**Proyecto#3 - El buscador**

**Algoritmos y Estructuras de Datos II**

**Fecha Entrega:** 11 de Mayo 2025

**Trabajo Individual**

**Objetivo**

Practicar con diccionarios y colas de prioridad.

**Introducción**

Supongamos que bajamos algunos libros en formato de texto de Internet (<http://www.gutenberg.org/browse/languages/es>) , por ejemplo: <http://www.gutenberg.org/cache/epub/2000/pg2000.txt>

Ahora quieres escribir un programita que te permita encontrar rápidamente fragmentos de texto como un buscador de google.

Para lograr esto, tu programa primero hace un índice que asocia cada palabra con una lista que contiene la posición donde empieza la palabra (cuenta los caracteres hasta llegar a esa posición)

Luego dado el índice, tu programa tendrá una interfaz (de tu elección) que te permita:

1. Hacer búsquedas tipo Google del libro: Por ejemplo puedes buscar "Don quijote de la mancha" y te saldrá todas las posiciones dentro de ese libro de manera ordenada con un resumen de 1 línea de texto antes, la línea donde aparecen esas palabras y la siguiente. (Obs: TODAS las palabras deberán aparecer, pero tienen que estar a una distancia máxima de 100 caracteres de distancia el uno del otro)

2. Que los resultados aparezcan paginados (es decir M cantidad a la vez), permitiendo ver los siguientes resultados y anteriores.

(Recomendación de interfaz: hacer una interfaz basado en texto.. esta es una clase de algoritmos, no de tu librería favorita para hacer interfaz gráfica)

El objetivo del proyecto es lograr esto en un tiempo O(MlogN + N), donde M es la cantidad de resultados a mostrar en una pagina dada, y N es la cantidad de palabras que hay en el libro.

**Metas de diseño:**

- Utilizar lo mínimo de RAM necesario en tu índice (va a ser MUY grande)

- Que las búsquedas sean rápidas

**Detalles de Implementación**

**El programa principal**

1. Empieza creando un programa que lea un archivo de texto en una subcarpeta relativa al programa. Se lee el archivo del libro línea por línea, divide cada línea en palabras y agrégalo a un diccionario como clave, y asocia una lista de lugares donde ocurre la palabra con dicha clave.

3. Ignora palabras demasiado comunes como: que, el, en, la, por, un, ... Haciendo esto puedes reducir MUCHO la cantidad de RAM sin perder calidad.

4. Ahora implementa una interfaz simple de usuario para que el usuario puedas hacer una búsqueda simple como por ejemplo: materia país pueblo. Esto encontrará pedazos de texto de aproximadamente 200 caracteres que resumen un lugar en el libro.. (Obs: omitir palabras chicas de la búsqueda.. ignóralas)

**Las estructuras de datos y algoritmos necesarios**

1. Diccionario: implementa un diccionario como una Tabla Hash. Puedes suponer que las claves son de tipo char\* para simplificar.

2. Lista: tal vez necesites alguna implementación de lista. Aunque reemplazar la lista por una cola de prioridades implementada como montículo estratégicamente será enormemente útil. Otra opción sería el uso inteligente de árboles y colas binomiales

3. Cola de prioridades: si quieres llegar a la meta de performance una cola de prioridades será necesaria.

**Forma de Entrega**

1. Un informe corto IMPRESO explicando cómo hiciste tu proyecto, cómo se usa, qué problemas viste con el proyecto, cómo lo resolviste, qué lograste, qué no lograste. El informe debe empezar con la frase:

**Certifico que este trabajo es INDIVIDUAL y que lo he completado respetando las normas de ética académica de la materia. Entiendo que si violo estas normas que seré reportado a las autoridades de la Facultad y estaré sujeto a sanciones que incluyen hasta expulsión de la Facultad.**

**Firma: \_\_\_\_\_\_\_\_\_     Aclaración: \_\_\_\_\_\_\_\_   Fecha: \_\_\_\_\_\_\_\_\_**

Y debes firmarlo. Si no firmas esta declaración **NO CORREGIREMOS TU TRABAJO.**

2. Enviar tu proyecto al classroom