

## Explicación de la Práctica 4, versión 2.

Para hacer la versión 2 de la práctica, se recomienda hacer la siguiente función main, o alguna versión de ésta.

```
int main()
{
    tIndicePalabras idx = nuevoIndice();
    tListaCadenas totales = nuevaListaCadenas();
    tMatriz L = inicMatriz(1, 0.0); //matriz de tam 1 y valor 0.0
    string nombreArchivo = pedirArchivo();
    cout << "Cargando..." << endl;
    crearTabla(nombreArchivo, idx, totales, L);
    cout << "Tabla indice creada." << endl;

1    cout << "matriz L: " << endl << L << endl;
2    cout << idx;
3    cout << "totales: " << totales << endl;

    tMatriz M = desdeEnlacesAMatriz(L);

4    cout << "C: " << endl << C << endl;
5    cout << "matriz M: " << endl << M << endl;

    tVector v = vectorPropio(M, 0.000001);

6    cout << "Vector v: " << endl << v << endl;

    string palabra = pedirPalabra();
    while (palabra != "fin")
    {
        tListaCadenas archivos = getValor(idx, palabra);

7        cout << "archivos: " << archivos << endl;

        tVector valores = valoresSubLista(archivos, totales, v);

8        cout << "valores: " << valores << endl;

        ordenaValores(archivos, valores);

9        cout << "2 archivos: " << archivos << endl;
10       cout << "2 valores: " << valores << endl;

        muestraArchivos(archivos, valores);
        palabra = pedirPalabra();
    }
} // main
```

El código que se pide en la práctica es el que no tiene a su izquierda un número, las líneas con número corresponden a couts que sirven para guiar la explicación de mañana viernes 27 de abril. La ejecución de este main envía los siguientes mensajes:

Dame el nombre del archivo raíz: indice.txt

Cargando...

Tabla indice creada.

<pre>1 cout &lt;&lt; "matriz L: "...</pre>	matriz L: 0, 0, 0, 0 1, 0, 1, 1 1, 0, 0, 1 1, 0, 0, 0
<pre>2 cout &lt;&lt; idx;</pre>	Esta tabla de indices tiene un tamnno: 15 Clave   Indice archivo: [indice.txt,file1.txt,file2.txt,file3.txt] dos: [file3.txt] el: [indice.txt] enlace: [file2.txt] enlaces: [indice.txt,file1.txt,file3.txt] es: [indice.txt] este: [file1.txt,file2.txt,file3.txt] esto: [indice.txt] no: [file1.txt] prueba: [indice.txt] raiz: [indice.txt] tiene: [indice.txt,file1.txt,file2.txt,file3.txt] tres: [indice.txt] un: [file2.txt] una: [indice.txt]
<pre>3 cout &lt;&lt; "totales: "...</pre>	totales: [indice.txt,file1.txt,file2.txt,file3.txt]
<pre>4 cout &lt;&lt; "C: "</pre>	0,85
<pre>5 cout &lt;&lt; "matriz M: "</pre>	matriz M: 0.0375, 0.25, 0.0375, 0.0375 0.320833, 0.25, 0.8875, 0.4625 0.320833, 0.25, 0.0375, 0.4625 0.320833, 0.25, 0.0375, 0.0375
<pre>6 cout &lt;&lt; "Vector v: "</pre>	Vector v: 1 3.38319 1.82875 1.28333

Que palabra quieres buscar? raiz

<pre>7 cout &lt;&lt; "archivos: "</pre>	archivos: [indice.txt]
<pre>8 cout &lt;&lt; "valores: "</pre>	valores: 1
<pre>9 cout &lt;&lt; "2 archivos: "</pre>	2 archivos: [indice.txt]
<pre>10 cout &lt;&lt; "2 valores: "</pre>	2 valores: 1

La lista de archivos donde aparece la palabra es:

indice.txt (1)

Que palabra quieres buscar? no

<pre>7 cout &lt;&lt; "archivos: "</pre>	archivos: [file1.txt]
<pre>8 cout &lt;&lt; "valores: "</pre>	valores: 3.38319
<pre>9 cout &lt;&lt; "2 archivos: "</pre>	2 archivos: [file1.txt]
<pre>10 cout &lt;&lt; "2 valores: "</pre>	2 valores: 3.38319

La lista de archivos donde aparece la palabra es:

file1.txt (3.38319)

Que palabra quieres buscar? un

7 cout << "archivos: "	archivos: [file2.txt]
8 cout << "valores: "	valores: 1.82875
9 cout << "2 archivos: "	2 archivos: [file2.txt]
10 cout << "2 valores: "	2 valores: 1.82875

La lista de archivos donde aparece la palabra es:

file2.txt (1.82875)

Que palabra quieres buscar? dos

7 cout << "archivos: "	archivos: [file3.txt]
8 cout << "valores: "	valores: 1.28333
9 cout << "2 archivos: "	2 archivos: [file3.txt]
10 cout << "2 valores: "	2 valores: 1.28333

La lista de archivos donde aparece la palabra es:

file3.txt (1.28333)

Que palabra quieres buscar? fin

---