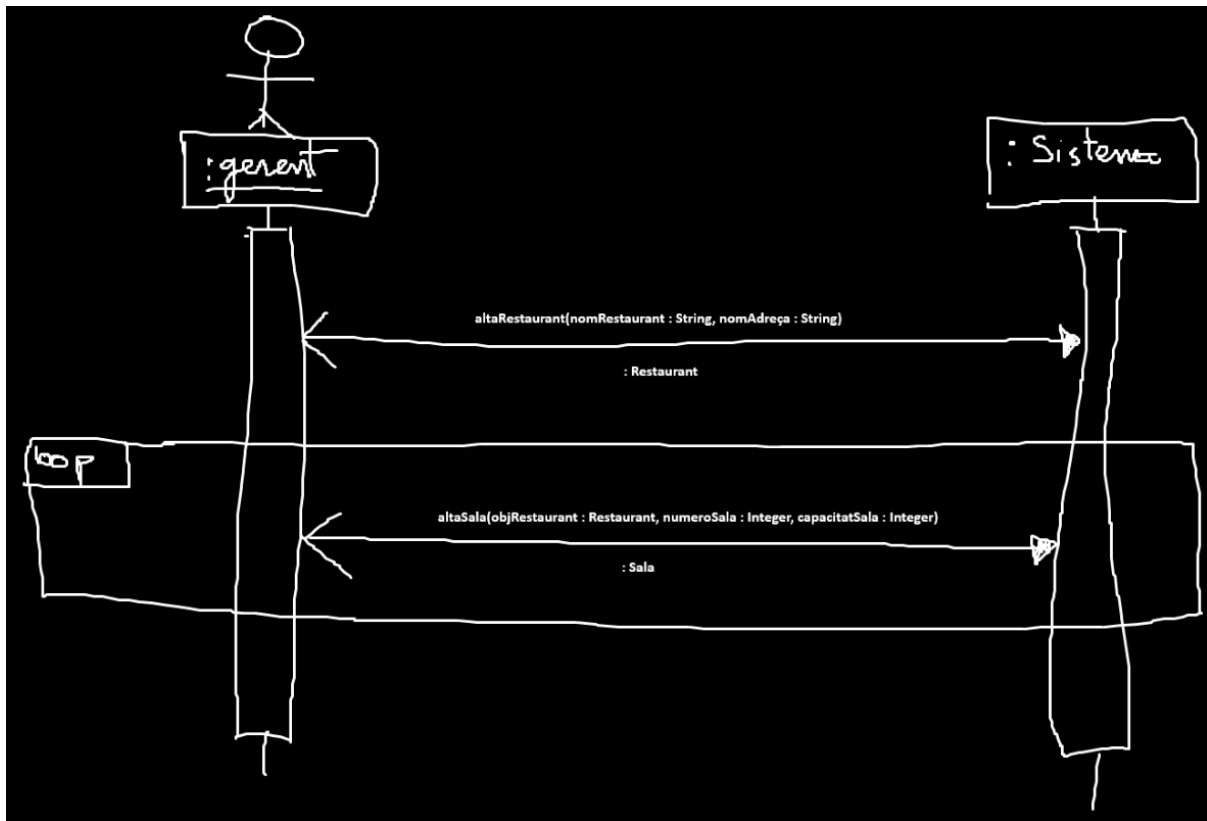


Alta Restaurant

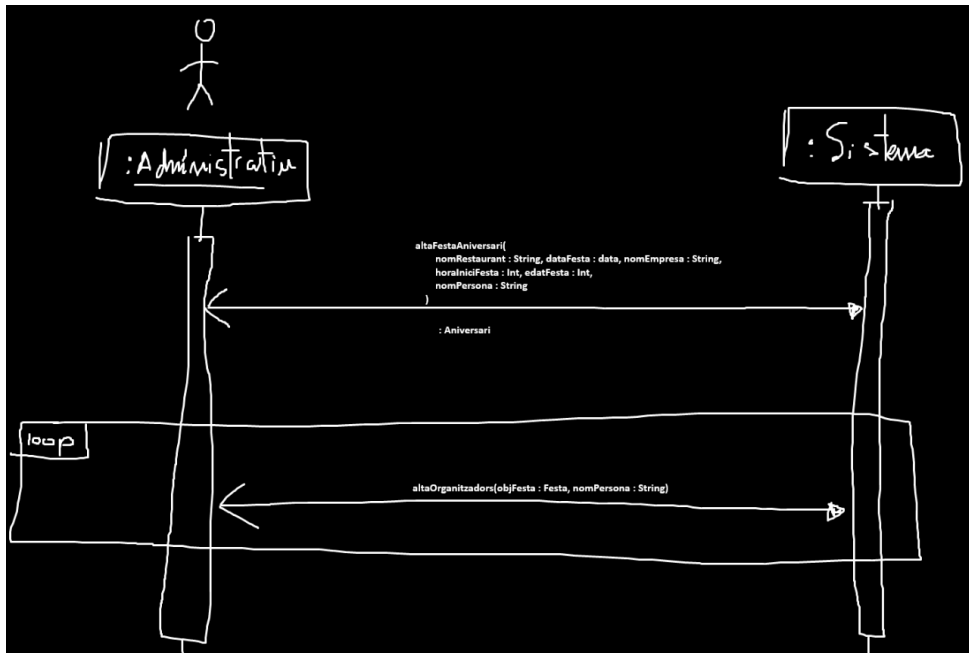


```

////////////////////
// Alta de Restaurant
////////////////////
context:
    Sistema::altaRestaurant(nomRestaurant : String, nomAdreça : String) : Restaurant
pre:
    // Menys de 100 restaurants
    Restaurant.allInstances()->size() < 100
post:
    Restaurant.allInstances()->exists(r |
        r.oclIsNew() AND
        r.nom = nomRestaurant AND r.adreça = nomAdreça AND
        r = result
    )

context:
    Sistema::altaSala(objRestaurant : Restaurant, numeroSala : Integer, capacitatSala : Integer) : Sala
pre:
    // Màxim 5 sales
    objRestaurant.sala.allInstances()->size() < 5
post:
    Sala.allInstances()->exists(s |
        s.oclIsNew() AND
        s.numero = numeroSala AND s.capacitat = capacitatSala AND
        s.restaurant = objRestaurant AND
        s = result
    )
  
```

Alta Festa d'Aniversari



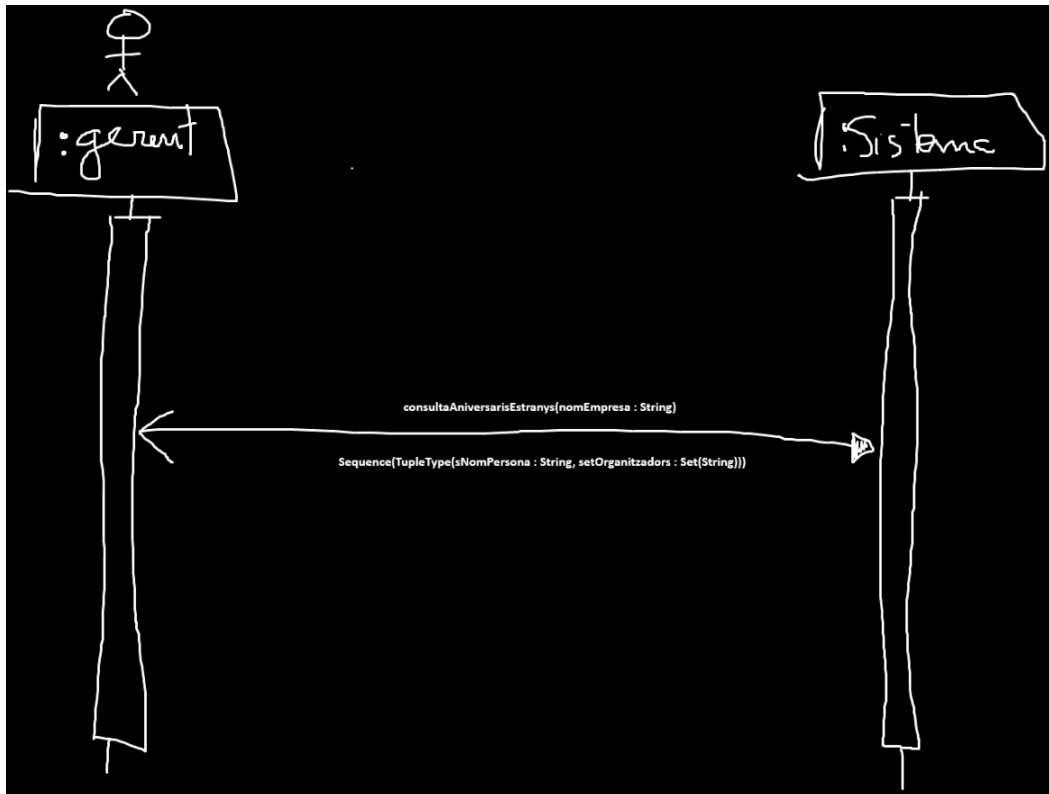
```

////////////////////
// Alta Festa d'Aniversari
////////////////////
context:
    Sistema::altaFestaAniversari(
        nomRestaurant : String, dataFesta : data, nomEmpresa : String,
        horaIniciFesta : Int, edatFesta : Int,
        nomPersona : String
    ) : Aniversari
pre:
    Restaurant.allInstances()->exists(r | r.nom = nomRestaurant) AND
    Data.allInstances()->exists(d | d.data = dataFesta) AND
    Empresa.allInstances()->exists(e | e.nom = nomEmpresa) AND

    // Comprovar que persona homenatjada no sigui assistent d'una altra festa
    NOT Persona.allInstances()->exists(p |
        p.nom = nomPersona AND
        p.participacions.data = dataFesta AND p.participacions.horaInici = horaIniciFesta
    )
post:
    Festa.allInstances()->exists(f |
        f.ocllsNew() AND
        f.restaurant.nom = nomRestaurant AND f.data.data = dataFesta AND f.empresa.nom = nomEmpresa AND
        f.horaInici = horaIniciFesta AND
        f.ocllsTypeOf(Aniversari) AND
        f.ocllsType(Aniversari).edat = edatFesta AND f.ocllsType(Aniversari).homenatjada.nom = nomPersona AND
        result = f
    )
    ocllsTypeOf(Aniversari)
context:
    // NOTA: En aquest cas també afegirem els participants donat que son els mateixos (Enunciat)
    Sistema::altaOrganitzadors(objFesta : Festa, nomPersona : String)
pre:
    // Comprovar que organitzador no siguin assistents d'una altra festa
    NOT Persona.allInstances()->exists(p |
        p.nom = nomPersona AND
        p.participant.data = objFesta.data AND p.participant.horaInici = objFesta.horaInici
    )
post:
    objFesta.organitzades->exists(o | o.nom = nomPersona) AND
    Assistent.allInstances()->exists(a |
        a.ocllsNew() AND
        a.festa = objFesta AND a.persona.nom = nomPersona
        a.pagat = true
    )
    )

```

Consulta Aniversaris Estranys



```

////////////////////
// Consulta Aniversaris Estranys
////////////////////
context:
    // NOTA: Necessitem "Sequence" o "Bag" donat que nom Persona es pot repetir.
    Sistema::consultaAniversarisEstranys(nomEmpresa : String)
    : Sequence(TupleType(sNomPersona : String, setOrganitzadors : Set(String)))
pre:
    // ∀ t ∈ e.tipusDeFesta.nom -> (∃ f ∈ Festa | f.empresa.nom = e.nom ∧ f.oclIsTypeOf(t))
    TipusDeFesta.allInstances()->forAll(t |
        Festa.allInstances()->exists(f |
            f.empresa.nom = nomEmpresa AND f.oclIsTypeOf(t)
        )
    )
body:
    let SeqAniversaris : Sequence(Aniversari) = Aniversari.allInstances()->select(a |
        a.empresa.nom = nomEmpresa AND
        a.participacions->size() > 30 AND
        a.participacions->excludes(a.homenatjada)
    )

    in SeqAniversaris->collect[sa |
        Tuple {
            sNomPersona = sa.homenatjada.nom,
            setOrganitzadors = sa.organitzades.nom
        }
    ]

```