Fecha: 22/02/2025

Informe Taller02

Autores: Santiago Mesa y Jeronimo Chaparro Tenorio.
Materia: Estructura de Datos.
Pontificia Universidad Javeriana

1. Introducción

Este informe presenta el diseño y la relación entre dos tipos abstractos de datos (TAD): **ArchivoTexto** y **ColaP**, implementados en lenguaje C++. Ambos TADs colaboran en la lectura de un archivo de texto, el procesamiento de sus líneas y la manipulación de palabras contenidas en el mismo.

2. Diseño del TAD "ArchivoTexto"

Definición: Este TAD representa la estructura de un archivo de texto y sus elementos principales, incluyendo el nombre del archivo, una subcadena a buscar y las líneas contenidas en el archivo.

Atributos:

- `nombreArchivo` (string): Nombre del archivo de texto a leer.
- `subcadena` (string): Subcadena a buscar en las palabras del archivo.
- `listaLineas` (list): Lista de líneas del archivo.

Métodos:

- `~ArchivoTexto()`: Destructor que limpia la lista de líneas.
- `void setNombreArchivo(string nombreArchivo)`: Asigna el nombre del archivo.
- `string getNombreArchivo()`: Devuelve el nombre del archivo.
- `void setSubcadena(string subcadena)`: Asigna la subcadena.
- `string getSubcadena()`: Devuelve la subcadena.
- `void setListaLineas(list<string> listaLineas)`: Asigna la lista de líneas.
- `list<string> getListaLineas()`: Devuelve la lista de líneas.
- `void leerArchivo()`: Lee el archivo, guarda las líneas en `listaLineas` y almacena la subcadena.

3. Diseño del TAD "ColaP"

Definición: Este TAD administra una cola de palabras con su correspondiente número de línea en el archivo de texto y se encarga de imprimir las palabras que contienen la subcadena tanto en su forma original como invertida.

Atributos:

- `colaPalabras` (queue<pair<string, int>>): Cola de palabras y el número de línea en la que aparecen.
- `archivoTexto` (ArchivoTexto): Objeto del TAD `ArchivoTexto` que contiene el archivo leído.
- `invSubcadena` (string): Subcadena invertida.

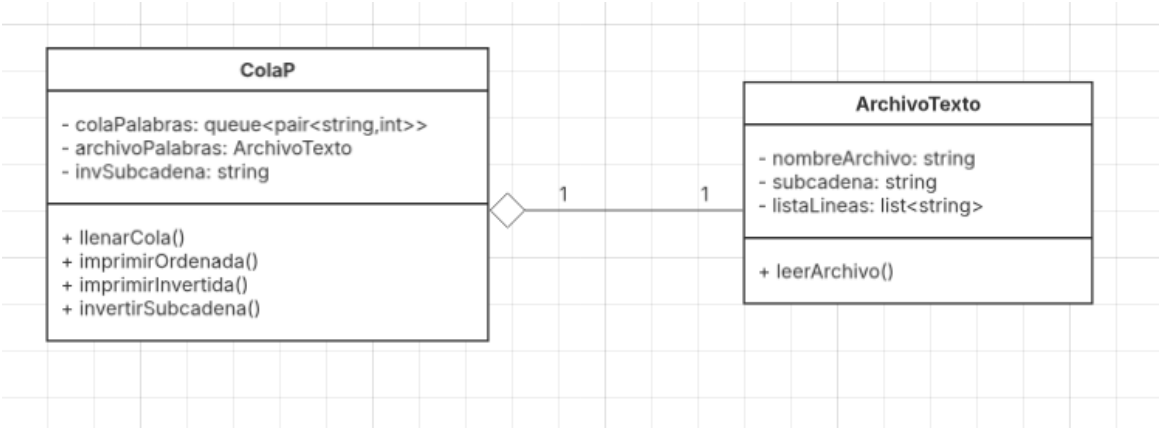
Métodos:

- `~ColaP()`: Destructor que limpia la cola de palabras.
- `void setColaPalabras(queue<pair<string, int>> colaPalabras)`: Asigna la cola de palabras.
- `queue<pair<string, int>> getColaParabras()`: Devuelve la cola de palabras.
- `void setArchivoTexto(ArchivoTexto archivoTexto)`: Asigna el objeto `archivoTexto`.
- `ArchivoTexto getArchivoTexto()`: Devuelve el objeto `archivoTexto`.
- `void setInvSubcadena(string invSubcadena)`: Asigna la subcadena invertida.
- `string getInvSubcadena()`: Devuelve la subcadena invertida.
- `void llenarCola()`: Llena la cola con las palabras del archivo y el número de línea.
- `void imprimirOrdenada()`: Imprime las palabras que contienen la subcadena en el orden original.
- `void imprimirInvertida()`: Imprime las palabras que contienen la subcadena invertida.
- `void invertirSubcadena()`: Invierte la subcadena y la almacena en `invSubcadena`.

4. Relación entre los TADs

El TAD `ColaP` contiene un objeto de tipo `ArchivoTexto`, lo que establece una relación de **composición**. A través de esta relación, `ColaP` accede a las líneas y la subcadena del archivo leído para realizar operaciones de búsqueda y manipulación de palabras.

Diagrama de Relación:



Nota: En el Diagrama de Relación no se pusieron los destructores, getters y setters de cada TAD.

5. Comandos para correr el código y plan de pruebas

Procedimiento para ejecutar las pruebas

1. **Compilar el programa**:
2. g++ -std=c++17 Taller02.cpp Clases.cpp -o programa
3. **Ejecutar el programa con diferentes archivos de entrada**:
4. ./programa entradas/ entrada1.txt
5. ./programa entradas/ entrada2.txt
6. ./programa entradas/ entrada3.txt

Tabla de Casos de Prueba

Caso de Prueba	Valores de Entrada	Resultado Esperado	Resultado Obtenido	Estado (Éxito/Falla)
Prueba 1: Archivo con palabras y subcadena presente	entrada1.txt (subcadena "co")	Lista de palabras que contienen "co" y "oc" con sus líneas	Lineas y las palabras que dicen “co”	☑ Éxito
Prueba 2: Archivo sin palabras que coincidan con la subcadena	entrada2.txt (subcadena "rpp") SE HACE MODIFICACION EN EL ARCHIVO PARA QUE NO APAREZCA LA PALABRA	Mensaje: "Número de palabras que contienen la subcadena: 0"	Numero de palabras que contienen la subcadena ordenada/invertida en el archivo: 0	☑ Éxito
Prueba 3: Archivo vacío	entrada_vacio.txt	Mensaje de error o "Número de palabras: 0"	Numero de palabras que contienen la subcadena	☑ Éxito

			ordenada/invertida en el archivo: 0	
Prueba 4: Archivo inexistente	entrada_inexistente.txt	Mensaje de error: "Error al abrir el archivo"	Error al abrir el archivo.	☑ Éxito
Prueba 5: Archivo con una sola palabra	entrada3.txt (subcadena "el")	Imprime la palabra y su línea	Con la Subcadena |el| ordenada: vuelve Linea: 14 pelo Linea: 15 el Linea: 22 Numero de palabras que contienen la subcadena ordenada en el archivo: 3	☑ Éxito

Conclusión

Este plan de pruebas proporciona una validación completa del programa, asegurando que maneje distintos escenarios correctamente. Se recomienda repetir las pruebas tras cualquier modificación en el código para mantener su fiabilidad.