KEY - LAYOUT GENERATOR DOCUMENTACIÓ OFICIAL





Projectes de programació. QT-2023/24 Facultat d'informàtica de Barcelona Universitat Politècnica de Catalunya

Josep Díaz Sosa | josep.diaz.sosa
Pau Mayench Caro | pau.mayench
Marc Expósito Francisco | marc.exposito.francisco
Víctor Hernández Barragán | victor.hernandez.barragan

CONTINGUTS

Descripció d'atributs i mètodes de la capa de domini Descripció de les classes de la capa de presentació Descripció de les classes de la capa de persistència	<i>3</i>

DESCRIPCIÓ D'ATRIBUTS I MÈTODES DE LA CAPA DE DOMINI

CLASSE < Usuari>

Aquesta classe fa referència als usuaris que faran servir el programa. Aquests usuaris poden ser Admin o usuaris corrents. Els usuaris tenen un nom d'usuari, una contrasenya, una mà prioritaria (Esquerra o Dreta) i un identificador per saber si es Admin o no. Els primers 2 atributs son Strings, mentres que el 3r atribut es un Enum i l'últim és un booleà.

Les operacions són principalment *setters* i *getters*, amb una associació amb la clase Teclat. Destaquen les operacions:

- isTeclat(id: Integer) :: boolean
 - Retorna true si el teclat amb ID id existeix dins de l'associació.
- addTeclat(id: Integer, teclat : Teclat)
 - Afegeix un teclat a l'associació amb id.
- eliminarTeclats(id : Integer)
 - o Elimina el teclat a l'associació amb id.
- getTeclats():: HashMap<Integer, Teclat>
 - Retorna el HashMap amb els teclats de l'usuari, junt amb el id corresponent

CLASSE < Teclat>

Aquesta classe representa un teclat, amb una string com a nom, el seu *layout* (expressat com una matriu de chars), el algorisme amb el que s'ha creat el teclat (instancia de la classe Algorisme) i la data de creació d'aquesta classe (expressat amb la classe Date). Cal destacar que el teclat no guarda el usuari, ja que el propi usuari, guarda els teclats en el seu Hashmap.

Les operacions de aquesta classe són principalment *setters* i *getters*. Algunes de les operacions destacables són:

- addFrequencies(Frequencies frequencies) ::
 - Associa la instancia de freqüència que es passa com a paràmetre a la instancia de teclat
- getInfo() :: Vector<String>
 - Retorna un vector amb la informació del teclat (com a diferents strings).
 En ordre del vector apareixen:
 - El nom del teclat
 - El nom del algoritme amb el que s'ha creat
 - La data de creació del teclat
 - El nom del alfabet de del input
 - I per últim cada fila de la matriu del layout

CLASSE <Text> (+ subclases)

La classe text representa tots els textos que el sistema permet guardar. Com cada tipus de text té les seves pròpies funcionalitats, s'han classificat en dues subclasses: TextPúblic i TextPredefinit. Els textos públics són aquells que els usuaris poden penjar al sistema per a que altres usuaris els agafin com a input per als seus teclats. Per altre banda, els textos predefinits, son aquells que existeixen el el sistema de forma "predefinida". Només els usuaris determinats com a administradors poden gestionar aquestes instancies. Aquests textos ens permeten donar als usuaris que no tenen cap

fitxer específic alguns exemples generals que poden arribar a usar com a input dels seus teclats.

Les funcionalitats principals s'implementen en les subclasses, ja que en la classe Text només trobem *setters* i *getters*.

En la classe TextPublic podem destacar la funcionalitat:

- getUsuariUsername():: String
 - Aquesta funció retorna el username que ha penjat el text públic

CLASSE <Frequencia>

La classe Freqüències és vital per al sistema. En ella es guarden i es processen les freqüències dels textos o els inputs que pot realitzar l'usuari per crear el seu teclat. Utilitza un TreeMap per guardar la llista de frequencies, es guarda el Alfabet al que pertany i també té un paràmetre on guarda el nombre de paraules que es troben (si es tracta d'un text d'on cal treure les freqüències). Aquesta classe és l'encarregada de processar els inputs per a la creació dels teclats, i per tant té diverses funcions per als diferents tipus de inputs que pot rebre. Algunes de les principals són:

- genera(String s, Alfabet[] llistaAlfabets) ::
 - és la funció encarregada de llegir un text (que rep en la variable string s).
 La funció té uns valors definits "prohibits", que la funció comprova en el text i si els detecta no els conta. La funció doncs, llegeix les paraules del text introduït i mentres detecta l'alfabet del propi text. Un cop acaba de processar tot el text, genera les freqüències de les paraules detectades
- llegir(String t, Alfabet[] llistaAlfabets) ::
 - és la funció encarregada de llegir l'input quan aquest són uns parells de paraules amb les seves freqüències. Aquesta funció processa el format amb el que s'ha introduït. També detecta l'alfabet del input, que es selecciona de l'array de Alfabets que té com a paràmetre

CLASSE < Alfabet>

Aquesta classe representa tots els alfabets que es permeten en el sistema. L'usuari el tria de forma indirecte en el moment d'introduir el input per a la generació del teclat. Aquesta classe té com a atributs el nom de l'alfabet, i el alfabet en si (és a dir, els caràcters que conformen l'alfabet). La majoria de les funcionalitats són setters i getters, a excepció de la funcionalitat principal, que es la que s'utilitza per detectar el alfabet de la col·lecció d'alfabets. La funció és la següent:

- determinaAlfabet(Character c, TreeMap<String, String> magatzemAlfabets) ::
 - Aquesta funció és la responsable de anar revisant els caràcters que rep i els compara amb els dels alfabets guardats en el sistema. Si detecta un dels alfabets, l'assigna.

CLASSE < Algorisme >

La classe algorisme representa els possibles algorismes de creació de teclats que podria arribar a usar un Usuari. Aquesta classe és una classe abstracte, ja que no té cap instància i les seves funcions no tenen cap implementació. La implementació es realitza en les seves jerarquies (que son els dos algorismes implementats en el sistema). La única funció interessant mencionable és:

- generarLayout(Frequencies frequencies, int columnes, int files) :: char[[[]
 - La implementació d'aquesta funció es troba en les seves subclasses, ja que cada una usa una estrategia de generació diferent

CLASSE <BranchAndBoundEager> (extends Algorisme)

La subclasse de algorisme implementada en aquesta primera entrega. Es l'encarregada de generar el layout que se li proporciona al teclat un cop s'ha realitzat l'input complert. Els dos atributs principals i destacables són la matriu de distàncies i la de proximitats entre caràcters (aquestes proximitats es basen en les freqüències que es calculen en la classe freqüència). Aquestes matrius són matrius de integers, característiques del problema QAP que tractem de solucionar amb aquest algorisme amb l'estratègia Branch and Bound (per més informació sobre la resolució del algorisme, veure el document Estructures de dades i Algorismes). A part dels setters i getters, la funcionalitat que destaca és la següent:

- generarLayout(Frequencies frequencies, int columnes, int files) :: char[[[]
 - Aquesta és la implementació de la funció mencionada en l'explicació de la classe Algorisme. És l'encarregada de generar el layout que li pertoca a partir de les freqüències mitjançant l'estratègia Branch and Bound (explicada en el document d'algorisme). La funció retorna directament la matriu de chars que pertoca al teclat amb totes les lletres col·locades segons la solució de l'algorisme.

CLASSE <SimulatedAnnieling> (extends Algorisme)

La subclasse de algorisme implementada en la segona entrega. Igual que la classe anterior, aquesta classes es una segona opció per la generació del layout que se i proporciona al teclat. A part de les matrius distancia i proximitat com la classe anterior, els altres atributs principals son els paràmetres per a que l'algorisme funcioni: la temperatura, el coolingRate i les interacions. Aquest és un algorisme de búsqueda metaheurística anomenada Simulated Annealing, un algorisme característic en la Intel·ligència Artificial, que utilitzem per resoldre el problema QAP (més informació al document d'Estructures de dades i Algorismes). A més, la funcionalitat que destaca, com l'anterior classe és:

generarLayout(Frequencies frequencies, int columnes, int files) :: char[][]

 Aquesta és la implementació que s'encarrega de generar el layout amb les freqüències mitjançant l'estratègia de búsqueda metaheurística mencionada anteriorment. La funció retorna una matriu de chars on cada caràcter representa el teclat format per l'algorisme.

DESCRIPCIÓ DE LES CLASSES DE LA CAPA DE PRESENTACIÓ

En aquest apartat es descriu breument la funcionalitat que aporta al programa cadascuna de les classes conformants de la capa de presentació, fent un repàs a les interaccions possibles a cadascuna mitjançant els seus widgets.

CLASSE < VistaPantallaIni>

L'únic mètode que conté, *mostra*, implementa la primera finestra que apareix al executar el programa, la qual mostra un label indicatiu dels autors, títol i dos botons: Un per a registrar-se al sistema mitjançant "Crea un compte", que farà que el sistema, a través del controlador salti al codi de **VistaRegistre**> i un botó "Iniciar Sessió", que serveix per a fer login al sistema en cas que l'usuari tingui ja un compte i envia a través del controlador a **VistaLogin**>.

CLASSE < VistaRegistre>

L'únic mètode que conté, *mostra*, implementa la finestra per a la qual un usuari pot donar-se d'alta al sistema a través d'emplenar els textField corresponents a la introducció del nom d'usari i contrassenya, així com un combo box de selecció de la mà bona de l'usuari, ja que de cara a la impressió del teclat això serà important de cara a quines tecles estaran més a prop del polze de l'usuari. El botó "Confirmar Registre" envia a la pantalla principal de **VistaMenuPrincipal**> i fa que el controlador de presentació cridi a la funció del controlador del domini que enregistra l'usuari al sistema amb les dades proporcionades.

CLASSE < VistaLogin>

Implementa la finestra per a fer login amb un perfil existent. Anàlogament a **<VistaRegistre>** inclou dos textField per a que l'usuari introdueixi username i contrasenya. El botó "Confirmar Login" farà que el controlador de presentació cridi a l'operació del controlador del domini que carrega l'usuari al sistema i el fagi servir per a totes les operacions futures. Envia a la finestra carregada per **<VistaMenuPrincipal>**.

CLASSE < VistaMenuPrincipal >

Carrega la finestra que representa el vestíbul principal del programa, ja que implementa els 4 botons corresponents a les funcionalitats disponibles al programa: Crear un teclat, Gestionar els teclats que l'usuari ja ha creat, Publicar un text públic al sistema i Gestionar els texts públics al sistema. En cas de tractar-se de l'usuari *Admin*, aquest veurà opcions addicionals que un usuari corrent no veuria: Gestionar texts predefinits i Gestionar Usuaris del sistema.

Conté, a més, un desplegable combo box que permet a l'usuari gestionar el seu perfil o bé fer logout.

CLASSE < VistaCreacioTeclat>

Carrega la finestra de configuració dels paràmetres personalitzables per a la creació d'una distribució de teclat. Primerament, implementa un desplegable *Combo Box* que permet seleccionar amb quin algorisme vol l'usuari que es generi la distribució: es llegeix del sistema quins hi han disponibles i es van introduïnt al desplegable com a opcions de forma dinàmica, de manera que si el sistema és modificat, automàticament sortiria l'opció com a disponible.

Addicionalment, implementa 5 botons que disparen el codi per a cadascuna de les 5 opcions que té l'usuari d'introduir les freqüències per a l'algorisme: *Entrar Freqüències Manualment, Entrar fitxer amb freqüències, Crear a partir d'un fitxer de text, Usar text públic i Usar text predefinit.*

CLASSE < VistaCreacioFreqManual >

Implementa la lògica a través de la qual l'usuari pot introduir manualment des de l'aplicació les freqüències de cada paraula a través d'un textArea, així com també un textField per a donar-li un nom que l'identifiqui al sistema. Un com premut el botó "Confirmar", executa l'operació del controlador de domini per a generar el teclat amb les opcions que ha anat escollint l'usuari com a paràmetres.

CLASSE < VistaCreacioFitxer>

Implementa la lògica necessària per a que l'usuari pugui introduïr a l'algorisme la informació necessària per a obtenir les freqüències a través de proporcionar un fitxer de text, de forma que aquesta vista és la mateixa tant per a l'opció *Entrar fitxer amb freqüències* com per a *Crear a partir d'un fitxer de text* (que sigui un fitxer de freqünències o un text en llenguatge humà es controla mitjançant un paràmetre depenent del botó premut). Disposa d'un botó que accedeix al sistema de fitxers que permet a l'usuari seleccionar el fitxer de text pla que decideixi fer servir per a l'algorisme, depenent del cas haurà d'escollir un fitxer de freqüències o de llenguatge humà. Conté el textField per a introduir el nom del teclat i el botó "Confirmar" executa l'operació per a generar el teclat amb l'algorisme seleccionat prèviament.

CLASSE < VistaCreacioTextos >

Implementa la lògica necessària per a que l'usuari pugui crear un teclat fent servir, o bé un text predefinit al sistema (per l'admin), o bé un text públic publicat per un altre usuari. Depenent si s'ha seleccionat *Usar text públic* o *Usar text predefinit*. La finestra que carrega aquesta classe mostra un quadre amb barra per a scrolling i els títols de tots els textos (predefinits o públics), dels quals l'usuari haurà de fer clic a un i introduir el nom per al teclat al textField. Al prémer "Confirmar", s'executa l'operació per a generar el teclat.

CLASSE < VistaGestionarTeclats >

Implementa la finestra que apareix al seleccionar *Gestionar els teclats*. Apareix un desplegable per cada teclat que té l'usuari emmagatzemat i li dóna 3 opcions a executar al teclat al prémer-lo: *Veure Teclat, Modificar Teclat, Eliminar Teclat*.

CLASSE < Vista Veure Teclat>

Implementa la finestra que apareix al seleccionar *Veure Teclat* del desplegable, mostra el nom del teclat, l'algorisme amb què va ser creat, l'alfabet dels caràcters que el conformen, la data de creació i, evidentment, la distribució de les tecles del teclat en forma de botons.

CLASSE < Vista Eliminar Teclat >

Implementa la finestra que apareix al seleccionar *Eliminar Teclat*, senzillament apareix un botó que l'usuari ha de prémer per a eliminar el teclat. Això es fa a mode de seguretat, per si l'usuari ha premut *Eliminar Teclat* per accident.

CLASSE < VistaModificarTeclat>

Implementa la finestra que apareix al seleccionar *ModificarTeclat*. Conté 3 botons: *Canviar nom del teclat, Canviar input del teclat, Canviar estratègia del teclat*. Depenent de quin sigui seleccionat, la pròpia classe farà saltar una finestra pop-up demanant a l'usuari els canvis pertinents relatius al nom/input (freqüències i, per tant, distribució)/algorisme, depenent quina selecció hagi fet. (Hi han 3 finestres pop-up, una per cadascuna de les opcions).

CLASSE < VistaGestioUsuari>

D'una forma similar a **VistaModificarTeclat**>, aquesta classe genera una finestra que mostra 3 opcions en forma de botons a l'usuari: *Canviar Contrasenya, Canviar la meva mà bona* i *ELIMINAR USUARI*. La pròpia classe s'encarrega de generar unes finestres Pop-Up que no destrueixen l'anterior finestra amb les opcions pertinents a cadascuna de les opcions.

CLASSE < VistaMenuGestioTextosPublics>

Aquesta vista mostra tots els títols dels textos que ha publicat l'usuari. Aquests són clicables. També es mostren dos botons, les accions dels quals les genera la pròpia classe en finestres Pop-Up que no destrueixen la principal de la classe. Aquests dos botons són: *Veure Text Públic* (el contingut d'aquest) i *Eliminar Text Públic*. Eliminar un text públic NO elimina els teclats del sistema creats mitjançant aquest text.

CLASSE < VistaMenuGestioTextosPredefinits >

Classe equivalent a < *VistaMenuGestioTextosPublics*>, però només accessible per l'usuari administrador al seleccionar al menú principal l'opció de gestionar textos predefinits.

CLASSE < VistaGestionar Usuaris >

Classe només accessible per a l'usuari *Admin*, aquesta classe implementa la finestra que mostra el username de tots els usuaris. Disposa d'un un botó que permet eliminar a l'usuari seleccionat i tots els textos que va publicar, així com un botó que permet promocionar al rang d'*Admin* a un usuari del sistema.

CLASSE < VistaPublicarTextPublic>

Classe que implementa la finestra a partir de la qual un usuari pot publicar un text al sistema per tal que un altre usuari pugui fer-lo servir per a crear el seu teclat (ja que potser té un interès similar per a la fi del teclat). Incorpora un selector de fitxers juntament amb un textField que haurà d'emplenar amb el títol que vol donar al mateix al sistema.

DESCRIPCIÓ DE LES CLASSES DE LA CAPA DE PERSISTÈNCIA

CLASSE < GestorTexts>

El GestorTexts s'encarrega de la gestió de textos dins de l'aplicació. Aquest gestor té la responsabilitat de guardar a disc els textos públics carregats pels usuaris, així com els textos predefinits que els administradors puguin voler establir. Funciona mitjançant dos arxius JSON separats: un per als textos públics i un altre per als textos predefinits. En cas que els fitxers JSON no existeixin o siguin eliminats, el GestorTexts generarà automàticament tres textos predefinits per defecte com a punt de partida.

A més, cada text públic es vincula amb el nom d'usuari de l'usuari que el va pujar, permetent una associació clara entre l'usuari i els seus textos. Les funcionalitats clau d'aquesta classe inclouen la capacitat de llegir, escriure, modificar i eliminar textos tant públics com predefinits.

CLASSE < GestorUsuaris>

El GestorUsuaris s'encarrega de la gestió dels usuaris de l'aplicació. Aquest gestor realitza operacions com la creació, lectura, actualització i eliminació de dades d'usuari, emmagatzemades en fitxers JSON. Inclou funcions per canviar contrasenyes, ajustar configuracions d'usuari (com la mà bona), i administrar els drets d'administrador. També maneja la creació i modificació de teclats personalitzats associats a cada usuari. En cas que els fitxers no existeixin, s'inicialitzen amb un usuari admin per defecte.

Cada usuari es guardat en un fitxer JSON individual amb un índex assignat arbitràriament, aquesta relació username índex es guarda en un altre JSON anomenat usuarisIndex on hi ha una entrada per cada usuari amb l'username i l'índex. D'aquesta manera quan un usuari està loguejat i ha de modificar els seus teclats o crear-ne de nous, no fa falta llegir i reescriure tots els teclats i usuaris de l'aplicació.

CLASSE < GestorAlfabets>

El GestorAlfabets és responsable de gestionar diferents alfabets emmagatzemats en l'aplicació. Aquesta classe permet la lectura i l'escriptura d'alfabets en un fitxer JSON, proporcionant un accés centralitzat als alfabets utilitzats dins de l'aplicació. Inclou una funció per obtenir un llistat de tots els alfabets disponibles, que es guarden en un fitxer JSON anomenat alfabets.json

CLASSE < GestorAlgorismes>

El GestorAlgorismes s'encarrega de la gestió dels diferents algorismes emprats en l'aplicació. Aquesta classe maneja la persistència d'algorismes utilitzats per a l'optimització dels dissenys de teclat. D'aquesta manera podem obtenir una llista de tots els algorismes disponibles. Es guarden els noms, "tipus de l'algorisme", en un fitxer anomenat algorismes.json