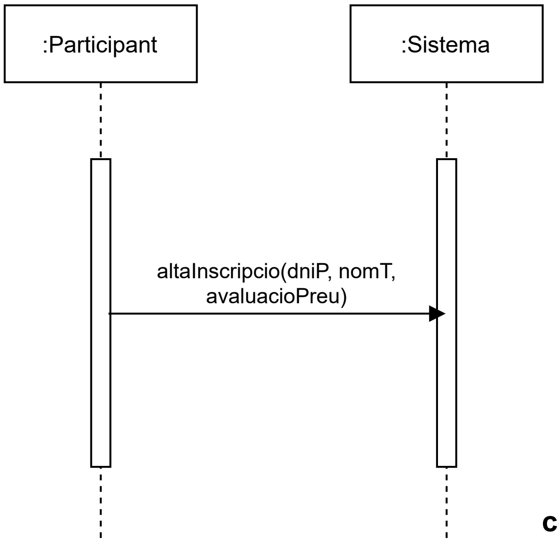
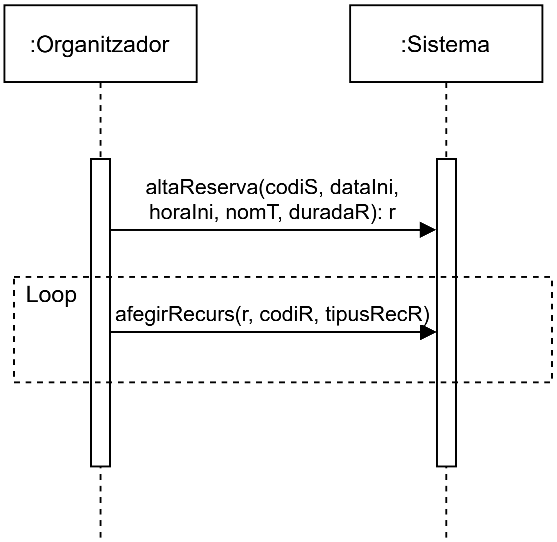
AltaInscripcio



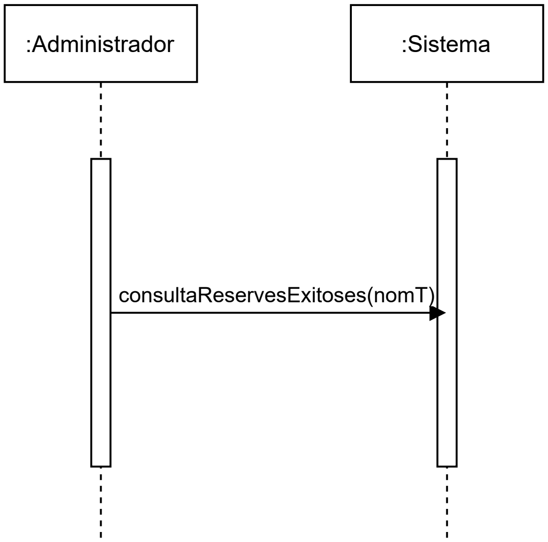
```
context: Sistema::altaInscripcio(dniP: String, nomT: String, avaluacioPreu: TPreu)

  pre: Taller.allInstances()->exists(t | t.nom = nomT) and Persona.allInstances()->exists(p |
  p.dni = dniP)

  post: Inscripcio.allInstances()->exists(i | i.oclIsNew() and i.taller.nom = nomT and
  i.participant.dni and

        if i.taller.tipusTaller = TallerDePagament then
            i.oclIsTypeOf(InscripcioDePagament) and
            i.oclAsType(InscripcioDePagament).pagada = false and
            i.oclAsType(InscripcioDePagament).avaluacioPreu = avaluacioPreu
        endif)
```

AltaReserva


:Organitzador
:Sistema
altaReserva(codiS, dataIni, horaIni, nomT, duradaR): r
Loop
afegirRecurs(r, codiR, tipusRecR)

```
context: Sistema::altaReserva(codiS: String, dataIni: Date, horaIni: Hora, nomT: String,
duradaR: Integer): Reserva

      pre: Sala.allInstances()->exists(s | s.codi = codiS) and

           Taller.allInstances()->exists(t | t.nom = nomT) and

           DataHora.allInstances()->exists(d | d.data = dataIni and d.hora = horaIni)

      post: Reserva.allInstances()->exists(r | r.oclIsNew() and r.sala.codi = codiS and

             r.datahora.data = dataIni and r.datahora.hora = horaIni and

             r.taller.nom = nomT and r.durada = duradaR and result = r)

  context: Sistema::afegirRecurs(r: Reserva, codiR: String, tipusRecR: String)

        post: if not Recurs.allInstances()@pre->exists(rec | rec.codi = codiR) then


                   Recurs.allInstances()->exists(rec | rec.oclIsNew() and rec.codi = codiR and

                 rec.tipusRec = tipusRecR and r.recurs = rec)

             else Recurs.allInstances()->exists(rec | r.recurs = rec)

             endif )
```

ConsultaReservesExitoses



```
context:  Sistema::consultaReservesExitoses(nomT: String) : Set (TupleType(codi: String,

data: Date, hora: Hora, correuEPartic: Set(String))

        pre: taller.allInstances() -> exists(t | t.nom = nomT and
             t.oclIsTypeOf(TalleDePagament)

        body: let reserves: Set(Reserva) = Reserva.allInstances()->

                  select(r | r.valoracio-> select(v | v.puntuacio = 5)->size() > 5)

              in result = reserves->collect(r | Tuple{codi = r.sala.codi; data = r.datahora.data;
                 hora = r.datahora.hora; correuEPartic = r.taller.participant.correuE} )
```