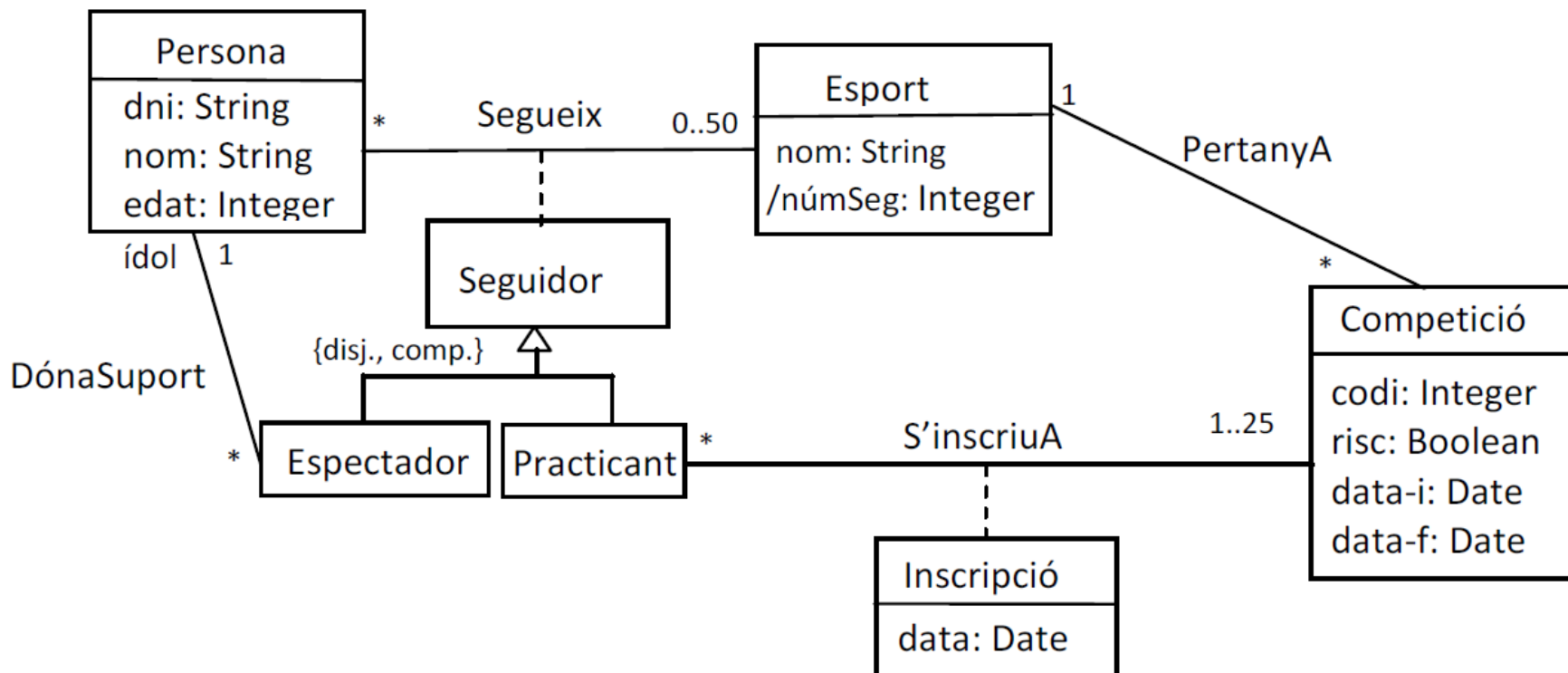


Problema de disseny en UML: Seguidors Esportius

Enunciat: esquema conceptual d'especificació



Enunciat: restriccions textuais

1. Claus externes: (Persona, dni); (Esport, nom); (Competició, codi)
2. Una persona menor de 18 anys no pot estar inscrita a competicions de risc.
3. La data d'una inscripció ha de ser anterior a la data d'inici de la competició.
4. L'esport del practicant d'una inscripció és el mateix que l'esport al qual pertany la seva competició.
5. L'ídol un espectador és practicant de l'esport de l'espectador.

Enunciat: informació derivada

1. /númSeg de la classe Esport és el total de seguidors d'aquell esport

Operació 1: *Obtenir Nombre Inscripcions*

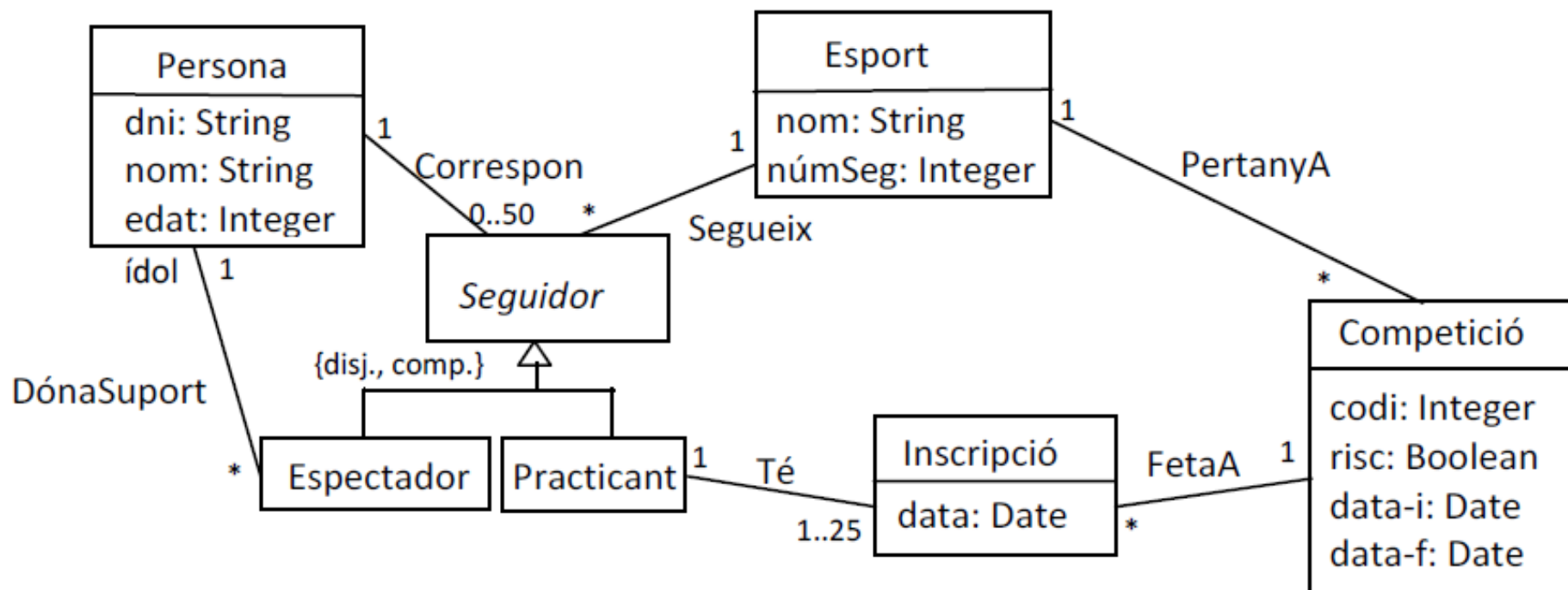
- **Operació:** `obtenirNombredInscripcions(dniP:String, nomE: String): TupleType(edat:Integer, númInscripcions: Integer)`
- **Pre:**
 - La persona amb *dniP* és seguidora de l'esport *nomE*
- **Body:**
 - Si el seguidor és “Practicant”, aleshores es retorna l'edat de la persona *dniP* i el nombre d'inscripcions que el practicant ha fet a competicions de risc de l'esport *nomE*.
 - Si el seguidor és “Espectador”, es retorna l'edat del seu ídol i el nombre d'inscripcions que l'ídol ha fet com a practicant a competicions de risc de l'esport *nomE*.

Operació 2: *Crea Practicant*

- **Operació:** `creaPracticant(dniP:String, nomE:String, codiC:Integer, dataInsc:Date)`
- **Pre:**
 - La persona *dniP* existeix
 - L'esport *nomE* existeix
 - La competició *codiC* existeix
- **Post:**
 - Es crea una instància de Practicant de la Persona *dniP* i l'esport *nomE*. Es crea una nova instància d'Inscripció definida pel practicant que s'acaba de crear i la competició *codiC*, amb la data *dataInsc*.

Diagrama de classes de disseny i contractes de les operacions de disseny

Solució: diagrama de classes de disseny



Restriccions textuais afegides

- *RT7: No hi pot haver dos Seguidors amb els mateixos Persona i Esport* *Clau associativa*
- *RT8: No hi pot haver dues Inscripcions amb els mateixos Practicant i Competició.*

Clau associativa

Solució:

contracte de disseny d'*obtenirNombreInscripcions*

- **Operació:** `obtenirNombredInscripcions(dniP:String, nomE: String): TupleType(edat:Integer, númInscripcions: Integer)`
- **Exc:**
 - [NoExisteixSeguidor] La persona amb *dniP* no és seguidora de l'esport *nomE*
- **Body:**
 - Si el seguidor és "Practicant", aleshores es retorna l'edat de la persona *dniP* i el nombre d'inscripcions que el practicant ha fet a competicions de risc de l'esport *nomE*.
 - Si el seguidor és "Espectador", es retorna l'edat del seu ídol i el nombre d'inscripcions que l'ídol ha fet com a practicant a competicions de risc de l'esport *nomE*.

Solució: contracte de disseny de *creaPracticant*

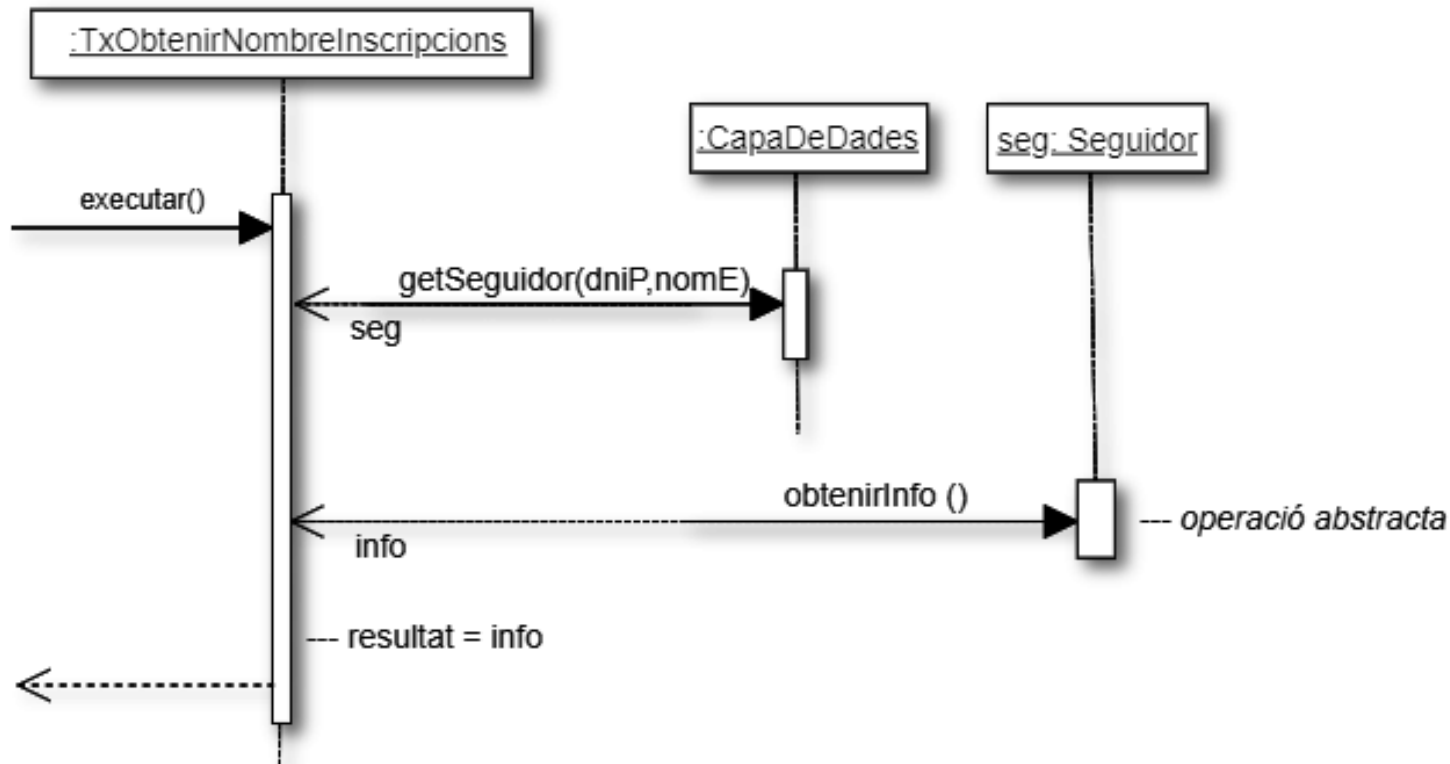
- Operació: *creaPracticant*(dniP:String, nomE:String, codiC:Integer, dataInsc:Date)
- Exc:
 - [NoExisteixPersona] La persona dniP no existeix
 - [NoExisteixEsport] L'esport nomE no existeix
 - [NoExisteixCompetició] La competició codiC no existeix
 - [jaExisteixSeguidor] la persona dniP ja és seguidor de l'esport nomE (RT7, clau externa de Seguidor)
 - [massaEsportiu] la persona dniP ja és seguidora de 50 esports (multiplicitat 0..50)
 - [massaPetit] la persona dniP és menor de divuit anys i la competició codiC és de risc (RT2)
 - [esportDiferent] la competició codiC no pertany a l'esport nomE (RT4)
 - [dataIncorrecta] dataInsc és posterior a data-i de la competició nomC (RT3)
- Post:
 - Es crea una instància de Practicant de la Persona *dniP* i l'esport *nomE*. Es crea una nova instància d'Inscripció definida pel practicant que s'acaba de crear i la competició *codiC*, amb la data *dataInsc*.
 - S'incrementa el valor de l'atribut *númSeg d'Esport*

Diagrama de seqüència: *obtenirNombreInsc*

Solució: contracte de disseny d'*obtenirNombreInscripcions*

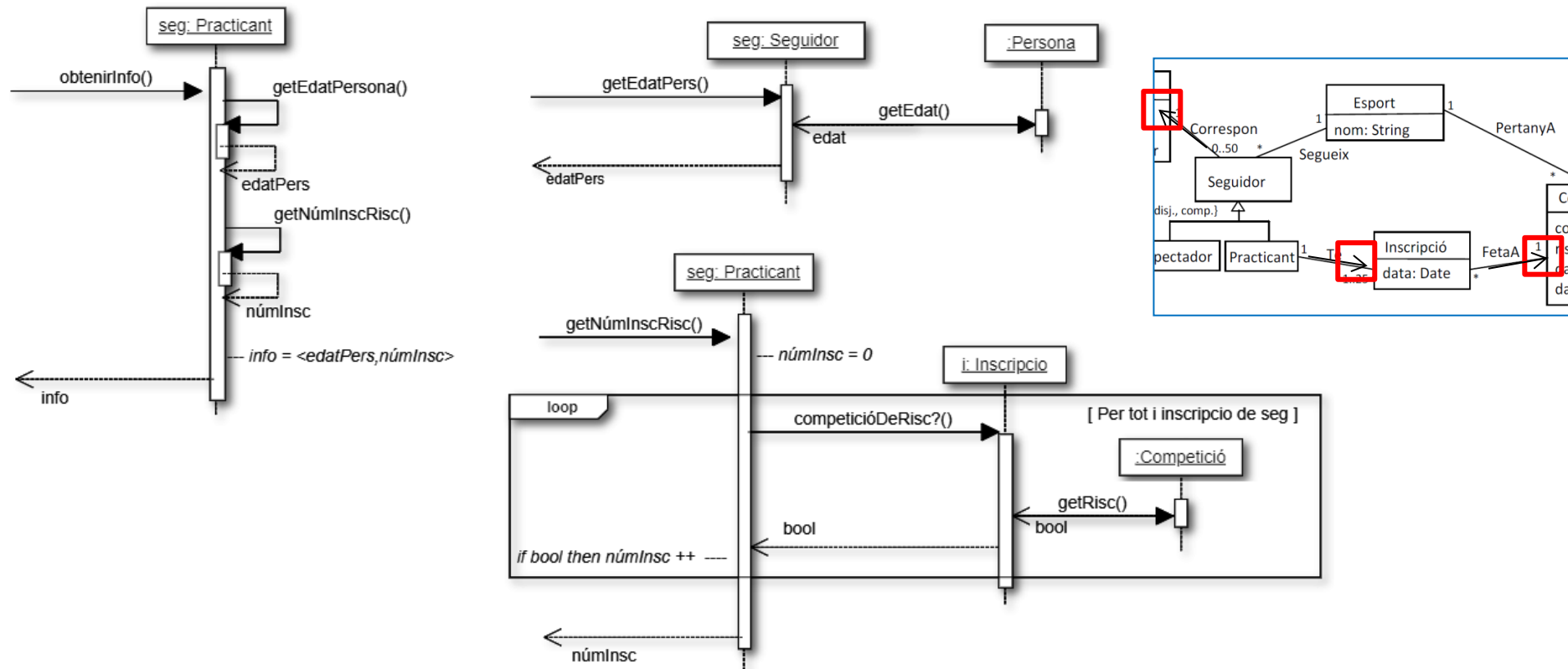
- **Operació:** `obtenirNombreInscripcions(dniP:String, nomE: String): TupleType(edat:Integer, númInscripcions: Integer)`
- **Exc:**
 - [NoExisteixSeguidor] La persona amb *dniP* no és seguidora de l'esport *nomE*
- **Body:**
 - Si el seguidor és “Practicant”, aleshores es retorna l'edat de la persona *dniP* i el nombre d'inscripcions que el practicant ha fet a competicions de risc de l'esport *nomE*.
 - Si el seguidor és “Espectador”, es retorna l'edat del seu ídol i el nombre d'inscripcions que l'ídol ha fet com a practicant a competicions de risc de l'esport *nomE*.

Executar



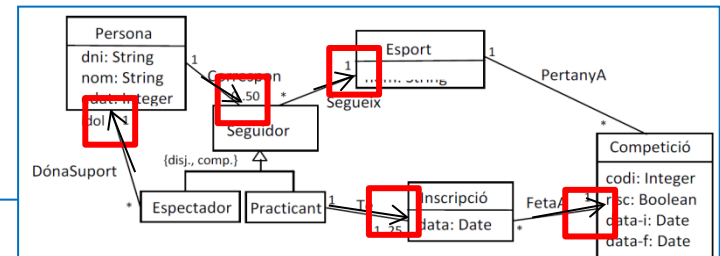
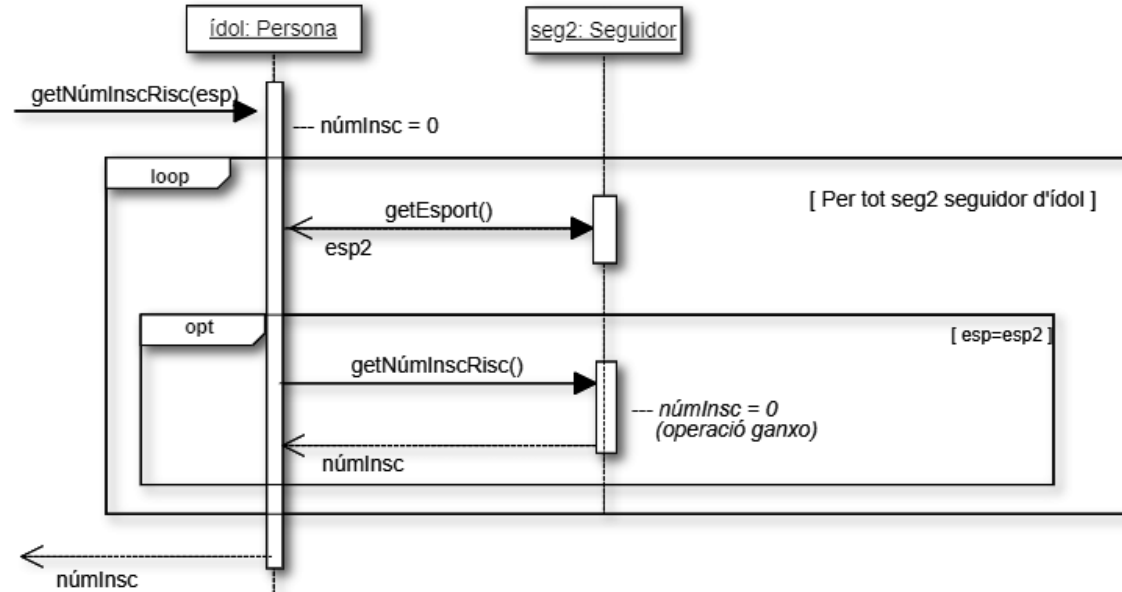
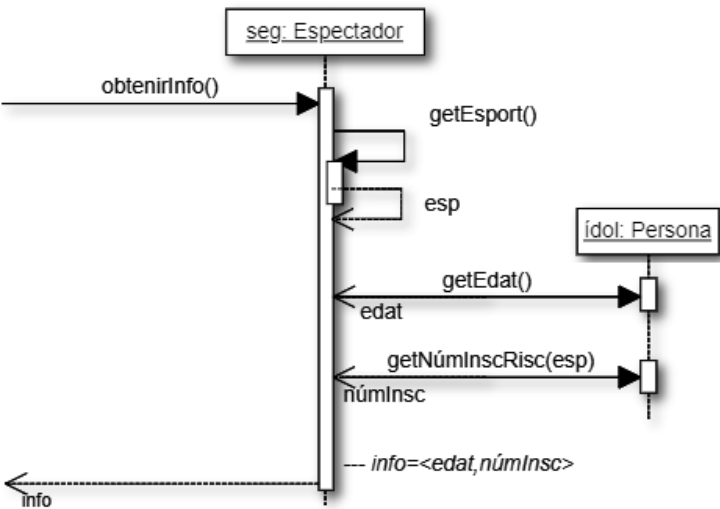
- Obtenim el seguidor de la Capa de Dades.
 - Això pot llençar l'excepció [NoExisteixSeguidor]
- Deleguem la resta de la operació a la classe Seguidor
- Ens guardem el resultat a l'atribut result del controlador transacció

Practicant::obtenirInfo



- Per al Practicant seg:
 - Obtenim el **dni** de la **Persona** practicant
 - Obtenim el seu nombre d'inscripcions a competicions de risc
- Afegim una tupla a info amb dniPers i númInsc

Espectador::obtenirInfo



- Per a l'Espectador seg:
 - Obtenim l'edat del seu ídol
 - Obtenim el nombre d'inscripcions del seu ídol a competicions de risc d'esp
- Afegim una tupla a info amb dniPers i númInsc

Navegabilitats resultants de la consultora

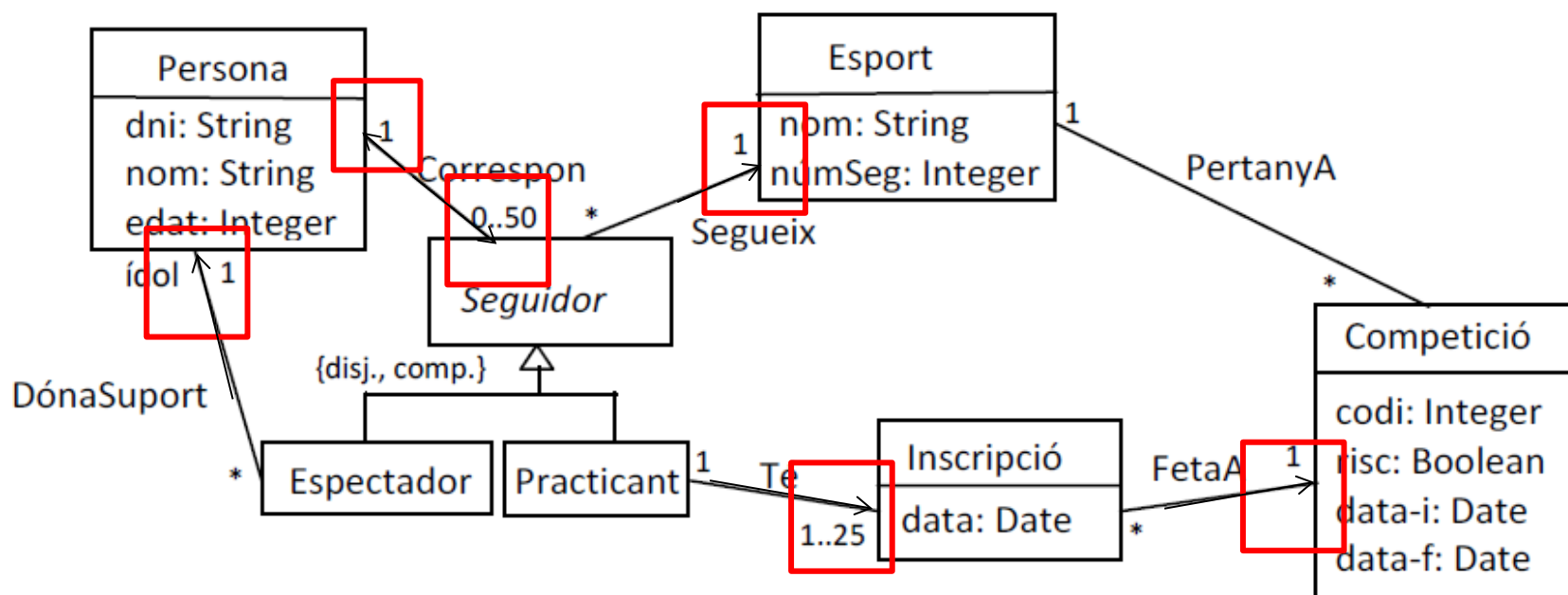


Diagrama de seqüència: creaPracticant

Solució: contracte de disseny de *creaPracticant*

- Operació: *creaPracticant*(dniP:String, nomE:String, codiC:Integer, dataInsc:Date)
- Exc:
 - [NoExisteixPersona] La persona dniP no existeix
 - [NoExisteixEsport] L'esport nomE no existeix
 - [NoExisteixCompetició] La competició codiC no existeix
 - [jaExisteixSeguidor] la persona dniP ja és seguidor de l'esport nomE
 - [massaEsportiu] la persona dniP ja és seguidora de 50 esports
 - [massaPetit] la persona dniP és menor de divuit anys i la competició codiC es de risc
 - [esportDiferent] la competició codiC no pertany a l'esport nomE
 - [dataIncorrecta] dataInsc és posterior a data-i de la competició nomC
- Post:
 - Es crea una instància de Practicant de la Persona *dniP* i l'esport *nomE*. Es crea una nova instància d'Inscripció definida pel practicant que s'acaba de crear i la competició *codiC*, amb la data *dataInsc*.
 - S'incrementa el valor de l'atribut *númSeg* d'*Esport*

Capa de Dades
Capa de Dades
Capa de Dades
Capa de Dades

Expert

Persona

Persona

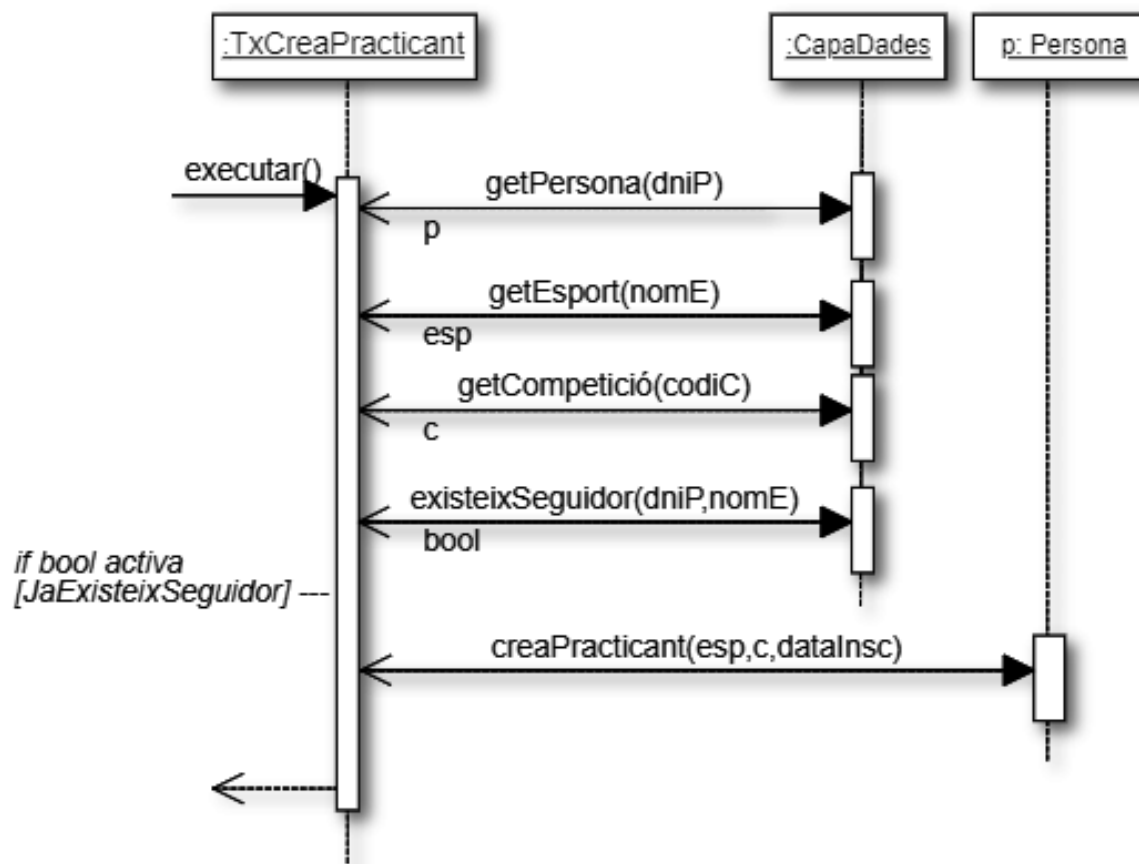
Competició o Esport

Competició

Persona o Esport

Esport

Executar

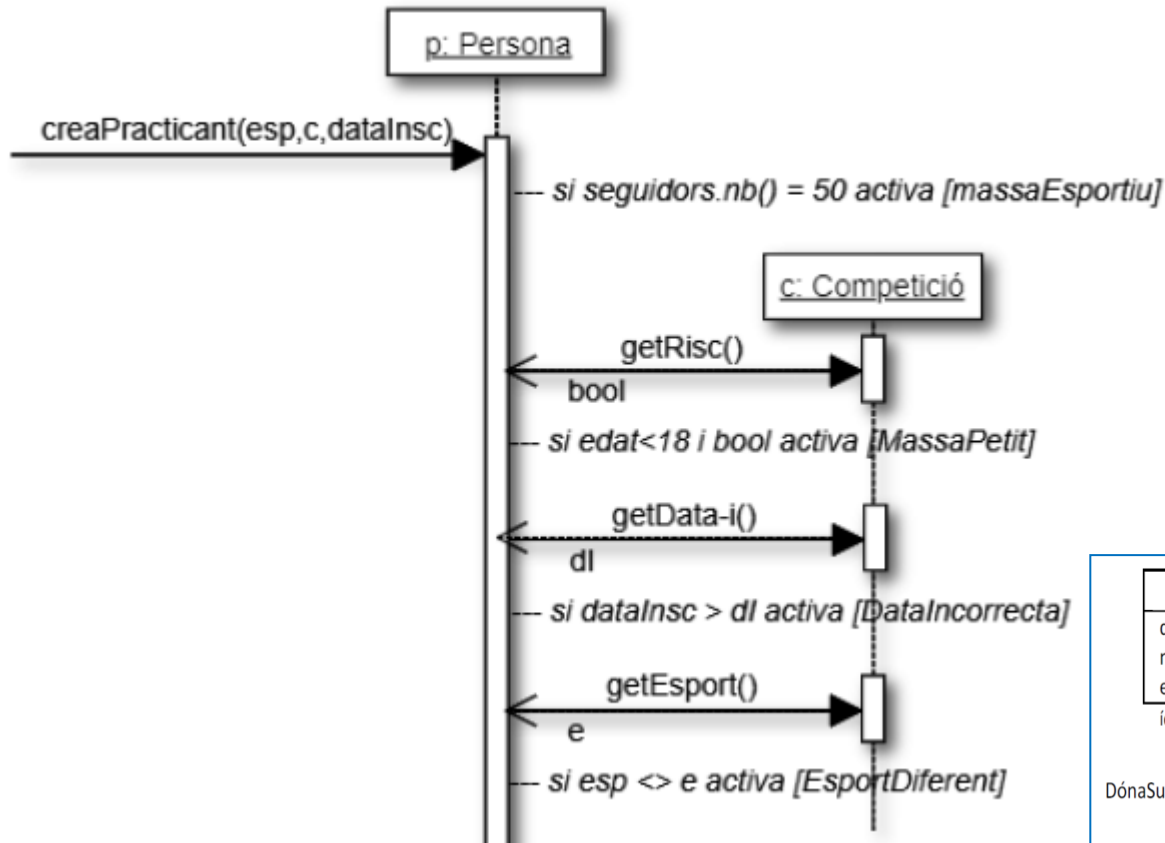


- **Obtenim els objectes**, cosa que pot llençar les excepcions:
 - `[NoExisteixPersona]`, `[NoExisteixEsport]`, `[NoExisteixCompetició]`
- Comprovem que **no existeixi el Seguidor**
- Finalment **deleguem a Presona**

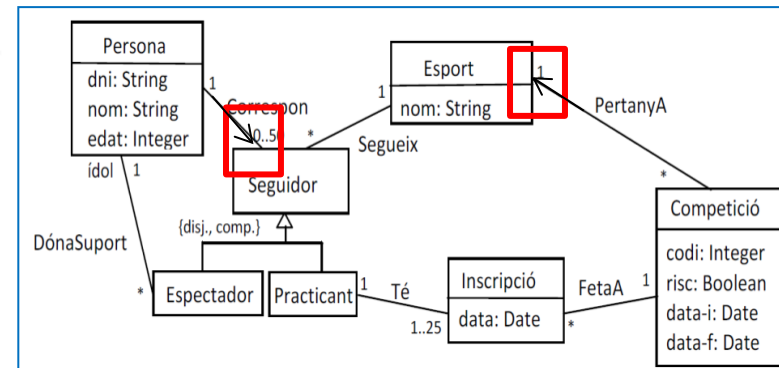
Persona::creaPracticant(esp,c,dataInsc)

- Què ha de fer aquesta operació?
 - Acabar de **comprovar les excepcions que queden** (o delegar en qui correspongui perquè ho faci):
 - [massaEsportiu] la persona dniP ja és seguidora de 50 esports
 - [massaPetit] la persona dniP és menor de divuit anys i la competició codiC és de risc
 - [esportDiferent] la competició codiC no pertany a l'esport nomE
 - [dataIncorrecta] dataInsc és posterior a data-i de la competició nomC
 - **Crear el Practicant i la seva Inscripció** tenint en compte TOTES les navegabilitats
 - Incrementar l'atribut **númSeg**

Persona::creaPracticant(esp,c,dataInsc) (I)

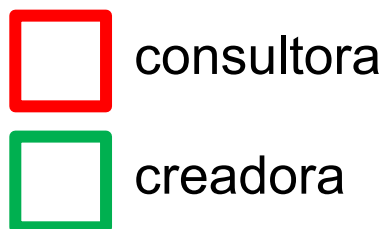
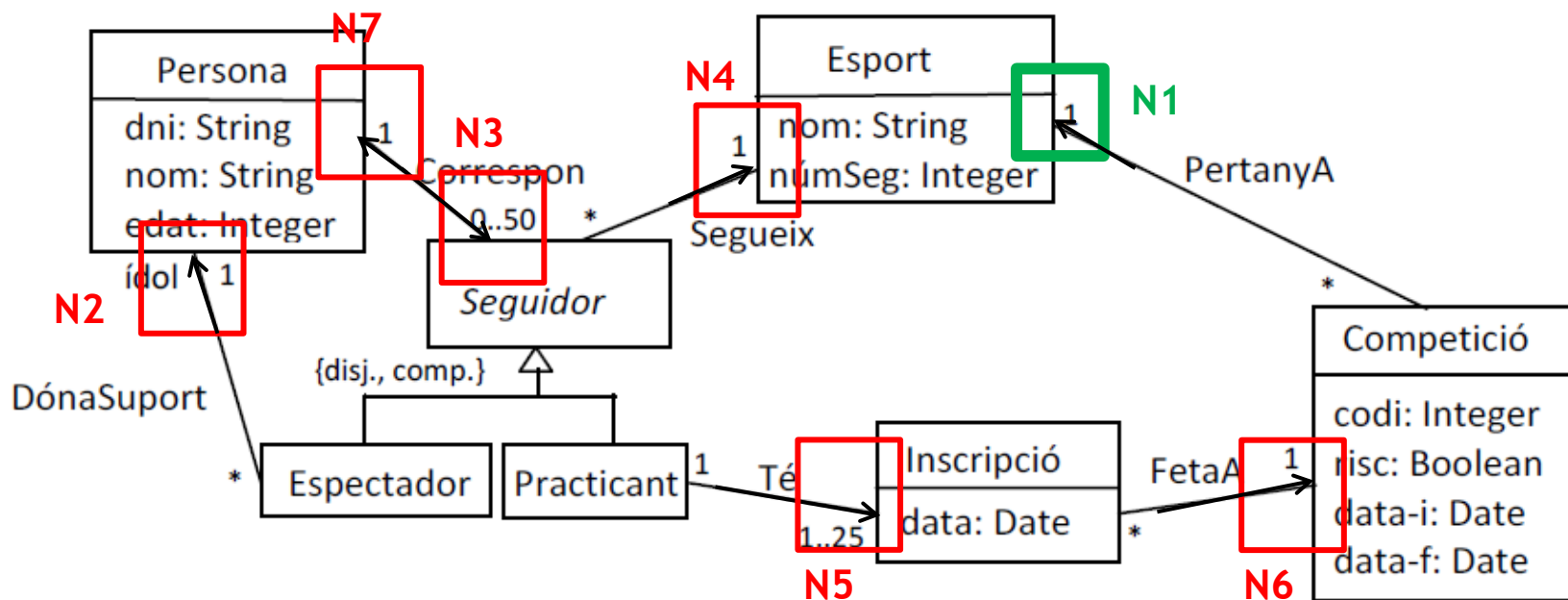


després crearem el Practicant i la Inscripció

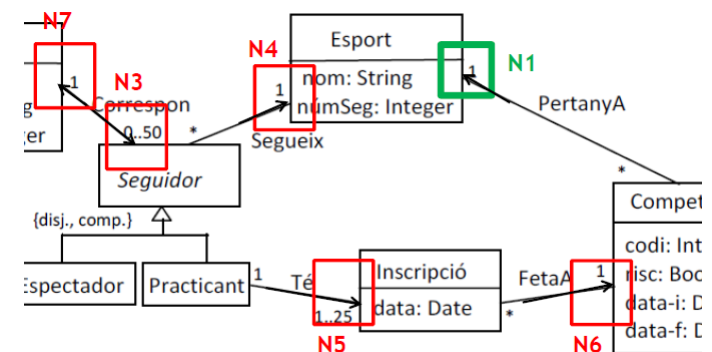
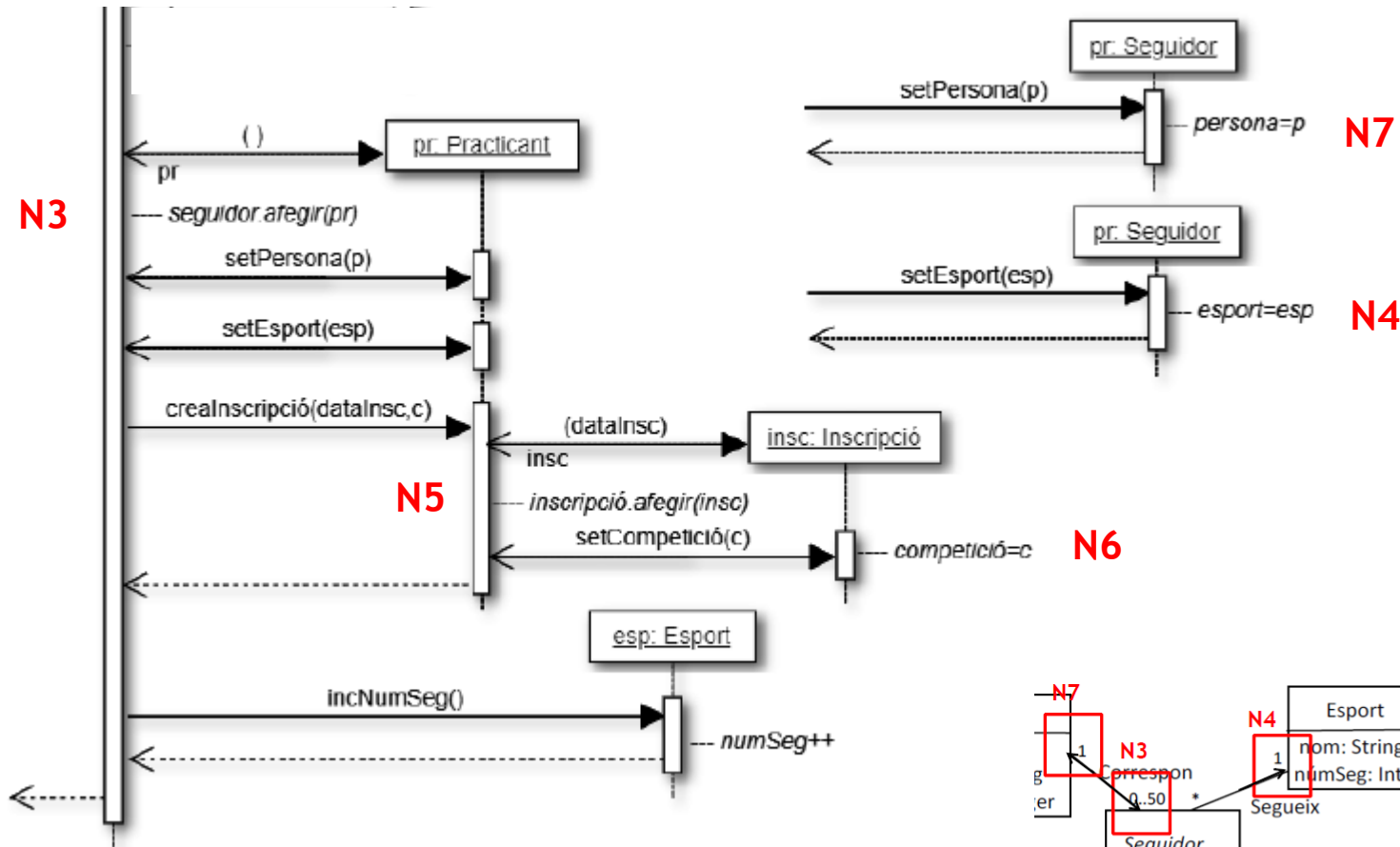


- Comprovem les excepcions: [massaEsportiu], [massaPetit], [dataIncorrecta], [esportDiferent]

Navegabilitats resultants



Persona::creaPracticant(esp,c,dataInsc) (II)



Problema de disseny en UML: Seguidors Esportius