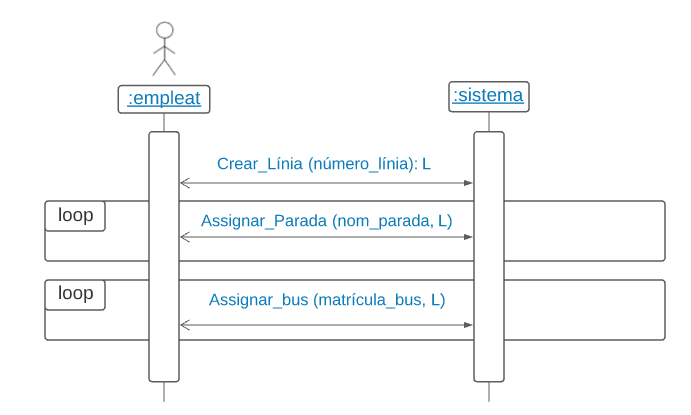
**4t PROBLEMA**

****

**Crear línia:**

Context:

Sistema::Crear\_Línia (número\_línia: Integer): Línia

Pre:

Post:

Línia.allInstances() -> exists (L | L.número = número\_línia

AND L.oclIsNew()

AND result = L)

**Assignar parada:**

Context:

Sistema::Assignar\_parada (nom\_parada: String, L: Línia)

Pre:

Post:

Parada.allInstances() -> exists (p | p.nom = nom\_parada

AND p.línia = L

AND p.oclIsNew())

**Assignar bus:**

Context:

Sistema::Assignar\_bus (matrícula\_bus: String, L: Línia)

Pre:

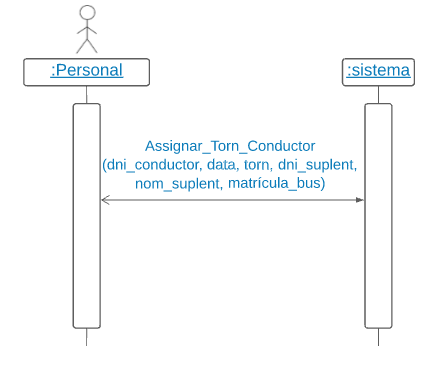
Bus.allInstances() -> exists (b | b.matrícula = matrícula\_bus

AND b.línia -> size() = 0)

Post:

Línia.allInstances() -> exists (L | L.número = número\_línia

AND L.bus.matrícula -> includes (matrícula\_bus))

****

**Assignar torn a conductor**

Context:

Sistema::Assignar\_Torn\_Conductor (dni\_conductor: String,

data: Date,

torn: TipusTorn,

dni\_suplent: String

nom\_suplent: String,

matrícula\_bus: String

)

Pre:

Conductor.allInstances() -> exists (c | c.dni = dni\_conductor)

AND Data.allInstances() -> exists (d | d.data = data)

AND Torn.allIsntances() -> exists (t | t.torn = torn)

AND Bus.allInstances() -> exists (b | b.matrícula = matrícula\_bus)

Post:

if not (Conductor.allInstances() -> exists (c | c.dni = dni\_suplent) then

Conductor.allInstances() -> exists (c | c.dni = dni\_suplent

AND c.oclIsNew()

AND c.nom = nom\_suplent)

endif

AND

TornTreballat.allInstances() -> exists (t | t.oclIsNew()

AND t.data.data = data

AND t.conductor.dni = dni\_conductor

AND t.torn.torn = torn

AND t.bus.matrícula = matrícula\_bus

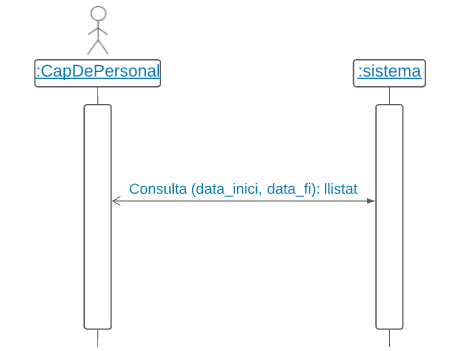
AND if (t.torn.torn = TipusTorn::nit) then

t.oclIsTypeOf(TornNocturn)

AND t.oclAsTypeOf(TornNocturn).suplent.dni = dni\_suplent

AND t.oclAsTypeOf(TornNocturn).suplent.nom = nom\_suplent

endif)

****

**Consulta conductors suplents**

Context:

Sistema::Consulta (data\_Inici: Date, data\_Fi: Date)

:llistat: Set(

tupleType(

dni\_suplent: String,

número\_línies: Set (Integer)))

Pre:

Conductor.allInstances() -> exists (c | c.TornNocturn -> size() > 0

AND not TornTreballat.allInstances() -> exists (t | t.conductor.dni = c.dni))

Body:

Let suplents: Set (Conductor) = Conductor.allInstances() -> select (c |

c.TornNocturn -> select (t | t.data.data >= data\_Inici

AND t.data.data <= data\_Fi) -> size() > 1)

in

suplents -> collect (c | Tuple {

dni\_suplent = c.dni,

AND número\_línies = c.línia.número

}

)