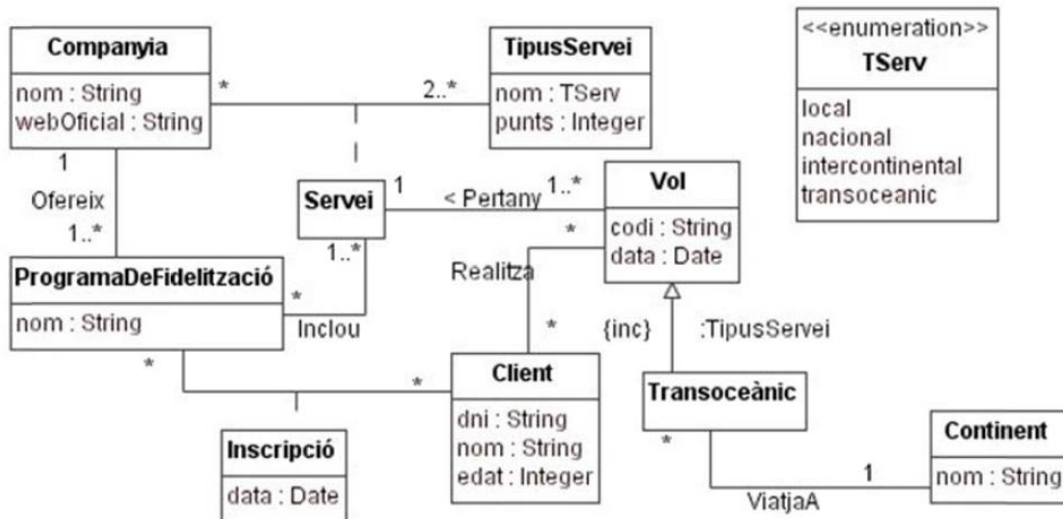


## 7è problema d'IES – 20 o 21 d'abril del 2015 – QP1415

Considereu un sistema per un consorci de companyies aèries. Les companyies associades disposen de diversos tipus de servei (local, nacional, intercontinental i transoceànic) i ofereixen programes de fidelització als seus clients per a què s'hi inscriguin. S'enregistren també els vols que pertanyen a un servei i els clients que realitzen els vols. La part rellevant de l'esquema conceptual de les dades en UML es mostra a la figura següent:



### Restriccions textuais

1. Claus externes: (Companyia, nom), (TipusServei, nom), (Continent, nom), (Vol, codi+ data), (Client, dni)
2. Una companyia no pot oferir dos o més programes de fidelització amb el mateix nom
3. Els serveis inclosos a un programa de fidelització són de la companyia que ofereix el programa
4. Un client no es pot inscriure a més d'un programa de fidelització de la mateixa companyia

Suposarem que el sistema a desenvolupar no ha de donar d'alta instàncies de les classes Companyia, TipusServei, ProgramaDeFidelització, Continent ni Client atès que hi ha un altre sistema que ho fa. El sistema ha de permetre efectuar les funcionalitats següents: alta de servei i vols, alta d'inscripció i consulta de clients sènior actius.

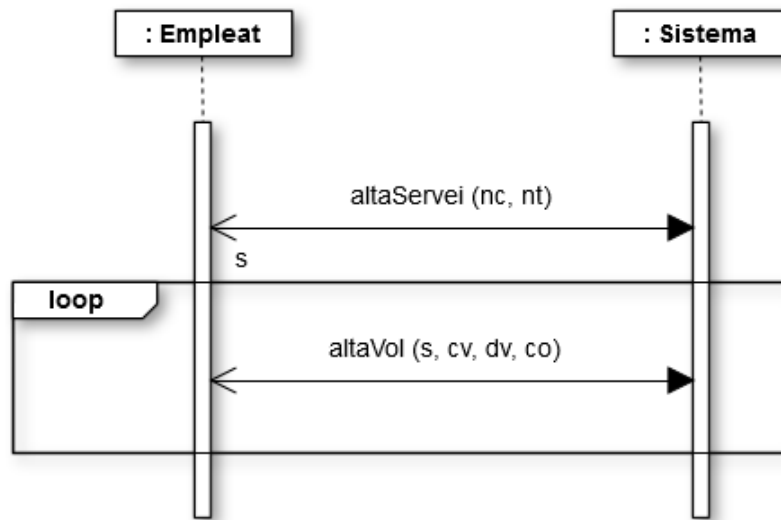
**AltaServeiIvols:** Quan un directiu vol donar d'alta un servei i els seus vols li demana a un empleat del consorci que ho faci. L'empleat indica al sistema la informació necessària per donar d'alta un servei. A continuació, per cada vol que ha de pertànyer al servei, l'empleat dona la informació necessària per donar d'alta el vol. Si es tracta d'un vol transoceànic, indica també el continent on viatja. Cal considerar que aquesta funcionalitat només es pot realitzar si la companyia del servei té menys de 20 programes de fidelització. Feu que la interacció per portar a terme aquesta funcionalitat requereixi més d'un esdeveniment.

**AltaInscripció:** Quan un client es vol inscriure a un programa de fidelització, ell mateix indica al sistema la informació necessària per identificar el programa de fidelització, el seu dni i la data de la inscripció. Només es pot fer aquesta funcionalitat si el client té com a mínim un vol que pertanyi a un servei inclòs al programa de fidelització on es vol inscriure i si la data de la inscripció és posterior o igual a l'actual. Feu que la interacció requereixi un sol esdeveniment.

**ConsultaClientsSèniorActius:** Quan un empleat vol fer una consulta de clients sènior actius, ell mateix introdueix al sistema un nom de companyia. El sistema mostra el llistat dels clients que tenen més de 40 anys i que han fet més de 3 vols transoceànics que pertanyen a serveis de la companyia donada. El llistat resultant mostra per cada client el dni i les dates de tots els vols que té enregistrats al sistema. Aquesta funcionalitat només s'executa si la companyia existeix.

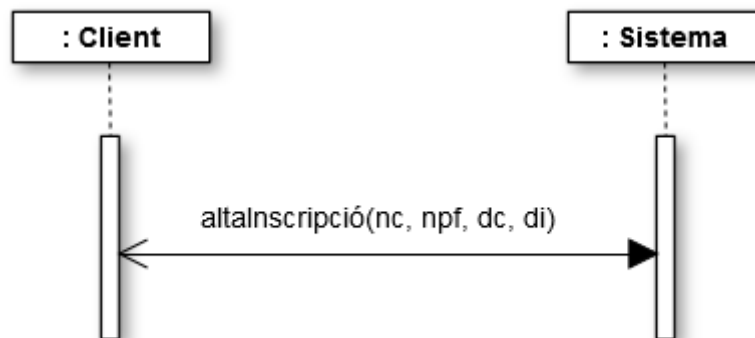
Es demana que feu mitjançant la notació UML:

- Diagrames de seqüència i contractes en OCL de les operacions corresponents a les funcionalitats anteriors.



**context:** Sistema::altaServei(nc:String, nt:TServ):Servei  
**pre:** Companyia.allInstances()->exists(c|c.nom=nc and c.programaDeFidelització->size()<20) and TipusServei.allInstances()->exists(t|t.nom=nt)  
**post:** Servei.allInstances()->exists(s|s.oclIsNew() and s.companyia.nom=nc and s.tipusServei.nom=nt and result=s)

**context:** Sistema::altaVol(s:Servei, cv:String, dv:Date, co:String)  
**pre:** s.tipusServei.nom=TServ::transoceànic implies Continent.allInstances()->exists(c|c.nom=co)  
**post:** Vol.allInstances()->exists(v|v.oclIsNew() and v.servei=s and v.codi=cv and v.data=dv and if s.tipusServei.nom=TServ::transoceànic then v.oclAsType(Transoceànic).continent.nom=co endif)



**context:** Sistema::altaInscripció(nc:String, npf:String, dc:String, di:Date)  
**pre:** di>=today() and Client.allInstances->select(cl|cl.dni=dc).vol.servei.programaDeFidelització->includes(Companyia.allInstances()->select(c|c.nom=nc).programaDeFidelització->select(pf|pf.nom=npf)))

**post:**        `Inscripció.allInstances()->exists(i|i.oclIsNew() and  
i.programaDeFidelització.nom=npf and  
i.programaDeFidelització.companyia.nom=nc and  
i.client.dni=dc and i.data=di)`



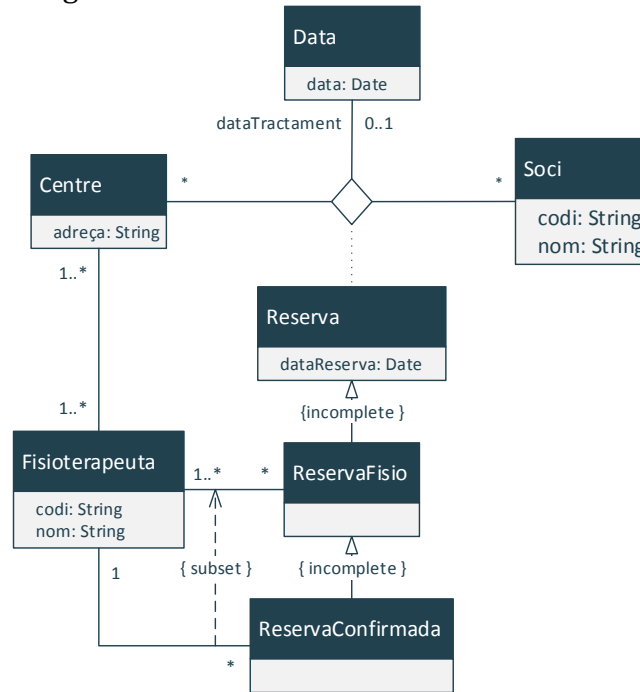
**context:**    `Sistema::consultarClients(nc:String):  
Set(TupleType(dnic: String, datesc: Set(Date)))`

**pre:**        `Companyia.allInstances()->exists(c|c.nom=nc)`

**body:**        `let clients:Set(Client)=Client.allInstances()->  
select(c|c.edat>40 and c.vol->select  
(v|v.servei.companyia.nom=nc and  
v.oclIsTypeOf(Transoceanic))->size(>3) in  
clients->collect(c|Tuple{dnic=c.dni,  
datesc=c.vol.data})`

## 8è problema d'IES – 23 o 24 d'abril del 2015 – QP1415

Un spa ofereix als seus socis la possibilitat de rebre tractaments als seus centres. Els socis poden fer reserves als centres per a rebre aquests tractaments. Per als tractaments de fisioteràpia, els socis tenen la possibilitat d'escollir un conjunt de possibles fisioterapeutes per a què un d'ells els faci el tractament. Les reserves de fisioteràpia s'han de confirmar i s'hi assigna el professional que definitivament farà el tractament. La part rellevant d'aquest esquema conceptual és la següent:



### Restriccions textuais

1. Claus externes: (Soci, codi), (Centre, adreça), (Fisioterapeuta, codi), (Data, data)
2. No pot haver més d'una reserva per a un soci a una mateixa data
3. La data de tractament ha de ser posterior a la data en què s'ha fet la reserva
4. Els fisioterapeutes seleccionats per a una reserva pertanyen al centre d'aquesta reserva

El sistema a desenvolupar no ha de donar d'alta Socis, Centres, Dates, ni Fisioterapeutes atès que hi ha un altre sistema que ho fa. No obstant, ha de ser capaç de crear una nova reserva, enregistrar la confirmació d'una reserva i llistar els fisioterapeutes que han estat demanats més cops.

Quan un soci vol fer una reserva per a un tractament indica la data en què el vol rebre i el centre. A la reserva haurà d'indicar els fisioterapeutes preferits que vol pel tractament. Feu que la interacció per portar a terme aquest cas d'ús requereixi més d'un esdeveniment.

Un coordinador s'encarregarà de confirmar una reserva encara no confirmada. En aquest cas, indica al sistema les dades necessàries per identificar la reserva que vol confirmar i el fisioterapeuta que farà el tractament. Feu que la interacció tingui un sol esdeveniment.

Quan un coordinador demana el llistat dels fisioterapeutes preferits, indica al sistema el rang de dates de la consulta. El sistema retorna, per a cada fisioterapeuta confirmat a algun tractament de qualsevol centre dins del rang de dates, el nom del fisioterapeuta i la ratio: nombre de vegades que ha estat confirmat per nombre de vegades sol·licitat (ambdós casos dins del rang de dates demanats).

Us demanem que feu mitjançant la notació UML:

- *Model del Comportament*: diagrames de seqüència de totes les funcionalitats especificades i contractes en OCL de totes les operacions que apareixen a aquests diagrames.

## **Solució**

### **Cas d'ús Nova Reserva**

**context** Sistema::novaReserva(centre: String, data: Data, soci: String, fisio: Boolean) : Reserva

**pre** existeixCentre: Centre.allInstances()->exists(c | c.adreça = centre)

**pre** existeixSoci: Soci.allInstances()->exists(s | s.codi = soci)

**post:**

```
    Reserva.allInstances()->exists(r |  
        r.ocllsNew() and r.dataReserva = avui() and  
        r.centre.adreça = centre and r.data.data = data and r.soci.codi = soci and  
        fisio implies r.ocllsTypeOf(ReservaFisio) and  
        result = r  
    )
```

**context** Sistema::solicitarFisio(r: ReservaFisio, fisio: String)

**pre** existeixFisio: Fisioterapeuta.allInstances()->exists(f | f.codi = fisio)

**post:**

```
    r.fisioterapeuta->includes(Fisioterapeuta.allInstances()->select(f | f.codi = fisio))
```

### **Cas d'ús Confirmar Reserva Fisio**

**context** Sistema::obtenirReserva(centre: String, data: Data, soci: String, fisio: String)

**pre** reservaÉsFisioNoConfirmada:

```
    ReservaFisio.allInstances()->exists(rf |  
        rf.centre.adreça = centre and rf.data.data = data and rf.soci.codi = soci and  
        not rf.ocllsTypeOf(ReservaConfirmada)  
    )
```

**pre** existeixFisio: Fisioterapeuta.allInstances()->exists(f | f.codi = fisio)

**post:**

```
    ReservaConfirmada.allInstances()->exists(rc |  
        rc.centre.adreça = centre and rc.data.data = data and rc.soci.codi = soci and  
        rc.fisioTractament = Fisioterapeuta.allInstances()->select(f | f.codi = fisio)  
    )
```

### Cas d'ús Llistar Fisios Preferits

**context** Sistema::llistarFisiosPreferits(dataInici: Data, dataFi: Data)  
: Set(TupleType(fisio: String, ratio: Number))

**pre** datesCoherents: dataFi >= dataInici

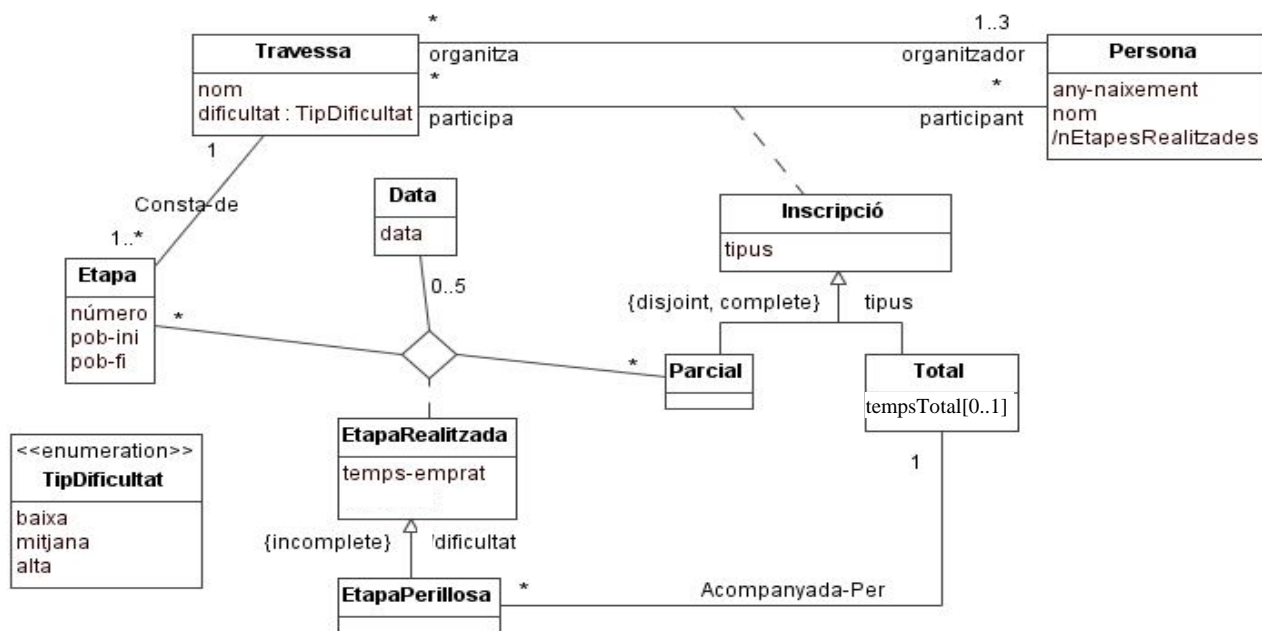
**body:**

```
let fisios = Fisioterapeuta.allInstances->select(f |  
    f.reservaConfirmada->select(rc |  
        rc.data.data >= dataInici and rc.data.data <= dataFi  
    )->size() > 0  
)  
in  
    fisios->collect(f | Tuple  
        { fisio = f.nom,  
          ratio =  
              f.reservaConfirmada->select(rc |  
                  rc.data.data >= dataInici and rc.data.data <= dataFi  
              )  
            /  
              f.reservaFisio->select(rf |  
                  rf.data.data >= dataInici and rf.data.data <= dataFi  
              )  
        })
```

*Aquesta entrega comptarà per a l'avaluació de la competència transversal comunicació eficaç oral i escrita. Per tant, heu de fer un document ben escrit on s'hi especifiqui la vostra solució de l'exemple i s'hi expliquin els motius que us han portat a aquesta solució i els dubtes que pugueu tenir sobre la mateixa. L'extensió d'aquesta justificació no pot ser inferior a mitja pàgina.*

Un centre excursionista necessita un sistema per gestionar informació de les travesses que ofereix. De cada travessa se'n coneixen les seves etapes, les persones que l'organitzen i les que hi estan inscrites. Una inscripció pot ser total o parcial. En aquest segon cas, el sistema haurà d'enregistrar també les etapes concretes que s'han efectuat i, si la travessa és de dificultat alta, la persona que l'acompanyava en aquella etapa. A continuació disposeu de l'especificació feta per aquest sistema.

Esquema conceptual d'especificació:



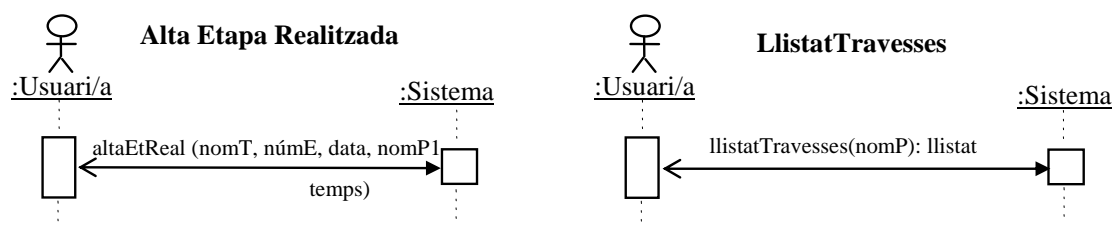
#### Informació derivada

1. /nEtapesRealitzades és el nombre d'etapes realitzades per una Persona

#### Restriccions textuais

1. Claus externes: (Travessa, nom), (Persona, nom), (Data, data).
2. Una travessa no pot tenir dues etapes amb el mateix número.
3. Els números d'una etapa d'una travessa són correlatius, començant per 1.
4. La travessa d'una etapa realitzada és la mateixa que la de la seva inscripció parcial.
5. Una etapa d'una travessa com a màxim pot tenir 25 etapes realitzades.
6. Una EtapaRealitzada és EtapaPerillosa si la dificultat de la travessa a la qual es realitza és 'alta'.
7. L'acompanyant i l'acompanyat d'una etapa perillosa han de fer la mateixa travessa.

Diagrama de seqüència d'esdeveniments del sistema:



Contracte de l'operació *alta d'Etapa Realitzada*:

**Operació:** *altaEtapaRealitzada* (nomT: String, númE: Integer, data: Date, nomP: String, temps: Integer)

**Pre:**

- Existeix l'etapa *númE* a la travessa *nomT*.
- Existeix una inscripció parcial de la persona *nomP* a la travessa *nomT*.
- La travessa *nomT* no té dificultat alta

**Post:**

- Es dona d'alta una instància d'*EtapaRealitzada* amb les dades indicades.

Contracte de l'operació *llistatTravesses*:

**Operació:** *llistatTravesses* (nomP: String): Set(TupleType(nomT:String, tempsTotal:Integer))

**Pre:**

- La persona *nomP* existeix.

**Body:**

- Per cada travessa a la que ha participat la persona *nomP*, es mostra el nom de la travessa i, si ja es coneix, el temps que la persona ha fet a la travessa. El temps es calcula de la manera següent:
  - si la inscripció de *nomP* a la travessa és parcial, i existeix alguna etapa realitzada per *nomP*, es mostra la suma dels temps de les etapes realitzades per *nomP* (si no existeix cap etapa realitzada no s'ha de mostrar el temps).
  - si la inscripció de *nomP* a la travessa és total, es mostra el seu temps total (si no existeix el seu temps total no s'ha de mostrar el temps).

Tenint en compte que:

- l'atribut *nEtapesRealitzades* ha de ser materialitzat.

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 els contracte de les operacions obtinguts com a conseqüència de la traducció de l'esquema d'especificació al de disseny.
- Diagrama de seqüència de les operacions *altaEtReal* i *llistatTravesses* 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.
- Fer una llista de totes les operacions que hagueu utilitzat en el vostre disseny. Indiqueu la signatura completa d'aquestes operacions i digueu quines són abstractes. Indiqueu al diagrama de classes de l'apartat a) la navegabilitat resultant del vostre disseny.





# Contracte de llistat Travesses

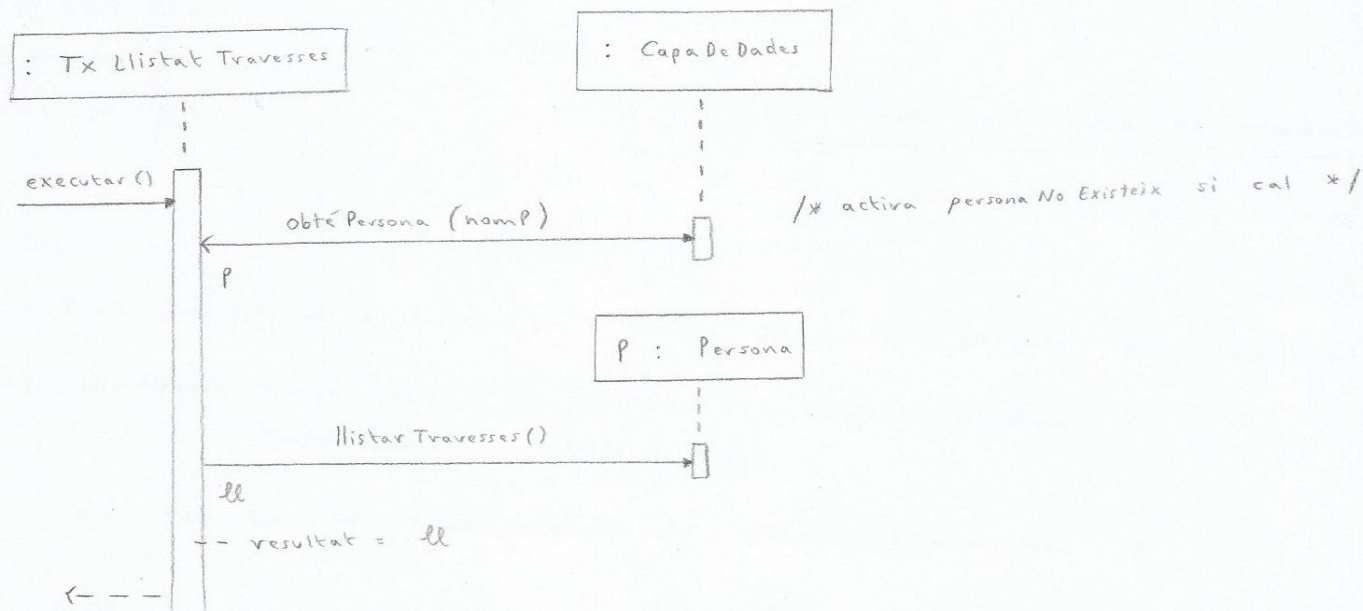
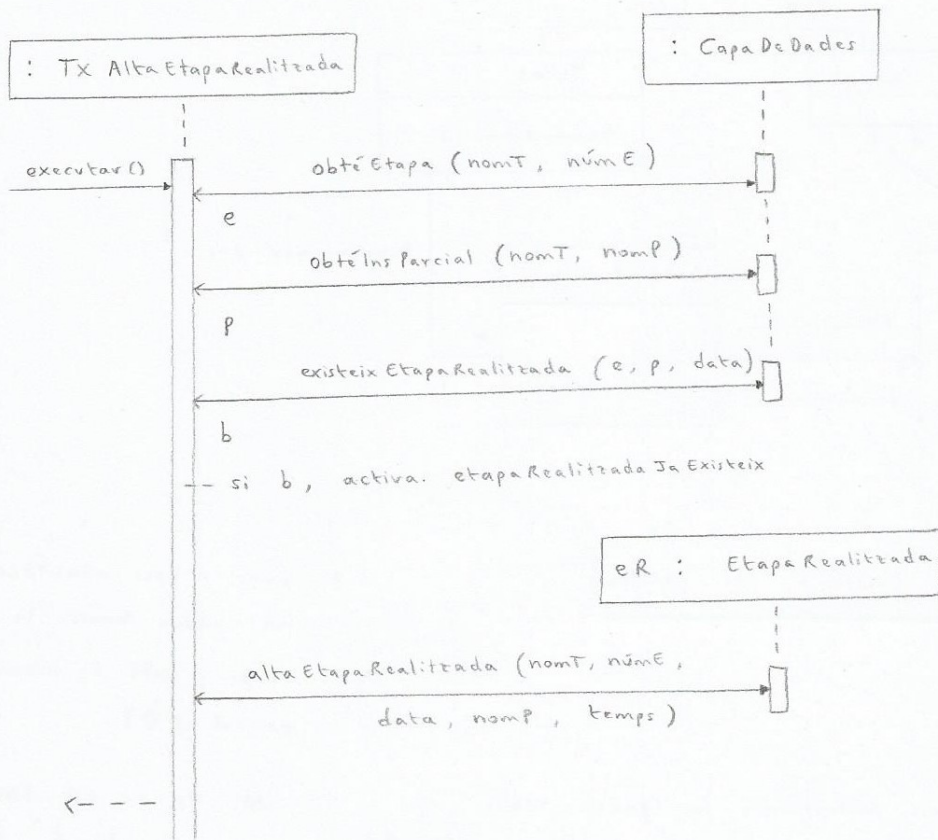
Operació : llistat Travesses (...)

Pre : -

Excepcions : personaNo Existeix : la persona identificada per nomP no existeix

Body : ...

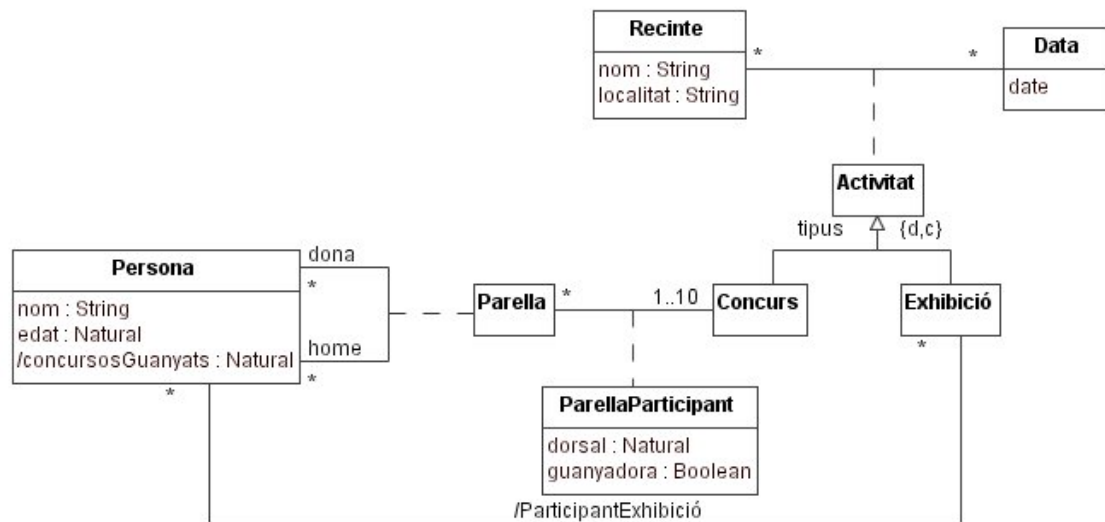
b)



## IES – 3er problema de disseny – 25/26 de maig del 2015 – QP1415

Una escola de ball organitza dos tipus d'activitat, concursos i exhibicions de ball. Les activitats s'organitzen en una data i un recinte determinat, del que en sabem el nom que l'identifica i la localitat on es troba. El sistema també emmagatzema el nom i l'edat de les persones que, per parelles, participen a un concurs, i se'ls hi assigna un número de dorsal al concurs. De cada concurs també s'enregistren les parelles guanyadores. A les exhibicions hi participen totes les persones que hagin estat guanyadores en algun concurs de l'any anterior a l'any de la celebració de l'exhibició. A continuació disposeu de l'especificació feta per aquest sistema.

### Esquema conceptual d'especificació:



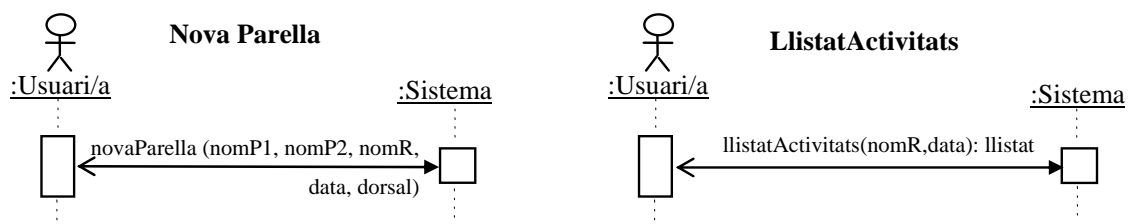
### **Restriccions textuais**

1. Claus externes: (Persona, nom), (Recinte, nom), (Data, data)
2. Els membres d'una parella han de ser diferents
3. Una persona no pot participar més d'una vegada al mateix concurs (com a membre de parelles diferents)
4. En un concurs no hi ha dorsals repetits
5. En un mateix dia no hi pot haver dues exhibicions

### **Informació derivada**

1. /concursosGuanyats d'una Persona és el nombre de concursos en que la persona ha format part d'una parella guanyadora
2. Les persones que participen a una exhibició són tots els que han estat guanyadors d'algun concurs de l'any anterior a l'any de l'exhibició

### Diagrama de seqüència d'esdeveniments del sistema:





Contracte de l'operació novaParella:

**Operació:** novaParella (nomP1: String, nomP2: String, nomR: String, data: Date, dorsal: Integer)

**Pre:**

- Existeix la parella identificada per *nomP1* i *nomP2*.
- Existeix el concurs identificat per *nomR* i *data*.

**Post:**

- Es dona d'alta una instància de ParellaParticipant definida per la parella (*nomP1*, *nomP2*) i pel concurs (*nomR*, *data*), amb el dorsal *dorsal* i l'atribut guanyadora a fals.

Contracte de l'operació llistatActivitats:

**Operació:** llistatActivitats (nomR: String, data:Date): Set(String)

**Pre:**

- L'activitat identificada per *nomR* i *data* existeix.

**Body:**

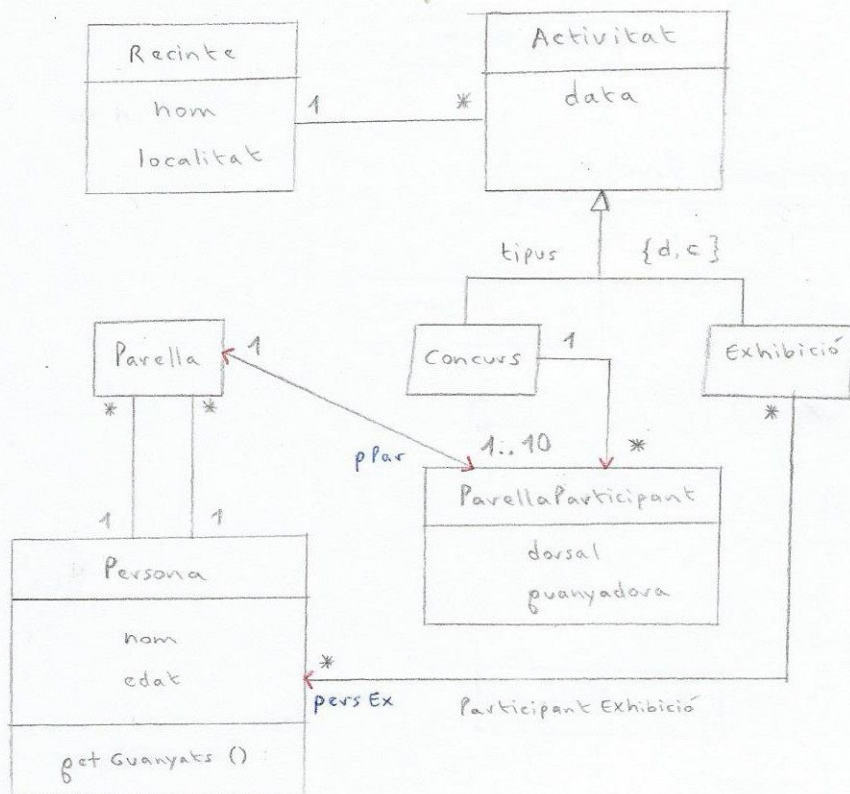
- Si l'activitat és una exhibició, es mostren els noms de totes persones que hi ha participat.
- Si l'activitat és un concurs, es mostren els noms de les dones de les parelles participants al concurs que no varen ser parelles guanyadores.

Tenint en compte que:

- l'associació *participantExhibició* ha de ser materialitzada.
- l'atribut *concursosGuanyats* ha de ser calculat

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 els contracte de les operacions obtinguts com a conseqüència de la traducció de l'esquema d'especificació al de disseny.
- Diagrama de seqüència de les operacions *novaParella* i *llistatActivitats* 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.
- Fer una llista de totes les operacions que hagueu utilitzat en el vostre disseny. Indiqueu la signatura completa d'aquestes operacions i digueu quines són abstractes. Indiqueu al diagrama de classes de l'apartat a) la navegabilitat resultant del vostre disseny.



## RT afegides

- Dues activitats no poden tenir la mateixa data i el mateix recinte
- No hi pot haver dues parelles amb els mateixos membres

## Contracte de nova Parella

Operació : novaParella (...)

Pre : -

- Exc :
1. parellaNo Existeix : no existeix la parella identificada per *nomP1* i *nomP2*
  2. concursNo Existeix : no existeix el concurs identificat per *nomR* i *data*
  3. parellaParticipant Ja Existeix : ja existeix una parella participant amb les dades introduïdes

Post : igual

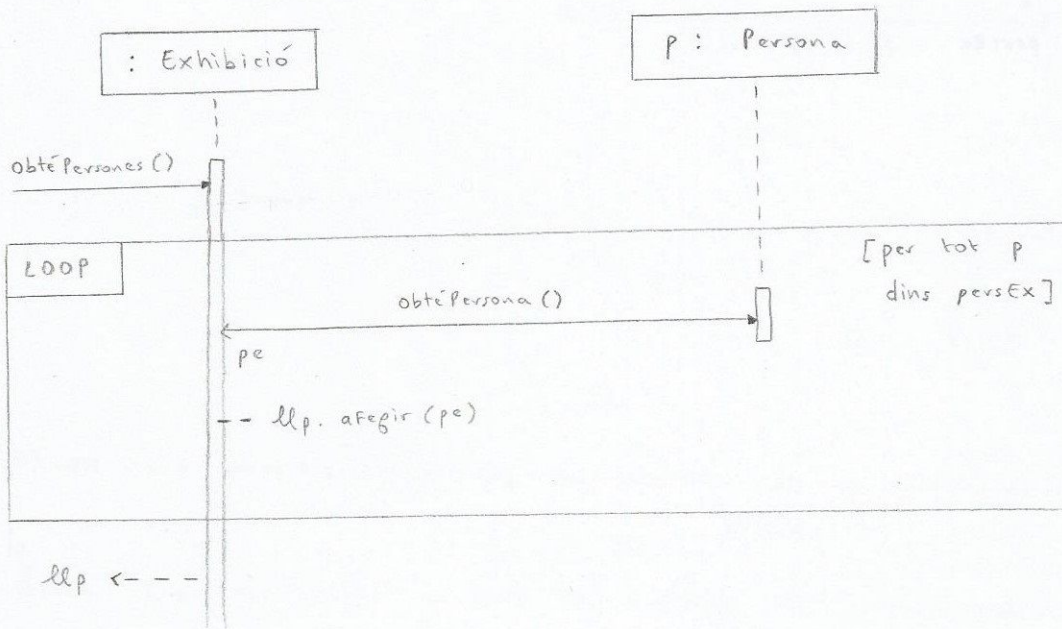
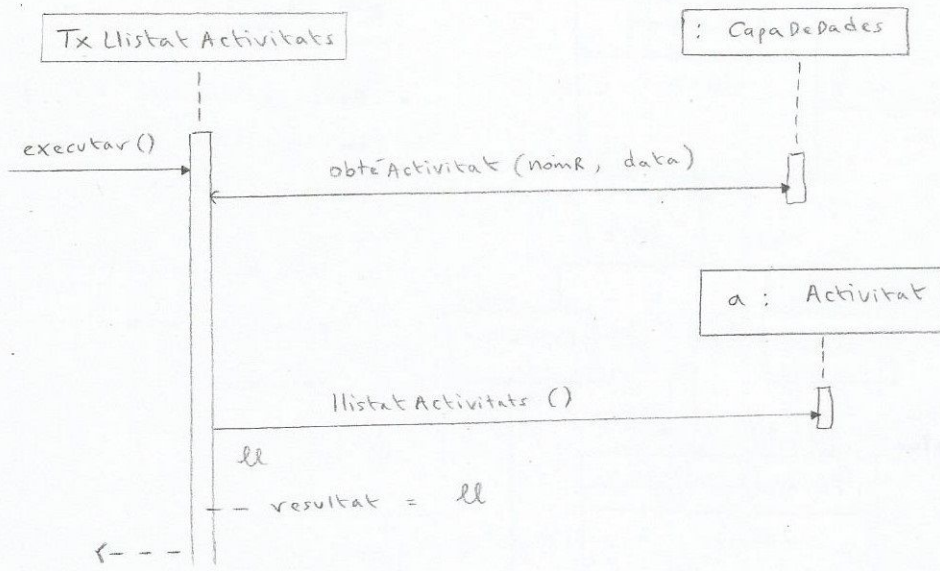
## Contracte de llistat Activitats

Operació : llistatActivitats (...) : set (string)

Pre : -

Exc : activitatNo Existeix : l'activitat identificada per *nomR* i *data* no existeix

Body : igual





alta Parella Participant (p, c, dorsal, nom P1, nom P2)

: Concurs

pP: Parella Participant

LOOP

[ $\forall pP$  de Parella Participant]

obtenir Dorsal()

d

existeix Part (nom P1)

existeix Part (nom P2)

comprovar 10 Part (p)

crear (dorsal, p)

mpp: Parella Participant

self.dorsal = dorsal  
self.guanyadora = fals

parella = p

p: Parella

associar PP (mpp)

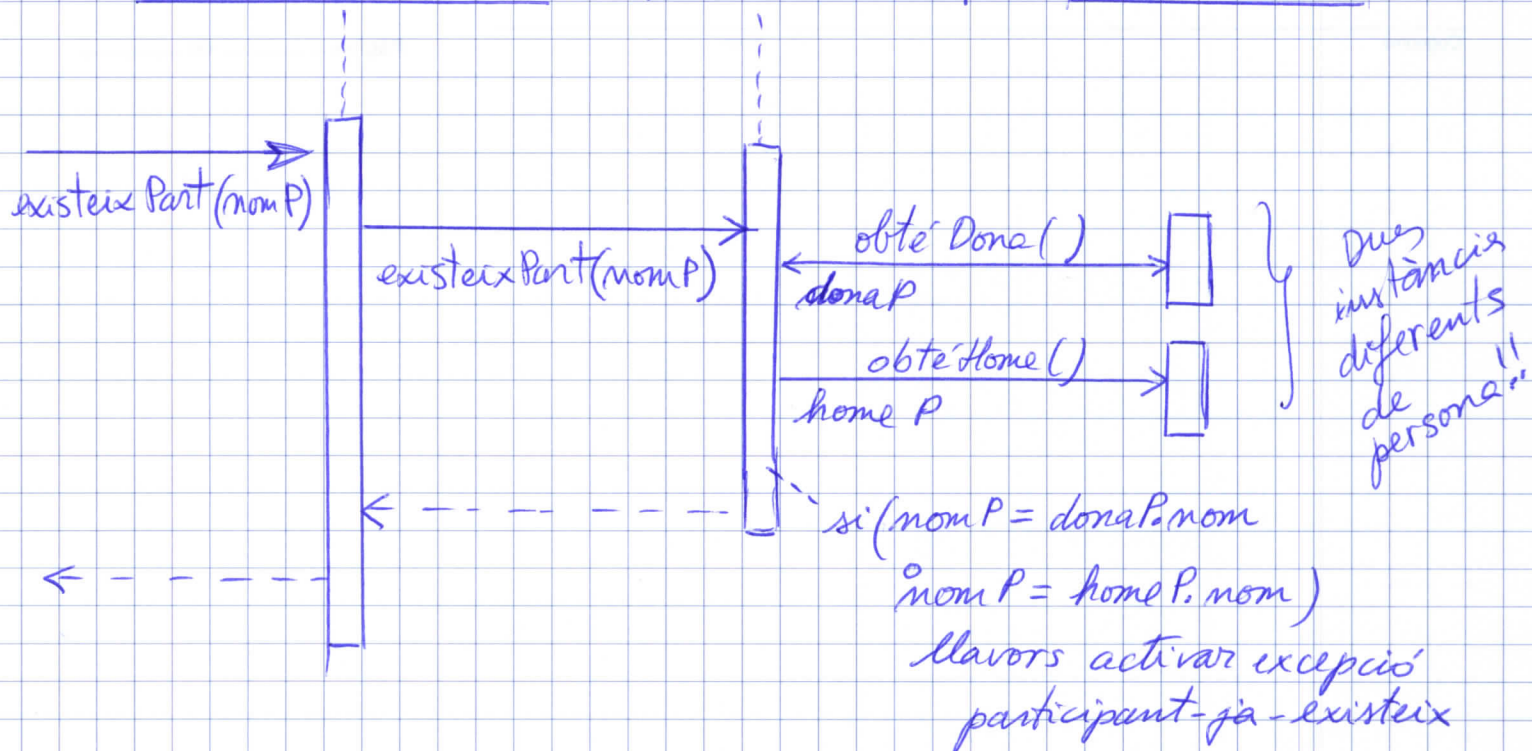
parella Participant =  
• afegir (mpp)

parella Participant. afegir (mpp)

pP: Parella Participant

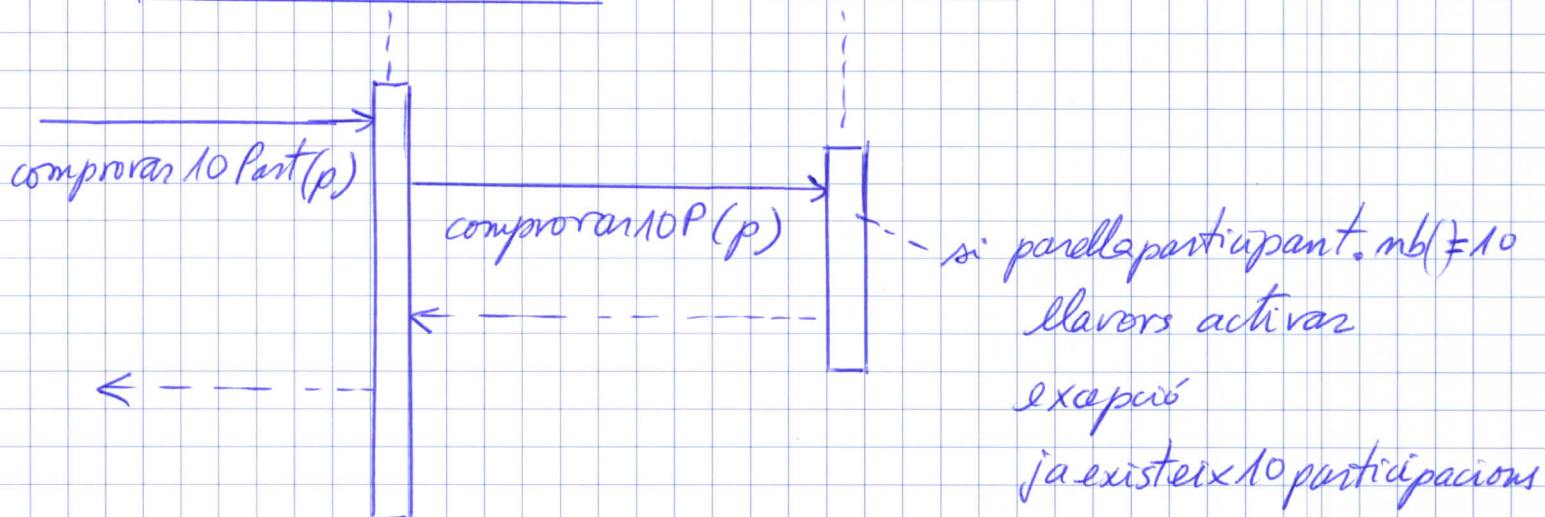
par: Parella

per: Persona



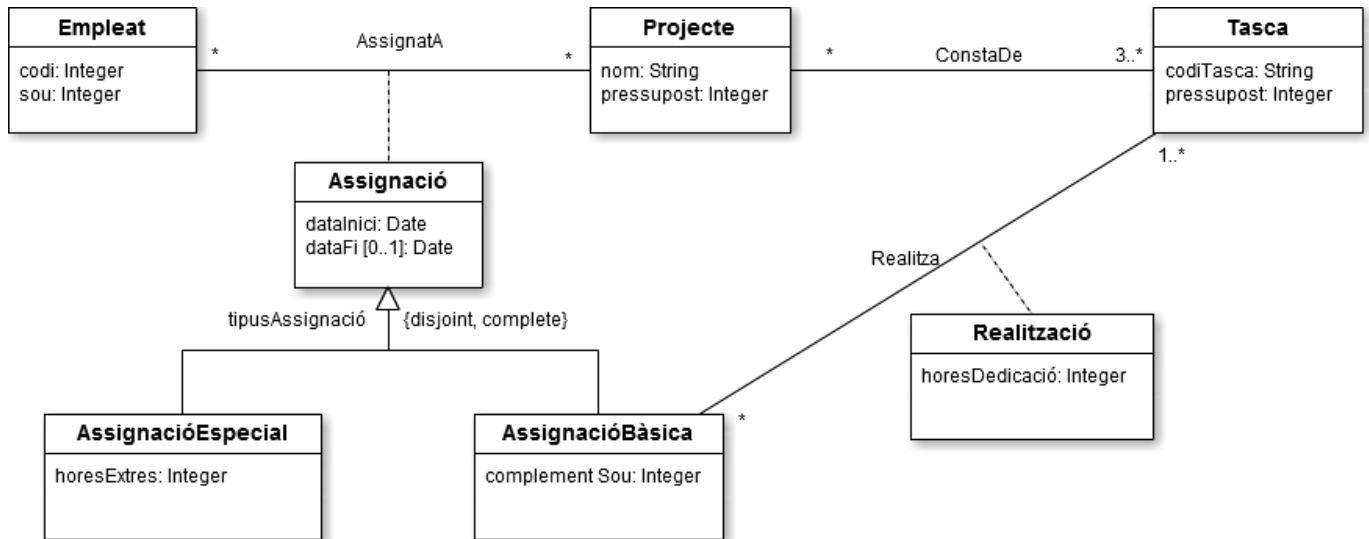
pP: Parella Participant

par: Parella





Considereu el sistema software d'una empresa que gestiona les assignacions d'empleats als seus projectes. Una assignació pot ser bàsica o especial. En el primer cas el sistema enregistra també les tasques que els empleats realitzen en el projecte, amb les hores de dedicació. En el segon, es guarda únicament el nombre d'hores extres que l'empleat fa en aquell projecte. La part rellevant de l'esquema conceptual en UML es mostra a la figura següent:



### Restriccions d'integritat:

1. Claus externes: (Empleat, codi) (Projecte, nom) (Tasca, codiTasca)
2. La dataFi d'una assignació (si té valor) no pot ser anterior a la seva dataInici
3. El pressupost d'un projecte ha de ser superior a la suma de pressupostos de les tasques de que consta
4. En una Realització, la seva tasca és una de les que consta el projecte de la seva AssignacióBàsica

Suposarem que el sistema no ha de donar d'alta instàncies de les classes Empleat, Projecte ni Tasca ja que hi ha un altre sistema encarregat de fer-ho. No obstant, ha de ser capaç d'afegir una tasca a un projecte, de crear una nova assignació bàsica i de consultar els projectes ben complementats.

**Afegir Tasca:** Quan un cap de projecte vol afegir una tasca a un projecte, ell mateix indica al sistema la informació necessària per tal de crear una nova instància de l'associació *ConstaDe*. Cal tenir en compte que aquesta funcionalitat només es pot fer si el pressupost de la tasca a afegir no fa que el pressupost del projecte sigui inferior a la suma dels pressupostos de les seves tasques. Feu que la interacció necessària per dur a terme aquesta funcionalitat requereixi un sol esdeveniment.

**Crear Assignació Bàsica:** Quan un cap de projecte vol donar d'alta una nova instància d'assignació bàsica, ell mateix indica al sistema la informació necessària per a fer-ho. O sigui, la informació de l'assignació bàsica que s'ha de crear i la de les seves múltiples realitzacions. Cal tenir en compte que aquesta funcionalitat només es pot portar a terme si l'empleat amb el que es crea l'assignació no té cap assignació especial. Feu que la interacció necessària per dur a terme aquesta funcionalitat requereixi més d'un esdeveniment.

**Consulta Projectes Ben Complementats:** Quan el director general vol consultar els projectes ben complementats, és ell mateix l'encarregat de demanar-ho al sistema. El sistema retorna, per cada projecte que té alguna assignació bàsica amb un complement de sou superior a 1000€ i un mínim de tres realitzacions, el nom del projecte i els codis de totes les tasques de les que consta.

Es demana que feu mitjançant la notació UML:

- a) Diagrama de seqüència i contractes en OCL de les operacions corresponents a les funcionalitats anteriors.

## Afegir Tasca (2,5 punts )

**context:** Sistema::AfegirTasca(nomP:String, codiT:String)

**pre:** Projecte.allInstances()->exists(p|p.nom=nomP) and  
Tasca.allInstances()->exists(t|t.codiTasca=codiT)

**post:** Projecte.allInstances()->select(p|p.nom=nomP).tasca.codiTasca  
->includes(codiT)

## Crear Assignació Bàsica (4,5 punts: 2,5 CrearAssignBàsica + 2 AfegirRealització )

**context:** Sistema::CrearAssignBàsica(codiE:Integer, nomP:String, dIni:Date, compS:Integer):AssignacióBàsica

**pre:** Projecte.allInstances()->exists(p|p.nom=nomP) and  
Empleat.allInstances()->exists(e|e.codi=codiE and  
e.assignació->select(a|a.oclIsTypeOf(AssignacióEspecial))->size()==0)

**post:** AssignacióBàsica.allInstances()->exists(a|a.oclIsNew() and  
a.empleat.codi=codiE and a.projcte.nom=nomP and  
a.dataInici=dIni and a.complementSou=compS and  
result=a)

**context:** Sistema::AfegirRealització(ab:AssignacióBàsica, codiT:String, hores:Integer)

**pre:** Tasca.allInstances()->exists(t|t.codiTasca=codiT)

**post:** Realització.allInstances()->exists(r|r.OclIsNew() and  
r.assignacióBàsica=ab and r.tasca.codiTasca=codi and  
r.horesDedicació=hores)

## Consulta Projectes Ben Complementats (3 punts )

**context:** Sistema::llProjBenComp():Set(Tupletype(nomP:String, cods:Set(String)) )

**pre:**

**body:** let projs:set(Projecte)=  
Projecte.allInstances()->select(p|p.assignació->exists(a|  
a.oclIsTypeOf(AssignacióBàsica) and  
a.oclAsType(AssignacióBàsica).complementSou>1000 and  
a.oclAsType(AssignacióBàsica).realització->size()>=3))  
in projs->collect(p| Tuple{nomP=p.nom, cods=p.tasca.codiTasca})