Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Parte 1)

Diagrama

Descripción generada automáticamente

La estructura esta compuesta por la etiqueta del carácter actual donde esta posicionado, en el caso que sea una palabra, se le agrega el numero de pagina en que se encuentra la palabra. A su vez contiene un enlace al siguiente carácter en caso de que no sea fin de palabra, con un array de caracteres que indica como se van componiendo las palabras.

Parte 2)

Dado un texto (palabra y paginas separadas por comas) debemos recorrer carácter por carácter de este hasta encontrar una coma (“,”), entonces se genera una nueva palabra desde el índice 0 hasta el valor donde se encontró la primer coma. Y el resto del texto se pasa como un String y se le hace una separación en array a cada valor separado por coma.

Entonces se inserta de manera común la palabra encontrada en el trie, pero antes de marcarlo como palabra a ese nodo donde termina la palabra, se le agrega el valor de las paginas donde esta la palabra que había sido pasado por parámetro.

Ej. ala, 1, 13, 22 🡪 Palabra: ala y int[] paginas: [1, 13, 22]

Precondiciones.

El texto debe tener los valores establecidos en para el alfabeto declarado, debe tener una como enseguida de la palabra y los valores numéricos deben ser separados por comas.

Inicialmente el trie debe estar vacío.

Postcondiciones.

El trie debe tener almacenadas las palabras dadas con sus números de páginas correspondientes.

El trie debe quedar correctamente formado.

Pseudo:

Void InsertarTextoPorLinea( texto)

Com

Entero i 🡨0

MIENTRAS texto.caracter(i) <> “,” HACER

i++,

FINMIENTRAS

Palabra 🡨 texto.subtexto(0 hasta i)

TNodoTrie nodo 🡨 este nodo

PARA CADA carácter en palabra HACER

Entero índice 🡨 valor de carácter

SI nodo.hijos[índice] = nulo ENTONCES

Nodo.hijos[índice] 🡨 nuevo TNodoTrie()

FIN SI

Nodo = nodo.hijos[índice]

FIN PARA

Nodo.paginas 🡨 obtenerPag(texto.subtexto(i + 2)

Nodo.esPalabra 🡨 true

FIN

obtenerPag(texto) devuelve int[ ] paginas

Com

String[ ] str 🡨 texto.separar( “, “ )

Int[ ] pag 🡨 new int [str.largo]

PARA CADA valor en str HACER

Pag[i] 🡨 (entero) valor

i++

FIN PARA

DEVUELVE pag

FIN

Parte 3)

Diagrama, Esquemático

Descripción generada automáticamente

a) programa 7 veces

b) proselitismo 4

c) cazadores 5

d) Altura 11

e) tamaño 45