

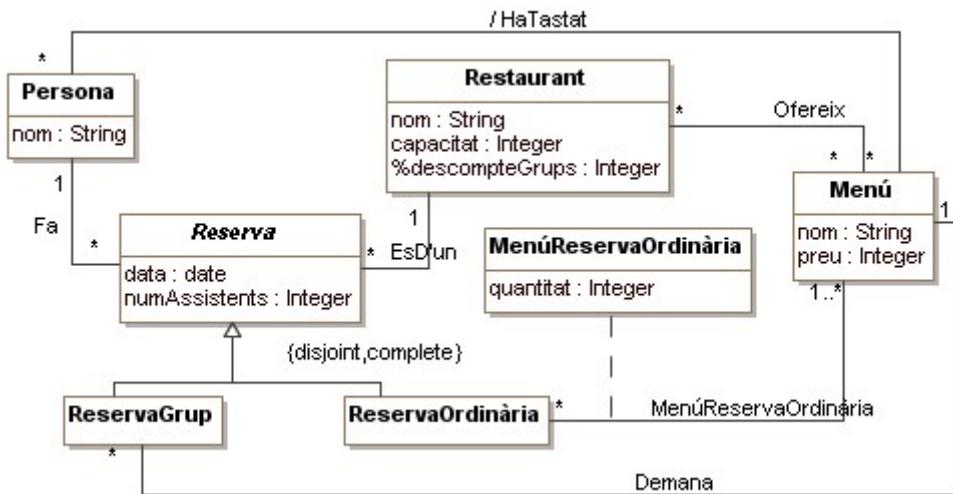
ARQUITECTURA DEL SOFTWARE

Unit 3.3: Presentation Layer Design

Exercicis Resolts

Exercici 3

Una cadena de restaurants que ofereixen menús per sopars ens ha demanat que li dissenyem una part d'un sistema software per gestionar les seves reserves. Els clients de la cadena fan reserves per un restaurant concret en una data determinada. Les reserves poden ser de dos tipus: les reserves de grup són reserves per a un conjunt de persones que demanen el mateix menú i ho fan en el moment de fer la reserva, i les reserves ordinàries són reserves per a una o més persones que demanen el menú que vulguin en el moment de sopar. Els restaurants ofereixen un descompte (%descompteGrups) en el preu del menú per a les reserves de grup. A continuació disposeu de l'esquema conceptual, de la descripció del disseny extern del cas d'ús *AltaReservaGrup*, del diagrama de seqüència dels esdeveniments del sistema del cas d'ús concret i dels contractes de la capa de domini corresponents:



R.I. Textuals:

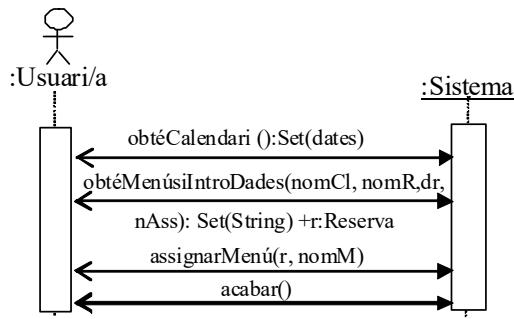
- Clau: (Persona, nom); (Restaurant, nom); (Menú, nom); (Reserva, nomPersona+data)
- Els menús de les reserves han de ser menús oferts pel restaurant on s'ha fet la reserva.
- La capacitat d'un restaurant ha de ser més gran que el sumatori dels assistents de les reserves d'aquell restaurant per una data determinada.
- Tots els atributs de tipus integer han de ser positius
- El numAssistents d'una reserva de grup ha de ser més gran que 5
- En una reserva ordinària el numAssistents ha de coincidir amb la quantitat de menús demandats

Informació derivada:

- HaTastat: relaciona les persones amb els menús que ha tastat o tastarà (per reserves de grup o ordinàries)

A continuació disposeu d'una descripció textual del disseny extern pel cas d'ús *AltaReservaGrup*:

1. El responsable de la central de reserves de la cadena de restaurants vol donar d'alta una nova reserva de grup. Selecciona l'opció corresponent i prem <<OK>>.
2. El sistema mostra una pantalla per introduir el nom del client, el nom del restaurant, el número de assistents i la data. La data a seleccionar es mostra en un calendari de dies a partir d'avui.
3. El responsable introduceix les dades anteriors i prem <<OK>>.
4. A continuació, el sistema enregistra la reserva de grup i mostra un desplegable amb tots els menús oferts pel restaurant.
5. El responsable selecciona un menú i prem <<OK>>.
6. El sistema enregistra que per a la reserva de grup creada es demana el menú seleccionat i informa al responsable. El responsable confirma la recepció del missatge. El cas d'ús acaba.
7. Si en alguna de les interaccions anteriors es produeix algun dels errors possibles, es mostra un missatge i acaba el cas d'ús.



context CapaDomini::obtéMenúsSiIntroDades (nomCl: String, nomR:String, dr:date, nAss:Integer): Set(String)

pre numAssOk: el nAss és més gran que 5 (RT)

exc clientNoExisteix: el client nomCl no existeix (R estructural- int referencial)

exc restaurantNoExisteix: el restaurant nomR no existeix (R estructural- int referencial)

exc restaurantSenseCapacitat: El restaurant no té capacitat per la reserva (RT)

exc reservaGrupExisteix: La reserva de grup ja existeix (RT Clau de reserva)

post reservaGrupCreada: s'ha creat la reserva de grup (r). (post original)

post obtéMenús: result= nom dels menús oferts pel restaurant nomR

post reservaGuardada: self.reserva:= r

context CapaDomini::assignarMenú (nomM: String)

pre menúServit: el menú nomM existeix

post assignaMenú: el menú s'assigna a la reserva

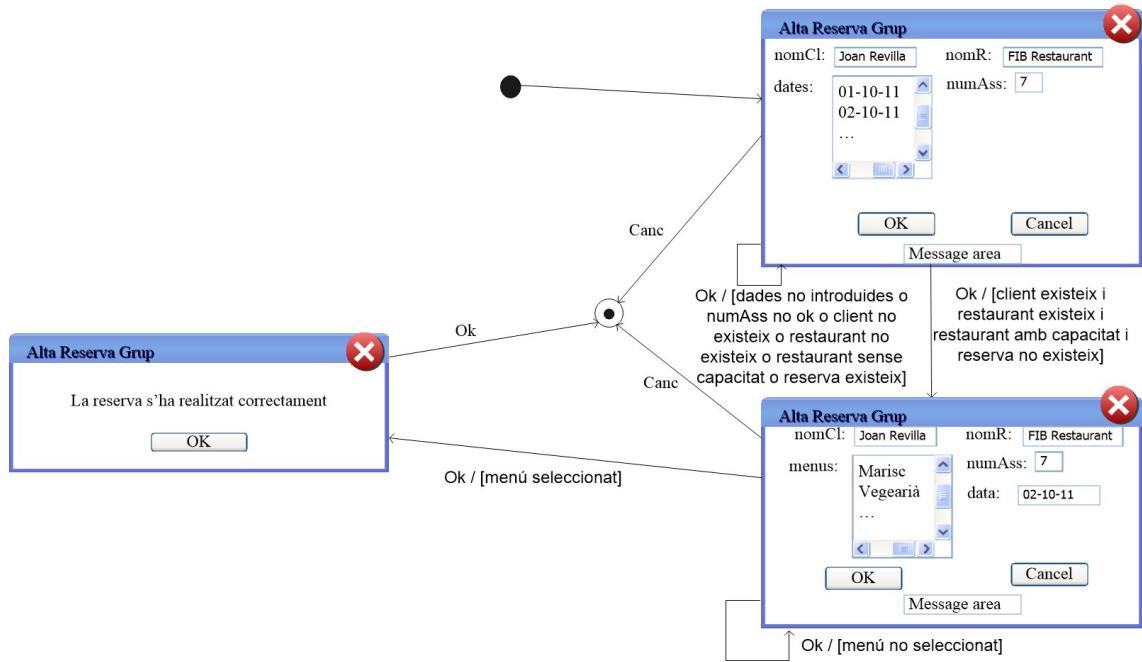
post haTastat: si la persona (de la reserva) no ha tastat el menú aleshores es dóna d'alta una instància de l'associació HaTastat entre la persona i el menú.

Es demana:

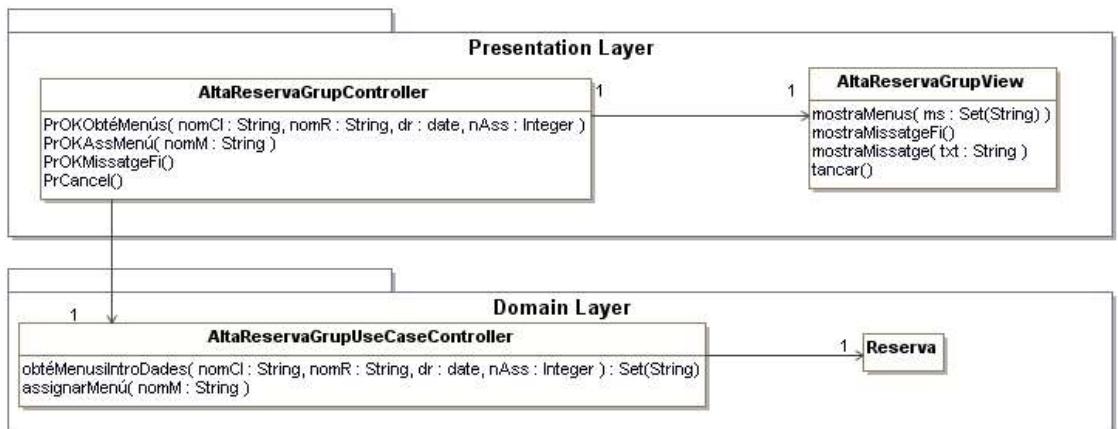
- Definiu el model navegacional de cas d'ús (descripció d'alt nivell).
- Diagrama de classes de la Capa de Presentació i de Domini (en aquesta capa només cal que definiu els controladors).
- Diagrames de seqüència de les operacions dels controladors de la Capa de Presentació.

Solució

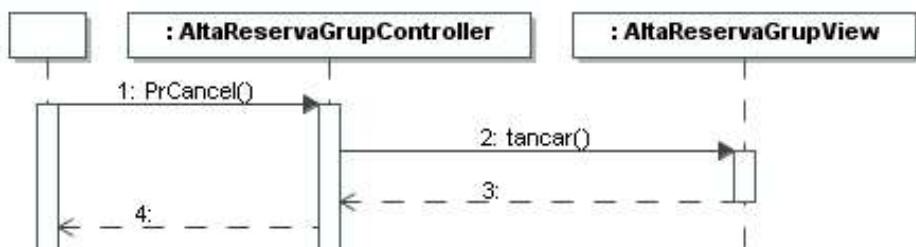
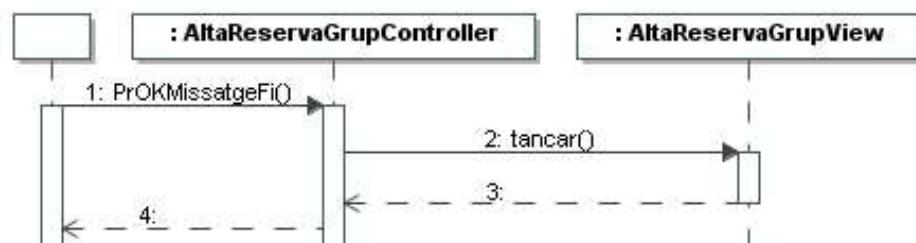
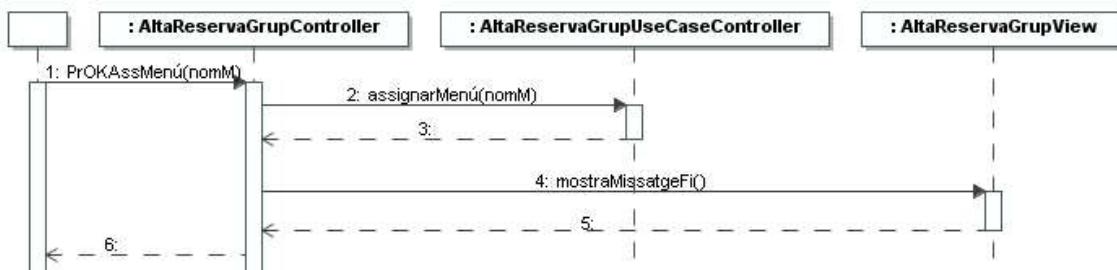
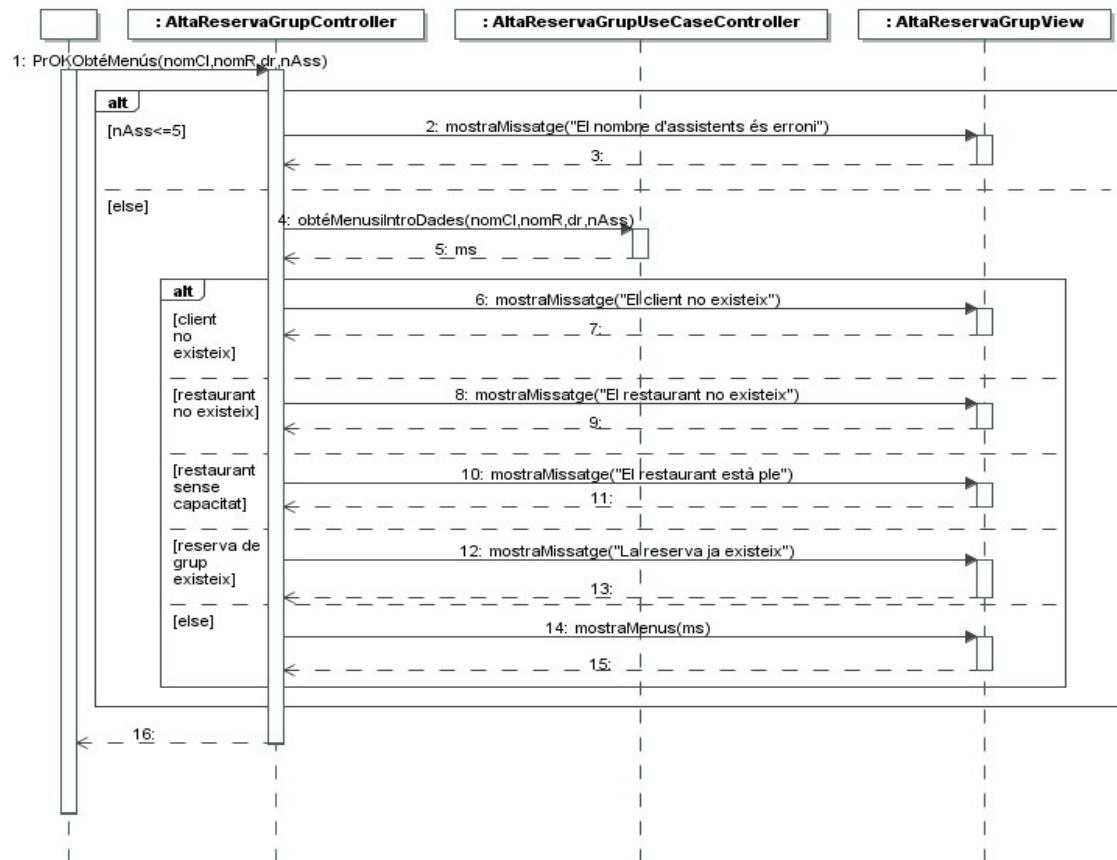
a) Model navegacional



b) Diagrama de classes

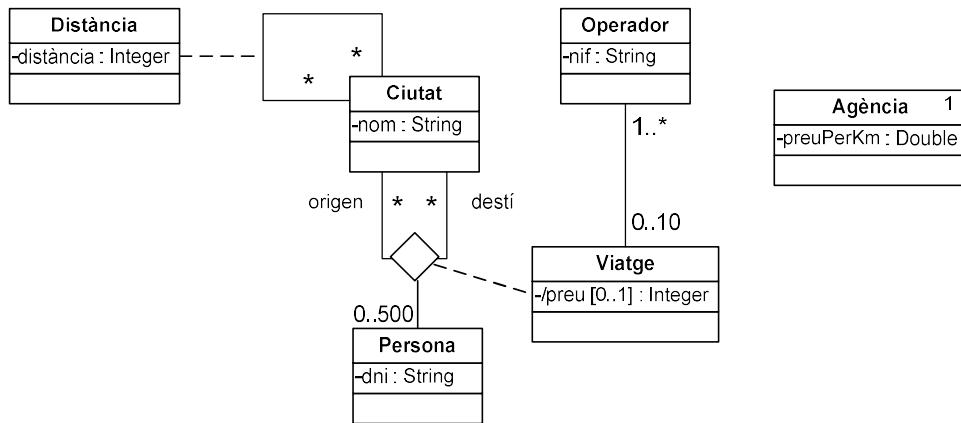


c) Diagrames de seqüència



Exercici 4

Una Agència de viatges organitza Viatges per a Persones d'una Ciutat origen a una Ciutat destí. Els viatges tenen un preu calculat en funció de la distància entre les ciutats i el preuPerKm fixat per l'agència. Cada viatge té assignats un o més Operadors potencials. A continuació disposeu de l'esquema conceptual, de la descripció del disseny extern del cas d'ús *AltaViatge*, del diagrama de seqüència dels esdeveniments del sistema del cas d'ús concret i dels contractes corresponents:

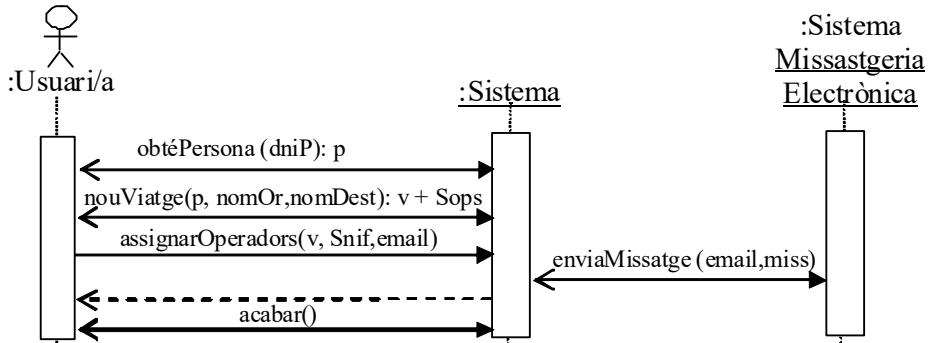


- RT1. Claus: (*Ciutat, nom*); (*Persona, dni*); (*Operador, nif*)
- RT2. Les ciutats *origen* i *destí* d'un viatge són diferents
- RT3. Una persona no pot tenir més de 100 viatges
...altres restriccions que no són d'interès per al problema

ID1. El *preu* d'un viatge és igual a la *distància* establerta entre la seva ciutat origen i destí, multiplicada pel *preuPerKm* estipulat per l'agència. Si les dues ciutats no estan directament relacionades per *Distància*, l'atribut *preu* té valor nul.

Tot seguit disposeu de la descripció textual del cas d'ús concret:

1. L'usuari introduceix el dni de la persona i prem <<OK>>.
2. A continuació, l'usuari introduceix les dues ciutats i prem <<OK>>.
3. El sistema enregistra el viatge i mostra en una llista tots els operadors amb menys de 10 viatges assignats que, per tant, poden ser assignats com a operadors potencials a aquest nou viatge.
4. Finalment, l'usuari marca en aquesta llista els operadors potencials d'aquest viatge que desitja, introduceix el mail del client i prem <<OK>>.
5. El sistema enregistra els operadors del viatge, informa a l'usuari i invoca al sistema extern de missatgeria electrònica per enviar un mail amb les dades del viatge al client.
6. L'usuari confirma la recepció del missatge i s'acaba el cas d'ús.
7. Si es produeix algun dels errors possibles, es mostra un missatge i acaba el cas d'ús.



context CapaDomini::obtéPersona (dniP:String)
exc personaNoExisteix: la persona amb dni *dniP* existeix.
post self.persona = persona amb dni *dniP*.

context CapaDomini::nouViatge (nomOr: String, nomDest: String): Sops:Set(string)
pre ciutatsDiferents: les ciutats origen i destí són diferents. (RT1)
exc ciutatOrigenNoExisteix: la ciutat amb nom *nomOr* no existeix. (R. referencial)
exc ciutatDestiNoExisteix: la ciutat amb nom *nomDest* no existeix. (R. referencial)
exc viatgeExisteix: el viatge per la persona, ciutat origen i destí existeix. (R clau viatge)
exc personaAmbMoltsViatges: la persona ja té 100 viatges. (RT2)
exc ciutatsAmbMoltsViatges: les ciutats origen i destí ja tenen 500 viatges. (R gràfica)
post viatgeCreat: crea una instància de viatge de *p* entre *nomOr* i *nomDest* i s'assigna el preu.
post self.viatge = viatge creat.
post result = nif dels operadors que tenen assignats menys de 10 viatges.

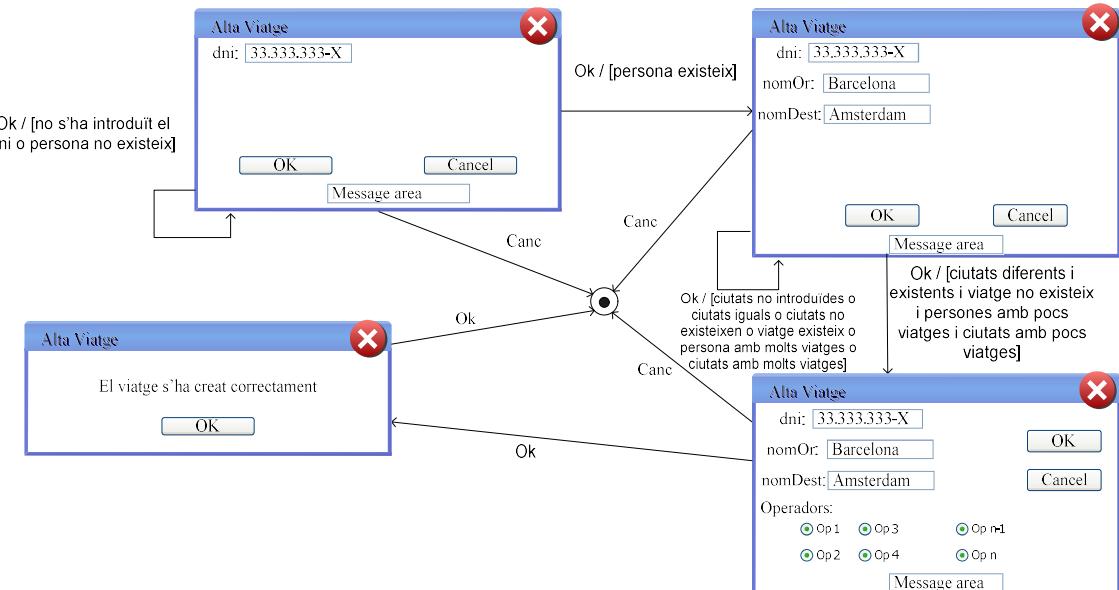
context CapaDomini::assignarOperadors (Snif: Set(String), email:String)
pre operadorsExisteixen: els operadors de Snif existeixen. (R. referencial)
pre operadorsSenseMaxViatges: els operadors de Snif no tenen 10 viatges assignats. (R. gràfica)
pre operadorsNoRepetits: si hi ha operadors a Snif, no hi ha repetits. (R. implícita no repetits associacions)
pre viatgeAmbOperadors: hi ha operadors a Snif. (R gràfica)
post operadorsAssociats: per tot nif *n* dins de *Snif*, associa el viatge *v* amb l'operador de nif *n*.
post enviaMail: s'invoca l'execució de l'operació *enviaMissatge* del sistema de *MissatgeriaElectrònica* amb l'email i el missatge de text amb les dades del viatge.

Es demana:

- (a) Definiu el model navegacional de cas d'ús (descripció d'alt nivell).
- (b) Diagrama de classes de la Capa de Presentació i de Domini (en aquesta capa només cal que definiu els controladors).
- (c) Diagrames de seqüència de les operacions dels controladors de la Capa de Presentació.

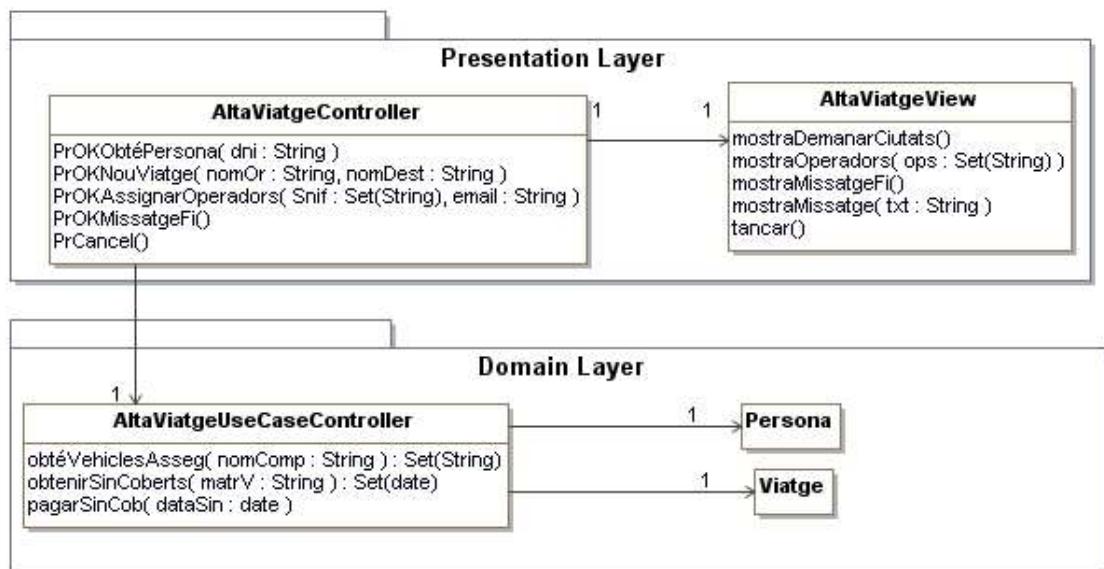
Solució

a) Model navegacional



El botó ok estarà desactivat fins que es seleccione un operador

b) Diagrama de classes



c) Diagrames de seqüència

