PRESENTACIÓ DE CAPES

Professor:
Iván Paz
Assignatura:
Projecte de Programació
Subgrup:
Subgrup-prop 41.5
Autors:
Axel David González Saldivar
axel.david.gonzalez@estudiantat.upc.edu
Marcel Masip Bagán
marcel.masip@estudiantat.upc.edu
Aleix Parrilla Martín
aleix.parrilla@estudiantat.upc.edu
Versió:
Lliurament versió 2.1

Facultat d'Informàtica de Barcelona (FIB)
Universitat Politècnica de Catalunya (UPC) - BarcelonaTech
2024

ÍNDEX

DESCRIPCIÓ DE CAPES

Descripció de Classes de la Capa de Domini

La capa de Domini és l'encarregada d'implementar la lògica per a fer que l'aplicació funcioni. Per tant és el conjunt d'operacions que manipula les dades i aplica les regles definides en les classes.

Classes principals:

- Producte

La classe producte defineix com es representa un producte i com es gestionen les seves similituds amb altres productes. Així doncs un producte té un nom i un Array de Doubles que seran les similituds. En aquesta classe es defineixen les regles basades en les similituds amb la funció similitudsCorrectes(), on ens assegurem que no poden ser null, i que han d'estar entre 0 i 1.

- Prestatge

La classe prestatge controla la distribució de productes. Així doncs, una instància de un prestatge representa el resultat dels algorismes, és a dir, una distribució circular amb similituds maximitzades entre els productes, on a més, no poden haver-hi productes repetits. Aquesta classe també ens permet modificar el resultat, intercanviar posicions dels productes i consultar els productes del prestatge.

- DistribucióKruskal

A partir de l'entrada donada, genera un prestatge de productes circular, de manera que el còmput total de similituds calculada a partir dels productes adjacents és màxim. En aquesta classe es troben els mètodes que comporten la 2-aproximació. I finalment la funció generarPrestatge() que ens donarà el prestatge final.
En la primera funció definirem regles de domini per a generar el Minimum Spanning Tree (en aquest cas Maximum): construirMST()

- Un MST no pot tenir cicles
    - S'eviten cicles utilitzant la funció find que ajuda a mantenir la connexió de components disjunts.
- Ha de connectar tots els productes
- Ha de maximitzar les similituds entre els productes
- Requereix com a mínim dos productes

En la segona funció es defineixen les regles sobre el cicle euleria generaCicleEuleria(List<Aresta> MST) :

- Es necessiten com a mínim tres productes per a poder formar-lo
- S'ha de passar exactament un cop per cada aresta que connecta dos productes
- Es dupliquen els vèrtex per assegurar que cada producte tingui grau parell (poden haver-hi productes duplicats)
- El resultat ha de ser un cicle tancat, és a dir, s'ha de tornar al producte inicial

La última funció recull els resultat dels mètodes per a conformar el prestatge final, de manera que es defineixen les regles sobre el cicle Hamiltonià:

- S'ha de visitar cada producte una vegada
    - S'eliminen els productes repetits que hi havien en el cicle Eulerià comportant una reducció de costos basada en la desigualtat triangular
- S'ha de tornar al vèrtex inicial (ha de ser un únic cicle)
- Es mantenen les regles establertes anteriorment

- <u>AlgorismeFB</u>

La classe AlgorismeFB implementa un algorisme de força bruta per trobar la millor solució possible per a la distribució del prestatge. És un algorisme robust peoò ineficient en escenaris amb grans quantitats de dades.

- La distribució trobada ha de maximitzar la similitud entre productes adjacents
- La distribució trobada es considera cíclica, sigui N productes, es considera també la similitud de producte[N-1] → producte[0]
- Cada producte apareix una vegada
- No es considera cap ordre previ entre els productes
- El backtracking assegura la completitud i exactitud de l'algorisme

Descripció de Classes de la Capa de Presentació

La capa de Presentació és l'encarregada de la interacció entre l'usuari i l'aplicació. La seva funció principal és mostrar la informació de forma comprensible i atractiva, així com rebre les accions o inputs de l'usuari. Aquesta capa tradueix les dades provinents de la lògica de negoci a formats visuals (interfícies gràfiques, pàgines web, etc.) i transmet les accions de l'usuari a les altres capes perquè siguin processades.

Classes Principals:

- PrincipalView

Aquesta classe és la classe principal de la interfície de la nostra aplicació. És la pantalla principal, i l'encarregada de cridar a cada un dels panells que formen aquesta interfície. Proporciona la comunicació entre la visualització de la pantalla i el controlador de presentació.

- MainView

La classe MainView serveix com a punt d'entrada inicial a la interfície d'usuari, actua com a pantalla inicial, el primer panell que veu l'usuari. La crida la PrincipalView. La seva funció és establir la navegació i els punts d'accés als mètodes principals de l'aplicació. És com un "menú principal" de l'aplicació, un panell inicial que facilita la navegació. La seva responsabilitat és oferir una interfície d'accés senzilla i organitzada per l'usuari. Està formada per 5 botons, els quals la seva funcionalitat és cridar a altres panells, aquests botons són els següents: Gestionar Productes, Generar Prestatge, Mostrar Cistella, Mostrar Prestatge i Tancar (el botó que surt i tanca l'aplicació).

- GenerarPrestView

Aquesta classe és una altra vista funcional de l'aplicació, ofereix a l'usuari dues opcions concretes per generar el prestatge a l'aplicació. Una de les opcions és generar-lo mitjançant l'algorisme de Força Bruta, per tal de dur a terme aquesta acció, l'usuari ha de clicar sobre aquest botó. L'altre opció és generar-lo mitjançant l'algorisme de 2-Aproximació. També tenim l'opció de tornar enrere, per si l'usuari s'ho pensa i finalment no vol generar cap prestatge.

- PrestatgeView

Aquesta classe és l'encarregada de mostrar el prestatge un cop aquest s'ha generat. Facilita la interacció de l'usuari amb les dades associades al prestatge i proporciona accions directes per a la seva gestió. Aquestes accions són les de guardar el prestatge, o les de modificar-lo.

- ModificarPrestatgeView

Aquesta classe és una vista dedicada a la modificació del prestatge dins de l'aplicació. Actua com una eina que permet a l'usuari veure l'estat actual del prestatge, fer ajustos i guardar o no els canvis. És una vista essencial per garantir que els usuaris puguin modificar la distribució del prestatge i guardar les dades a l'aplicació.

- MostrarPrestatgeView

La classe MostrarPrestatgeView és una altra vista funcional dins d'aquesta capa. Permet a l'usuari veure la distribució del prestatge guardat en el sistema, i ofereix opcions per poder modificar o eliminar aquest prestatge. Com altres vistes també té l'opció de tornar enrere, per si l'usuari vol sortir d'aquesta pantalla sense realitzar cap actualització en l'aplicació.

- EliminarPrestatgeView

Aquesta classe proporciona a l'usuari una interfície per confirmar l'eliminació d'un prestatge. Aquest panell actua com un diàleg de confirmació per evitar eliminacions accidentals. Això garanteix que l'eliminació d'un prestatge es realitzi de manera intencionada, donant a l'usuari l'opció de confirmar o cancel·lar l'acció.

- MostrarCistellaView

La classe MostrarCistellaView és una vista que permet a l'usuari visualitzar el contingut de la cistella, és a dir, l'usuari veu tots els productes que estan guardats en la base de dades de l'aplicació. Aquest panell és exclusiu per veure només els productes de l'aplicació, l'usuari té l'opció de tornar enrere per anar a la pantalla principal.

- GestionarProdView

Aquesta classe permet a l'usuari fer la gestió dels productes, el panell mostra tots els productes que hi ha guardats a l'aplicació. Aquesta vista proporciona la possibilitat d'afegir productes a la base de dades de l'aplicació, aquesta acció t'envia al panell corresponent per afegir un o més prodcutes. Així mateix, l'usuari seleccionant un producte en aquesta vista podrà accedir a un panell de selecció de l'article, on el podrà modificar (modificar les seves característiques) o eliminar de la base de dades de l'aplicació.

- SeleccioProdView

La classe SeleccioProdView és un panell en el què l'usuari pot modificar el producte que hagi seleccionat (això li envia a una altra pantalla), o pot eliminar el producte definitivament de la base de dades. També es proporciona a l'usuari la opció de sortir d'aquesta finestra, per si al final no vol ni modificar ni eliminar el producte que ha seleccionat.

- ModificarProdView

Aquesta classe és l'encarregada de mostrar a l'usuari una interfície en la que pot modificar les característiques del producte prèviament seleccionat. En aquesta vista es mostren a l'usuari les similituds del producte. L'usuari pot seleccionar la que posteriorment voldrà modificar. Aquesta acció l'envia a un panell de modificació exclusiu per al grau de similitud del producte. També es permet a l'usuari sortir de la finestra si finalment no vol modificar cap característica del producte seleccionat.

- SeleccioSimilitudView

La classe SeleccioSimilitudView és un panell en el que l'usuari ha d'escriure el nou grau de similitud que vol canviar pel grau prèviament seleccionat del producte també anteriorment seleccionat. Es permet a l'usuari sortir d'aquest panell si finalment no vol modificar el grau de similitud seleccionat.

- AfegirProducteView

Aquesta classe és una vista que proporciona la possibilitat d'afegir un producte a mà, mitjançant la funcionalitat de l'aplicació, o també afegir un o més productes mitjançant fitxers de text que l'usuari adjunta. Si l'usuari decideix afegir el producte a mà, haurà d'escriure el nom del nou producte i els nous graus de similitud d'aquest mateix. Aquest panell proporciona a l'usuari l'opció de guardar el producte a la base de dades de l'aplicació. També es permet a l'usuari sortir de la finestra, per si finalment no vol afegir cap producte en el sistema.

Descripció de Classes de la Capa de Persistencia

La Capa de Persistència serà, en essència, la que s'ocuparà de la gestió d'emmagatzematge d'arxius persistens del projecte. Serà capaç de llegir i escriure les classes Producte i Prestatge (en concret, arxius .json).

Classes principals:

- Gestor_Producte

Aquesta classe és com el gestor de fitxers dels productes. Tot el que tingui a veure amb guardar o carregar informació d'un producte ho fa aquesta classe. Quan es crea un producte nou, ella s'encarrega de fer-li un fitxer JSON i guardar-lo al directori *data/productes*. Si es vol recuperar un producte, només cal demanar-ho, i aquesta classe llegeix el fitxer corresponent i el torna a l'aplicació.
De forma similar, també podrà modificar un producte, per exemple la seva similitud, farà una lectura del JSON, precisarà i efectuarà els canvis necessaris, i tornarà a desar-lo.

També pot esborrar productes, que en realitat és tan simple com esborrar el fitxer JSON del producte en qüestió. A més, si s'ha de canviar alguna cosa, com el grau de similitud entre productes, aquesta classe ho actualitza al fitxer. Tot això ho fa de manera força automàtica utilitzant la llibreria Gson per treballar amb fitxers JSON.

- Gestor_Prestatge

Aquesta classe és molt semblant a *Gestor_Producte*, però en lloc de gestionar productes, gestiona prestatges (en el nostre cas, només un). La seva feina és desar el prestatge sencer en un únic fitxer JSON i recuperar-lo quan sigui necessari. També pot eliminar-lo, esborrant el fitxer corresponent.

El prestatge es guarda al directori *data/prestatge*. Si el directori no existeix, aquesta classe el crea. És útil quan necessites tenir tota la informació d'un prestatge guardada o quan vols carregar-la per seguir treballant amb ella. També té la funció de modificar el prestatge, i aquesta modificació es basarà simplement en fer un *swap* de dos ID's de productes.

Fluxe de connexions

Els controladors es comunicaràn entre ells de la següent forma:

Usuari <-> Presentació <-> Domini <-> Persistència <-> Base de Dades

D'aquesta forma tindrem funcionalitats altament predefinides per cada una de les capes, sent cada una la responsable d'un conjunt de funcionalitats concretes que no s'encreuen entre elles.