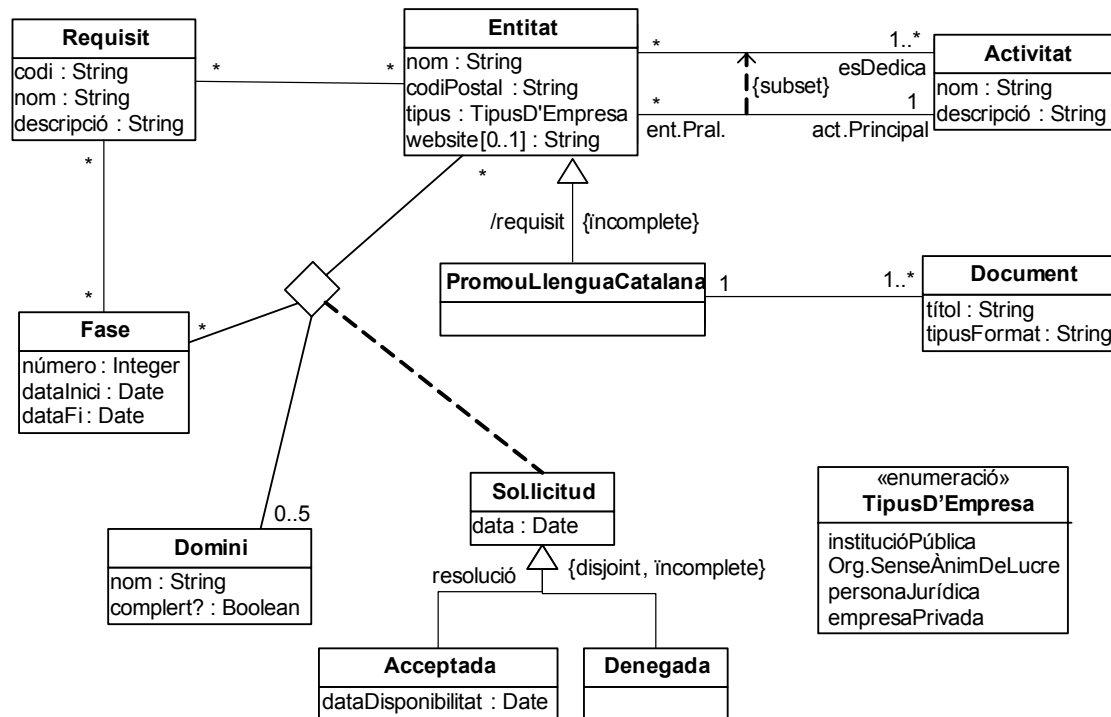


## 1er Problema – Model Conceptual de les Dades (3,5 punts)



## Restriccions d'Integritat Textuals:

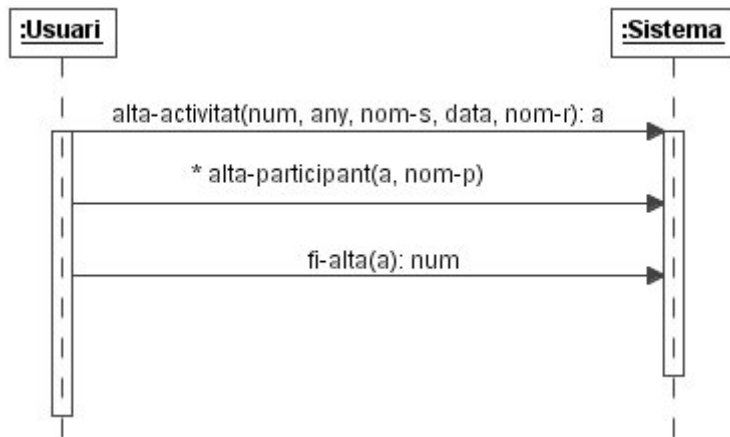
- Claus externes: (Entitat: nom), (Activitat, nom), (Requisit, codi), (Fase, número), (Document, títol), (Domini, nom)
- Els números, identificadors de fase, han de ser números correlatius
- Tota entitat que fa una sollicitud en una fase ha de complir els requisits que són necessaris en aquella fase.
- La data de sollicitud ha d'estar inclosa dins del període de la fase en que es fa la sollicitud
- La data de disponibilitat ha de ser posterior a la data de sollicitud
- El nom del domini ha de ser un nom d'entre 3 y 62 caràcteres de l'alfabet català
- Un domini pot tenir com a màxim una sollicitud acceptada
- Una entitat no pot fer una petició d'un domini en una fase si el domini ja està acceptat o bé si l'entitat l'ha sol·licitat en una fase anterior i aquesta encara no ha estat resolta.

---

## 2on Problema – Model del Comportament (3,5 punts)

---

### Cas d'ús Alta Activitat



**Operació:** `alta-activitat(num-a: Integer, any: Integer, nom-s: String, data: Date, nom-r: String): Activitat`

**Semàntica:** Enregistra una nova activitat amb les dades indicades

**Precondició:**

- `data->notEmpty()` implies `nom-r->notEmpty()` and `nom-r->notEmpty` implies `data->notEmpty()`
- `nom-r->notEmpty()` implies `Restaurant.allInstances()->exists(r|r.nom = nom-r)`
- `Soci.allInstances()->exists(s|s.nom = nom-s)`

**Postcondició:** `a.oclIsNew()` and `a.oclIsTypeOf(Activitat)` and `a.numero=num-a` and `a.any=any` and `a.organitzador.nom=nom-s` and `if data->notEmpty()` then `a.oclIsTypeOf(Sopar)` and `a.data=data` and `a.restaurant.nom=nom-r` endif

**Sortida:** `result = a`

O bé afegir un paràmetre que indiqui si l'activitat és un sopar, i comprovar a la precondició que si és sopar llavors la data i el nom-r no són buits.

**Operació:** `alta-participant(a: Activitat, nom-p: String)`

**Semàntica:** Associa el soci nom-p com a participant a l'activitat a

**Precondició:**

- `Soci.allInstances()->exists(s|s.nom=nom-p)`
- `not a.participant->exists(p|p.nom=nom-p)`

**Postcondició:** `a.participant->exists(p|p.nom=nom-p)`

**Operació:** `fi-alta(a: Activitat): Integer`

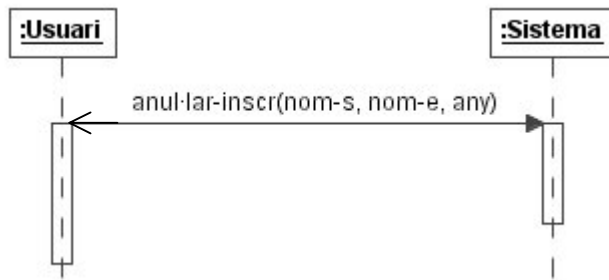
**Semàntica:** Finalitza l'alta de participants a l'activitat a i en retorna el nombre total

**Precondició:** -

**Postcondició:** -

**Sortida:** `result = a.participant->size()`

## Cas d'ús Anul·lar Inscripció



**Operació:** anul·lar-inscr(nom-s: String, nom-e: String, any: Integer)

**Semàntica:** Dóna de baixa la inscripció del soci nom-s a l'esport nom-e a l'any any, si no ha estat pagada. A més, si l'esport es queda sense inscripcions també es dóna de baixa l'esport.

**Precondició:** `Inscripcio.allInstances()->exists(i|i.soci.nom = nom-s and i.esport.nom = nom-e and i.any.any = any and i.pagada = false)`

**Postcondició:** `not Inscripcio.allInstances()->exists(i|i.soci.nom = nom-s and i.esport.nom = nom-e and i.any.any = any)`

## Cas d'ús Consultar Esports sense Organitzadors



**Operació:** `llistat-esports(any: Int): Set(TupleType(nom-e: String, socis: Set(String)))`

**Semàntica:** Retorna un llistat de tots els esports per als quals els socis que hi tenen inscripcions de l'any indicat no han organitzat cap sopar dels realitzats en el mateix any

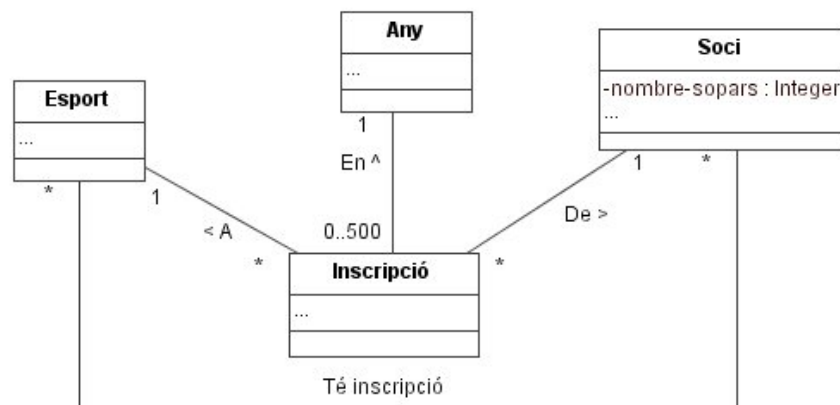
**Pre:** `Any.allInstances()->exists(a| a.any = any)`

**Post:** -

**Sortida:**

```
let esports: Set(Esport) = Esport.allInstances()->select(e | e.inscripció-> select(i | i.any.any = any).soci.act-org.oclAsType(Sopar)->select(s | s.any=any)->isEmpty())
in
result = esports-> collect(e | Tuple{nom-e = e.nom, socis = e.inscripcio-> select(i | i.any.any = any).soci.nom})
```

## NORMALITZACIÓ



(La resta del diagrama queda igual)

RI Textuals (afegides):

- No hi pot haver dues inscripcions amb els mateixos Esport, Any i Soci.
- Un soci es pot inscriure com a màxim a 3 esports en un any.
- En un any es poden inscriure com a màxim 100 socis a cada esport.

S'elimina la restricció 5, ja que ara és gràfica.

Contractes:

- *alta-activitat*:
  - o Afegim a la precondició:
    - No existeix ja una activitat amb *num-a* (control clau activitat)
    - El soci *nom-s* no ha organitzat ja una activitat en l'any *any* (control restricció 2)
    - Si l'activitat és un sopar, no n'hi ha ja un altre amb la mateixa *data* (control restricció 3)
    - Si l'activitat és un sopar, la *data* pertany a l'any *any* (control restricció 4)
  - o Afegim a la postcondició:
    - Si l'activitat és un sopar, s'incrementa en 1 el valor de l'atribut *nombre-sopars* del soci amb *nom-s* (materialització de *nombre-sopars*)
- *fi-alta*:
  - o Afegim a la precondició:
    - El soci organitzador de l'activitat *a* és també un dels participants (control subset)
- *anul·lar-inscr*:
  - o Afegim a la postcondició:
    - Si el soci *nom-s* no té cap altra inscripció a l'esport *nom-e*, s'esborra la instància de '*Té inscripció*' que relaciona aquest soci amb aquest esport (materialització de '*Té inscripció*')

La resta queda igual.