



Alta Exemplar

```
1 context:
2   Sistema::AltaExemplar(nomEx:String, sexeEx:TipusSexe, nomEsp:String, minKg: int, maxKg:int, esFert:Bool)
3
4 pre:
5   Especie.allInstances()->exists(esp|esp.nom = nomEsp) and
6   Zona.allInstances()->exists(z | z.especie.nom->includes(nomEsp) and
7   z.especie->size() >= 2)
8
9
10 post:
11   Exemplar.allInstances->exists(ex |
12     ex.oclIsNew() and
13     ex.nom = nomEx and
14     ex.sexe = sexeEx and
15     ex.especie.nom = nomEsp and
16     if nomEsp = "Lleo" then
17       ex.oclIsTypeOf(ExemplarLleo) and
18       ex.oclAsType(ExemplarLleo).minimKg = minKg and
19       ex.oclAsType(ExemplarLleo).maximKg = maxKg and
20       if sexeEx = TipusSexe::Femella then
21         ex.oclIsTypeOf(ExemplarLleoFemella) and
22         ex.oclAsType(ExemplarLleoFemella).fertil = esFert
23       endif
24     endif
25   )
```

Alta Zona

```
1 context:
2   Sistema::AltaZona(nomZ:String):Zona
3
4 pre:
5
6 post:
7   Zona.allInstances()->exists(z |
8     z.nom = nomZ and
9     z.oclIsNew() and
10    result = z)
11
12
13 context:
14   Sistema::AssignaEspecie(zonaNova:Zona, nomEsp:String):Especie
15
16 pre:
17
18 post:
19   if Especie.allInstances()->exists(esp |
20     esp.zona = zonaNova and
21     esp.nom = nomEsp and
22     result = esp
23   )
24   and if not Especie.allInstances()->exists(esp | esp.nom = nomEsp) then
25     Especie.allInstances()->exists(esp |
26       esp.nom = nomEsp and
27       esp.oclIsNew() and
28       result = esp
29     )
30
31
32 context:
33   Sistema::AltaLoc(espAssig:Especie, nomP:String, estatC:TipusEstat)
34
35 pre:
36   Pais.allInstances()->exists(p | p.nom = nomP)
37
38 post:
39   localitzacio.allInstances() -> exists(l |
40     l.estatConservacio = estatC and
41     l.pais.nom = nomP and
42     l.especie = espAssig and
43     l.oclIsNew()
44   )
```

ConsultaAgressionsRellevants

```
1 context: Sistema::ConsultaAgr(nomAgressor::String)
2   :Set(tupletype (
3     dataAgressio:Date,
4     nomsFerits:Set(String)
5   ))
6
7 pre:
8   Exemplar.allInstances()->exists(ex |
9     ex.nom = nomAgressor and
10    not ex.especie.localitzacio->exists(loc |
11      loc.estatConservacio=TipusEstat::EnPerillGreu)
12  )
13
14 body:
15   let llistaAgr: set(Agressio) = Agressio.allInstances()->select(agr |
16     agr.agressor.nom = nomAgressor and
17     agr.agredit.especie.localitzacio->exists(loc |
18       loc.estatConservacio=TipusEstat::EnPerillGreu)
19   )
20   in result = llistaAgr -> collect {
21     ag | tuple {
22       dataAgressio = ag.data.data
23       nomsFerits = ag.ferit.nom
24     }
25   }
```