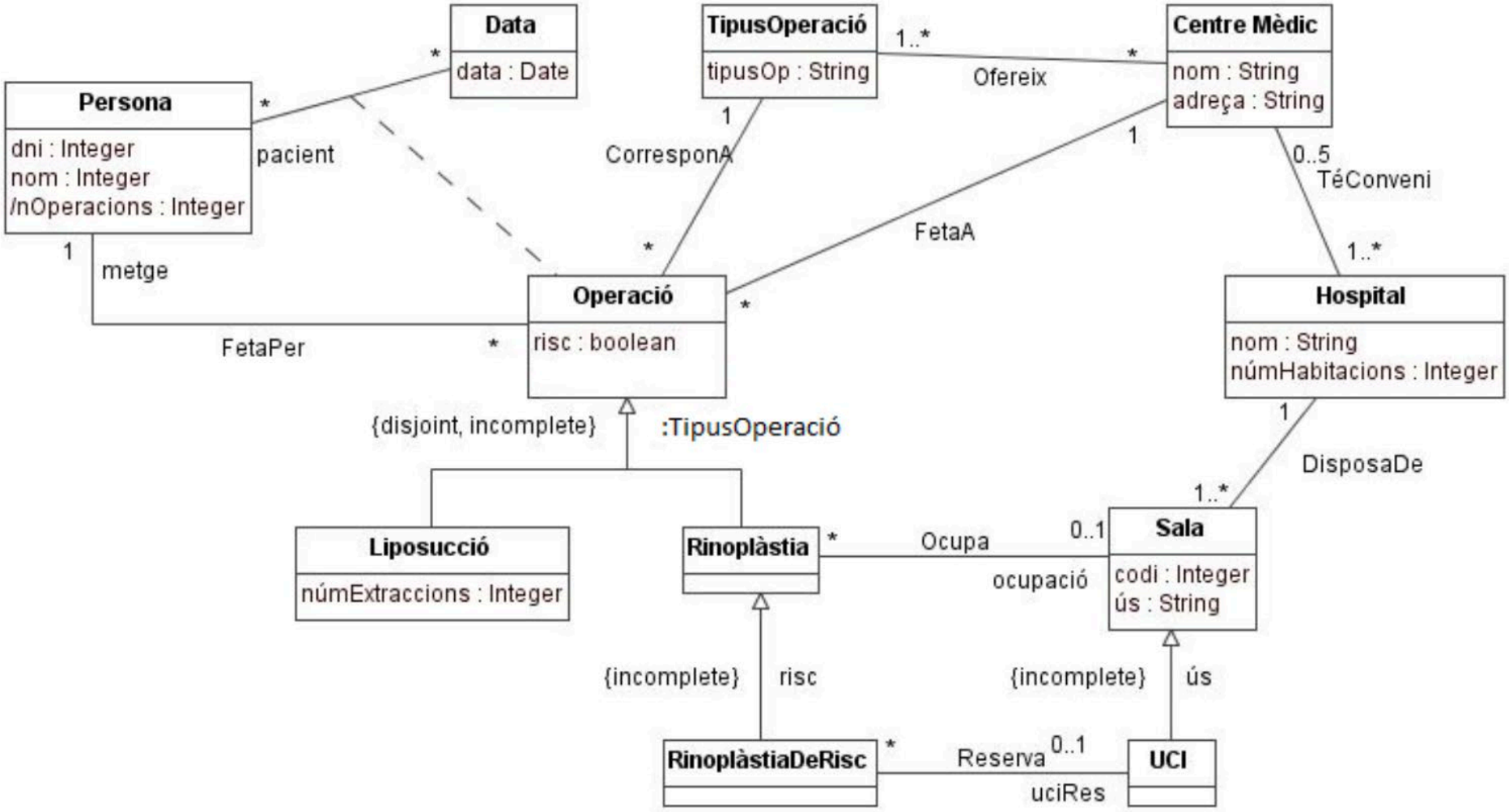
Problema de disseny en UML: Centres Mèdics

Enunciat: Presentació

Una consorci de centres mèdics necessita un sistema software que gestioni la informació de les operacions de cirurgia estètica que realitza.

Enunciat: Diagrama de Classes de Disseny

aquest sistema.



Enunciat: Restriccions Textuals

- Claus externes: (Persona, dni), (Data, data), (TipusOperació, tipusOp), (Centre Mèdic. Nom), (Hospital, nom).
- El metge i el pacient d'una operació no poden coincidir.
- El tipus que correspon a una operació és un dels que ofereix el centre mèdic on es fa l'operació.
- Un metge fa un màxim de tres operacions diàries per a un mateix centre mèdic.
- Si una Rinoplàstia té risc=cert, aleshores ha de ser del tipus 'RinoplàstiaDeRisc'.
- Un hospital no pot disposar de dues sales amb el mateix codi.
- La sala que ocupa una rinoplàstia ha de ser d'un hospital que té conveni amb el centre on es fa l'operació.
- L''UCI reservada per una rinoplàstia de risc ha de ser del mateix hospital de la sala que ocupa.

Operació 1: getNumRinoplàsties

- **Operació:** getNumRino(dniP: Integer, nomCM: String, nomH: String): Integer
- **Exc:**
  - [NoExisteixPersona]: la Persona identificat per *dniP* no existeix.
  - [NoExisteixCentreMedic]: El Centre Mèdic identificat per *nomCM* no existeix.
  - [NoExisteixHospital]: l'Hospital identificat per *nomH* no existeix.

- **Body:**
  - Es retorna el nombre de vegades que la persona *nomP* ha sigut pacient d'una rinoplàstia de risc feta al centre mèdic *nomCM* i que ha reservat una UCI de l'hospital *nomH* i on el metge que fa l'operació hagi sigut pacient com a mínim de 10 operacions.

Operació 2: ProgramarRinoplàstiaRisc

- **Operació:** progRinoRisc (dniP: Integer, data: Date, dniM: Integer, nomC: String)
- **Exc:**
  - [NoExisteixPersona]: el Pacient identificat per *dniP* no existeix.
  - [NoExisteixMetge]: El Metge identificat per *dniM* no existeix.
  - [NoExisteixCentreMedic]: El Centre Mèdic identificat per *nomC* no existeix.
  - [NoExisteixTipusRinoplàstia]: La beguda identificada per *nomBeguda* no existeix.
  - [OperacióExisteix]: ja existeix l'operació amb Data, data i pacient amb dniP
  - [Autooperació]: el metge i el pacient son la mateixa persona (RT2)
  - [CentreMedicIncorrecte]: El centreMèdic no ofereix Rinoplastia (RT3)
  - [MetgeLimitOperacions]: El metge ja ha fet 3 operacions (RT 4)

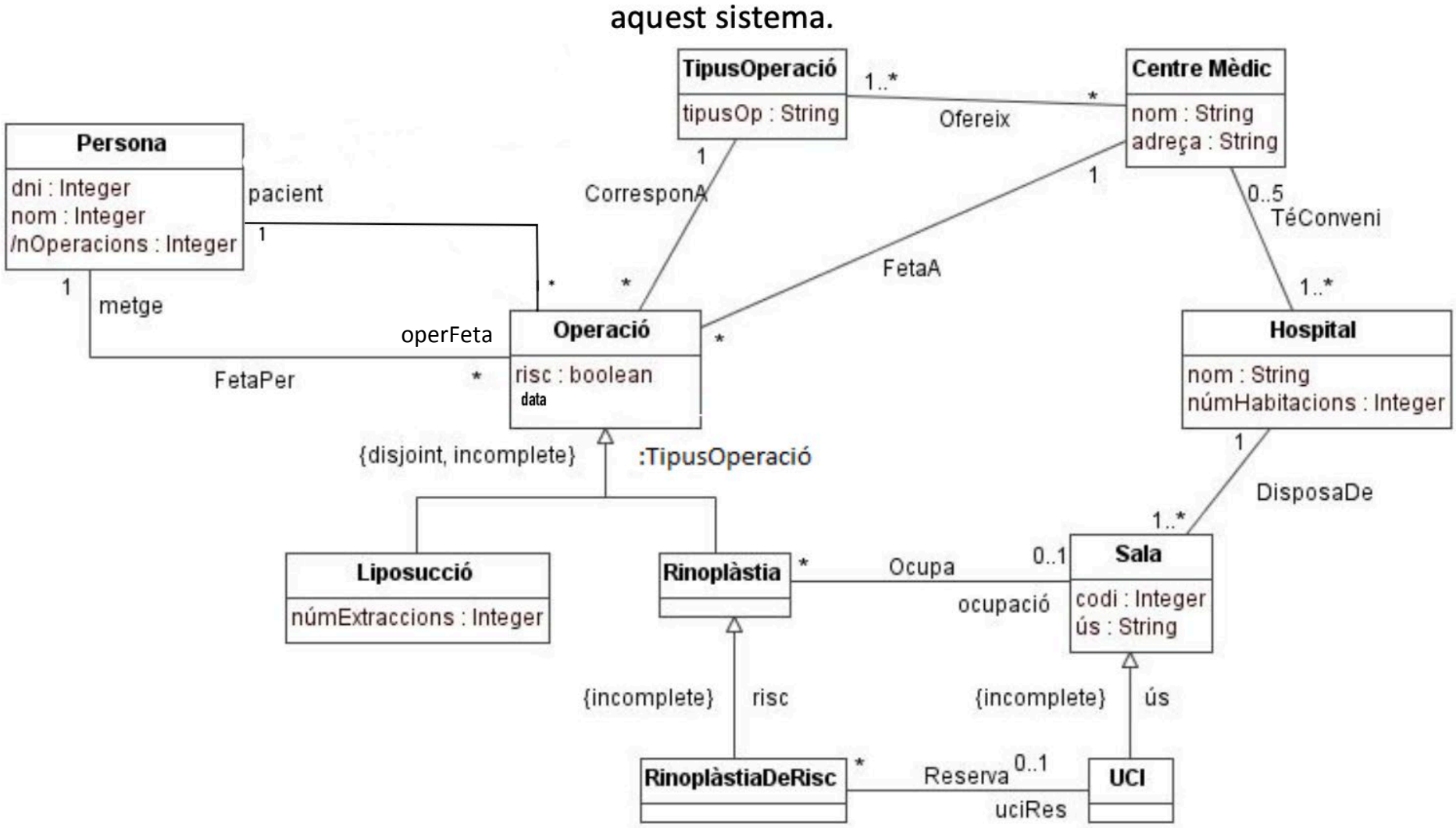
- **Post:**
  - Es dóna d'alta una instància de RinoplàstiaDeRisc amb els atributs i les associacions corresponents entre la Persona *dniP* i la Data *data*.
  - (extra) nOperacions ha d'incrementar-se

Enunciat: Què es demana?

- Diagrama de classes de disseny obtingut a partir de l'esquema conceptual de les dades, indicant explícitament les restriccions d'integritat que apareixen o desapareixen, i els contracte de les operacions obtinguts com a conseqüència de la traducció de l'esquema d'especificació al de disseny.
- Assumint que es vol utilitzar el controlador transacció, dissenyeu el diagrama de seqüència de l'operació *getNumRinoplàsties* i de totes les operacions que siguin invocades en aquest diagrama de seqüència. Poseu comentaris de tot el que no hi aparegui de forma explícita. Indiqueu clarament les operacions abstractes.
- Assumint que es vol utilitzar el controlador transacció i que l'atribut *nOperacions* de *Persona* és materialitzat, dissenyeu el diagrama de seqüència de l'operació *programarRinoplàstiaRisc* i de totes les operacions que siguin invocades en aquest diagrama de seqüència. Poseu comentaris de tot el que no hi aparegui de forma explícita. Indiqueu clarament les operacions abstractes.
- Indiqueu al diagrama de classes de l'apartat a) la navegabilitat resultant del vostre disseny.

Solució Diagrama

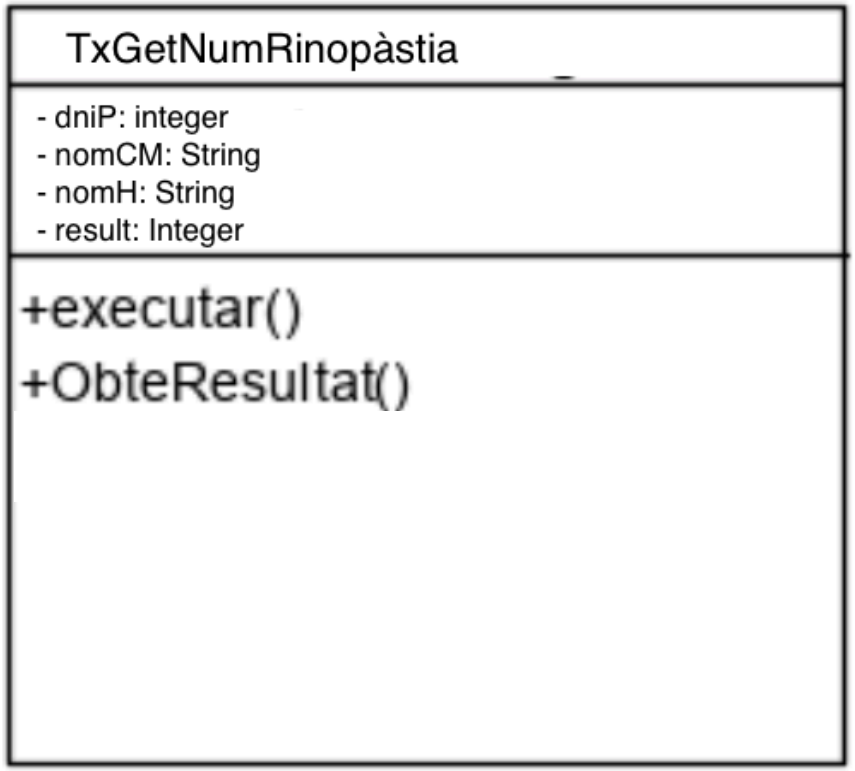
Solució Diagrama



RT: No hi ha 2 operacions amb el matrix pacient i data

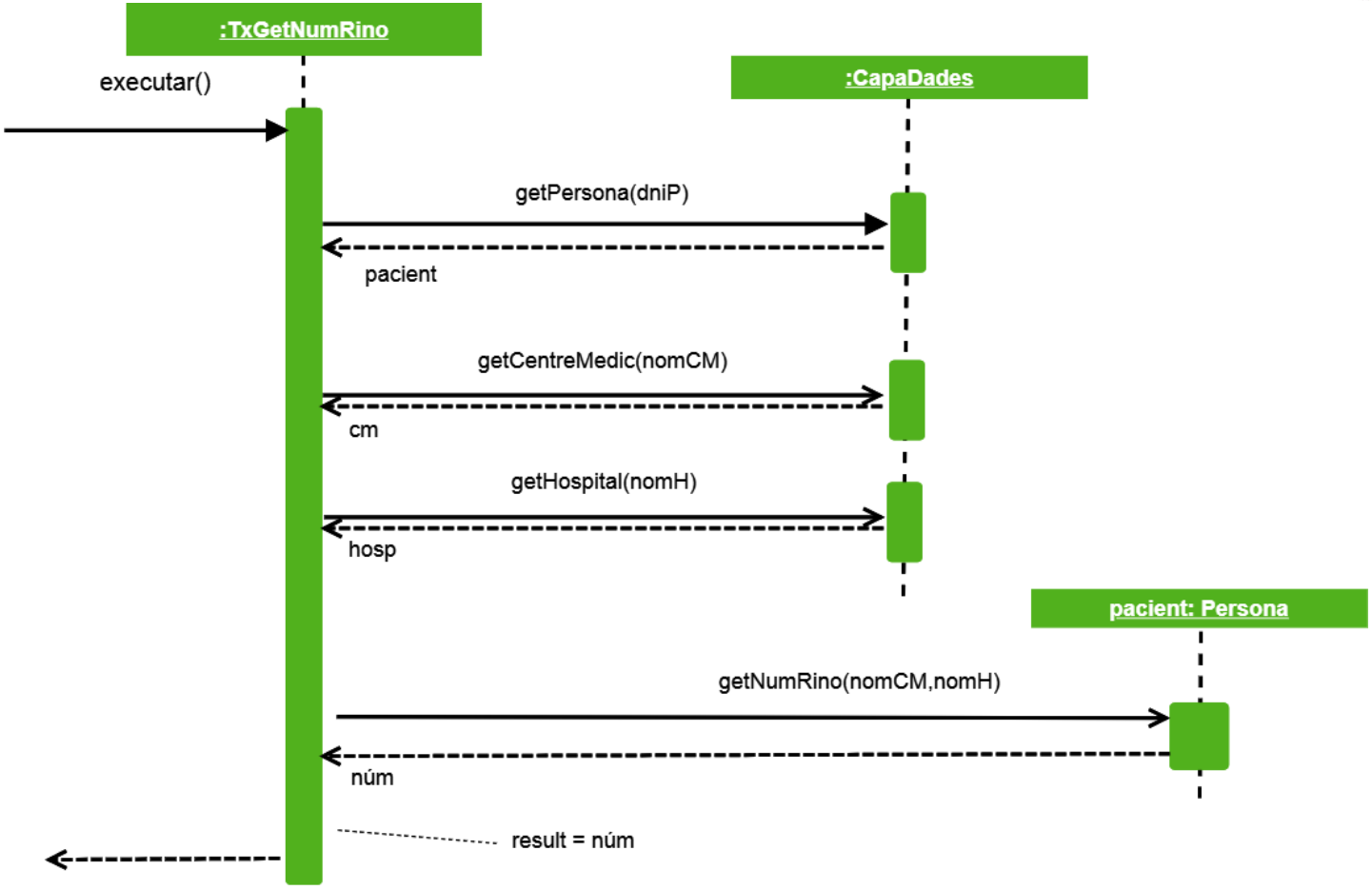
Solució getNumRinoplàstia

getNumRinoplàstia



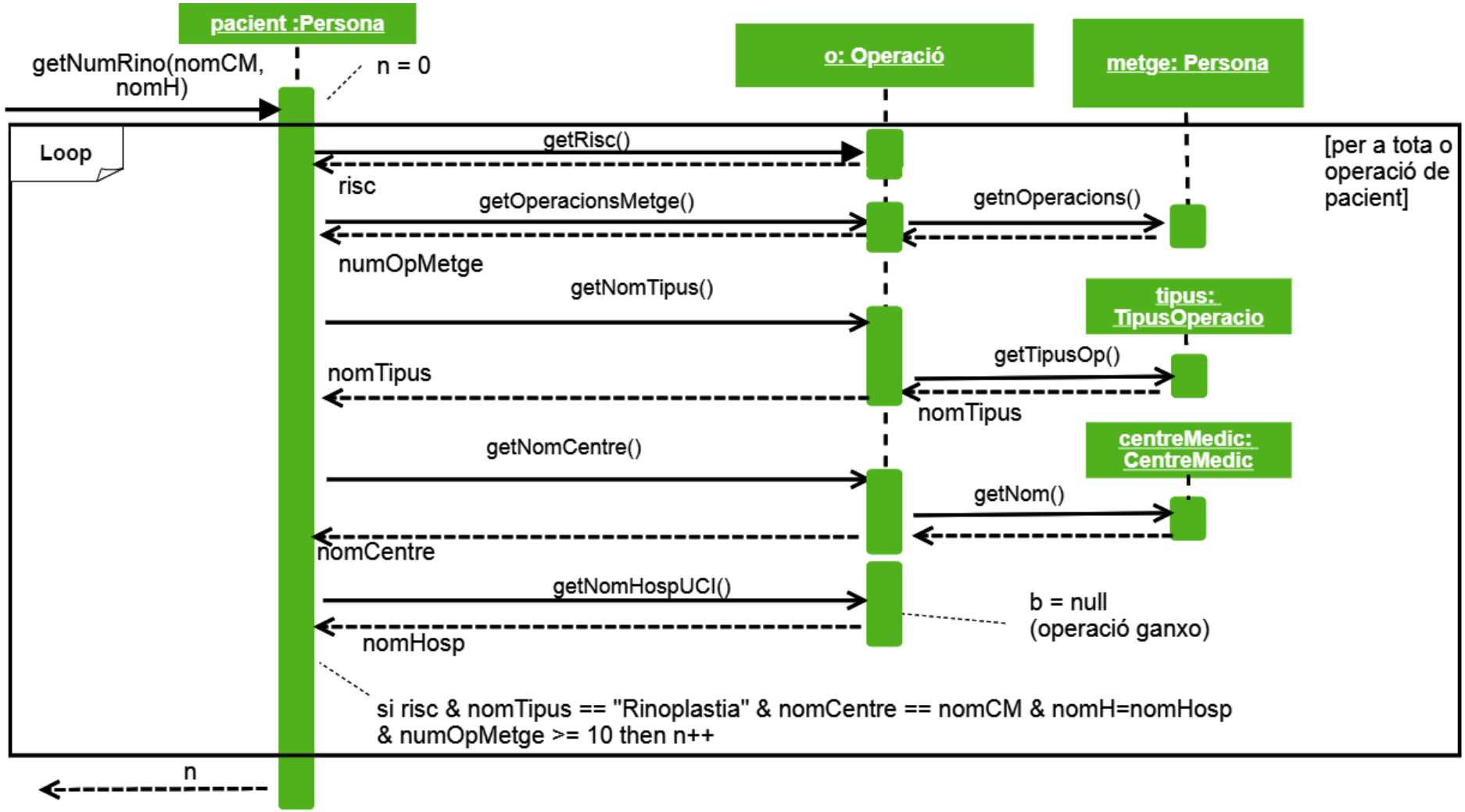
- Creem una classe que segueix el *Controlador Transacció* que conté
  - Un atribut per cada paràmetre de l'operació
  - Un atribut resultat del tipus retornat per la operación
  - Una operación *Executar*()
  - Una operació de *ObteResultat*()

Executar



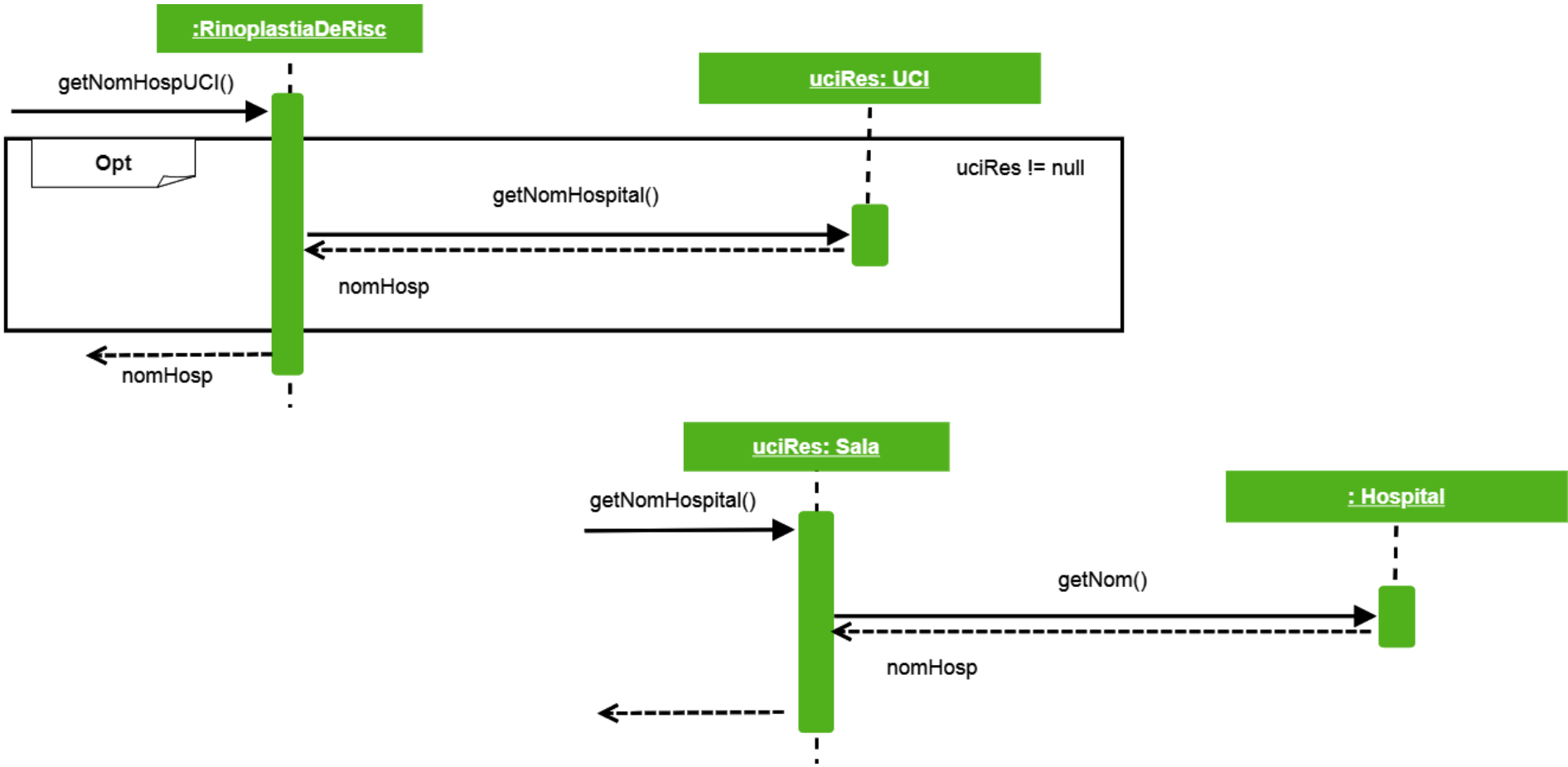
- Obtenim Persona, Centre Medic de la Capa de Dades.
  - Això pot llençar la Excepció [NoExisteixPersona], [NoExisteixCM], [NoExisteidHospital]
- Deleguem la resta de la operació a la classe Persona, al pacient
- Ens guardem el resultat

Persona::getNumRinoplasties



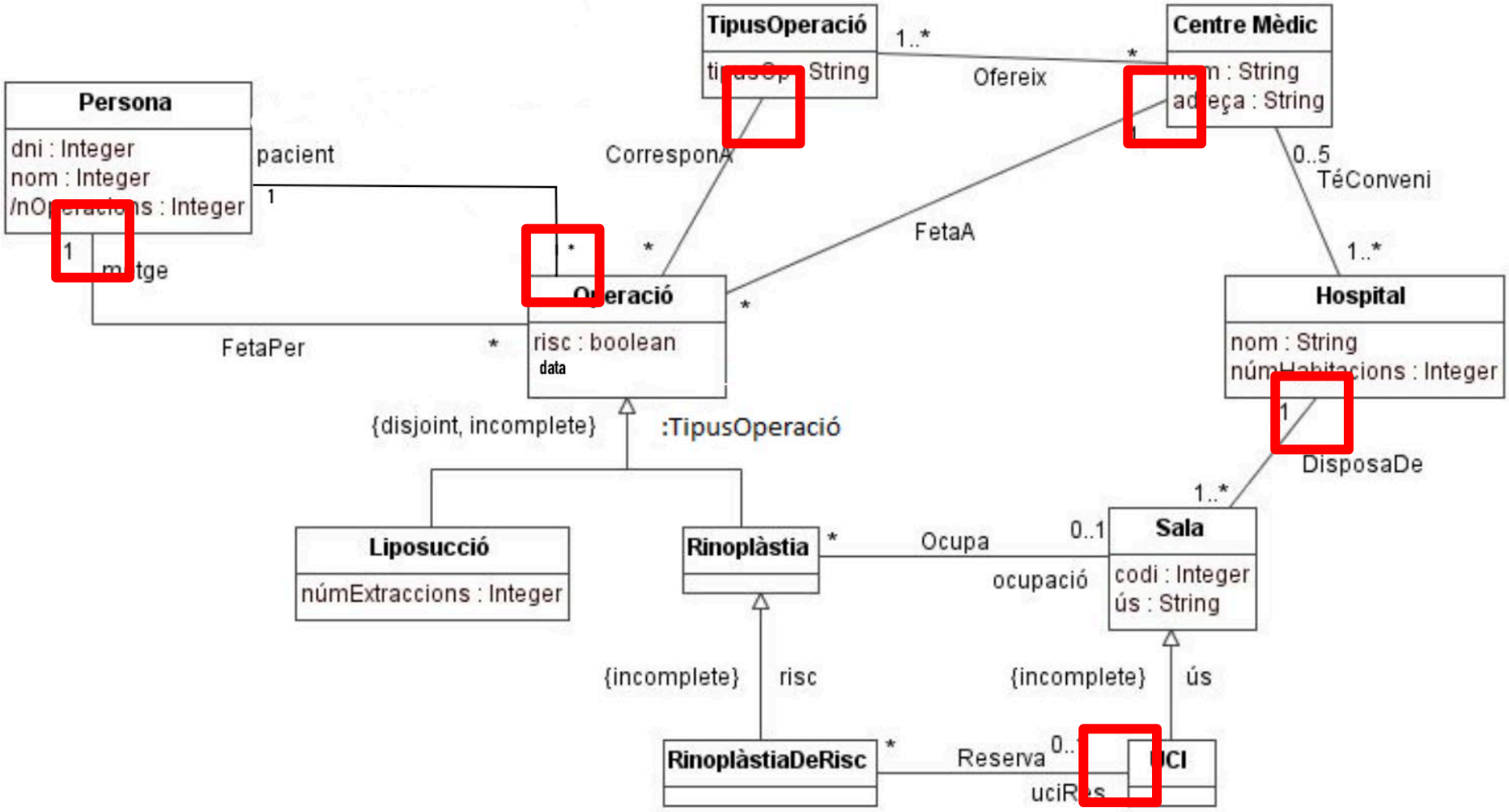
- Per cada operació del pacient, mirem si aquesta satisfà les condicions per ser afegida a la llista
- La operación reservaUCI() ha de ser implementada a les subclasses d'Operació

RinoplàstiadeRisc::getNomHospUCI



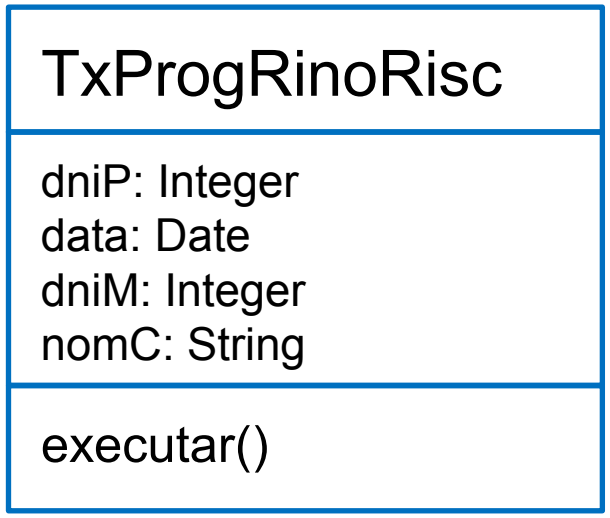
Noves Navegabilitats i Operacions

aquest sistema.



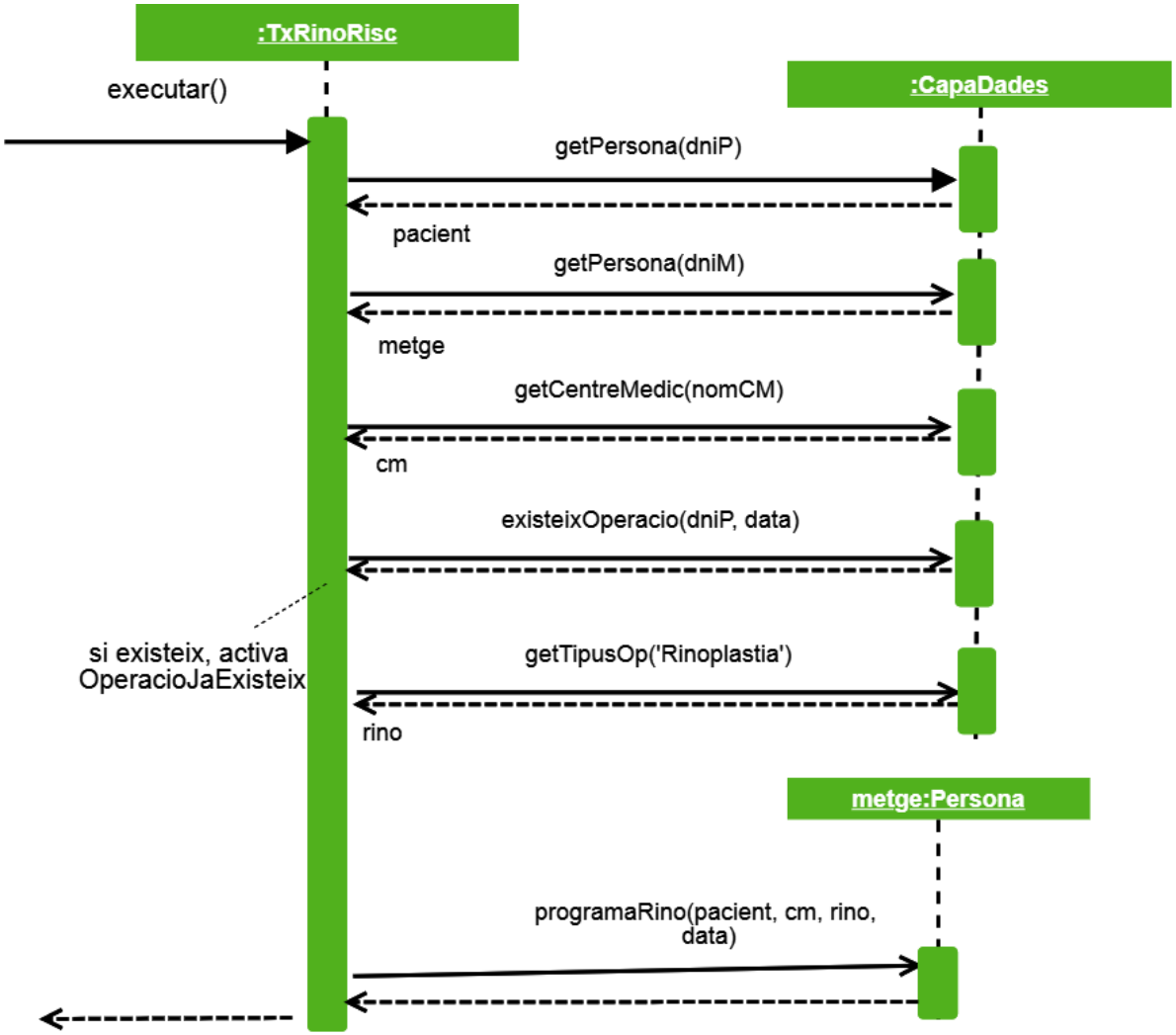
Solució programarRinoplastiaDeRisc

txProgRinoRisc



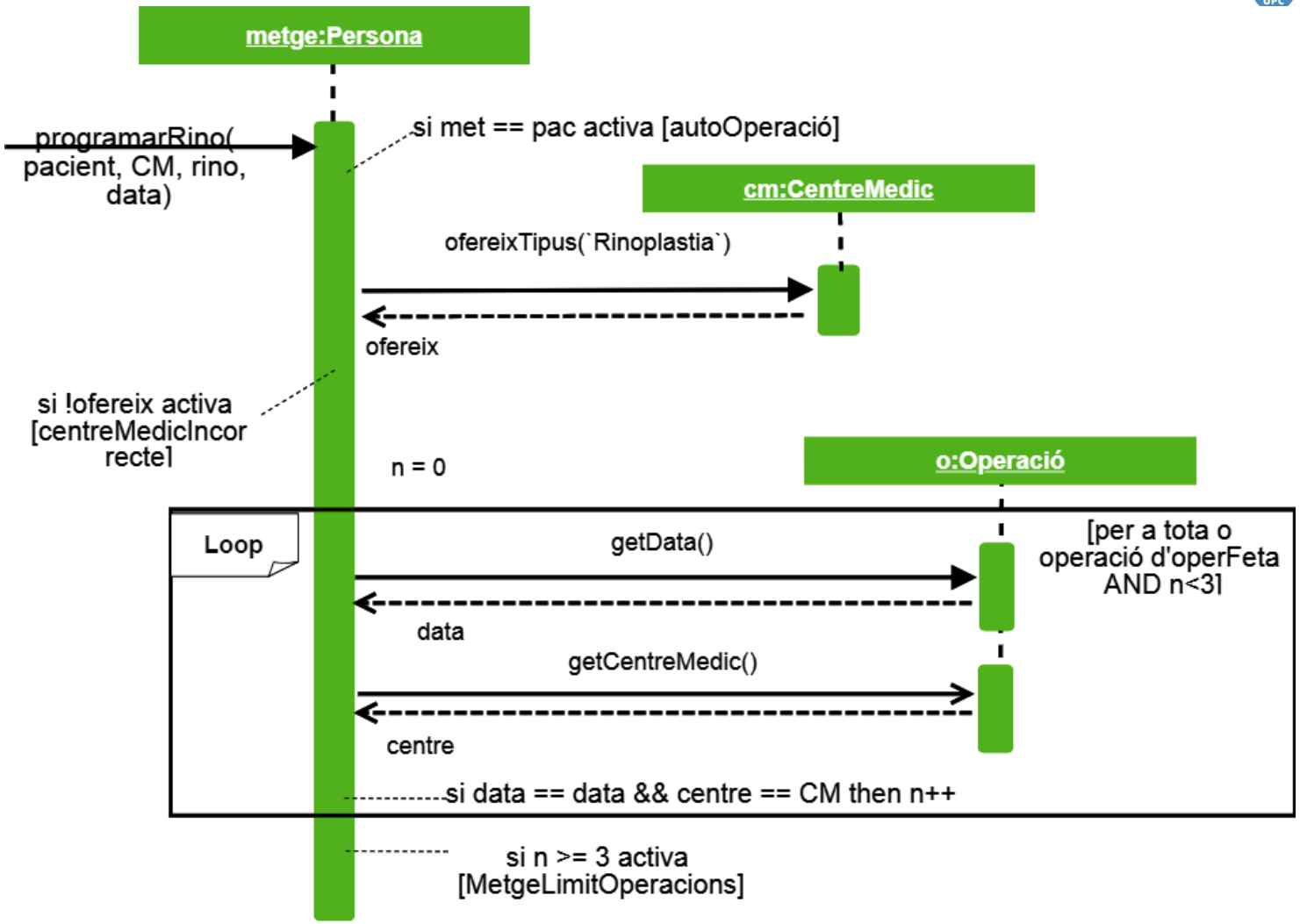
- Creem una clase que segueix el *Controlador Transacció* que conté
  - Un atribut per cada paràmetre de la operación
  - Una operación *Executar*()
  - Una operacióCrear
- En aquest cas no cal resultat perquè l'operació no retorna res

Executar



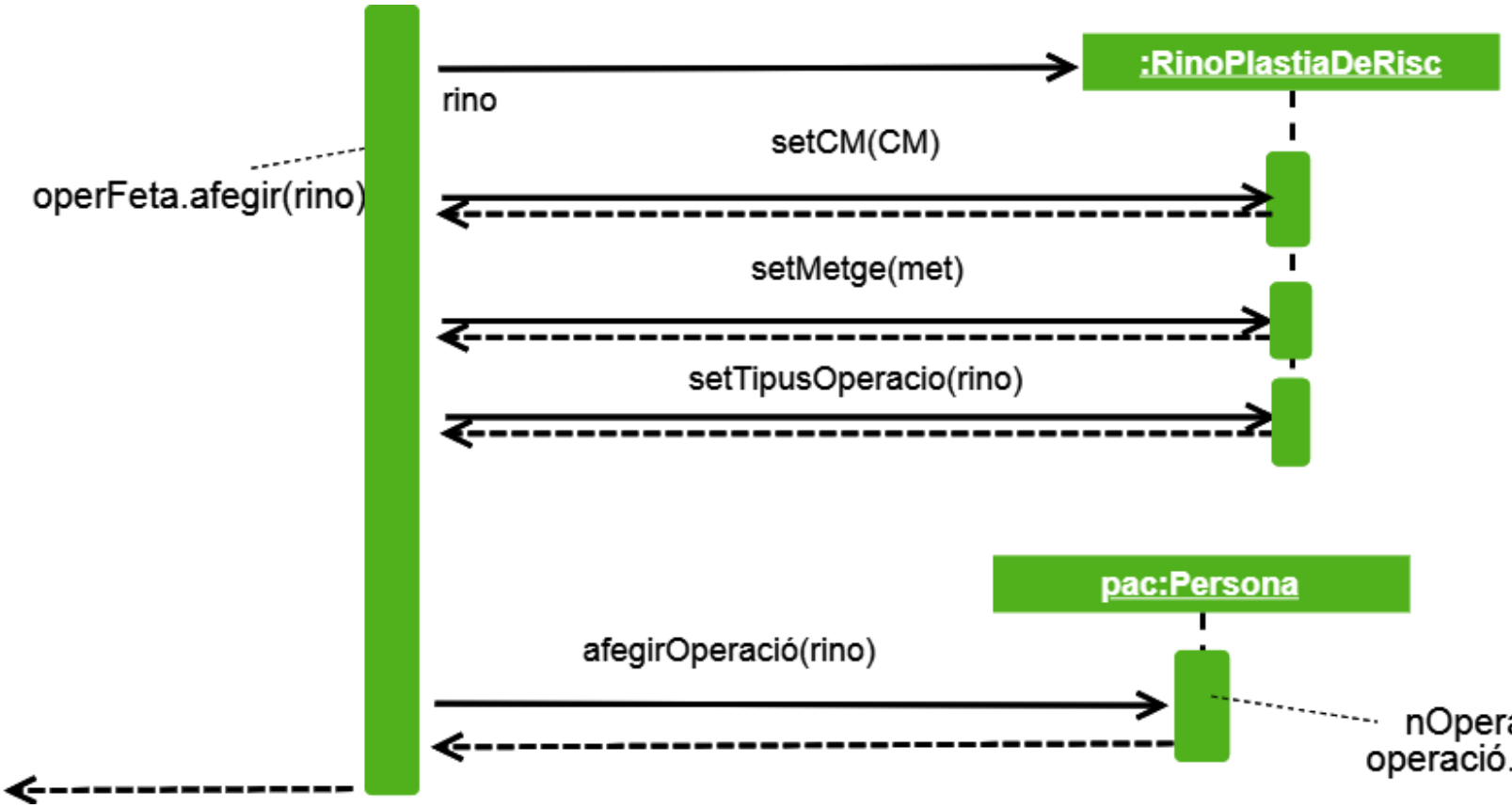
- Primer obtenim la informació, cosa que llençarà les excepcions:
  - [NoExisteixPersona]
  - [NoExisteixMetge]
  - [NoExisteixCM]
  - [JaExisteixOperacio]
- Finalment deleguem a metge:Persona

Persona::ProgramarRino (Part 1)



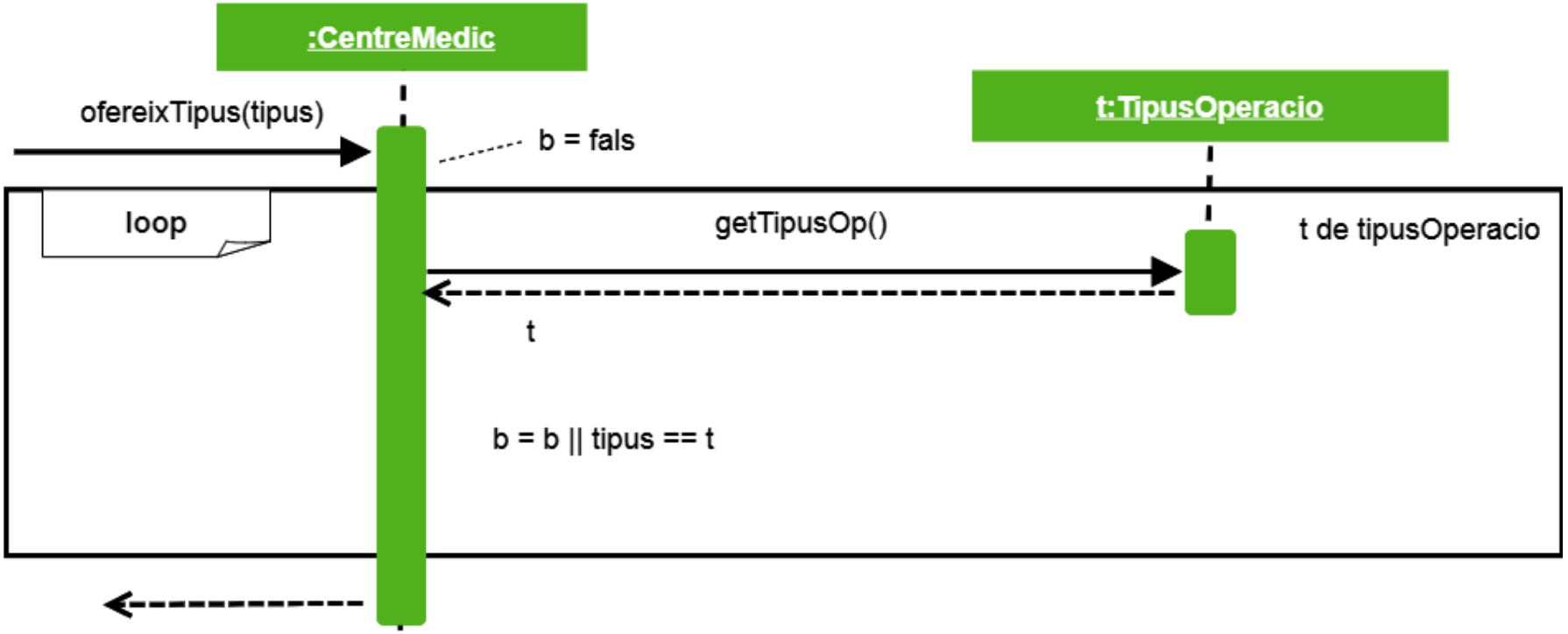
- Fem les comprovacions de: [autoOperacio], [CenreMedicIncorrecte], [MetgeLimitOps]
- Consultem tipus rino

Persona::ProgramarRino (Part 2)



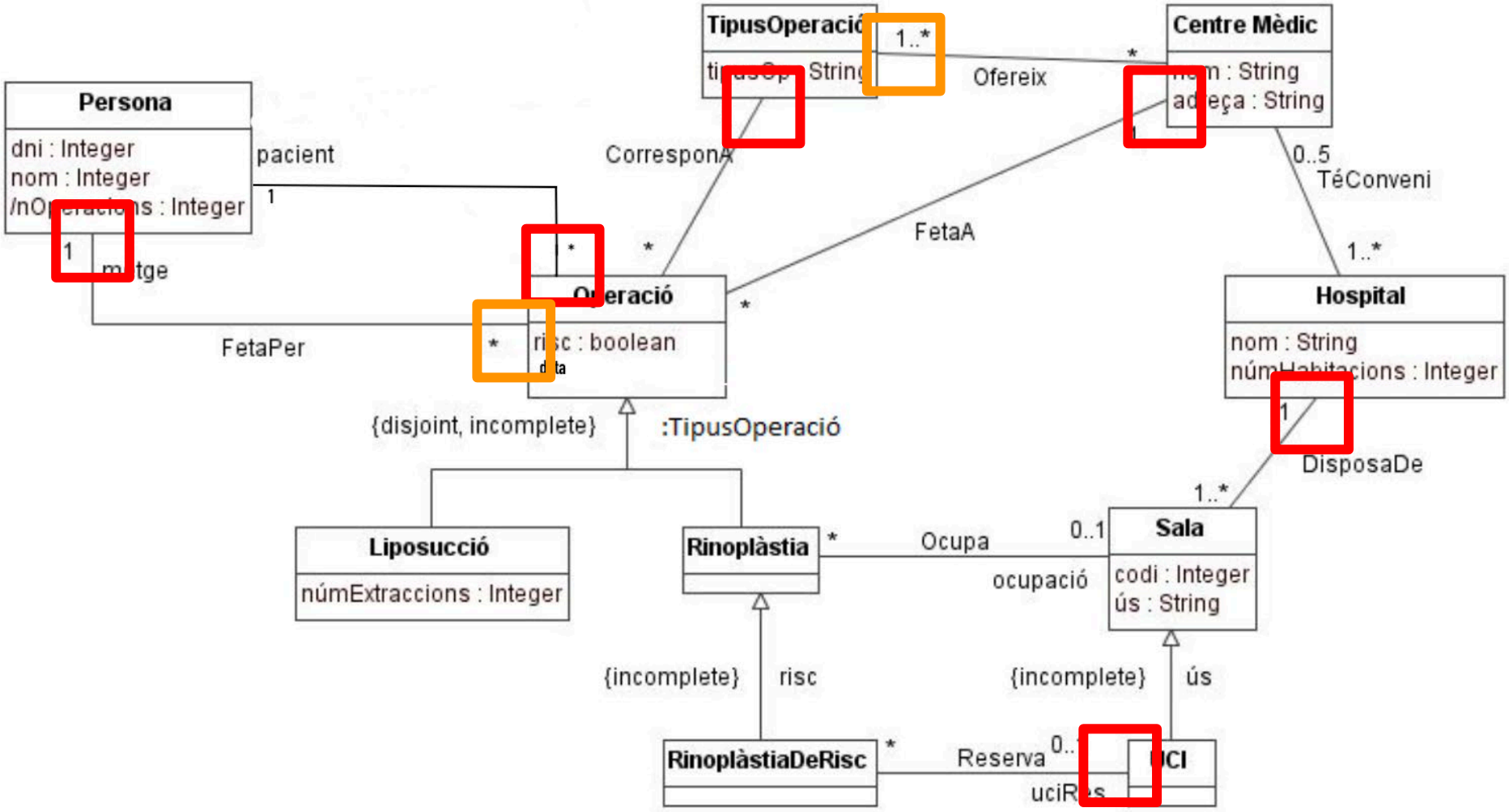
- Ara ja podem crear l'objecte nou
  - Assignem tots els objectes relacionats
  - Afegim l'operació a metge i a persona

CentreMedic::ofereixTipus



Noves Navegabilitats i Operacions

aquest sistema.



Problema de disseny en UML: Centres Mèdics