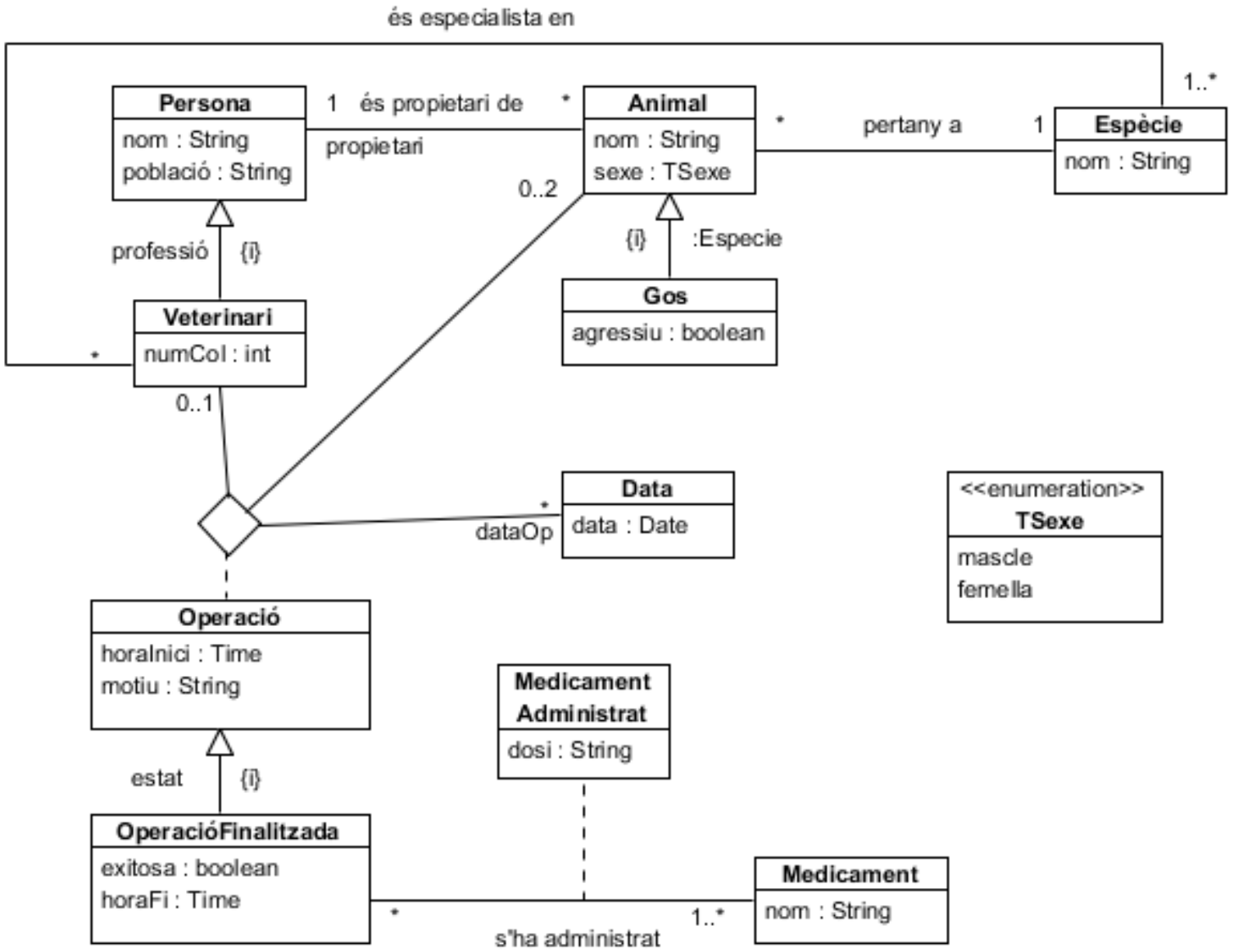
Clínica Veterinària

Solució

Esquema Conceptual de Dades



Restriccions textuals

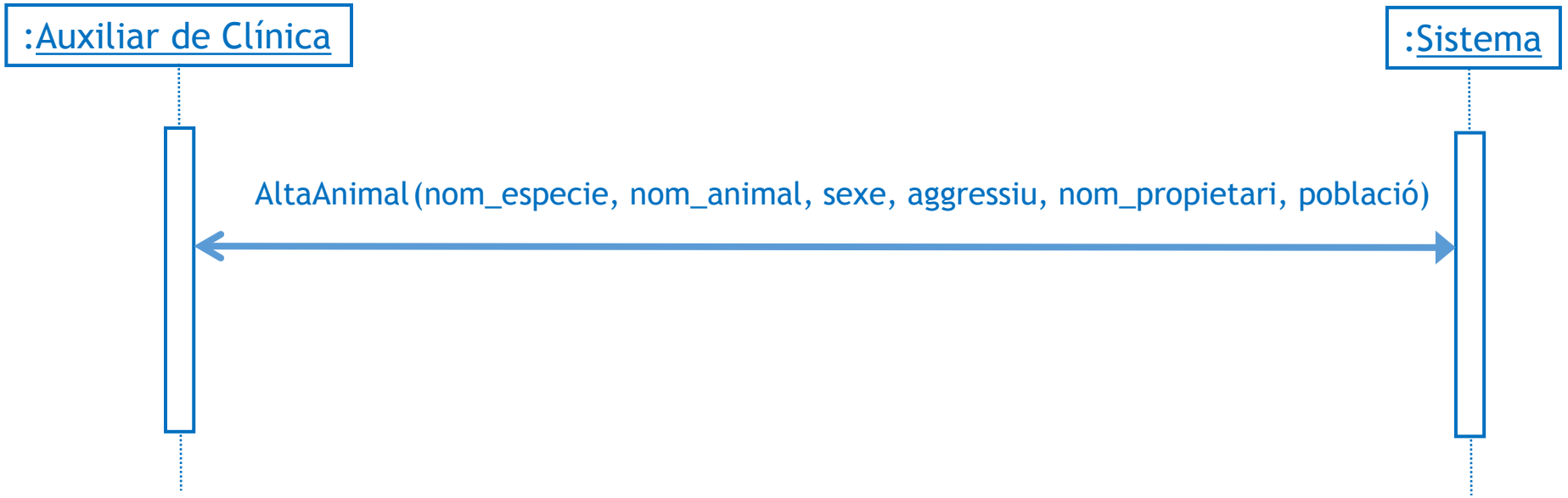
- Claus externes: (Animal, nom), (Persona, nom), (Veterinari, numCol), (Data, data), (Espècie, nom), (Medicament, nom)

- Per cada operació finalitzada, horaFi > hora Inici

- Un Veterinari no pot fer dues operacions que se solapin temporalment.

El sistema a desenvolupar no ha de donar d'alta **Veterinari, Espècie, Medicament** ni **Data** ja que hi ha un altre sistema encarregat de fer-ho

Alta Animal – Diagrama de Seqüència



- **Alta Animal**: Quan l'auxiliar de clínica vol donar d'alta un nou animal a la clínica, ell mateix introdueix la informació necessària per fer-ho. És a dir, tota la informació requerida per a qualsevol animal. Si el propietari de l'animal no està registrat en el sistema, s'haurà de crear en aquell moment (però no com a veterinari).

Contracte operació – AltaAnimal -Ll. natural

- **Operació**: AltaAnimal(nom_especie: String, nom_animal:String, sexe: TSexe, agressiu: boolean, nom_propietari: String, població: String)

- **Precondicions**:
  - Existeix la Especie amb nom = nom_especie
- **Postcondicions**
  - Si el propietari amb nom=nom_propietari no existeix
    - Crea una nova instància de Persona amb nom=nom_propietari I població = població
  - S'ha creat una instància A d'Animal amb nom = nom_animal associat amb la especie que te nom=nom_especia
  - Si l'animal era un Gos
    - A és de tipus gos i agresiu = agressiu.

Contracte de Operació – AltaAnimal - OCL

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona
FIB
Departament d'Enginyeria de Serveis i Sistemes d'Informació
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Context:

```
Sistema::AltaAnimal(nom_especie: String, nom_animal:String, sexe: TSexe,
agressiu: boolean, nom_propietari: String, població: String)
```

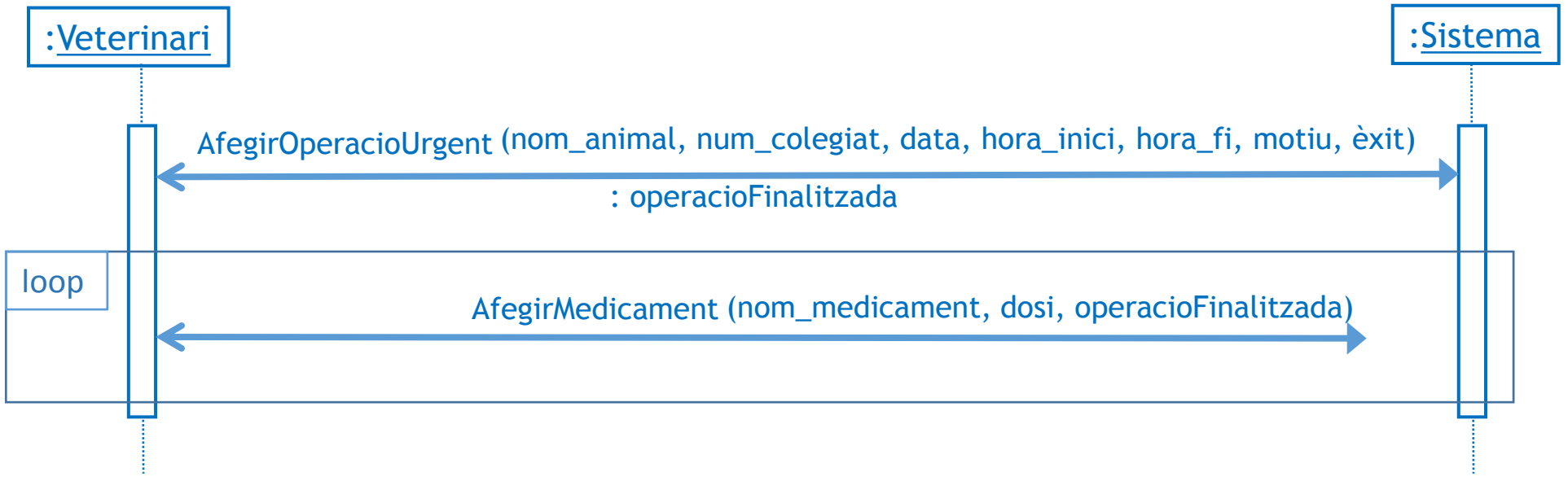
Pre:

```
Especie.allInstanes()->exists(e | e.nom = nom_especie)
```

Post:

```
If not (Persona.allInstances()@pre->Exists(p | p.nom = nom_persona) then
    Persona.allInstances()->Exists(p |      p.nom=nom_persona
                                            and p.població=nom_població
                                            and p.oclIsNew())
Endif
And
Animal.allIstances()->Exists(a |  a.oclIsNew()
                                  and a.nom=nom_animal
                                  and a.propietari.nom = nom_propietari
                                  and a.sexe = sexe
                                  and a.especie.nom = nom_especie
                                  and if (nom_especie='Gos') then
                                        a.oclIsTypeOf(Gos)
                                        and a.oclAsType(Gos).agressiu = agressiu
                                      endif )
```

Alta Operació Urgent - DSS



Alta Operació Urgent: En els cas de les operacions urgents, aquestes són donades d'alta un cop ja han finalitzat. Per fer-ho, el veterinari indica al sistema totes les dades necessàries per crear una operació finalitzada. En aquest cas, podeu assumir que l'animal ja està donat d'alta al sistema. A més, aquesta funcionalitat només pot realitzar-se si el veterinari que realitza l'operació és especialista en l'espècie de l'animal que opera. Feu que la interacció necessària per dur a terme aquesta funcionalitat requereixi més d'un esdeveniment.

AfegirOperacióUrgent – CO – Ll. natural

- **Operació**: AfegirOperacioUrgent(nom_animal: String, num_col·legiat: int, data: Date, hora_inici: Time, hora_fi: Time, motiu: String, èxit: Boolean):OperacióFinalitzada
- **Precondicions**:
  - L'animal amb nom = nom_animal existeix
  - El veterinari amb numCol = num_col·legiat existeix
  - El veterinari és especialista en l'espècie de l'animal
- **Postcondicions**
  - S'ha creat una OperacióFinalitzada OP amb els valors de exitosa, horaFi, hora Inici, motiu, animal, data i veterinari intoduïts
- **Sortida**
  - OP

AfegirOperacióUrgent – CO – OCL

Context:

```
Sistema::AfegirOperacioUrgent(nom_animal: String, num_col·legiat: int, data: Date,
hora_inici: Time, hora_fi: Time, motiu: String, èxit: Boolean): OperacioFinalitzada
```

pre:

```
Animal.allInstances()->exists(a | a.nom = nom_animal)
And Veterinari.allInstances()->exists(v | v.numCol = num_col·legiat)
And Veterinari.allInstances()->exists(v |
    v.numCol = num_col·legiat and v.especie->includes(
            Animal.allInstances()->select(a | a.nom = nom_animal).especie)
```

post:

```
OperacioFinalitzada.allInstances()->exists(of |
                                        of.oclIsNew()
                                        and of.existosa = exit
                                        and of.horaFi = hora_fi
                                        and of.hora Inici = hora_inici
                                        and of.motiu = motiu
                                        and of.dataOp = data
                                        and of.veterinari.numCol = num_col·legiat
                                        and of.animal.nom = nom_animal
                                        and result = of)
```

AfegirMedicament – CO – Ll. natural

- **Operació**: AfegirMedicament(nom_medicament: String, dosi: String, of: OperacioFinalitzada)
- **Precondicions**:
  - El medicament amb el nom nom_medicament existeix
- **Postcondicions**
  - S'ha creat una nova instància de MedicamentAdministrat relacionada amb la operació i el medicament introducits amb la dosi demanada.

AfegirMedicament– CO – OCL

Context:

```
Sistema::AfegirMedicament(nom_medicament: String, dosi: String, of: OperacioFinalitzada)
```

pre:

```
Medicament.allInstances()->exists(m | m.nom = nom_medicament)
```

post:

```
MedicamentAdministrat.allInstances() -> exists (ma |
                                                ma.oclIsNew()
                                                and ma.dosi = dosi
                                                and ma.medicament.nom = nomM
                                                and ma.operacioFinalitzada = of)
```

Consulta Gossos Perillosos - DSS

- **Consulta Gossos Perillosos no Esterilitzats**: Podria ser el cas que l'auxiliar de clínica volgués consultar quins gossos agressius hi ha donats d'alta que no han estan sotmesos ni tenen programada una operació amb motiu "esterilització", per tal de recomanar als seus propietaris que la facin. Per fer-ho, l'auxiliar indica el nom de la població al sistema. El sistema retorna una llista amb el nom del gos, el nom del propietari i un boolèa indicant si aquest últim també és veterinari, per tots aquells gossos agressius tals que: el propietari del gos viu a la població indicada per paràmetre, el gos no està esterilitzat (ni aquesta està programada), i el gos és mascle (en les femelles l'esterilització no millora l'agressivitat). Per poder dur a terme aquesta funcionalitat és necessari que hi hagi una persona com a mínim que visqui a la població indicada i que sigui propietària d'almenys un animal.

Consulta Gossos Perillosos – CO – Ll. natural

- **Operació**: ConsultaGossosPerillosos(nom_població: String): llistat
- **Precondicions**:
  - A la població indicada hi ha d'haver algun propietari d'un animal
- **Postcondicions**
  - El sistema retorna una llista amb
    - Nom del gos
    - Nom del propietari
    - Booleà si el propietari és veterinari
  - Per al següents casos
    - Gossos Agressius
    - Mascles
    - Propietari viu a la població amb nom = nom_població
    - No te cap operació d'esterilització (ni feta ni programada)

Consula Gossos Perillosos – CO – OCL

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona
FIB
Departament d'Enginyeria de Serveis i Sistemes d'Informació
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Context:

```
Sistema::ConsultaGossoPerillosos(nom_població: String)
    : llistat: Set(
            tupletype (
                    nom_animal: String, nom_propietari: String, es_veterinari: boolean))
```

pre:

```
Persona.allInstances()->exists(p |
                                p.població = nom_població
                                and p.animal->notEmpty())
```

body:

```
Let gossos:Set(Gos) =
    Gos.allInstances()->select( g |
                                g.agressiu = true
                                and g.sexe = Tsexe::mascle
                                and g.propietari.població = nom_població
                                and not g.operació.motiu->includes("Esterilització"))
In
Result = gossos->collect(gos | Tuple {
                                nom_animal = gos.nom, nom_propietari = gos.propietari.nom,
                                es_veterinari = gos.propietari.oclIsTypeOf(Veterinari)
})
```