Universitat Politècnica de Catalunya

Facultat d'Informàtica de Barcelona

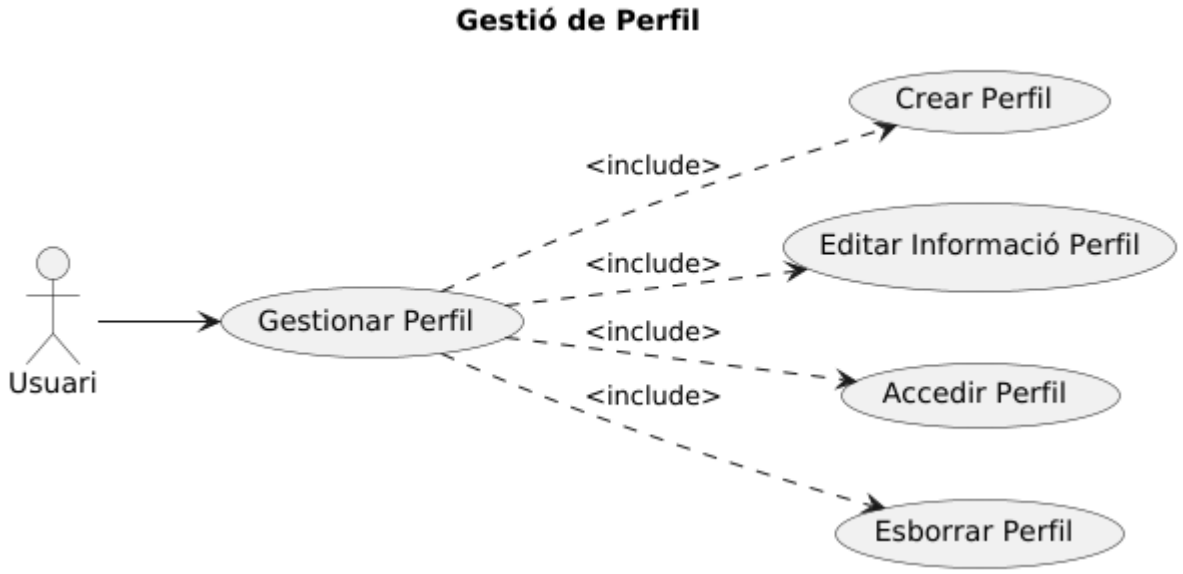
Distribució de productes a un supermercat

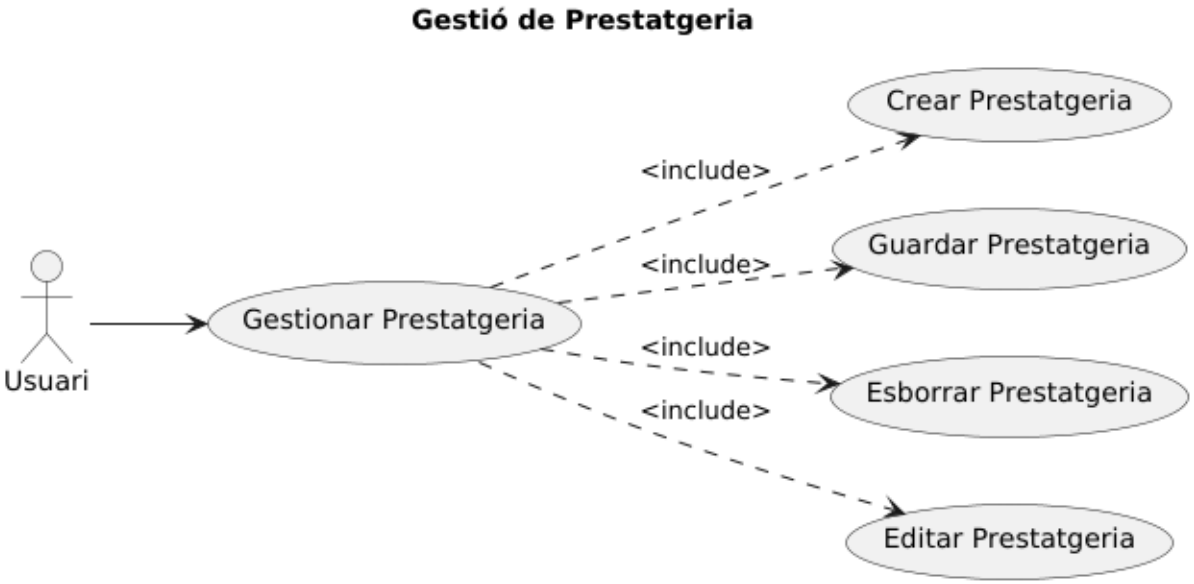
Grup 31.5
Èric Díez Apolo (eric.diez)
Pol Carnicer González (pol.carnicer)
Francesc Pérez Venegas (francesc.perez.venegas)
Aleix Montero Ponce (aleix.montero)

PROP - Carles Arnal Castello
18 de Novembre del 2024

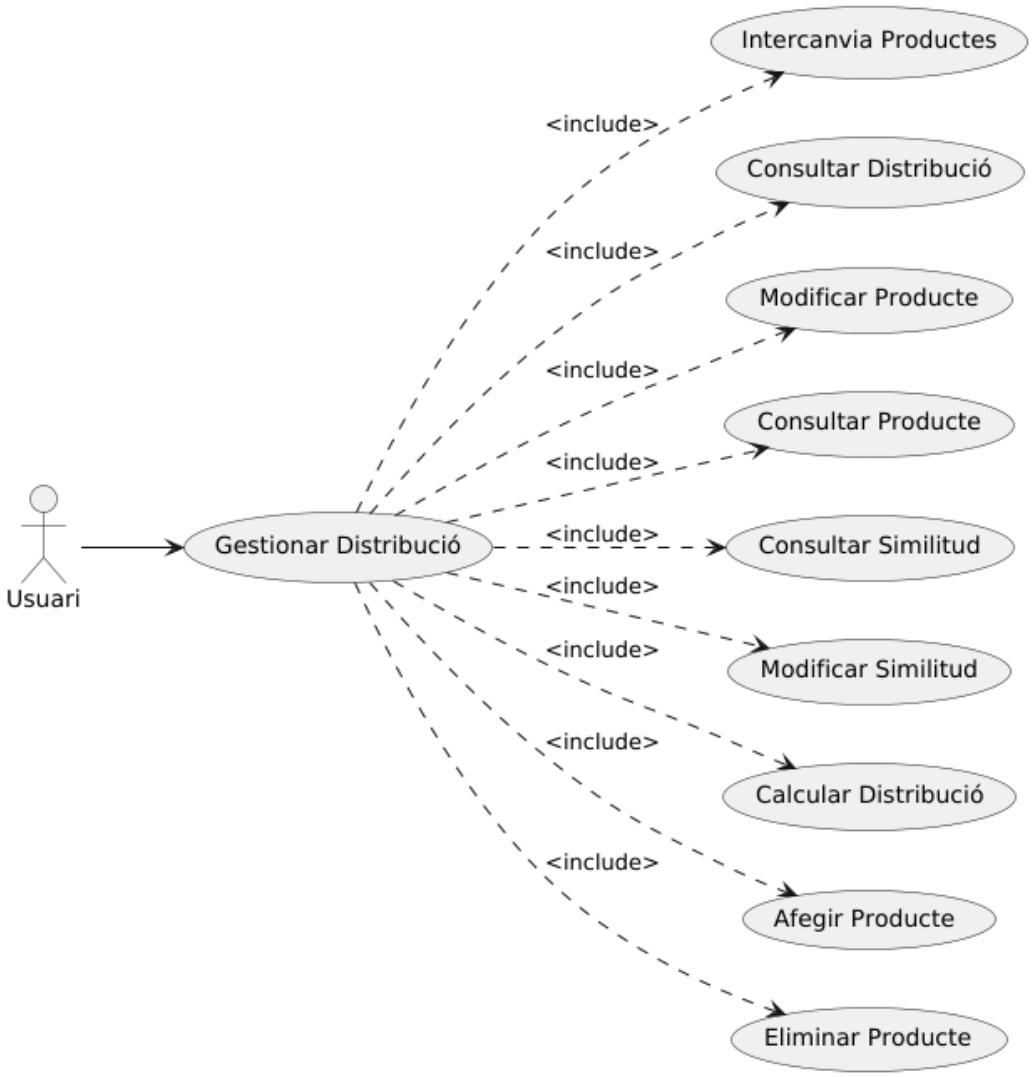
Índex

1. Diagrama de casos d'ús

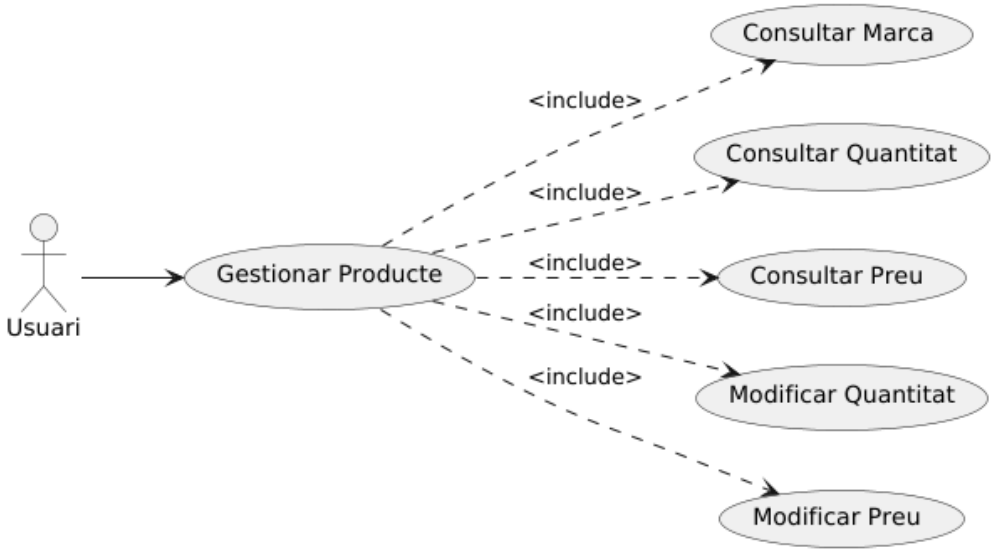


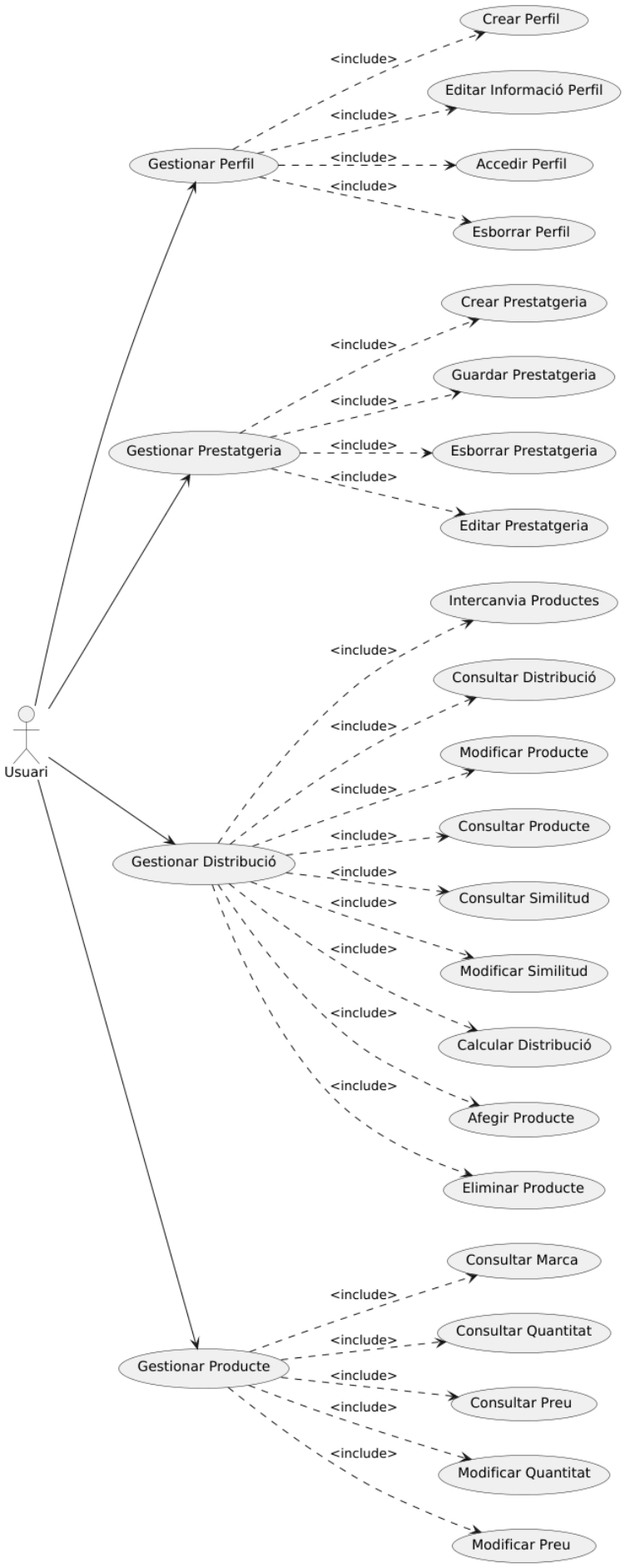


Gestió de Distribució



Gestió de Producte




Crear Perfil
<include>
Editar Informació Perfil
<include>
Gestionar Perfil
<include>
Accedir Perfil
<include>
Esborrar Perfil
Crear Prestatgeria
<include>
Guardar Prestatgeria
<include>
Gestionar Prestatgeria
<include>
Esborrar Prestatgeria
<include>
Editar Prestatgeria
Intercanvia Productes
<include>
Consultar Distribució
<include>
Modificar Producte
<include>
Consultar Producte
<include>
Gestionar Distribució
<include>
Consultar Similitud
<include>
Modificar Similitud
<include>
Calcular Distribució
<include>
Afegir Producte
Eliminar Producte
Usuari
Consultar Marca
<include>
Consultar Quantitat
<include>
Gestionar Producte
<include>
Consultar Preu
<include>
Modificar Quantitat
<include>
Modificar Preu

1.1 Descripció casos d'ús

A continuació, donarem una descripció detallada dels diferents casos d'ús que tenim:

Cas d'ús 1: Crear perfil

Nom: Crear perfil

Actor: Nou usuari

Comportament (diàleg entre els actors i el sistema):

1. Usuari: Selecciona l'opció de crear perfil a l'aplicació.
2. Sistema: Mostra un formulari que demana informació com nom i contrasenya.
3. Usuari: Omple el formulari amb la seva informació.
4. Sistema: Valida la informació introduïda.
5. Sistema: Si tot és correcte, crea el perfil d'usuari i mostra un missatge de confirmació.
6. Usuari: Rep la confirmació.

Errors possibles i cursos alternatius:

Cas d'ús 2: Accedeix perfil

Nom: Accedeix perfil

Actor: Usuari registrat

Comportament (diàleg entre els actors i el sistema):

1. Usuari: Selecciona l'opció de carregar perfil a l'aplicació.
2. Sistema: Demana introduir usuari i contrasenya.
3. Usuari: Introdueix el nom d'usuari i després la contrasenya.
4. Sistema: Si les credencials són correctes, carrega la informació del perfil associat a l'usuari.
5. Usuari: Visualitza la pantalla de gestió de prestatgeries.

Errors possibles i cursos alternatius:

- Error: Les credencials són incorrectes.
    - Curs alternatiu: El sistema informa l'usuari de l'error i ofereix l'opció de tornar a intentar-ho.

Cas d'ús 3: Esborrar perfil

Nom: Esborrar perfil

Actor: Usuari registrat

Comportament (diàleg entre els actors i el sistema):

1. Usuari: Selecciona l'opció "Esborrar perfil".
2. Sistema: Mostra per pantalla la llista dels perfils creats.
3. Usuari: Selecciona quin usuari vol esborrar.
4. Sistema: Esborra totes les dades associades amb el perfil de l'usuari i envia missatge de confirmació.
5. Usuari: Rep la confirmació que el seu perfil ha estat eliminat.

Errors possibles i cursos alternatius:

- Error: No hi ha cap usuari creat per eliminar.
    - Curs alternatiu: El sistema envia missatge d'error i torna a la pantalla principal.
- Error: L'usuari selecciona una opció fora del rang de les possibles.
    - Curs alternatiu: El sistema mostra per pantalla l'error que l'opció seleccionada no està disponible i torna al menú.

Cas d'ús 4: Editar informació perfil

Nom: Editar informació perfil

Actor: Usuari registrat

Comportament (diàleg entre els actors i el sistema):

1. Usuari: Selecciona l'opció "Editar perfil".
2. Sistema: Mostra per pantalla una llista dels perfils creats.
3. Usuari: Selecciona el perfil a editar.
4. Sistema: Dona opció a editar el nom d'usuari o la contrasenya.
5. Usuari: Selecciona una opció.
    1. Usuari: Selecciona editar nom d'usuari.
        1. Sistema: Demana que s'escrigui el nou nombre d'usuari.
        2. Usuari: Escriu el nou nom d'usuari

3. Sistema: Actualitza el nom d'usuari de l'usuari seleccionat, envia missatge de confirmació i torna al menú principal.

2. Usuari: Selecciona canviar contrasenya.
    1. Sistema: Demana que s'escrigui la contrasenya actual de l'usuari.
    2. Usuari: Escriu la contrasenya actual.
    3. Sistema: Demana que escrigui la nova contrasenya.
    4. Usuari: Escriu la nova contrasenya.
    5. Sistema: Actualitza la contrasenya de l'usuari seleccionat, envia missatge de confirmació i torna al menú principal.

Errors possibles i cursos alternatius:

- Error (5.b.iii): La contrasenya actual no és correcta.
    - Curs alternatiu: El sistema indica un missatge d'error de contrasenya incorrecta i torna al menú.
- Error (5.a.iii): L'usuari ja existeix.
    - Curs alternatiu: El sistema indica els requisits necessaris per a la nova contrasenya i sol·licita una nova entrada.
- Error (3): No hi ha cap usuari creat per eliminar.
    - Curs alternatiu: El sistema envia missatge d'error i torna a la pantalla principal.
- Error (4, 6): L'usuari selecciona una opció fora del rang de les possibles.
    - Curs alternatiu: El sistema mostra per pantalla l'error que l'opció seleccionada no està disponible i torna al menú.

Cas d'ús 5 : Crea Prestatgeria

Nom: Crea Prestatgeria

Actor: Usuari

Comportament (diàleg entre els actors i el sistema):

1. Usuari: Selecciona l'opció "Crea Prestatgeria"
2. Sistema: Mostra per pantalla un formulari que demana el nom de la prestatgeria i l'altura d'aquesta.
3. Usuari: Escriu el nom de la prestatgeria i la seva altura.
4. Sistema: Valida la informació introduïda i envia missatge de confirmació per pantalla.

Errors possibles i cursos alternatius:1

- Error (4): Ja existeix una prestatgeria amb el mateix nom a l'introduït per l'usuari
    - Curs alternatiu: El sistema informa amb un missatge a l'usuari que el nom no és vàlid i torna al menú.

Cas d'ús 6 : Guarda Prestatgeria

Nom: Guarda Prestatgeria

Actor: Usuari

Comportament (diàleg entre els actors i el sistema):

1. Usuari: Seleccionen l'opció "Guarda Prestatgeria"
2. Sistema: Guarda i sobreescriu la prestatgeria

Errors possibles i cursos alternatius:

Cas d'ús 7: Borra Prestatgeria

Nom: Borra Prestatgeria

Actor: Usuari

Comportament (diàleg entre els actors i el sistema):

1. Usuari: Selecciona l'opció "Borra Prestatgeria"
2. Sistema: Mostra per pantalla una llista amb les prestatgeries creades.
3. Usuari: Selecciona la prestatgeria que vol esborrar.
4. Sistema: Esborra totes les dades associades a la prestatgeria i mostra un missatge de confirmació.
5. Usuari: Rep la confirmació que la prestatgeria ha estat eliminada.

Errors possibles i cursos alternatius:

- Error (4, 6): L'usuari selecciona una opció fora del rang de les possibles.
    - Curs alternatiu: El sistema mostra per pantalla l'error que l'opció seleccionada no està disponible i torna al menú.

Cas d'ús 8: Edita Prestatgeria

Nom: Edita prestatgeria

Actor: Usuari registrat

Comportament (diàleg entre els actors i el sistema):

1. Usuari: Selecciona l'opció "Editar prestatgeria".
2. Sistema: Mostra per pantalla una llista de les prestatgeries creades.
3. Usuari: Selecciona la prestatgeria a editar.
4. Sistema: Mostra el nom i altura actuals i pregunta si vol canviar el nom.
5. Usuari: Selecciona una opció.
    1. Usuari: Indica que si
        1. Sistema: Demana que s'escrigui el nou nombre de prestatgeria.
        2. Usuari: Escriu el nou nom de prestatgeria.
        3. Sistema: Actualitza el nom de la prestatgeria seleccionada i envia missatge de confirmació.
    2. Usuari: Indica que no.
6. Sistema: Pregunta si es vol canviar l'altura.
    1. Usuari: Indica que sí.
        1. Sistema: Demana que s'escrigui la nova altura.
        2. Usuari: Escriu el nou valor de l'altura.
        3. Sistema: Actualitza el valor de l'altura de la prestatgeria i envia missatge de confirmació.

Errors possibles i cursos alternatius:

- Error (2): No hi ha prestatgeries creades.
    - Curs alternatiu: El sistema indica un missatge d'error informant que no hi ha prestatgeries i torna al menú.
- Error (4): L'usuari selecciona una opció fora del rang de les possibles.
    - Curs alternatiu: El sistema mostra per pantalla l'error que l'opció seleccionada no està disponible i torna al menú.

Cas d'ús 9: Consultar distribució

Nom: Consultar distribució

Actor: Usuari

Comportament (diàleg entre els actors i el sistema):

1. Usuari: Selecciona l'opció "Consultar Distribució.
2. Sistema: Mostra per pantalla la distribució sencera i pregunta si es vol editar.
1. Usuari: Indica que sí.
    1. Sistema: Mostra opció per intercanviar productes.
2. Usuari: Indica que no.

Errors possibles i cursos alternatius:

- Error (2): No hi ha cap producte a la distribució.
    - Curs alternatiu: El sistema mostra per pantalla l'error que no hi ha productes a la distribució i torna al menú.

Cas d'ús 10: Intercanviar productes

Nom: Intercanviar productes

Actor: Usuari

Comportament (diàleg entre els actors i el sistema):

1. Usuari: Indica que sí que vol editar una distribució quan l'estava consultant.
2. Sistema: Dona una llista dels productes de la distribució i demana el primer.
3. Usuari: Indica el primer producte a intercanviar.
4. Sistema: Demana el segon producte.
5. Usuari: Indica el segon producte a intercanviar.
6. Sistema: El sistema intercanvia els dos productes de posició.

Errors possibles i cursos alternatius:

- (Error: No es poden intercanviar 2 productes iguals
    - Curs alternatiu: El sistema informa amb un missatge a l'usuari que no es poden intercanviar 2 productes iguals.)
- Error (4, 6): L'usuari selecciona una opció fora del rang de les possibles.
    - Curs alternatiu: El sistema mostra per pantalla l'error que l'opció seleccionada no està disponible i torna al menú.

Cas d'ús 11: Afegeix Producte

Nom: Afegeix Producte

Actor: Usuari

Comportament (diàleg entre els actors i el sistema):

1. Usuari: Selecciona l'opció "Afegir Producte".
2. Sistema: Demana el nou del producte a afegir.
3. Usuari: Escriu el nom del nou producte.
4. Sistema: Demana introduir la marca, preu i quantitat del producte.
5. Usuari: Introdueix els atributs corresponents.
6. Sistema: Dona opció a afegir la similitud amb altres productes i envia missatge de confirmació.

Errors possibles i cursos alternatius:

- Error: El producte ja existeix.
    - Curs alternatiu: El sistema envia missatge d'error per pantalla que ja existeix el producte i torna al menú.
- Error: El preu és menor que 0.
    - Curs alternatiu: El sistema mostra per pantalla l'error que el preu és incorrecte (menor que 0) i torna al menú.
- Error: La quantitat és menor que 0.
    - Curs alternatiu: El sistema mostra per pantalla l'error que la quantitat és incorrecte (menor que 0) i torna al menú.

Cas d'ús 12: Borra Producte

Nom: Borra Producte

Actor: Usuari

Comportament (diàleg entre els actors i el sistema):

1. Usuari: Selecciona l'opció "Elimina Producte"
2. Sistema: Mostra per pantalla una llista amb tots els productes de la distribució.
3. Usuari: Selecciona el producte que vol eliminar.
4. Sistema: Elimina el producte de la distribució i envia un missatge de confirmació.

Errors possibles i cursos alternatius:

- Error (4): Producte no existeix?
    - Curs alternatiu: El sistema cancel·la l'esborrament i retorna a l'usuari a la gestió de prestatge.
- Error (4): L'usuari selecciona una opció fora del rang de les possibles.
    - Curs alternatiu: El sistema mostra per pantalla l'error que l'opció seleccionada no està disponible i torna al menú.

Cas d'ús 13: Consulta Productes

Nom: Consulta Productes

Actor: Usuari

Comportament (diàleg entre els actors i el sistema):

1. Usuari: Selecciona l'opció "Consultar Productes"
2. Sistema: Mostra per pantalla tot el conjunt de productes que formen part de la distribució.

Errors possibles i cursos alternatius:

- Error (2): No hi ha productes a la distribució
    - Curs alternatiu: El sistema mostra el missatge d'error per pantalla que no hi ha productes a la distribució.

Cas d'ús 14: Consulta Similitud

Nom: Consulta Similitud

Actor: Usuari

Comportament (diàleg entre els actors i el sistema):

1. Usuari: Selecciona l'opció "Consulta Similituds".
2. Sistema: Mostra per pantalla el conjunt de similituds de cada producte de la distribució amb la resta.

Errors possibles i cursos alternatius:

- Error (2): No hi ha productes a la distribució
    - Curs alternatiu: El sistema mostra el missatge d'error per pantalla que no hi ha productes a la distribució.

Cas d'ús 15: Modifica Similitud

Nom: Modifica Similitud

Actor: Usuari

Comportament (diàleg entre els actors i el sistema):

1. Usuari: Selecciona l'opció "Modifica Similituds".
2. Sistema: Mostra per pantalla tot el conjunt de productes per a escollir de quin canviar les similituds.
3. Usuari: Selecciona el producte del qual vol canviar similituds.
4. Sistema: Mostra per pantalla la resta de productes per escollir amb qui es vol canviar la similitud.
5. Usuari: Selecciona el producte i modifica la similitud.
6. Sistema: Pregunta si es vol canviar alguna més i actualitza la similitud.

Errors possibles i cursos alternatius:

- Error (2): No hi ha productes a la distribució
    - Curs alternatiu: El sistema mostra el missatge d'error per pantalla que no hi ha productes a la distribució.
- Error (4, 6): L'usuari selecciona una opció fora del rang de les possibles.
    - Curs alternatiu: El sistema mostra per pantalla l'error que l'opció seleccionada no està disponible i torna al menú.
- Error (6): Valor de similitud incorrecte ($x < 0$ i $100 < x$)
    - Curs alternatiu: El sistema mostra el missatge d'error per pantalla que el valor ha d'estar entre 0 i 100 i torna al menú.

Cas d'ús 16: Modifica Producte

Nom: Modifica Producte

Actor: Usuari

Comportament (diàleg entre els actors i el sistema):

1. Usuari: Selecciona l'opció "Modificar Producte".
2. Sistema: Mostrarà per pantalla els productes de la distribució i demana que es seleccioni un.
3. Usuari: Selecciona el producte que vol modificar.
4. Sistema: Mostra per pantalla tots els atributs i li pregunta si el vol editar.
5. Usuari: Indica que sí.
6. Sistema: Indica que es doni un nou preu i quantitat.
7. Usuari: Escriu el preu i quantitat modificats del producte.
8. Sistema: Actualitza els atributs i envia missatge de confirmació per pantalla.

Errors possibles i cursos alternatius:

- Error (8): El nou preu és menor que 0.
    - Curs alternatiu: El sistema mostra per pantalla l'error que el preu és incorrecte (menor que 0) i torna al menú.
- Error (8): La nova quantitat és menor que 0.
    - Curs alternatiu: El sistema mostra per pantalla l'error que la quantitat és incorrecte (menor que 0) i torna al menú.

Cas d'ús 17: Calcula Distribució

Nom: Calcula distribució

Actor: Usuari

Comportament (diàleg entre els actors i el sistema):

1. Usuari: Selecciona l'opció "Calcular Distribució".
2. Sistema: Dona a escollir entre les dues estratègies de càlcul de distribució.
3. Usuari: Selecciona l'estratègia que vol.
4. Sistema: Fa el càlcul i mostra per pantalla la distribució segons el càlcul fet

Errors possibles i cursos alternatius:

- Error (4): L'usuari selecciona una opció fora del rang de les possibles.
- Curs alternatiu: El sistema mostra per pantalla l'error que l'opció seleccionada no està disponible i torna al menú.

Cas d'ús 18: Consulta Quantitat

Nom: Consulta Quantitat

Actor: Usuari

Comportament (diàleg entre els actors i el sistema):

1. Usuari: Indica que vol modificar un producte per així veure la seva informació.
2. Sistema: Mostrarà per pantalla els productes de la distribució i demana que es seleccioni un.
3. Usuari: Indica el producte que vol consultar.
4. Sistema: Mostra per pantalla tots els atributs incloent la quantitat.

Errors possibles i cursos alternatius:

- Error (4): L'usuari selecciona una opció fora del rang de les possibles.
    - Curs alternatiu: El sistema mostra per pantalla l'error que l'opció seleccionada no està disponible i torna al menú.

Cas d'ús 19: Consulta Preu

Nom: Consulta Preu

Actor: Usuari

Comportament (diàleg entre els actors i el sistema):

1. Usuari: Indica que vol modificar un producte per així veure la seva informació.
2. Sistema: Mostrarà per pantalla els productes de la distribució i demana que es seleccioni un.
3. Usuari: Indica el producte que vol consultar.
4. Sistema: Mostra per pantalla tots els atributs incloent el preu.

Errors possibles i cursos alternatius:

- Error (4): L'usuari selecciona una opció fora del rang de les possibles.
    - Curs alternatiu: El sistema mostra per pantalla l'error que l'opció seleccionada no està disponible i torna al menú.

Cas d'ús 20: Modifica Preu

Nom: Modifica Preu

Actor: Usuari

Comportament (diàleg entre els actors i el sistema):

1. Usuari: Selecciona l'opció "Modificar Producte".
2. Sistema: Mostrarà per pantalla els productes de la distribució i demana que es seleccioni un.
3. Usuari: Indica el producte que vol consultar.
4. Sistema: Mostra per pantalla tots els atributs incloent el preu i li pregunta si el vol editar.
5. Usuari: Indica que sí.
6. Sistema: Indica que es doni un nou preu.
7. Usuari: Escriu el preu modificat del producte.
8. Sistema: Actualitza el preu i envia missatge de confirmació per pantalla.

Errors possibles i cursos alternatius:

- Error (8): El nou preu és menor que 0.
    - Curs alternatiu: El sistema mostra per pantalla l'error que el preu és incorrecte (menor que 0) i torna al menú.

Cas d'ús 21: Modifica Quantitat

Nom: Modifica Quantitat

Actor: Usuari

Comportament (diàleg entre els actors i el sistema):

1. Usuari: Indica que vol modificar un producte.
2. Sistema: Mostrarà per pantalla els productes de la distribució i demana que es seleccioni un.
3. Usuari: Indica el producte que vol consultar.
4. Sistema: Mostra per pantalla tots els atributs incloent la quantitat li pregunta si el vol editar.
5. Usuari: Indica que sí.
6. Sistema: Indica que es doni una nova quantitat.
7. Usuari: Escriu la nova quantitat del producte.
8. Sistema: Actualitza la quantitat i envia missatge de confirmació per pantalla.

Errors possibles i cursos alternatius:

- Error (8): La nova quantitat és menor que 0.
    - Curs alternatiu: El sistema mostra per pantalla l'error que la quantitat és incorrecte (menor que 0) i torna al menú.

Cas d'ús 22: Consulta Marca

Nom: Consulta Marca

Actor: Usuari

Comportament (diàleg entre els actors i el sistema):

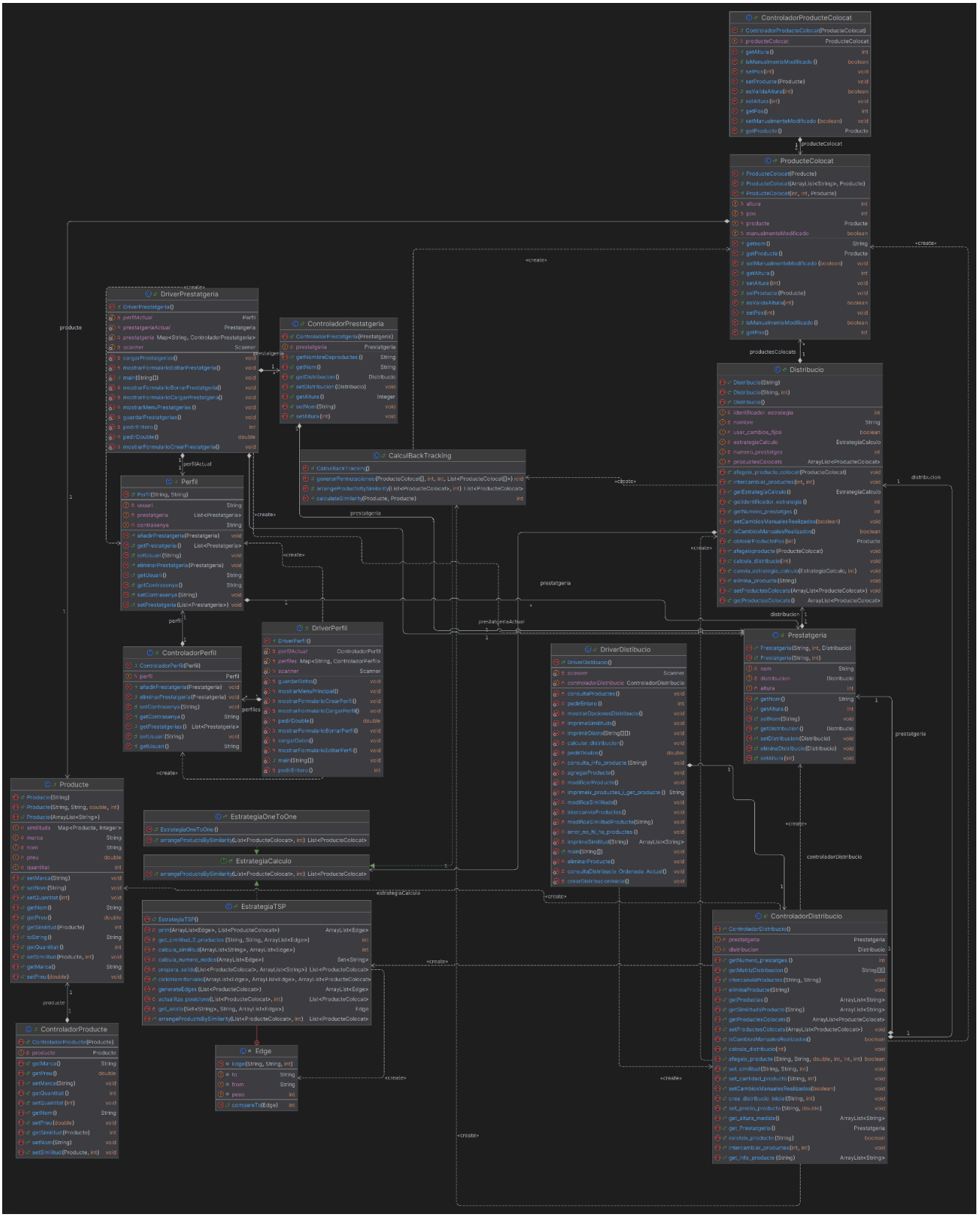
1. Usuari: Indica que vol modificar un producte per així veure la seva informació.
2. Sistema: Mostrarà per pantalla els productes de la distribució i demana que es seleccioni un.
3. Usuari: Indica el producte que vol consultar.
4. Sistema: Mostra per pantalla tots els atributs incloent la marca.

Errors possibles i cursos alternatius:

- Error (4): L'usuari selecciona una opció fora del rang de les possibles.
    - Curs alternatiu: El sistema mostra per pantalla l'error que l'opció seleccionada no està disponible i torna al menú.

2. Diagrama de model conceptual

2.1 Disseny del diagrama del model conceptual



2.2 Descripció de les classes

2.2.1 Perfil

La classe perfil és la classe que conté les dades de l'usuari (usuari i contrasenya) i que emmagatzema les diferents prestatgeries creades per l'usuari. S'encarrega principalment de la gestió dels usuaris i prestatgeries d'aquests, creant ambdós i poden eliminar-los i editar-los.

Els atributs d'aquesta clase son:

- Usuari(String): Emmagatzema el nom d'usuari associat al perfil.
- Contrasenya(String): Emmagatzema la contrasenya corresponent al perfil.
- Prestatgeria(List<Prestatgeria>): Una llista d'objectes Prestatgeria que representen les estanteries associades al perfil.

Els mètodes son:

- añadirPrestatgeria(Prestatgeria prestatgerias): Afegeix una estanteria a la llista de prestatgeries del perfil.
- eliminarPrestatgeria(Prestatgeria prestatgerias): Elimina una estanteria de la llista de prestatgeries del perfil.

2.2.2 Prestatgeria

La classe Prestatgeria proporciona una forma de gestionar i configurar prestatgeries mitjançant atributs com el nom, l'altura i una distribució associada. Permet establir o modificar la distribució per organitzar els productes dins de la prestatgeria, així com eliminar-la si cal.

Els atributs d'aquesta clase son:

- Nom(String): Emmagatzema el nom associat a la prestatgeria
- Distribució(Distribucio): Representa a la distribució de la prestatgeria.
- Altura(Integer): Emmagatzema l'altura de la prestatgeria / nombre de prestatges.

Els mètodes son:

- Prestatgeria(String, int, Distribucio): Inicialitza una prestatgeria amb un nom donat, una altura i la distribucio associada.
- Prestatgeria(String, int): Inicialitza una prestatgeria amb un nom donat, una altura.
- eliminaDistribució(Distribucio distribucio): Elimina la distribució de la prestatgeria

2.2.3 Distribució

La classe Distribució proporciona una forma de gestionar i organitzar productes en una estructura mitjançant una llista. Permet afegir, eliminar, intercanviar i ordenar productes, amb flexibilitat per utilitzar diferents algoritmes de càlcul. A més, pot gestionar restriccions com la fixació de certs productes en posicions específiques.
Els atributs d'aquesta clase son:

- nombre (String): El nom de la distribució.
- numero_prestatges (int): El nombre de prestatgeries en la distribució.
- usar_cambios_fijos (boolean): Indica si s'han realitzat canvis fixes en la distribució. Si és true, els productes amb l'atribut fixe es mantindran a la seva posició. Si és false, els productes es poden ordenar sense restriccions.
- productesColocats (ArrayList<ProducteColocat>): Llista d'objectes ProducteColocat que representen els productes que han estat col·locats dins de la distribució.
- estrategiaCalculo (EstrategiaCalculo): Una estratègia utilitzada per calcular la distribució dels productes.
- identificador_estrategia (int): Un identificador que indica el tipus d'estratègia utilitzada.

Els mètodes son:

- Distribucio(String nom): Inicialitza una distribució amb un nom donat, amb 0 prestatgeries.
- Distribucio(String nom, int numero_prestatge): Inicialitza una distribució amb un nom i un número de prestatgeries especificat.
- Distribucio(): Constructor per defecte que inicialitza una distribució amb 0 prestatgeries i utilitza BackTracking com a estratègia.
- afegeix_producte_colocat(ProducteColocat producteColocat): Afegeix un producte col·locat a la llista de productes de la distribució i incrementa el número de prestatgeries.
- obtenirProductePos(int pos): Retorna el producte situat a la posició indicada de la llista de productes col·locats.
- elimina_producte(String producte): Elimina un producte de la distribució basant-se en el nom del producte.
- intercambiar_productes(int seleccion1, int seleccion2): Intercanvia dos productes col·locats a les posicions indicades dins de la llista de productes col·locats.
- afegeixproducte(ProducteColocat producteColocat): Afegeix un producte col·locat a la llista de productes col·locats.

- isCambiosManualesRealizados(): Retorna true si s'han realitzat canvis manuals fixos en la distribució, false en cas contrari.
- calcula_distribucio(int altura): Calcula i ordena la distribució dels productes utilitzant l'estratègia de càlcul actual, tenint en compte l'altura proporcionada.
- canvia_estrategia_calculo(EstrategiaCalculo estrategia, int id_estrategia): Canvia l'estratègia de càlcul per una nova estratègia, juntament amb el seu identificador.

2.2.4 Producte

La classe producte representa a cadascun dels productes d'una distribució. Aquesta classe s'encarrega principalment d'emmagatzemar les dades de cada producte així com el nom, marca, preu i quantitat.
Els atributs d'aquesta clase son:

- nom (String): El nom del producte.
- marca (String): La marca del producte.
- preu (double): El preu del producte.
- quantitat (int): La quantitat disponible del producte.
- similituds (Map<Producte, Integer>): Un mapa que emmagatzema les similituds entre aquest producte i altres productes. La clau és un altre objecte Producte i el valor és un valor enter que indica el percentatge de similitud entre els productes (en un rang de 0 a 100).

Els mètodes son:

- Producte(String nom, String marca, double preu, int quantitat): Inicialitza un producte amb el nom, marca, preu i quantitat proporcionats.
- Producte(ArrayList<String> a): Inicialitza un producte a partir d'una llista de strings, amb valors per al nom, marca, preu (convertit a double) i quantitat (convertida a int).
- Producte(String nom): Inicialitza un producte amb el nom especificat i crea un mapa de similituds buit.

2.2.5 Producte Col·locat

La classe producte col·locat representa a cadascun dels productes d'una distribució un cop ja organitzats segons un dels algorismes o manualment. Aquesta classe s'encarrega principalment d'emmagatzemar les dades de la posició i en quina altura es troba cada producte.
Els atributs d'aquesta classe son:

- Altura(Integer): Representa a quina altura de la prestatgeria (el prestatge) està situat el producte.
- Pos(Int): Representa la posició del producte dins el prestatge.
- Producte(Producte): Una instància de la classe Producte que conté tota la informació del producte associat.
- ManualmenteModificado(boolean): Indica si la col·locació del producte ha estat modificada manualment, i s'inicialitza a false.

Els mètodes son:

- ProducteColocat(int pos, int altura, Producte producte): Assigna una posició, una altura i un producte a la instància creada. Inicialitza el camp manualmenteModificado com false.
- ProducteColocat(ArrayList<String> info, Producte prod): Construeix un objecte a partir d'una llista d'informació (info) i una instància de Producte. Interpreta els valors de pos, altura i si ha estat modificat manualment.
- ProducteColocat(Producte producte): Construeix un objecte només amb la informació del producte. Els altres camps queden per inicialitzar.
- esValidaAltura(Integer): Comprova si l'altura del producte és vàlida respecte a l'altura màxima de la prestatgeria. Retorna true si és vàlida, false en cas contrari.
- isManualmenteModificado(): Obté l'estat de si la col·locació ha estat modificada manualment.

2.2.6 Controlador Prestatgeria

El controlador de Prestatgeria s'encarrega de la gestió dels mètodes de la classe Prestatgeria. Es comunica amb Prestatgeria per tal de realitzar totes les funcionalitats que fan referència a aquest. I fa de pont entre la classe Prestatgeria i el seu driver corresponent.

2.2.7 Controlador Producte

El controlador de Producte s'encarrega de la gestió dels mètodes de la classe Producte. Es comunica amb Producte per tal de realitzar totes les funcionalitats que fan referència a aquest.

2.2.8 Controlador Producte Col·locat

El controlador de Producte Col·locat no deixa de tenir la mateixa funcionalitat que el controlador de Producte amb Producte, però aquest ho fa amb la clase Producte Col·locat.

2.2.9 Controlador Distribució

El controlador de Distribució és qui controla o té la majoria de lògica del programa, qui fa de pont entre el driver de distribució i la classe distribució on tenim la majoria de funcionalitats importants, com la de càlcul de distribució, la de mostrar la distribució per pantalla…

2.2.10 Controlador Perfil

El controlador de Perfil s'encarrega de la gestió dels mètodes de la classe Perfil. Es comunica amb Perfil per tal de realitzar totes les funcionalitats que fan referència a aquest. I fa de pont entre la classe Perfil i el seu driver corresponent.

2.2.11 Estrategia Cálculo

La estratègia “Estrategia Cálculo” és una interfície (no té codi). Serveix de pont entre la classe distribució i les implementacions dels algorismes de càlcul.

2.2.12 Estrategia TSP

La estratègia TSP es una clase que implementa la interfície “Estrategia Calculo” d'acord amb el patró Estratègia de disseny de Software. En aquesta classe es fa el càlcul de la nostra solució al problema del TSP adaptat al nostre problema de trobar la distribució amb les màximes similituds. Aquesta classe retorna una llista de productes ordenada i els productes que la componen ja tenen la seva posició i altura dels seus atributs actualitzades.

Els mètodes d’aquesta clase son:

- List<ProducteColocat> arrangeProductsBySimililarity(List<ProducteColocat>, int): Ordena els productes per similitud utilitzant un arbre de cobertura mínima (MST) i un cicle hamiltonià.
- List<ProducteColocat> prepara_salida(List<ProducteColocat>, ArrayList<String>) :Prepara la sortida convertint el cicle hamiltonià de cadenes a text a objectes `ProducteColocat`.
- ArrayList<Edge> generateEdges(List<ProducteColocat>):Genera arestes entre els productes basant-se en la seva similitud.
- private ArrayList<Edge> prim(ArrayList<Edge> edges, List<ProducteColocat>):Implementa l'algorisme de Prim per generar un arbre de cobertura mínima (MST) a partir de les arestes.

- int calcula_similitud(ArrayList<String>, ArrayList<Edge>):Calcula la puntuació de similitud per a una solució donada.
- int get_similitud_2_productes(String s, String s1, ArrayList<Edge>):Obté la similitud entre dos productes.
- Set<String> calcula_numero_nodos(ArrayList<Edge>) :Calcula el nombre de nodes únics en el cicle eulerià.
- cicloHamiltoniano(ArrayList<Edge>,ArrayList<Edge>, ArrayList<ProducteColocat>): Genera un cicle hamiltonià a partir del cicle eulerià.
- get_arista(Set<String>, String, ArrayList<Edge>): Obté la millor aresta per continuar el cicle hamiltonià.

  -actualitza_posicions(List<ProducteColocat>, int): Actualitza les posicions dels productes basant-se en la solució i l'altura.

2.2.14 Edge

Edge és una classe auxiliar que hem creat per a poder realitzar l'algorisme que implementa la nostra solució al problema TSP que hem implementat per a ordenar la nostra distribució.
La classe Edge consta de 2 strings, un string to i un string from que representen els dos productes als extrems de l'artista i un valor "peso" que indica la similitud entre els 2 productes.

Els atributs son:

- From(String): Fa referència a un producte de la similitud.
- To(String): Fa referència a l'altre producte de la similitud.
- Peso(Integer): Integer que representa la similitud en dos productes.

Els mètodes son:

- int compareTo(Edge o):

2.2.13 Estrategia OneToOne

La estrategia One to One és una classe que implementa la interfície "Estratègia Cálculo". La utilitzem en el test de la classe Distribució, no deixa de ser un stub de la classe de càlcul que retorna la mateixa distribució que se li ha ficat.

Els mètodes son:

- List<ProducteColocat>arrangeProductsBySimilarity(List<ProducteColocat>, int): Aquest mètode l'utilitzem per a fer tests i retorna la mateixa distribució d'entrada.

2.2.15 CalculBackTracking

La classe càlcul BackTracking no deixa de ser una altra Estratègia de càlcul que implementa la interfície “Estrategia Cálculo”. Aquesta calcula amb un algorisme de força bruta la millor distribució possible, la que té les similituds màximes, a canvi de trigar exponencialment i, per tant, amb grans quantitats de productes pot trigar bastant a fer el càlcul.

Aquesta classe no te atributs específics, llavors els seus mètodes serien:

- arrangeProductsBySimilarity(List<ProducteColocat> productes, int altura):
  Aquesta funció s'encarrega de disposar els productes en funció de la seva similitud. El procés inclou calcular quantes columnes (altura) es poden fer per distribuir els productes i generar permutacions dels productes no modificats manualment.
  Després es calcula la disposició amb la millor similitud total, tenint en compte els productes adjacents i el seu percentatge de similitud.
  Finalment, es retornen els productes disposats amb les posicions i altures ajustades.

- generarPermutaciones(ProducteColocat[] productos, int index, int size, List<ProducteColocat[]> permutaciones):

  Aquesta funció genera totes les permutacions possibles dels productes no modificats manualment.
  Utilitza un algorisme recursiu per intercanviar els elements i crear les permutacions, les quals es guarden en una llista.

- calculateSimilarity(Producte p1, Producte p2):
  Calcula la similitud entre dos productes. Utilitza el mètode getSimilitud de la classe Producte per obtenir la similitud entre els productes p1 i p2.
  Si algun dels productes és nul, retorna 0 com a similitud.

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

A continuació es mostra una taula amb el repartiment de les classes que ha implementat cada membre de l'equip.

Eric Díez	Pol Carnicer	Aleix Montero	Francesc Pérez
Càlcul BackTracking	Perfil	Ctrlr Distribució	Estrategia TSP i Test Estrategia TSP
Ctrlr Producte Col·locat	Ctrlr Prestatgeria	Distribució	Estrategia One to One
Producte Col·locat	Prestatgeria	Estrategia Càlcul	Producte
Test Producte Col·locat	Ctrlr Perfil	Edge	Ctrlr Producte
Test Càlcul BackTracking	Test Perfil	Test Distribució	Test Producte
Main	Driver Perfil	Driver Distribució	Driver Producte

4. Estructures de dades i algorismes utilitzats

Aquí s'explica cadascuna de les estructures de dades que hem necessitat per crear les nostres prestatgeries i distribucions, i els algorismes utilitzats per trobar la millor distribució possible de productes segons les seves similituds.

4.1 Estructures de dades

4.1.1 ArrayList

Un ArrayList és una estructura de dades que implementa una llista dinàmica, proporcionant una forma eficient de gestionar col·leccions d'elements que poden canviar de mida durant l'execució d'un programa. A diferència d'un array tradicional, que té una mida fixa, un ArrayList pot ampliar-se automàticament quan es necessita més espai per emmagatzemar nous elements.

Aquesta estructura està basada en un array intern, però permet afegir o eliminar elements sense preocupar-se per la capacitat inicial. Internament, quan l'ArrayList arriba a la seva capacitat màxima, es redimensiona automàticament per adaptar-se a més elements.

Hem utilitzat ArrayList per a gairebé tot, ja que ens sembla una estructura de dades molt versatil per exemple, la nostra classe distribució té un ArrayList de Productes Col·locats o com la clase Perfil té un ArrayList de Prestatgeries.

4.1.2 HashMap

Un HashMap és una estructura de dades que emmagatzema parelles clau-valor, on cada clau és única i s'associa a un valor específic. Aquesta estructura permet una cerca, inserció i eliminació molt ràpides, ja que els elements es distribueixen en buckets (contenidors) utilitzant una funció hash. El HashMap és ideal quan es vol accedir ràpidament a un valor mitjançant la seva clau associada, en comptes d'iterar a través de tota la col·lecció.

En el nostre projecte, utilitzem un HashMap per gestionar la similitud entre productes. En lloc de tenir una estructura lineal on es comparen tots els productes entre si, cada Producte té un HashMap associat on les claus són altres Productes i els valors són les similituds entre aquests productes.

4.1.3 List

Una llista és una estructura de dades que permet emmagatzemar elements de manera ordenada, on cada element es guarda en un node que conté el valor de l'element i una referència (enllaç) al següent element de la llista.

A diferència d'altres estructures com els arrays, les llistes no tenen una mida fixa i poden créixer o disminuir de manera dinàmica a mesura que s'afegeixen o eliminen elements.
Diferents membres de l'equip han escollit ArrayList o List indiferentment a l'hora de programar les seves clases.

4.1.4 PriorityQueue

Una Priority Queue és una estructura de dades que opera de manera similar a una cua normal, però amb una diferència clau: els elements que s'afegeixen a la cua tenen associada una prioritat. Els elements amb més alta prioritat es processen abans que aquells amb menor prioritat, independentment de l'ordre en què van ser afegits a la cua.

La PriorityQueue l'hem utilitzat per a realitzar el càlcul del Maximum Spanning tree. Aquest càlcul és part de l'algorisme d'aproximació del TSP que hem implementat.

4.1.5 Set

Un Set és una estructura de dades que emmagatzema elements únics, és a dir, no permet duplicats. A diferència d'altres col·leccions com les llistes o les cues, un Set no manté l'ordre dels elements, però garanteix que cada element només aparegui una vegada. Aquesta estructura és útil quan es necessita assegurar que no hi hagi repeticions de valors en la col·lecció.

Aquesta estructura de dades l'hem utilitzat per a l'algoritme d'aproximació del problema del TSP, per a la comprovació de nodes visitats, ja que les funcions de add() i de contains() ens serveixen molt per a la comprovació de nodes visitats en l'algoritme.

4.1.6 Array -> String [] []

Un array bidimensional de strings és una estructura de dades que emmagatzema elements de tipus string en forma de matriu, és a dir, una taula amb files i columnes. Aquesta estructura es pot veure com un conjunt de llistes d'element d'un tipus específic, on cada fila pot tenir un nombre variable de columnes. Els arrays bidimensionals es poden utilitzar per representar dades tabulars o taules de valors.

En el cas concret de String[][], cada element de la matriu és una cadena de text, i l'indexació es fa mitjançant dues dimensions: una per accedir a les files i una altra per les columnes. La mida de l'array es defineix quan es crea l'array, però un cop creat, no pot variar.

Aquesta estructura de dades l'hem utilitzat sobretot per a passar les dades entre la capa de domini a la capa de presentació, en aquest cas els drivers.

4.2 Algorismes

4.2.1 Algorisme de Backtracking

El backtracking és una tècnica de cerca que explora totes les possibles solucions a un problema de manera sistemàtica i retrocedeix quan troba un camí que no condueix a una solució vàlida. Aquest enfocament és especialment útil per resoldre problemes d'optimització i combinatòria, però és molt costós computacionalment amb temps exponencial.

En el nostre projecte, aquest algorisme de força bruta, s'utilitza per trobar la millor distribució possible de productes en una prestatgeria, on s'intenta que tots aquests sempre estiguin al costat d'un altre producte amb el qual tinguin una similitud elevada. Aquest algorisme calcula la millor distribució a diferència de l'algorisme implementat que no ho calcula perfectament, però té un cost molt superior computacionalment respecte l'altre.

4.2.1 Algorisme Travelling Salesman Problem (TSP)

El TSP (Travelling Salesman Problem) és un problema d'optimització que, originalment, consisteix a trobar el camí de cost mínim que passi per cada una de les ciutats i retorni al punt d'origen. En el nostre cas hem resolt el problema aplicant el mateix principi, però en comptes de trobar el camí de cost mínim, hem calculat el camí de cost màxim.

Com és un problema NP-complet, no coneixem cap manera de resoldre el problema en temps polinòmic.

Nosaltres hem implementat la nostra versió de la 2-Aproximació del TSP, obtenint un Maximum Spanning Tree amb l'algoritme de Prim del nostre graf d'entrada que hem creat, ja que nosaltres estem treballant amb una Llista de Productes i no amb un graf. Amb el resultat, que és un conjunt d'arestes ponderades no dirigides, hem construït un graf dirigit, fent que cada aresta no dirigida passin a ser 2 arestes dirigides ponderades. Amb això ja hem obtingut un cicle eulerià, però ara mateix apareixen nodes repetits, així que hem calculat un cicle hamiltonià. L'algorisme del càlcul del cicle hamiltonià que hem implementat recorre el graf dirigit ponderat i construeix un cicle hamiltonia per cada node del graf i es queda amb el de major pes. D'aquesta manera obtenim una distribució que és pitjor que la calculada amb "Backtracking" però segons els nostres tests triga molt menys.

D'aquesta manera obtenim una distribució representada en una llista, on els productes de la llista tenen similitud màxima.

Cada producte col·locat té un atribut que és posició i altura que es calcula posteriorment al càlcul.