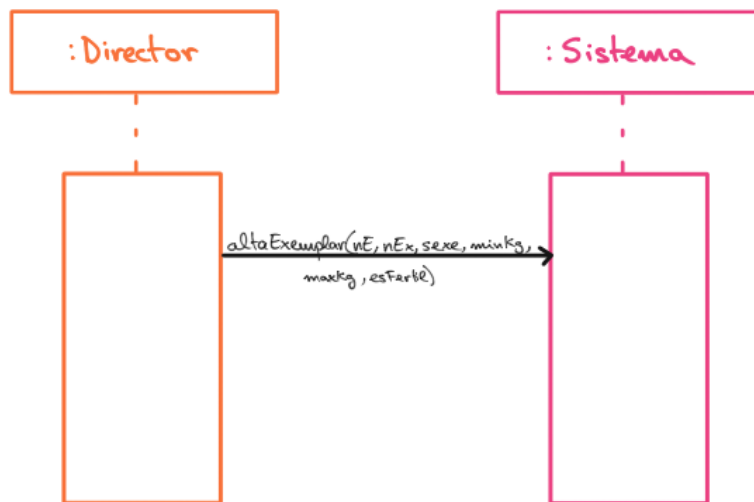




# Exercici 1 - Zoo

## Alta Exemplar

Quan el director del zoo vol donar d'alta a un exemplar d'una espècie, ell mateix introdueix la informació necessària per fer-ho. És a dir, tota la informació requerida per a qualsevol exemplar. Aquesta funcionalitat només pot realitzar-se si l'espècie de l'exemplar ja existeix, i aquesta espècie està assignada a almenys una zona que té, com a mínim, dues espècies assignades.



```
Context: Sistema::altaExemplar ( nE: String, nEx: String, sexe: TipusSexe ) : E

pre:   Especie.allInstances() -> exists ( e | e.nom = nE
      and e.zona -> exists ( z | z.Especie -> size() >= 2 )
      and e.zona -> exists ( z | z.Especie.size() >= 2 )

post:   Exemplar.allInstances() -> exists ( eX | eX.oclIsNew()
      and eX.nom = nEx
      and eX.sexe = sexe
      and eX.Especie.nom = nE
      and result = eX )

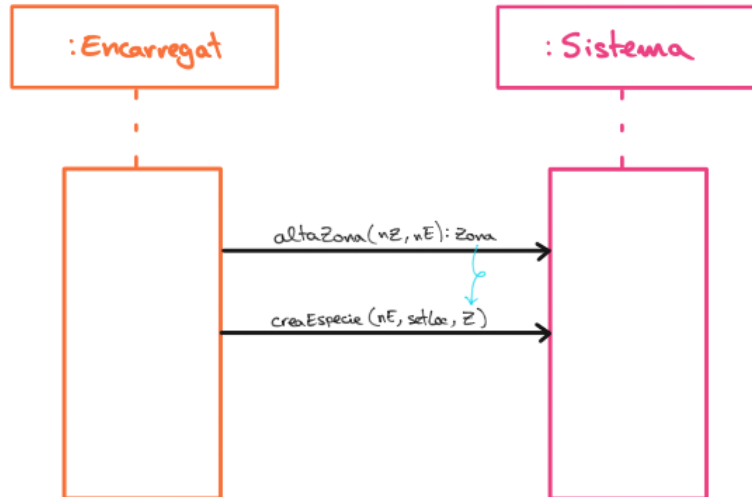
Context: Sistema::altaLleo ( E: Exemplar, minKg: Integer, maxKg: Integer, esFertil: bool )

pre:

post:   E.oclAsType ( ExemplarLleo ).minimKg = minKg
      and E.oclAsType ( ExemplarLleo ).maximKg = maxKg
      and if E.sexe = "Femeni" then
        E.fertil = esFertil
      endif
```

## Alta Zona

Quan l'encarregat d'infraestructures del zoo vol donar d'alta una zona, introdueix la informació de la zona, i la informació de les espècies que hi haurà assignades. S'ha de considerar que si l'espècie no existeix en el sistema, s'ha de crear. Feu que la interacció per dur a terme aquesta funcionalitat requereixi més d'un esdeveniment.



```

Context: Sistema::altaZona ( nZ: Sting, nE: set ( String ) ) : Z

pre:

post:  Zona.allInstances() -> exists ( z | z.ocIsNew()
      and z.nom = nZ
      and z.especie = nE
      and result = z )

Context: Sistema::creaEspecie ( nE: String, setLoc: set ( TupleType ( nP: String, estatC: TipusEstat ) ), Z: Zona )

pre:

post:  if not ( Especie.allInstances()@pre -> exists ( e | e.nom = nE ) ) then
      Especie.allInstances() -> exists ( e |
        e.ocIsNew()
        and e.nom = nE
        and setLoc -> forAll ( loc |
          Localtzacio.allInstances() -> exists ( l |
            l.ocIsNew()
            and l.estatConservacio = loc.estatC
            and l.pais.nom = loc.nP )))
      endif

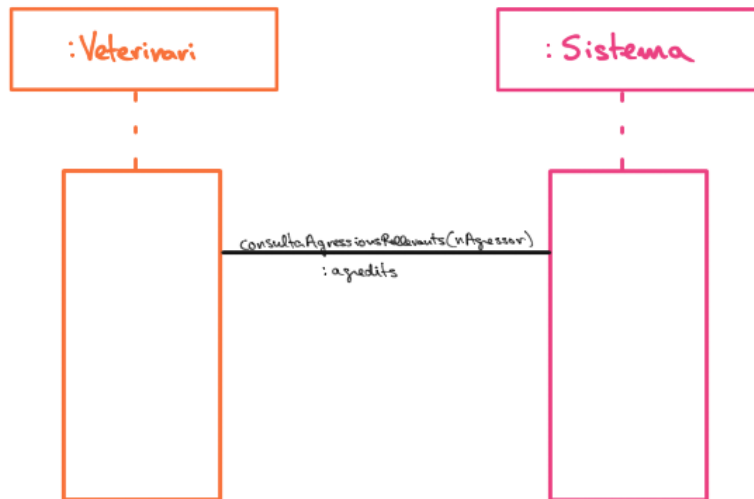
      Zona.allInstances() -> exists ( z | z.nom = Z.nom
        and z.especie -> includes ( nE ) )

```

## Consulta Agressions Rellevants

Quan el veterinari del zoo vol consultar les agressions rellevants d'un exemplar, ell introdueix el nom de l'exemplar agressor. El sistema mostra informació de cada agressió en la què participa aquest exemplar com a agressor, i on l'exemplar agredit és d'una espècie que està "EnPerillGreu" en almenys un país. Per a cadascuna d'aquestes

agressions, es mostra la data de l'agressió, i el nom dels exemplars que en varen resultar ferits. Aquesta funcionalitat només es pot realitzar si l'exemplar agressor és d'una espècie que està "SensePerill" a tots els països on es localitza.



```

Context: Sistema::consultaAgressionsRellevants ( nAgressor: String ) : agredits : set ( TupleType ( dt: data, nA: set ( String ) ) )

pre:   Exemplar.allInstances() -> exists ( eX | eX.nom = nAgressor
      and eX.especie -> forAll ( e | e.especie.localitzacio.estatConservacio = TipusEstat::SensePerill )

body:  let agredits : set ( Agressio ) = Agressio.allInstances() -> select ( a |
      a.agressor.nom = nAgressor
      and a.agredit.especie.localitzacio.allInstances() -> exists ( eLoc |
        eLoc.estatConservacio = TipusEstat::EnPerillGreu ) )
    in result = agredits -> collect ( a |
      Tuple { dt = a.data.data,
              nA = a.ferit.nom })
  
```



Ignasi Fibla Figuerola < ignasi.fibla@estudiantat.upc.edu >  
 Gerard Madrid Miró < gerard.madrid@estudiantat.upc.edu >