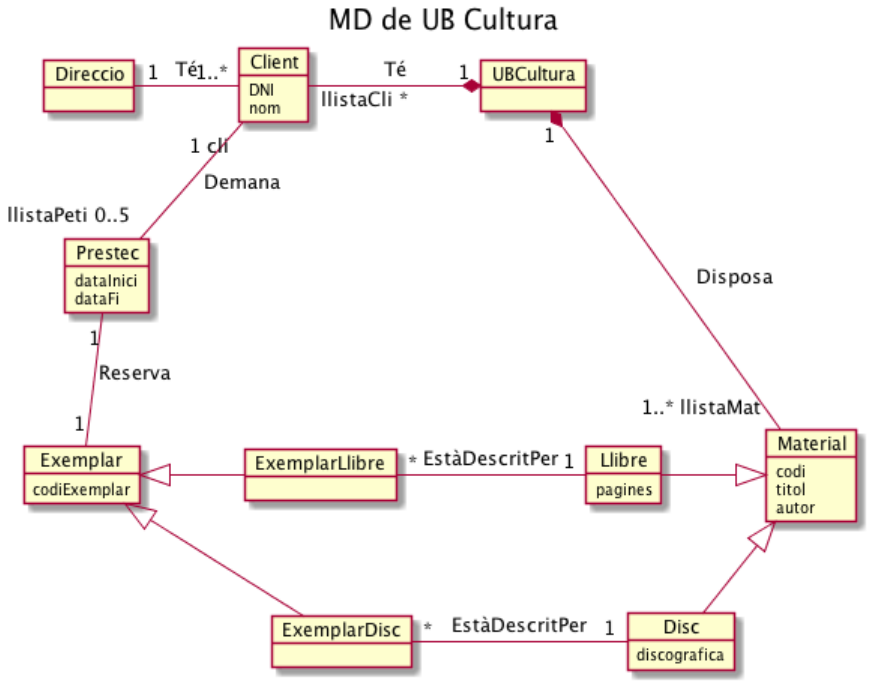
Disseny de Software: Patrons GRASP (exemple UBCultura)
Curs 2021-22

UB-Cultura: Com fer el disseny aplicant patrons GRASP? Cal donar les responsabilitats a les classes que els hi toqui. Per això s'apliquen els patrons Creador, Expert i Controlador. En aquest exemple es mostra com es decideix a cada pas del disseny i com va creixen el Model de Classes del Domini.

En el problema d'UB-Cultura que es va especificar en el Tema 2:



Suposa que cal implementar aquestes funcionalitats, seguint els patrons GRASP

1 - cercar Client per DNI
2 - afegir client (dni, nom, direccio)
3 – Afegir préstec d'un material a un client (dni, idMaterial, data_inici, data_final)

NOTA : Aquesta solució està explicada pas a pas. Per a la pràctica 2, només us demanem les taules on es resumeixen els patrons usats a cada història d'usuari, NO heu de detallar incrementalment el Model de Domini ni el diagrama de seqüència que aquí es detallen

En primer lloc cal destacar que s'estructura el projecte amb el patró arquitectònic Model-Vista-Controlador. Les funcionalitats cercar client per DNI i afegir client es serviran des del controlador a la vista. el Controlador és l'encarregat de cridar a classes del model per a delegar la feina.

Per a determinar el Diagrama de Classes final del Controlador i el Model, es tria cada funcionalitat per veure com es dissenyarà al llarg de les classes usant el patró expert (GRASP). Es important aquí veure com es van delegant les responsabilitats per les diferents classes.

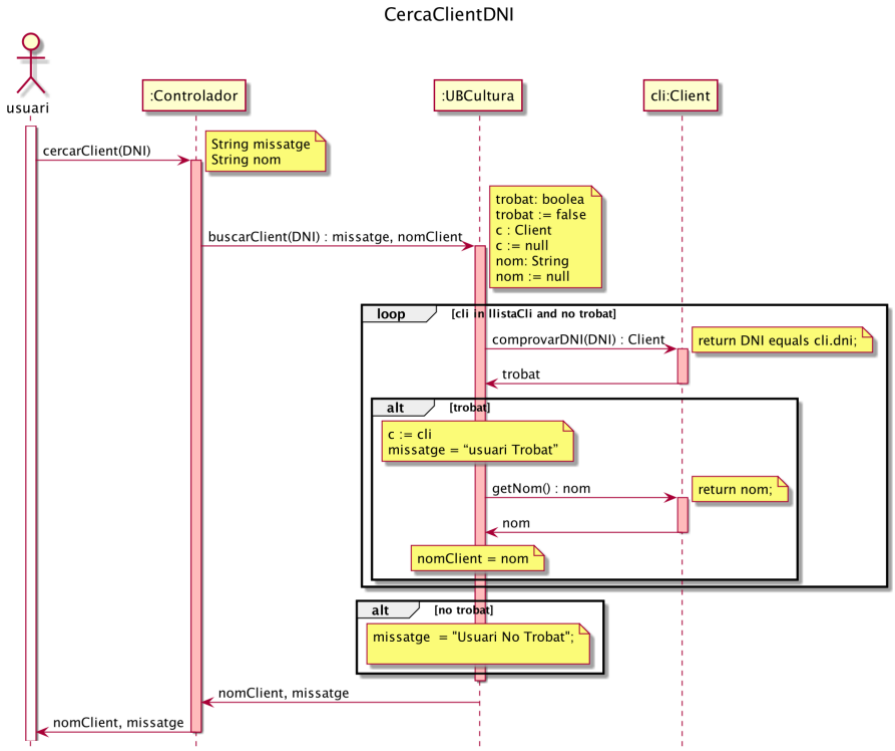
1. **Cercar Client Per DNI.**

Pas 1. Es defineix el Controlador que servirà CercarClientPerDNI a la Vista. En principi, es fa una classe Controlador, per servir aquestes peticions tot i que sempre s'ha de vigilar que no es vulneri el principi de Single Responsability (SOLID), o que es vulneri el menys possible, tot aplicant el patró **Expert** de GRASP. En cas que el Controlador quedés poc cohesionat i amb alt acoblament amb d'altres classes, caldria dividir-lo en diferents Controladors especialitzats.

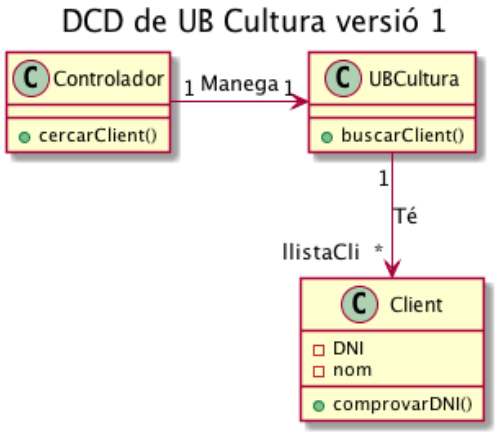
Pas 2. Quan ens plantegem aquesta utilitat, ens preguntem primer qui es la classe Experta en la informació de clients. En aquest cas la resposta és UBCultura, ja que té la llista de Clients i en tot cas serà Client la classe que podrà garantir si la informació del DNI es igual a un altre o no.

Pas 3. Ara cal que reparteixis les responsabilitats per les diferents classes del teu model de domini per a començar a fer el diagrama de classes. Això ho pots fer ajudant-te del Diagrama de Seqüència o analitzant les classes que estaran implicades del teu Model de Domini. en aquest cas seran les classes Controlador, UBCultura i Client.

Un cop les has repartit (tal com indica el Diagrama de Seqüència que hi ha a continuació), dibuixem el Diagrama de Classes que hem obtingut. Fixa't que el diagrama de classes de disseny té fletxes a les associacions que indiquen la navegabilitat entre classes i a cada classe es posen tant els atributs com els mètodes que calen per implementar la funcionalitat desitjada.



El Diagrama de Classes de Disseny, afegint la navegabilitat entre classes, és:

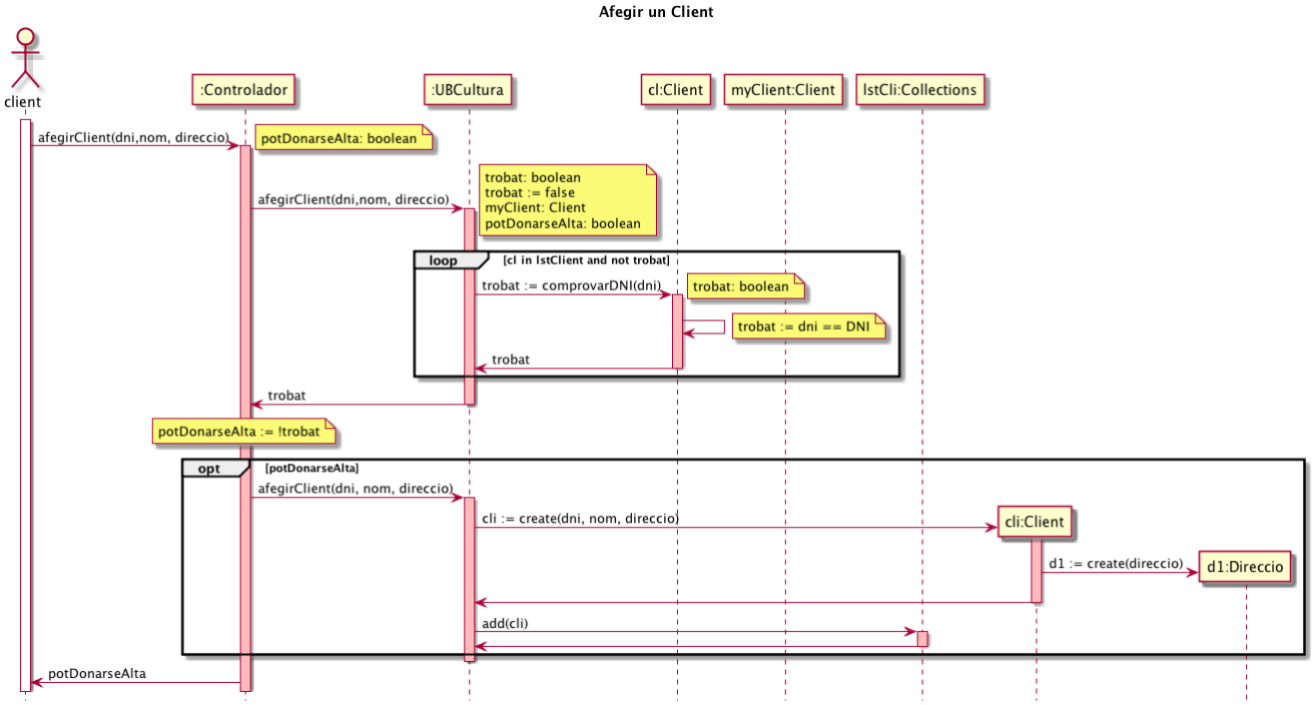


Test d'acceptació	Criteri GRASP	Breu explicació de les classes canviades
cercarClientPerDNI	Controlador	Ofereix la utilitat de cercarClient a la Vista (o test)
	Expert	Repartiment de buscarClient UBCultura i comprovarDNI(a la classe Client per que es l'experta en la informació de DNI

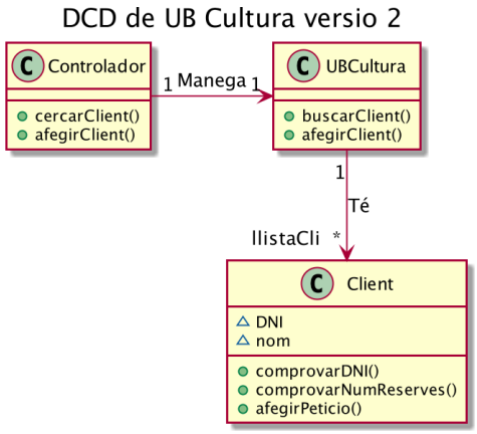
	Baix Acoblament	El Controlador només coneix UBCultura i no la classe Client, per tant es manté el baix acoblament de Controlador amb la resta de classes.

2. Afegir Client (dni, nom, direcció) retorna cert si s'ha pogut afegir per què és un client nou o fals si el client ja existeix.

Sobre el Diagrama de Classes anterior s'han d'afegir noves classes. Ara es torna a avaluar qui té les responsabilitat de veure si ja existeix el client (patró EXPERT) i quina classe crea el nou client (patró CREADOR), tot repartint la feina des de la classe Controlador (patró CONTROLADOR).



On el Diagrama de Classes de Domini quedarà amb una classe més, Direcció i haurem afegit més mètodes a les classes ja existents.



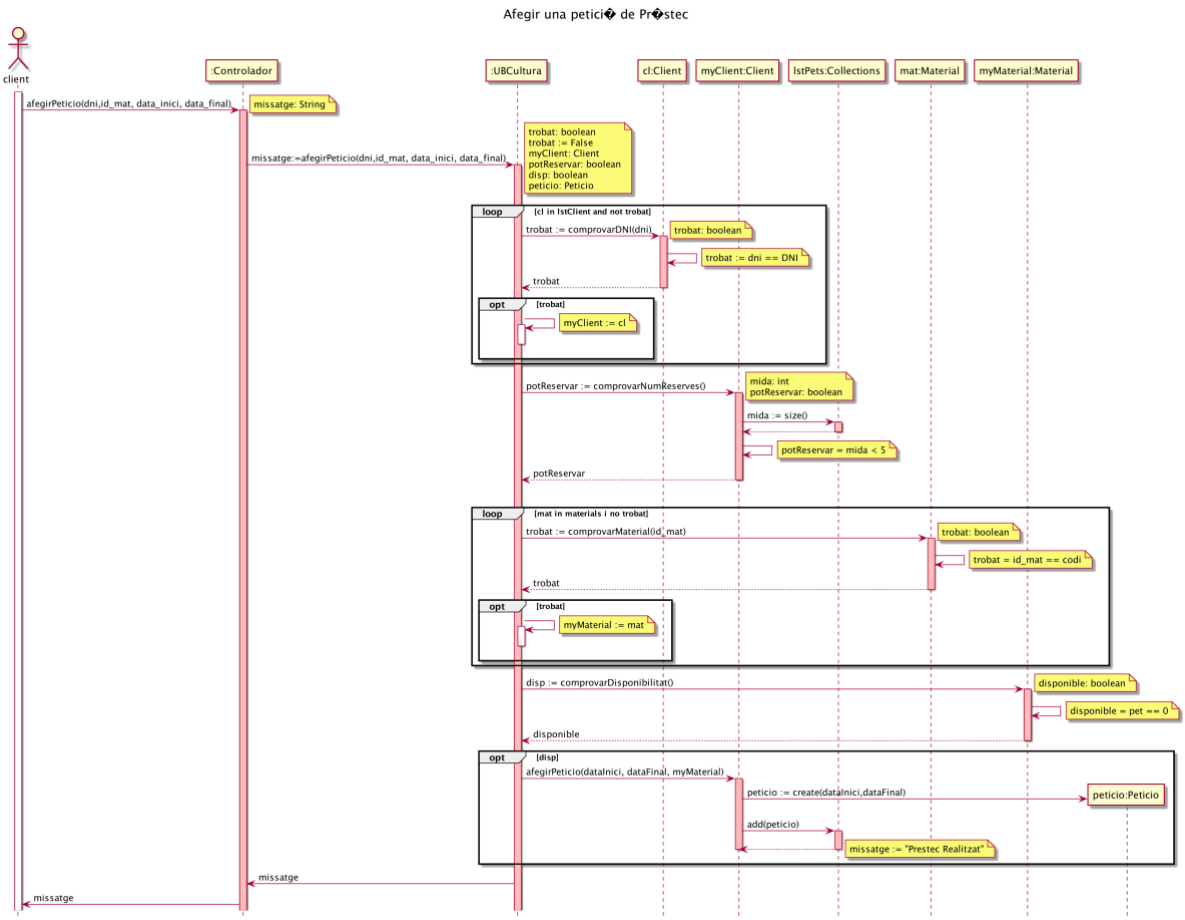
Fixa't que aquí el Controlador comença a baixar la seva cohesió ja que serveix més utilitats a la Vista, tot i que com encara tracta només amb Clients es podria pensar que encara té una única responsabilitat (segons SOLID)

Test d'acceptació	Criteri GRASP	Breu explicació de les classes canviades
cercarClientPerDNI	Controlador	Ofereix la utilitat de afegir Clients a la Vista (o test)
	Expert	Repartiment de buscarClient UBCultura i comprovarDNI(a la classe Client per que es l'experta en la informació de DNI)
	Creador	afegirClient: te la responsabilitat de la creació UBCultura ja que conté la llista de clients (el new de Client es farà a la classe UBCultura) i així Controlador no coneixerà el tipus Client
	Baix Acoblament	El Controlador només coneix UBCultura i no la classe Client, per tant es manté el baix acoblament de Controlador amb la resta de classes.

3. Si es volgués afegir una petició de préstec, quines crides necessitaries? Quin diagrama de classes et quedaria? Raona tal i com s'ha fet fins ara en els dos apartats anteriors.

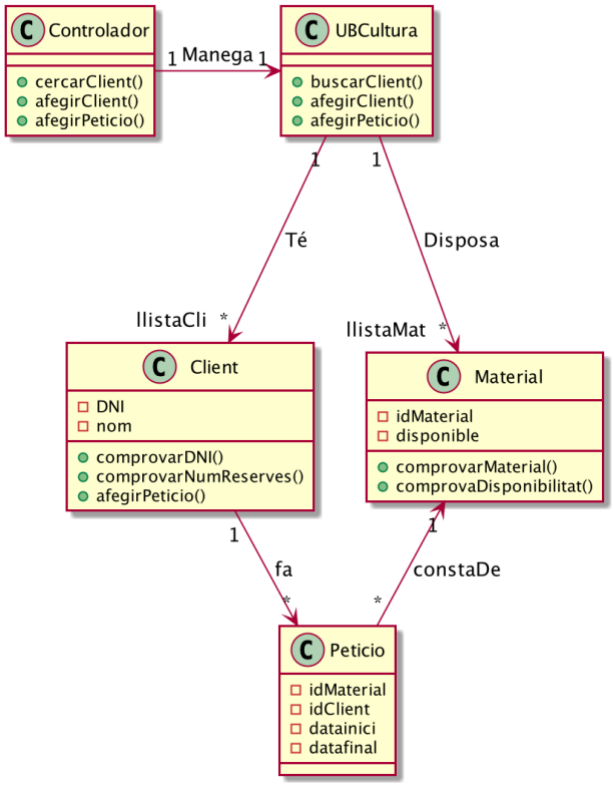
afegirPrestec(dni, id_material, data_inici, data_final) retorna CERT si tot ha anat bé o FALS en el cas que no es puguin afegir més préstecs en aquestes dates pel client.

Aquest és un possible Diagrama de Seqüència:



Com seria el seu Diagrama de Classes amb totes les funcionalitats anteriors?

DCD de UB Cultura versio 3



Test d'acceptació	Criteri GRASP	Breu explicació de les classes canviades
cercarClientPerDNI	Controlador	Ofereix la utilitat de afegirPeticio a la Vista (o test)
	Expert	Repartiment de afegirPeticio a UBCultura i Client (a la classe Client per que es l'experta en la informació de préstec)
	Creador	afegirPrestec: te la responsabilitat de la creació Client ja que conté la llista dels préstecs que té (el new de Prestec es farà a la classe Client) i així Controlador o UBCultura no coneixeran el tipus Prestec
	Baix Acoblament	El Controlador només coneix UBCultura i no la classe Client ni Material ni Prestex, per tant es manté el baix acoblament de Controlador amb la resta de classes.
	Baixa Cohesio	El Controlador comença a tenir baixa cohesió en servir moltes responsabilitats a la Vista. Per tant aquí podríem pensar en realitzar una Façana (amb una Indireccio o Fabricacio Pura)....

(4) Ara es volen també tenir vendes i no només préstecs, com canviaria això el teu diagrama de classes? Quins patrons aplicaries?