

Solució del cas pràctic.

Generalitats: l'enunciat (que ja té alguns anys) demana el desenvolupament d'un bon client de correu. Avui en dia podríem cobrir de sobres el que es demana amb qualsevol client de correu com Outlook o Outlook Express de Microsoft, el mòdul de correu de Mozilla (abans Netscape), etc. per tant no caldria fer-lo, sinó triar quin del existents en el mercat s'adapta millor al problema plantejat.

La finalitat d'aquest enunciat es veure com aplicar la metodologia que us hem proposat a la producció d'una especificació, per tant, no donarem el problema per resolt triant un producte del mercat, sinó que mirarem d'especificar-ne un seguint els passos de la metodologia. Amb el temps disponible per discutir-lo ens haurem de limitar a fer una primera passada sense poder arribar a l'últim detall de l'especificació.

RNF:

- Interfície usuari i factors humans: dóna poques pistes, sembla que ho faci quan diu "[...] en un entorn universitari [...]" però és en un exemple i no fa referència al context d'execució. Sembla que el programa hagi d'estar orientat a un únic usuari per ajudar-lo gestionar millor el correu sense haver-li de demanar gaires coneixement addicionals.
- Documentació: no se'n parla.
- Consideracions de hardware: no n'hi ha cap. Malgrat tot es podria fer alguna inferència, però no prou significativa, ja que diu "L'accés hauria de poder ser per pantalla, però també imprimint-los" (a més, aquest "però també" fa pensar en un cert penediment).
- Característiques de rendiment: tampoc n'hi ha.
- Tractament d'errors i condicions extremes: no hi cap referència.
- ...

De fet, no hi cap referència a cap dels apartats que segueixen a la llista del l'Allen i l'Horacio i tampoc a l'entorn on s'ha d'executar, per tant, suposarem vàlids els RNF típics de l'assignatura (no els detallarem aquí ja que són part del vostre treball).

Llista de frases **nominals** (classes i atributs):

- sistema de bases de dades documentals senzill
- emmagatzemament de correu electrònic || sistema de correu electrònic
- carta || documents || correspondència || missatges
- "garbage mail"
- persona que envia || emissor
- text
- encapçalament
- usuari
- entorn universitari
- congressos
- cursos
- correspondència acumulada
- tipus de cartes

- format (a l'emmagatzemar)
- sistema de prioritats entre tipus de documents o entre documents del mateix tipus.
- contenidors
- criteris (d'accés)
- formats de visualització
- índex (dels contenidors)
- resum (dels missatges)

Llista d'expressions **verbals** (funcionalitats)

- construir un sistema de bases de dades documentals.
- rebre carta (qualsevol persona ens pot enviar una carta) || mirar el correu rebut || llegir els missatges
- llegir primer els missatges més urgents || classificar automàticament els documents
- respondre (els missatges)
- convocar reunions
- definir els diferents tipus de cartes que vol emmagatzemar
- accedir-hi per diferents criteris
- visualització

Hi ha elements que són fruit d'un determinat soroll en el text, com els de l'exemple d'acumulació de correu "entorn universitari", "congressos", "cursos", o la funcionalitat de "convocar reunions" que sense una agenda associada no és altre cosa que una tramesa de missatges.

El tema del format d'emmagatzematge és un dels que no tenen gaire sentit en els sistemes de correu d'avui en dia. Cada sistema de correu sol haver definit el seu format d'emmagatzematge del missatge complet i deixa jugar a l'usuari amb la forma de veure'ls, és a dir, el deixa gestionar-se les vistes.

De fet, el correu circula per *internet* en un únic format (que es coneix per les sigles SMTP) basat en el codi ASCII de 7 bits i s'usa una conversió de 8 a 7 bits (MIME) per adjuntar fitxers en altres formats. Per exemple, un text en ASCII estès caldrà convertir-lo a un fitxers de caràcters de 7 bits. Per entendre's amb SMTP (i MIME) aquests sistemes disposen del convertidor (*gateway*) adequat. (Les relacions entre el client de correu i el servidor es gestionen amb algun protocol estàndard, com MAPI, POP3 o IMAP4.) Per tant, obviarem aquest tema dels formats d'emmagatzematge. El dels formats de visualització no, i en tot cas en el moment de definir les vistes de consulta del correu ens ocuparíem de dotar-les amb la flexibilitat adequada.

Observació: una base de dades documental (BDD) no és una cosa senzilla. Es tracta d'una eina destinada a emmagatzemar i tractar informació sense estructurar, com articles de revistes o de premsa, bibliografia, etc. i poder-hi fer recerques complexes. Un sistema de gestió de base de dades documental (SGBDD) sol ser una eina complexa, allunyada del que es fa en l'àmbit d'aquesta assignatura, per tant, de cara al que es demana haurem d'oblidar aquesta expressió per poc encertada. Segurament l'autor vol donar a entendre que es tracta de manipular informació sense estructura o amb una estructura poc significativa (bàsicament un missatge té una capçalera amb

unes poques dades estructurades, un cos o text i documents adjunts que poden ser qualsevol cosa).

Una altra trampa de l'enunciat és quan parla de fer un *resum*. Fer un resum d'un text és una activitat força complexa i difícil d'automatitzar. Potser quan a les assignatures d'intel·ligència artificial (IA) us parlin del tractament del llenguatge natural entendreu millor encara el que vull dir. No podem plantejar-nos, doncs, de fer cap resum. En el món de la missatgeria electrònica s'entén que el camp *assumepte* conté un resum superconcentrat del contingut, aquest és l'únic resum que tenim i podem tractar del missatge.

Començarem per estudiar l'estructura de dades, i després, un cop establerta quina informació conté, quina cal mantenir i manipular i quins són els rols de cada classe en el sistema, estudiarem els casos d'ús necessaris en conseqüència.

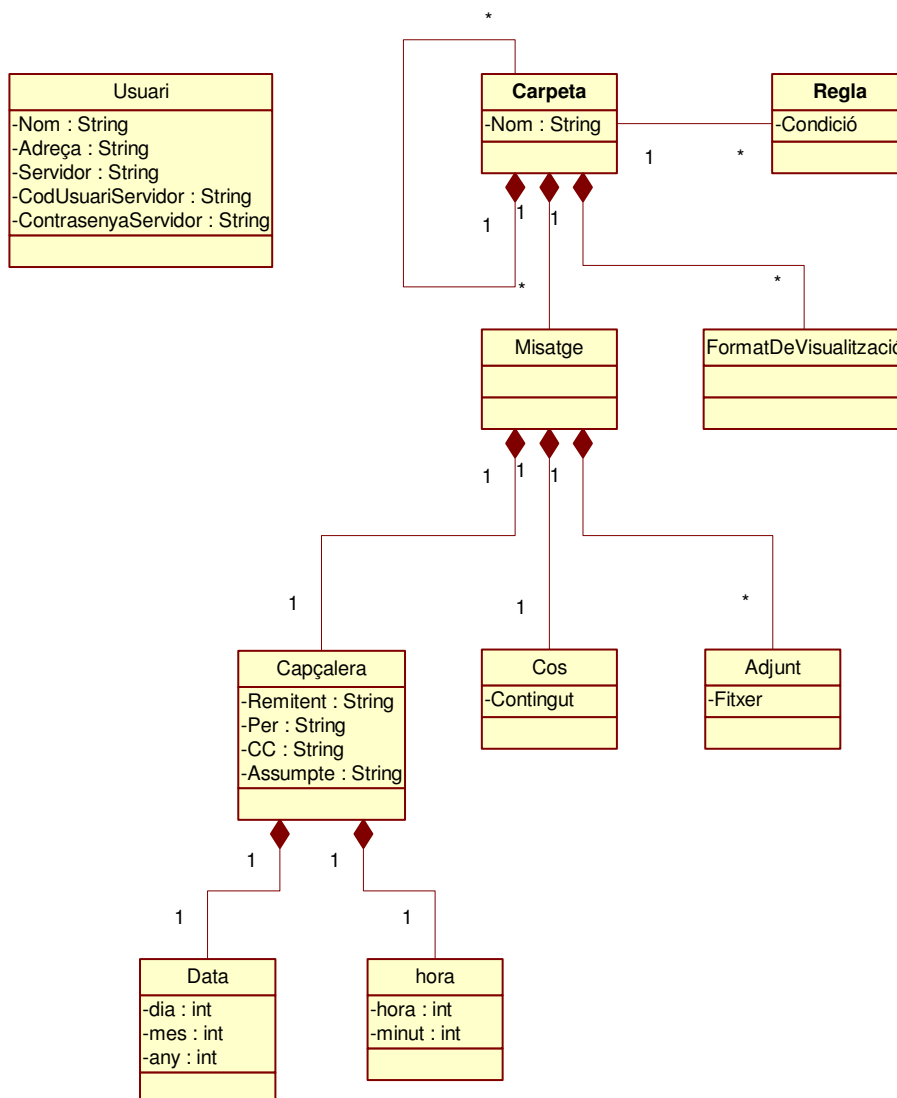
Podem treballar amb quatre classes bàsiques: el missatge, el contenidor (que per adaptar-nos al que hi ha al mercat és millor dir-ne *carpeta*), les regles de classificació del correu entrant i l'usuari. El missatge conté com a atributs, una capçalera (que pot ser una agregació ja que té una estructura definida) un cos (que avui en dia pot tenir format text o HTML, fonamentalment) i documents adjunts (que circulen per *internet* en format MIME). De documents adjunts pot no haver-n'hi cap, o haver-n'hi simplement un, o més d'un.

La carpeta haurà de tenir nom, i podrà contenir missatges i carpetes. Podríem pensar en diferenciar entre la carpeta de treball (on classificar el correu entrant) i carpetes històriques (on emmagatzemar el correu que ha estat llegit i tramitat). Aquesta diferenciació es pot fer simplement amb estructura de l'arbre de carpetes, no cal que les carpetes tinguin unes característiques especials, per tant no les diferenciarrem. En canvi, se sol fer un procés que permeti passar a magatzems especials (fitxers exportables o gravables en CD o cinta) la informació que fa un temps determinat que no es toca (que és històrica o obsoleta).

Les regles, han d'estar escrites en un determinat llenguatge que convindria definir, la tendència és basar-lo en l'àlgebra de Boole, però com que això sol resultar complex per a l'usuari se solen obtenir mitjançant un assistent de manera que l'usuari no arriba mai a veure aquesta àlgebra a no ser que se li permeti a través d'algun accés avançat. A més, s'hauran de poder ordenar les regles de manera que unes manin sobre les altres (prioritats), això es pot fer simplement establint l'ordre d'execució. Cada regla farà que els missatges que es rebin i que la satisfacin s'emmagatzemin en una carpeta concreta en el moment de rebre'ls. Aquest moment serà, normalment, el moment de posar en marxa el client o també es pot donar l'opció a l'usuari de desaparar una funcionalitat d'obtenir missatges del servidor, aquesta també haurà d'aplicar les regles als missatges. És a dir, que un missatge que satisfà una regla determinada s'emmagatzema en la carpeta corresponent i no es comprova si satisfà cap altre de les regles que quedin a la llista i que encara no han estat comprovades. En una carpeta hi podran emmagatzemar missatges tantes regles com es vulgui. Sempre hi ha d'haver una carpeta on s'emmagatzemin els missatges que no satisfan cap regla (sol ser la safata d'entrada, *bandeja de entrada* o *inbox*).

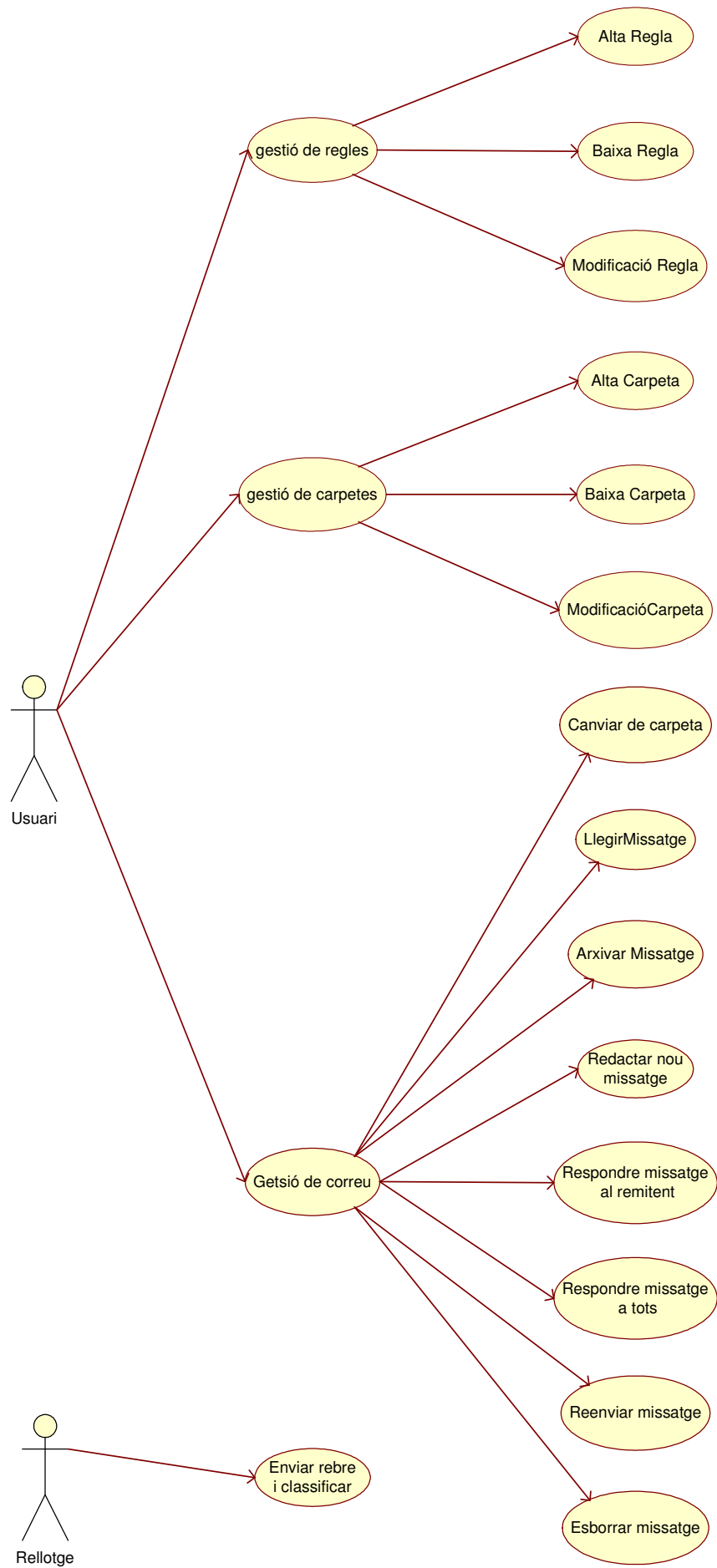
L'estructura de la classe *Usuari* ha de servir per emmagatzemar la seva identificació, aquesta se sol usar per a la tramesa de missatges: nom i a/e (adreça electrònica), i les dades de validació en el servidor de correu: nom o identificació del servidor, codi d'usuari i contrasenya. Si es pensa en un client multiusuari, cosa que no s'especifica en l'enunciat, s'hauria de preveure la necessitat de validar cada usuari del sistema amb un codi d'usuari i una contrasenya (aquest conjunt codi/contrasenya hauria de poder ser diferent del que s'usa pel compte del servidor de correu. En tot cas es pot relacionar amb els que usi per validar-se al sistema operatiu. S'intenta sempre d'unificar la validació del compte amb la del servidor de correu, però no sempre és possible, especialment si el servidor és remot i si accedeix amb POP3 o IMAP4).

Una primera proposta pel **diagrama de classes** podria ser:



Si el sistema fos multiusuari caldria una associació de l'usuari amb les seves carpetes, però si és per a un únic usuari en podem prescindir.

Lligar els formats de visualització a les carpetes és una opció, però no la única. Per exemple, en l'Outlook els formats de visualització de les carpetes de correu solen servir per a totes les carpetes, tot i que Outlook també permet fer-ne d'ús privat, només per a una carpeta.



Un primer **diagrama de casos d'ús** podria ser aquest. Algunes de les funcionalitats

com respondre el missatge a tots i alguna altra no ha sortit de l'enunciat, sinó del coneixement d'altres clients de correu.

Per simplificar he obviat les funcionalitats de gestió de la classe usuari, de les visualitzacions i aquelles que permetrien imprimir. També observareu que manca alguna manera del canviar el format de visualització de la carpeta: en aquest cas es podria fer que en triar la carpeta amb que es vol treballar es triés el format de visualització que es desitgi.

Si s'hagués pensat en un emmagatzematge automàtic de la informació històrica, l'usuari **rellotge** es faria càrrec del traspàs de missatges vells a l'històric, caldria, però que l'usuari en pogués definir adequadament la configuració (amb les funcionalitats corresponents), bàsicament la freqüència d'execució, i el temps que fa que no es toquen els missatges a traspasar.

Un darrer aspecte a revisar és que pot ser convenient de dotar d'alguna eina que permeti a l'usuari disparar la funcionalitat d'enviar, rebre i classificar el correu rebut, en el diagrama de casos d'ús no s'ha previst.

En aquest punt fóra bo **revisar** profundament els tres documents amb què s'ha treballat (RNF i diagrames de classes i de casos d'ús). Per exemple, en el diagrama de classes es podria estudiar si la relació entre *carpeta* i *missatge* podria també ser d'associació, i també la relació entre *carpetes*, tot i que no sembli tan natural.

També convindria completar el diagrama de casos d'ús amb els aspectes comentats i d'altres que puguin sortir.

Moltes vegades aquesta revisió, abans de fer una primera proposta, requereix d'una nova entrevista amb l'usuari per acabar de concretar els aspectes que no hagin quedat prou clars.

Finalment, un cop definits tots els aspectes que han quedat oberts o per concretar, hauríem de fer el **text descriptiu** amb la primera proposta per a l'usuari o client. En aquest document de descripció textual, a més de descriure el problema cada dia més greu de les acumulacions de correu i del correu que no ens interessa o escombraria, caldrà justificar i descriure la solució que es proposa.

La gestió de carpetes no és difícil de justificar, en tot cas només destaca el fet que no es diferencien les històriques de les de treball. Amb les carpetes podríem discutir el tema de les diferents visualitzacions possibles.

A més, haurem de justificar la necessitat de mantenir les dades de l'estructura Usuari. Se'n necessita una per compte de correu. Fixeu-vos que aquest és un aspecte no hem estudiat amb els diagrames anteriors i que hauríem d'acabar de definir abans de seguir, revisant els diagrames convenientment.

La gestió de missatges, tot i ser una mica més complexa, no us ha de resultar gaire més difícil de justificar. Només caldrà explicar quines funcionalitats poden ser necessàries per gestionar-los i aquesta descripció i justificació no pot resultar complexa pels usuaris habituals d'un sistema de correu (com vosaltres). Només s'hauria de tenir una cura especial en plantejar les eines d'arxiu, especialment les eines automàtiques de passar a un fitxer a part allò que fa un temps determinat que no s'hi acce-

deix. Observeu que explicant això us podeu adonar que hi falta informació a l'estructura de dades del missatge: la data i hora del darrer cop que s'hi ha accedit.

Finalment, i l'he deixada pel final perquè aquesta és la part més delicada del text ja que és la que en realitat justifica l'existència del programa no pas perquè hagi de ser l'última o hagi d'anar al final, s'haurà de parlar de les regles: comesa, formació i gestió. La comesa és la de permetre classificar el correu rebut en carpetes diferent de la de defecte (la safata d'entrada). Caldrà explicar també que cada regla s'associa a una carpeta, que permetrà triar per un determinat concepte o per més d'un (si es permeten regles múltiples amb operacions "i" i "o"), determinar quins son aquests conceptes (contingut del camp remitent, contingut dels camps *To* i *CC*, contingut del camp assumpte, contingut del cos del missatge, documents adjunts, etc.) Per acabar caldrà afegir que es disposarà d'un assistent que guiarà l'usuari en el moment de crear i modificar les regles, assistent que no serà necessari per esborrar-les. Una opció que caldria definir també és si es disposarà d'una gestió avançada d'altres i modificacions de regles directament sobre àlgebra de Boole.

La descripció en el text de com es formen les regles hauria de ser poc formal i al nivell que pugui entendre l'usuari. En tot cas s'hi afegirà un annex (que s'ha d'esmentar en el text com a referència) amb una descripció més rigorosa o formal del llenguatge de formació de les regles.