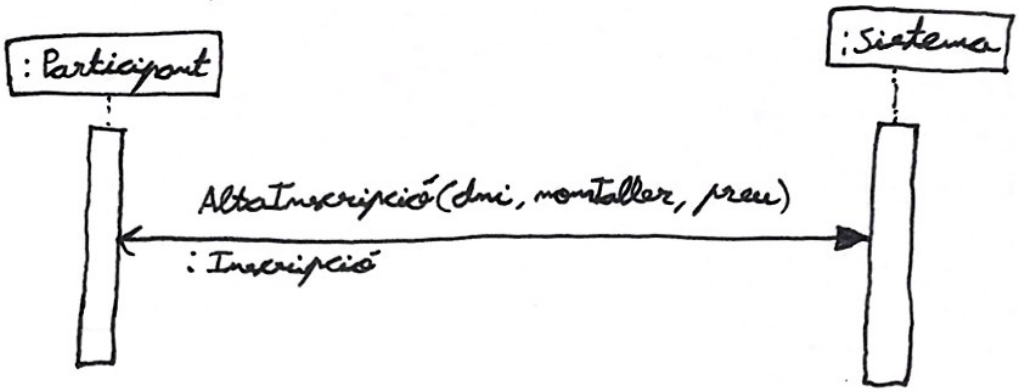
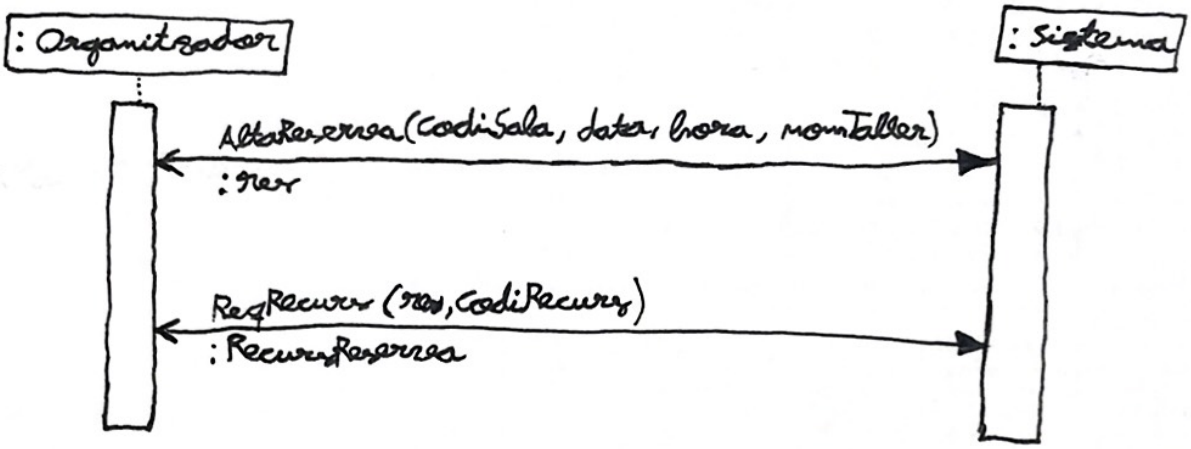
7è problema d'IES-QT2122

ANDREU ORENSANZ

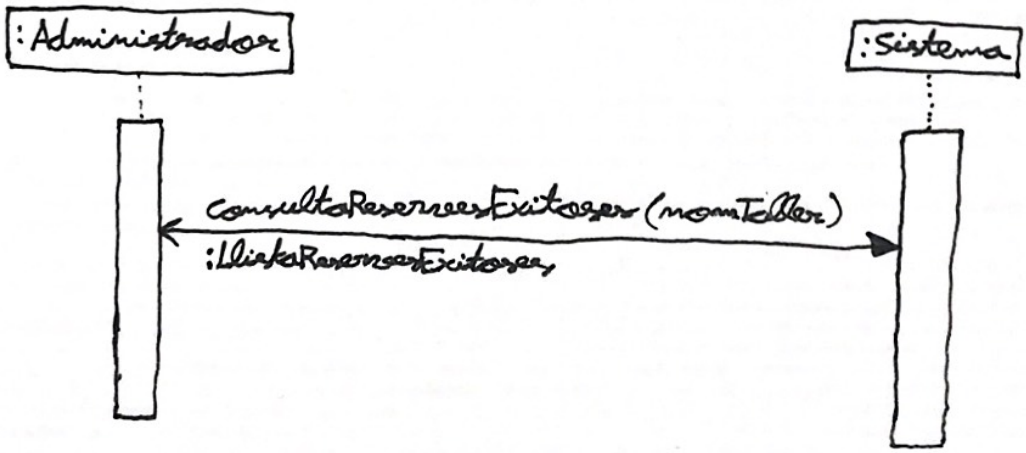
Alta d'Inscripció



Alta de Reserva



Consulta Reserves Exitoses



```
--Alta d'Inscripció
context Sistema::AltaInscripcio(dni: String, nomTaller: String, price: TPreu):Inscripcio
pre: Participant.allInstances() -> (p | p.dni = dni) and
    Taller.allInstances() -> (t | t.nom = nomTaller)
post: Inscripcio.allInstances() -> Exists(i | i.oclIsNew() and
                                              i.Participant.dni = dni and
                                              i.Taller.nom = nomTaller and
                                              if i.tipusIns == InscripcioDePagament then
                                              i.avaluacioPreu = price and i.pagada = false endif
                                              and result = i)
```

```
--Alta de Reserva
context Sistema::AltaReserva(codiSala: String, data: Date, hora: Time, nomTaller: String):res
pre: Sala.allInstances() -> Exists(s | s.codi = codiSala)
    DataHora.allInstances() -> Exists(dh | dh.data = data and dh.hora = hora)
    Taller.allInstances() -> Exists(t | t.nom = nomTaller)
post: Reserva.allInstances() -> Exists(r | r.oclIsNew() and
                                           r.Sala.codi = codiSala and
                                           r.Taller.nom = nomTaller and
                                           r.DataHora.data = data and
                                           r.DataHora.hora = hora and
                                           result = r)

context Sistema::ReqRecurs(res: Reserva, codiRecurs: String):RecursReserva
pre: Reserva.allInstances() -> Exists(r | r = res)
    Recurs.allInstances() -> Exists(rec | rec.codi = codiRecurs)
post: Reserva.allInstances() -> Exists(r | r.oclIsNew() and
                                           r.Recurs.codi -> includes(codiRecurs) and
                                           r = RecursReserva and
                                           result = r)

```

```
--Consulta Reserves Exitoses
context Sistema::consultaReservesExitoses(nomTaller: String):
      Set(tupletype(codiSala: String, data: Date, hora: Time, correus: String))LlistaReservesExitoses
pre: Taller.allInstances() -> Exists(t | t.nom = nomTaller) and
    Inscripcio.allInstances() -> select(i | if i.tipusIns = "InscripcioDePagament" then
                                            i.oclAsType(InscripcioDePagament).pagada = true and
                                            i.oclAsType(InscripcioDePagament).avaluacioPreu = TPreu::barat endif) -> size() > 3
body: let resultat: Set(Reserva) = Reserva.allInstances() -> select(r | r.Taller.nom = nomTaller and
                                                                        r.Valoracio -> select(v | v.puntuacio = TPuntuacio::5))
                                                                        -> size() > 5
     in result = resultat -> collect(r | Tuple {
                                         codiSala = r.Sala.codi,
                                         data = r.DataHora.data,
                                         hora = r.DataHora.hora,
                                         correus = r.Taller.Participant.correuE
                                        })
```