

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.

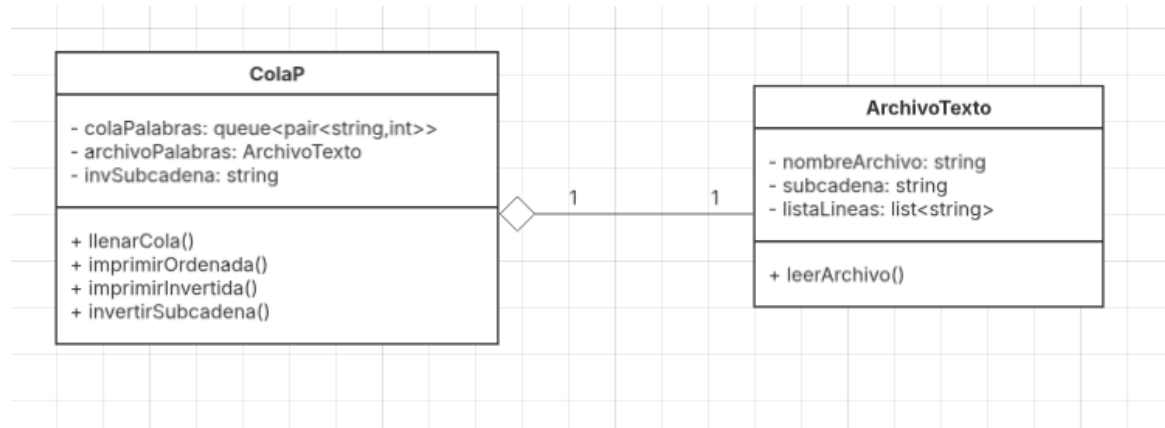
### Métodos:

- `~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

En una terminal poner :

```
g++ Clases.cpp Taller02.cpp -o nombre_ejecutable
```

Luego para correr el ejecutable:

```
./nombre_ejecutable nombre_entrada.txt
```

En caso de que las entradas esten en dentro de otra carpeta seria:

```
./nombre_ejecutable nombre_carpeta/nombre_entrada.txt
```