Gestor de teclats

Lliurament 2

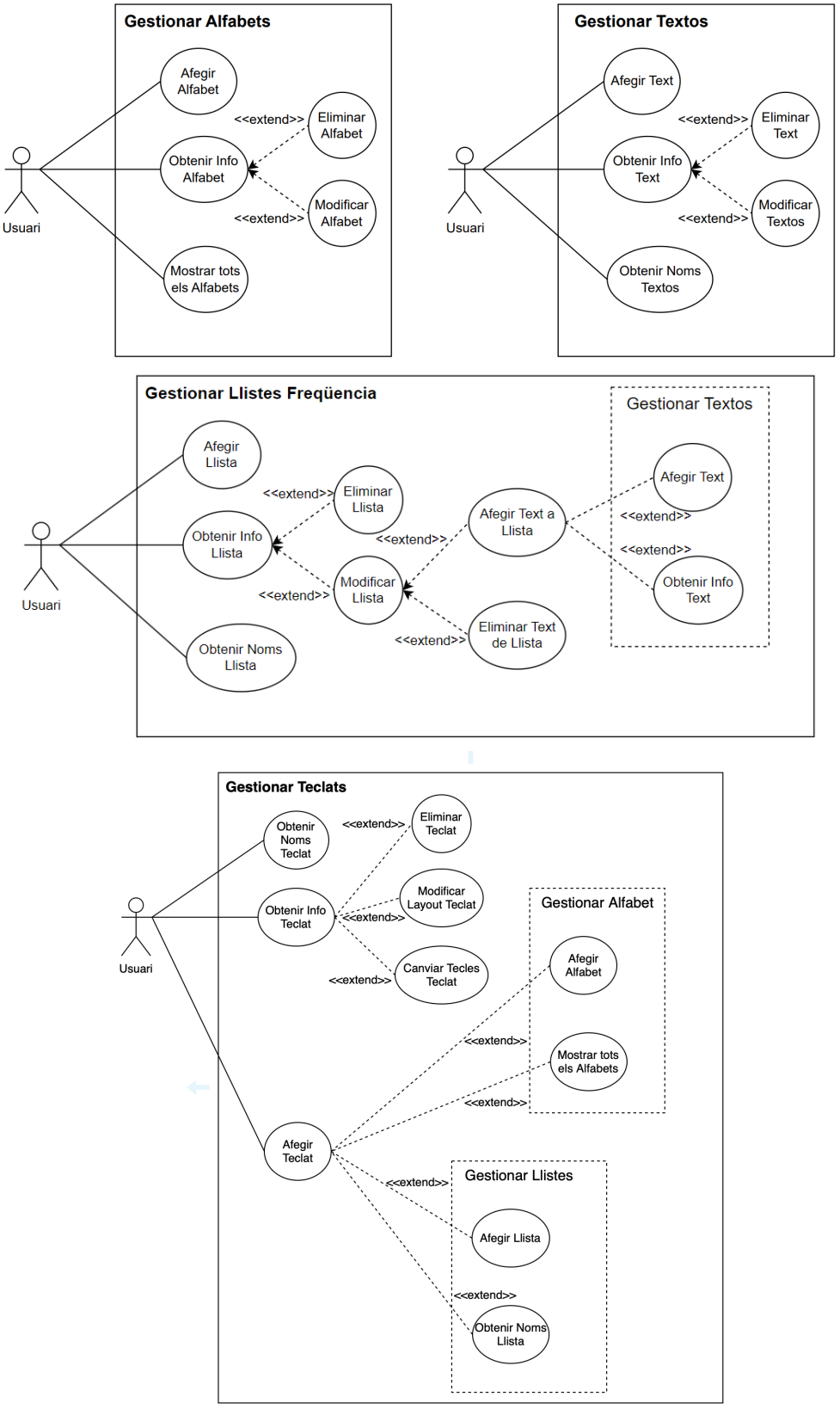
Grup 31.2

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

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 línea.
    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 línea.
    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 FREQÜÈNCIA:

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.

Errors possibles 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 esta 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'us 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.

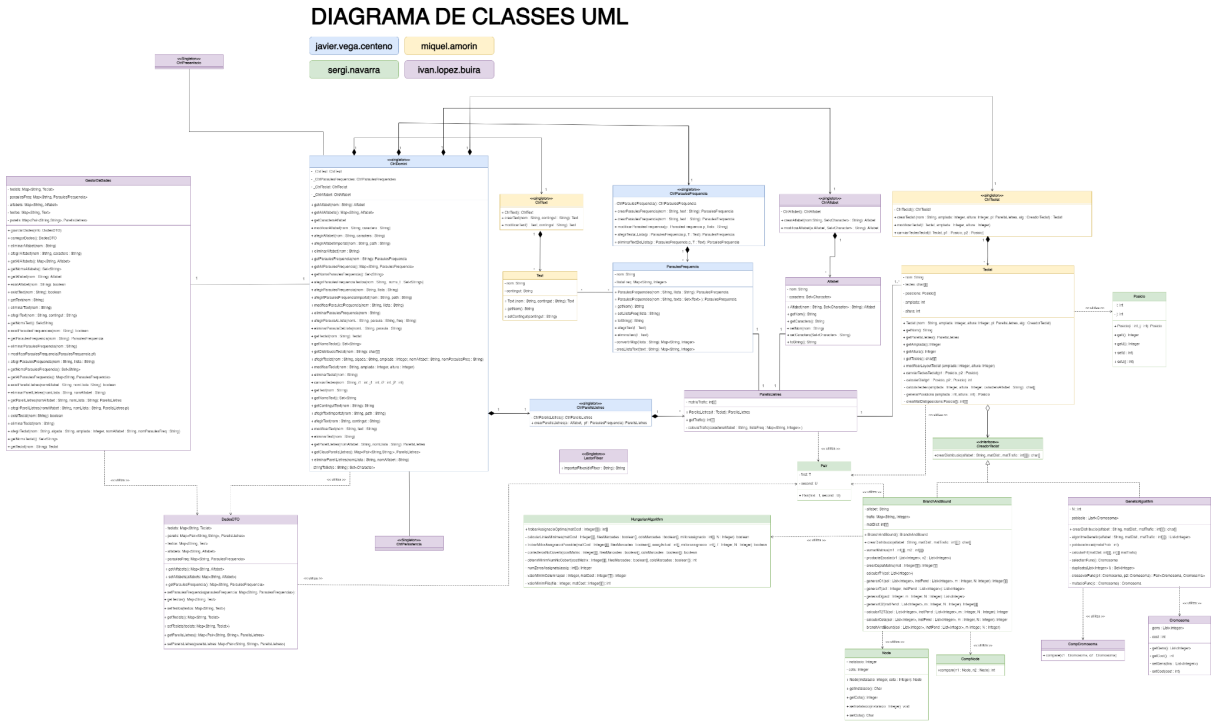
Nom: Eliminar Text de Llista

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.

2. Diagrama del model conceptual de Domini



a. Diagrama

Claus externes: (Alfabet, nom), (ParaulesFrequencia, nom), (Teclat, nom), (ParellsLletres, Alfabet::nom + ParaulesFrequencia::nom), (Text, nom)

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 quals 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'usarà 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'utlitzarà. 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 clase 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 clase també porta registre dels Textos amb els que s'ha creat amb la capacitat de poder afegir o eliminar algún dels textos.

Atributs:

- **Map<String, Integer> llistaFreq:** Mapa on la clau és la paraula i el valor es la freqüència.
- **String nom:** Nom que identifica a la clase.
- **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 clase a partir de strings en format paraula:frequencia;
- **ParaulesFrequencia(String nom, Set<Text> t):** creadora de Paraules Freqüència 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 freqüència 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 emmagatzemada i amb els diferents controladors (CtrlTeclat, CtrlText, CtrlAlfabet, CtrlLlistaFreqüencia i CtrlLlistaParells) per implementar les funcionalitats del software. Conté una instància de cada controlador , del lector de fitxer i del gestor de dades.

Mètodes:

- **modificarAlfabet(String nom, String caràcters):** 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, Posició p1, Posició 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 alfabets: crear, modificar. Conté una instancia 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 l'encarregada de crear i modificar les instàncies de Paraules Frequencies.
Mètodes:

- **crearParaulesFrequencia(String nom, String text)**: Crea una Llista de Paraules Freqüències a partir de una string amb el format paraula:freqüència; i la retorna per poder afegir-la al gestor de Dades.
- **crearParaulesFrequenciaText(String nom, Set<Text> t):** Crea una Llista de Paraules Freqüències a partir de una serie de Textos i la retorna per poder afegir-la al gestor de Dades.
- **modificarParaulesFrequencies(ParaulesFrequencia 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'apliciació 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 crear la interfície CreadorTeclat, la qual és implementada per totes les classes que contenen la lògica dels diferents algoritmes 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.
- **ParaulesFrequencia pf:** Una instància de la classe ParaulesFrequencia 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, ParaulesFrequencia pf):** Constructor de la classe que rep una instància de l'alfabet i una instància de ParaulesFrequencia 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, ParaulesFrequencia pf):** Crea una llista de parells de lletres a partir d'un alfabet i d'una llista de paraules-frequencies.

GestorDades

Aquesta clase és l'encarregada de gestionar les dades del sistema. Actua com a caché de manera que només es carrega i guarda les dades de la capa de persistència al iniciar la aplicació i tancar-la respectivament. El gestor només es comunica amb el Controlador de Domini de manera que no hi haurà acoblament.

Atributs:

- **Map<String, Teclat> teclats:** Un HashMap amb clau de tipus String i valor de tipus Teclat que guarda tots els teclats del sistema.
- **Map<String, ParaulesFrequencia> paraulesFreq:** Un HashMap amb clau de tipus String i valor de tipus ParaulesFrequencia que guarda totes les llistes de Paraules amb les seves frequencies del sistema.
- **Map<String, Alfabet> alfabets:** Un HashMap amb clau de tipus String i valor de tipus Alfabet que guarda tots els alfabets del sistema.
- **Map<Pair<String,String>, ParellsLletres> parells:** Un HashMap amb clau de tipus Pair de (String - String) i valor de tipus ParellsLletres que guarda totes les llistes de ParellsLletres del sistema.
- **Map<String, Text> textos:** Un HashMap amb clau de tipus String i valor de tipus Text que guarda tots els textos del sistema.

Mètodes:

- **guardarDades (info : DadesDTO):** Guarda cada atribut del DTO a la propia clase.
- **carregarDades(): DadesDTO:** Crea un objecte DadesDTO on guardarà totes les dades que es trobin a la propia classe i retornarà el DTO.

> Després per cada atribut tenim:

- **exist(clau : String):** Retorna cert si existeix en el Map la clau passada per paràmetre
- **getAll()**: Retorna tots els elements del Map.
- **getNoms():** Retorna les claus de tots els elements del Map.
- **afegir(Objecte: O):** Afegeix al Map una entrada amb l'identificador de l'objecte com a clau i l'objecte O com a valor.
- **eliminar(clau : String):** Elimina del Map l'entrada amb la clau passada per paràmetre.

DadesDTO

Aquesta classe es tracta d'un DTO (Data Transfer Unit) en la cual guardarem totes les dades que vulguem passar a una altra capa. Utilitzem el DTO per poder serialitzar l'objecte i poguer transferir dades més còmodament.

Atributs:

- **Map<String, Teclat> teclats:** Un HashMap amb clau de tipus String i valor de tipus Teclat que guarda els teclats a transferir.
- **Map<String, ParaulesFrequencia> paraulesFreq:** Un HashMap amb clau de tipus String i valor de tipus ParaulesFrequencia que guarda les ParaulesFreqüència a transferir.
- **Map<String, Alfabet> alfabets:** Un HashMap amb clau de tipus String i valor de tipus Alfabet que guarda els alfabets a transferir.
- **Map<Pair<String,String>, ParellsLletres> parells:** Un HashMap amb clau de tipus Pair de (String - String) i valor de tipus ParellsLletres que guarda els ParellsLletres a transferir.
- **Map<String, Text> textos:** Un HashMap amb clau de tipus String i valor de tipus Text que guarda els textos a transferir.

GeneticAlgorithm

Aquesta classe implementa els mètodes necessaris per poder resoldre el Quadratic Assignment Problem mitjançant l'algoritme genètic

Atributs:

- **int N:** Representa la mida de la població de cada generació.
- **List<Cromosoma> població:** Representa la població actual, cada individu de la població es un cromosoma.

Mètodes:

- **crearDistribucio(alfabet: String, matDist: int[][], matTrafic: int[][]): char[]:** Mètode encarregat d'iniciar l'algoritme genètic per crear una distribució del teclat òptima. Retorna un vector de caràcters que representa la millor solució trobada. Implementa la interfície CreadorTeclat.

Cromosoma:
Aquesta classe representa una solució possible dins de l'algoritme genètic. Una població de l'algoritme està formada per N cromosomes

Atributs:

- **List<Integer> gens:** Cada cromosoma està format per una llista Integers on el valor d'un element és la instal·lació i la posició en la llista és alguna possible posició.
- **int cost:** Representa el cost total del cromosoma, és a dir el sumatori de cada instal·lació per el seu cost de tràfic.

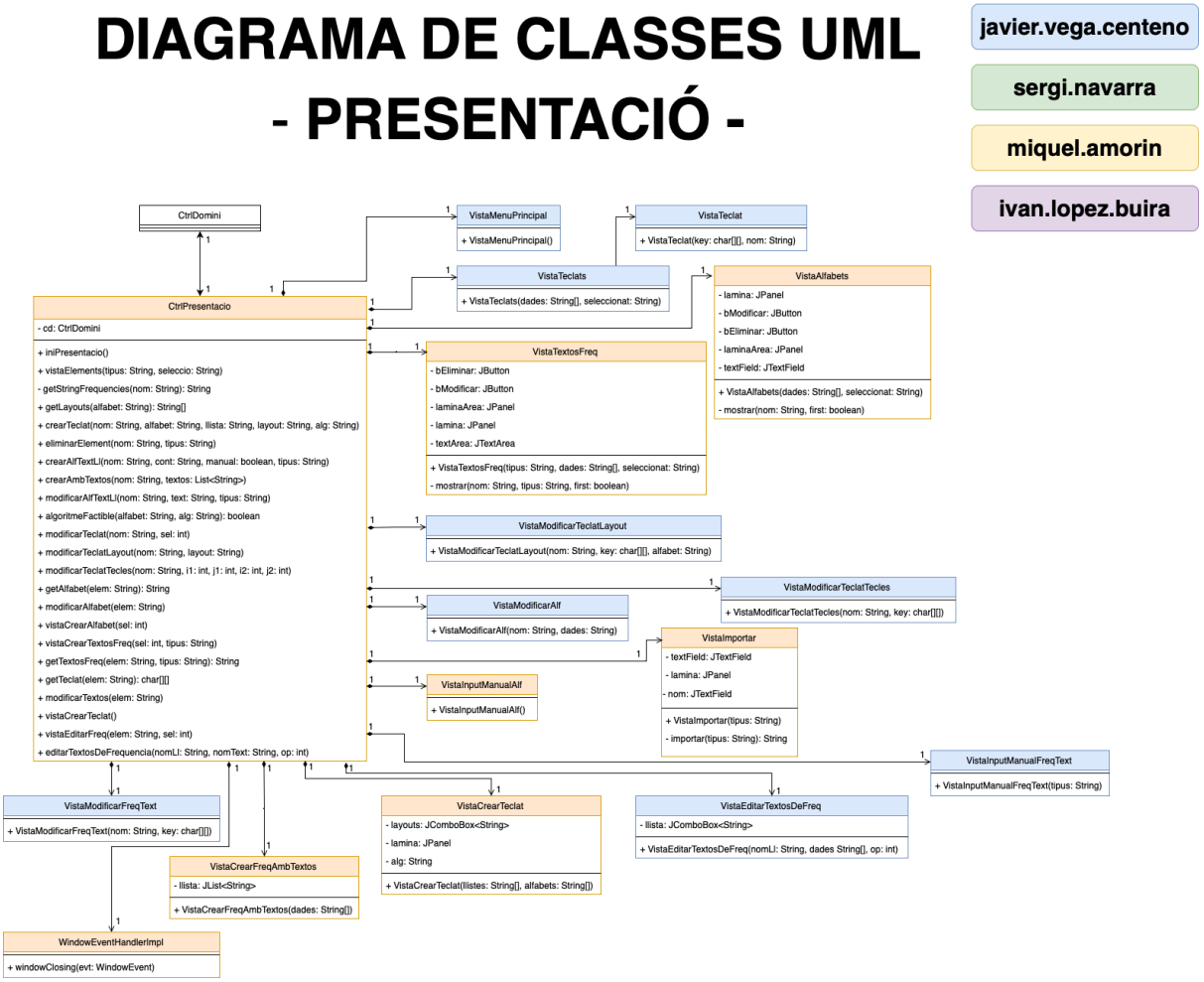
CompCromosoma:
Com que a la hora d'avaluar el millor cromosoma de la població hem de trobar el cost inferior, hem de poder ordenar la població en funció del cost. La classe CompCromosoma implementa la interfície Comparator que ens permet definir l'ordre en el qual s'ordenen els cromosomes dins de la llista.

Mètodes:

- **compare(n1: Cromosoma, n2 : Cromosoma):** Donats dos cromosomes n1 i n2, retorna zero si el cost de n1 és igual al d'n2, un valor més petit que zero si el cost de n1 és més petita que el d'n2 i un valor més gran que zero si el cost d'n1 és més gran que el d'n2.

3. Diagrama del model conceptual de Presentació

Diagrama



Descripció de les classes

VistaAlfabets

Resum: Aquesta vista és l'encarregada de gestionar els alfabets. Es compon d'un desplegable que conté tots els alfabets registrats en el sistema, un botó de creació que mostra un pop-up amb dues opcions (Manualment, Importar), la primera porta a VistaInputManualAlf i la segona a VistaImportar. Quan seleccionem un alfabet del desplegable, es mostra una vista prèvia de l'alfabet i s'activen els botons d'eliminar i de modificar, el botó modificar porta a VistaModificarAlf. També té un botó de retrocés.

Atributs:

- **JTextField textField:** Camp de text per a mostrar el contingut de l'alfabet.
- **JPanel laminaArea:** Panell que conté tots els elements de la vista
- **JButton bEliminar:** Botó per eliminar un alfabet seleccionat.
- **JButton bModificar:** Botó per a modificar un Alfabet, et porta a VistaModificarAlf.
- **JPanel lamina:**

Mètodes:

- **VistaAlfabets(dades : String[], seleccionat : String):** Constructor de la classe. Configura la interfície gràfica per a la visualització dels alfabets. Si el paràmetre seleccionat és not null el desplegable té marcat per defecte aquesta opció.

VistaCrearFreqAmbTextos

Resum: VistaCrearFreqAmbTextos permet crear una nova llista de freqüències a partir d'un o més textos guardats en l'aplicació. Conté un espai de text per a poder introduir el nom de la llista de freqüències, un espai on es mostren els diferents textos disponibles i que permet seleccionar un o diversos, un botó Crear per a crear la nova llista a partir dels textos seleccionats i que seguidament t'envia a VistaTextosFreq i un botó de retrocés.

Atributs:

- **JList<String> llista:** Llista visual que permet la selecció dels textos.

Mètodes:

- **VistaCrearFreqAmbTextos(dades: String[]):** Constructora de la classe que permet crear llistes de freqüències a partir de textos.

VistaCrearTeclat

Resum: Aquesta vista té diversos components que ens permeten crear un teclat. Hi ha un espai de text on posar el nom del teclat, dos desplegables per a poder seleccionar un alfabet i una llista de freqüències (al costat de cada desplegable hi ha un botó de crear per si volem afegir un alfabet/llesta de freqüències al moment), una vegada hem seleccionat un alfabet es desbloqueja un altre desplegable que permet seleccionar un layout. També tenim dos algorismes per a seleccionar (Branch And Bound i Genetic) si seleccionem l'algorisme B&B, però l'alfabet és massa gran sortirà un pop-up avisant-nos que el teclat trigarà molt temps a generar-se i ens preguntarà si volem canviar l'algorisme pel Genetic. Finalment, hi ha el botó de crear i el de retrocés.

Atributs:

- **JPanel lamina:** Panell que conté tots els elements de la vista
- **JComboBox<String> layouts:** Llista desplegable per a seleccionar layouts.
- **String alg:** Variable per a emmagatzemar l'algorisme seleccionat.

Mètodes:

- **VistaCrearTeclat(llistes: String[], alfabets: String[]):** Constructora de la classe. Configura la interfície gràfica per a la creació d'un teclat.

VistaEditarTextosDeFreq

Resum: Aquesta vista s'utilitza per a afegir/eliminar un text que forma part d'una llista de freqüències. Té un desplegable amb els possibles textos a afegir/eliminar, un botó Afegir/Eliminar que modifica la llista de freqüències segons l'opció en la qual estiguem i un botó de retrocés.

Atribut:

- **JComboBox<String> llista:** Llista desplegable per a seleccionar els textos.

Mètodes:

- **VistaEditarTextosDeFreq(nomLl: String, dades String[], op: int):** Constructor de la classe. Configura la interfície gràfica per a editar textos d'una llista de freqüències. El paràmetre op indica si la vista s'usa per a eliminar (op == 0) o per a afegir (op == 1) textos.

VistaImportar

Resum: VistaImportar permet importar un element des del nostre dispositiu. Es compon d'un espai per a posar nom a l'element i un botó Seleccionar Fitxer que obre el navegador d'arxius del sistema. Finalment, disposa d'un espai per a visualitzar el path del fitxer una vegada importat, un botó Crear que crea el nou element i un botó de retrocés.

Atribut:

- **JPanel lamina:** Panell que conté tots els elements de la vista
- **JTextField textField:** Camp de text per a mostrar la ruta de l'arxiu seleccionat.
- **JTextField nom:** Camp de text on es mostra el nom del fitxer importat (l'usuari el pot modificar).

Mètodes:

- **VistaImportar(tipus: String):** Constructora de la classe. Configura la interfície gràfica per a la importació de dades.

VistaInputManualAlf

Resum: Aquesta vista permet crear un nou alfabet de manera manual. Es compon d'un espai per a posar nom a l'alfabet i un altre per a introduir el nou alfabet. També conté un botó de retrocés i un botó Crear que crea el nou alfabet en el sistema.

Mètodes:

- **VistaInputManualAlf():** Constructor de la classe. Configura la interfície gràfica per a la creació manual d'alfabets.

VistaInputManualFreqText

Resum: Aquesta vista s'utilitza tant per a crear textos com llistes de freqüències de manera manual segons el que li indiquem des del CtrlPresentació. Conté un espai de text per a introduir el nom del nou element, un altre espai on introduir el contingut de l'element, botó Crear per a crear el nou element i un botó per a retrocedir.

Mètodes:

- **VistaInputManualFreqText(tipus: String):** Constructora de la classe. Configura la interfície gràfica per a la creació manual d'elements del tipus especificat (Textos o Freqüències).

VistaMenuPrincipal

Resum: Aquesta vista s'encarrega de mostrar la primera pantalla que apareix en executar el programa. És un menú amb diverses opcions en forma de botons: Teclats, Textos, Llistes de frequencia i Alfabets, cadascun d'ells ens porta a diferents vistes on gestionar els diferents elements del sistema. També tenim un botó Sortir que tanca el programa.

Mètodes:

- **VistaMenuPrincipal():** Constructora de la finestra del menú principal encarregada de mostrar els 5 botons que permeten saltar de vistes i sortir del programa.

VistaModificarAlf
Resum: VistaModificarAlf conté un espai on es mostra l'alfabet i on podem modificar-lo, un botó Modificar per a acceptar els canvis i un botó de retrocés.
Mètodes:

- **VistaModificarAlf(nom: String, dades: String):** Constructora de la classe. Configura la interfície gràfica per a la modificació d'un alfabet.

VistaModificarFreqText
Resum: Aquesta vista s'utilitza tant per modificar textos com llistes de freqüències (de manera manual) segons el que li indiquem desdel CtrlPresentació. Té una zona amb la informació de l'element que podem canviar per a modificar-lo, un botó Modificar per a confirmar els canvis i un botó de retrocés.
Mètodes:

- **VistaModificarFreqText(nom: String, key: char[][]):** Constructora de la classe. Configura la interfície gràfica per a la modificació de textos o llistes de freqüència.

VistaModificarTeclatLayout
Resum: Aquesta vista ens permet canviar el layout del teclat. En la vista podem veure un teclat (amb la forma del layout triat) fet amb botons. Hi ha un desplegable que mostra els layouts possibles per a seleccionar i un botó Modificar que efectua el canvi en el sistema. També hi ha un botó de retrocés.
Mètodes:

- **VistaModificarTeclatLayout(nom: String, key: char[][], alfabet: String):** Constructora de la classe. Configura la interfície gràfica per a la modificació del layout d'un teclat.

VistaModificarTeclatTecles
Resum: Aquesta vista ens permet intercanviar les posicions de dues tecles del teclat. En la vista podem veure un teclat (amb la forma del layout triat) fet amb botons. Quatre desplegables ens permeten seleccionar les files i les columnes de les posicions a intercanviar, estan programats per a mostrar només números de fila/columnes vàlids i així evitar errors. Finalment, tenim el botó Modificar que confirma els canvis i un botó de retrocés.
Mètodes:

- **VistaModificarTeclatTecles(nom: String, key: char[][]):** Constructora de la classe. Configura la interfície gràfica per a la modificació de les tecles d'un teclat.

VistaTeclat
Resum: VistaTeclat ens mostra un teclat (amb la forma del layout triat) fet amb botons. També té tres botons: Modificar, Eliminar i retrocés. El botó Modificar mostra un pop-up amb dues opcions: Layout, que porta a VistaModificarTeclatLayout i Tecles, que porta a VistaModificarTeclatTecles.
Mètodes:

- **VistaTeclat(key: char[][], nom: String):** Constructora de la classe. Configura la interfície gràfica per a mostrar un teclat, amb opcions per a modificar, eliminar o tornar.

VistaTeclats

Resum: Aquesta vista només té tres elements: un desplegable que mostra els diferents teclats creats, un botó per a crear teclats i un botó de retrocés. Una vegada seleccionat un teclat del desplegable, s'obre VistaTeclat, el botó de crear porta a VistaCrearTeclat.

Mètodes:

- **VistaTeclats(dades: String[], seleccionat: String):** Constructora de la classe. Configura la interfície gràfica per a mostrar una llista de teclats.

VistaTextosFreq

Resum: Aquesta vista s'utilitza tant per a mostrar textos com llistes de freqüències segons el que li indiquem des del CtrlPresentació. Disposa d'un desplegable per a seleccionar els elements i d'un quadre de text on previsualitzar la informació i un botó per a crear un nou element. Una vegada seleccionat l'element es pot accedir als botons Eliminar i Modificar. Finalment hi ha un botó de retrocés.

Opció de mostrar text:

El botó per a crear un nou element mostra dues opcions (Manualment, Importar), la primera porta a VistaInputManualFreqText i la segona a VistaImportar. El botó Modificar et porta a VistaModificarFreqText

Opció de mostrar llistes de freqüències:

El botó de crear un nou element mostra tres opcions (Manualment, Importar, Textos) la primera porta a VistaInputManualFreqText, la segona a VistaImportar i la tercera a VistaCrearFreqAmbTextos. El botó Modificar et mostra un pop-up amb tres opcions (Eliminar Text, Afegir Text, Editar Llista) les dues primeres opcions porten a VistaEditarTextosDeFreq mentre que la tercera porta a VistaModificarFreqText.

Atributs:

- **JTextArea textArea:** Camp de text utilitzat per a mostrar el contingut d'un text o llista de freqüència.
- **JPanel lamina:** Panell que conté tots els elements de la vista
- **JPanel laminaArea:** Panell que conté el textField i la barra d'scroll per mostrar el text o la llista de freqüències
- **JButton bModificar:** Botó per modificar un element seleccionat.
- **JButton bEliminar:** Botó per eliminar un element seleccionat.

Mètodes:

- **VistaTextosFreq(tipus: String, dades: String[], seleccionat: String):** Constructora de la classe. Configura la interfície gràfica per a mostrar una llista de textos o llistes de freqüència. L'atribut tipus determina si la vista s'usa sobre Textos o sobre Llistes de Freqüències.

WindowEventHandlerImpl

Resum: S'encarrega de gestionar el guardar les dades en el moment de tancar la persistència. És una extensió de la classe WindowAdapter

Mètodes:

- **windowClosing():** Crida a la funció de Control Presentació de guardar dades i tanca el programa.

CtrlPresentacio

Resum: La classe CtrlPersistencia s'encarrega de gestionar la comunicació entre les vistes. Conté una instància de la classe CtrlDomini per a poder relacionar-se amb la capa de Domini. El CtrlPersistencia demana al CtrlDomini les dades que necessiten les diferents vistes i li envia la informació dels canvis (Crear, Modificar i Eliminar) produïts.

Atributs:

- **CtrlDomini cd:** Instància del controlador de domini.

Mètodes:

- **iniPresentacio():** Mostra en pantalla la finestra del menú principal (VistaMenuPrincipal).
- **vistaElements(tipus: String, seleccio: String):** Aquesta funció té diferent comportament segons el paràmetre *tipus*, però la seva funció principal és mostrar elements per pantalla. Segons el paràmetre tipus es demanarà la informació necessària al CtrlDomini i després es mostrarà per pantalla les finestres que permeten veure els elements del sistema ja siguin teclats (VistaTeclats), alfabets (VistaAlfabets), textos (VistaTextosFreq) o llistes de freqüències (VistaTextosFreq).
- **getStringFrequencies(nom: String): S tring:** Demana al CtrlDomini que la llista de freqüències de paraules corresponent, la formata correctament per a poder mostrar-los per pantalla i els envia a la vista/funció que els ha demanat.
- **getLayouts(alfabet: String): String[]:** Demana al CtrlDomini que li passi els layouts possibles per a l'alfabet determinat, els formata correctament per a poder mostrar-los per pantalla i els envia a la vista/funció que els ha demanat.
- **crearTeclat(nom: String, alfabet: String, llista: String, layout: String, alg: String):** Envia al CtrlDomini les dades necessàries per a crear un teclat (nom del teclat, de l'alfabet i de la llista de freqüències, el layout i l'algorisme a utilitzar), després crida a vistaElements per a anar a VistaTeclats.
- **eliminarElement(nom: String, tipus: String):** Aquesta funció demana al CtrlDomini que elimini l'objecte del tipus indicat pel paràmetre *tipus* i el nom corresponent, després crida a la funció vistaElement.
- **crearAlfTextLl(nom: String, cont: String, manual: boolean, tipus: String):** Aquesta funció serveix per a crear un alfabet, un text o una llista de freqüències segons indiqui l'atribut tipus. La manera en què es crearà l'element és definida per l'atribut manual (manual == true → de manera manual; manual == false → de forma importada). Li mana al CtrlDomini el nom i el contingut de l'element i després crida a la funció vistaElement.
- **crearAmbTextos(nom: String, textos: List<String>):** Li mana al CtrlDomini un set de textos perquè creï una llista de paraules de freqüències amb aquests textos. Crida a la funció vistaElements.
- **modificarAlfTextLl(nom: String, text: String, tipus: String):** La funció mana al CtrlDomini la informació necessària per a modificar un Alfabet, Text o Llista de Freqüències segons indiqui l'atribut tipus, després crida a vistaElements.
- **algoritmeFactible(alfabet: String, alg: String): boolean:** Calcula que l'algorisme triat per a crear el teclat es pot utilitzar sense causar una espera massa gran. En cas afirmatiu return és true i al contrari return és false.
- **modificarTeclat(nom: String, sel: int):** Depenent del paràmetre *sel* modificarem el teclat canviant el layout o intercanviant dues tecles. En el primer pas demana al CtrlDomini l'alfabet del qual està creat el teclat i mostra per pantalla la finestra que permet modificar el layout d'un teclat (VistaModificarTeclatLayout). Si *sel* indica que

volem intercanviar dues tecles, es mostra per pantalla la finestra que permet intercanviar les seves posicions (VistaModificarTeclatTecles).

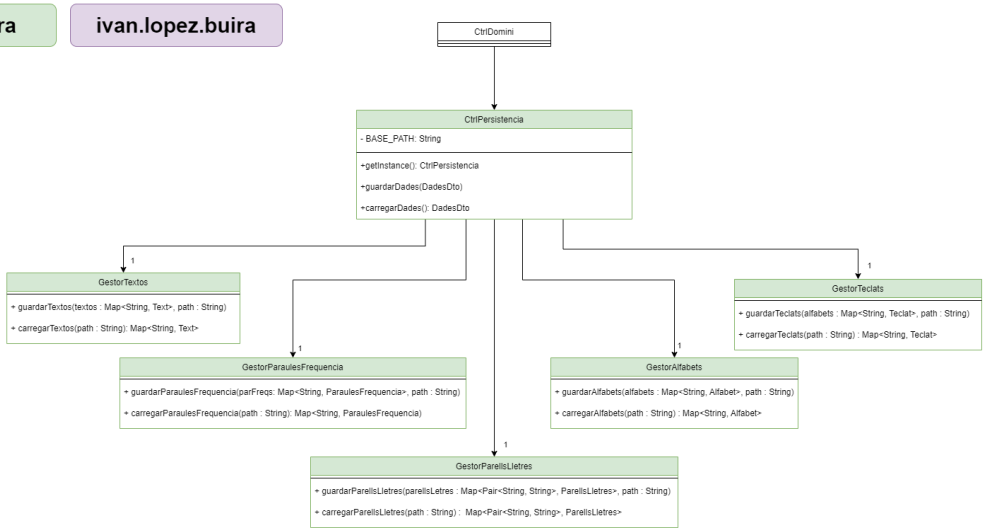
- **modificarTeclatLayout(nom: String, layout: String):** Envia al CtrlDomini les dades necessàries per a modificar el layout d'un teclat (nom del teclat i nou layout), després crida a vistaElements per a anar a VistaTeclats.
- **modificarTeclatTecles(nom: String, i1: int, j1: int, i2: int, j2: int):** Envia al CtrlDomini les dades necessàries per a intercanviar dues tecles d'un teclat d'un teclat (nom del teclat i les posicions de les dues tecles a intercanviar), després crida a vistaElements per a anar a VistaTeclats.
- **getAlfabet(elem: String): String:** Demana al CtrlDomini els caràcters que formen l'alfabet amb nom *elem* i l'envia a la vista/funció que l'ha demanat.
- **modificarAlfabet(elem: String):** Demana al CtrlDomini els caràcters que formen l'alfabet amb nom *elem*, després mostra en pantalla la finestra que permet modificar un alfabet (VistaModificarAlf).
- **vistaCrearAlfabet(sel: int):** Mostra en pantalla la finestra per a crear un alfabet de manera manual (VistaInputManualAlf) o per a importar un element (VistaImportar) segons el que li indiqui el paràmetre sel. En cas de no fer cap de les opcions anteriors crida a la funció vistaElements.
- **vistaCrearTextosFreq(sel: int, tipus: String):** En aquesta funció podrem crear un text o una llista de freqüències depenent del paràmetre *sel*. Si *sel* == 0 es mostra en pantalla la finestra que permet crear un element (text/llesta) de manera manual (VistaInputManualFreqText), si *sel* == 1 es mostra en pantalla la finestra que permet importar un element i si *sel* == 2 se li demana al CtrlDomini el nom de tots els textos emmagatzemats en el sistema i es diu a la funció VistaCrearFreqAmbTextos. Finalment, si sel presa qualsevol altre valor es diu a la funció vistaElements.
- **getTextosFreq(elem: String, tipus: String): String:** Li demana al CtrlDomini el contingut del text o de la llista de paraules de freqüència identificada pel paràmetre elem i el retorna a la vista/funció que l'hagi demanat.
- **getTeclat(elem: String): char[][]:** Demana al CtrlDomini el teclat amb nom *elem* i ho envia a la vista/funció que ho ha demanat.
- **modificarTextos(elem: String):** Demana al CtrlDomini el contingut del text amb nom elem i mostra en pantalla la finestra que permet modificar els textos (VistaModificarFreqText).
- **vistaCrearTeclat():** Demana al CtrlDomini els noms de totes les llistes de freqüència i de tots els alfabets i mostra la finestra que permet crear un teclat (VistaCrearTeclat).
- **vistaEditarFreq(elem: String, sel: int):** Amb aquesta funció podem modificar una llista de freqüències de tres maneres diferents: Eliminant un text dels quals formen les llista, en aquest cas demanem al CtrlDomini els textos que formen la llista i vam mostrar per pantalla la finestra que permet eliminar un text d'una llista (VistaEditarTextosDeFreq); Afegint un text a una llista, demanem al CtrlDomini els textos que no estan afegits a la llista i vam mostrar per pantalla la finestra que permet afegir un text a una llista (VistaEditarTextosDeFreq); Finalment modificant manualment la llista mostrant per pantalla la finestra que permet modificar manualment una llista (VistaModificarFreqText).
- **editarTextosDeFrequencia(nomLl: String, nomText: String, op: int):** Li demana al CtrlDomini que afegeixi/elimini el text nomText de la llista de freqüències nomLl, després crida a la funció vistaElement.

4. Diagrama del model conceptual de Persistència

a. Diagrama

DIAGRAMA DE CLASSES UML - PERSISTÈNCIA

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



b. Descripció de classes

GestorTextos
Resum: Aquesta classe s'encarrega de guardar i carregar la informació relacionada amb els textos a un fitxer de text pla.
Mètodes:

- **guardarTextos(textos : Map<String, Text>, path : String)**: Guarda el fitxer ubicat a la direcció indicada pel paràmetre *path* la informació dels textos que se li passen també com a paràmetre.
- **carregarTextos(path: String): Map<String, Text>**: Retorna la informació dels textos guardada en el fitxer ubicat a la direcció indicada pel paràmetre *path*.

GestorParaulesFrequencia
Resum: Aquesta classe s'encarrega de guardar i carregar un conjunt de llistes de paraules amb les seves frequencies a un fitxer de text pla.
Mètodes:

- **guardarParaulesFrequencia(parFreqs : Map<String, ParaulesFrequencia>, path : String)**: Guarda el fitxer ubicat a la direcció indicada pel paràmetre *path* totes les llistes de paraules i freqüències que conté la variable *parFreqs.*
- **carregarParauesFrequencia(path: String): Map<String, ParaulesFrequencia>**: Retorna el conjunt de llistes de paraules i freqüències contingudes al fitxer indicat pel paràmetre *path*.

GestorParellsLletres
Resum: S'encarrega de guardar i carregar a un fitxer de text pla un conjunt d'instàncies de la classe ParellsLletres.
Mètodes:

- **guardarParellsLletres(parellsLletres: Map<Pair<String, String>, ParellsLletres>, path: String)**: Guarda al fitxer ubicat a la direcció indicada pel paràmetre *path* totes les instàncies de la classe ParellsLletres indicades pel paràmetre *parellsLletres*.
- **carregarParellsLletres(path: String): Map<Pair<String, String>, ParellsLletres>:** Retorna el conjunt d'instàncies de la classe ParellsLletres guardades en el fitxer ubicat a la direcció indicada pel paràmetre *path*.

GestorAlfabets
Resum: S'encarrega de guardar i carregar a un fitxer de text pla un conjunt d'instàncies de la classe Alfabet.
Mètodes:

- **guardarAlfabets(alfabets : Map<String, Alfabet>, path : String)**: Guarda al fitxer ubicat a la direcció indicada pel paràmetre path totes les instàncies de la classe Alfabet indicades pel paràmetre *alfabets*.
- **carregarAlfabets(path: String): Map<String, Alfabet>:** Retorna el conjunt d'instàncies de la classe Alfabet guardades en el fitxer ubicat a la direcció indicada pel paràmetre *path*.

GestorTeclats
Resum: Aquesta classe s'encarrega de guardar i carregar la informació relacionada amb els teclats a un fitxer de text pla.
Mètodes:

- **guardarTeclats(teclats : Map<String, Teclat>, path : String):** Guarda al fitxer ubicat a la direcció indicada pel paràmetre path la informació dels teclats que se li passen amb la variable teclats.
- **carregarTeclats(path: String): Map<String, Teclat>:** Retorna la informació dels teclats guardada en el fitxer ubicat a la direcció indicada pel paràmetre *path*.

CtrlPersistencia

Resum: És la classe encarregada de gestionar la comunicació entre les capes de domini i persistència.

Atributs:

- **BASE_PATH: String**: Conté la direcció de la carpeta on es guardarán tots els fitxers necessaris per assolir la persistència de les dades.
- **gestorAlfabets: GestorAlfabets**: Conté una instància de la classe GestorAlfabets que s'utilitzarà per guardar i carregar la informació dels alfabets.
- **gestorTextos: GestorTextos**: Conté una instància de la classe GestorTextos que s'utilitzarà per guardar i carregar la informació dels textos.
- **gestorParellsLletres: GestorParellsLletres**: Conté una instància de la classe GestorParellsLletres que s'utilitzarà per guardar i carregar la informació de les llistes de parells de lletres.
- **gestorParaulesFrequencia: GestorParaulesFrequencia**: Conté una instància de la classe GestorParaulesFrequencia que s'utilitzarà per guardar i carregar la informació de les llistes de paraules i freqüències.
- **gestorTeclats: GestorTeclats**: Conté una instància de la classe GestorTeclats que s'utilitzarà per guardar i carregar la informació dels teclats.

Mètodes:

- **getInstance(): CtrlPersistencia:** Com que la classe és singleton hem definit aquest mètode per poder obtenir-ne la instància.
- **guardarDades(DadesDto dades):** Rep com a paràmetre un DTO (Data Transfer Object) que conté tota la informació que la capa de domini li transmet a la capa de dades per a que la emmagatzemi en fitxers de text.
- **carregarDades(): DadesDto:** Agafa totes les dades emmagatzemades als diferents fitxers i les empaqueta en un DTO per transferir-les a la capa de domini.

5. Estructures de dades i algorismes utilitzats

a. ParellsLletres

i. Matriu d'Arrays d'enters: S'utilitza per guardar el flux d'una llista de Freqüències per un Alfabet donat. Els índexs son dos caràcters i conté el número aparicions d'aquest parell de lletres. Principalment utilitzem la matriu només per obtenir un valor concret per tant el cost és només **O(1)**

b. Alfabet

i. Set de Character: S'utilitza per guardar els alfabets. 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, ParaulesFrequencies i ParellsLletres. Com a clau primària té el Nom de cada objecte de les classes que guarda (que els identifiquen). En aquest cas, el cost de buscar un element al mapa, inserir o eliminar es de **O(log n).**

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. Aquesta estructura l'utilitzem per imprimir el teclat el que té un cost de **O(n)**, després també intercanviem elements de l'array pero això és constant **O(1).**

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.

iv. El cost d'aquest algoritme sense tenir en compte la implementació de l'hungarian algorithm que s'explicarà a continuació és de **O(n^4)**

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.

iii. El cost de l'algoritme es en el pitjor dels casos **O(n!)**

g. Genetic Algorithm

i. Llistes d'enters: Utilitzem una llista d'Integers on a cada element de la llista, el valor representa la instal·lació i la posició en la llista la ubicació.

ii. Genetic Algorithm: Aquest algoritme consisteix en, a partir d'una població inicial de cromosomes (possibles solucions), obtenir una nova generació a partir dels millors individus de la població i modificarlos genèticament en una petita quantitat per anar trobant la millor solució després de N generacions.

L'algoritme te diversos passos i els costos corresponents:

1. Creació de la població inicial amb N cromosomes. **O(n * m)**
2. Fitness: Calculem un cost per a cada cromosoma per comprovar com de bona és la solució. **O(n * m^2)**
3. Selection: Mitjançant el Tournament Selection obtenim els individus als quals seran els progenitors de la pròxima generació. **O(n log n)**
4. Crossover: Emparellem dos cromosomes dels seleccionats que resultaran en un fill, utilitzarem el One-Point Crossover. **O(m)**
5. Mutation: Es realitza un canvi random en els gens d'un cromosoma, nosaltres utilitzarem Swap Mutation. Aquests Cromosomes resultants seran la nova generació. **O(1)**
6. Es repeteixen tots els passos desde el 2 fins les iteracions dessitjades. Cada iteració de l'algoritme es una nova generació. **O(i)**

Per tant, analitzant el cost total de l'algoritme trobem que té un cost de **O(n * m^2 * i)** on:

- n: Mida població
- m: mida alfabet
- i: número de generacions dessitjades.