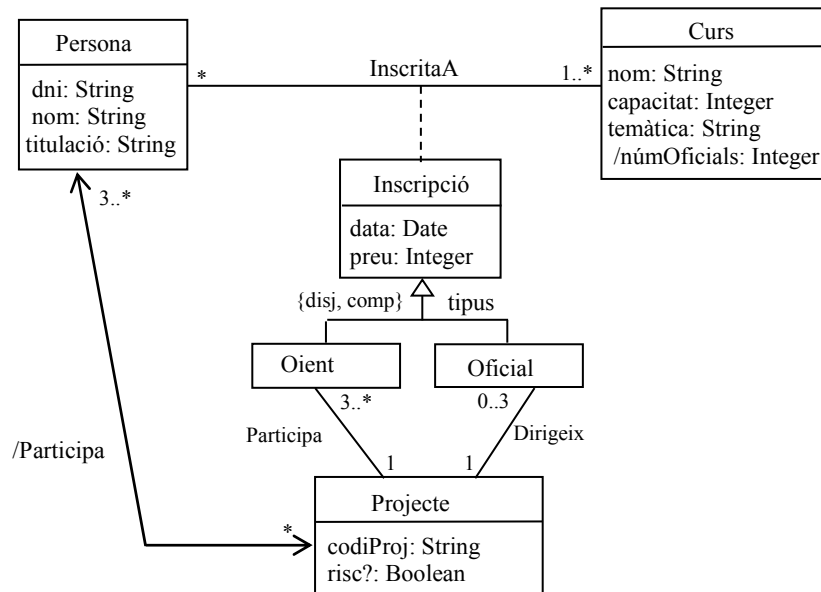
Post:

- Es dona d'alta una instància de Professional definida per dni, nomEsp i nomCentre. Es crea l'associació entre aquesta instància i la competició amb nomComp. S'incrementa en 1 l'atribut númProf del centre esportiu on dni practica nomEsport.



4. Una acadèmia està interessada a dissenyar un sistema software que gestioni la informació de les inscripcions als cursos que ofereix. Un curs s'identifica per nom i se n'enregistra també la seva capacitat i temàtica. Els alumnes de l'acadèmia s'inscriuen als cursos oferts. Una inscripció d'una persona a un curs pot ser com a oient o bé oficial. De les inscripcions com a oient se'n coneixen els projectes en què es participa mentre que de les inscripcions oficials se'n sap el projecte que es dirigeix. Dels projectes s'enregistra el seu codi i si són de risc o no. A continuació disposeu de l'especificació d'aquest sistema.

Esquema conceptual d'especificació:



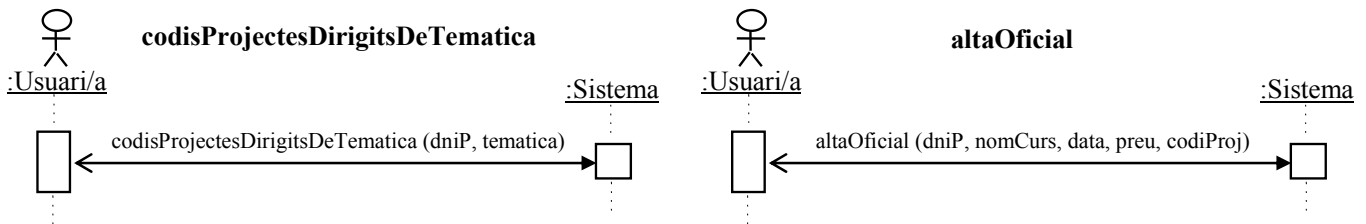
Restriccions d'integritat textuais:

- Claus classes no associatives: (Persona, dni); (Curs, nom); (Projecte, codiProj).
- Una persona no pot estar inscrita a dos (o més) cursos amb la mateixa temàtica.
- Un curs pot tenir com a màxim 10 inscripcions com a oient.
- Un curs no pot tenir més inscripcions oficials que la seva capacitat.
- Una persona no pot dirigir un projecte en el que participa com a oient (via inscripcions diferents).
- Només les persones amb titulació='enginyer' poden dirigir projectes de risc.

Info. derivada:

- *Participa* relaciona una Persona amb els projectes en els que participa per les seves inscripcions com a Oient
- *númOficials* d'un Curs: indica el nombre d'inscripcions oficials d'aquell curs.

Diagrames de seqüència d'esdeveniments del sistema:



Contractes de les operacions *codisProjectesDirigitsDeTematica* i *alta Oficial*:

Operació: *codisProjectesDirigitsDeTematica* (dniP: String, tematica: String): Set(String)

Pre:

- Existeix una persona *dniP*

Body:

- Retorna el conjunt dels codis dels projectes dirigits per la persona *dniP* a través d'una inscripció oficial per un curs de temàtica *tematica*.

Operació: *altaOficial* (dniP: String, nomCurs: String, data: Date, preu: Int, codiProj: String)

Pre:

- Existeix una persona *dniP*
- Existeix el curs *nomCurs*
- Existeix el projecte *codiProj*

Post:

- Es dona d'alta una inscripció oficial entre la persona *dniP* i el curs *nomCurs* amb la data *data* i preu *preu*. Es crea una instància de l'associació *Dirigeix* entre el projecte *codiProj* i la nova inscripció oficial.

Tenint en compte que:

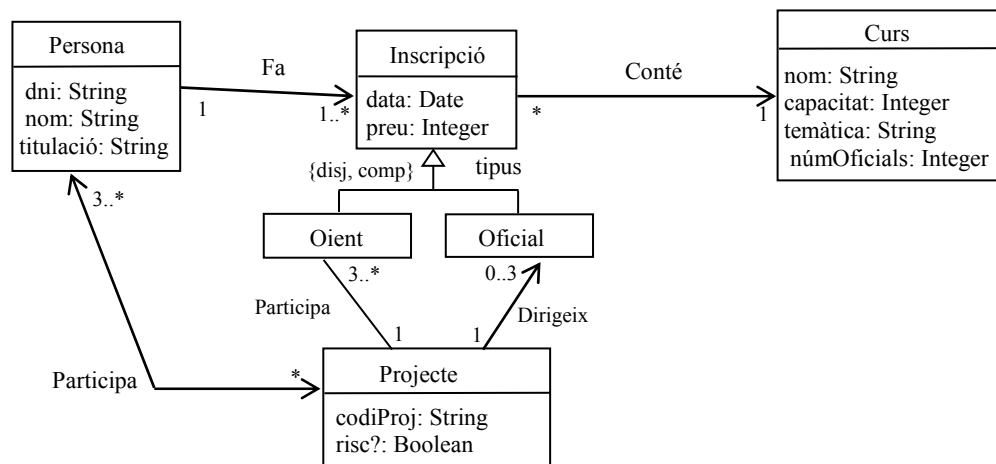
- L'atribut derivat *númOficials* i l'associació derivada *Participa* han de ser materialitzats.
- La navegabilitat de l'associació derivada *Participa* és doble, tal i com es mostra a l'esquema conceptual d'especificació.

Es demana:

- Diagrama de classes de disseny obtingut a partir de l'esquema conceptual de les dades, indicant explícitament les restriccions d'integritat que apareixen o desapareixen.
- El contracte de les operacions *codisProjectesDirigitsDeTematica* i *altaOficial* obtingut com a conseqüència de la traducció de l'esquema d'especificació al de disseny.
- Diagrames de seqüència de les operacions *codisProjectesDirigitsDeTematica* i *altaOficial* i de totes les operacions que siguin invocades en aquestes operacions. Poseu comentaris de tot el que no hi aparegui de forma explícita.
- Indiqueu al diagrama de classes de l'apartat a) la navegabilitat resultant del vostre disseny.

Solució:

- Diagrama de classes de disseny:



Restriccions d'integritat afegides:

~~No poden existir 2 instàncies d'inscripció amb els mateixos Persona i Curs.~~ (subsumida per Rest. textual 2)

- Contracte de disseny de les operacions *codisProjectesDirigitsDeTematica* i *alta Oficial*

Operació: *codisProjectesDirigitsDeTematica* (dniP: String, tematica: String): Set(String)

Pre:

Excepcions:

- *noExisteixPersona*: no existeix una persona *dniP*

Body:

- Retorna el conjunt dels codis dels projectes dirigits per la persona *dniP* a través d'una inscripció oficial per un curs de temàtica *tematica*

Operació: *altaOficial* (dniP: String, nomCurs: String, data: Date, preu: Int, codiProj: String)

Pre:

Excepcions:

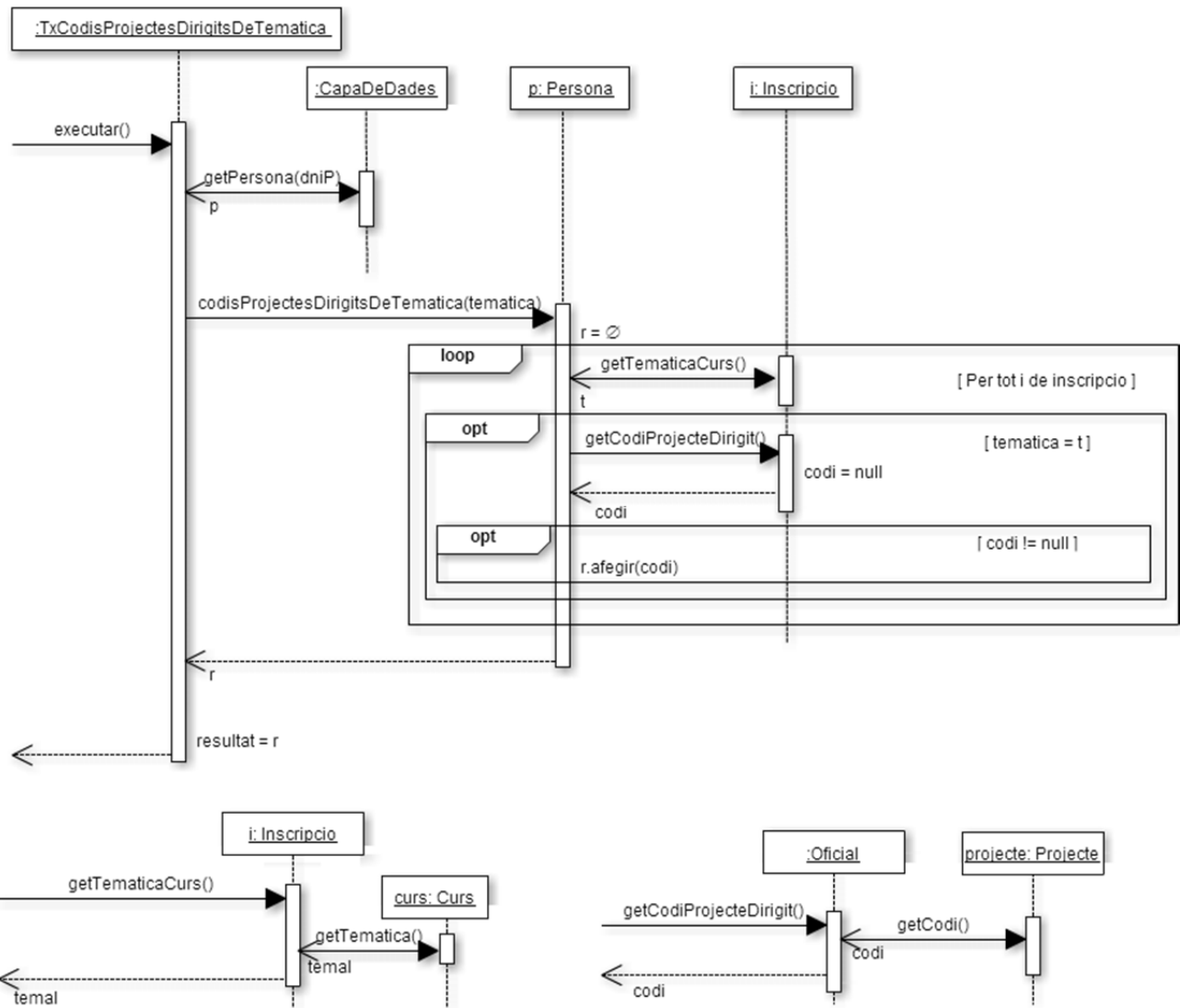
- *noExisteixPersona*: no existeix una persona *dniP*
- *noExisteixCurs*: no existeix el curs *nomCurs*
- *noExisteixProjecte*: no existeix el projecte *codiProj*
- *jaTéInscripcióEnLaTemàtica*: la persona *dniP* ja està inscrita en un curs de la mateixa temàtica que *nomCurs*
- *massaGent*: els oficials inscrits al curs *nomCurs* superen la seva *capacitat*
- *participaAlProjecte*: la persona *dniP* participa al projecte *codiProj*
- *noÉsEnginyer*: el projecte *codiProj* és de risc i la persona *dniP* no té la titulació d'enginyer
- *jaTé3Directors*: el projecte *codiProj* ja té 3 directors

NOTA: no cal comprovar que no existeixi ja la inscripció oficial perquè aquesta excepció ja està subsumida per l'excepció *jaTéInscripcióEnLaTematica*

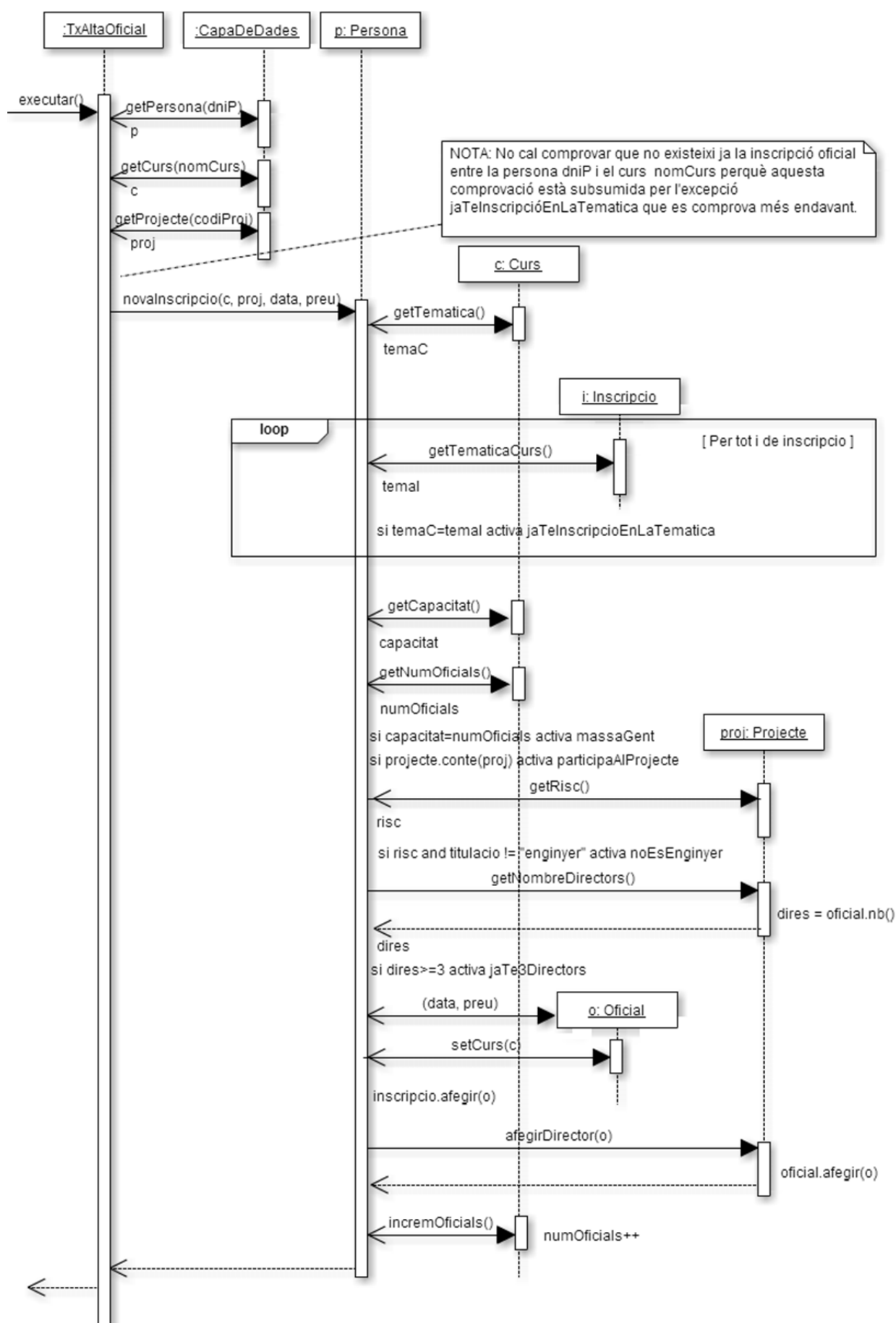
Post:

- Es dona d'alta una inscripció com a oficial entre la persona *dniP* i el curs *nomCurs* amb la data *data* i preu *preu*. Es crea una instància de l'associació *Dirigeix* entre el projecte *codiProj* i la nova inscripció oficial
- *L'atribut númOficials del curs nomCurs s'incrementa en una unitat*

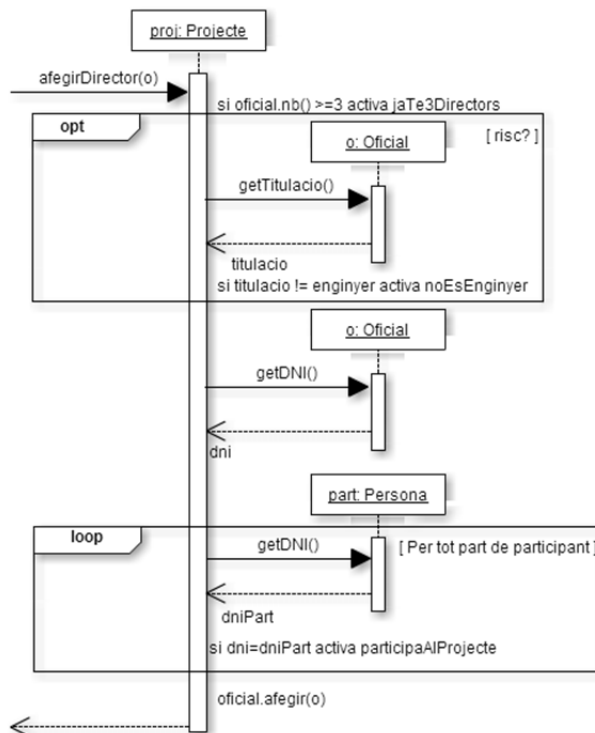
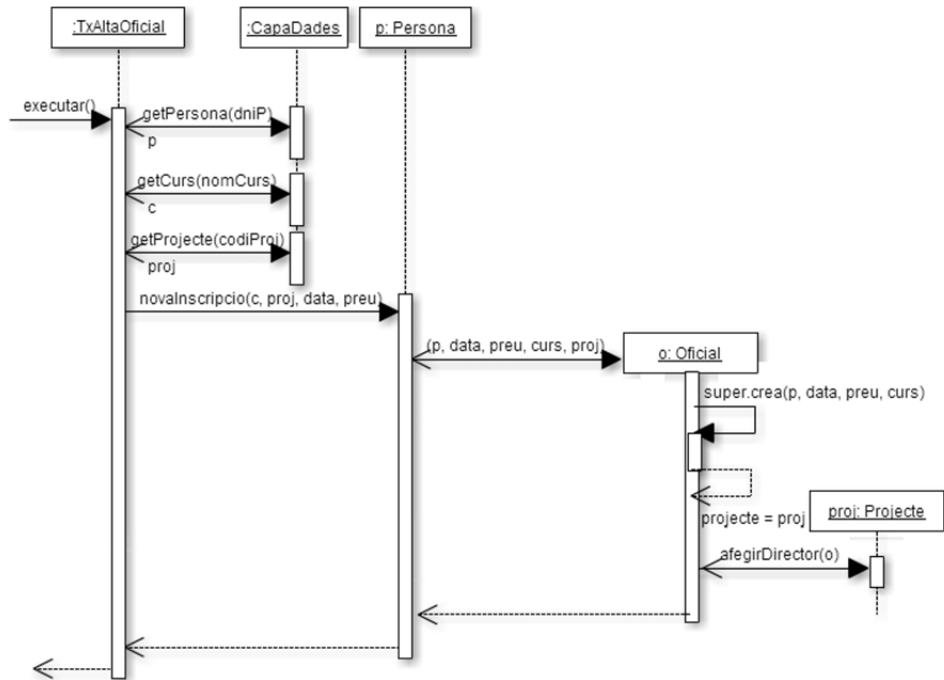
- Diagrama de seqüència de l'operació *codisProjectesDirigitsDeTematica*

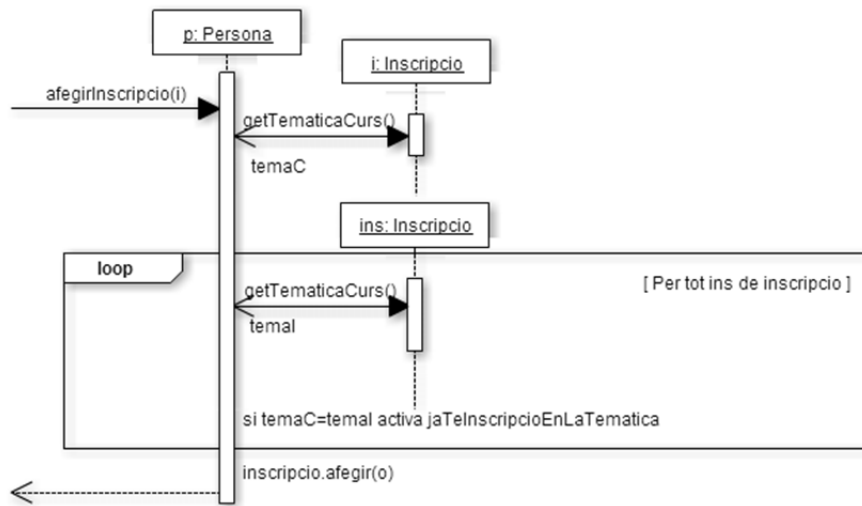
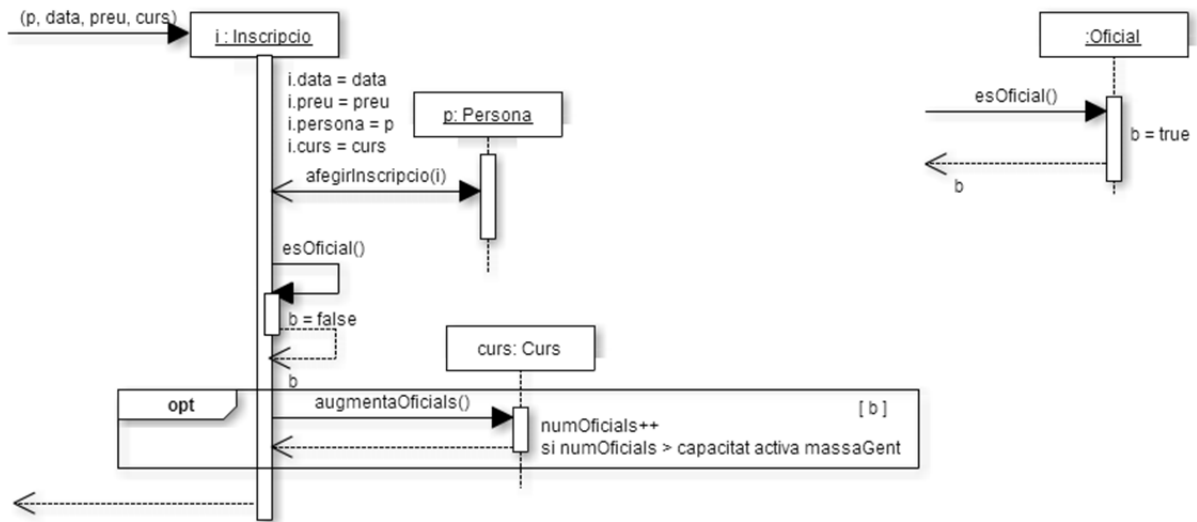


- Diagrama de seqüència de l'operació *altaOficial*



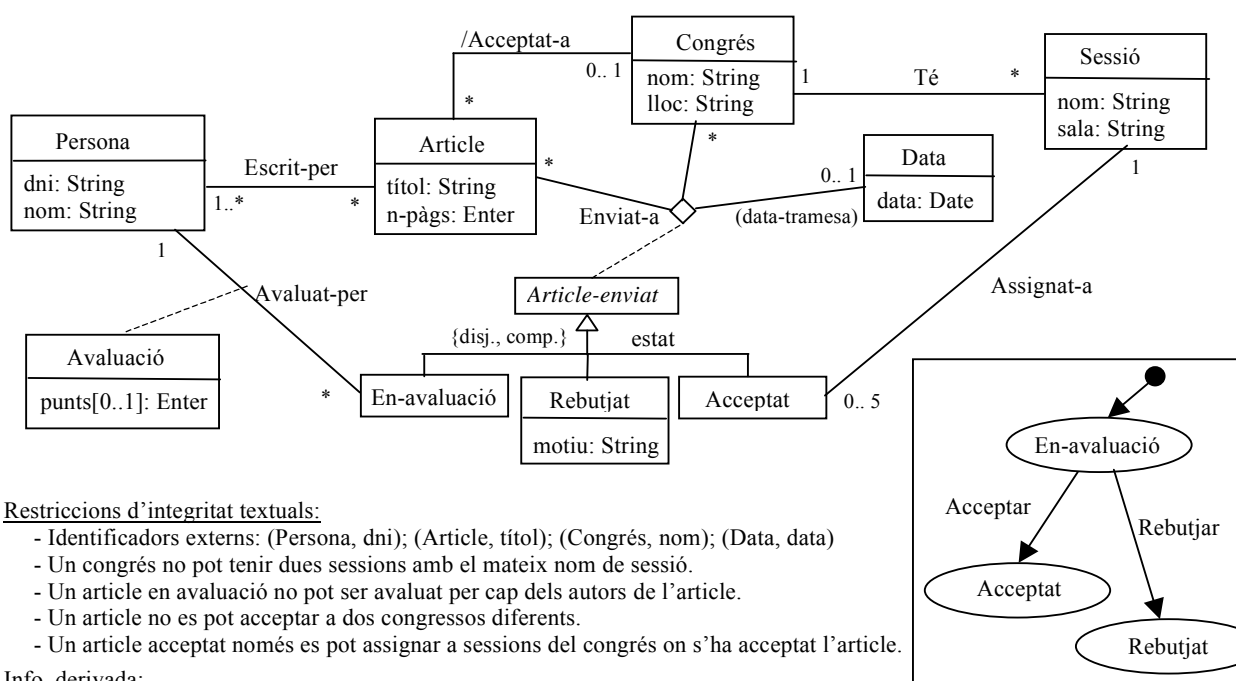
Solució alternativa de l'operació altaOficial (on s'ha definit una operació constructora perquè s'encarregui de fer el tractament de les excepcions)





5. Una empresa especialitzada en l'organització de congressos ens ha demanat que li dissenyem una part d'un sistema software per gestionar el procés de selecció dels articles que són enviats a un congrés. Els articles s'identifiquen per títol i tenen un cert nombre de pàgines. Un article és escrit per diverses persones (identificades per dni i de qui se'n sap el nom) i es pot enviar a un congrés (identificat per nom i de qui es guarda també el lloc on es farà) en una certa data. Inicialment, un article enviat a un congrés està en avaluació. L'avaluació la fa una persona que ha d'assignar una puntuació a l'article. Un article en avaluació pot ser rebutjat o acceptat. En cas que es rebutgi, el sistema enregistra el motiu de no acceptació. Si s'accepta, es guarda la sessió del congrés a la que s'assigna l'article.

Esquema conceptual d'especificació:



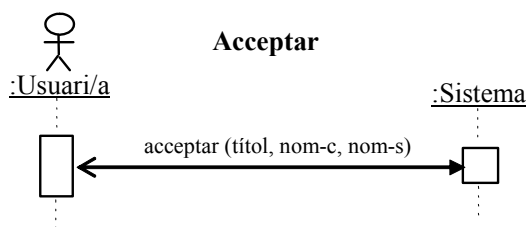
Restriccions d'integritat textuals:

- Identificadors externs: (Persona, dni); (Article, títol); (Congrés, nom); (Data, data)
- Un congrés no pot tenir dues sessions amb el mateix nom de sessió.
- Un article en avaluació no pot ser avaluat per cap dels autors de l'article.
- Un article no es pot acceptar a dos congressos diferents.
- Un article acceptat només es pot assignar a sessions del congrés on s'ha acceptat l'article.

Info. derivada:

- *Acceptat-a* permet obtenir el congrés al que s'ha acceptat un article i els articles acceptats a un congrés.

Diagrama de seqüència d'esdeveniments del sistema:



Contracte de l'operació *acceptar*:

Operació: acceptar (títol:String, nomCong:String, nomSessió:String)

Semàntica: l'article títol és acceptat al congrés nomCong i s'assigna a la sessió nomSessió.

Pre:

- L'article títol és un article en avaluació al congrés nomCong i té una puntuació superior a 5.

Post:

- L'article en avaluació passa a ser un article acceptat. Es crea l'associació Assignat-a entre l'article acceptat i la sessió nomSessió del congrés nomCong. S'elimina l'associació Avaluat-per de l'article que deixa d'estar en avaluació.

Tenint en compte que:

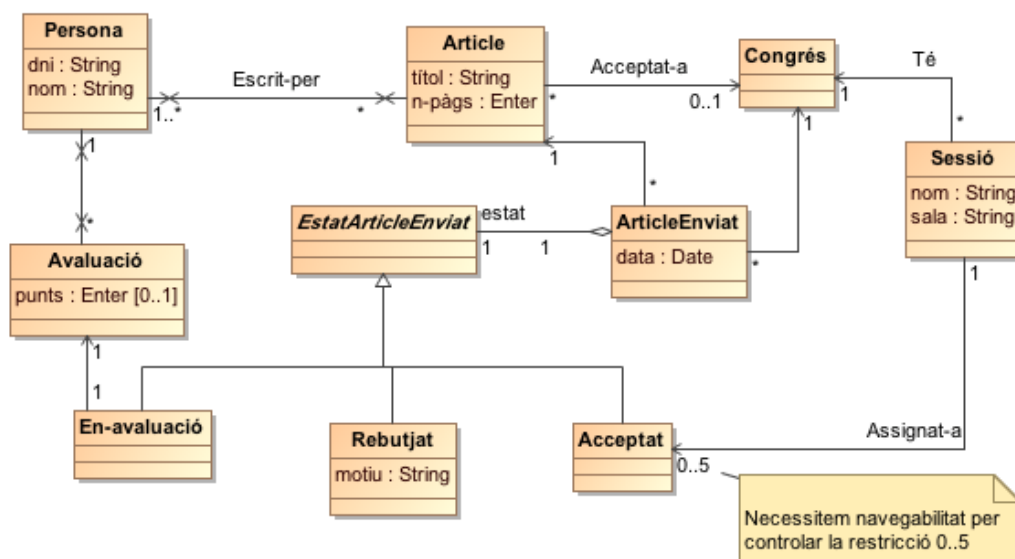
- L'associació *Acceptat-a* ha de ser materialitzada.

Es demana:

- Diagrama de classes de disseny obtingut a partir de l'esquema conceptual de les dades, indicant explícitament les restriccions d'integritat que apareixen o desapareixen, i el contracte de l'operació *acceptar* obtingut com a conseqüència de la traducció de l'esquema d'especificació al de disseny
- Diagrames de seqüència de l'operació del sistema *acceptar* i de totes les operacions que siguin invocades en aquesta operació. Cal que indiqueu als mateixos diagrames de seqüència la **signatura completa de les operacions**, que digueu si són **abstractes/ganxo** i que poseu **comentaris** de tot el que no hi aparegui de forma explícita. Indiqueu al diagrama de classes de l'apartat a) la **navegabilitat** resultant del vostre disseny.

Solució:

- Diagrama de classes de disseny obtingut a partir de l'esquema conceptual de les dades, indicant explícitament les restriccions d'integritat que apareixen o desapareixen, i el contracte de l'operació *acceptar* obtingut com a conseqüència de la traducció de l'esquema d'especificació al de disseny**



Restriccions d'integritat que cal afegir:

RI5: Un article no es pot enviar dues vegades a un congrés (0..1 ternària)

Un article no es pot enviar dues vegades a un congrés en una data concreta (estructural ternària) (subsumida per RI5)

No pot haver dues instàncies d'Avaluació amb els mateixos Persona i En-avaluació (estructural Avaluació) (subsumida per l'1 al costat Avaluació)

Context CapaDeDomini::acceptar(titol:String, nom-c:String, nom-s:String)

exc: article-enviat-no-existeix (part de la pre original)

exc: article-enviat-no-en-avaluació (part de la pre original)

exc: puntuació-insuficient (part de la pre original)

exc: sessió-completa (0..5)

exc: article-ja-acceptat (RI3)

exc: sessió-no-existeix (RI4)

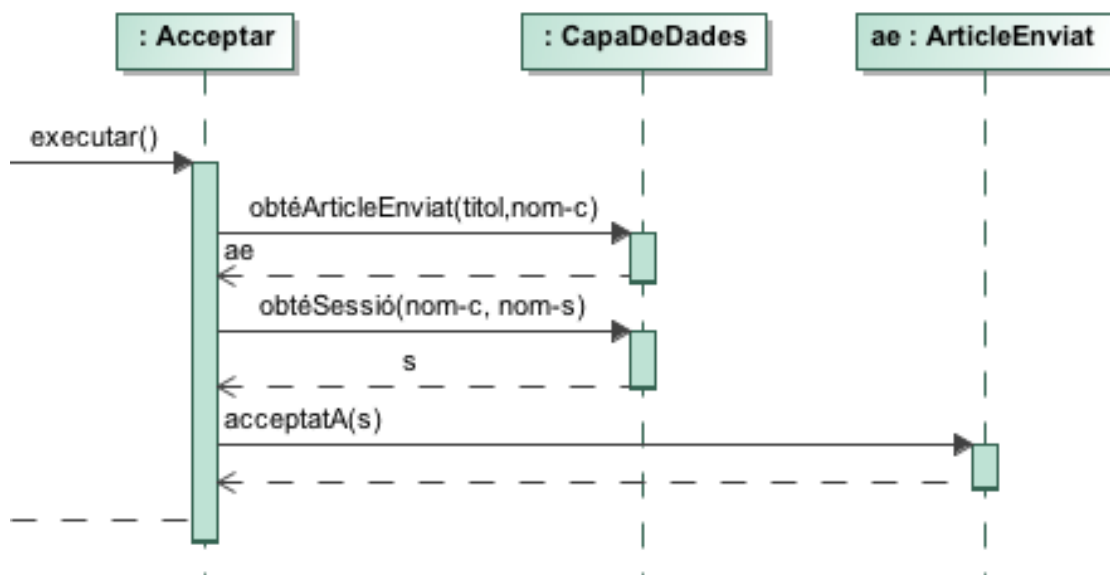
post: (original) L'article en avaluació passa a ser un article acceptat. Es crea l'associació Assignat-a entre l'article acceptat i la sessió nomSessió del congrés nomCong. S'elimina l'associació Avaluat-per de l'article que deixa d'estar en avaluació.

post: S'associa l'article al congrés a través de l'associació Acceptat-a"

NOTA: En aquest cas, hem decidit dividir la precondició original en tres precondicions abans de fer el pas al contracte de disseny. D'aquesta manera, el contracte de disseny podrà distingir per quin dels tres motius (no existeix l'article enviat, l'article enviat no està en avaluació o l'avaluació no té prou punts) ha fallat l'operació.

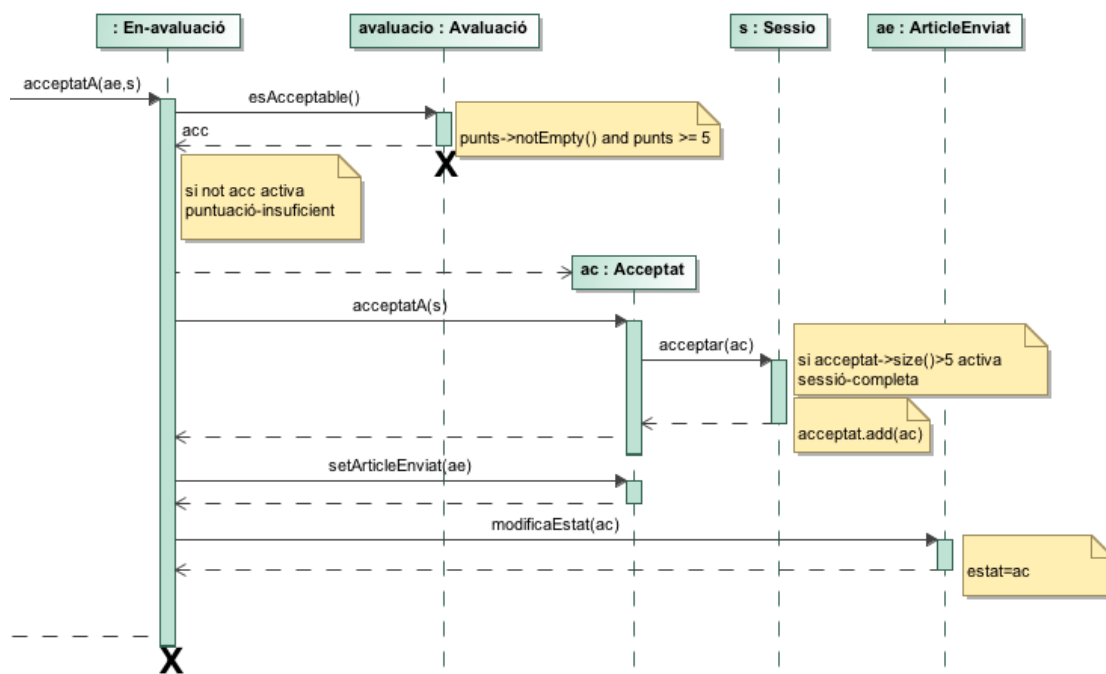
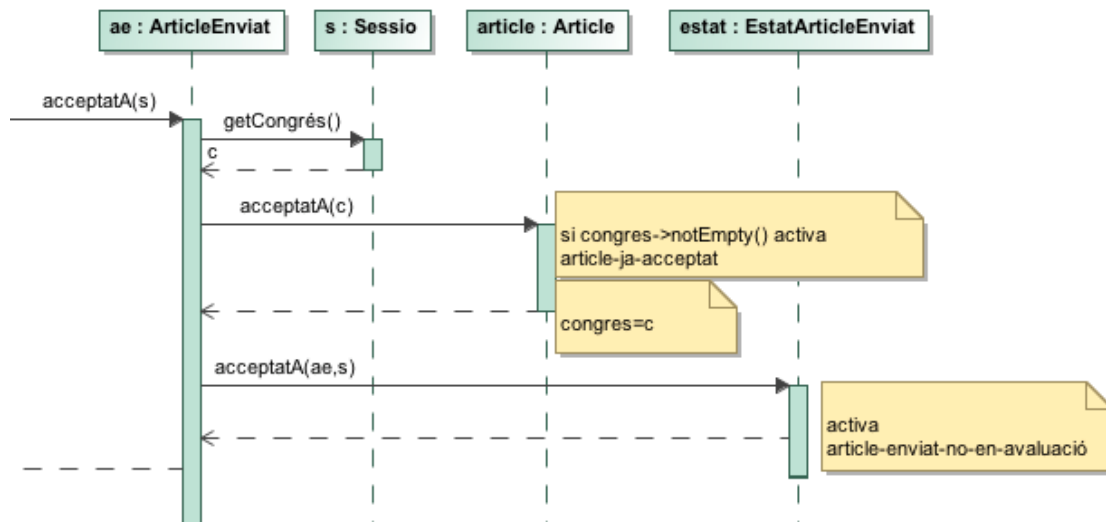
Hagués estat igualment correcte tenir una única excepció que cobris tots tres casos. Això sí, sempre cal tenir en compte que el disseny ha de ser coherent amb el contracte i, per tant, ha de comprovar els mateixos casos d'excepcions amb les mateixes excepcions que apareixen al contracte.

- b) **Diagrames de seqüència de l'operació del sistema *acceptar* i de totes les operacions que siguin invocades en aquesta operació. Cal que indiqueu als mateixos diagrames de seqüència la signatura completa de les operacions, que digueu si són abstractes/ganxo i que poseu comentaris de tot el que no hi aparegui de forma explícita. Indiqueu al diagrama de classes de l'apartat a) la navegabilitat resultant del vostre disseny**



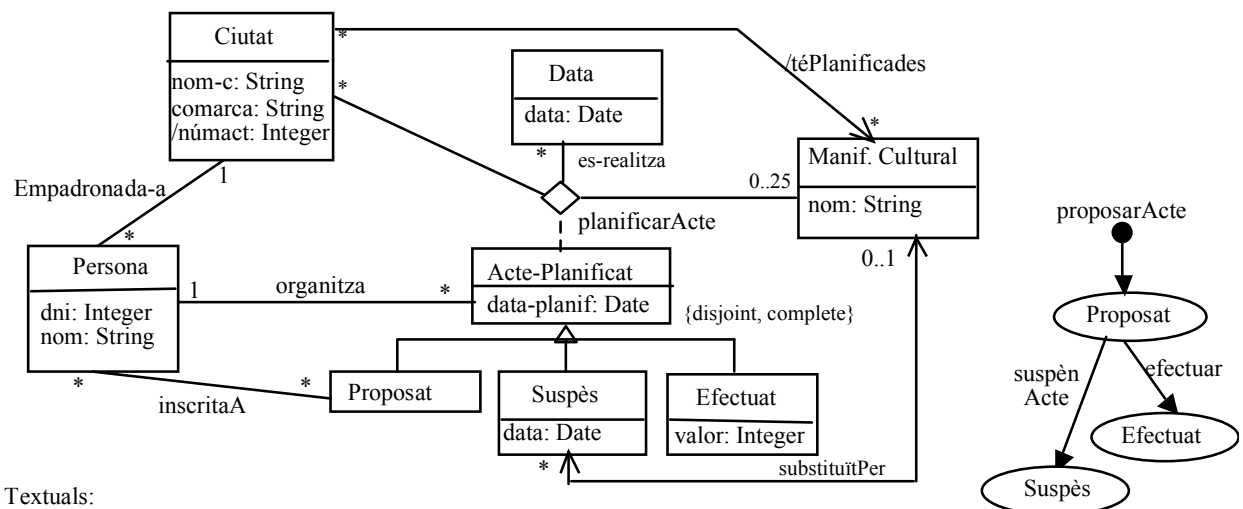
NOTA: Podriem haver passat el nom de la sessió en comptes d'anar-lo a buscar a la capa de dades però no ens haguéssim estalviat cap acoblament. A canvi, hauriem tingut una solució més llarga ja que si només tenim el nom llavors hem de buscar la sessió entre les sessions del congrés iterant per la col·lecció de sessions associades al congrés. En aquest cas ens hagués aparegut una navegabilitat més (de Congrés a Sessió)

NOTA: En aquest diagrama no hem capturat cap de les dues excepcions que ens pot activar la capa de dades (article-enviat-no-existeix i sessió-no-existeix) degut al fet que el nostre contracte ens permet propagar-les.



Cal tenir en compte que no hi ha navegabilitat de Persona a Avaluació (i per això podem destruir l'instància d'Avaluació sense notificar a la Persona a la qual pertany l'avaluació) i que tampoc hi ha navegabilitat d'Acceptat a Sessió (si fos el cas, caldria afegir "sessió=s" en algun punt del constructor d'Acceptat).

6. Una agrupació de ciutats ens ha demanat que li dissenyem una part d'un sistema software per gestionar els actes culturals que s'hi planifiquen. L'agència disposa d'informació de les ciutats que volen planificar actes, identificades per nom-c; i de les manifestacions culturals que es poden planificar. Un acte es planifica a una ciutat, per una determinada manifestació i per a una certa data, i s'enregistra també el dia en què s'ha fet la planificació de l'acte i la persona que organitza l'acte. Les persones s'identifiquen per dni i se'n coneix també la ciutat d'empadronament. Mentre l'acte està proposat, el sistema guarda informació de les persones que s'hi inscriuen. L'acte es pot acabar suspenent o bé efectuant. En el primer cas, es pot programar una manifestació cultural que substitueixi la manifestació cultural de l'acte que se suspèn i que es farà a la mateixa ciutat i data de l'acte suspès. En el segon cas, cal enregistrar la valoració que es fa de l'acte.



- Claus classes no associatives: (Persona, dni); (Data, data); (Ciutat, nom-c); (Manif.Cultural, nom)
- La persona que organitza un acte ha d'estar empadronat a la ciutat on es fa l'acte
- Data de realització d'un acte planificat > data de suspensió (si n'hi ha) > data planificació de l'acte
- Un acte planificat no es pot suspendre si entre els seus inscrits hi ha la persona que l'organitza

- númact: és el nombre d'actes proposats a aquella ciutat
- téPlanificades: una Ciutat té planificades un conjunt de Manifestacions Culturals

Diagrama de seqüència d'esdeveniments del sistema:

Contracte de l'operació suspènActe:

Operació: suspènActe (nomCiutat: String, dni: Integer, nomM1:String, dataActe: Date, nomM2: String, dataSusp: Date)

Semàntica: Suspendre el dia d-susp un acte proposat i assignar-li una manifestació substitutiva.

Precondicions: Existeix un acte planificat *a*, proposat, per a la ciutat nomCiutat en la data dataActe de la manifestació nomM1 i dni és l'organitzador de l'acte *a*.

Postcondicions:

- L'acte planificat passa de Proposat a Suspès, amb data=dataSusp, i s'eliminen les seves associacions Inscrita-a amb Persona.
- Si nomM2 no és nul, es crea una nova ocurrència de l'associació *substituïtPer* entre l'acte suspès i la manifestació nomM2.

Contracte de l'operació llistarManifs:

Operació: llistarManifs (nomCiutat:String, dni:Integer): Set(String)

Semàntica: Es retorna una llista amb el nom de les manifestacions que tenen algun acte planificat a nom-c, que està organitzat per dni i que té més de 100 inscrits que no són de nom-c.

Precondicions: La persona dni està empadronada a la ciutat nomCiutat.

Postcondicions:

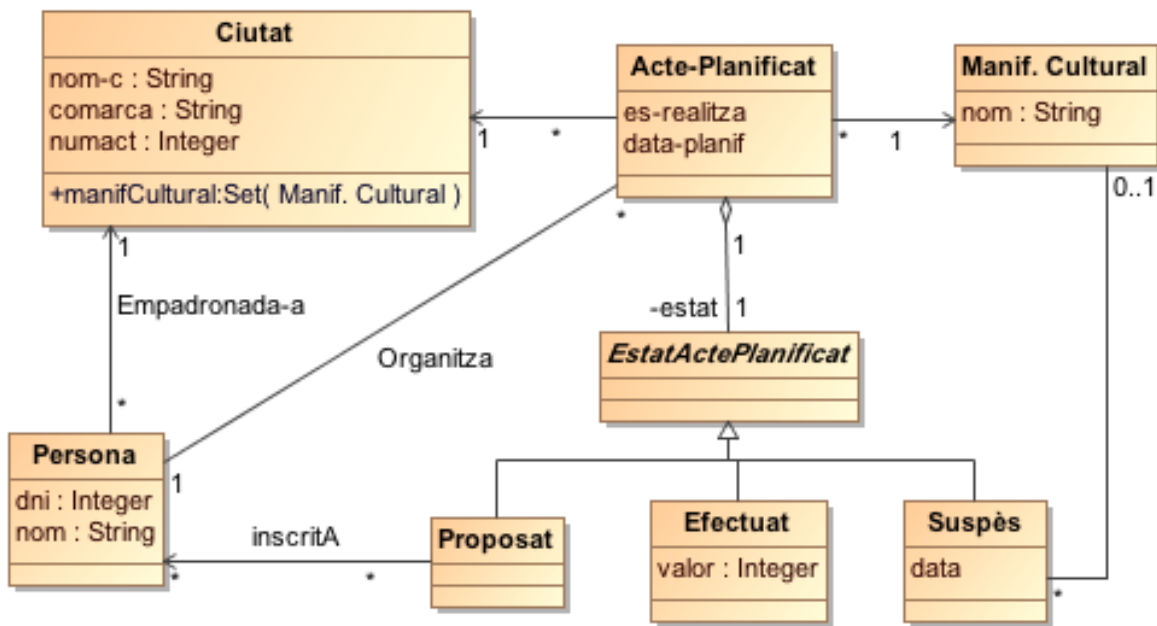
- Es retorna una llista que conté el nom de totes les activitats que tenen algun acte planificat a la ciutat nomCiutat, que està organitzat per dni i que té més de 100 inscrits que no estan empadronats a la ciutat nomCiutat.

Suposant que:

- L'associació *téPlanificades* és calculada i té la navegabilitat indicada a l'esquema conceptual. L'atribut *númact* és materialitzat. L'associació *substituïtPer* té navegabilitat doble.

Es demana:

- a) Diagrama de classes de disseny obtingut a partir de l'esquema conceptual de les dades, indicant explícitament les restriccions d'integritat que apareixen o desapareixen, i els contractes de les operacions *suspènActe* i *llistarManifs* obtinguts com a conseqüència de la traducció de l'esquema d'especificació al de disseny



NOTA: La navegabilitat de Manif. Cultural a Suspès és doble. La d'Organitza també és doble.

Restriccions d'integritat afegides:

RI4: No es poden planificar dos actes amb els mateixos (Ciutat, Manfí. Cultural i Data) (estructural ternària)

RI5: No es poden planificar més de 25 actes a la mateixa ciutat en una data concreta (0..25)

Contractes:

Context CapaDeDomini::suspènActe(nomCiutat:String, dni:Integer, nomM1:String, dataActe:Date, nomM2:String, dataSusp:Date)

exc: acte-planificat-no-existeix (pre original: Existeix un acte planificat a per a la ciutat nomCiutat en la data dataActe de la manifestació nomM1)

exc: acte-no-proposat (pre original: l'acte a és proposat)

exc: dni-no-organitzador (pre original: dni és l'organitzador de l'acte a)

exc: no-existeix-manifestació (estructural, si nomM2 no és nul)

exc: organitzador-inscrit (RI3)

exc: data-incorreccta (RI2)

post: (postcondicions originals)

post: Es decremента el valor de numact a la ciutat amb nom nomCiutat

Context CapaDeDomini::l·listarManifs(nomCiutat:String, dni:Integer):Set(String)

exc: no-empadronada (pre original)

exc: persona-no-existeix (pre original)

exc: ciutat-no-existeix (pre original)

post: (post original)

- b) Diagrames de seqüència de de les operacions del sistema *suspènActe* i *l·listarManifs* i de totes les operacions que siguin invocades en aquesta operació. Cal que indiqueu als mateixos diagrames de seqüència la **signatura completa de les operacions**, que digueu si són **abstractes/ganxo** i que poseu **comentaris** de tot el que no hi aparegui de forma explícita. Indiqueu al diagrama de classes de l'apartat a) la **navegabilitat** resultant del vostre disseny.

Comencem per l'operació de consulta per tal de veure quines navegabilitats necessitem.

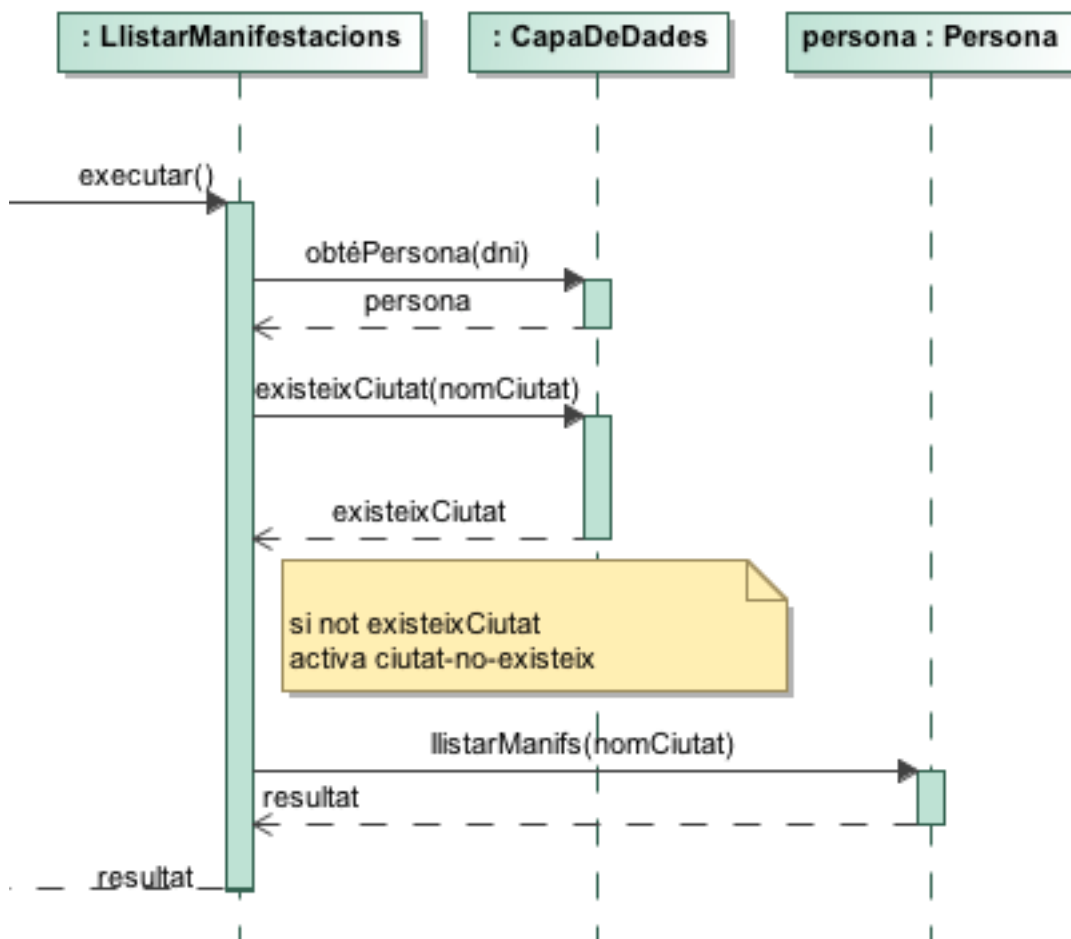
Context CapaDeDomini::l·listarManifs(nomCiutat:String, dni:Integer):Set(String)

exc: no-empadronada

exc: persona-no-existeix

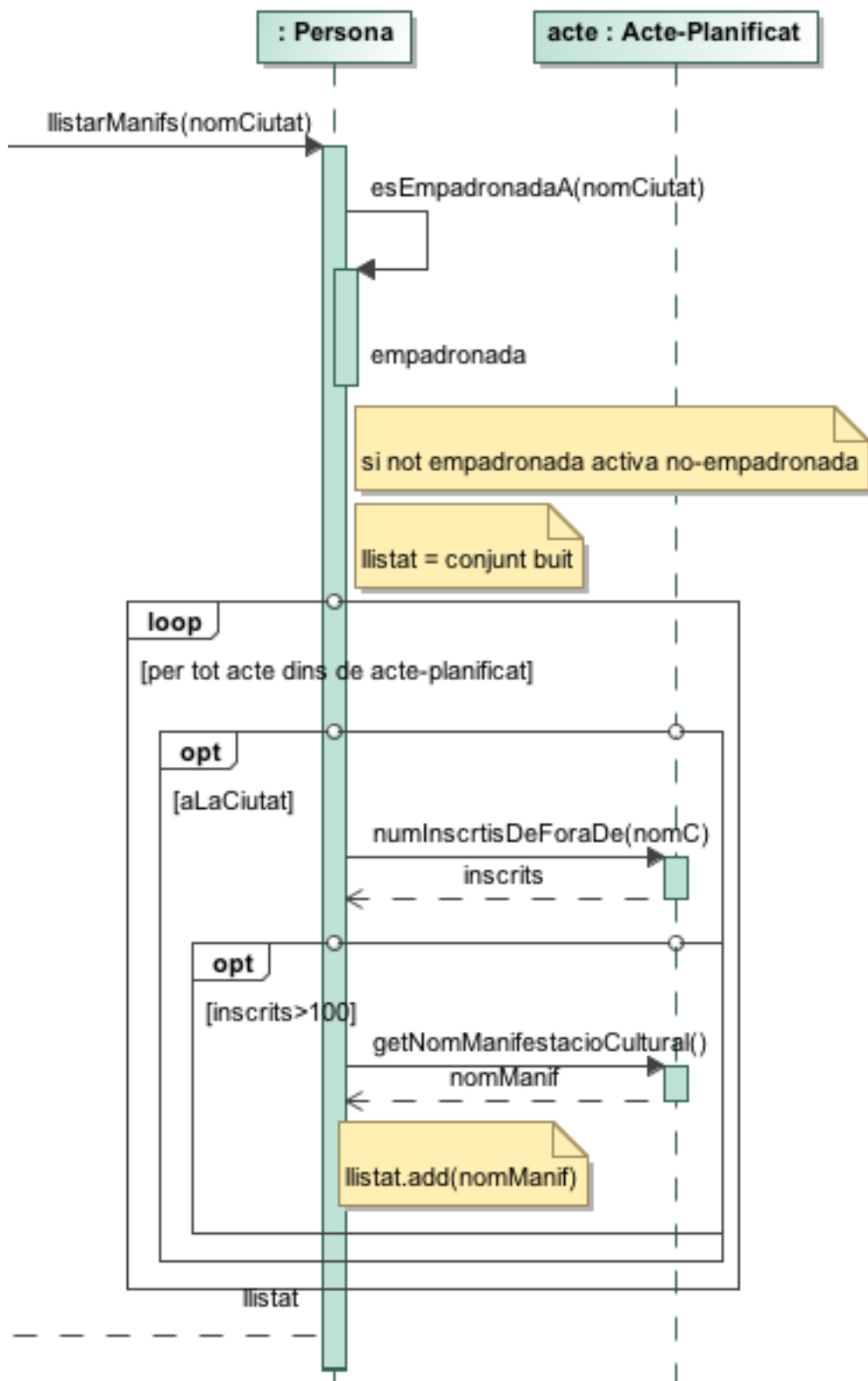
exc: ciutat-no-existeix

post: Es retorna una llista que conté el nom de totes les activitats que tenen algun acte planificat a la ciutat nomCiutat, que està organitzat per dni i que té més de 100 inscrits que no estan empadronats a la ciutat nomCiutat.

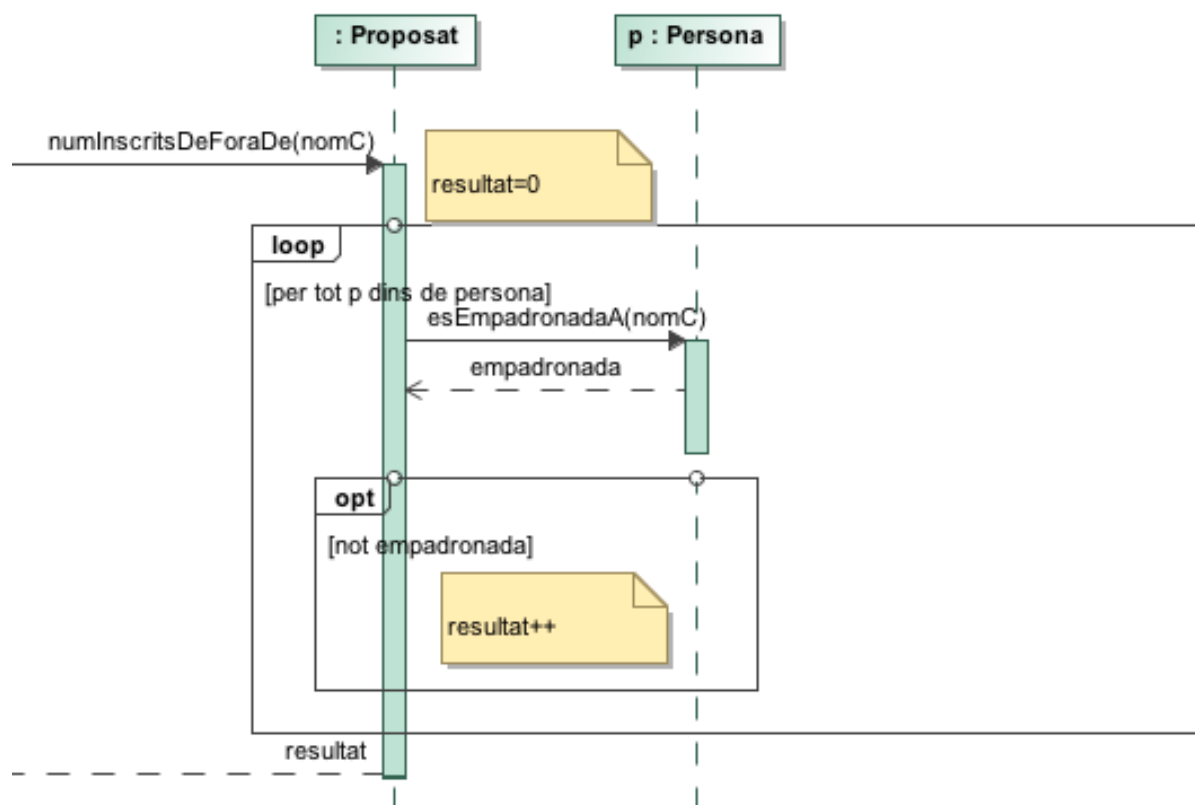
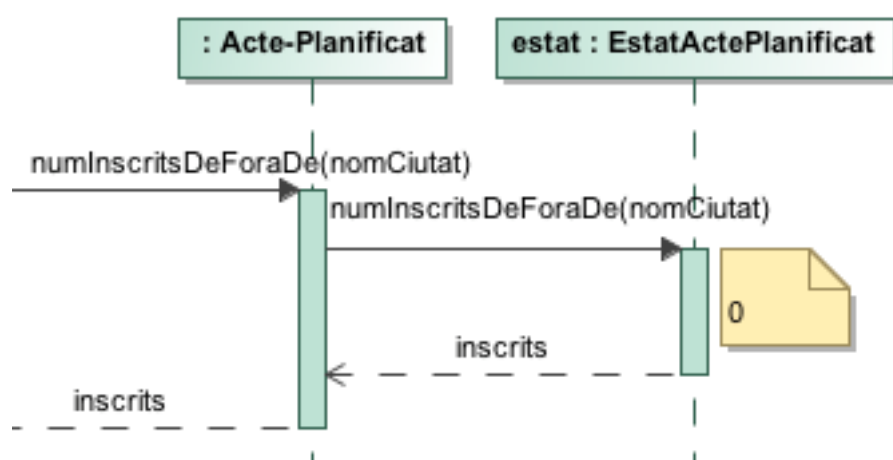
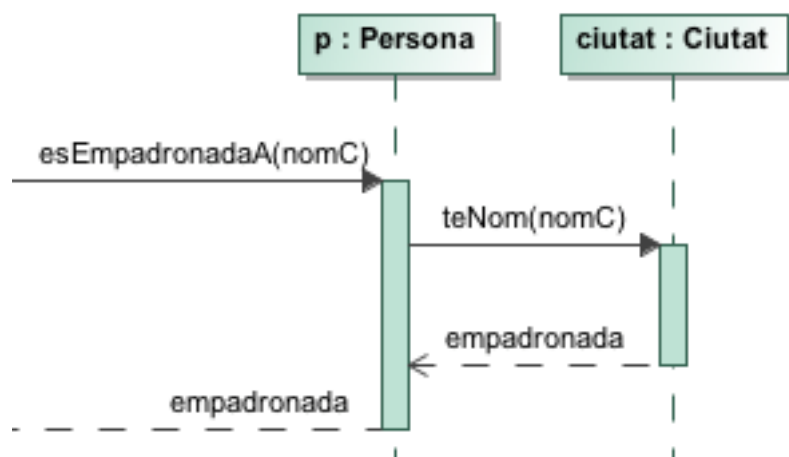


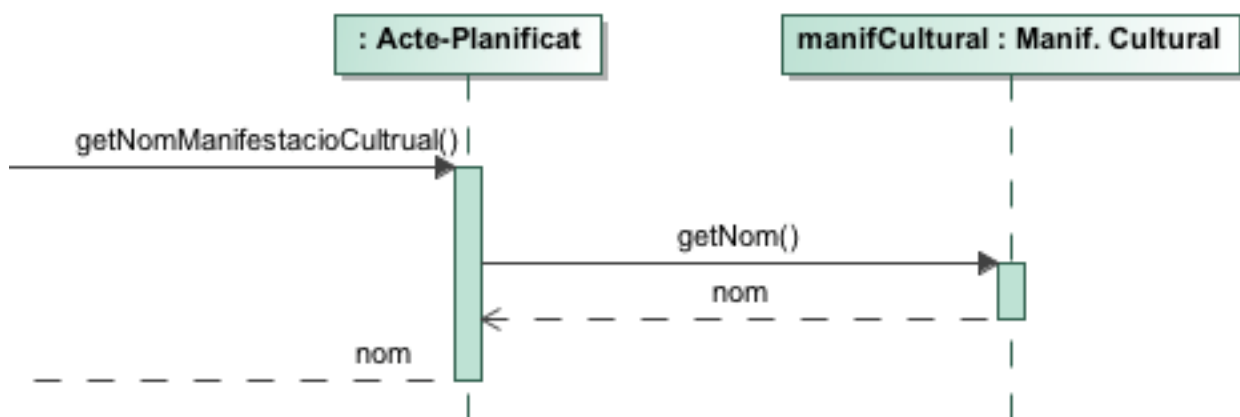
Podriem haver anat a buscar totes les manifestacions culturals i, per cada manifestació cultural, veure si té algun acte planificat que complís els criteris.

També podriem haver començat per la ciutat i obtenir la manifestació cultural dels actes que organitza i compleixen els criteris.



NOTA: Com que tenim una restricció d'integritat que diu que "La persona que organitza un acte ha d'estar empadronada a la ciutat on es fa l'acte", no cal que verifiquem si l'Acte està planificat o no a la ciutat (nomCiutat) ja que aquesta restricció d'integritat ens assegura que és així.





Context CapaDeDomini::suspènActe(nomCiutat:String, dni:Integer, nomM1:String, dataActe:Date, nomM2:String, dataSusp:Date)

exc: acte-planificat-no-existeix (pre original: Existeix un acte planificat a per a la ciutat nomCiutat en la data dataActe de la manifestació nomM1)

exc: acte-no-proposat (pre original: l'acte a és proposat)

exc: dni-no-organitzador (pre original: dni és l'organitzador de l'acte a)

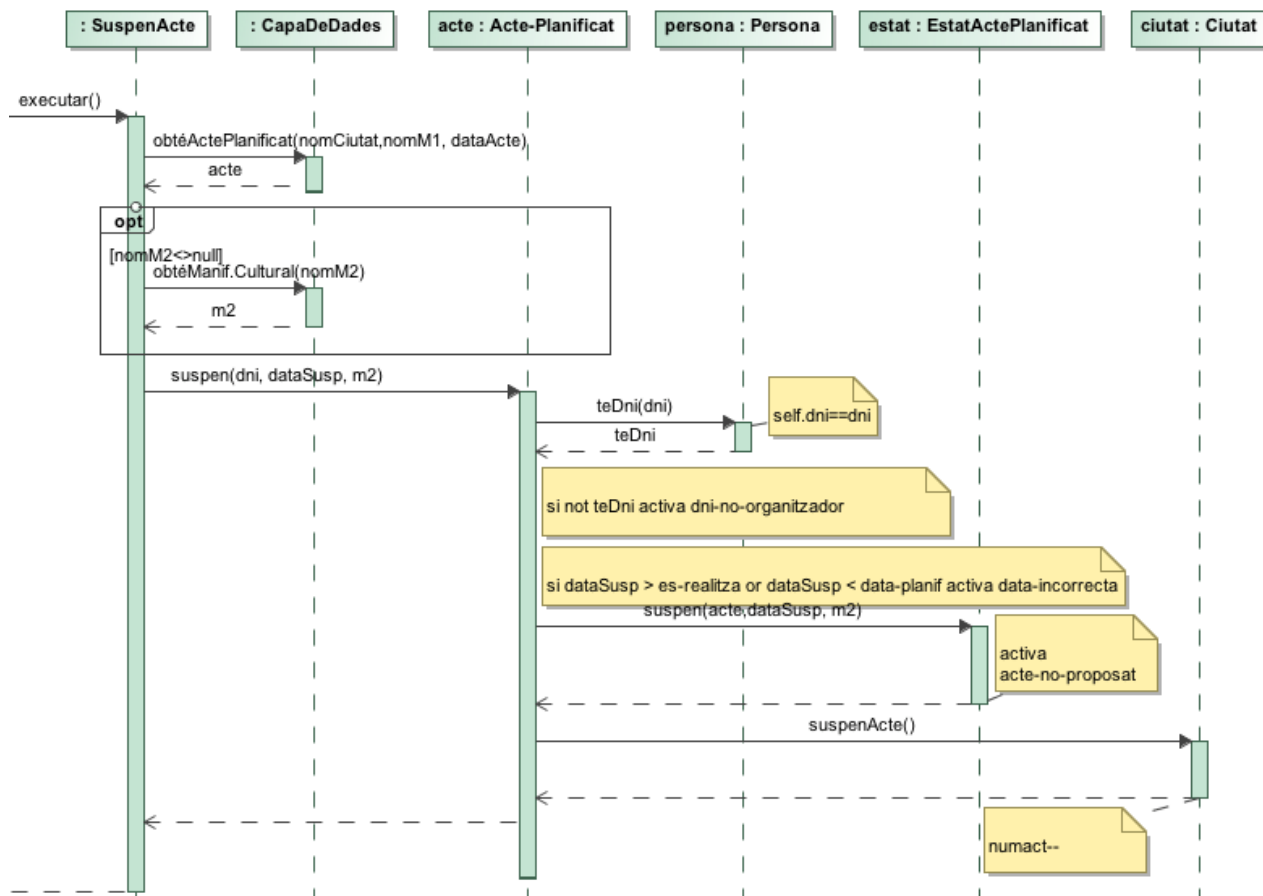
exc: no-existeix-manifestació (estructural, si nomM2 no és nul)

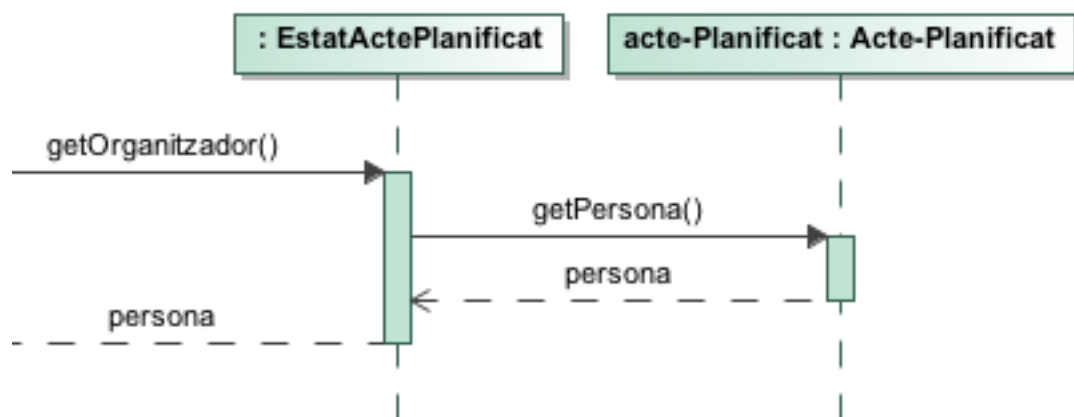
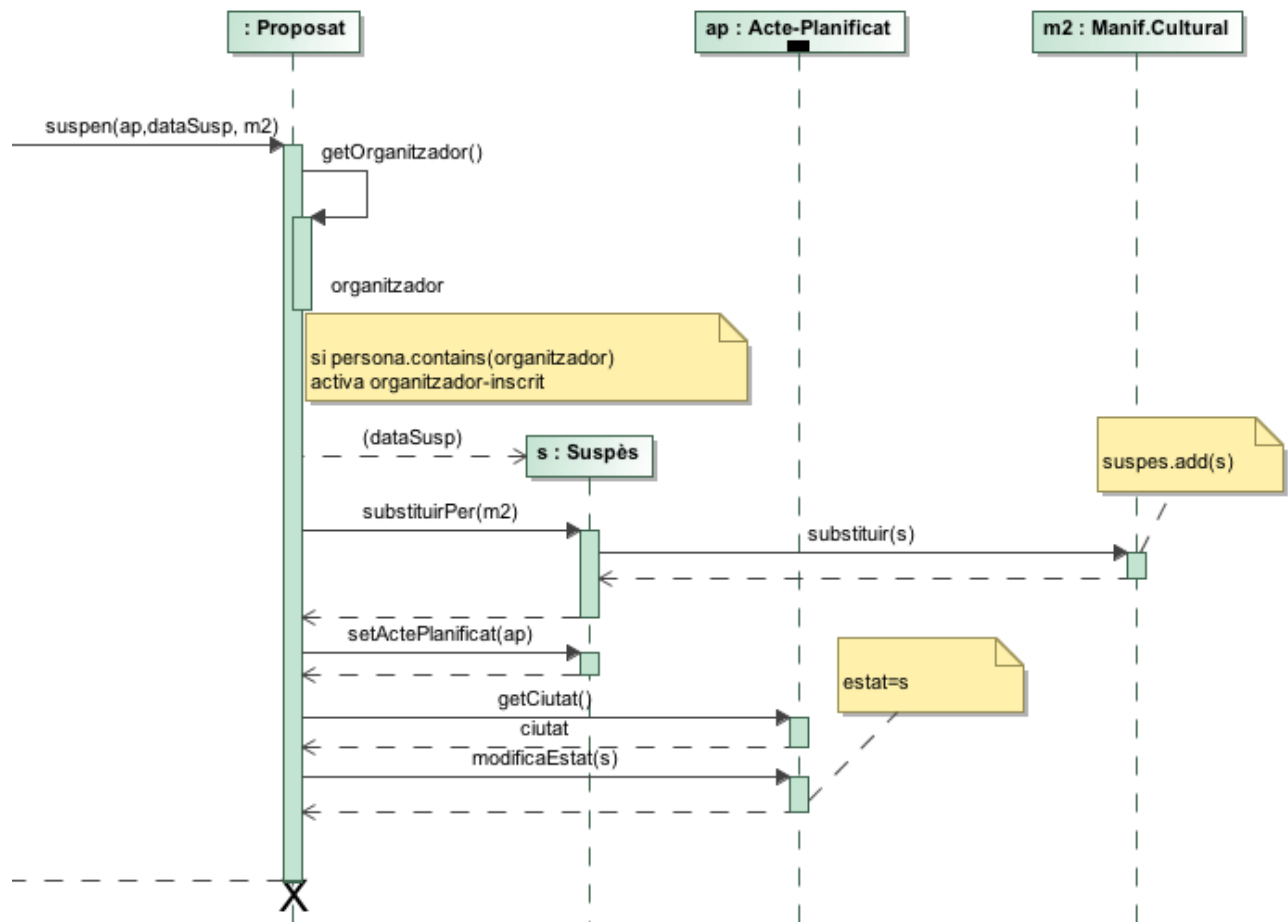
exc: organitzador-inscrit (RI4)

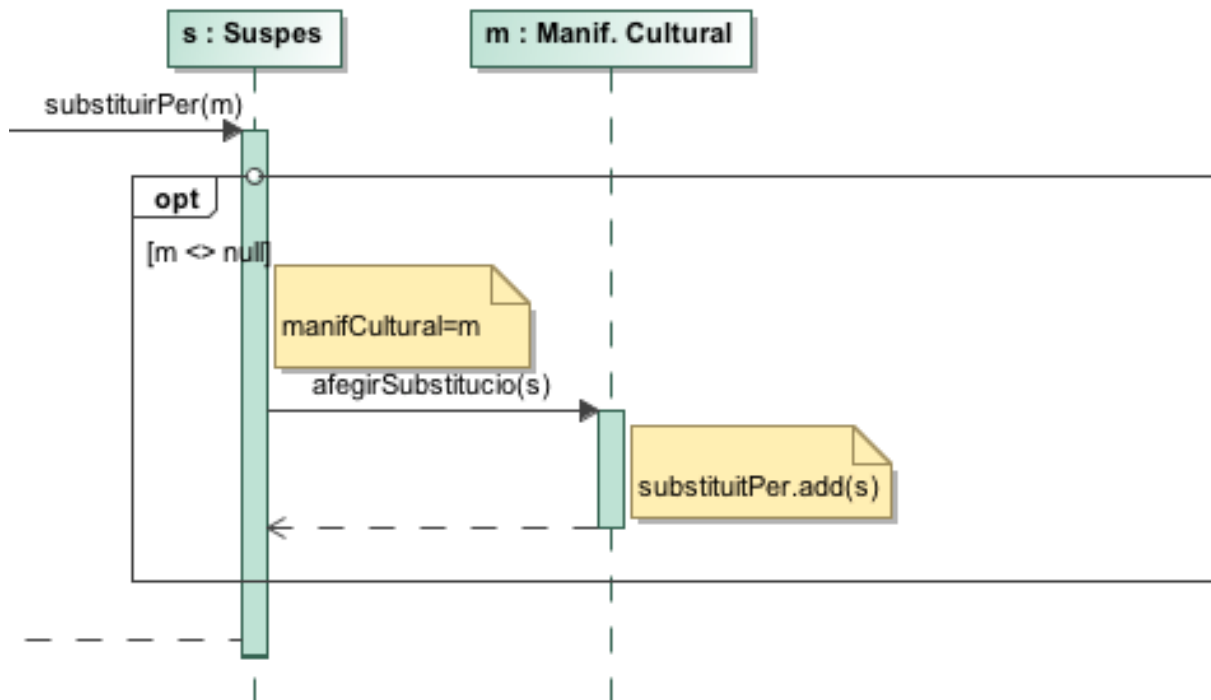
exc: data-incorrecata (RI2)

post: (postcondicions originals)

post: Es decrementa el valor de numact a la ciutat amb nom nomCiutat







NOTA: Si haguéssim volgut tenir en compte la tercera restricció d'integritat (una manifestació cultural no pot substituir un acte suspès si ja hi ha algun acte planificat d'aquesta manifestació a les mateixes ciutats i data de l'acte suspès) l'operació **aefegirSubstitucio(s)** s'hagués dissenyat de manera que es controlés l'esmentada condició.