

Descripció dels mètodes i atributs de les classes

Grup 43.4:

Samuel Cilleruelo González (samuel.cilleruelo)

Sara Díaz Morales (sara.diaz.morales)

David Durban Rodrigues (david.durban)

Mateu Villaret Abio (mateu.villaret)

Versió del lliurament: 3.0

Índex

1. Domini
 - 1.1. CtrlDomini
 - 1.2. CtrlEntrada
 - 1.3. CtrlAlgorisme
 - 1.4. Collections
 - 1.5. Entrada
 - 1.6. Text
 - 1.7. LlistaParaules
 - 1.8. Teclat
 - 1.9. Usuari
 - 1.10. Alfabet
 - 1.11. Netejadora
 - 1.12. Processadora
 - 1.13. QAP
 - 1.14. Node
 - 1.15. SimulatedAnnealing

2. Persistència
 - 2.1. CtrlPersistencia
 - 2.2. GestorUsuari
 - 2.3. GestorTeclat
 - 2.4. GestorAlfabet
 - 2.5. GestorEntrada
 - 2.6. GestorIds

3. Presentació
 - 3.1. CtrlPresentacio
 - 3.2. VistaLogin

- 3.3. VistaSingIn
- 3.4. VistaPrincipal
- 3.5. VistaUsuari
- 3.6. VistaTeclat
- 3.7. VistaAlfabet
- 3.8. VistaEntrada

1. Domini

1.1 CtrlDomini

- **Nom:** CtrlDomini
- **Descripció:**
 - Controlador responsable d'establir la comunicació entre la capa de disseny i la de persistència, així com gestiona l'entrada i sortida del programa i tots els casos d'ús.
- **Descripció dels atributs:**
 - ctrl: Instància Singleton
 - usuariActual: Objecte usuari que conté el que està actualment actiu
 - ctrlEntrada: Instància de la classe CtrlEntrada
 - ctrlAlgorisme: Instància de la classe CtrlAlgorisme
 - ctrlPersistencia: Instància de la classe CtrlPersistencia
- **Descripció dels mètodes:**
 - CtrlDomini()::CtrlDomini:
Constructor de domini que instancia els controladors i inicialitza l'usuariActual a null.
 - obtenirInstancia()::static CtrlDomini:
Obté la instància de la classe i la crea en cas de no tenir-la.
 - iniciarSessio(String nomUsuari, String contrasenya)::void:
Carrega l'usuari identificat per nomUsuari de persistència i el guarda a domini, carregant també tots els seus teclats, alfabetes i entrades; si i només si no hi ha cap usuari actiu, en cas contrari es llança una excepció. Si l'usuari identificat per nomUsuari no existeix o la contrasenya és incorrecta es llança de la mateixa forma una excepció.
 - tancarSessio()::void:

L'usuari actiu passa a ser null i s'esborra de domini tota la informació al respecte. Requereix que hi hagi un usuari actiu, en cas contrari llancem excepció.

- `crearUsuari(String nomUsuari , String contrasenya)::void:`

Crea un usuari nou amb el nom i la contrasenya donats i les llistes d'ids buides, i l'assigna a l'atribut de la classe. Es guarda l'usuari també a persistència. Les excepcions que es llancen son: si ja hi ha una sessió iniciada, si ja existeix l'usuari amb `nomUsuari`, si `nomUsuari` és massa curt o massa llarg, si té espais, si la contrasenya és massa curta o massa llarga, si la contrasenya conté espais o si la contrasenya no compleix els requeriments de contrasenya segura.

- `eliminarUsuari():void:`

Elimina tota la informació de l'usuari que té sessió iniciada i l'esborra també de persistència, juntament amb els seus teclats, alfabets i entrades. En cas que no hi hagi un usuari actiu es llança excepció.

- `modificarContrasenyaUsuari(String novaContrasenya)::void:`

Canvia la contrasenya de l'usuari, la qual passa a tenir el contingut que s'indica al paràmetre. Es llança excepció en cas que no hi hagi cap sessió iniciada, i en els mateixos casos on no es compleix que la contrasenya sigui vàlida, afegint el cas que s'introdueixi la mateixa que ja hi havia.

- `consultarTeclats():ArrayList<String>:`

Retorna una llista amb els ids i els noms dels teclats d'un usuari. Si no s'ha iniciat sessió es llança excepció.

- `consultarAlfabets():ArrayList<String>:`

Retorna una llista amb els ids i els noms dels alfabets d'un usuari . Si no s'ha iniciat sessió es llança excepció.

- `consultarEntrades():ArrayList<String>:`

Retorna una llista amb els ids i els noms de les entrades d'un usuari. Si

no s'ha iniciat sessió es llança excepció.

- `crearTeclat(String nomTeclat, int idAlfabet , int idEntrada)::void:`

Crea un nou teclat amb el nom indicat i amb els alfabet i entrades que corresponen als ids passats. Abans de passar-lo a l'algorisme per la creació, es neteja l'entrada de tal forma que es passi en un format correcte. Amb aquesta informació l'algorisme retorna un String amb el contingut del teclat i allà és quan es crea i es guarda a persistència. Si no s'ha iniciat sessió es llança excepció.

- `eliminarTeclat(int idTeclat)::void:`

Elimina el teclat identificat per idTeclat, i mira si ha d'eliminar també el seu alfabet o entrada associats. Si no s'ha iniciat sessió es llança excepció.

- `modificarTeclat(int idTeclat, boolean nouAlfabet, int nould, int opcio)::void:`

Modifica el teclat identificat per idTeclat. Es reassigna un nou alfabet o una nova entrada de forma que es torna a recalculat el contingut del teclat. Si no s'ha iniciat sessió es llança excepció.

- `consultarTeclat(int idTeclat)::String[]:`

Retorna una array d'Strings amb el contingut del teclat identificat per idTeclat. Si no s'ha iniciat sessió es llança excepció.

- `crearAlfabet(String nomAlfabet, String alfabet)::void:`

Crea un nou alfabet amb el nom i contingut entrat com a paràmetres. Si no s'ha iniciat sessió es llança excepció.

- `eliminarAlfabet(String idAlfabet)::void:`

Elimina l'alfabet identificat per idAlfabet en cas que no tingui teclats associats, altrament posa el seu atribut eliminat a cert. Si no s'ha iniciat sessió es llança excepció.

- `modificarAlfabet(int idAlfabet, String alfabet)::void:`

Modifica el contingut de l'alfabet identificat per idAlfabet. Si no s'ha iniciat sessió es llança excepció.

- `consultarAlfabet(int idAlfabet)::String[]:`

Retorna una array d'Strings amb el contingut de l'alfabet identificat per idAlfabet. Si no s'ha iniciat sessió es llança excepció.

- `crearEntrada(String nomEntrada, String tipus, String entrada)::void:`

Crea una nova entrada amb els atributs que indiquen els paràmetres. Si no s'ha iniciat sessió es llança excepció.

- `eliminarEntrada(int idEntrada)::void:`

Elimina l'entrada identificada per idEntrada en cas que no tingui teclats associats, altrament posa el seu atribut eliminat a cert. Si no s'ha iniciat sessió es llança excepció.

- `modificarEntrada(int idEntrada, String entrada)::void:`

Modifica el contingut de l'entrada identificada per idEntrada, que passa a ser l'String donat per paràmetre. Si no s'ha iniciat sessió es llança excepció.

- `consultarEntrada(int idEntrada)::String[]:`

Retorna una array d'Strings amb el contingut de l'entrada identificada per idEntrada. Si no s'ha iniciat sessió es llança excepció.

- `existeixUsuari(String nomUsuari)::boolean:`

Retorna cert si l'usuari identificat per nomUsuari existeix.

- `comprovarContrasenya(String nomUsuari, String contrasenya)::boolean:`

Retorna cert si la contrasenya donada coincideix amb la de l'usuari identificat per nomUsuari.

- `contrasenyaCorrecta(String contrasenya)::boolean:`

Retorna si la contrasenya donada té com a mínim una majúscula, un dígit i un símbol.

1.2. CtrlEntrada

- **Nom:** CtrlEntrada
- **Descripció:**
 - Controlador que s'encarrega de comunicar les classes d'entrada, teclat i alfabet amb les altres classes funcionals i principalment té com a funció la creació, modificació i eliminació de les primeres.
- **Descripció dels atributs:**
 - ctrl: Instància Singleton
 - colleccions: Instància de la classe Collections
 - netejadora: Instància de la classe Netejadora
- **Descripció dels mètodes:**
 - CtrlEntrada()::CtrlEntrada:
Constructor de la classe, instancia les classes Collections i Netejadora.
 - obtenirInstancia()::CtrlEntrada:
Obté la instància de la classe i la crea en cas de no tenir-la.
 - carregarTeclats(String[] teclats)::void:
Crea els teclats donats per l'array d'Strings, on cada element conté tota la informació del teclat, amb els atributs de la classes separats per comes.
 - carregarAlfabet(String[] alfabet)::void:
Crea els alfabet donats per l'array d'Strings, on cada element conté tota la informació de l'alfabet, amb els atributs de la classes separats per comes.
 - carregarEntrades(String[] entrades)::void:
Crea les entrades donades per l'array d'Strings, on cada element conté tota la informació de l'entrada, amb els atributs de la classes separats per comes.
 - carregarSeguentsIds(Integer[] ids)::void:

Carrega a Collections els següents ids a assignar.

- `reset():void`:

Esborra totes les dades de l'usuari que hi ha a Collections.

- `crearTeclat(int idTeclat, String nomTeclat, int idAlfabet, int idEntrada, String teclat)::int`:

Crea un nou teclat amb tots els atributs donats, i en cas que l'idTeclat sigui -1 consulta quin s'ha d'assignar i incrementa la variable. Guarda aquest objecte a Collections. Retorna l'id del teclat creat.

- `eliminarTeclat(int idTeclat)::void`:

Elimina el teclat identificat per idTeclat esborrant-lo de la col·lecció, i el desvincula també de seves associacions. En cas que l'alfabet o entrada associats a aquest teclat tinguessin l'atribut eliminat a cert, s'esborren també de Collections.

- `modificarTeclat(int idTeclat, boolean nouAlfabet, int nould, String teclat)::void`:

Modifica el contingut del teclat reassignant l'alfabet o l'entrada, i assigna el seu id al nou alfabet/entrada associat. En cas que l'alfabet/entrada antics passin a no tenir cap teclat associat i tinguin l'atribut eliminat a cert, s'esborren de Collections.

- `consultarTeclat(int idTeclat)::String[]`:

Retorna una array d'Strings amb tota la informació del teclat identificat per idTeclat.

- `consultarIdAlfabet(int idTeclat)::int`:

Retorna l'id de l'alfabet associat al teclat identificat per idTeclat.

- `consultarIdEntrada(int idTeclat)::int`:

Retorna l'id de l'entrada associada al teclat identificat per idTeclat.

- `comprovarNomTeclat(String nomTeclat)::void`:

Llança excepció si l'usuari té algun teclat amb nom=nomTeclat.

- `consultarNomTeclat(int idTeclat)::String:`

Retorna el nom del teclat identificat per idTeclat.

- `crearAlfabet(int idAlfabet, String nomAlfabet, boolean eliminat, ArrayList<Integer> idTeclats, String alfabet)::int:`

Crea un nou alfabet amb tots els atributs donats, i en cas que l'idAlfabet sigui -1 consulta quin s'ha d'assignar i incrementa la variable. Guarda aquest objecte a Collections. Retorna l'id de l'alfabet creat.

- `eliminarAlfabet(int idAlfabet)::boolean:`

Elimina l'alfabet identificat per idAlfabet. Si l'alfabet no té teclats associats s'esborra directament, en cas contrari el marca com a eliminat mitjançant un boolean, però es conserva fins que no tingui cap teclat associat només essent visible pels seus teclats associats, però no per l'usuari. Retorna cert si l'alfabet té algun teclat associat.

- `modificarAlfabet(int idAlfabet, String alfabet)::void:`

Modifica el contingut de l'alfabet identificat per idAlfabet. Si el contingut del nou alfabet és idèntic al de l'antic es llança excepció.

- `consultarAlfabet(int idAlfabet)::String[]:`

Retorna una array d'Strings amb tota la informació de l'alfabet identificat per idAlfabet. Si l'alfabet encara existeix, però té l'atribut eliminat a cert es llança excepció.

- `consultarContingutAlfabet(int idAlfabet)::String:`

Retorna un String amb el contingut de l'alfabet identificat per idAlfabet.

- `obtenirAlfabetNetejat(int idAlfabet)::Set<Character>:`

Retorna l'alfabet identificat per idAlfabet netejat per tal que l'algorisme pugui crear el teclat.

- `comprovarNomAlfabet(String nomAlfabet)::void:`

Llança excepció si l'usuari té algun alfabet amb nom=nomAlfabet.

- alfabetEliminat(int idAlfabet)::Boolean:

Retorna cert si l'alfabet identificat per idAlfabet està marcat com a eliminat.

- consultarNomAlfabet(int idAlfabet)::String:

Retorna el nom de l'alfabet identificat per idAlfabet.

- crearEntrada(int idEntrada, String nomEntrada, boolean eliminat, String tipus, ArrayList<Integer> idTeclats, String contingutEntrada):: int:

Crea una nova entrada amb tots els atributs donats, i en cas que l'idEntrada sigui -1 consulta quin s'ha d'assignar i incrementa la variable. Guarda aquest objecte a Collections. Retorna l'id de l'entrada creada.

- eliminarEntrada(int idEntrada): boolean

Elimina l'entrada identificada per idEntrada. Si l'entrada no té teclats associats s'esborra directament, en cas contrari el marca com a eliminada mitjançant un boolean, però es conserva fins que no tingui cap teclat associat només essent visible pels seus teclats associats, però no per l'usuari. Retorna cert si l'entrada té algun teclat associat.

- modificarEntrada(int idEntrada, String entrada)::void:

Modifica el contingut de l'entrada identificada per idEntrada. Si el contingut de la nova entrada és idèntic al de l'antiga es llança excepció.

- consultarEntrada(int idEntrada)::String[]:

Retorna una array d'Strings amb tota la informació de l'entrada identificada per idEntrada. Si l'entrada encara existeix, però té l'atribut eliminat a cert es llança excepció.

- obtenirEntradaNetejada(int idEntrada, Set<Character> alfabet):: HashMap<String, Integer>:

Retorna un HashMap que conté l'entrada identificada per idEntrada netejada, segons el contingut de l'alfabet donat.

- `comprovarNomEntrada(String nomEntrada)::void:`
Comprova que no existeix una entrada amb el mateix nom que `nomEntrada` a l'usuari.
- `entradaEliminada(int idEntrada)::Boolean:`
Retorna cert si l'entrada identificada per `idEntrada` està marcada com a eliminada.
- `consultarNomEntrada(int idEntrada)::String:`
Retorna el nom de l'entrada identificada per `idEntrada`.

1.3. CtrlAlgorisme

- **Nom:** CtrlAlgorisme
- **Descripció:**
 - Controlador que s'encarrega de comunicar les classes processadora, QAP i SimulatedAnnealing amb les altres classes i principalment té com a funció la crida a l'algorisme, la qual ens permet obtenir el necessari per a poder formar el teclat.
- **Descripció dels atributs:**
 - `ctrlalgorisme`: Instància Singleton
 - `qap`: Instància de la classe QAP
 - `sa`: Instància de la classe SimulatedAnnealing
 - `processadora`: Instància de la classe Processadora
- **Descripció dels mètodes:**
 - `CtrlAlgorisme()::CtrlAlgorisme:`
Constructor privat que instancia les classes associades.
 - `obtenirInstancia()::static CtrlAlgorisme:`
Obté la instància de la classe i la crea en cas de no tenir-la.
 - `crearTeclat(alfabet: Set<Character>, entrada: HashMap<String,Integer> , int opcio):: Character[][]:`

Crea una matriu de chars, fa una crida a processadora per generar el paràmetre que es passa a l'algorisme. S'escull l'algorisme a utilitzar segons el paràmetre opcio. Es processa el que retorna l'algorisme perquè tingui el format desitjat i el resultat és el que es retorna.

1.4. Collections

- **Nom:** Collections
- **Descripció:**
 - La classe Collections s'utilitza per emmagatzemar les col·leccions de teclats, alfabet i entrades d'un usuari, a més de guardar i actualitzar els següents ids a assignar d'aquestes classes.
- **Descripció dels atributs:**
 - coleccio: Instància Singleton
 - teclats: HashMap que emmagatzema els objectes Teclats d'un usuari amb idTeclat com a clau
 - alfabet: HashMap que emmagatzema els objectes Alfabet d'un usuari amb idAlfabet com a clau
 - entrades: HashMap que emmagatzema els objectes Entrada d'un usuari amb idEntrada com a clau
 - nomsTeclats: Llista de Strings que conté tots els noms dels teclats d'un usuari
 - nomsAlfabet: Llista de Strings que conté tots els noms dels alfabet d'un usuari
 - nomsEntrades: Llista de Strings que conté tots els noms de les entrades d'un usuari
 - seguentIdTeclat: Enter que representa l'id a assignar al pròxim teclat
 - seguentIdAlfabet: Enter que representa l'id a assignar al pròxim alfabet
 - seguentIdEntrada: Enter que representa l'id a assignar a la següent entrada

- **Descripció dels mètodes:**

- Collections():Collections:

Creadora de la classe que inicialitza les llistes buides i posa els següents ids a assignar a 1.

- obtenirInstancia():Collections:

Obté la instància de la classe i la crea en cas de no tenir-la.

- consultarTeclat(int idTeclat)::Teclat:

Retorna l'objecte teclat identificat per idTeclat, en cas de no existir llança excepció.

- consultarAlfabet(int idAlfabet)::Alfabet:

Retorna l'objecte alfabet identificat per idAlfabet, en cas de no existir llança excepció.

- consultarEntrada(int idEntrada)::Entrada:

Retorna l'objecte entrada identificat per idEntrada, en cas de no existir llança excepció.

- consultarNomTeclat(String nomTeclat)::boolean:

Retorna cert si a la llista teclats existeix un teclat amb nom = nomTeclat.

- consultarNomAlfabet(String nomAlfabet)::boolean:

Retorna cert si a la llista alfabet existeix un alfabet amb nom = nomAlfabet.

- consultarNomEntrada(String nomEntrada)::boolean:

Retorna cert si a la llista entrades existeix una entrada amb nom = nomEntrada.

- consultarSeguentIdTeclat():int:

Retorna el següent id de teclat a assignar.

- consultarSeguentIdAlfabet():int:

Retorna el següent id d'alfabet a assignar.

- `consultarSeguentIdEntrada():int:`
Retorna el següent id d'entrada a assignar.
- `reset():` Buida tots els atributs de la classe, excepte els ids.
- `afegirTeclat(int idTeclat, String nomTeclat, Teclat teclat)::void:`
Afegeix a teclats l'objecte teclat amb clau idTeclat, i afegeix el nom del teclat a nomsTeclats.
- `afegirAlfabet(int idAlfabet, String nomAlfabet, Alfabet alfabet)::void:`
Afegeix a alfabets l'objecte alfabet amb clau idAlfabet, i afegeix el nom de l'alfabet a nomsAlfabets.
- `afegirEntrada(int idEntrada, String nomEntrada, Entrada entrada)::void:`
Afegeix a entrades l'objecte entrada amb clau idEntrada, i afegeix el nom de l'entrada a nomsEntrades.
- `eliminarTeclat(int idTeclat, String nomTeclat)::void:`
Elimina de teclats l'element amb clau idTeclat i treu també el seu nom de la llista nomsTeclats.
- `eliminarAlfabet(int idAlfabet, String nomAlfabet)::void:`
Elimina d'alfabets l'element amb clau idAlfabet i treu també el seu nom de la llista nomsAlfabets.
- `eliminarEntrada(int idEntrada, String nomEntrada)::void:`
Elimina d'entrades l'element amb clau idEntrada i treu també el seu nom de la llista nomsEntrades.
- `assignarIds(Integer[] ids)::void:`
Reassigna el contingut del següents ids a assignar als valors que hi ha al paràmetre ids, on ids[0] correspon al de teclats, ids[1] el d'alfabets i ids[2] al d'entrades.

1.5.Entrada

- **Nom:** Entrada
- **Descripció:**
 - La classe Entrada és una classe abstracta que fa de base per a les entrades tipus text i llista de paraules. Representa una entrada que conté una llista de paraules amb les seves freqüències associades.
- **Descripció dels atributs:**
 - idEntrada: Identificador de l'entrada
 - nom: Nom de l'entrada
 - eliminat: Booleà que indica si l'entrada ha estat eliminada
 - LP: HashMap de les paraules i les seves freqüències
 - idTeclats: Llista d'identificadors dels seus teclats associats
- **Descripció dels mètodes:**
 - Entrada():Entrada:
Creadora de la classe.
 - consultarLlistaParaules():HashMap<String, Integer>:
Retorna el contingut de l'entrada en format llista de paraules.
 - afegirIdTeclat(Integer id)::void:
Afegeix l'id passat com a paràmetre a la llista idTeclats.
 - consultarNomEntrada():String:
Retorna el nom de l'entrada.
 - consultarEliminat():boolean:
Retorna el valor de l'atribut eliminat.
 - consultarIdTeclat(Integer idTeclat)::boolean:
Retorna cert si l'idTeclat està dins de la llista d'ids.
 - teTeclats():boolean:
Retorna cert si té algun teclat associat.

- eliminarEntrada():void:
Posa l'atribut eliminat a cert.
- eliminarIdTeclat(Integer id)::void:
Elimina l'id passat com a paràmetre de la llista idTeclats.
- modificarContingut(HashMap<String, Integer> LP)::void:
Modifica l'entrada i elimina les referències de teclat.
- ompleHashMap(String llista)::HashMap<String, Integer>:
Converteix les dades d'String a HashMap, i retorna això.

1.6. Text

- **Nom:** Text
- **Descripció:**
 - La classe Text és una subclasse d'Entrada i s'ocupa principalment de la manipulació del seu contingut.
- **Descripció dels atributs:**
 - T: cadena de caràcters que guarda el contingut de l'entrada tipus text
- **Descripció dels mètodes:**
 - Text(int idEntrada, String nomEntrada, boolean eliminat, ArrayList<Integer> idTeclats, String x)::Text:
Constructora de la classe Text que inicialitza tots els atributs de la superclasse.
 - consultarText():String:
Retorna el contingut del text.
 - modificar_original(String s)::void:
Modifica text per un nou i crida a la seva super classe per actualitzar també la Llista de paraules creades a partir d'aquest.
 - textaLp():void:

Transforma l'entrada text en un map que representa la Llista de paraules amb freqüències.

1.7. LlistaParaules

- **Nom:** LlistaParaules
- **Descripció:**
 - La classe LlistaParaules hereda la classe Entrada i representa les funcions de Llista de paraules amb freqüències.
- **Descripció dels atributs:**
- **Descripció dels mètodes:**
 - `LlistaParaules(int id, String nomEntrada, boolean eliminat, ArrayList<Integer> idTeclats, String llista)::LlistaParaules:`
Constructora de la classe LlistaParaules que inicialitza tots els atributs de la superclasse.
 - `modificar_original(String s)::void:`
Neteja el contingut de LP i crida a la super classe per actualitzar la Llista de paraules creades a partir de l'String entrat.

1.8. Teclat

- **Nom:** Teclat
- **Descripció:**
 - La classe Teclat representa un teclat amb la distribució més òptima segons l'algorisme triat per un cert alfabet i una certa entrada.
- **Descripció dels atributs:**
 - `idTeclat:` Identificador del teclat
 - `nomTeclat:` Nom del teclat
 - `idAlfabet:` Identificador de l'alfabet al qual està associat
 - `idEntrada:` Identificador de l'entrada al qual està associada

- teclat: Matriu de caràcters que representa la disposició de les tecles del teclat
- **Descripció dels mètodes:**
 - Teclat(Integer idTeclat, String nomTeclat, Integer idAlfabet, Integer idEntrada, String teclat)::Teclat:
Creadora de la classe que inicialitza els atributs amb els paràmetres passats i converteix el teclat a matriu de caràcters.
 - consultarTeclat()::Character[][]:
Retorna el contingut del teclat en forma matricial.
 - consultarNomTeclat()::String:
Retorna el nom del teclat.
 - consultarIdEntrada()::Integer:
Retorna l'id de l'entrada associada.
 - consultarIdAlfabet()::Integer:
Retorna l'id de l'alfabet associat.
 - reassignarEntrada(String teclat , Integer idEntrada)::void:
Reassigna l'entrada associada al teclat i actualitza la matriu de caràcters amb la nova recalculació de la disposició.
 - reassignarAlfabet(String teclat, Integer idAlfabet)::void:
Reassigna l'alfabet associat al teclat i actualitza la matriu de caràcters amb la nova recalculació de la disposició.

1.9. Usuari

- **Nom:** Usuari
- **Descripció:**
 - La classe Usuari s'ocupa de guardar la informació de cada usuari.
- **Descripció dels atributs:**

- nomUsuari: Identificador de l'usuari
- contrasenya: Contrasenya de l'usuari
- teclats: Llista que conté els ids de tots els teclats que té l'usuari
- alfabet: Llista que conté els ids de tots els alfabetes que té l'usuari.
- entrades: Llista que conté els ids de totes les entrades que té l'usuari.
- **Descripció dels mètodes:**
 - Usuari(String nom, String contrasenya)::Usuari
Constructora d'usuari que inicialitza els seus atributs amb els paràmetres passats, excepte les llistes que s'inicialitzen com una array buida.
 - consultarNomUsuari()::String
Retorna el nom de l'usuari.
 - consultarContrasenya()::String
Retorna la contrasenya de l'usuari.
 - consultarTeclats()::ArrayList<Integer>
Retorna la llista que conté els ids de tots els teclats que té l'usuari.
 - consultarAlfabet()::ArrayList<Integer>
Retorna la llista que conté els ids de tots els alfabetes que té l'usuari.
 - consultarEntrades()::ArrayList<Integer>
Retorna la llista que conté els ids de totes les entrades que té l'usuari.
 - modificarContrasenya(String novaContrasenya)::void:
Modifica la contrasenya de l'usuari, canviant-la per novaContrasenya.
 - afegirTeclat(Integer idTeclat)::void:
Afegeix l'identificador de teclat idTeclat a la llista teclats.
 - afegirAlfabet(Integer idAlfabet)::void:
Afegeix l'identificador d'alfabet idAlfabet a la llista alfabetes.
 - afegirEntrada(Integer idEntrada)::void:
Afegeix l'identificador d'entrada idEntrada a la llista entrades.

- `eliminarTeclat(Integer idTeclat)::void:`
Elimina l'identificador de teclat `idTeclat` de la llista teclats.
- `eliminarAlfabet(Integer idAlfabet)::void:`
Elimina l'identificador d'alfabet `idAlfabet` de la llista alfabet.
- `eliminarEntrada(Integer idEntrada)::void:`
Elimina l'identificador d'entrada `idEntrada` de la llista entrades.

1.10. Alfabet

- **Nom:** Alfabet
- **Descripció:**
 - La classe Alfabet guarda la informació corresponent als alfabet.
- **Descripció dels atributs:**
 - `idAlfabet`: Identificador de l'alfabet
 - `nomAlfabet`: Nom de l'alfabet
 - `eliminat`: Booleà que indica si l'alfabet ha estat eliminat
 - `idTeclats`: Llista d'identificadors dels teclats associats
 - `alfabet`: Conjunt de caràcters únics que formen l'alfabet
- **Descripció dels mètodes:**
 - `Alfabet(int idAlfabet, String nomAlfabet, Boolean eliminat, ArrayList<Integer> idTeclats, String alfabet)::Alfabet:`
Constructor de Alfabet que inicialitza tots els atributs de la classe i converteix l'alfabet a un Set de caràcters.
 - `afegirIdTeclat(Integer idTeclat)::void:`
Afegeix l'id passat com a paràmetre a la llista `idTeclats`.
 - `consultarIdTeclat(Integer idTeclat)::boolean:`
Retorna cert si l'`idTeclat` està dins de la llista d'ids.
 - `consultarAlfabet()::Set<Character>:`
Retorna el contingut de l'alfabet.

- `consultarNomAlfabet():String`:
Retorna el nom de l'alfabet.
- `consultarEliminat():boolean`:
Retorna el valor de l'atribut eliminat.
- `eliminarAlfabet():void`:
Posa l'atribut eliminat a cert.
- `eliminarIdTeclat(Integer idTeclat):void`:
Elimina l'id passat com a paràmetre de la llista idTeclats.
- `modificarAlfabet(Set<Character> nouAlfabet):void`:
Modificar els caràcters de l'alfabet, mantenint el mateix id, nom i la resta d'atributs.
- `teTeclats():boolean`:
Retorna cert si té algun teclat associat.
- `netejarAlfabet(Set<Character> abc):: Set<Character>`:
Retorna l'alfabet netejat, és a dir acceptant només caràcters que siguin lletres. El paràmetre d'entrada es l'alfabet brut.

1.11. Netejadora

- **Nom:** Netejadora
- **Descripció:**
 - Classe encarregada de netejar tan l'alfabet com l'entrada abans de crear un teclat.
- **Descripció dels atributs:**
- **Descripció dels mètodes:**
 - `netejarEntrada(Set<Character> AbcNet, HashMap<String, Integer> LP) ::HashMap<String, Integer>`:
Neteja l'entrada a partir de l'alfabet net i la retorna.

- `esTrobaAbc(Set<Character> Abc, Character c)::boolean:`
Retorna cert si un caràcter es troba a l'alfabet.
- `netejarLletres(Character c)::Character:`
Converteix un caràcter que sigui una lletra a minúscula sense accents (les neteja) i retorna el resultat.

1.12. Processadora

- **Nom:** Processadora
- **Descripció:**
 - Classe encarregada de convertir les dades que li arriben de fora (Alfabet i Entrada netejats) al format que necessita l'algorisme per treballar.
- **Descripció dels atributs:**
 - `processadora:` Instància Singleton
- **Descripció dels mètodes:**
 - `Processadora()::Processadora:`
Constructora de Processadora.
 - `permAPos(Set<Character> alfabet, int[] permutacio)::String:`
Transforma la permutació que genera l'algorisme en un String per a que el teclat rebi el paràmetre desitjat, i retorna aquest String.
 - `ProcessaDades(Set<Character> Alfabet, Map<String, Integer> LP) ::int[][]:`
Converteix Alfabet i Entrada en una matriu d'enters amb les repeticions de cada lletra. Aquesta matriu ens permet saber la freqüència que una lletra es trobi rere un altre.
 - `obtenirInstancia()::Processadora:`
Obté la instància de la classe i la crea en cas de no tenir-la.

1.13. QAP

- **Nom:** QAP
- **Descripció:**
 - Classe encarregada de calcular les dades necessàries que ens permeten formar el teclat fent ús de l'algorisme QAP.
- **Descripció dels atributs:**
 - qap: Instància Singleton
 - matrix: Matriu d'instal·lacions i tràfic
 - distancia: Matriu de distàncies de les ubicacions
 - permutacions: Millor distribució d'instal·lacions trobada. Registra a on s'ubica cada instal·lació
 - CostCota: Cost Cota de la millor distribució trobada
 - N: Nombre d'instal·lacions
 - tamSol: Registra el nombre mínim de línies o columnes per cobrir tots els zeros d'una matriu. És utilitzat al Hungarian Algorithm
 - SquareInRow: Registra si una fila està marcada. És utilitzat al Hungarian Algorithm
 - SquareInCol: Registra si una columna està marcada. És utilitzat al Hungarian Algorithm
- **Descripció dels mètodes:**
 - QAP()::QAP:
Constructora de QAP
 - ObtenirInstancia()::QAP:
Obté la instància de la classe i la crea en cas de no tenir-la.
 - ejecutarQAP(mat: int[][]):int[]:
Mètode d'optimització QAP.
 - evolution()::void:
Calcula els successors al primer símbol per a la construcció de la aproximació greedy.

- `first_candidate():void`:
Calcula el primer símbol a colocar en una instal·lació. Utilitzat pel greedy.
- `Primer_terme(perm: int[] , dist: double[][]):double`:
Calcula el primer terme de la cota de Gilmore-Lawler. S'utilitza durant el branch & bound.
- `crear_matdis():void`:
Crea la matriu de distàncies entre ubicacions. S'utilitzarà per calcular el cost de colocar una instal·lació en una ubicació.
- `cost_newnode(perm: int[] , dist: double[][] , indx: int):double`:
Calcula el cost que suposaria colocar una instal·lació no emplaçada, en la k-ésima ubicació.
- `crear_mathung(perm: int[] , colocats: int):double[][]`:
Crea la matriu amb tots els costs que suposaria colocar una instal·lació no emplaçada, en les ubicacions lliures. Es farà ús d'aquesta matriu en el hungarian.
- `Treure_minims(mat: double[][]):void`:
Donat la matriu parametre, treu el mínim de cada fila a tota la fila i el mínim de cada columna a tota la columna.
- `Lineas_minimo(mat: double[][]):void`:
Gestiona el càlcul de la assignació més completa possible.
- `Asig_zeros(mat: double[][], fasig: ArrayList<Point> , level: int , rowHasSquare: int[], colHasSquare: int[]):void`:
Obté l'assignació més completa possible, és a dir amb el major nombre de files amb un zero assignat. Ho fa amb l'ús de backtracking.
- `MarcarCol(mat: double[][] , MarcColumnes: int[] , MarcFiles: int[]):void`:
Marca totes les columnes que tinguin algun zero en una fila marcada. Si ha marcat una columna no marcada, crida a MarcarFiles.
- `MarcarFiles(mat: double[][] , MarcColumnes: int[] , MarcFiles: int[]):void`:

Marca totes les files que tinguin una assignació en una columna marcada. Si ha marcat una fila no marcada, crida a MarcarColumnes.

- ValorMinNoCobert(double[][] mat, int[] MarcColumnes, int[] MarcFiles)::double:

Troba el valor mínim no cobert de la matriu.

- AjustarMatriu(mat: double[][] , MarcColumnes: int[] , MarcFiles: int[] , valMin: double):: void:

Resta el valor mínim no cobert als nombres no coberts i suma el valor mínim no cobert als nombres coberts per dues línies.

- PerCobrir(mat: double[][])::void:

Gestiona el càlcul d'AjustarMatriu, generant els paràmetres pertinents.

- Hungarian_Algorithm(mat: double[][])::double: Fa el càlcul del Hungarian Algorithm.

1.14. Node

- **Nom:** Node

- **Descripció:**

- Classe encarregada de conservar les dades necessàries d'un estat. Serà utilitzada durant el branch & bound del algorisme QAP.

- **Descripció dels atributs:**

- permutacions(int[]): Registra la permutació del node, és a dir les instal·lacions i les ubicacions
- CostCota: Double que registra el cost de cota de la permutació
- colocats: int que registra quantes instal·lacions estan ubicades
- level: int que registra la primera ubicació lliure

- **Descripció dels mètodes:**

- creadora(n:int):

Creadora del primer Node.

- creadora(p:int[],cost:double,col:int,lev:int):

Creadora de Node sucesor.

- `consultar_perm(i:int)::int:`
Retorna el valor de la ubicació de la instal·lació i.
- `consultar_totperm()::int[]:`
Retorna la permutació.
- `consultar_totperm_length()::int:`
Retorna el nombre d'instal·lacions total.
- `consultar_level()::int:`
Retorna el level.
- `consultar_cost()::double:`
Retorna el CostCota.
- `consultar_colocats()::int:`
Retorna el nombre de instal·lacions col·locades.

1.15. SimulatedAnnealing

- **Nom:** SimulatedAnnealing
- **Descripció:**
 - Classe encarregada de calcular les dades necessàries que ens permeten formar el teclat fent ús de l'algorisme simulated annealing.
- **Descripció dels atributs:**
 - `sa`: Atribut que és una instància estàtica de la pròpia classe per implementar el singleton
- **Descripció dels mètodes:**
 - `SimulatedAnnealing()::SimulatedAnnealing:`
Constructora de SimulatedAnnealing.
 - `ObtenirInstancia()::SimulatedAnnealing:`
Obté instància singleton de SimulatedAnnealing.

- `ejecutarSimulatedAnnealing(mat: int[][]):int[]:`
Execució de l'algorisme de cerca local SimulatedAnnealing.
- `getHeuristicValue(perm: int[] , dist: double[][], matrix: int[][]):double:`
Càlcul primer terme de la cota de Gilmore-Lawler. S'utilitza com a heurístic per poder realitzar la cerca local.
- `crear_matdis(distancia: double[][]):void:`
Crea la matriu de distàncies entre ubicacions. S'utilitza per calcular el cost de colocar una instal·lació en una ubicació.
- `swap(idxi:int,idxj:int,perm:int[]):void:`
Operador que s'utilitza durant l'algorisme, intercanvia la posició de dues tecles.
- `evolution(matrix:int[][],permutacions:int[]):void:`
Calcula els successors al primer símbol per a la construcció de la aproximació greedy.
- `first_candidate(matrix:int[][],perm:int[]):void:`
Calcula el primer símbol a colocar en una instal·lació. Utilitzat per el greedy.
- `getTemp(t: int):double:`
Obté la temperatura de l'algorisme.

2. Persistència

2.1 CtrlPersistencia

- **Nom:** CtrlPersistencia
- **Descripció:**
 - Controlador responsable d'establir la comunicació entre la capa de domini i les dades guardades en disc.
- **Descripció dels atributs:**
 - ctrlPersistencia: Instància Singleton
 - gstUsuari: Instància de la classe GestorUsuari
 - gstTeclat: Instància de la classe GestorTeclat
 - gstAlfabet: Instància de la classe GestorAlfabet
 - gstEntrada: Instància de la classe GestorEntrada
 - gstIds: Instància de la classe GestorIds
- **Descripció dels mètodes:**
 - CtrlPersistencia():::CtrlPersistencia:
Creadora de la classe que inicialitza tots els gestors de persistència.
 - obtenirInstancia():::static CtrlPersistencia:
Obté la instància de la classe i la crea en cas de no tenir-la.
 - carregarSeguentsIds():::Integer[]:
Retorna els següents ids a carregar en una array d'enters, d'aquesta forma: [idTeclat, idAlfabet, idEntrada].
 - carregarUsuari(String nomUsuari)::String:
Retorna un String amb les dades de l'usuari identificat per nomUsuari.
 - carregarContrasenya(String nomUsuari)::String:
Retorna la contrasenya de l'usuari identificat per nomUsuari.
 - guardarUsuari(String nomUsuari, String contrasenya)::void:

Guarda a disc un les dades d'un nou usuari amb el nom i contrasenya indicats, i els vectors d'ids inicialment buits.

- `eliminarUsuari(String nomUsuari)::void:`

Elimina de persistència l'usuari identificat per `nomUsuari`, així com els seus teclats, alfabet i entrades.

- `modificarContrasenya(String novaContrasenya, String nomUsuari)::void:`

Modifica a disc la contrasenya de l'usuari identificat per `nomUsuari`.

- `carregarTeclats(String nomUsuari)::String[]:`

Retorna la informació de tots els teclats de l'usuari identificat per `nomUsuari`.

- `guardarTeclat(int idTeclat, String nomTeclat, int idAlfabet, int idEntrada, String teclat, String nomUsuari)::void:`

Guarda a disc les dades d'un nou teclat amb la informació passada als paràmetres; també afegeix l'id del teclat a l'alfabet i l'entrada associats, així com a l'usuari que l'ha creat.

- `eliminarTeclat(int idTeclat, String nomUsuari)::void:`

Elimina de persistència el teclat identificar per `idTeclat`, eliminant també aquest id de les llistes d'ids de teclats del seu alfabet i entrada associats, així com de l'usuari al qual pertany.

- `modificarTeclat(int idTeclat, boolean nouAlfabet, int nould, String nouTeclat)::void:`

Modifica a disc el teclat identificat per `idTeclat`, canviant el seu contingut i assignant-li un nou alfabet en cas que el booleà passat com a paràmetre sigui cert, altrament es reassigna l'entrada. Es fan els canvis necessaris també en les llistes d'ids de teclats d'alfabet o entrada.

- `carregarAlfabet(String nomUsuari)::String[]:`

Retorna la informació de tots els alfabet de l'usuari identificat per `nomUsuari`.

- `guardarAlfabet(int idAlfabet, String nomAlfabet, String alfabet, String nomUsuari)::void:`
 Guarda a disc les dades d'un nou alfabet amb la informació donada per mitjà dels paràmetres, i s'afegeix l'idAlfabet a la llista d'ids dels alfabet de l'usuari que l'ha creat.
- `eliminarAlfabet(int idAlfabet, String nomUsuari)::void:`
 Elimina de persistència l'alfabet identificat per idAlfabet i el seu id de la llista d'ids d'alfabets de l'usuari creador en cas que no tingui cap teclat associat, altrament posa el seu atribut eliminat a cert.
- `modificarAlfabet(int idAlfabet, String contingutAlfabet)::void:`
 Modifica a disc el contingut de l'alfabet identificat per idAlfabet.
- `carregarEntrades(String nomUsuari)::String[]:`
 Retorna la informació de totes les entrades de l'usuari identificat per nomUsuari.
- `guardarEntrada(int idEntrada, String nomEntrada, String tipus, String entrada, String nomUsuari)::void:`
 Guarda a disc les dades d'una nova entrada amb la informació proporcionada pels paràmetres, i afegeix l'id de l'entrada a la llista d'ids d'entrades de l'usuari creador.
- `eliminarEntrada(int idEntrada, String nomUsuari)::void:`
 Elimina de disc l'entrada identificada per idEntrada, igualment amb el seu id de la llista d'ids d'entrades de l'usuari que l'ha creat en cas que el seu atribut eliminat estigui a cert. En cas contrari, posa aquest atribut a cert.
- `modificarEntrada(int idEntrada, String contingutEntrada)::void:`
 Modifica a persistència el contingut de l'entrada identificada per idEntrada.

2.2 GestorUsuari

- **Nom:** GestorUsuari
- **Descripció:**
 - Classe que s'encarrega de carregar, guardar i mantenir actualitzat a disc tota la informació dels usuaris.
- **Descripció dels atributs:**
 - `gstUsuari`: Instància Singleton
 - `path`: String que conté la ruta on es troba el fitxer on s'emmagatzemen tots els usuaris
- **Descripció dels mètodes:**
 - `GestorUsuari()::GestorUsuari`:
Creadora de la classe, crea un nou fitxer que té com a nom "usuaris.txt" i inicialitza la variable `path` al camí corresponent al fitxer.
 - `obtenirInstancia()::static GestorUsuari`:
Obté la instància de la classe i la crea en cas de no tenir-la.
 - `carregarUsuari(String nomUsuari)::String`:
Retorna tota la informació de l'usuari identificat per `nomUsuari`, fent una cerca al fitxer i agafant tota la línia quan el troba.
 - `guardarUsuari(String nomUsuari, String contrasenya, ArrayList<Integer> teclats, ArrayList<Integer> alfabets, ArrayList<Integer> entrades)::void`:
Guarda tota la informació proporcionada via paràmetres d'un usuari en format CSV a la última línia del fitxer indicat per la variable `path`.
 - `eliminarUsuari(String nomUsuari)::void`:
Elimina tota la informació de l'usuari identificat per `nomUsuari` del fitxer que es troba a la variable `path`, esborrant tota la seva línia.

2.3 GestorTeclat

- **Nom:** GestorTeclat
- **Descripció:**
 - Classe que s'encarrega de carregar, guardar i mantenir actualitzat a disc tota la informació dels teclats.
- **Descripció dels atributs:**
 - `gstTeclat`: Instància Singleton
 - `path`: String que conté la ruta on es troba el fitxer on s'emmagatzemen tots els teclats
- **Descripció dels mètodes:**
 - `GestorTeclat()::GestorTeclat`:
Creadora de la classe, crea un nou fitxer que té com a nom "teclats.txt" i inicialitza la variable `path` al camí corresponent al fitxer.
 - `obtenirInstancia()::static GestorTeclat`:
Obté la instància de la classe i la crea en cas de no tenir-la.
 - `carregarTeclat(int idTeclat)::String`:
Retorna tota la informació del teclat identificat per `idTeclat`, fent una cerca al fitxer i agafant tota la línia quan el troba.
 - `guardarTeclat(int idTeclat, String nomTeclat, int idAlfabet, int idEntrada, String contingutTeclat)::void`:
Guarda tota la informació proporcionada via paràmetres d'un teclat en format CSV a la última línia del fitxer indicat per la variable `path`.
 - `eliminarTeclat(int idTeclat)::void`:
Elimina tota la informació del teclat identificat per `idTeclat` del fitxer que es troba a la variable `path`, esborrant tota la seva línia.

2.4 GestorAlfabet

- **Nom:** GestorAlfabet
- **Descripció:**
 - Classe que s'encarrega de carregar, guardar i mantenir actualitzat a disc tota la informació dels alfabetes.
- **Descripció dels atributs:**
 - `gstAlfabet`: Instància Singleton
 - `path`: String que conté la ruta on es troba el fitxer on s'emmagatzemen tots els alfabetes
- **Descripció dels mètodes:**
 - `GestorAlfabet()::GestorAlfabet`:
Creadora de la classe, crea un nou fitxer que té com a nom "alfabets.txt" i inicialitza la variable `path` al camí corresponent al fitxer.
 - `obtenirInstancia()::static GestorAlfabet`:
Obté la instància de la classe i la crea en cas de no tenir-la.
 - `carregarAlfabet(int idAlfabet)::String`:
Retorna tota la informació de l'alfabet identificat per `idAlfabet`, fent una cerca al fitxer i agafant tota la línia quan el troba.
 - `guardarAlfabet(int idAlfabet, String nomAlfabet, Boolean eliminat, ArrayList<Integer> idTeclats, String contingutAlfabet)::void`:
Guarda tota la informació proporcionada via paràmetres d'un alfabet en format CSV a la última línia del fitxer indicat per la variable `path`.
 - `eliminarAlfabet(int idAlfabet)::void`:
Elimina tota la informació de l'alfabet identificat per `idAlfabet` del fitxer que es troba a la variable `path`, esborrant tota la seva línia.

2.5 GestorEntrada

- **Nom:** GestorEntrada
- **Descripció:**
 - Classe que s'encarrega de carregar, guardar i mantenir actualitzat a disc tota la informació de les entrades.
- **Descripció dels atributs:**
 - `gstEntrada`: Instància Singleton
 - `path`: String que conté la ruta on es troba el fitxer on s'emmagatzemen totes les entrades
- **Descripció dels mètodes:**
 - `GestorEntrada()::GestorEntrada`:
Creadora de la classe, crea un nou fitxer que té com a nom "entrades.txt" i inicialitza la variable `path` al camí corresponent al fitxer.
 - `obtenirInstancia()::static GestorEntrada`:
Obté la instància de la classe i la crea en cas de no tenir-la.
 - `carregarEntrada(int idEntrada)::String`:
Retorna tota la informació de l'entrada identificada per `idEntrada`, fent una cerca al fitxer i agafant tota la línia quan el troba.
 - `guardarEntrada(int idEntrada, String nomEntrada, Boolean eliminat, String tipus, ArrayList<Integer> idTeclats, String contingutEntrada)::void`:
Guarda tota la informació proporcionada per mitjà de paràmetres d'una entrada en format CSV a la última línia del fitxer indicat per la variable `path`.
 - `eliminarEntrada(int idEntrada)::void`:
Elimina tota la informació de l'entrada identificada per `idEntrada` del fitxer que es troba a la variable `path`, esborrant tota la seva línia.

2.6 GestorIds

- **Nom:** GestorIds
- **Descripció:**
 - Classe que s'encarrega de carregar, guardar i mantenir actualitzades a disc les següents ids a assignar de les classes teclat, alfabet i entrada.
- **Descripció dels atributs:**
 - `gstIds`: Instància Singleton
 - `path`: String que conté la ruta on es troba el fitxer on s'emmagatzemen els tres ids
- **Descripció dels mètodes:**
 - `GestorIds()::GestorIds`:
Creadora de la classe, crea un nou fitxer que té com a nom "ids.txt" i l'inicialitza amb tres 1 separats per línies (si no existeix el fitxer); també assigna el camí del fitxer a la variable `path`.
 - `obtenirInstancia()::static GestorIds`:
Obté la instància de la classe i la crea en cas de no tenir-la.
 - `carregarIds()::Integer[]`:
Retorna una array d'enters amb els tres ids que es troben al fitxer indicat per la variable `path` en aquest ordre: [`idTeclat`, `idAlfabet`, `idEntrada`].
 - `actualitzarIds(int idTeclat, int idAlfabet, int idEntrada)::void`:
Esborra el fitxer existent i en crea un de nou amb els mateixos valors que hi havia, però sumant a cada un el valor que s'indica als paràmetres.

3. Presentació

3.1 CtrlPresentacio

- **Nom:** CtrlPresentacio
- **Descripció:**
 - Controlador responsable d'establir la comunicació entre la capa de domini i la interacció amb l'usuari i la part gràfica.
- **Descripció dels atributs:**
 - ctrlPresentacio: Instància Singleton
 - ctrlDomini: Instància Singleton del controlador de domini
- **Descripció dels mètodes:**
 - CtrlPresentacio():: CtrlPresentacio:
Constructor privat per a inicialitzar la classe.
 - obtenirInstancia()::static CtrlPresentacio:
Obté una classe única de la classe CtrlPresentacio.
 - iniPresentacio():: static void:
Fa la crida per mostrar la vista VistaLogin des d'on l'usuari podrà iniciar sessió.
 - vistaSignUp()::static void:
Fa la crida per mostrar la vista VistaSingUp des d'on l'usuari podrà registrar-se.
 - vistaPrincipal():: static void:
Fa la crida per mostrar la vista VistaPrincipal amb el menú principal de l'usuari.
 - vistaUsuari():: static void:
Fa la crida per mostrar la vista VistaUsuari, que conté el menú de gestió d'un usuari.
 - vistaTeclat()::static void:

Fa la crida per mostrar la vista vistaTeclat, que conté el menú de gestió dels teclats de l'usuari.

- vistaEntrada()::static void:

Fa la crida per mostrar la vista VistaEntrada, que conté el menú de gestió de les entrades de l'usuari.

- vistaAlfabet():: static void:

Fa la crida per mostrar la vista VistaAlfabet, que conté el menú de gestió dels alfabetes de l'usuari.

- crearUsuari(String nomUsuari, String contrasenya)::static void:

Amb els atributs passats per paràmetre, que són el nom de l'usuari i la seva contrasenya, fa una crida a CtrlDomini per comprovar els atributs, crear el perfil i posteriorment s'inicia la sessió.

- eliminarUsuari():: static void:

Fa la crida a CtrlDomini per a eliminar l'usuari.

- modificarContrasenyaUsuari(String novaContrasenya)::static void:

Amb novaContrasenya passat per paràmetre, fa una crida a CtrlDomini per a modificar la seva contrasenya.

- iniciarSessio(String nomUsuari, String contrasenya)::static void:

Amb els atributs passats per paràmetre, que són el nom de l'usuari i la seva contrasenya, fa una crida a CtrlDomini per comprovar els atributs i iniciar sessió.

- tancarSessio()::static void:

Fa la crida a CtrlDomini per tancar la sessió de l'usuari.

- crearTeclat(String nomTeclat, int idAlfabet, int idEntrada, int opcio) : :static void:

Amb el nomTeclat, l'idAlfabet, l'idEntrada i l'opció de l'entrada escollida que l'usuari proporciona, crida a CtrlDomini per a la creació d'un teclat

amb els paràmetres proporcionats.

- `eliminarTeclat(int idTeclat)::static void:`

Fa la crida a `CtrlDomini` per eliminar un teclat concret identificat per l'`idTeclat` que ha introduït l'usuari.

- `consultarTeclats()::static ArrayList<String>:`

Consulta i posteriorment mostra per pantalla els noms dels teclats associats a l'usuari.

- `consultarTeclat(int idTeclat)::static String[]:`

Fa una consulta a `CtrlDomini` del teclat amb identificador `idTeclat` que s'ha passat per paràmetre i retorna un array de chars que representa la distribució dels caràcters del teclat.

- `modificarTeclat(int idTeclat, boolean nouAlfabet, int nouId, int opció)::static void:`

Fa una crida a `CtrlDomini` amb l'id del teclat que es vol modificar, el boolean `nouAlfabet` que indica si es reassigna l'entrada o l'alfabet, el nou id de l'element reassignat i l'opció d'algoritme proporcionats per l'usuari a través de la vista `VistaTeclat` per a la modificació del teclat.

- `crearAlfabet(String nomAlfabet, String alfabet)::static void:`

Crida a `CtrlDomini` per a crear un nou alfabet amb nom `nomAlfabet` i amb el contingut `Alfabet` que l'usuari ha introduït a través de la `VistaAlfabet`.

- `eliminarAlfabet(int idAlfabet)::static void:`

Fa la crida a `CtrlDomini` per tal d'eliminar l'alfabet amb identificador `idAlfabet` que l'usuari ha proporcionat mitjançant la vista `VistaAlfabet`.

- `consultarAlfabet()::static ArrayList<String>:`

Consulta i posteriorment mostra per pantalla els noms dels alfabet associats a l'usuari.

- `consultarAlfabet(int idAlfabet)::static String[]:`

Fa una consulta a CtrlDomini de l'alfabet amb identificador idAlfabet que l'usuari ha introduït a través de la vista VistaAlfabet i retorna un array de strings que serà el contingut de l'alfabet i que serà mostrat per pantalla mitjançant la mateixa vista.

- `modificarAlfabet(int idAlfabet, String alfabet)::static void:`

Rep de la vista VistaAlfabet l'idAlfabet i el nou contingut de l'alfabet que ha proporcionat l'usuari i fa la crida a ctrlDomini per a realitzar la modificació.

- `crearEntrada(String nomEntrada, String tipus, String entrada)::static void:`

Crida a CtrlDomini per a crear una nova entrada segons el nom de l'entrada, el tipus i el seu contingut d'aquesta que l'usuari haurà introduït a través de la vista VistaEntrada.

- `eliminarEntrada(int idEntrada)::static void:`

Fa la crida a CtrlDomini per tal d'eliminar l'entrada amb identificador idEntrada que l'usuari ha proporcionat.

- `ConsultarEntrades()::static ArrayList<String>:`

Consulta i retorna els noms de les entrades associades a l'usuari, que seran mostrades posteriorment per pantalla.

- `consultarEntrada(int idEntrada)::static String[]:`

Fa una consulta a CtrlDomini de l'entrada amb identificador idEntrada que l'usuari ha introduït i retorna un array de strings que serà mostrat per pantalla.

- `modificarEntrada(int idEntrada, String entrada)::static void:`

Rep l'idEntrada i el nou contingut de l'entrada que ha proporcionat l'usuari i fa la crida a ctrlDomini per a realitzar la modificació.

3.2 VistaLogin

- **Nom:** VistaLogin
- **Descripció:**
 - Vista Inicial del programa, l'usuari ha d'iniciar sessió o pot anar a la vista VistaSignUp per registrar-se.
- **Descripció dels atributs:**
 - usernameField(JTextField): camp per introduir el nom d'usuari
 - passwordField(JPasswordField): camp per introduir la contrasenya d'usuari
 - loginButton(JButton): Botó per iniciar sessió
 - signInButton(JButton): Botó per canviar a la vista VistaSignUp per registrar-se
 - mainPanel(JPanel): Panel principal de la vista
 - ctrlPresentacio(CtrlPresentacio): Instància singleton de CtrlPresentacio
- **Descripció dels mètodes:**
 - VistaLogin()::VistaLogin:
Constructora de la vista VistaLogin

3.3 VistaSignIn

- **Nom:** VistaSignIn
- **Descripció:**
 - Vista per registrar-se. L'usuari pot canviar a la vista VistaLogin per iniciar sessió.
- **Descripció dels atributs:**
 - usernameField(JTextField): Camp per introduir i emmagatzemar el nom del nou usuari
 - passwordField(JPasswordField): Camp per introduir i emmagatzemar la contrasenya del nou usuari

- password2Field(JPasswordField): Camp per introduir i verificar la contrasenya del nou usuari
- signInButton(JButton): Botó per registrar-se
- loginButton(JButton): Botó per anar a la vista VistaLogin per iniciar sessió
- mainPanel(JPanel): Panel principal de la vista
- ctrlPresentacio(CtrlPresentacio): Instància singleton de CtrlPresentacio
- **Descripció dels mètodes:**
 - VistaSignIn():VistaSignIn:
Constructora de la vista VistaSignIn.

3.4 VistaPrincipal

- **Nom:** VistaPrincipal
- **Descripció:**
 - Vista principal del programa. Dona accés al tractament dels 4 objectes principals del programa: Usuari, Teclat, Alfabet, Entrada.
- **Descripció dels atributs:**
 - usuariButton(JButton): Botó encarregat d'obrir la vistaUsuari
 - alfabetButton(JButton): Botó encarregat d'obrir la vistaAlfabet
 - entradaButton(JButton): Botó encarregat d'obrir la vistaEntrada
 - teclatButton(JButton): Botó encarregat d'obrir la vistaTeclat
- **Descripció dels mètodes:**
 - VistaPrincipal():VistaPrincipal
Constructora de la vista VistaPrincipal.

3.5 VistaUsuari

- **Nom:** VistaUsuari
- **Descripció:**

- Vista encarregada de gestionar el tractament d'Usuari
- **Descripció dels atributs:**
 - eliminarButton(JButton): Botó per eliminar l'usuari.
 - modificarButton(JButton): Botó per obrir la perspectiva per modificar contrasenya d'usuari.
 - tancarButton(JButton): Botó per tancar la sessió
 - consultarButton(JButton): Botó per consultar la informació de l'usuari
 - enrereButton(JButton): Botó per tornar a la vista anterior
- **Descripció dels mètodes:**
 - VistaUsuari():VistaUsuari:
Constructora de la vista VistaUsuari.
 - mostrarPanelModificar(): void:
Mostra per pantalla la informació i els camps necessaris per modificar la contrasenya de l'usuari.

3.6 VistaTeclat

- **Nom:** VistaTeclat
- **Descripció:**
 - Vista encarregada de gestionar el tractament de les funcionalitats de teclat.
- **Descripció dels atributs:**
 - messageLabel1(JLabel): Missatge de text per les funcionalitats del teclat
 - consultarButton(JButton): Botó per consultar un teclat
 - crearTeclatButton(JButton): Botó per obrir la perspectiva per crear un teclat
 - teclatTable(JTable): Taula encarregada de mostrar tots els teclats de l'usuari
 - eliminarButton(JButton): Botó per obrir la perspectiva per eliminar un

teclat

- modificarButton(JButton): Botó per obrir la perspectiva per modificar un teclat
- enrereButton(JButton): Botó per tornar a la vista anterior
- idsTeclats(HashMap<String, Integer>): Map que relaciona el nom d'un alfabet amb el seu identificador.

- **Descripció dels mètodes:**

- VistaTeclat():VistaTeclat:
Constructora de la vista VistaTeclat
- mostrarPanelCrearTeclat(): void:
Mostra el panel per recollir les dades i crear un teclat.
- mostrarPanelModificarTeclat(): void:
Mostra el panel per recollir les dades i modificar un teclat.
- mostrarModEntrada(): void:
Mostra per pantalla un missatge d'avis.
- carregarDades(): void:
Carrega els teclats que pertanyen a l'usuari.
- actualitzarTable(): void:
Actualitza teclatTable amb els teclats de l'usuari.

3.7 VistaAlfabet

- **Nom:** VistaAlfabet

- **Descripció:**

- Vista encarregada de gestionar el tractament de les funcionalitats d'alfabet

- **Descripció dels atributs:**

- consultarButton(JButton): Botó per consultar un alfabet

- messageLabel1(JLabel): Missatge de text per les funcionalitats d'alfabet
- crearAlfabetButton(JButton): Botó per obrir la perspectiva per crear un alfabet
- alfabetTable(JTable): Taula que mostra els alfabets de l'usuari
- eliminarButton(JButton): Botó per obrir la perspectiva per eliminar un alfabet
- modificarButton(JButton): Botó per obrir la perspectiva per modificar un alfabet
- enrereButton(JButton): Botó per tornar a la vista anterior
- idsAlfabets(HashMap<String, Integer>): Map que relaciona el nom d'un alfabet amb el seu identificador.
- **Descripció dels mètodes:**
 - VistaAlfabet()::VistaAlfabet
Constructora de la vista VistaAlfabet.
 - mostrarPanelCrear():: void:
Mostra el panel per recollir les dades i crear un alfabet.
 - mostrarPanelModificar():: void:
Mostra el panel per recollir les dades i modificar un alfabet.
 - carregarDades():: void:
Carrega els alfabets que pertanyen a l'usuari.
 - actualitzarTable():: void:
Actualitza alfabetTable amb els alfabets de l'usuari.

3.8 VistaEntrada

- **Nom:** VistaEntrada
- **Descripció:**
 - Vista encarregada de gestionar el tractament de les funcionalitats d'Entrada.

- **Descripció dels atributs:**

- consultarButton(JButton): Botó per consultar una entrada
- messageLabel1(JLabel): Missatge de text per les funcionalitats d'alfabet
- entradaTable(JTable): Taula que mostra les entrades de l'usuari
- crearButton(JButton): Botó per crear una entrada
- eliminarButton(JButton): Botó per obrir la perspectiva per eliminar una entrada
- modificarButton(JButton): Botó per obrir la perspectiva per modificar una entrada
- enrereButton(JButton): Botó per tornar a la vista anterior
- idsEntrades(HashMap<String, Integer>): Map que relaciona el nom d'un alfabet amb el seu identificador.

- **Descripció dels mètodes:**

- VistaEntrada()::VistaEntrada
Constructora de la vista VistaEntrada.
- mostrarPanelCrear():: void:
Mostra el panel per recollir les dades i crea una entrada.
- mostrarPanelModificar():: void:
Mostra el panel per recollir les dades i modificar una entrada.
- carregarDades():: void:
Carrega les entrades que pertanyen a l'usuari.
- actualitzarTable():: void:
Actualitzar entradaTable amb les entrades de l'usuari.