

Projecte de Programació:

Documentació

Grup 5.5

Casals Vilaplana, Jaume (jaume.casals.vilaplana)
Fernandez Clemente, Adrià (adria.fernandez.clemente)
Juanola Ametller, Martí (marti.juanola)
Sturlese Ruiz, Pol (pol.sturlese)

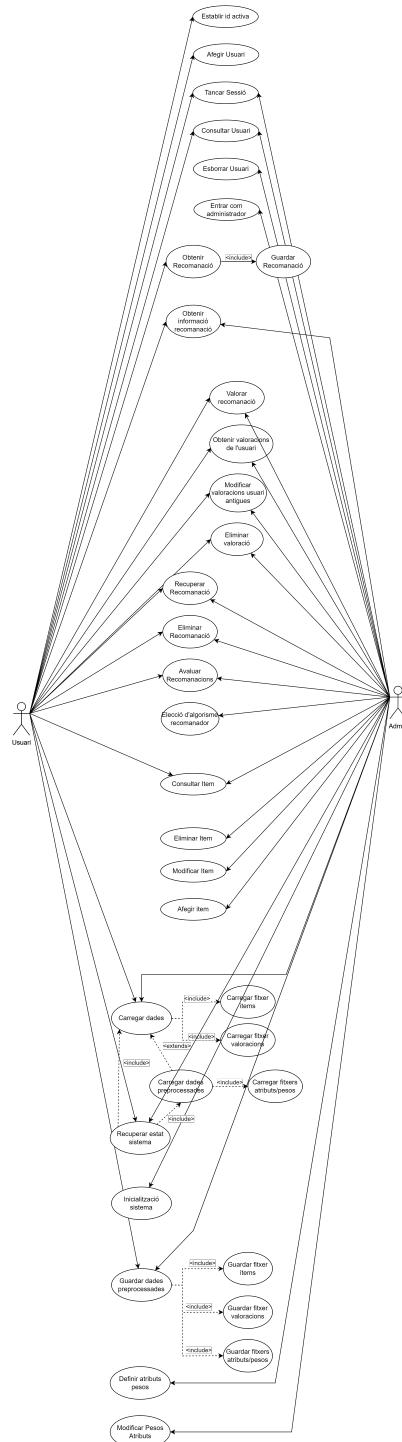
Curs 2021-2022, Semestre de Tardor
Versió 2.0

Continguts

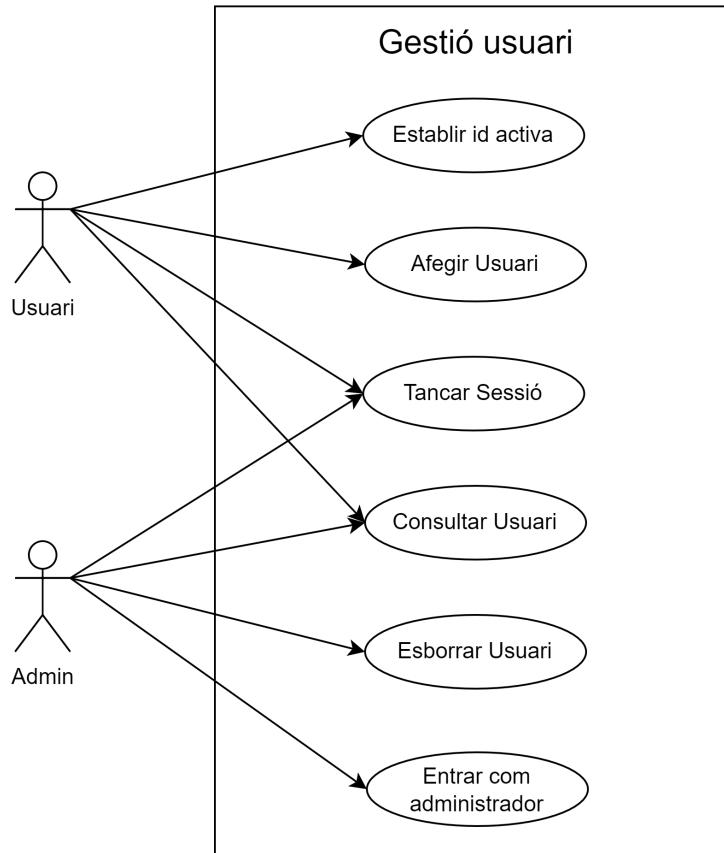
1 Casos d'Us	2
1.1 Diagrama	2
1.2 Explicació	3
2 Diagrama de Classes	12
3 Repartició de les Classes	19
4 Documentació de les Classes	20
4.1 Documentació de la capa de persistència	20
4.1.1 Controlador Presentació	20
4.1.2 Vista Inicial	20
4.1.3 Vista Escollir Atributs	20
4.1.4 Vista Sessió Nova	20
4.1.5 Vista Items	20
4.1.6 Vista Afegir Item	21
4.1.7 Vista Modificar Item	21
4.1.8 Vista Canviar Pesos	21
4.1.9 Vista Informacio Item	21
4.1.10 Vista Error	21
4.1.11 Vista Principal	21
4.1.12 Vista Test Algorisme	22
4.1.13 Vista Modificar Algorisme	22
4.1.14 Vista Total Usuaris	22
4.1.15 Vista Usuari	22
5 Llibreries Externes	23
6 Algorismes i Estructures de Dades	24
6.1 ControladorDominiAlgorisme	24
6.2 ItemValoracioEstimada	24
6.3 CollaborativeFiltering	24
6.4 Centroide	24
6.5 ContentBasedFiltering	24
6.6 HybridFiltering	25
6.7 Controlador Persistencia	25
6.8 Controlador Load	25
6.9 Controlador Save	26
6.10 Usuari	26
6.11 Conjunt Usuaris	26
6.12 Recomanació	26
6.13 Conjunt Recomanacions	26
6.14 Ítem	26
6.15 Conjunt d'Ítems	26

1 Casos d'Us

1.1 Diagrama



1.2 Explicació



Establir ID activa

- *Actor:* Usuari no administrador.
- *Comportament:* Estableix la ID de l'usuari actual, que s'usarà en les recomanacions.
- *Errors possibles i cursos alternatius:* El camp de ID està buit.

Afegir Usuari

- *Actor:* Usuari no administrador.
- *Comportament:* Crea un nou usuari buit amb la ID donada.
- *Errors possibles i cursos alternatius:* L'usuari ja existeix.

Tancar Sessió

- *Actor:* Ambdós.
- *Comportament:* Permet retornar a la selecció d'usuari i per tant intercanviar entre usuari i administrador.
- *Errors possibles i cursos alternatius:* -

Consultar Usuari

- *Actor:* Ambdós.
- *Comportament:* Retorna la ID d'un usuari i totes les seves valoracions.
- *Errors possibles i cursos alternatius:* L'usuari no existeix. L'actor és usuari no administrador i sol·licita informació d'un usuari diferent a ell mateix.

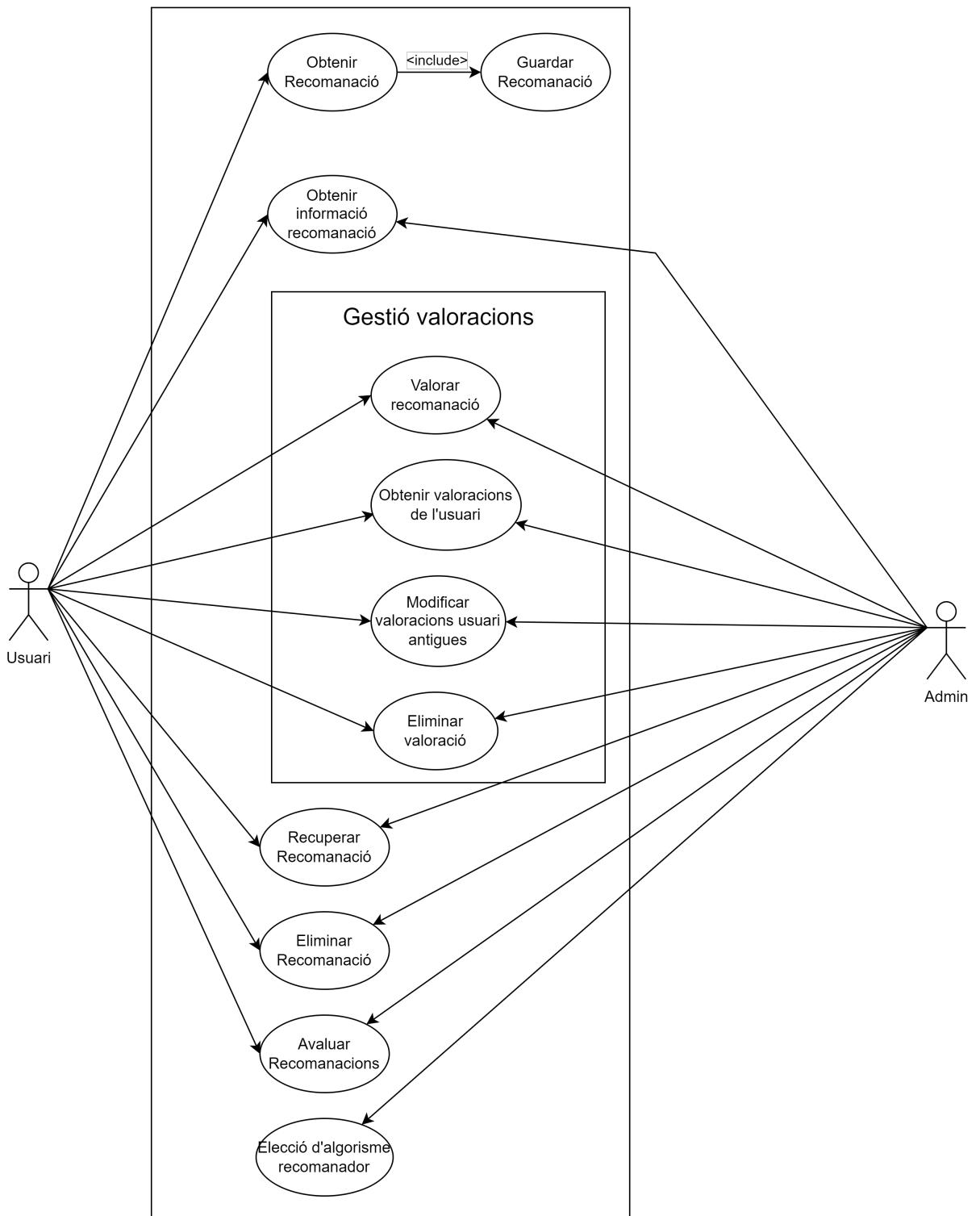
Entrar com a administrador

- *Actor:* Usuari administrador.
- *Comportament:* Estableix a l'usuari actiu com a administrador, i per tant permet realitzar totes les funcionalitats reservades als usuaris administradors.
- *Errors possibles i cursos alternatius:* Usuari i contrasenya equivocada (en implementació real).

Esborrar Usuari

- *Actor:* Usuari administrador.
- *Comportament:* Elimina un usuari i les seves valoracions.
- *Errors possibles i cursos alternatius:* L'usuari no existeix.

Recomanacions



Obtenir recomanació

- *Actor:* Usuari
- *Comportament:* Dona a l'usuari de la ID activa un conjunt d'items recomanats per l'algorisme recomanador, i la guarda al sistema.
- *Errors possibles i cursos alternatius:* No s'han pogut obtenir recomanacions. No hi ha items. Error de l'algorisme recomanador.

Guardar recomanació

- *Actor:* Usuari
- *Comportament:* Afegeix al conjunt de recomanacions una nova instància, sense valorar.
- *Errors possibles i cursos alternatius:* Ja existeix la recomenació de l'item per a l'usuari.

Obtenir informació recomanació

- *Actor:* Usuari administrador
- *Comportament:* Desplega els atributs d'un item recomanat.
- *Errors possibles i cursos alternatius:* L'usuari no és administrador i intenta obreir informació recomanació d'un usuari diferent a ell mateix.

Valorar recomanació

- *Actor:* Usuari administrador
- *Comportament:* Crea una valoració d'un item recomanat per a l'usuari de la ID activa.
- *Errors possibles i cursos alternatius:* L'usuari no és administrador i intenta valorar recomanacions d'un usuari diferent a ell mateix.

Obtenir valoracions de l'usuari

- *Actor:* Usuari administrador
- *Comportament:* Desplega totes les valoracions fetes per l'usuari d'ID activa i permet les funcionalitats de modificar-les si en té.
- *Errors possibles i cursos alternatius:* L'usuari no és administrador i intenta demanar valoracions d'un usuari diferent a ell mateix.

Modificar valoracions antigues usuari

- *Actor:* Usuari administrador
- *Comportament:* Canvia la valoració d'un item feta per l'usuari d'ID activa.
- *Errors possibles i cursos alternatius:* L'usuari no és administrador i intenta modificar recomanacions d'un usuari diferent a ell mateix.

Eliminar valoració

- *Actor:* Usuari administrador.
- *Comportament:* Elimina la valoració d'un item feta per l'usuari d'ID activa.
- *Errors possibles i cursos alternatius:* L'usuari no és administrador i intenta eliminar recomanacions d'un usuari diferent a ell mateix.

Elecció d'algorisme recomanador

- *Actor:* Usuari administrador.
- *Comportament:* Permet seleccionar l'algorisme recomanador entre les opcions Collaborative filtering, Content-based filtering i Hybrid approach.
- *Errors possibles i cursos alternatius:* -

Avaluar Recomanacions

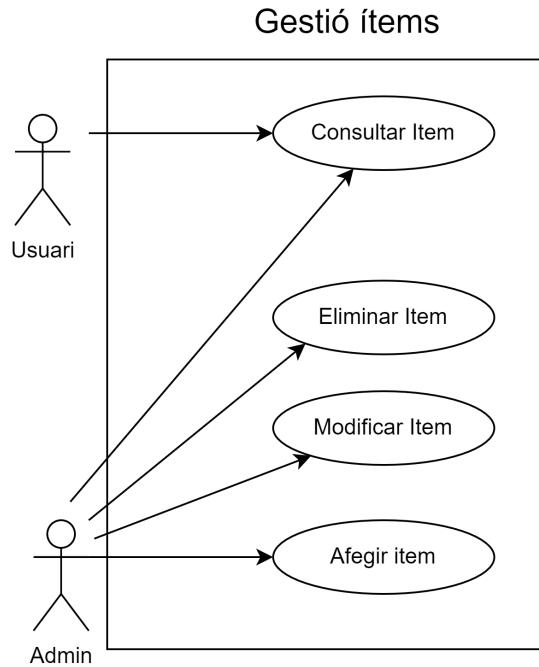
- *Actor:* Usuari administrador.
- *Comportament:* Calcula el DCG i DCG normalitzat de l'algorisme recomanador mitjançant els fitxers Known i Unknown.
- *Errors possibles i cursos alternatius:* No existeixen els fitxers Known i Unknown

Recuperar Recomanació

- *Actor:* Usuari administrador
- *Comportament:* Mostra les recomanacions antigues que ha fet el sistema, si n'hi ha.
- *Errors possibles i cursos alternatius:* L'usuari no és administrador i intenta recuperar recomanacions d'un usuari diferent a ell mateix.

Eliminar Recomanació

- *Actor:* Usuari administrador
- *Comportament:* Elimina una recomenació del conjunt de recomanacions.
- *Errors possibles i cursos alternatius:* L'usuari no és administrador i intenta eliminar recomanacions d'un usuari diferent a ell mateix. No existeix la recomanació.



Afegir Item

- *Actor:* Usuari administrador.
- *Comportament:* Afegeix un item al conjunt d'items.
- *Errors possibles i cursos alternatius:* Ja existeix l'item. Atributs incorrectes.

Eliminar Item

- *Actor:* Usuari administrador.
- *Comportament:* Elimina un item del conjunt d'items.
- *Errors possibles i cursos alternatius:* L'item no existeix.

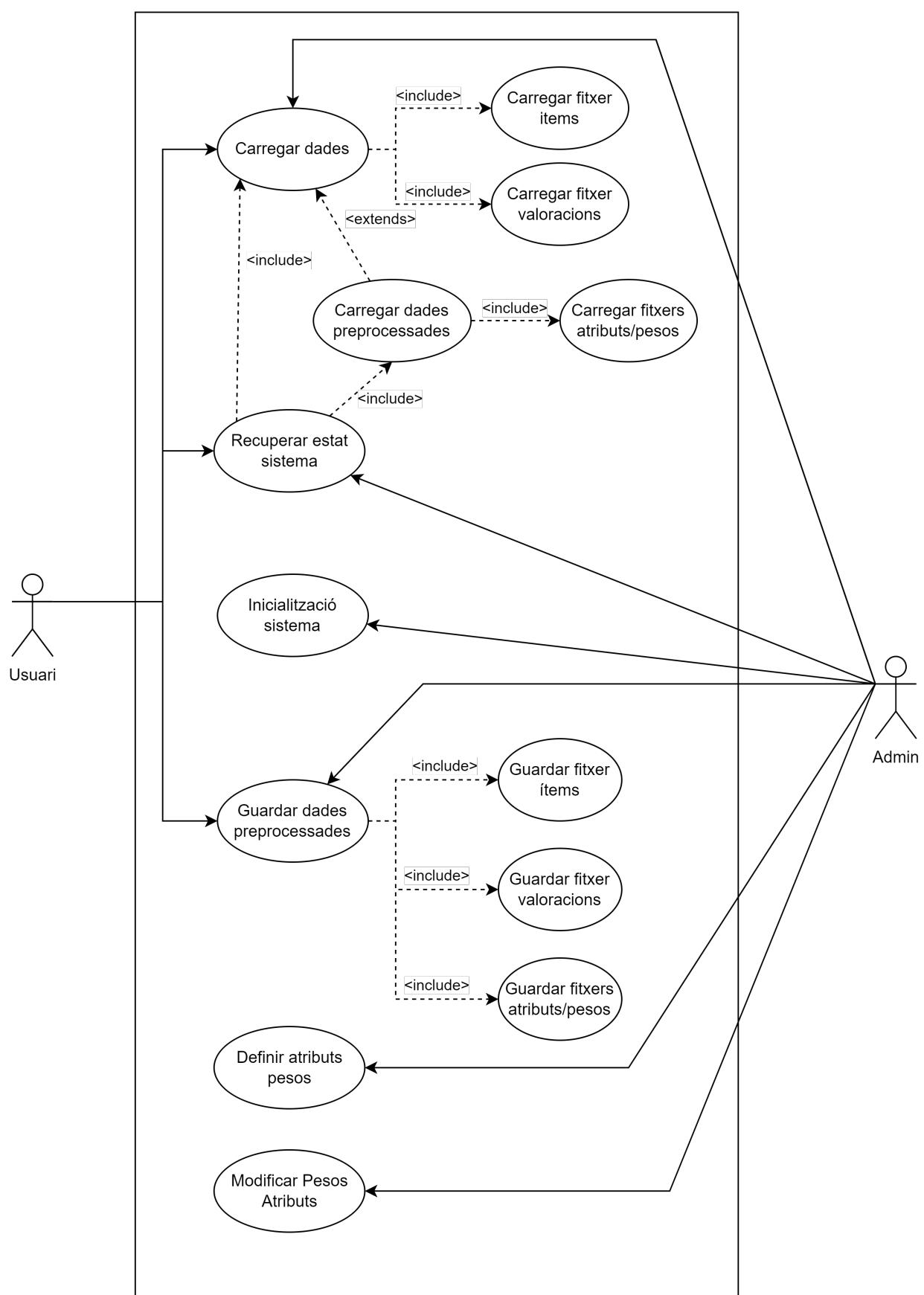
Modificar Item

- *Actor:* Usuari i administrador.
- *Comportament:* Modifica els atributs d'un item.
- *Errors possibles i cursos alternatius:* L'item no existeix.

Consultar Item

- *Actor:* Usuari.
- *Comportament:* Obte els atributs d'un item.
- *Errors possibles i cursos alternatius:* L'item no existeix.

Gestió sistema



Carregar dades

- *Actor:* Usuari administrador.
- *Comportament:* Carrega al programa una sèrie d'arxius. Aquest cas d'us serà més útil per la primera execució, ja que no inclou els pesos dels atributs.
- *Errors possibles i cursos alternatius:* Arxius incorrectes.

Carregar fitxer items

- *Actor:* Usuari administrador.
- *Comportament:* Carrega al programa un arxiu amb un conjunt d'items.
- *Errors possibles i cursos alternatius:* Arxiu incorrecte

Carregar fitxer valoracions

- *Actor:* Usuari administrador.
- *Comportament:* Carrega al programa un arxiu amb un conjunt de valoracions d'items fetes per usuaris. Es permet carregar un arxiu amb només la capçalera per si es vol començar *tabula rasa*.
- *Errors possibles i cursos alternatius:* Arxiu incorrecte

Carregar dades preprocessades

- *Actor:* Usuari administrador.
- *Comportament:* Carrega al programa l'estat guardat d'una execució anterior. Inclou les funcionalitats anteriors però afegint la càrrega de la definició d'atributs, els seus pesos i dades necessàries pel correcte funcionament de l'algorisme i de les classes usades.
- *Errors possibles i cursos alternatius:* Arxius incorrectes

Recuperar estat sistema

- *Actor:* Usuari administrador.
- *Comportament:* Carrega al programa l'últim estat guardat del sistema, eliminant les dades que no hagin estat guardades en el moment de fer-ho.
- *Errors possibles i cursos alternatius:* S'intenta fer el guardat amb dades invàlides

Inicialització sistema

- *Actor:* Administrador.
- *Comportament:* Inicialitza el sistema a partir de les dades d'item que entra l'administrador.
- *Errors possibles i cursos alternatius:* S'intenta fer la inicialització amb dades invàlides

Guardar dades preprocessades

- *Actor:* Usuari administrador.
- *Comportament:* Guarda en els fitxers corresponent l'estat del programa.
- *Errors possibles i cursos alternatius:* Existeixen dades incorrectes

Guardar fitxer items

- *Actor:* Usuari administrador.
- *Comportament:* Guarda en un fitxer la informació pertanyent als ítems.
- *Errors possibles i cursos alternatius:* Existeixen dades incorrectes

Guardar fitxer valoracions

- *Actor:* Usuari administrador.
- *Comportament:* Guarda en un fitxer la informació pertanyent a les valoracions.
- *Errors possibles i cursos alternatius:* Existeixen dades incorrectes

Guardar fitxer atributs/pesos

- *Actor:* Usuari administrador.
- *Comportament:* Guarda en diversos fitxers la informació addicional pertanyent a l'execució del programa que facilita la càrrega posterior de les dades.
- *Errors possibles i cursos alternatius:* Existeixen dades incorrectes

Definir atributs pesos

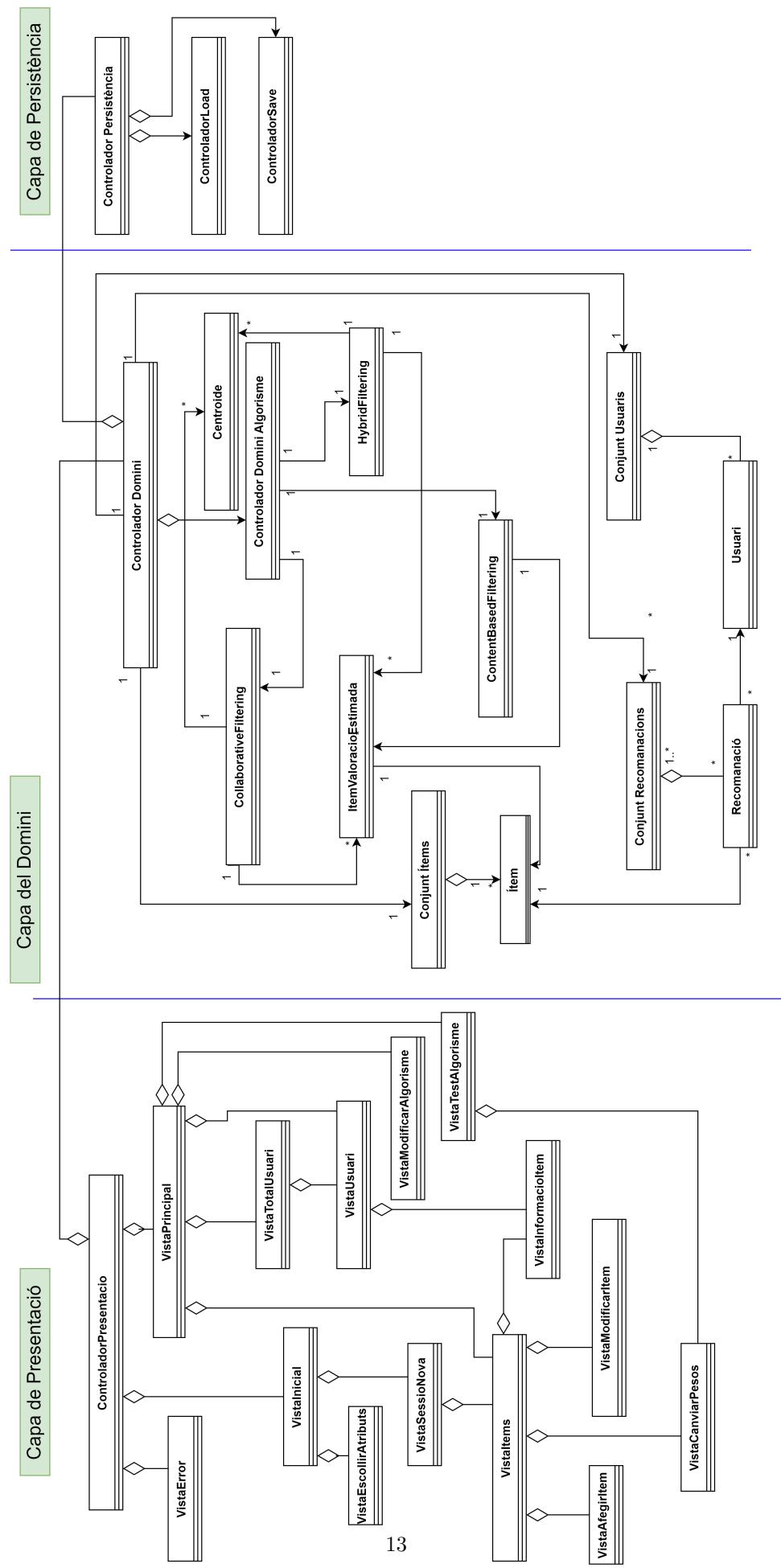
- *Actor:* Administrador.
- *Comportament:* Permet la definició dels pesos que tenen cada un dels atributs en l'execució de l'algorisme.
- *Errors possibles i cursos alternatius:* Existeixen dades incorrectes

Modificar atributs pesos

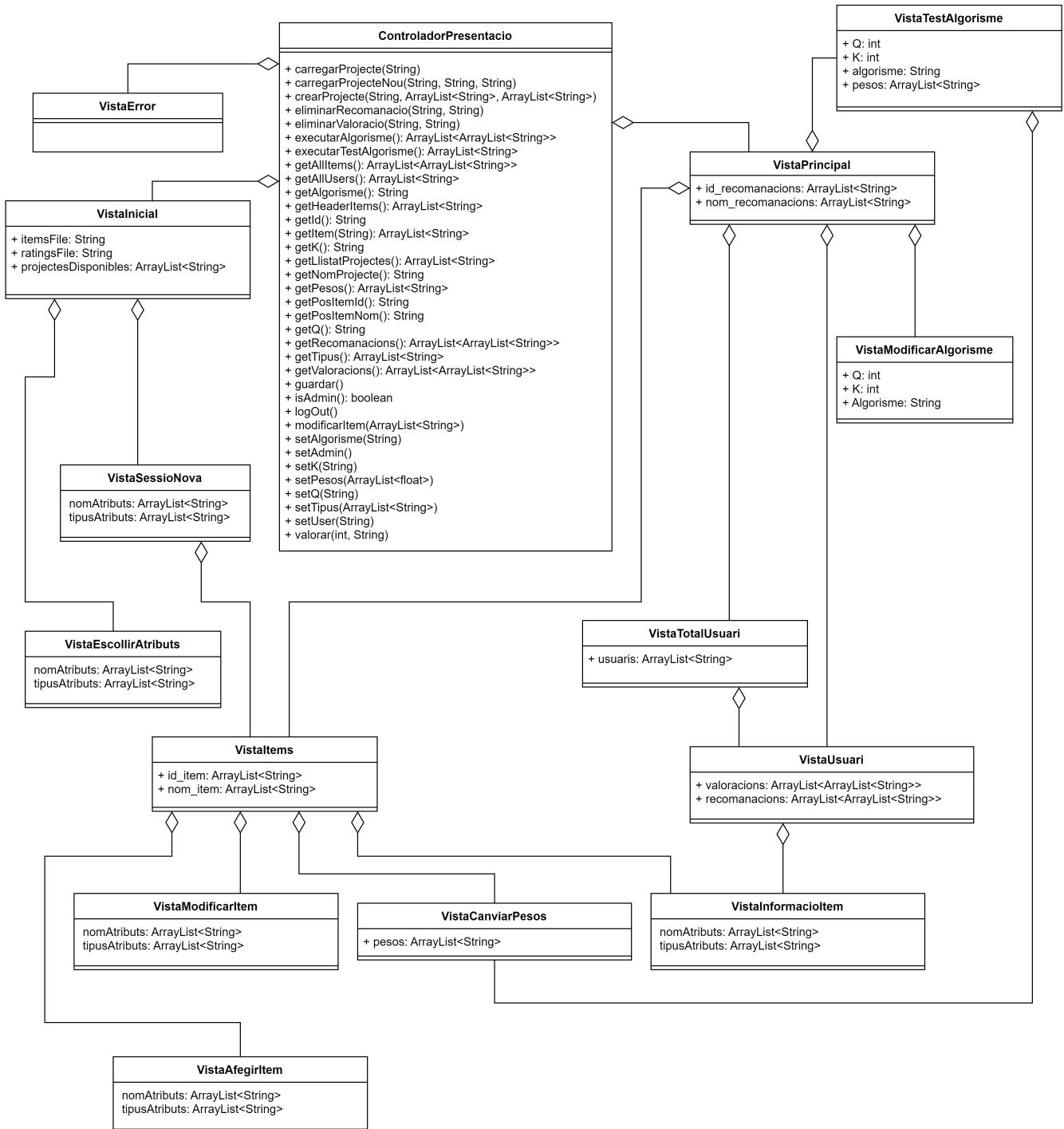
- *Actor:* Administrador.
- *Comportament:* Permet la modificació dels pesos que tenen cada un dels atributs en l'execució de l'algorisme.
- *Errors possibles i cursos alternatius:* Existeixen dades incorrectes

2 Diagrama de Classes

A l'entrega digital s'ha adjuntat el diagrama complert en format *html* per si es vol veure amb major detall.

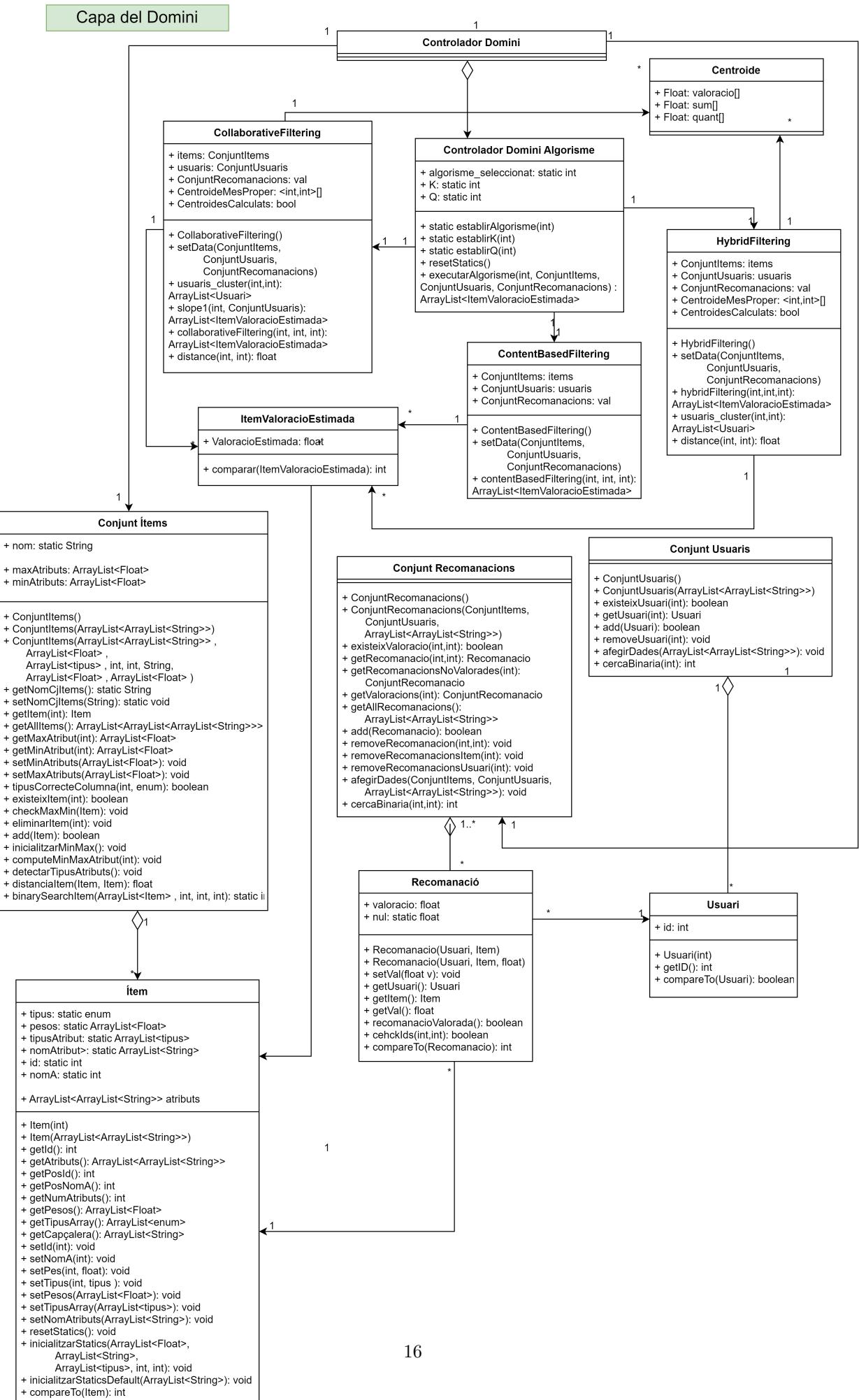


Capa de Presentació



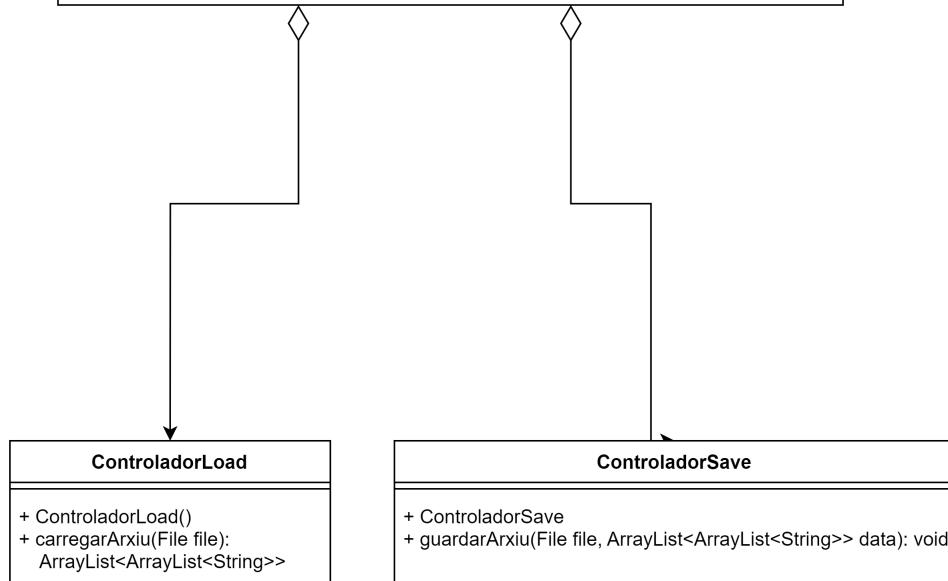
Controlador Domini

Controlador Domini	
+ id: static int + admin: static boolean	
+ ControladorDomini() + login(int): void + loginAdmin(): void + logout(): void + getActiveUserId(): String + isAdmin(): boolean + loadSession(String): void + createSession(String,String,String): void + createEmptySession(String,ArrayList<String>, ArrayList<String>): void + saveSession(): void + runTest(): ArrayList<String> + getAllItems(): ArrayList<ArrayList<ArrayList<String>>> + getItem(int): ArrayList<ArrayList<String>> + getPosIdItem(): String + getPosNomItem(): String + getHeader(): ArrayList<String> + setNomItem(String): void + addlItem(ArrayList<ArrayList<String>>): void + removeItem(String): void + editItem(ArrayList<ArrayList<String>>): void + getPesos(): ArrayList<String> + setPesos(ArrayList<String>): void + getTipus(): ArrayList<String> + setTipus(ArrayList<String>): void + getAllRecomanacionsUsuari(String): ArrayList<ArrayList<String>> + getAllValoracionsUsuari(String): ArrayList<ArrayList<String>> + setValoracio(String,String,String): void + removeValoracio(String,String): void + removeRecomanacio(String,String): void + createRecomanacions(): void + getK(): String + getQ(): String + getAlgorisme(): String + setK(String): void + setQ(String): void + setAlgorisme(String): void + getAllUsuaris(): Array<String> + addUsuari(String): void + removeUsuari(String): void + getTipusAsStrings(): ArrayList<String> + getNomProjecte(): String + getAllProjectes(): ArrayList<String> + resetData(): void	1 1 1 1



Capa de Persistència

Controlador Persistència
<pre>+ carpeta: File + dades: File</pre>
<pre>+ ControladorPersistencia() + getNomProjecte(): String + llistatCarpetes(): ArrayList<String> + existeixenDadesPreprocesades(): boolean + existeixenTestos(): boolean + escollirProjecte(string s): void + sortirDelProjecte(): void + carregarArxiuCarpeta(String s): ArrayList<ArrayList<String>> + carregarRecomanacionsCarpeta(): ArrayList<ArrayList<String>> + carregarItemsCarpeta(string file): ArrayList<ArrayList<String>> + carregarPesosAtributs(): ArrayList<String> + carregarTipusAtributs(): ArrayList<String> + carregarTestKnown(): ArrayList<ArrayList<String>> + carregarTestUnknown(): ArrayList<ArrayList<String>> + carregarFitxerExtern(string s): ArrayList<ArrayList<String>> + carregarMaxAtributsItems(): ArrayList<String> + carregarMinAtributsItems(): ArrayList<String> + carregarEstat(): ArrayList<String> + crearProjecte(String s): void + guardarDades(ArrayList<ArrayList<String>> D, String s): void + guardarRecomanacions(ArrayList<ArrayList<String>> D): void + guardarItems(ArrayList<String> head, ArrayList<ArrayList<ArrayList<String>>> body): void + guardarPesosAtributs(ArrayList<Float> pesos): void + guardarTipusAtributs(ArrayList<String> tip): void + guardarMaxAtributsItems(ArrayList<String> max_atr): void + guardarMinAtributsItems(ArrayList<String> min_atr): void + guardarEstat(ArrayList<String>): void</pre>



Restriccions Textuals:

- Cada usuari té un Conjunt de Recomanacions de les seves recomanacions no valorades i un Conjunt de Recomanacions de les seves recomanacions valorades
- El controlador de domini té un Conjunt Recomanacions amb totes les recomanacions de tots els usuaris, valorades i sense valorar
- Les classes HybridFiltering i CollaborativeFiltering tindran tantes instàncies de Centroide com el paràmetre K

3 Repartició de les Classes

L'implementació de les classes s'ha realitzat de la següent forma:

Pol Sturlese:

- Controlador Persistència + Driver
- Controlador Load + Driver + JUnit
- Controlador Save + Driver + JUnit
- Capa de Presentació
- Driver StringOperations

Martí Juanola:

- Controlador Domini
- Recomanacio + Driver
- ConjuntRecomanacions + Driver
- ConjuntUsuaris + Driver
- Usuari + Driver
- Driver Item

Adrià Fernández:

- ItemValoracioEstimada + Driver
- ControladorDominiAlgorisme + Driver
- CollaborativeFiltering
- ContentBasedFiltering
- Hybrid Filtering
- Centroide

Jaume Casals:

- Item
- Capa de Presentació
- ConjuntItems + Driver
- StringOperations
- UnionIntersection + Driver

4 Documentació de les Classes

La documentació de les classes de la capa de domini i la capa persistència es pot trobar en l'arxiu *html Javadoc* entregat digitalment. La documentació de les classes de la capa de presentació, es troba en el següent apartat.

4.1 Documentació de la capa de persistència

A continuació es mostra la documentació de les classes de la capa de persistència que, a part del controlador, han sigut dissenyades tenint en ment les pantalles que representen.

4.1.1 Controlador Presentació

El controlador d'aquesta capa conté la funció principal d'aquest projecte que inicialitza les altres capes i controla quina pantalla es mostrada en cada moment a l'usuari. Conté, sobretot, funcions per cridar a capes inferiors i funcions per obrir finestres noves.

4.1.2 Vista Inicial

Tal i com s'indica en el nom, aquesta és la primera finestra que es mostra al iniciar el programa, conté botons, camps de text i desplegables per permetre l'usuari:

- Identificar si és administrador o usuari.
- Quin identificador d'usuari vol utilitzar (en cas de ser usuari).
- Quin nom de projecte vol utilitzar (en cas de crear un nou projecte).
- L'opció de crear una sessió nova (només si és administrador).
- L'opció de crear una sessió a partir de fitxers, i uns buscadors per escollir-los.
- L'opció de carregar les dades d'un projecte creat anteriorment.

A partir de les opcions escollides (sessió nova, sessió nova a partir de fitxers i carregar dades anteriors) es poden obrir diferents vistes (Vista sessió nova, Vista escollir atributs o Vista Principal, respectivament).

4.1.3 Vista Escollir Atributs

Aquesta vista es mostra després de crear una sessió a partir de un fitxer d'ítems i un fitxer de valoracions. Aquí es mostra un llistat dels atributs actuals amb uns tipus generats automàticament pel programa, i s'indica a l'usuari que revisi que els tipus siguin correctes i indiqui, obligatoriament, quin del atributs representa el nom de l'ítem.

Al finalitzar, s'obre la Vista Principal.

4.1.4 Vista Sessió Nova

Aquesta vista es creada justament després de crear una sessió nova, i permet a l'usuari indicar quins atributs tindrà la seva sessió i de quin tipus serà cada atribut.

Al finalitzar, s'obre la Vista Items.

4.1.5 Vista Items

Aquesta vista es creada quan es volen mostrar tots els ítems que conté actualment un projecte i es vol donar l'opció d'afegir, eliminar, consultar o modificar algun ítem.

Inicialment es mostrarà els noms dels ítems i opcions al seu costat per consultar en cas de ser usuari o modificar i eliminar en cas de tenir permisos d'administrador. En un lloc apart, es mostrarà un botó per afegir ítems i un botó per mostrar els pesos, ambdues opcions només si es tenen permisos d'administrador.

Al utilitzar qualsevol funcionalitat ens portarà a una vista diferent:

- Modificar ítem: Obre la Vista Modificar Item.
- Consultar ítem: Obre la Vista Informació Item.

- Afegir ítem: Obre la Vista Afegir Item.
- Modificar pesos: Obre la Vista Canvia Pesos.
- Finalitzar: Tanca la vista actual i retorna les modificacions a la vista anterior.
- Eliminar ítem: No obre cap vista nova.

4.1.6 Vista Afegir Item

La Vista Afegir Item es crida per la Vista Items quan es necessari afegir un ítem nou.

Aquesta vista conté el nom dels atributs actuals, el seu tipus i espais per permetre a l'usuari omplir les dades del nou ítem.

Al finalitzar, es tanca la vista i es retornen les dades del nou ítem.

4.1.7 Vista Modificar Item

Vista Modificar Items es creada per Vista Items quan es selecciona un ítem que es vol modificar.

Es mostren els atributs actuals de l'ítem, juntament amb el nom de cada atribut i el seu tipus. Els atributs es poden modificar.

Al finalitzar, es tanca la vista i es retonen els nous atributs.

4.1.8 Vista Canviar Pesos

Vista Canviar Pesos es creada per Vista Items quan un administrador indica que es vol canviar els pesos dels atributs del projecte actual.

Es mostren els pesos actuals de l'ítem, juntament amb el nom de cada atribut. Els pesos es poden modificar amb un *slider*.

Al finalitzar, es tanca la vista i es retonen els nous pesos.

4.1.9 Vista Informacio Item

Vista Informacio Item es creada per Vista Items quan s'indica que es volen consultar els atributs d'un ítem del projecte actual.

Es mostren els atributs actuals de l'ítem, juntament amb el nom de cada atribut i el seu tipus. Els atributs no es poden modificar.

Al finalitzar, es tanca la vista.

4.1.10 Vista Error

Vista Error és una vista que es crida quan es vol notificar a l'usuari d'un error, tant per culpa del usuari com un error del sistema.

4.1.11 Vista Principal

Aquesta vista és l'encarregada de mostrar la major part de funcionalitats del programa. Té funcionalitats compartides per l'usuari i l'administrador, i algunes que són específiques de cada un dels dos rols mencionats. Les funcionalitats compartides són les següents:

- Accés a la finestra d'obtenció d'informació de tots els ítems.
- Accés a la finestra de testeig de l'algorisme recomanador.
- Guardar l'estat del programa actual.
- Recuperar l'últim estat guardat, eliminant així tots els canvis recents que no s'hagin emmagatzemat.
- Sortir al menú d'elecció de projecte (és a dir, anar a la Vista Inicial).

A part d'aquestes funcions, l'administrador té també accés a unes funcions pensades per canviar de vista i mostrar certes dades. Les funcionalitats són les següents:

- Accés a la finestra d'obtenció d'informació de tots els usuaris

- Accés a la finestra de modificació de dades de l'algorisme

L'usuari, per altre banda, té accés a un botó que el porta a la vista amb la seva informació d'usuari. També té un apartat on pot demanar al programa que executi l'algorisme i li recomani un cert nombre d'objectes que es podrà valorar des de la mateixa finestra. Les recomanacions mostrades estaran disponibles també a la Vista Usuari. Es podrà obtenir més informació de les recomanacions obtingudes fent click sobre la recomanació.

4.1.12 Vista Test Algorisme

Aquesta vista és l'encarregada del testeig de l'algorisme i les recomanacions que s'obtenen. En aquesta vista es pot seleccionar els paràmetres K i Q de l'algorisme, així com l'algorisme que es vol usar pel càlcul i els pesos que tenen els atributs. Com que és una finestra de testeig, per defecte hi seràn els valors que hi hagués durant l'execució, però qualsevol canvi que es faci en aquesta vista serà temporal. D'aquesta manera, tant l'usuari com l'administrador poden provar les combinacions que prefereixin sense tenir cap efecte en el programa en general.

La vista aquesta fà us de dos arxius: ratings.db.known.csv i ratings.db.unknown.csv. Com que la part de testeig està pensada per a usuaris experts, la introducció d'aquests fitxers s'haurà de fer manualment a la carpeta del projecte de dins de dades.

4.1.13 Vista Modificar Algorisme

Aquesta vista és l'encarregada del control de les dades de l'algorisme. Es poden modificar els paràmetres K i Q i l'algorisme usat en l'execució del programa.

4.1.14 Vista Total Usuaris

Aquesta vista pot ser accedita només per l'administrador i mostra tots els usuaris que hi ha al sistema. L'administrador podrà tant eliminar un usuari com veure la seva informació: és a dir, accedir a la Vista Usuari mostrant la informació de l'usuari seleccionat.

4.1.15 Vista Usuari

Aquesta vista és l'encarregada de mostrar les dades d'un usuari. És a dir, mostra la seva id i les seves valoracions i recomanacions. En aquesta vista es permet modificar o eliminar les diverses recomanacions i valoracions. També es pot accedir a una vista amb l'informació de les recomanacions fent click sobre aquesta.

Aquesta vista pot ser accedita tant per l'usuari com per l'administrador. En el cas de l'usuari, només pot veure la seva informació, mentre que l'administrador té accés a la informació d'un usuari qualsevol (s'haurà escollit en una altra vista, la Vista Total Usuaris).

5 Llibreries Externes

Per l'implementació del treball s'han emprat les següents llibreries estàndards de Java:

- *ArrayList*
- *HashMap*
- *Collections*
- *Math*
- *File*
- *FileReader*
- *FileWriter*
- *Random*

Addicionalment, hem emprat la llibreria *Swing* per a l'elaboració de la capa de presentació. No hem emprat cap llibrería externa a Java.

6 Algorismes i Estructures de Dades

6.1 ControladorDominAlgorisme

Aquesta classe fa servir l'estructura de dades ArrayList per emmagatzemar els resultats dels algorismes recomanadors. No implementa cap algorisme recomanador donat que aquest el realitzen les respectives classes.

Els algorismes que si fa servir són el càlcul de DCG, IDCG i NDCG descrits a la documentació (que fan servir el càlcul de potències i logaritmes de la llibreria Math) i l'algorisme d'ordenació de llistes que ens dona la llibreria Collections, que hem usat per ordenar les valoracions estimades d'Unknown com prèviament descrit.

Cal mencionar que per al càlcul del DCG es tenen en compte les valoracions estimades iguals com a la mateixa posició. Si s'arrodoneixen les valoracions estimades al múltiple de 0.5 més proper el DCG millora substancialment, donat que moltes valoracions tindran la mateixa posició (resultat de sobre 0.8 en el normalitzat). En el nostre cas, però, hem determinat que no arrodonirem les valoracions estimades donat que son una estimació, la qual cosa repercuteix negativament sobre el DCG tot i obtenir els mateixos resultats (resultat de sobre 0.4 en el normalitzat).

6.2 ItemValoracioEstimada

Aquesta classe senzilla no empra cap estructura de dades ni algorisme, excepte una funció que donat un ItemValoracióEstimada retorna el major entre l'instància actual i el parametre (la qual cosa ens permet l'ordenació mitjançant la funcionalitat Sort de la llibreria Collections).

6.3 CollaborativeFiltering

Aquesta classe empra les estructures de dades ConjuntItems, ConjuntUsuaris i ConjuntRecomanacons, que són extensions de la classe ArrayList. També fa servir un array de la classe Centroide, Fem servir un HashMap en l'atribut ClosestCentroid, que ens guarda per a cada usuari quin és el seu centroide més proper. També emprem la classe ItemValoracióEstimada, que ens permet ordenar les recomanacions segons la seva valoració estimada.

D'algorismes hem usat els algorismes descrits a la documentació per organitzar els usuaris en clústers d'usuaris semblants (k-Means) i per posteriorment obtenir valoracions estimades (Slope-1).

6.4 Centroide

Emmagatzema tres HashMaps: valoració, sum i quant. Aquests mapes emmagatzemen una variable per a cada item, i ens serveixen per calcular les distàncies i recalcular les mitjanes.

6.5 ContentBasedFiltering

Per a aquesta classe hem emprat l'estructura de dades ArrayList per a emmagatzemar un conjunt d'ItemValoracióEstimada que ens permet ordenar un conjunt d'ítems segons la seva valoració estimada.

Per a la part d'algorisme, hem implementat l'algorisme proposat a la documentació de k-NearestNeighbours. L'hem implementat amb aquestes dues consideracions que hem vist adients:

La similitud entre dos ítems (aquesta funció es troba a conjunt d'ítems, però hem vist adient descriure-la aquí) l'hem calculat sumant la similitud dels seus atributs multiplicada per la seva ponderació. La similitud dels atributs es calcula així:

- Si són valors numèrics, la diferència dels valors dividida entre la diferència del valor màxim i mínim de tots els ítems.
- Si són dates es tracta igual que amb números però passant les dates a dies.
- Si són valors booleans 1 si són iguals, 0 altrament.

- Si són strings la distància d'editat (calculada amb programació dinàmica) dividida entre la mida màxima de les dues strings.
- Si es un conjunt de strings (és a dir, tags) és la mida de la intersecció dividit entre la mida de la unió.

Per decidir com ordenar les valoracions i obtenir-ne les Q millors, la documentació proposa ordenar-les per nombre d'aparicions, tot i que ho deixa obert a discussió. Nosaltres hem decidit que ens semblava una millor alternativa multiplicar la similitud per la valoració de l'ítem semblant i agafar aquest valor més alt com a valoració estimada de l'ítem. Això valora més els ítems semblants a ítems molt ben valorats, i ens retorna una valoració estimada el qual facilitarà la implementació d'estratègies híbrides.

Finalment, l'algorisme demana definir un “threshold” a partir dels quals es consideren les valoracions de l'usuari, per a només agafar les positives. Nosaltres hem considerat adequat agafar el valor 3.5, però hem plantejat altres alternatives com trobar la valoració en la posició $\lfloor \text{mida} * 0.75 \rfloor$ que es podria calcular eficientment usant l'algorisme de selecció.

6.6 HybridFiltering

Aquesta classe empra les estructures de dades ConjuntItems, ConjuntUsuaris i ConjuntRecomanacons, que són extensions de la classe ArrayList. També fa servir un array de la classe Centroide, Fem servir un HashMap en l'atribut ClosestCentroid, que ens guarda per a cada usuari quin és el seu centroide més proper. També emprem la classe ItemValoracióEstimada, que ens permet ordenar les recomanacions segons la seva valoració estimada.

L'algorisme implementat consisteix en:

- Separar el conjunt d'usuaris en clusters seguint el procediment de k-Means
- Obtenir tots els ítems amb valoració ≥ 4.5 de les valoracions dels usuaris al cluster que conté l'usuari a ser recomanat
- Sobre aquest conjunt d'ítems executar k-NN, amb el procediment descrit a l'apartat ContentBased-Filtering.
- Retornar els Q items amb major valoració estimada.

6.7 Controlador Persistencia

Aquesta classe usa com a estructura de dades la classe ArrayList. Aquesta és usada per a passar i rebre les dades cap a capes superiors, normalment en forma de Matriu bidimensional de Strings, tot i que en algun cas s'opera amb matrius unidimensionals i tridimensionals. Per a la implementació d'algunes funcions, es considera la primera fila de l'array bidimensional la capçalera de la taula.

L'atribut de la classe estat emmagatzema 6 atributs que usen classes del domini, i manté la informació fins a l'hora de guardar-la usant una ArrayList. També usa una altra estructura proporcionada per java: la classe File. Aquesta permet obtenir informació sobre els arxius i directoris de l'ordinador, crear-los i eliminar-los.

Respecte als algorismes usats, aquesta classe no n'usa cap llevat de recorreguts sencers de taules.

6.8 Controlador Load

Aquesta classe usa com a estructura de dades la classe ArrayList, la classe File i la classe FileReader. ArrayList és usat per a mantenir en una taula bidimensional les dades que es van llegint en forma de strings. La classe File és usada com a input de la funció per saber de quin arxiu s'ha de llegir les dades. La classe FileReader instància un nou objecte d'aquest tipus a partir de la File, i ens permet fer operacions de lectura de l'arxiu obert.

Aquesta classe no usa cap algorisme llevat del recorregut de l'arxiu que es llegeix. Se suposa un arxiu en format .csv i es fa un recorregut de taula.

6.9 Controlador Save

Aquesta classe usa com a estructura de dades la classe ArrayList, la classe File i la classe FileWriter. ArrayList és usat per a mantenir en una taula bidimensional les dades que es passen per paràmetre a la funció principal de guardat. La classe File és usada com a input de la funció per saber a quin arxiu s'ha d'escriure les dades. La classe FileWriter instància un nou objecte d'aquest tipus a partir de la File, i ens permet fer operacions d'escriptura de l'arxiu obert.

Aquesta classe no usa cap algorisme llevat del recorregut de la taula. Es guarden els valors de la taula en format .csv.

6.10 Usuari

Usuari empra l'estructura de dades ConjuntRecomanacions que es tracta d'una extensió d'ArrayList. No empra cap algorisme.

6.11 Conjunt Usuaris

Aquesta classe es tracta d'una extensió de l'estructura de dades ArrayList. Addicionalment, donat que el conjunt d'usuaris es troba ordenat per ID hem implementat l'algorisme BinarySearch, que ens permet cercar un usuari en cost asimptòtic logarítmic en comptes del cost lineal que té la cerca per defecte de l'ArrayList.

6.12 Recomanació

La classe recomanació posseeix un Usuari, un Item i un float opcional Valoració. No empra cap altra estructura de dades ni algorisme.

6.13 Conjunt Recomanacions

Aquesta classe es tracta d'una extensió de l'estructura de dades ArrayList, ordenat primer per ID ítem i després per ID d'usuari. Ho hem decidit així, ja que ja podem accedir a les valoracions d'un usuari fent usuari.getValoracions().

Addicionalment, donat que el conjunt d'usuaris es troba ordenat, hem implementat l'algorisme BinarySearch, que ens permet cercar una recomanació en cost asimptòtic logarítmic en comptes del cost lineal que té la cerca per defecte de l'ArrayList.

6.14 Ítem

La classe Ítem disposa de 3 ArrayList estàtiques, que representen les capçaleres dels atributs, els tipus dels atributs i els pesos dels atributs. També disposa d'una ArrayList privada amb els valors dels atributs. Addicionalment fa servir l'estructura de dades Enum que representa els tipus que pot tenir un atribut.

6.15 Conjunt d'Ítems

La classe ConjuntItems es tracta d'una extensió de la classe ArrayList. Donat que el conjunt d'ítems es troba ordenat per ID hem implementat l'algorisme BinarySearch, que ens permet cercar un ítem en cost asimptòtic logarítmic en comptes del cost lineal que té la cerca per defecte de l'ArrayList.

Aquesta classe també disposa d'un algorisme per calcular la similitud entre dos Ítems, que invoca una funció per calcular la similitud entre dos atributs. Aquests algoritmes estan descrits a ContentBasedFiltering.