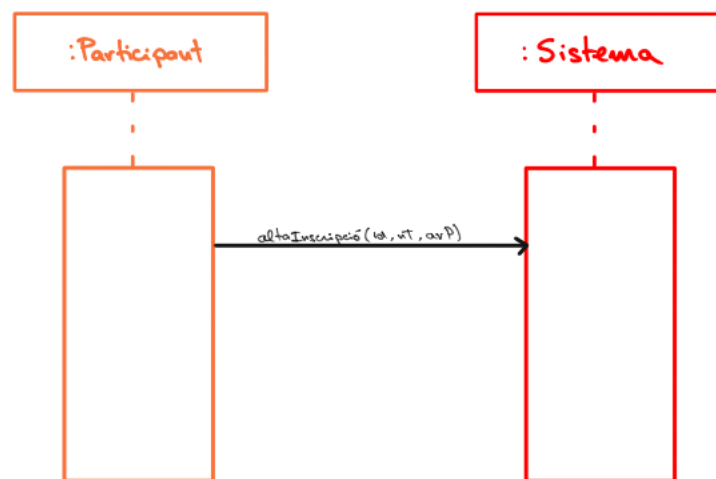




Exercici 3 - Taller

Alta Inscripció

Quan un participant vol inscriure's a un taller ell mateix introdueix el seu dni i el nom del taller al que vol inscriure's. Si el taller és de pagament, per defecte es considerarà inscripció no pagada i el participant haurà d'indicar l'avaluació que fa del seu preu. Considereu que una inscripció només es pot fer abans del començament del taller. Feu que la interacció necessària per dur a terme aquesta funcionalitat requereixi només un esdeveniment.

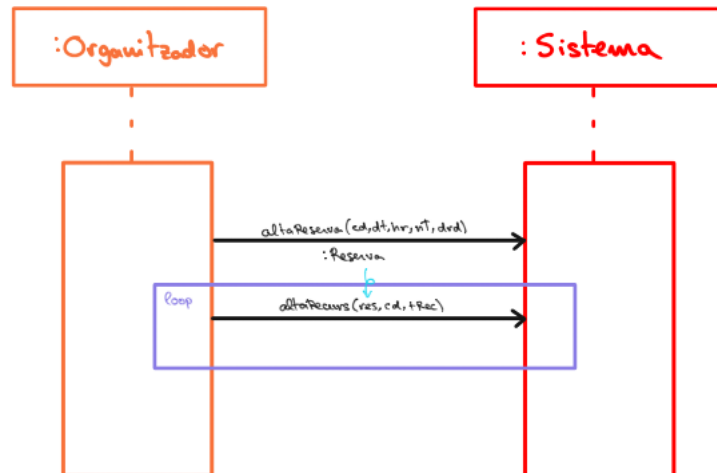


```
Context: Sistema::AltaInscripcio ( id: String, nT: String, avP: TPuntuacio )
pre: -- Suposem que la persona esta donada d'alta
    Taller.allInstances() -> exists ( t |
        t.nom = nT
        and t.dataIni > now() )

post: Taller.allInstances() -> exists ( t |
    t.nom = nT
    and if ( t.IsTypeOf( TallerDePagament ) ) then
        Inscripcio.allInstances() -> exists ( i |
            i.ocIsNew()
            and i.taller = nT
            and i.participant.dni = id
            and i.AsTypeOf ( InscripcioDePagament ).pagada = FALSE
            and i.AsTypeOf ( InscripcioDePagament ).avaluacioPreu = avP )
        else
            Inscripcio.allInstances() -> exists ( i |
                i.ocIsNew()
                and i.taller = nT
                and i.participant.dni = id )
            endif
        )
```

Alta Drets de Publicació

Quan un organitzador d'un taller vol fer una reserva de sala, introdueix al sistema les dades necessàries per fer-ho. Per cada recurs que es requereixi a la reserva, haurà d'informar-lo. Si el recurs és nou s'haurà de donar d'alta. Feu que la interacció necessària per dur a terme aquesta funcionalitat requereixi més d'un esdeveniment.



Context: Sistema::AltaReserva (cd: String, dt: Date, hr: Hora, nT: String, drd: Integer) : Reserva

```

pre:
  Sala.allInstances() -> exists ( s |
    s.codi = cd )
  and
  DataHora.allInstances() -> exists ( d |
    d.data = dt
    and d.hora = hr )
  and
  Taller.allInstances() -> exists ( t |
    t.nom = nT )
  
```

```

post:
  Reserva.allInstances() -> exists ( r |
    r.ocIsNew()
    and r.durada = drd
    and r.sala.codi = cd
    and r.datahora.hora = hr
    and r.datahora.data = dt
    and r.taller.nom = nT
    and result = r )
  
```

Context: Sistema::AltaRecurs (res: Reserva, cd: String, tRec: String) -- LOOP

```

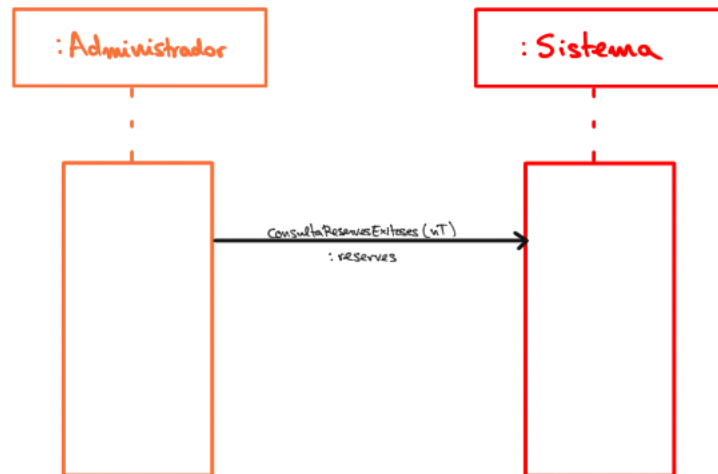
pre:
  
```

```

post:
  Recurs.allInstances() -> exists ( r |
    r.ocIsNew()
    and r.codi = cd
    and r.tipusRec = tRec
    and r.reserva = res )
  
```

Consulta Agressions Rellevants

Quan un administrador vol consultar les reserves més exitoses d'un taller de pagament indica al sistema el nom del taller. El sistema retorna, per a cada reserva amb més de 5 valoracions amb puntuació de 5 d'aquest taller: el codi de la sala reservada, la data i hora de la reserva i el conjunt de correus de participants inscrits. Aquesta funcionalitat solament pot demanar-se si hi ha com a mínim 3 inscripcions al taller pagades i amb avaluació de barat.




```

Context: Sistema::ConsultaReservesExitoses ( nT: String ) : reserves: Set ( TupleType (cd: String, dt: Date, hr: Hora, correus: Set ( corre
pre:  Inscricio.allInstances() -> select ( i |
      i.taller.nom = nT
      and i.pagada = TRUE
      and i.avaluacioPreu = TPreu::barat ) -> size() > 3

body:  let listR : Set ( Reserva ) = Reserva.allInstances() -> select ( r |
      r.taller.nom = nT
      and r.valoracions.allInstances() -> select ( v |
        v.puntuacio = TPuntuacio::5 ) -> size() > 5 )

      in result = listR -> collect ( r |
        Tuple { cd = r.sala.codi,
                dt = r.datahora.data,
                hr = r.datahora.hora,
                correus = r.taller.participant.correuE})
  
```

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