

Distribució de productes a un supermercat

Entrega 1
Versió del lliurament: 1.0

PROP - QT2425

Subgrup 21.1

Roger Cot Londres: roger.cot@estudiantat.upc.edu

Nadia Khier: nadia.khier@estudiantat.upc.edu

David Mas Escudé: david.mas@estudiantat.upc.edu

David Sanz Martínez: david.sanz.martinez@estudiantat.upc.edu

| | |
|---|-----------|
| 1. Diagrama casos d'ús..... | 3 |
| 1.1. Explicació dels casos d'ús..... | 4 |
| 2. Diagrama del model conceptual..... | 17 |
| 2.1. Descripció model conceptual..... | 17 |
| 2.2. Descripció de les classes..... | 17 |
| 2.2.1. User..... | 17 |
| 2.2.2. CjtUsers..... | 18 |
| 2.2.3. LlistaProductes..... | 19 |
| 2.2.4. CjtLlistesProductes..... | 20 |
| 2.2.5. Distribucion..... | 21 |
| 2.2.6. CjtDistribuciones..... | 22 |
| 2.2.7. AlgoritmoVoraz..... | 23 |
| 2.2.8. Aproximation_Kruskal_ILS..... | 23 |
| 2.2.9. menuController..... | 24 |
| 2.2.10. UserController..... | 24 |
| 2.2.11. listController..... | 24 |
| 2.2.12. distributionController..... | 24 |
| 3. Relació de les classes implementades per membre de l'equip..... | 25 |
| 4. Estructures de dades i algorismes..... | 25 |
| 4.1. Estructures de dades..... | 25 |
| 4.2. Algorismes..... | 28 |
| 4.2.1. Algoritme voraç..... | 28 |
| 4.2.2. Algoritme d'aproximació..... | 28 |

1. Diagrama casos d'ús

A continuació es presenta el diagrama de casos d'ús, que es troba també a DOCS/DiagramaCasosdUs.jpg



1.1. Explicació dels casos d'ús

CAS 1

Nom: Tancar simulador

Actor: Usuari

Comportament:

1. L'usuari es troba en la pàgina principal del sistema.
2. De les opcions que es mostren al menú, escolleix tancar el simulador.
3. Es tancarà el programa i no es guardarà cap llista ni distribució de productes.

Errors possibles i cursos alternatius:

2a) L'usuari escolleix una altra opció. Si no existeix, el sistema avisa a l'usuari i torna a mostrar el menú.

3a) No es tanca el simulador o es queda penjat.

CAS 2

Nom: Afegir llista de productes

Actor: Usuari

Comportament:

1. L'usuari es troba en la pàgina principal del sistema.
2. De les opcions que es mostren al menú, escolleix afegir una nova llista de productes.
3. L'usuari escolleix com vol introduir la llista o sortir al menu principal:
 - 3.1. Via terminal:
 - 3.1.1. L'usuari introdueix els productes en el format demanat.
 - 3.1.2. L'usuari introdueix els graus de similitud en el format demanat.
 - 3.2. Via fitxer:
 - 3.2.1. L'usuari introdueix el nom del fitxer.
 - 3.3. Sortir:
4. L'usuari escull si vol guardar o descartar la llista.
 - 4.1. Guardar
 - 4.1.1. L'usuari introdueix el nom amb el que vol guardar la llista.
 - 4.1.2. Es guarda la llista amb el nom introduït.
 - 4.2. Descartar
 - 4.2.1. El sistema surt del menú d'opcions i mostra el menú principal.

Errors possibles i cursos alternatius:

2a) L'usuari escolleix una altra opció. Si no existeix, el sistema avisa a l'usuari i torna a mostrar el menú.

3a) L'usuari escolleix una opció inexistente. El sistema notifica a l'usuari de que l'opció que ha indicat es invalida i torna a mostrar les opcions d'input de llistes.

3.1.1a), 3.1.2a) i 3.2.1a) L'usuari no respecta el format demanat. Un missatge l'avisava i pot tornar a intentar-ho. (Inclou el cas en el que el Grau de Similitud no esta entre 0 i 1).

3.2.1b) No es troba el fitxer amb aquest nom. El sistema dóna error i torna al menu principal.

3.1.2b) L'usuari està definint el Grau de Similitud d'un producte que no existeix. El sistema avisa a l'usuari de l'error.

4a) L'usuari escull una opció invalida. El sistema avisa de l'error i torna a mostrar el menú d'opcions de guardat.

- 4.1.2a) Ja existeix una llista amb aquest nom. L'usuari és avisat i el pot tornar a introduir.
- 4.1.2b) L'usuari ha introduït un camp buit. El sistema l'avisat i ha de tornar a introduir un nom.

CAS 3

Nom: Gestió llistes de productes

Actor: Usuari

Comportament:

1. L'usuari es troba en la pàgina principal del sistema
2. L'usuari selecciona l'opció de gestionar llistes de productes
3. L'usuari pot realitzar diferents accions, i escolleix una: Des del menú de gestió de llistes de productes, l'usuari pot seleccionar l'opció de:
 - **Modificar llista de productes:** Permet modificar una de les llistes de productes ja existents i guardar-la.
 - **Visualitzar llista de productes:** Permet veure tots els productes que conté una llista de productes ja creada.
 - **Esborrar llista de productes:** Permet esborrar una llista de productes.
 - **Volver al menú principal:** El sistema redirige al usuari al menú principal

Errors possibles i cursos alternatius:

- 2a) L'usuari escolleix una altra opció. Si no existeix, el sistema avisa a l'usuari i torna a mostrar el menú.
- 3a) No hi ha llistes que gestionar. El sistema l'avisat i torna al menú principal.
- 3b) L'usuari introdueix una opció invàlida. El sistema l'avisat i li torna a demanar una opció vàlida.

CAS 4

Nom: Modificar llista de productes

Actor: Usuari

Comportament:

1. L'usuari es troba a la gestió de llistes i escolleix modificar una llista de productes.
2. L'usuari indica el nom de la llista de productes que vol modificar.
3. El sistema ofereix 4 accions:
 - **Afegir productes.**
 - **Eliminar productes.**
 - **Modificar el grau de similitud dels productes.**
 - **Canviar el nom de la llista.**
4. L'usuari escolleix una opció i segueix les instruccions sobre el format per efectuar la modificació.
5. El sistema guardarà els canvis.

Errors possibles i cursos alternatius:

- 1a) Indica una opció invàlida. El sistema avisa i torna a mostrar al menú de gestió.
- 3a) La llista amb aquest nom no existeix. El sistema avisa i torna a preguntar per el nom de la llista.
- 4a) Escolleix una acció incorrecta. El sistema avisa i torna al menú de gestió de llistes.

4b) Provoca un error dels casos d'ús: **(9: Afegir productes a una llista), (10: Eliminar productes d'una llista), (11:Gestió grau de similitud), (8: Canviar nom llista de productes)**

CAS 5

Nom: Visualitzar llista de productes

Actor: Usuari

Comportament:

1. L'usuari es troba al menu de gestió de llistes on es mostren els noms de totes les llistes de producte.
2. L'usuari indica que vol visualitzar una llista.
3. El sistema demana a l'usuari el nom de la llista que vol visualitzar.
4. L'usuari introdueix el nom de la llista que vol veure.
5. El sistema mostra la llista de productes. També apareixen els graus de similitud si n'hi ha.

Errors possibles i usos alternatius:

- 2a) Indica una opció invàlida. El sistema avisa i torna a mostrar al menú de gestió.
- 5a) La llista amb aquest nom no existeix. El sistema avisa a l'usuari i torna a preguntar per el nom de la llista.

CAS 6

Nom: Esborrar llista de productes

Actor: Usuari

Comportament:

1. L'usuari es troba al menu de gestió de llistes on es mostren els noms de totes les llistes de producte.
2. L'usuari indica que vol esborrar una llista.
3. El sistema demana a l'usuari el nom de la llista que vol esborrar.
4. L'usuari introdueix el nom de la llista que vol esborrar.
5. El sistema demana confirmació a l'usuari:
El sistema mostra un missatge de confirmació indicant que l'acció és irreversible i preguntant si l'usuari està segur de voler esborrar la llista.
6. L'usuari confirma o cancel·la l'acció:
 - 6.1. Si l'usuari confirma l'acció, el sistema procedeix a esborrar la llista de productes seleccionada.
 - 6.2. Si l'usuari cancel·la l'acció, la llista no es modifica i torna al menu de gestió de llistes.
7. El sistema notifica l'usuari:
Un cop esborrada la llista, el sistema informa l'usuari que la llista ha estat eliminada amb èxit i torna al menu de gestió de llistes.

Errors possibles i cursos alternatius:

- 2a) Indica una opció invàlida. El sistema avisa i torna a mostrar el menú de gestió.
- 5a) La llista amb aquest nom no existeix. El sistema avisa a l'usuari i torna a preguntar per el nom de la llista.

5b) La llista indicada és l'única llista en el conjunt de llistes. El sistema mostra un missatge informant a l'usuari.

6a) L'usuari indica una opció invàlida. El sistema avisa a l'usuari i torna a preguntar si vol o no eliminar la llista permanentment.

7a) Si l'usuari ha cancel·lat l'acció, la llista no s'ha esborrat.

7b) Si la llista eliminada era l'única que quedava ja no torna al menú de gestió de llistes sino al menú principal.

CAS 7

Nom: Canviar nom llista de productes

Actor: Usuari

Comportament:

1. L'usuari es troba en el **Pas 3) CAS 4: Modificar llista de productes**.
2. L'usuari indica que vol canviar el nom d'una llista.
3. El sistema demana a l'usuari el nou nom per la llista.
4. L'usuari introdueix el nou nom de la llista.
5. El sistema indica que el canvi s'ha efectuat i mostra el canvi de nou antic a nou nom.

Errors possibles i cursos alternatius:

2a) Indica una opció invàlida. El sistema avisa i torna a mostrar al menú de gestió.

5a) El nou nom és igual a l'antic, el sistema l'avisa perquè l'usuari n'introdueixi un altre.

5b) El nou nom ja existeix a una altra llista, el sistema l'avisa perquè l'usuari n'introdueixi un altre.

CAS 8

Nom: Afegir productes a una llista

Actor: Usuari

Comportament:

1. L'usuari es troba en el **Pas 3) CAS 4: Modificar llista de productes**.
2. L'usuari selecciona l'opció d'afegir productes.
3. L'usuari introdueix els productes en el format que demana el sistema.
4. S'afegeixen els productes demanats i el sistema retorna al menú de gestió de llistes.

Errors possibles i cursos alternatius:

2a) Indica una opció invàlida. El sistema avisa i torna a mostrar al menú de gestió.

3a) L'usuari tracta d'afegir un producte que ja està a la llista, això no modifica la llista i el sistema avisa a l'usuari dels productes repetits.

3b) L'usuari no introdueix cap producte i deixa el camp en blanc. El sistema mostra a l'usuari que ha de insertar almenys un producte per poder continuar i torna a preguntar per la inserció de productes.

CAS 9

Nom: Eliminar productes d'una llista

Actor: Usuari

Comportament:

1. L'usuari es troba en el **Pas 3) CAS 4: Modificar llista de productes**.
2. L'usuari selecciona l'opció d'eliminar productes.
3. L'usuari introdueix els productes en el format que demana el sistema.

4. S'elimina el/s producte/s i els seus graus de similitud amb els altres productes.

Errors possibles i cursos alternatius:

- 2a) Indica una opció invàlida. El sistema avisa i torna a mostrar al menú de gestió.
- 3a) L'usuari tracta d'esborrar un producte que no és a la llista, això no modifica la llista, el sistema avisa a l'usuari i torna al menú de gestió de llistes.
- 4a) L'usuari ha introduït tots els productes de la llista. El sistema avisa a l'usuari que la llista quedaria buida, i pregunta si vol eliminar la llista per complet. Si escull:
 - **No:** Torna al menú de gestió de llistes.
 - **Si:** El sistema executa el **Pas 5) CAS 7: Esborrar llista de productes.**
 - **Opció invàlida:** El sistema torna a preguntar per una opció vàlida.

CAS 10

Nom: Gestió grau de similitud

Actor: Usuari

Comportament:

1. L'usuari es troba en el **Pas 3) CAS 4: Modificar llista de productes.**
2. L'usuari selecciona l'opció de gestionar els graus de similitud dels productes.
3. El sistema indica a l'usuari el format d'entrada.
4. L'usuari introdueix els nous graus de similitud entre productes.
5. El sistema guarda els canvis i mostra una taula amb les columnes:
 - **Producte 1:** nom
 - **Producte 2:** nom
 - **GdS Antic:** valor antic
 - **GdS Nou:** valor nou

Errors possibles i cursos alternatius:

- 2a) Indica una opció invàlida. El sistema avisa i torna a mostrar al menú de gestió.
- 5a) L'usuari no respecta el format demanat. Un missatge l'avisa i pot tornar a intentar-ho. (Inclou el cas en el que el Grau de Similitud no està entre 0 i 1).
- 5b) L'usuari està definint el Grau de Similitud d'un producte que no existeix. El sistema avisa a l'usuari de l'error.

CAS 11

Nom: Generar nova distribució

Actor: Usuari

Comportament:

1. L'usuari es troba en la pàgina principal del sistema.
2. L'usuari selecciona l'opció de generar una nova distribució.
3. El sistema demana de quina llista es vol generar la distribució.
4. L'usuari introdueix el nom de la llista de la qual vol generar la distribució.
5. El sistema demana a l'usuari les dimensions de la prestatgeria i dona a escollir com vol introduir les dimensions de la prestatgeria:
 - 5.1. Escriure dimensions manualment.
 - 5.1.1. L'usuari introdueix les dimensions en el format demanat.
 - 5.2. Ajustament automàtic de dimensions.

6. L'usuari escull l'algorisme amb els paràmetres descrits.
7. El sistema genera la distribució i l'usuari té l'opció de:
 - 7.1. Guardar-la
 - 7.2. Guardar-la i exportar-la (**CAS 19**)
 - 7.3. Descartar-la

Errors possibles i cursos alternatius:

- 2a) L'usuari escolleix una altra opció. Si no existeix, el sistema avisa a l'usuari i torna a mostrar el menú.
- 3a) No hi ha llistes guardades. El sistema notifica de l'error i torna al menú principal.
- 4a) Introdueix el nom d'una llista que no existeix. El sistema notifica de l'error i torna al menú principal.
- 5a) L'usuari selecciona una opció invàlida. El sistema avisa a l'usuari i torna a demanar que seleccioni una opció.
 - 5.1.1a) Les dimensions de la prestatgeria són massa petites per al nombre de productes de la llista. El sistema informa de l'error i deixa tornar a intentar.
 - 5.1.1b) No es respecta el format d'entrada. El sistema notifica de l'error i torna a demanar les dimensions.
- 6a) L'usuari selecciona una altra opció. Si no existeix, el sistema avisa a l'usuari i torna a preguntar quin algorisme vol executar.
- 7a) L'usuari selecciona una opció invàlida. El sistema avisa a l'usuari i torna a preguntar com vol procedir amb la distribució generada.

CAS 12

Nom: Gestió de distribucions

Actor: Usuari

Comportament:

1. L'usuari accedeix a la gestió de distribucions a través del menú principal.
2. L'usuari selecciona una de les opcions que ofereix el menú. Des del menú de gestió de distribucions, l'usuari pot:
 - **Modificar distribució:** Accedir a distribucions existents per ajustar els productes o ubicacions associats.
 - **Visualitzar distribució:** Consultar les distribucions creades per veure els detalls.
 - **Esborrar distribució:** L'usuari pot eliminar distribucions que ja no són necessàries
 - **Tornar al menú principal:**

Errors possibles i cursos alternatius:

- 1a) L'usuari escolleix una altra opció. Si no existeix, el sistema avisa a l'usuari i torna a mostrar el menú.
- 1b) L'usuari no té cap distribució guardada, el sistema avisa a l'usuari i torna a imprimir el menú principal.
- 2a) L'usuari selecciona una opció invàlida. El sistema avisa a l'usuari i torna a demanar que seleccioni una opció.

CAS 13

Nom: Visualitzar distribució

Actor: Usuari

Comportament:

1. L'usuari selecciona la opció de visualitzar una distribució.
2. El sistema demana el nom de la distribució que es vol visualitzar.
3. L'usuari introdueix el nom de la distribució.
4. El sistema mostra la distribució dels productes en la prestatgeria amb un format visual, junt amb quin algoritme s'ha emprat.

Errors possibles i cursos alternatius:

- 1a) L'usuari selecciona una opció invàlida. El sistema avisa a l'usuari i torna a demanar que seleccioni una opció.
- 4a) El nom introduït per l'usuari no es correspon amb cap distribució. El sistema notifica de l'error i torna a demanar un nom.

CAS 14

Nom: Esborrar distribució

Actor: Usuari

Comportament:

1. L'usuari selecciona esborrar una distribució.
2. El sistema demana el nom de la distribució que es vol esborrar.
3. L'usuari introdueix el nom.
4. El sistema demana confirmació.
5. Si l'usuari ho confirma, el sistema elimina la distribució i torna al menú de gestió de distribucions.

Errors possibles i cursos alternatius:

- 1a) L'usuari selecciona una opció invàlida. El sistema avisa a l'usuari i torna a demanar que seleccioni una opció.
- 4a) El nom introduït per l'usuari no es correspon amb cap distribució. El sistema notifica de l'error i torna a demanar un nom.
- 4b) A més, si només té una distribució, el sistema a l'usuari l'avisava de que aquella distribució és l'única que té.
- 5a) L'usuari no ho confirma i torna al menú de gestió de distribucions.
- 5b) L'usuari ha confirmat la eliminació de la seva última distribució i per tant torna al menú principal.

CAS 15

Nom: Modificar distribució

Actor: Usuari

Comportament:

1. L'usuari selecciona l'opció de modificar una distribució.
2. El sistema demana el nom de la distribució que es vol modificar.
3. L'usuari introdueix el nom de la distribució.
4. El sistema ofereix a l'usuari:

- **Moure productes:**
- **Eliminar productes:**
- **Canviar el nom de la distribució:**
- **Tornar al menú de gestió de productes**

Errors possibles i cursos alternatius:

- 1a) L'usuari selecciona una opció invàlida. El sistema avisa a l'usuari i torna a demanar que seleccioni una opció.
- 4a) El nom introduït per l'usuari no es correspon amb cap distribució. El sistema notifica de l'error i torna al menú de gestió de distribució.

CAS 16

Nom: Moure productes de la distribució

Actor: Usuari

Comportament:

1. L'usuari selecciona l'opció de moure productes, desde el menú de gestió de distribucions.
2. El sistema mostra la distribució seleccionada en el **pas 2) del cas 15: Modificar distribució**, junt amb el algorisme emprat per a generar-lo i demana a l'usuari el nom de dos productes.
3. El usuari introdueix el nom dels productes que canviaran de posició.
4. El sistema canvia la posició dels productes, imprimeix la distribució canviada i torna al menú de gestió de distribucions.

Errors possibles i cursos alternatius:

- 1a) L'usuari escolléix una altra opció. Si no existeix, el sistema avisa a l'usuari i torna a mostrar el menú.
- 4a) L'usuari ha introduït un (o dos) producte(s) inexistent(s) o un format incorrecte, el sistema avisa de l'error i torna a preguntar a l'usuari.

CAS 17

Nom: Eliminar productes de la distribució

Actor: Usuari

Comportament:

1. L'usuari selecciona l'opció d'eliminar productes de la distribució, desde el menú de gestió de distribucions.
2. El sistema mostra la distribució seleccionada en el **pas 2) del cas 15: Modificar distribució**, junt amb el algorisme emprat per a generar-lo i demana a l'usuari el nom dels productes que han de ser eliminats.
3. L'usuari introdueix els noms en el format demanat.
4. El sistema confirma l'eliminació dels productes imprimint-los, imprimeix la distribució canviada i torna al menú de gestió de distribucions.

Errors possibles i cursos alternatius:

- 1a) L'usuari escolléix una altra opció. Si no existeix, el sistema avisa a l'usuari i torna a mostrar el menú.

4a) L'usuari ha introduït un producte inexistent en la distribució. El sistema avisa a l'usuari. Si no ha produït ningun canvi, no imprimeix la distribució y deixa tornar a intentar a l'usuario.

4b) L'usuari ha introduït tots els productes restants de la distribució, el sistema avisa a l'usuari que això no és possible i li pregunta si vol eliminar la distribució.

- **No:** Torna al menu de gestió de distribucions.
- **Si:** El sistema executa el **Pas 4) CAS 14: Esborrar distribucions.**
- **Opció invalida:** El sistema torna a preguntar per una opció valida.

CAS 18

Nom: Canviar el nom de la distribució

Actor: Usuari

Comportament:

1. L'usuari selecciona l'opció de canviar el nom de la distribució, desde el menú de gestió de distribucions.
2. El sistema demana a l'usuari que introdueixi el nou nom de la distribució.
3. L'usuari escriu el nou nom de la distribució.
4. El sistema accepta i torna al menu de gestió de distribucions.

Errors possibles i cursos alternatius:

1a) L'usuari escolleix una altra opció. Si no existeix, el sistema avisa a l'usuari i torna a mostrar el menú.

4a) El nom ha escrit un nom ja existent (potser el mateix nom que la distribució que està editant), el sistema avisa de l'error i torna preguntar per a un nou nom de la distribució.

CAS 19

Nom: Exportar distribució

Actor: Usuari

Comportament:

1. L'usuari ha seleccionat el pas **7.2) CAS 11: Generar nova distribució.**
2. El sistema demana el nom amb el que es guardarà la distribució.
3. L'usuari introdueix el nom.
4. El sistema guardarà la distribució en el sistema.
5. El sistema exportarà la distribució amb el nom indicat per l'usuari i afegirà el nom de l'usuari al final (per evitar repeticions entre usuaris). Es guardara en un directori outputs prèviament creat.

Errors possibles i cursos alternatius:

4a) L'usuari ha introduït un nom de distribució ja existent. El sistema l'avisa i demana que introdueixen un nom diferent.

4b) L'usuari no ha introduït res. El sistema l'avisa i demana que introdueixen un nom.

5b) El directori outputs no existeix. El sistema dona error d'escriptura. I demana a l'usuari que crei el directori o que abandoni el guardat de la distribució.

CAS 20

Nom: Registrar usuari

Actor: Usuari

Comportament:

1. El sistema mostra a l'usuari el menú d'accés al sistema.
2. L'usuari selecciona l'opció de registre.
3. El sistema mostra el camp de registre:
 - Nom d'usuari
 - Nom de la persona
 - Contrasenya
4. L'usuari completa cadascun dels camps.
5. El sistema accepta els valors introduïts i mostra el menú principal de l'aplicació.

Errors possibles i cursos alternatius:

2a) L'usuari escolleix una altra opció. Si no existeix, el sistema avisa a l'usuari i torna a mostrar el menú d'accés al sistema.

5a) L'usuari ha introduït un username ja existent. El sistema avisa de l'error i tornar a mostrar els camps de registre.

5b) L'usuari ha deixat en blanc algun dels camps. El sistema demana a l'usuari que completi el camp.

CAS 21

Nom: Inici de sessió

Actor: Usuari

Comportament:

1. El sistema mostra a l'usuari el menú d'accés al sistema.
2. L'usuari selecciona l'opció d'inici de sessió.
3. El sistema mostra el camp de registre:
 - Nom d'usuari
 - Contrasenya
4. L'usuari completa cadascun dels camps.
5. El sistema accepta els valors introduïts i mostra el menú principal de l'aplicació.

Errors possibles i cursos alternatius:

2a) L'usuari escolleix una altra opció. Si no existeix, el sistema avisa a l'usuari i torna a mostrar el menú d'accés al sistema.

5a) El username indicat per l'usuari no està registrat en l'aplicació. El sistema avisa de l'error i tornar a mostrar el menú d'accés.

5b) L'usuari ha introduït una contrasenya incorrecta d'un usuari registrat. El sistema avisa de l'error i tornar a mostrar el menú d'accés.

CAS 22

Nom: Abandonar la sessió (Logout)

Actor: Usuari

Comportament:

1. L'usuari es troba en el menú principal.
2. L'usuari selecciona l'opció de logout.
3. El sistema torna al menú d'accés a l'aplicació.

Errors possibles i cursos alternatius:

2a) L'usuari escolleix una altra opció. Si no existeix, el sistema avisa a l'usuari i torna a mostrar el menú.

CAS 23

Nom: Gestionar missatges

Actor: Usuari

Comportament:

1. L'usuari es troba en la pàgina principal del sistema.
2. De les opcions que es mostren al menú, escolleix entrar en el registre de missatges personal.
3. El sistema imprimeix la bústia d'entrada junt amb els missatges rebuts, classificats en nous i llegits, i seguidament els missatges enviats per l'usuari.
4. El sistema demana a l'usuari que seleccioni una de les següents opcions:
 - 4.1. **Enviar un missatge:** L'usuari envia un missatge a un altre usuari.
 - 4.2. **Seleccionar un missatge:** L'usuari selecciona un missatge per posteriorment veure'l.
 - 4.3. **Sortir:** Surt al menú.
5. L'usuari escull una opció de les anteriors.

Errors possibles i cursos alternatius:

- 2a) L'usuari escolleix una altra opció. Si no existeix, el sistema avisa a l'usuari i torna a mostrar el menú.
- 3a) L'usuari no té missatges rebuts o missatges enviats, o ambdós i el sistema no imprimeix cap missatge (rebut, enviat o ambdós).
- 5a) L'usuari selecciona una opció invàlida. El sistema avisa i torna a mostrar les opcions disponibles de gestió de missatges.

CAS 24

Nom: Enviar missatges

Actor: Usuari

Comportament:

1. L'usuari ha seleccionat l'opció **d'enviar un missatge**.
2. El sistema demana a l'usuari que introdueixi el nom d'un usuari registrat l'aplicació.
3. L'usuari introdueix un nom d'usuari.
4. El sistema demana a l'usuari que seleccioni una de les següents opcions:
 - 4.1. **Enviar llista de productes:**
 - 4.1.1. El sistema mostra al llista de productes de l'usuari i demana a l'usuari que introdueixi un nom.
 - 4.1.2. L'usuari introdueix el nom d'una llista i el sistema envia la llista seleccionada a l'usuari prèviament definit.
 - 4.2. **Enviar distribució de productes:**
 - 4.2.1. El sistema mostra la distribució de productes de l'usuari i demana a l'usuari que introdueixi un nom.
 - 4.2.2. L'usuari introdueix el nom d'una distribució i el sistema envia la distribució seleccionada a l'usuari prèviament definit.
 - 4.3. **Tornar:** Torna al menú de gestió de missatges.

Errors possibles i cursos alternatius:

- 2a) El sistema torna a mostrar el menú de gestió de missatges perquè no hi ha més usuaris registrats apart d'ell.

3a) L'usuari introdueix un nom incorrecte:

- El nom d'usuari no coincideix amb cap dels usuaris registrats, el sistema avisa a l'usuari i mostra una llista amb els possibles usuaris a enviar el missatge, seguidament torna a demanar a l'usuari que introdueixi un nom d'usuari.

- El nom d'usuari es ell mateix, el sistema avisa a l'usuari i mostra una llista amb els possibles usuaris a enviar el missatge, seguidament torna a demanar a l'usuari que introdueixi un nom d'usuari.

4a) L'usuari selecciona una opció invàlida. El sistema avisa i torna a demanar a l'usuari que seleccioni una de les opcions.

4.1a) L'usuari no té cap llista de productes guardada per enviar, el sistema l'avis i torna al menú de gestió de missatges.

4.2a) L'usuari no té cap distribució guardada per enviar, el sistema l'avis i torna al menú de gestió de missatges.

4.1.2a) El nom introduït per l'usuari no coincideix amb ninguna llista, el sistema l'avis i torna a demanar un nom.

4.2.2a) El nom introduït per l'usuari no coincideix amb ninguna distribució, el sistema l'avis i torna a demanar un nom.

CAS 25

Nom: Seleccionar missatges

Actor: Usuari

Comportament:

1. L'usuari ha seleccionat l'opció de **seleccionar un missatge**.
2. El sistema demana a l'usuari escriure l'índex d'un dels missatges que es mostra en el menú de gestió de missatges o escriure 0 per tornar al menú de gestió de missatges.
3. L'usuari introdueix el número del missatge que vol seleccionar.
4. El sistema mostra a l'usuari les següents opcions:
 - 4.1. Si el missatge seleccionat era un missatge rebut, el sistema imprimeix el contingut del missatge en pantalla juntament amb informació (remitent, destinatari, tipus de contingut i data) i pregunta a l'usuari si vol **guardar el missatge (Cas 26)**.
 - 4.2. Si el missatge seleccionat era un missatge enviat, el sistema imprimeix el contingut del missatge en pantalla juntament amb informació (remitent, destinatari, tipus de contingut i data) i torna a la **gestió de missatges (Cas 23)**.

Errors possibles i cursos alternatius:

2a) L'usuari selecciona una opció invàlida. El sistema avisa i torna a demanar a l'usuari que seleccioni una de les opcions.

4a) L'usuari ha escrit 0, per tant, torna al menú de gestió de missatges.

4b) L'usuari escriu l'índex d'un missatge inexistent o en un format incorrecte. El sistema avisa de l'error i torna al menú de gestió de missatges.

CAS 26

Nom: Guardar missatges

Actor: Usuari

Comportament:

1. L'usuari ha seleccionat un missatge rebut.
2. El sistema ha imprés el missatge juntament amb la opció de guardar o no guardar-lo. Mostra a l'usuari ambdues opcions.
3. L'usuari selecciona una opció.
 - 3.1. L'usuari selecciona la opció de guardar el contingut del missatge.
 - 3.1.1. El contingut del missatge (una llista o una distribució) s'afegeix a la biblioteca de llistes o distribucions de l'usuari. Es guarda amb el nom amb el qual s'ha rebut.
 - 3.2. L'usuari selecciona No guardar el contingut del missatge. El sistema torna a mostrar el menú de gestió de missatges.

Errors possibles i cursos alternatius:

- 2a) Errors possibles i cursos alternatius: 4a) i 4b) del **CAS 25: Seleccionar missatges**.
- 2b) L'usuari ha seleccionat un missatge enviat. El sistema només mostra el missatge i torna al menú de gestió de missatges.
- 3a) L'usuari selecciona una opció invàlida. El sistema avisa a l'usuari i torna a demanar a l'usuari que seleccioni una de les opcions.
- 3.1.1a) Si la biblioteca de l'usuari ja té una instància amb aquest nom, s'afegeix una cadena de caràcters "(X)" com a sufix del nom, on "X" és un índex numèric per diferenciar la nova instància de l'objecte respecte a les anteriors.

- **String name:** Nom real de l'usuari.
- **String password:** Contrasenya de l'usuari per accedir al sistema.
- **int mensajesNuevos:** Nombre de missatges nous sense llegir a la safata d'entrada.
- **List<Mensaje> inbox:** Llista de missatges rebuts per l'usuari.
- **List<Mensaje> sentMessages:** Llista de missatges enviats per l'usuari.
- **CjtDistribuciones cjtDistribuciones:** Conjunt de distribucions associades a l'usuari.
- **CjtLlistesProductes cjtLlistesProductes:** Conjunt de llistes de productes associades a l'usuari.

Funcionalitats:

- Gestiona i actualitza els missatges rebuts i enviats: permet afegir missatges a la safata d'entrada o a la llista de missatges enviats.
- Proporciona accés als atributs com el nom d'usuari, el nom real, la contrasenya, les llistes de productes i les distribucions.
- Controla el comptador de missatges nous, que es pot consultar o restablir.
- Ofereix una representació textual de la informació bàsica de l'usuari.

2.2.2. CjtUsers

Aquesta classe té una instància única.

S'encarrega de gestionar un conjunt d'usuaris registrats a l'aplicació. Proporciona funcionalitats per afegir, eliminar i consultar usuaris, així com per verificar la presència d'un usuari en el conjunt. Aquesta classe és una implementació única que actua com a contenidor de tots els usuaris.

Atributs:

- **Map<String, User> usersList:** Mapa que associa el nom d'usuari (**username**) amb l'objecte **User** corresponent. Permet l'accés eficient als usuaris registrats.

Funcionalitats:

- **boolean containsUser(String username):** Verifica si un usuari amb un nom específic existeix al conjunt.
- **void addUser(User user):** Afegeix un usuari al conjunt. El nom d'usuari ha de ser únic.
- **User getUser(String username):** Obté l'usuari associat a un nom d'usuari específic. Retorna **null** si no existeix.
- **int getSize():** Retorna el nombre total d'usuaris al conjunt.
- **List<String> getAllUsernames():** Retorna una llista amb tots els noms d'usuaris registrats.
- **boolean isEmpty():** Indica si el conjunt d'usuaris està buit.

Descripció general:

Aquesta classe centralitza la gestió dels usuaris en un únic conjunt. Permet operacions bàsiques per mantenir i consultar els usuaris registrats, assegurant un accés ràpid gràcies a

l'ús d'un mapa. Es pot utilitzar en controladors per gestionar les accions relacionades amb la sessió i la interacció entre usuaris.

2.2.3. LlistaProductes

S'encarrega de gestionar una llista de productes i una matriu de similitud entre ells. Proporciona funcionalitats per manipular la llista, establir i obtenir graus de similitud entre productes, i verificar la seva existència.

Atributs:

- **String name:** Nom de la llista de productes.
- **Map<String, Map<String, Float>> similarityMatrix:** Matriu que representa els graus de similitud entre els productes. Cada clau és el nom d'un producte, i el valor associat és un altre mapa que enllaça altres productes amb els seus graus de similitud.

Funcionalitats:

- **Gestió de la llista de productes:**
 - **Set<String> getList():** Retorna el conjunt de noms dels productes registrats a la llista.
 - **boolean insertarProducte(String product):** Afegeix un producte a la llista si no està registrat. Retorna **true** si s'ha afegit correctament o **false** si ja existeix.
 - **void eliminarProducte(String product):** Elimina un producte de la llista, incloent-hi totes les seves associacions de similitud.
- **Gestió de la matriu de similitud:**
 - **Float getSimilarity(String prod1, String prod2):** Obté el grau de similitud entre dos productes. Retorna **null** si algun dels productes no existeix.
 - **void setSimilarity(String prod1, String prod2, float GdS):** Estableix el grau de similitud entre dos productes. Actualitza la matriu en ambdós sentits (prod1 → prod2 i prod2 → prod1).
- **Altres funcionalitats:**
 - **boolean productExists(String product):** Verifica si un producte existeix a la llista.
 - **Map<String, Map<String, Float>> getSimilarityMatrix():** Retorna la matriu de similitud completa.
 - **String getName():** Obté el nom de la llista de productes.
 - **void setNom(String n):** Assigna un nom a la llista.

Descripció general:

Aquesta classe és útil per gestionar col·leccions de productes i definir relacions entre ells basades en graus de similitud. Això permet modelar associacions o agrupaments en funció d'aquests valors, essent especialment útil per a aplicacions de recomanació, classificació o cerca.

2.2.4. CjtLlistesProductes

Classe que gestiona un conjunt de llistes de productes. Proporciona funcionalitats per afegir, eliminar i consultar llistes de productes, així com verificar-ne l'existència.

Atributs:

- **Map<String, LlistaProductes> conjuntoListas:**
Mapa que associa un nom de llista amb una instància de la classe **LlistaProductes**.

Funcionalitats:

- Gestió del conjunt de llistes:
 - **int getSize()**
Retorna el nombre total de llistes registrades en el conjunt.
 - **boolean isEmpty()**
Verifica si el conjunt de llistes està buit. Retorna **true** si no hi ha cap llista, i **false** en cas contrari.
 - **Set<String> getListas()**
Retorna un conjunt amb els noms de totes les llistes emmagatzemades al conjunt.
 - **boolean contiene(String name)**
Verifica si existeix una llista amb el nom indicat dins el conjunt.
- Gestió d'elements individuals:
 - **void agregarLista(String name, LlistaProductes lista)**
Afegeix una nova llista de productes al conjunt. El nom de la llista serveix com a clau, i l'objecte **LlistaProductes** és el valor associat.
 - **LlistaProductes obtenerLista(String nombre)**
Retorna l'objecte **LlistaProductes** associat al nom indicat. Si no existeix, retorna **null**.
 - **void eliminarListaDelConjunto(String nombre)**
Elimina del conjunt la llista de productes amb el nom indicat. Si no existeix, no fa res.

Descripció general:

Aquesta classe serveix com a contenidor centralitzat per gestionar múltiples llistes de productes (**LlistaProductes**). És especialment útil en aplicacions que requereixen organitzar i manipular diferents col·leccions de productes per a funcionalitats com comparacions, recomanacions o categoritzacions.

Ús potencial

- Organitzar múltiples categories o grups de productes, com "Electrodomèstics", "Mobles", o "Decoració".
- Facilitar l'accés ràpid a qualsevol llista per nom.

- Mantenir consistència i simplificar l'estructura del codi quan es treballa amb diverses col·leccions de productes.

2.2.5. Distribucion

Classe que representa un conjunt de productes organitzats dins d'una distribució espacial definida per dimensions i gestionada per un algoritme que s'escull amb una *estratègia*. Aquesta classe permet la creació, manipulació i consulta de distribucions.

Atributs:

- **Pair<Integer, Integer> Prestage**
Dimensions de la distribució representades com a parell **<altura, longitud>**.
- **List<String> Dist**
Llista de productes presents a la distribució.
- **String nom**
Nom identificador de la distribució.
- **Estrategia estrategia**
Algoritme utilitzat per definir l'ordre dels productes en la distribució.
- **Map<String, Integer> mapa**
Mapa que associa el nom de cada producte amb la seva posició a la distribució.

Funcionalitats:

Gestió de la distribució:

- **Distribucion(Pair<Integer, Integer> prestige, List<String> dist, String nom)**
Constructor que inicialitza la distribució amb dimensions, productes i nom especificats.
- **Distribucion(int h, int w)**
Constructor que inicialitza una distribució amb dimensions **<altura, longitud>**.

Manipulació de la distribució:

- **void generateDist(Map<String, Map<String, Float>> LlistaProductes)**
Genera la llista de productes en la distribució segons l'algoritme d'estratègia proporcionat. També crea el mapa de posicions dels productes.
- **void editarDist(String nom1, String nom2)**
Intercanvia les posicions de dos productes a la distribució.
- **void eliminarProducte(String nom)**
Elimina un producte de la distribució, tant de la llista com del mapa de posicions.

Descripció general:

La classe **Distribución** permet gestionar de forma dinàmica col·leccions de productes disposats en un espai virtual o físic. És especialment útil en aplicacions de logística, planificació d'espais, o fins i tot jocs i simulacions.

Ús potencial

- Organització de productes en un magatzem o botiga.
- Visualització i optimització d'espais en aplicacions de decoració.
- Simulació de disposicions segons estratègies d'optimització.

2.2.6. CjtDistribuciones

Classe que gestiona un conjunt de distribucions organitzades per noms i representades com a objectes de tipus **Distribucion**. Aquesta classe permet l'administració de distribucions amb funcionalitats com afegir, eliminar, consultar i obtenir noms.

Atributs:

- **Map<String, Distribucion> conjuntoDistribuciones**
Mapa que associa cada distribució amb el seu nom identificador.

Funcionalitats:

Gestió del conjunt de distribucions

- **CjtDistribuciones()**
Constructor que inicialitza un conjunt buit de distribucions.

Consultes generals

- **int getSize()**
Retorna el nombre total de distribucions contingudes en el conjunt.
- **Set<String> getNames()**
Retorna un conjunt amb els noms de totes les distribucions emmagatzemades.
- **boolean contiene(String name)**
Verifica si existeix una distribució amb el nom especificat dins del conjunt.
- **boolean isEmpty()**
Retorna **true** si el conjunt de distribucions està buit; en cas contrari, retorna **false**.

Operacions sobre distribucions

- **void agregarDistribucion(String name, Distribucion distribucion)**
Afegeix una nova distribució al conjunt, identificada pel nom proporcionat.
- **Distribucion obtenerDistribucion(String nombre)**
Retorna l'objecte **Distribucion** associat amb el nom especificat. Si no existeix, retorna **null**.
- **void eliminarDistribucionDelConjunto(String nombre)**
Elimina la distribució identificada pel nom donat del conjunt.

Descripció general:

CjtDistribuciones simplifica la gestió d'un conjunt de distribucions. Cada distribució és únicament identificada pel seu nom i pot ser manipulada de manera eficient gràcies a l'ús d'un mapa (estructura clau-valor).

Ús potencial

- Administració d'una base de dades d'espais decoratius, magatzems o plans de distribució.
- Desenvolupament d'aplicacions que requereixen una gestió eficient de col·leccions d'objectes estructurats.
- Organització i recuperació d'informació d'una col·lecció de configuracions personalitzades per a decoració o simulació logística.

2.2.7. AlgoritmoVoraz

Descripció:

El **AlgoritmoVoraz** és un algorisme que utilitza **backtracking** per trobar la millor ordre de productes en funció de la similitud entre ells. Busca maximitzar la suma de les similituds entre productes consecutius en una llista

Atributs:

- **best_sim**: Emmagatzema la major similitud acumulada trobada.
- **lordenada**: Llista amb la millor ordre de productes calculat.
- **Matrix**: Matriu que emmagatzema les similituds entre els productes.
- **visitados**: Llista per rastrejar els productes ja processats.

Funcionalitat:

Explicat al punt 4.2.1

2.2.8. Aproximation_Kruskal_ILS

Descripció:

L'algorisme **Aproximation_Kruskal_ILS** implementa una aproximació per resoldre problemes de rutes i cicles en grafos utilitzant l'algorisme de **Kruskal** combinat amb la **cerca local iterativa (ILS)**. L'objectiu és trobar solucions aproximades minimitzant els costos, generant **arbres d'expansió mínima (MST)** i transformant-los en **cicles hamiltonians**.

Atributs i Classes Internes:

- **Edge**: Representa una aresta entre dos productes amb un pes associat.
- **MFS**: Estructura per gestionar conjunts disjunts mitjançant unió i cerca (per a l'algorisme de Kruskal).
- **parent**: Mapa que emmagatzema l'arrel de cada conjunt.

- **rank**: Mapa que emmagatzema el rang de cada conjunt per optimitzar la unió.

Flux de l'Algorisme:

Explicat al punt 4.2.2

2.2.9. menuController

S'encarrega de gestionar la interfície d'usuari a l'aplicació, permetent la navegació pel menú principal i les diferents opcions relacionades amb la gestió d'usuaris, llistes de productes i distribucions. Aquesta classe interactua amb el controlador d'usuaris (UserController) per delegar tasques específiques.

Atributs:

- **UserController userController**: Instància del controlador d'usuaris, utilitzada per interactuar amb les funcionalitats relacionades amb els usuaris, les llistes de productes i les distribucions.

2.2.10. UserController

S'encarrega de la gestió dels mètodes de la classe User, proporcionant mètodes de register, login, logout... Inicialitza els controladors de llista i distribució. Gestiona els inputs i outputs per pantalla corresponents a una sessió d'usuari.

Atributs:

- **CjtUsers cjtUsers**: Conté el conjunt de tots els usuaris registrats a l'aplicació.
- **User currentUser**: Representa l'usuari actual que ha iniciat sessió.
- **listController lc**: Controlador que s'encarrega de gestionar les llistes de productes de l'usuari actual.
- **distributionController dc**: Controlador que s'encarrega de gestionar les distribucions de productes de l'usuari actual.

2.2.11. listController

S'encarrega de la gestió dels mètodes de les classes de LlistaProductes i CjtLlistesProductes. Proporciona mètodes per afegir, eliminar i editar llistes, incloent-hi mètodes per gestionar els graus de similitud entre productes.

Atributs:

- **CjtLlistesProductes cjt**: Representa el conjunt de llistes de productes gestionades per la classe listController.

2.2.12. distributionController

S'encarrega de la gestió dels mètodes de les classes de Distribucion i CjtDistribucion. Proporciona mètodes per afegir, eliminar, editar i exportar distribucions, així com per generar-les aplicant un algorisme de distribució.

3. Relació de les classes implementades per membre de l'equip

| David Sanz | David Mas | Nadia Khier | Roger Cot |
|------------|----------------|-------------------|-------------------|
| Tests | Distribucion | CjtDistribucion | CjtUsers |
| Voraz | Estrategia | CjtLlistesProd | Aproximació |
| Estrategia | DistController | Mensaje | formatException |
| | | Pair | LlistaProductes |
| | | DistController | User |
| | | DomainController | DomainController |
| | | ListController | mensajeController |
| | | mensajeController | Singleton |
| | | DomainDriver | userController |

4. Estructures de dades i algorismes

4.1. Estructures de dades

1. CjtDistribuciones:

○ Map (HashMap):

Propòsit: Emmagatzemar i gestionar un conjunt de distribucions.

Estructura: L'atribut **conjuntDistribucions** es un **Map<String, Distribucion>**, on:

- **Clau:** Representa el nom únic de cada distribució.
- **Valor:** Es la instància de la classe **Distribucion** associada al nom.

Avantatges:

1. **Recerca eficient:** Permet accedir a les distribucions en temps promig constant $O(1)$ mitjançant el seu nom.
2. **Gestió flexible:** Facilita l'addició, eliminació i verificació de l'existència de distribucions.
3. **Accés a claus:** El mètode **getNames** utilitza **keySet()** per retornar un conjunt de noms de distribucions sense emmagatzemar duplicats.

2. CjtLlistesProductes:

○ Map (HashMap):

Propòsit: Emmagatzemar i gestionar un conjunt de llistes de productes.

Estructura: L'atribut **conjuntoListas** és un **Map<String, LlistaProductes>**, on:

- **Clau:** Representa el nom únic de cada llista de productes.
- **Valor:** És la instància de la classe **LlistaProductes** associada al nom.

Avantatges: Ídem que **CjtDistribuciones**.

3. CjtUsers:

- **List(ArrayList):**

Propòsit: Retornar els noms d'usuari en forma de llista ordenada.

Estructura: S'utilitza al mètode **getAllUsernames()**, que converteix les claus del **Map** en una **ArrayList<String>**

Avantatges:

1. **Accés seqüencial:** Ideal per retornar els noms d'usuari de manera ordenada i fàcilment iterables.

- **Map (HashMap):**

Propòsit: Emmagatzemar i gestionar un conjunt d'usuaris.

Estructura: L'atribut **usersList** és un **Map<String, User>**, on:

- **Clau:** Representa el nom únic de cada usuari (username).
- **Valor:** És la instància de la classe **User** associada a aquest nom.

Avantatges: Ídem que **CjtDistribuciones**.

4. Distribucion:

- **Pair**

Propòsit: Emmagatzemar les dimensions de la distribució (alçada i longitud).

Estructura: **Pair<Integer, Integer> Prestage**, on:

- El primer element representa l'alçada.
- El segon element representa la longitud.

Ús: Es fa servir per definir i consultar les dimensions de la distribució.

- **List (ArrayList)**

Propòsit: Gestionar la llista ordenada de productes que formen part de la distribució.

Estructura: **List<String> Dist**, on cada element és el nom d'un producte.

Avantatges:

1. Permet l'accés seqüencial als noms dels productes.
2. És fàcilment iterable i manipulable.

- **Map (HashMap)**

Propòsit: Mantenir una associació entre els noms dels productes i les seves posicions dins la distribució.

Estructura: **Map<String, Integer> mapa**, on:

- La clau (**String**) és el nom del producte.
- El valor (**Integer**) és la seva posició a la distribució.

Ús:

- Cerca ràpida de la posició d'un producte ($O(1)$).
- Manipulació eficient durant operacions com intercanviar o eliminar productes.

- **Map anidat**

Propòsit: Rebre les dades d'entrada per generar una distribució mitjançant una estratègia.

Estructura: `Map<String, Map<String, Float>>` al mètode `generateDist`.

- El primer nivell del mapa (`Map<String, Map<String, Float>>`) associa el nom d'un producte amb un altre `Map`.
- El segon nivell del mapa (`Map<String, Float>`) enllaça altres productes amb un grau de similitud (`Float`).

Ús: El mapa anidat s'utilitza al mètode `generateDist`, on s'envia com a entrada a l'estratègia definida. L'estratègia processa aquestes relacions per decidir l'ordre final dels productes.

5. LlistaProductes:

○ Map anidat

Propòsit: Gestionar una llista de productes amb una matriu de similituds entre ells.

Estructura: `Map<String, Map<String, Float>>` `similarityMatrix`, on:

- El primer nivell del mapa (`Map<String, Map<String, Float>>`) associa el nom d'un producte amb un altre mapa.
- El segon nivell del mapa (`Map<String, Float>`) conté els productes relacionats amb un grau de similitud (de tipus `Float`).

Ús: Aquest mapa s'utilitza per emmagatzemar la matriu de similituds entre productes. Quan es vol obtenir la similitud entre dos productes, es consulta aquest mapa. També permet afegir nous productes i eliminar-los juntament amb les seves relacions de similitud. A més, es poden actualitzar els valors de similitud entre productes mitjançant el mètode `setSimilarity`.

6. User:

○ List (ArrayList)

Propòsit: Gestionar una llista de productes amb una matriu de similituds entre ells.

Estructura: `List<Mensaje>` `inbox`, `List<Mensaje>` `sentMessages`.

Els dos atributs són llistes (`List<Mensaje>`) que emmagatzemen els missatges rebuts (`inbox`) i els enviats (`sentMessages`).

Ús: Aquestes llistes s'utilitzen per gestionar els missatges del usuari, emmagatzemant els missatges rebuts i els enviats. El comptador de missatges nous (`mensajesNuevos`) s'incrementa cada vegada que s'afegeix un nou missatge a la safata d'entrada.

4.2. Algorismes

4.2.1. Algoritme voraç

L'AlgoritmeVoraz és un algorisme que utilitza backtracking per trobar la millor distribució de productes en funció de la similitud entre ells. Busca maximitzar la suma de les similituds entre productes consecutius en una llista.

1) Atributs

- a) **best_sim**: Emmagatzema la major similitud acumulada trobada.
- b) **lordenada**: Llista amb la millor ordre de productes calculat.
- c) **Matrix**: Matriu que emmagatzema les similituds entre els productes.
- d) **visitats**: Llista per rastrejar els productes ja processats.

2) Mètodes

a) **backtracking**

Genera totes les combinacions possibles de productes per trobar la millor.

- Si es completa una llista, calcula la seva similitud acumulada.
- Si la similitud és millor que l'anterior, actualitza els resultats.
- Continua construint la llista fins processar tots els productes.

b) **calcular_sim**

Suma les similituds entre els productes consecutius en una llista parcial.

Ajuda a avaluar si una combinació actual és millor que l'anterior.

c) **executeAlg**

Inicialitza les variables i executa l'algorisme per trobar la millor llista.

- Rep una matriu de similituds com a entrada.
- Torna l'ordre òptim de productes.

4.2.2. Algoritme d'aproximació

Aquest algoritme implementa l'algoritme d'aproximació de Kruskal amb Cerca Local Iterativa (ILS), utilitzat per trobar solucions aproximades a problemes de rutes amb cost mínim. El procés es desenvolupa en diverses etapes per generar un cicle Hamiltonià a partir d'un arbre d'expansió mínima (MST) i optimitzar-lo mitjançant tècniques com la cerca local i el redisseny d'arcs. A continuació es descriu cada part de l'algoritme:

1) Definició de les classes internes:

- a) **Edge**: Representa una aresta entre dos nodes amb el seu pes associat (en aquest cas, la similitud entre productes).
- b) **MFS (Union-Find)**: Implementa l'estructura de conjunts disjunts per gestionar la unió i la cerca de conjunts de nodes durant l'algoritme de Kruskal, prevenint la formació de cicles en el MST.

2) Algoritme de Kruskal (obtenir MST):

- a) S'extreuen totes les arestes possibles entre els nodes del graf, les quals es classifiquen per pes (similitud).
- b) Utilitzant l'estructura MFS, s'uneixen els nodes, afegint només aquelles arestes que no causin un cicle, obtenint així un arbre d'expansió mínima (MST).

3) Cerca Local Iterativa (ILS):

- a) L'algoritme utilitza l'algoritme 2-opt, que realitza millores locals en el MST invertint l'ordre de subllistes d'arestes per reduir el cost global.
- b) S'introdueix una **perturbació aleatòria** en cada iteració per evitar quedar-se en òptims locals, millorant contínuament la solució en un màxim de 100 iteracions.

4) Generació del graf dirigit:

- a) Es crea un graf dirigit a partir de l'MST optimitzat, amb les arestes que representen les connexions entre nodes de manera que cada node apunta als seus veïns.

5) Generació del Cicle Eulerià:

- a) Mitjançant un recorregut DFS (cerca en profunditat), es construeix un cicle Eulerià a partir del graf dirigit, prioritizant les connexions més fortes (les que tenen més similitud entre els nodes).

6) Transformació en un Cicle Hamiltonià:

- a) El cicle Eulerià es transforma en un cicle Hamiltonià eliminant nodos repetits, de manera que cada producte aparegui una sola vegada en el cicle final.

7) Execució de l'algoritme:

El mètode *executeAlg* integra tots els passos anteriors:

1. Calcula el MST amb Kruskal.
2. Optimitza el MST mitjançant ILS.
3. Genera el grafo dirigit i el cicle Eulerià.
4. Finalment, transforma el cicle Eulerià en un cicle Hamiltonià i el retorna com a solució.