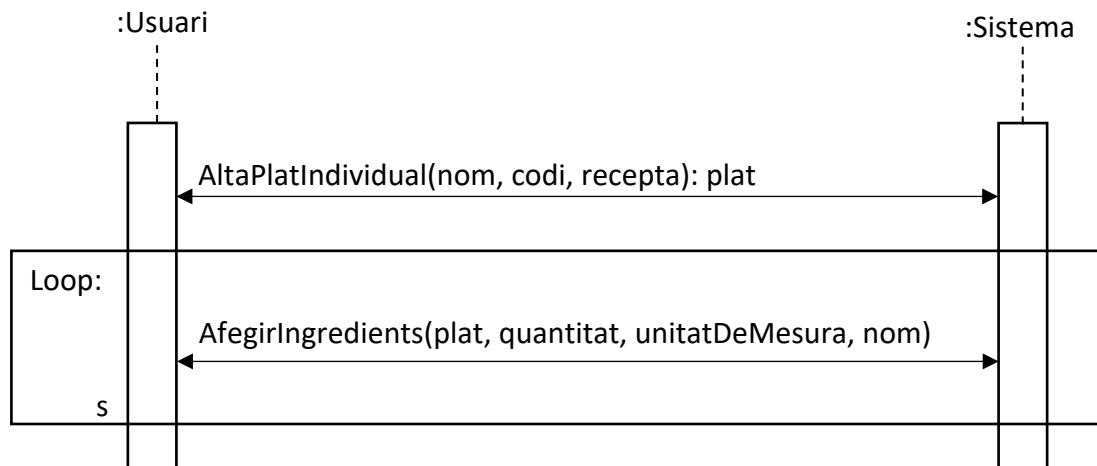


## 2on control IES QT2122 – Model del Comportament

1.

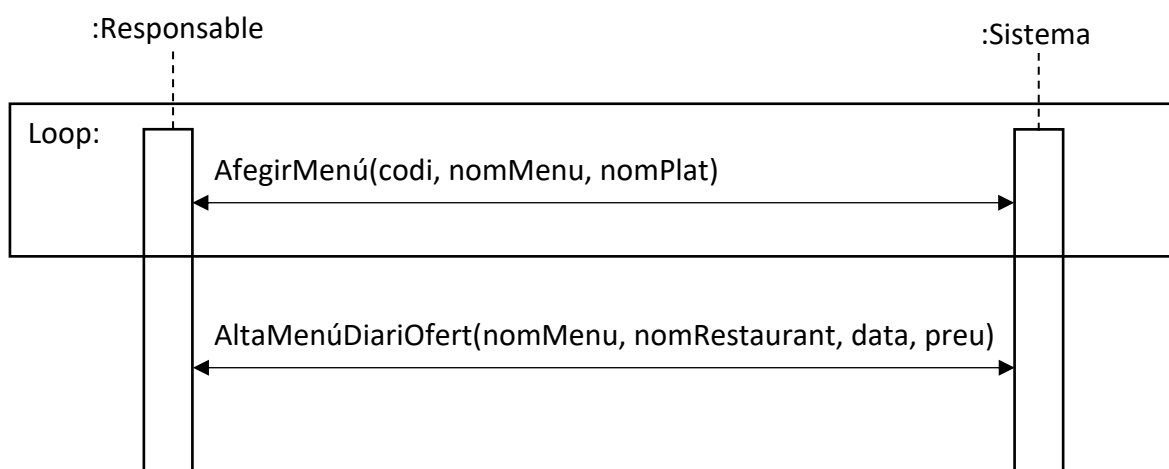


Alta Individual

```

context Sistema::AltaPlatIndividual(nom: String, codi: String, recepta: String): plat
pre:
post: PlatIndividual.allInstances() -> Exists(p | p.ocllsNew() and
|                                     p.nom = nom and
|                                     p.codi = codi and
|                                     p.recepta = recepta and
|                                     result = p)

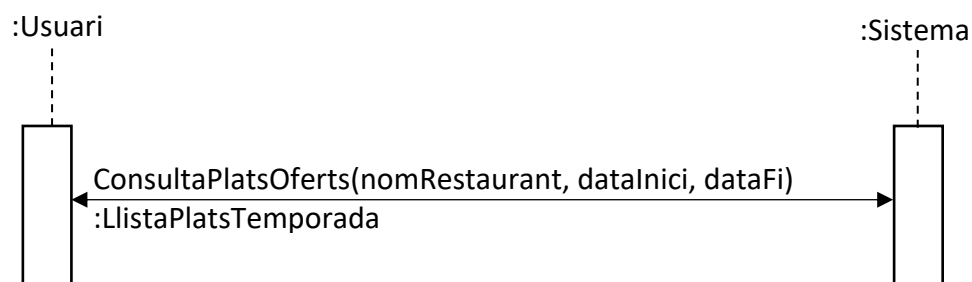
context Sistema::AfegirIngredients(plat: PlatIndividual, quantitat: Integer, unitatDeMesura: String, nom: String)
pre: Ingredient.allInstances() -> Exists(i | i.nom = nom)
post: Mesura.allInstances() -> Exists(mes | mes.ocllsNew() and
|                                     mes.PlatIndividual = pl and
|                                     mes.quantitat = quantitat and
|                                     mes.unitatDeMesura = unitatDeMesura
|                                     mes.Ingredient.nom = nom)
    
```



## Confecció Menú Diari

```
context Sistema::AltaMenu(codi: String, nomMenu: String, nomPlat: String)
pre: PlatIndividual.allInstances() -> Exists(p | p.nom = nomPlat)
post: if not Menu.allInstances()@pre -> Exists(m | m.nom = nomMenu) then
    Menu.allInstances() -> Exists(m | m.oclIsNew() and
        m.nom = nomMenu and
        m.codi = codi and
        m.PlatIndividual.nom = nomPlat) endif

context Sistema::AltaMenuDiariOfert(nomMenu: String, nomRestaurant: String, data: Date, preu: Decimal)
pre: Menu.allInstances() -> Exists(m | m.nom = nomMenu)
    Restaurant.allInstances() -> Exists(r | r.nom = nomRestaurant and
        r.PlatIndividual -> size() <= 100)
post: MenuDiariOfert.allInstances() -> Exists(menu | menu.oclIsNew() and
    menu.Menu.nom = nomMenu and
    menu.Restaurant.nom = nomRestaurant and
    menu.Data.data = data and
    menu.preu = preu)
```



## Consulta Plats Oferts

```
context Sistema::ConsultaPlatsOferts(nomRestaurant: String, dataInici: Date, dataFi: Date): Set(TupleType(codi: String,
    nomIngredients: Set(String)))
pre: MenuDiariOfert.allInstances() -> select(menu | menu.Data.data >= dataIni and
    menu.Data.data <= dataFi and
    menu.Restaurant.nom = nomRestaurant and
    menu.preu < 10) -> size() >= 10 -- Com a mínim 10 menús diaris de menys de 10€
body: let plats: Set(PlatIndividual) = PlatIndividual.allInstances() -> select(p | p.Restaurant.Data.data >= dataIni and
    p.Restaurant.Data.data <= dataFi and
    p.Restaurant.nom = nomRestaurant)
    in result = plats -> collect(p | Tuple {
        codi = p.codi,
        nomIngredients = p.Ingredient.nom -> asSet()
    })
```