

Gestor de teclats

Lliurament 1.0

Grup 31.2

Iván López Buira

@ivan.lopez.buira

Javier Vega Centeno

@javier.vega.centeno

Sergi Navarra Parés

@sergi.navarra

Miquel Amorín Díaz

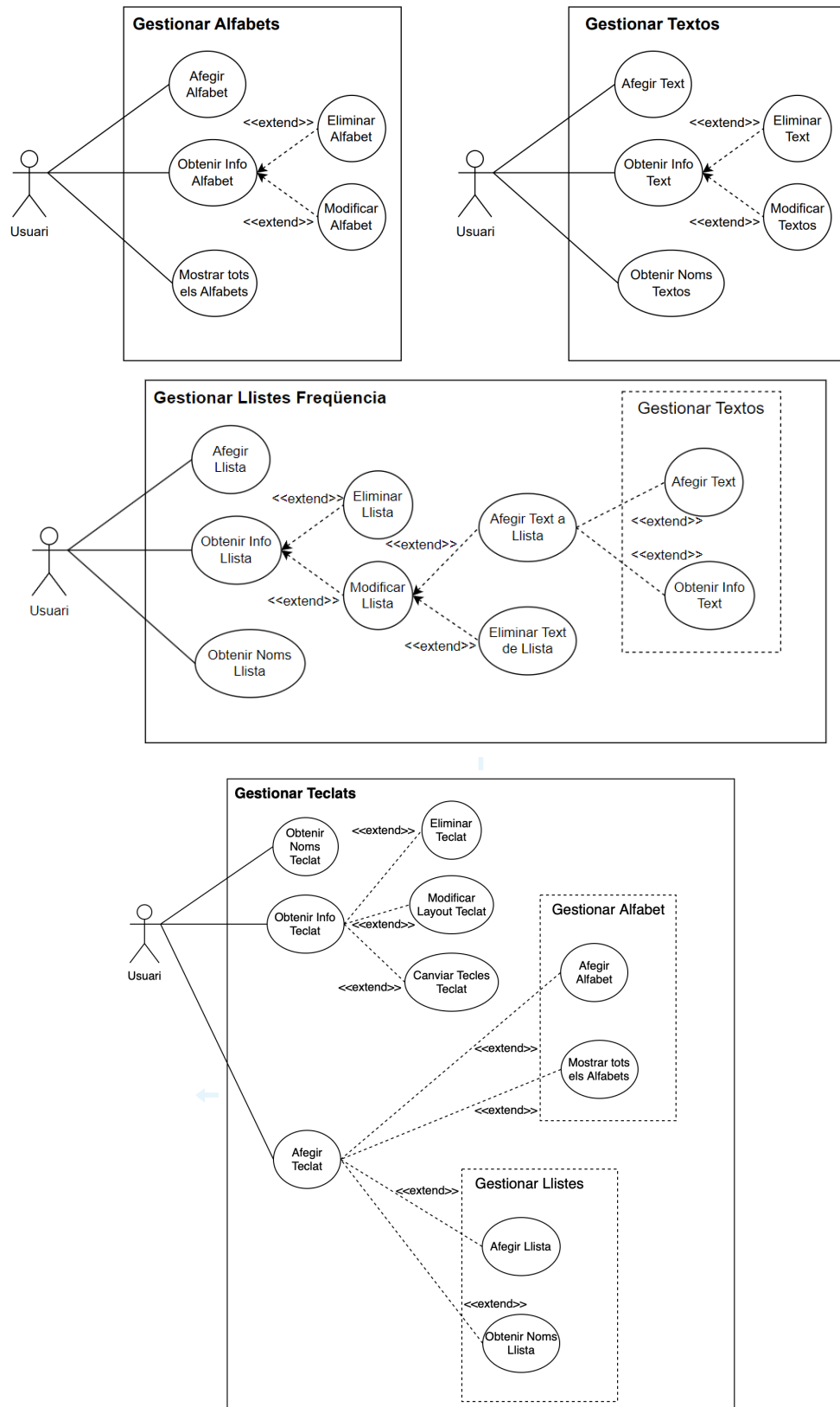
@miquel.amorin

Índex

1. Diagrama de casos d'ús.....	3
a. Diagrama de casos d'ús.....	3
b. Descripció dels casos d'ús.....	4
2. Diagrama del model conceptual.....	11
a. Diagrama.....	11
b. Descripció de les classes.....	12
Teclat.....	12
Text.....	12
Alfabet.....	12
ParaulesFrequencia.....	13
CtrlDomini.....	13
CtrlTeclat.....	14
CtrlText.....	14
CtrlAlfabet.....	15
CtrlParaulesFrequència.....	15
Posició.....	15
Pair.....	15
Node.....	15
CompNode.....	16
CreadorTeclat.....	16
BranchAndBound.....	16
HungarianAlgorithm.....	16
LectorFitxer.....	16
ParellsLletres.....	17
CtrlParellsLletres.....	17
3. Estructures de dades i algorismes utilitzats.....	18
a. ParellsLletres.....	18
b. GestorDades.....	18
c. Teclat.....	18
d. BranchAndBound.....	18
e. HungarianAlgorithm.....	18

1. Diagrama de casos d'ús

a. Diagrama de casos d'ús



b. Descripció dels casos d'ús

GESTIONAR ALFABETS:

Nom: Afegir Alfabet

Actor: Usuari

Comportament:

1. L'Usuari indica al sistema que vol crear un nou alfabet
2. El sistema mostra les opcions per introduir el nou alfabet:
 - a. Introducció Manual
 - i. El sistema demana a l'usuari una serie de caracteres en una mateixa línia.
 - b. Importar de Fitxer
 - i. El sistema demana a l'usuari el path del fitxer on es troba l'alfabet a importar.
3. L'usuari aporta tota la informació necessària per crear el L'Alfabet
4. El sistema indica a l'usuari que l'alfabet s'ha creat satisfactòriament.

Errors possibles i cursos alternatius:

- 3a. El nom de l'alfabet introduït ja existeix. El sistema indica l'error. Torna al punt 3.

Nom: Obtenir Info Alfabet

Actor: Usuari

Comportament:

1. L'usuari indica al sistema que vol obtenir la informació d'un alfabet en concret.
2. El sistema mostra la informació de l'alfabet (nom i contingut).
3. El sistema mostra l'opció d'eliminar (**Cas d'ús Eliminar Alfabet**) i modificar (**Cas d'ús Modificar Alfabet**)

Nom: Eliminar Alfabet

Actor: Usuari

Comportament:

1. L'usuari indica al sistema que vol eliminar l'alfabet.
2. El sistema demana a l'usuari un el nom de l'alfabet del que vol eliminar.
3. L'Usuari introdueix el nom de l'alfabet objectiu.
4. El sistema indica que l'alfabet s'ha eliminat satisfactòriament.

Errors possibles i cursos alternatius:

- 3a. L'alfabet amb el nom introduït per l'usuari no existeix. El sistema indica l'error. Torna al punt 3.

Nom: Modificar Alfabet

Actor: Usuari

Comportament:

1. L'Usuari indica al sistema que vol modificar l'alfabet.
2. El sistema demana a l'usuari un el nom de l'alfabet del que vol eliminar.
3. L'usuari aporta tota la informació necessària per modificar l'alfabet (nom i contingut).
4. El sistema indica que l'alfabet s'ha modificat satisfactòriament.

Errors possibles i cursos alternatius:

3a. L'alfabet amb el nom introduït per l'usuari no existeix. El sistema indica l'error. Torna al punt 3.

Nom: Mostrar tots els Alfabets

Actor: Usuari

Comportament:

1. L'usuari indica al sistema que vol obtenir tots els alfabets.
2. El sistema mostra a l'usuari tots els alfabets amb el seu nom i el seu contingut.

GESTIONAR TECLATS:

Nom: Afegir Teclat

Actor: Usuari

Comportament:

1. L'usuari indica que vol crear un teclat introduïnt el seu nom
2. El sistema mostra:
 - a. La llista de tots els alfabets (**cas d'ús de Mostrar tots els Alfabets**).
 - b. Una llista de les Llistes de Paraules Freqüències (**cas d'ús d'Obtenir Noms Llista**).
 - c. Opció d'afegir un Alfabet (**cas d'ús d'Afegir Alfabet**)
 - d. Opció d'afegir una Llista (**cas d'ús d'afegir Llista**)
3. L'usuari indica si vol afegir algun Alfabet o Llista i, al acabar, selecciona els desitjats.
4. El sistema mostra una llista de Layouts possibles i una llista d'algoritmes.
5. L'usuari selecciona el layout i l'algoritme desitjat.
6. El sistema crea el teclat, mostra un missatge de creació satisfactòria i mostra el contingut (**cas d'ús d'Obtenir Info Teclat**)

Errors possibles i cursos alternatius:

- 1a. El teclat ja existeix, el sistema mostra un missatge d'error, tornem al punt 1.

Nom: Obtenir Noms Teclats

Actor: Usuari

Comportament:

1. L'usuari indica al sistema que vol una llista amb els noms de tots els teclats.
2. El sistema retorna una llista amb els noms de tots els teclats.
3. L'usuari pot clicar a un dels teclats (**cas d'ús de Obtenir info Teclat**)

Nom: Obtenir Info Teclat

Actor: Usuari

Comportament:

1. L'usuari indica al sistema que vol mostrar la distribució del teclat indicant el seu nom.
2. El sistema retorna la distribució del teclat i mostra les opcions d'eliminar, modificar layout o intercanviar dues tecles
3. L'usuari escull l'opció desitjada
4. Anem al cas d'ús d'eliminar, modificar layout o intercanviar dues tecles d'un teclat

Nom: Eliminar Teclat

Actor: Usuari

Comportament:

1. L'usuari indica al sistema que vol eliminar el Teclat.
2. El sistema elimina el Teclat i mostra un missatge d'eliminació satisfactoria.

Nom: Modificar Layout Teclat

Actor: Usuari

Comportament:

1. L'usuari indica al sistema que vol canviar el layout d'un teclat donant el nom d'un Teclat.
2. El sistema mostra els layouts possibles.
3. L'usuari selecciona el layout desitjat.
4. El sistema modifica el layout del teclat i recalcula la distribució, mostra un missatge de modificació satisfactoria

Nom: Canviar Tecles Teclats

Actor: Usuari

Comportament:

1. L'usuari indica al sistema que vol canviar dues tecles donant el nom d'un Teclat i dues Posicions.
2. El sistema intercanvia els caràcters de les posicions designades, mostra un missatge de modificació satisfactoria

Errors possibles i cursos alternatius:

- 2b. La posició no és vàlida, el sistema llença un missatge d'error, tornem al punt 1.

GESTIONAR TEXTS:

Nom: Afegir Text

Actor: Usuari

Comportament:

1. L'Usuari indica al sistema que vol crear un nou Text
2. El sistema mostra les opcions per introduir el nou Text:
 - a. Introducció Manual
 - i. El sistema demana a l'usuari una serie de caracteres en una mateixa línia.
 - b. Importar de Fitxer
 - i. El sistema demana a l'usuari el path del fitxer on es troba el Text a importar.
3. L'usuari aporta tota la informació necessària per crear el Text.
4. El sistema indica que s'ha creat satisfactòriament el Text.

Errors possibles i cursos alternatius:

- 3a. El nom introduït ja existeix. El sistema notifica l'error. Torna al punt 3.

Nom: Obtenir Noms Textos

Actor: Usuari

Comportament:

1. L'usuari indica al sistema que vol una llista amb els noms de tots els Textos.
2. El sistema retorna una llista amb els noms de tots els Textos.
3. L'usuari pot clicar a un dels teclats (**cas d'ús de Obtenir info Text**)

Nom: Obtenir Info Text

Actor: Usuari

Comportament:

1. L'usuari indica al sistema que vol veure el contingut d'un text
2. El sistema mostra el contingut del text. A més mostra els botons eliminar i modificar
3. L'usuari clica l'opció desitjada (**cas d'ús Eliminar Text**) o (**cas d'ús de Modificar Text**)

Errors possibles i cursos alternatius:

Nom: Eliminar Text

Actor: Usuari

Comportament:

1. L'usuari indica al sistema que vol eliminar un text
2. El sistema elimina el Text i mostra un missatge d'eliminació satisfactoria

Nom: Modificar Text

Actor: Usuari

Comportament:

1. L'usuari indica al sistema que vol modificar un Text indicant el seu nou contingut
2. El sistema modifica el Text i mostra un missatge d'eliminació satisfactoria

GESTIONAR LLISTES FREQUÈNCIA:

Nom: Afegir Llista

Actor: Usuari

Comportament:

1. L'Usuari indica al sistema que vol crear una nova Llista de freqüències.
2. El sistema mostra les opcions per crear la llista:
 - a. Introducció Manual
 - i. El sistema demana a l'usuari que introdueixi una serie de paraules amb la seva freqüència amb el format: "paraula:freqüència; ..."
 - b. Importació Fitxer
 - i. El sistema demana a l'usuari el path del fitxer on es troba la llista a importar.
 - c. Creació a partir de Textos
 - i. El sistema mostra una serie de Textos (**cas d'ús de Obtenir Noms Textos**)
3. L'usuari aporta tota la informació necessària per crear la Llista.
4. El sistema indica que la Llista de freqüències s'ha creat satisfactòriament.

Error possible i cursos alternatius:

- 2a. El format introduït és incorrecte. El sistema notifica de l'error. Torna al punt 2.
- 2b. El path no existeix. El sistema notifica de l'error. Torna al punt 2.
- 3a. El nom de la Llista de Freqüències introduït ja existeix. El sistema indica l'error. Torna al punt 3.

Nom: Obtenir Noms Llistes

Actor: Usuari

Comportament:

1. L'usuari indica al sistema que vol una llista amb els noms de totes les llistes.
2. El sistema retorna una llista amb els noms de totes les llistes de Freqüència.
3. El sistema ofereix la opció d'obtenir informació d'alguna de les llistes (**Cas d'ús Obtenir Info Llista**)

Nom: Obtenir Info Llista

Actor: Usuari

Comportament:

1. L'usuari indica que vol mostrar la informació d'una llista
2. El sistema mostra el contingut de la llista i mostra les opcions d'eliminar i modificar
3. L'usuari escull l'opció desitjada
4. Anem al cas d'ús d'eliminar llista o modificar llista

Nom: Eliminar Llista

Actor: Usuari

Comportament:

1. L'usuari indica al sistema que vol eliminar una llista
2. El sistema elimina la llista
3. El sistema indica que la llista de freqüències s'ha eliminat satisfactòriament.

Nom: Modificar Llista

Actor: Usuari

Comportament:

1. L'usuari indica que vol modificar una llista
2. El sistema mostra el contingut de la llista (cas d'ús d'Obtenir info llista) i els textos amb els que està feta.
3. L'usuari modifica el contingut o indica que vol eliminar (cas d'ús d'Eliminar text de llista) o afegir (cas d'ús d'Afegir text a llista).
4. El sistema modifica la llista.
5. El sistema indica que la llista de freqüències s'ha modificat satisfactòriament.

Nom: Afegir Text a Llista

Actor: Usuari

Comportament:

1. L'usuari indica que vol afegir un text a una llista
2. El sistema mostra una llista dels textos disponibles (Cas d'ús d'Obtenir Noms Textos)
3. L'usuari selecciona el text desitjat
4. El sistema l'afegeix a la llista.
5. El sistema indica que la llista de freqüències s'ha modificat satisfactòriament.

Nom: Eliminar Text de Llista

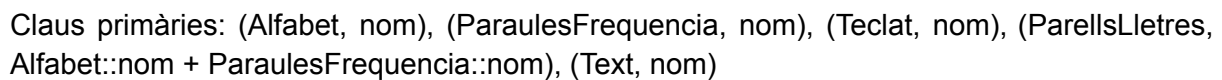
Actor: Usuari

Comportament:

1. L'usuari indica que vol eliminar un text del conjunt dels textos amb els que s'ha creat una llista.
2. El sistema mostra una llista dels textos que han format la llista.
3. L'usuari selecciona el text desitjat
4. El sistema elimina la llista i notifica que el text s'ha eliminat satisfactòriament.
5. El sistema indica que la Llista de freqüències s'ha modificat satisfactòriament.

a. Diagrama

javier.vega.centeno miquel.amorin
sergi.navarra ivan.lopez.buira



b. Descripció de les classes

Teclat

Resum: És la classe que conté tots els elements que formen un Teclat. També té una instància de ParellsLletres i una de CreadorTeclat, amb les què es relaciona, per una banda, crear la matriu de fluxos i per l'altra decidir l'algoritme a utilitzar.

Atributs:

- **char[] Tecles:** Array que conté la distribució de tecles
- **int amplada, int altura:** Integers que defineixen l'alçada i amplada del teclat
- **String nom:** El nom del teclat que l'identifica
- **ParellLletres pl:** instància de la classe ParellLletres que calcula la matriu de flux
- **CreadorTeclat alg:** instància de l'algoritme que s'utilitzarà per calcular les tecles.

Mètodes:

- **Teclat (nom : String, amplada: Integer, altura: Integer, pl: ParellsLletres, alg : CreadorTeclat):** Creadora de Teclat, es defineix el nom, altura, amplada, el parellLletres amb el que obtenir el flux i l'algoritme que s'utilitzarà. Posteriorment es calculen les tecles del teclat.
- **modificarLayoutTeclat(Integer amplada, Integer altura):** modifica l'alçada del Teclat i recalcula la nova distribució
- **canviarTeclesTeclat(Posicio p1, Posicio p2):** intercanvia les lletres de les dues posicions donades.

Text

Resum: La classe Text conté un String que emmagatzema el text i un String que guarda el seu nom.

Atributs:

- **String text:** Emmagatzema el contingut del text
- **String nom:** És el nom que identifica el Text

Mètodes:

- **Text(String nom, String contingut):** Creadora de text amb el nom i el contingut especificat

Alfabet

Resum: La classe Alfabet conté una llista de caràcters representant un alfabet qualsevol, aquest alfabet pot ser introduït per terminal o importat desde un fitxer.

Atributs:

- **Set<Character> caràcters:** una lista de caràcters que representa l'alfabet.
- **String nom:** identifica a l'alfabet.

Mètodes:

- **Alfabet(String nom, Set<Character> chars):** Constructora de la classe alfabet.

ParaulesFrequencia

Resum: La classe Paraules Frequencia, identificada per un nom, conté una serie de Strings associades a un valor, sent aquesta la freqüència amb la que apareix la paraula en la entrada proporcionada, que pot ser una serie de Textos, entrada manual o importada de un fitxer.

Aquesta classe també porta registre dels Textos amb els que s'ha creat amb la capacitat de poder afegir o eliminar algun dels textos.

Atributs:

- **Map<String, Integer> llistaFreq:** Mapa on la clau es la paraula i el valor es la freqüència.
- **String nom:** Nom que identifica a la classe.
- **Set<Text> textos:** Set de textos amb els quals ha sigut creada la instància de ParaulesFrequencia [Pot ser Null].

Mètodes:

- **ParaulesFrequencia(String nom, String ll):** constructora de la classe a partir de strings en format paraula:frequencia;
- **ParaulesFrequencia(String nom, Set<Text> t):** creadora de Paraules Frequencia a partir de una serie de Textos.
- **afegirText(Text t):** Afegir Text al set de textos amb els que s'ha creat la Llista de Paraules frequencia i es recalcula
- **eliminaText(Text t):** idem anterior pero eliminant un text.

CtrlDomini

Resum: La classe CtrlDomini s'encarrega de gestionar la comunicació entre les diferents capes. Conté una instància de si mateixa per fer-la singleton. Es relaciona amb la capa de presentació per comunicar-se amb l'usuari, amb la capa de dades per gestionar la informació enmagatzemada y amb els diferents controladors (CtrlTeclat, CtrlText, CtrlAlfabet, CtrlLlistaFreqüència i CtrlLlistaParells) per implementar les funcionalitats del software. Conté una insància de cada controlador , del lector de fitxer i del gestor de dades.

Mètodes:

- **modificarAlfabet(String nom, String characters):** permet modificar els caràcters associats a un alfabet existent.
- **afegirAlfabet(String nom, String alf):** afegeix un nou alfabet amb el nom i els caràcters especificats.
- **afegirAlfabetImportat(String nom, String path):** importa un alfabet des de l'ubicació del fitxer especificada pel camí *path*.
- **eliminarAlfabet(String nom):** elimina l'alfabet amb el nom especificat.
- **afegirTeclat(String nom, Integer amplada, Integer altura, String nomAlfabet, String nomParaulesFreq):** afegeix un nou teclat amb les característiques especificades.
- **modificarTeclat(String nom, Integer amplada, Integer altura):** modifica les dimensions d'un teclat existent.
- **canviarTecles(String nom, int i1, int j1, int i2, int j2):** canvia les posicions de dues tecles en un teclat existent.
- **eliminarTeclat(String nom):** elimina el teclat amb el nom especificat.
- **modificarText(String nom, String text):** modifica el contingut del text existent amb el nom especificat.

- **afegirTextImportat(String nom, String path):** importa un text des de l'ubicació del fitxer especificada pel camí *path*.
- **afegirText(String nom, String contingut):** afegeix un nou text amb el nom i el contingut especificats.
- **eliminarText(String nom):** elimina el text amb el nom especificat.
- **modificarParaulesFrequencia(String nom, String llista):** modifica les paraules de freqüència de la llista existent amb el nom especificat.
- **afegirParaulesFrequenciaTextos(String nom, Set<String> noms_t):** afegeix una nova llista de paraules de freqüència amb el nom especificat, a partir dels textos donats.
- **afegirTextALlista(String nomLlista, String nomText):** afegeix un text existent a una llista de paraules de freqüència existent.
- **eliminarTextLlista(String nomLlista, String nomText):** elimina un text d'una llista de paraules de freqüència existent.
- **afegirParaulesFrequencia(String nom, String llista):** afegeix una nova llista de paraules de freqüència amb el nom i la llista de paraules especificats.
- **afegirParaulesFrequenciaImportat(String nom, String path):** importa una llista de paraules de freqüència des de l'ubicació del fitxer especificada pel camí *path*.
- **eliminarParaulesFrequencia(String nom):** elimina una llista de paraules de freqüència amb el nom especificat.
- **eliminarParellsLletres(String nomLlista, String nomAlfabet):** elimina el parell de lletres associat a una llista de paraules de freqüència i un alfabet.

CtrlTeclat

Resum: La classe CtrlTeclat s'encarrega de gestionar els objectes de la classe Teclat, crear-los i modificar-los. Conté una instància de si mateixa per fer-la singleton. Es relaciona amb CtrlDomini per rebre les interaccions de l'usuari i amb Teclat per poder crear i modificar els objectes.

Mètodes:

- **crearTeclat(String nom, Integer amplada, Integer altura, ParellsLletres pl, CreadorTeclat alg):** crida a la classe Teclat per crear una instància d'ella.
- **modificarTeclat(Teclat t, Integer amplada, Integer altura):** donat un Teclat, modifica l'altura i l'amplada i calcula la nova distribució.
- **canviarTeclesTeclat(Teclat t, Posicio p1, Posicio p2):** donat un teclat i dues posicions, intercanvia les posicions de les tecles donades.

CtrlText

Resum: La classe CtrlText s'encarrega de gestionar els objectes de la classe Text, crear-los i modificar-los. Conté una instància de si mateixa per fer-la singleton. Es relaciona amb CtrlDomini per rebre les interaccions de l'usuari i amb Text per poder crear i modificar els objectes.

Mètodes:

- **modificarText(Text t, String text):** Donat un text i un contingut, posa el nou contingut al text t.
- **crearText(String nom, String contingut):** Crea un text amb el nom i el contingut especificat.gm

CtrlAlfabet

Resum: Aquesta classe s'encarrega de gestionar les funcionalitats associades als alfabet: crear, modificar. Conté una instància de si mateix per fer-la singleton.

Mètodes:

- **crearAlfabet(String nom, Set<Character> alf):** Crea alfabet
- **modificarAlfabet(Alfabet a, Set<Character> chars):** modifica el alfabet i retorna el mateix per poder afegir-lo al gestor de dades.

CtrlParaulesFreqüència

Resum: La següent classe s'encarrega de gestionar les funcionalitats que envolten a les Llistes de Paraules Freqüència. És la encarregada de crear i modificar les instàncies de Paraules Freqüències.

Mètodes:

- **crearParaulesFreqüència(String nom, String text):** Crea una Llista de Paraules Freqüències a partir d'una string amb el format paraula:freqüència; i la retorna per poder afegir-la al gestor de Dades.
- **crearParaulesFreqüènciaText(String nom, Set<Text> t):** Crea una Llista de Paraules Freqüències a partir d'una sèrie de Textos i la retorna per poder afegir-la al gestor de Dades.
- **modificarParaulesFreqüències(ParaulesFreqüència p, String llista):** Modifica la llista de paraules freqüència que es passa per paràmetre amb el contingut de l'string.

Posició

Resum: La classe Posició representa una posició de la matriu del teclat.

Atributs:

- **Integer i**
- **Integer j**

Mètodes:

- **Posició(Integer i, Integer j):** Crea una posició assignant els valors d'i i j passats per paràmetre.

Pair

Resum: Aquesta classe la utilitzem com a estructura de dades capaç d'emmagatzemar un parell de dades de qualsevol tipus.

Node

Resum: La classe Node representa una solució factible dins de l'arbre de solucions al fer el procés de branching a l'algoritme de branch and bound.

Atributs:

- **Integer instal·lació:** És la instal·lació que s'ha afegit a una solució parcial en una ubicació determinada.
- **Integer cota:** Representa la cota de la millor solució obtenible a partir de la solució parcial obtinguda al afegir una nova instal·lació.

CompNode

Resum: Com que a la hora de fer el branch and bound utilitzem una estratègia eager, els nodes no explorats els guardem a una cua amb prioritat. La classe CompNode implementa la interfície Comparator que ens permet definir l'ordre en el qual s'ordenen els nodes dins la cua.

Mètodes:

- **compare(Node n1, Node n2):** Donats dos nodes n1 i n2, retorna zero si la cota de n1 és igual a la de n2, un valor més petit que zero si la cota de n1 és més petita que la de n2 i un valor més gran que zero si la cota n1 és més gran que la de n2.

CreadorTeclat

Resum: Com que l'aplicació ha d'oferir dos mètodes per trobar una distribució d'un teclat a partir d'un alfabet i les freqüències de les paraules en aquest alfabet, hem decidit aplicar el patró de disseny strategy per tal de poder canviar l'algoritme amb el qual s'ha creat un teclat en temps d'execució. Per poder aplicar aquest patró hem creat la interfície CreadorTeclat, la qual és implementada per totes les classes que contenen la lògica dels diferents algorismes encarregats de generar les distribucions.

BranchAndBound

Resum: És la classe que implementa tots els mètodes necessaris per resoldre un Quadratic Assignment Problem a través del mètode branch and bound.

Mètodes:

- **crearDistribucio(alfabet : String, matDist:, matTrafic : int[][]):** Mètode encarregat d'iniciar el procés de branch and bound per crear una distribució del teclat òptima. Retorna un vector de caràcters que representa la millor solució trobada.

HungarianAlgorithm

Resum: La classe HungarianAlgorithm implementa un algoritmo que resol un problema d'assignació lineal donada una matriu de costos.

Mètodes:

- **trobarAssignacioOptima(matCost : Integer[][]):** Mètode que implementa l'Hungarian Algorithm per trobar l'assignació òptima en una matriu de costos. Retorna un vector on l'índex de cada element correspon a la fila de la matriu i el valor a la columna.

LectorFitxer

Resum: La classe LectorFitxer és una classe Singleton dissenyada per a la lectura de fitxers.

Mètodes:

- **importarFitxer(String path):** Aquest mètode rep una ruta de fitxer com a paràmetre i retorna un String que conté el contingut del fitxer. Utilitza la classe InputStream per llegir el contingut del fitxer i llança una excepció si el fitxer no es troba.

ParellsLletres

Resum: La classe ParellsLletres té la finalitat de gestionar informació sobre la freqüència de parells de lletres en un alfabet donat i les paraules associades amb aquesta freqüència. A continuació, es descriuen els seus atributs i mètodes públics:

Atributs:

- **int[][] trafic:** Una matriu d'enters que representa la freqüència de parells de lletres. Les files i les columnes d'aquesta matriu estan associades amb els caràcters de l'alfabet, i els valors de la matriu indiquen la freqüència de transicions de la lletra associada a la fila cap a la lletra associada a la columna.
- **ParaulesFrecuencia pf:** Una instància de la classe ParaulesFrecuencia que conté informació sobre la freqüència de paraules.
- **Alfabet a:** Una instància de la classe Alfabet que representa l'alfabet associat amb els parells de lletres.

Mètodes:

- **ParellsLletres(Alfabet a, ParaulesFrecuencia pf):** Constructor de la classe que rep una instància de l'alfabet i una instància de ParaulesFrecuencia com a paràmetres i inicialitza els atributs de la classe. Llança una excepció si es produeix un error en el càlcul del tràfic

CtrlParellsLletres

Aquesta classe és l'encarregada de crear les llistes de parells de lletres necessàries per crear els Teclats.

Mètodes:

- **crearParellLletres (Alfabet a, ParaulesFrecuencia pf):** Crea una llista de parells de lletres a partir d'un alfabet i d'una llista de paraules-freqüències.

3. Estructures de dades i algorismes utilitzats

a. ParellsLletres

- i. Matriu d'Arrays: S'utilitza per guardar el flux d'una llista de Freqüències per un Alfabet donat. Els índexs son dos caràcters i conté el número d'aparicions d'aquest parell de lletres.

b. Alfabet

- i. Set de Character: S'utilitza per guardar els alfabetes. Utilitzem un Set ja que un alfabet no pot tenir caràcters repetits.

c. GestorDades

- i. Map: Utilitzat per guardar les instàncies de cada Classe: Alfabet, Teclat, Text, ParaulesFreqüències i ParellsLletres. Com a clau primària té el Nom de cada objecte de les classes que guarda (que els identifiquen).

d. Teclat

- i. Array: Utilitzat per guardar els caràcters del teclat. A l'hora de mostrar el teclat s'ha de tenir en compte l'atribut amplada per transformar-ho a matriu.

e. BranchAndBound

- i. Matriu d'enters: Una matriu d'enters per guardar el flux/tràfic entre parells de caràcters, on cada número de fila o columna correspon a la posició del caràcter en l'alfabet proporcionat. També s'utilitza una matriu d'enters per guardar la distància entre posicions.
- ii. Llista d'enters: S'utilitzen llistes d'enters per mantenir un registre dels caràcters que s'han emplaçat i els que falten per emplaçar en tot moment.
- iii. Algorisme Branch and Bound: Aquest algorisme és utilitzat per calcular la millor distribució possible minimitzant la suma de les distàncies entre ubicacions multiplicades pel tràfic que hi ha de passar. Per calcular les cotes s'ha utilitzat el mètode de Gilmore-Lawler.

f. HungarianAlgorithm

- i. Matriu d'enters: Hem utilitzat una matriu d'enters que representen els costos d'assignar una instal·lació determinada en una ubicació. Les files corresponen a les instal·lacions i les columnes a les ubicacions.
- ii. Hungarian Algorithm: Aquest algoritme ens permet resoldre un problema d'assignació lineal, el qual consisteix en assignar cada instal·lació a una ubicació de tal manera que el cost total sigui el menor possible. Aquesta tasca forma part del càlcul de la cota de Gilmore-Lawler.