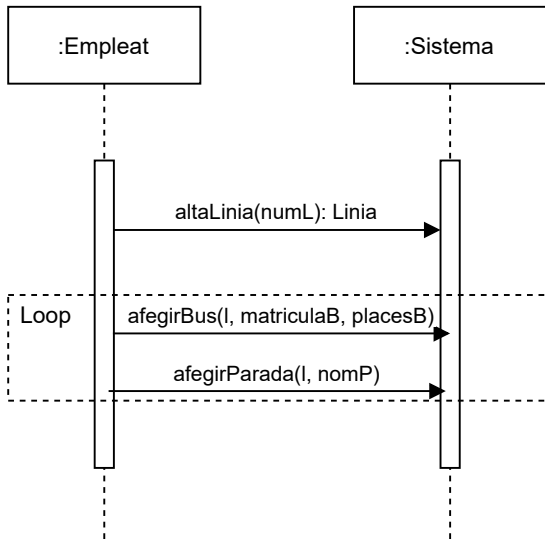


## AltaLinia



**context:** Sistema::altaLinia(numL: String): Linia

**pre:** --

**post:** Linia.allInstances()->exists(l | l.ocllsNew() and l.numero = numL and resultat = l)

**context:** Sistema::afegirBus(l: Linia, matriculaB: String, placesB: Integer)

**pre:** Bus.allInstances()->exists(b | b.matricula = matriculaB and b.places = placesB)

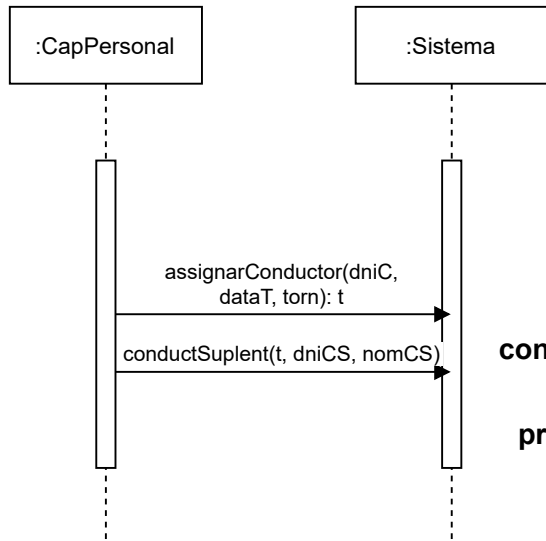
**post:** Bus.allInstances()->exists(b | b.matricula = matriculaB and b.places = placesB and l.bus = b)

**context:** Sistema::afegirParada(l: Linia, nomP: String)

**pre:** Parada.allInstances()->exists(p | p.nom = nomP)

**post:** Parada.allInstances()->exists(p | p.nom = nomP and l.parada = p)

## AssignarConductor



**context:** Sistema::assignarConductor(dniC: String, dataT: Data, torn: TipusTorn) : TornTreballat

**pre:** Conductor.allInstances()->exists(c | c.dni = dniC)

**post:** TornTreballat.allInstances()->exists(tt | tt.ocllsNew() and tt.data.data = dataT and tt.conductor.dni = dniC and tt.torn.torn = TipusTorn and

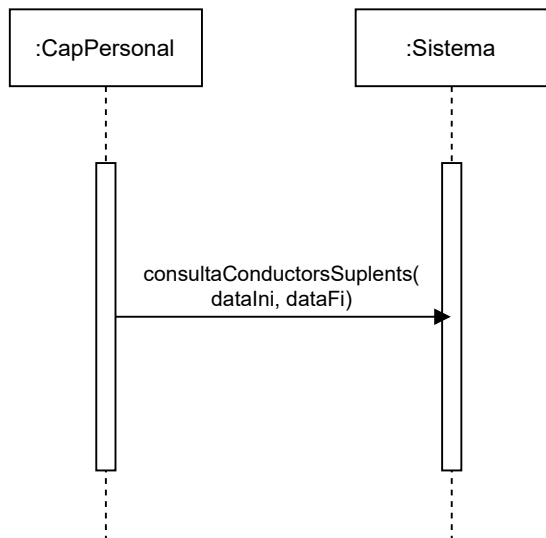
if tt.torn.torn = "TornNocturn" then

**context:** Sistema::conductSuplent(t: TornTreballat, dniCS: String, nomCS: String)

**pre:** TornTreballat.allInstances()->exists(tt | tt.data = t.data and tt.conductor = t.conductor and tt.torn = t.torn and tt.IsTypeOf(TornNocturn))

**post:** if not Conductor.allInstances()->exists(c | c.suplent.tornnocturn= t.ocllsNew(TornNocturn) then Conductor.allInstances()->exists(cs | cs.ocllsNew() and cs.dni = dniCS and cs.nom = nomCS and cs.suplent.tornnocturn= t)

## ConsultaConductorsSuplents



**context:** Sistema::consultaConductorsSuplents(dataIni: Date, dataFi: Date) : Set (TupleType(dni: String, linies: Set(Integer)))

**pre:** Conductor.allInstances()->exists(c | c.suplent.tornnocturn->size())>0 and not TornTreballar.allInstances()->exists(tt | tt.conductor.dni = c.dni and tt.ocllsTypeOf(TornNocturn) )

**body:** let conductors: Set(Conductor) = Conductor.allInstances()->select(c | TornTreballat.allInstances()->select(tt | tt.torn.torn != "TornNocturn" and tt.data.data > dataIni and tt.data.data < dataFi and tt.conductor.dni = c.dni)->size())>1 in conductors->collect(c | Tuple{dni = c.dni; linies=c.linia.numero})