

```

context: Sistema::AltaPersona(
    n_persona: string
    poblacio: string
) :Persona

pre:
post:
    Persona.allInstances() -> Exists(p |
        p.nom = n_persona and
        p.poblacio = poblacio and
        p.oclIsNew() and
        result = p)
  
```

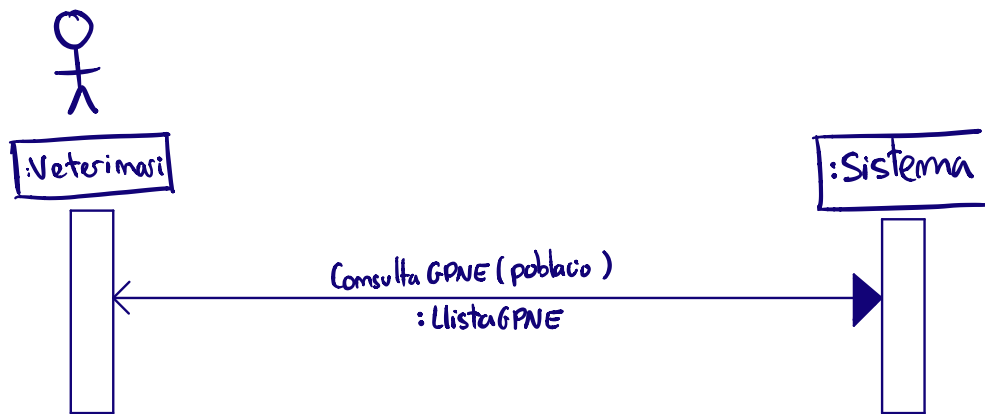
```

context: Sistema::AltaAnimal(
    n_animal: string
    sexe: TSexe
    n_especie: string
    propietari:Persona
) :Animal

pre:
    Persona.allInstances() -> Exists(p | p = propietari) and
    Especie.allInstances() -> Exists(e | e.nom = n_especie and not e.nom = "gos")

post:
    Animal.allInstances() -> Exists(a | a.nom = n_animal and
        a.sexe = sexe and
        a.persona = propietari and
        a.oclIsNew() and
        result = a)
  
```

```
context: Sistema::AltaGos(  
  n_animal:  string  
  sexe:      TSexe  
  n_especie: string  
  propietari: Persona  
  agressiu:  Bool  
) :Gos  
  
pre:  
  Persona.allInstances() -> Exists(p | p = propietari) and  
  Especie.allInstances() -> Exists(e |  
    e.nom = n_especie and  
    e.nom = "gos"  
  )  
  
post:  
  Animal.allInstances() -> Exists( g |  
    g.nom = n_animal and  
    g.sexe = sexe and  
    g.agressiu = agressiu and  
    g.persona = propietari and  
    g.oclIsNew() and  
    result = g  
  )
```



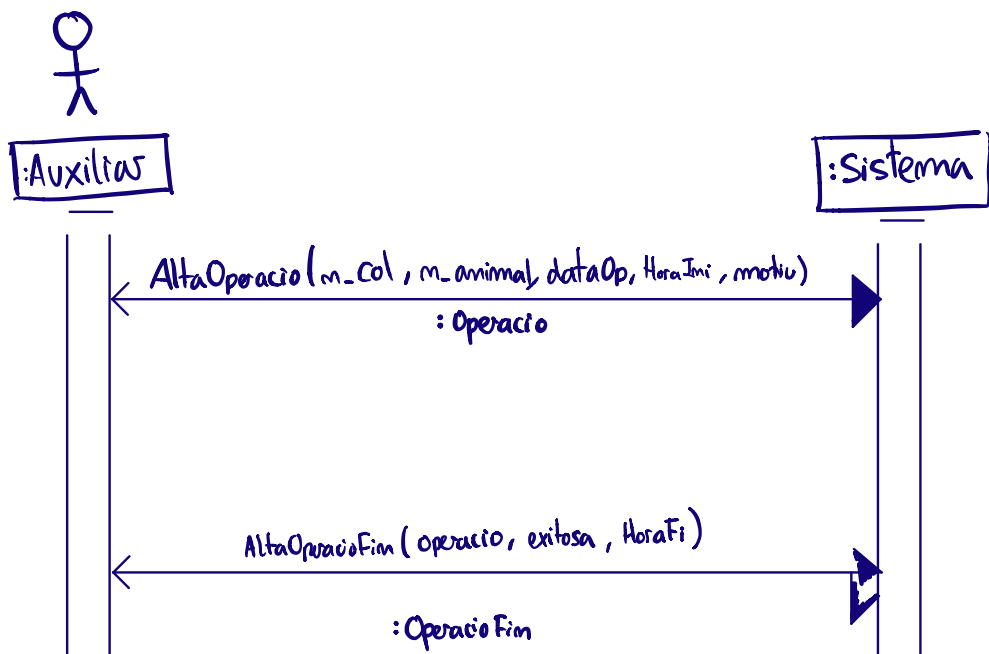
```

context: Sistema::ConsultaGPNE(
  poblacio:  string
) :set(tupletype(
  nom_gos:    string
  nom_prop:   string
  es_vet:     Bool
))

pre:
  Persona.allInstances() -> Exists( p | p.poblacio = poblacio and p.animal.size() > 0 )
body:
  if a.especie = "Gos" then

    result = Animal.allInstances() ->
      select( a | a.oclIsTypeof(Gos) and
        a.oclAsType(Gos).agressiu and
        a.sexe = TSexe::mascle and
        and
        not a.operacio.motiu = "Esterilitzacio"
      ) -> collect( c1 |
        tuple { nom_gos = c1.nom,
          nom_prop = c1.persona.nom,
          es_vet = c1.persona.oclIsTypeof(Veterinari)
        }
      )
  end if
end body

```



```

context: Sistema::AltaOperacio(
  n_col:      int
  n_animal:   string
  dataOp:     Date
  HoraIni:    Time
  motiu:      string
) :Operacio

```

pre:

```

Veterinari.allInstances() - Exists( v |
  v.numCol = n_col and
  v.especie.animal.nom -> includes(n_animal)
)

```

post:

```

Operacio.allInstances() -> Exists( op |
  op.horaInici = HoraIni and
  op.motiu = motiu and
  op.oclIsNew() and
  result = op
)

```

```

context: Sistema::AltaOperacioFin(
  operacio:   Operacio
  HoraFi:     Time
  exitosa:    Bool
) :OperacioFinalitzada

```

pre:

```

Operacio.allInstances() - Exists( o | o = operacio)

```

post:

```

OperacioFinalitzada.allInstances() -> Exists( opf |
  opf.horaFi = HoraFi and
  opf.exitosa = exitosa and
  opf.operacio = operacio and
  opf.oclIsNew() and
  result = opf
)

```