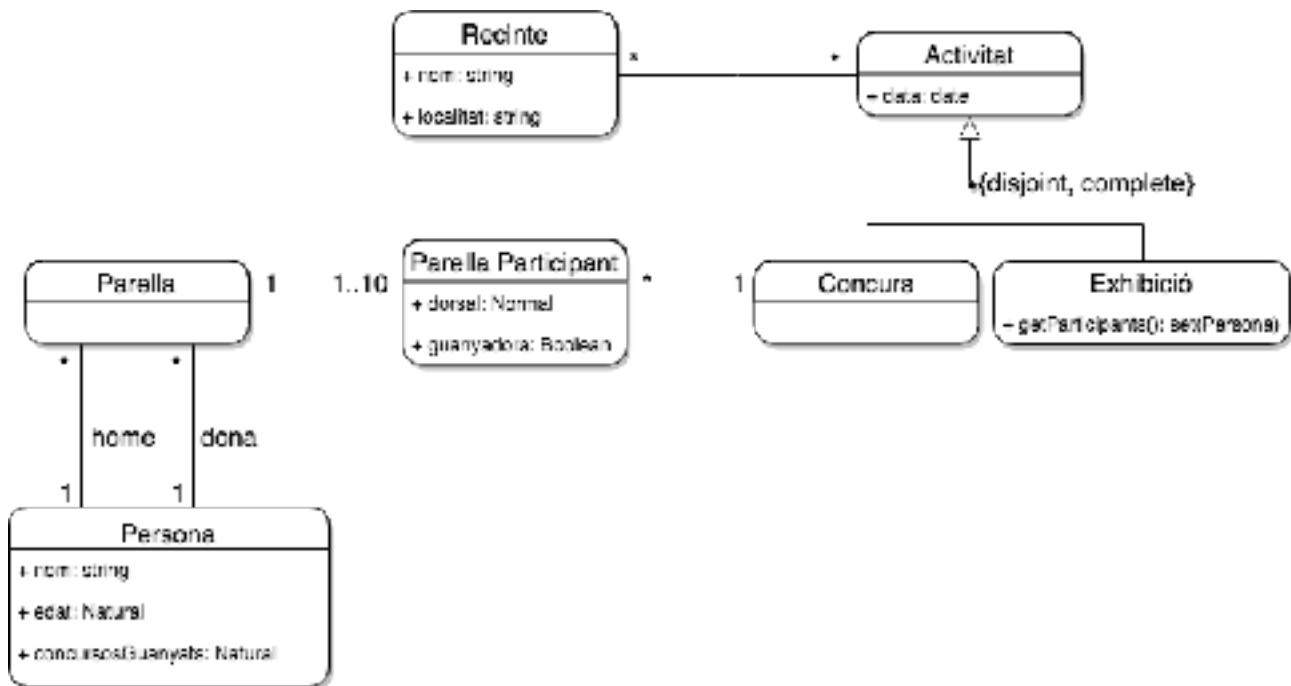


## A. Diagrama de classes de disseny:



### Restriccions d'integritat:

1. Claus externes: (Persona, nom), (Recinte, nom)
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
6. No poden existir dues activitats en el mateix recinte i dia.
7. No poden existir dues parelles amb el mateix home i dona.

### **Contracte de l'operació novaParellaParticipant:**

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

#### **Exc:**

- Ja\_dorsal: Ja existeix una parella amb el dorsal
- No\_parella: No existeix una parella identificada per nomP1 i nomP2.
- No\_concurs: No existeix concurs amb 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)

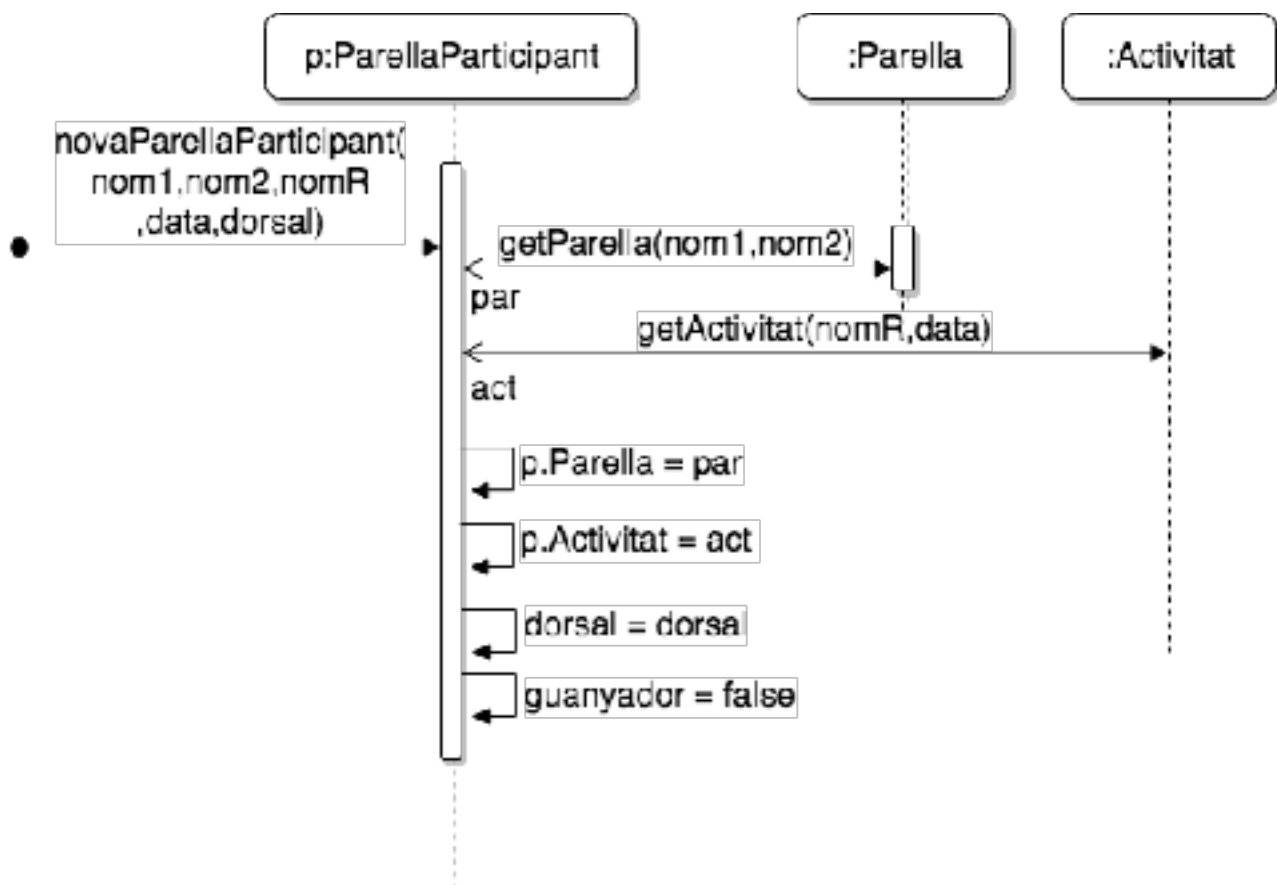
**Exc:**

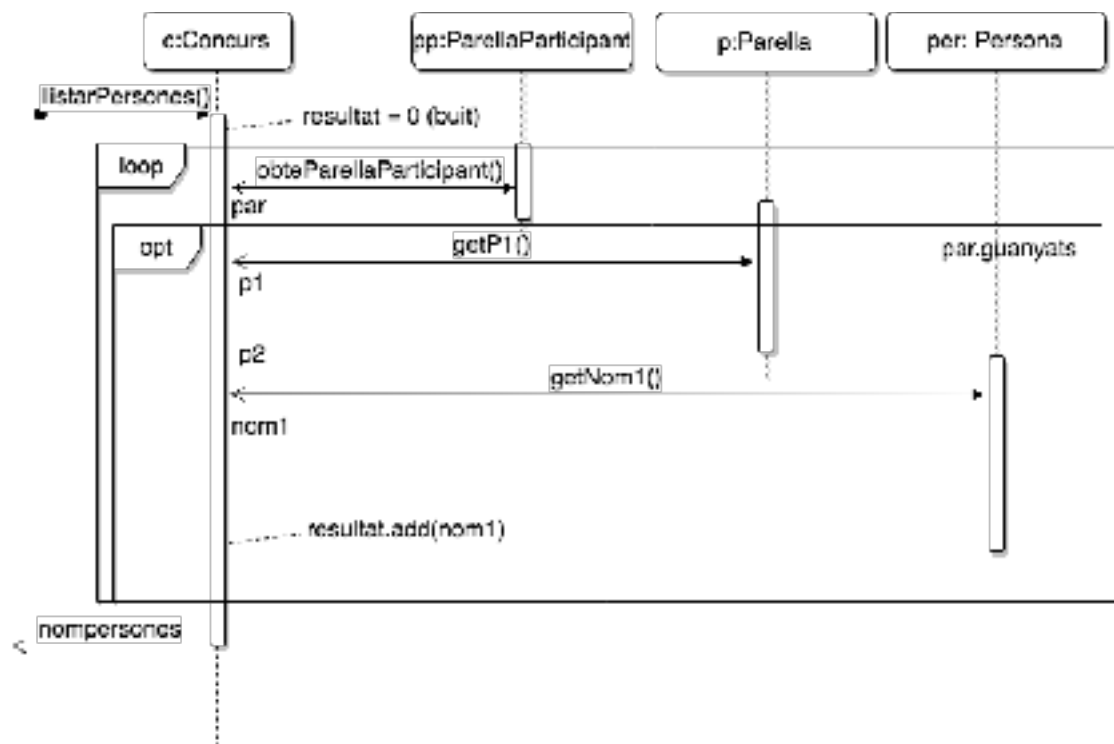
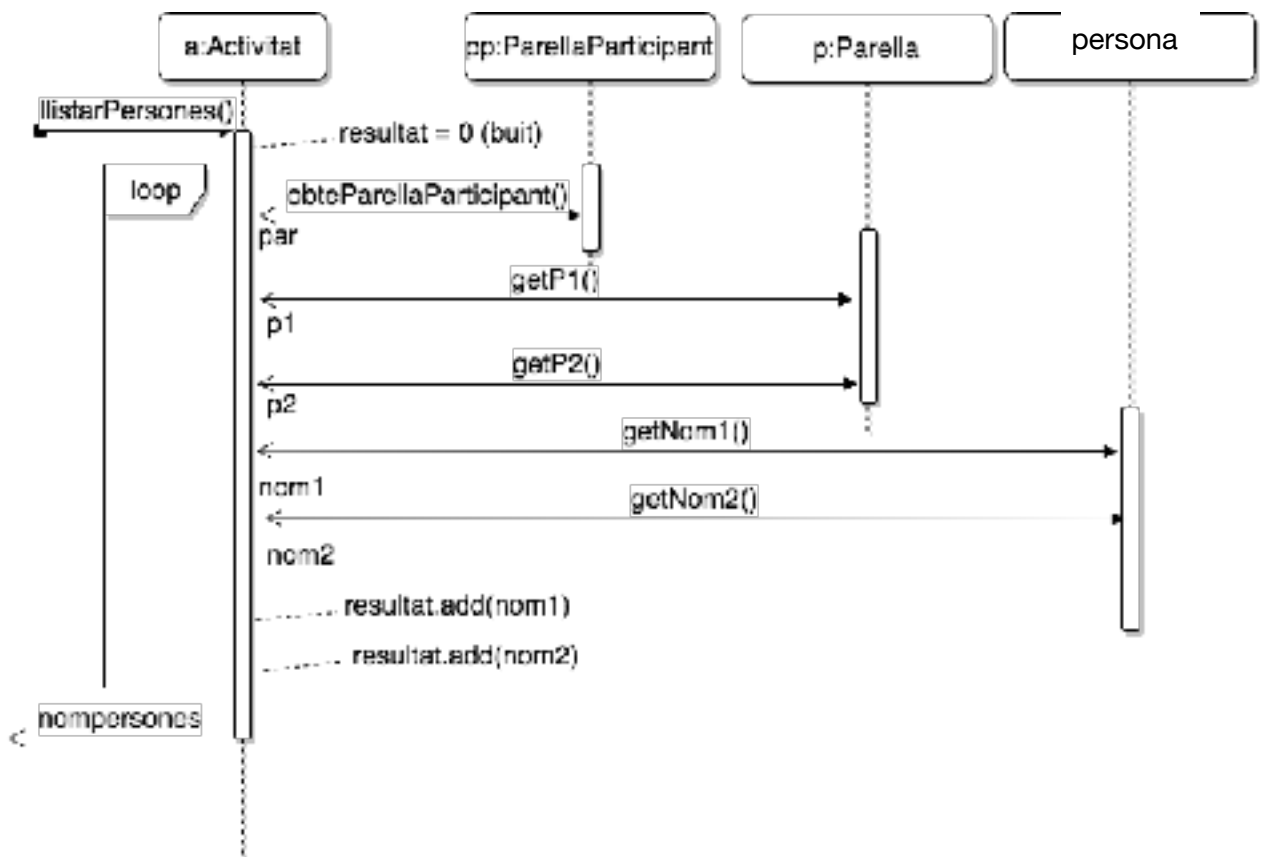
- No\_activitat: No existeix l'activitat identificada per nomR.

**Body:**

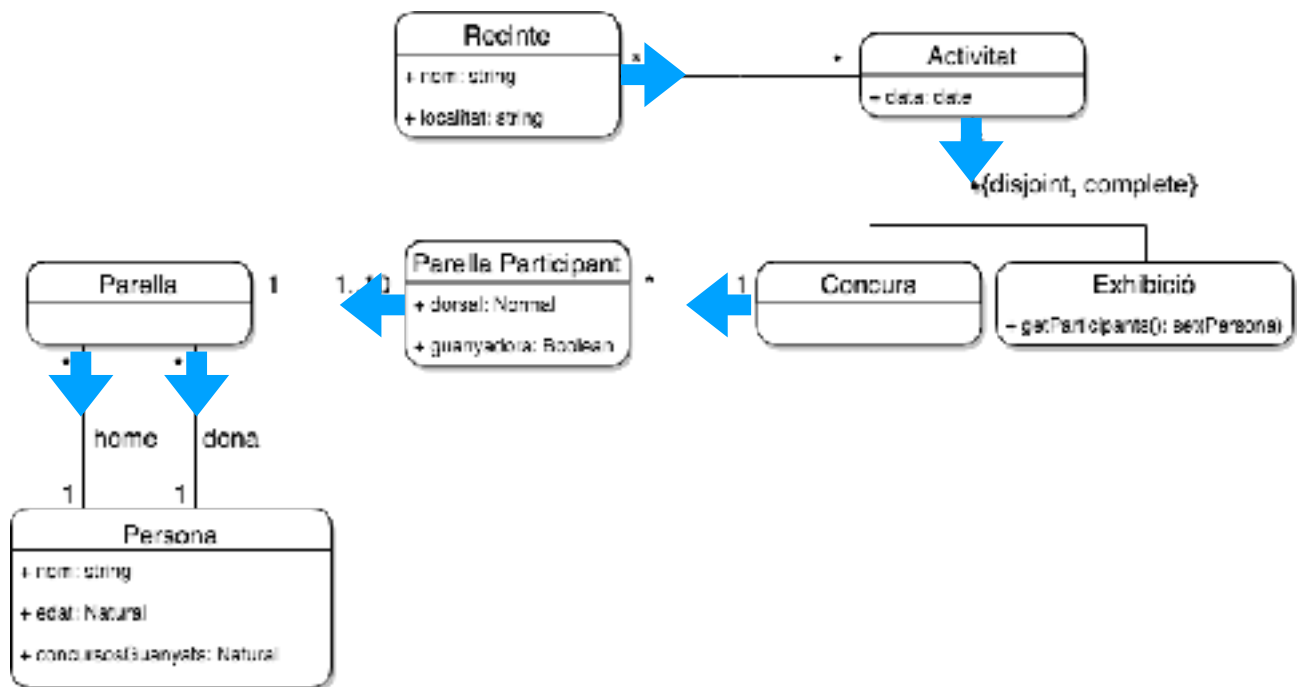
- Si l'activitat és una exhibició, es mostren els noms de totes persones que hi ha participat.
- Si l'activitat és un concurs, es mostren els noms de les dones de les parelles participants al concurs que no varen ser parelles guanyadores.

B. Diagrama de seqüència:





## C. Navegabilitat:



## Competència Transversal:

En l'exemple, per poder transformar-lo al diagrama de disseny cal eliminar les associatives. En aquest cas comencem eliminant la classe data i afegim l'atribut a Activitat, definint-la com a classe. Afegint la restricció d'integritat corresponent aconseguim que no existeixi més d'una activitat per dia i recinte. Les multiplicitats no varien.

També ens cal crear la classe ParellaParticipant entre Parella i Concurs, eliminant així la associativa. En aquest cas no ens cal afegir-li cap atribut a aquesta. Les multiplicitats, com en la majoria de ternàries s'intercanvien, és a dir, la de concurs es situa a la banda de parellaparticipant amb parella i la de parella a la contrària. La resta esdevenen uns. D'aquesta manera aconseguim que cada parellaparticipant pugui participar només en un concurs i que cada parella només pugui participar 10 vegades com a màxim.

Finalment eliminem la associativa de la classe persona i la convertim en classe. Assignem les multiplicitats de les relacions home i dona. Aquests permeten que donada una persona, aquesta pugui participar a més d'una parella, però una parella ha de tenir, per força, un participant home i un de dona.

Centrant-nos en les restriccions, no ens cal afegir que una parella no pugui apuntar-se més d'una vegada en un concurs, ja que la restricció 3 ja s'en ocupa.

Pel que fa a les operacions, gairebé no varien. Només fa falta afegir restriccions trivials per comprovar les dades que s'ens passen a l'hora de crear una nova instància de parella participant: Activitat, Persones i Dorsal.

I finalment, fixant-nos en els diagrames de seqüència, veiem que és necessari obtenir la parella de la classe omònima passant-li el nom dels dos participants i també l'activitat a partir del nom del recinte i la data, que són els que la identifiquen. Un cop obtinguts els dos elements, els assignem a la nova instància de parella, afegint, a més a més, el dorsal i posant a fals el bool de guanyador.

Pel que fa al segon diagrama, ens cal crear dos operacions, la genèrica i el polimorfisme. Ambdues utilitzant un loop per recórrer totes les parelles participants. La genèrica obté la parella participant i d'aquesta n'obté les dues persones. Aquestes, es passen cap a la classe persona per obtenir el nom dels dos. A partir d'aquí, només ens cal fer un `.add()` per afegir-los al set resultat.

Per contrapartida, el polimorfisme sobreescriu la funció. La nova només obté la persona dona de la parella si el bool de guanyadors és cert. A continuació, n'obté el nom i l'afegeix al set resultat. Al finalitzar el loop es retorna aquest set.