Informe Técnico: Proyecto 1 - Buscador con Índice Invertido

Autor: Lorenzo Figueroa Wlack
RUT: 19489501-1
Curso: Estructuras de Datos
Fecha: 13 Junio 2025

Resumen Ejecutivo

Este informe presenta el desarrollo de un sistema de búsqueda de términos en documentos, basado en la construcción de un índice invertido implementado manualmente mediante listas enlazadas simples. El proyecto aborda las necesidades básicas de un motor de búsqueda textual, incluyendo lectura y normalización de palabras, eliminación de stopwords y consulta por intersección de términos. El sistema fue implementado en lenguaje C, con una estructura modular orientada a la claridad y al uso responsable de memoria dinámica.

Estructuras de Datos Utilizadas

- DocNode: Lista enlazada simple para almacenar los identificadores (doc_id) de documentos donde aparece una palabra.
- WordNode: Lista enlazada simple que almacena una palabra y su sublista de documentos (doc_list).
- InvertedIndex: Contenedor principal del índice, apunta al primer WordNode.
- StopWordNode y StopWordList: Estructuras análogas para manejar la lista enlazada de palabras vacías (stopwords).

Cada inserción en el índice garantiza orden ascendente por doc_id y evita duplicados. Las palabras también son insertadas sin repetir.

Procesamiento de Archivos

- Se cargan las stopwords desde un archivo externo (stopwords_english.dat.txt).
- Se recorren los archivos de texto indicados vía argumentos (argv[i]).
- Cada archivo se procesa palabra por palabra, normalizando caracteres y descartando stopwords.
- Las palabras válidas se insertan en el índice invertido con su respectivo doc_id.

Búsqueda de Términos

- Se solicita al usuario una línea de entrada.
- Se divide en palabras con split_words, luego se normalizan y se eliminan aquellas presentes en la lista de stopwords.
- Si la consulta contiene una sola palabra, se accede directamente a su lista de documentos.
- Si contiene varias, se aplica una intersección de listas de documentos ordenadas, entregando sólo aquellos documentos donde aparecen **todas** las palabras.

Scripts y Automatización

- build.sh: Compila el proyecto con make.
- run.sh: Ejecuta el programa con rutas por defecto o argumentos personalizados.
- search.sh: Permite ejecutar el programa con consultas específicas desde terminal.
- test-run.sh: Compila y ejecuta una prueba demostrativa.

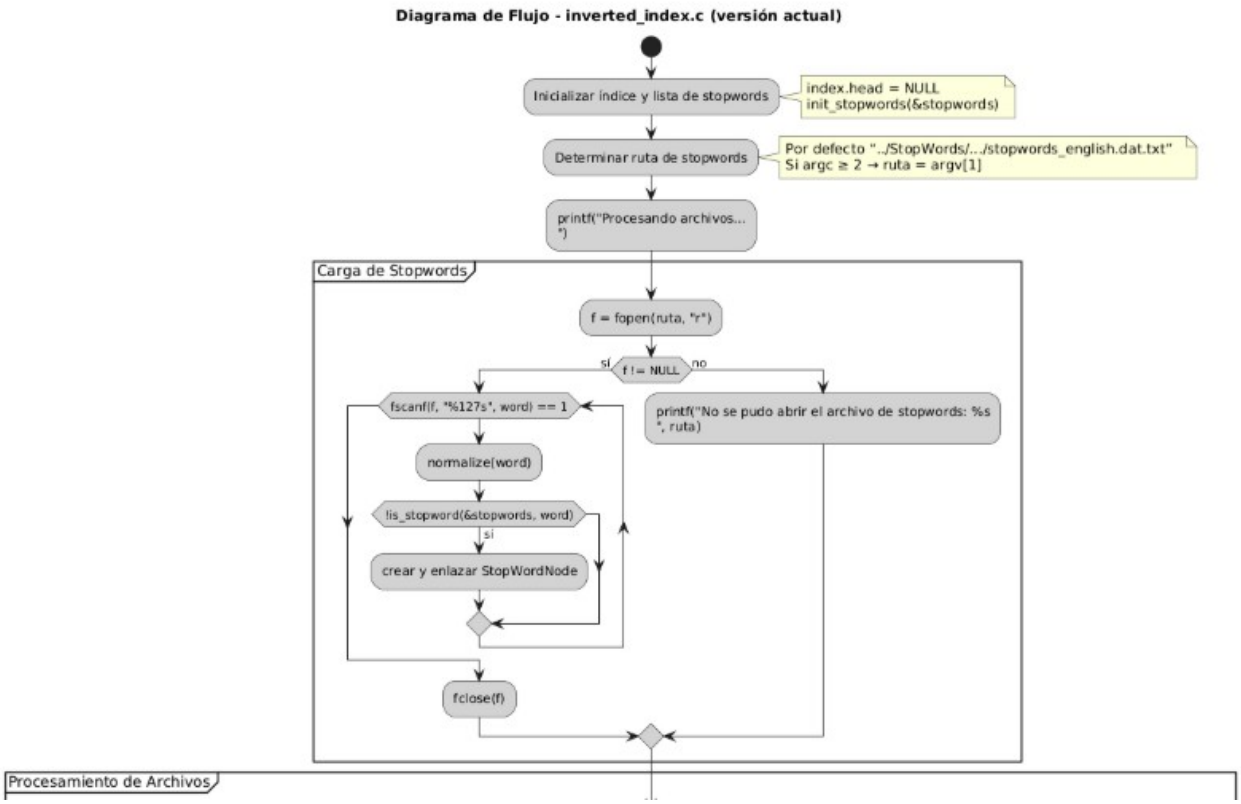
Todos los scripts están diseñados para sistemas UNIX y admiten rutas relativas para facilitar portabilidad.

Diagrama de Flujo

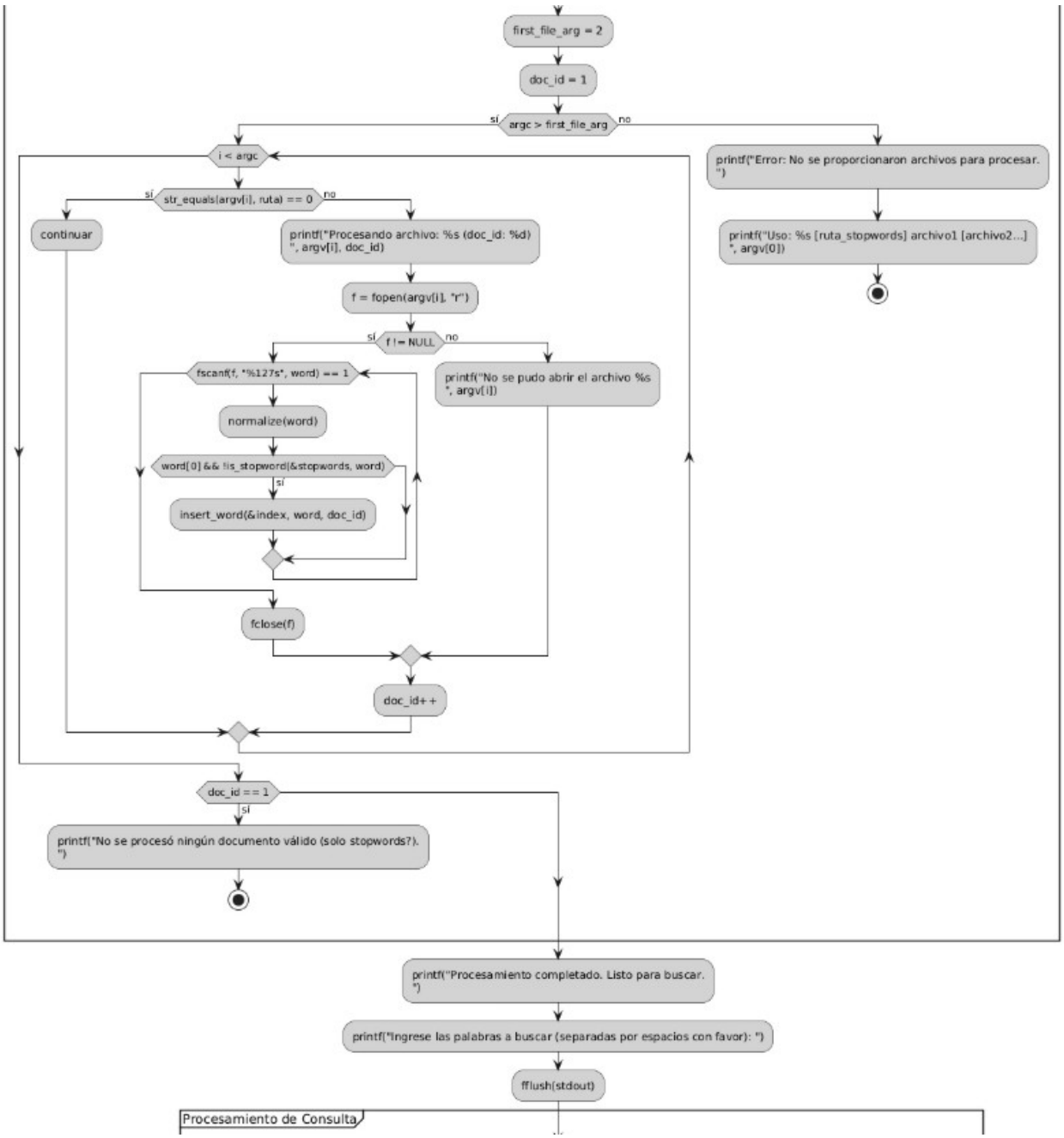
Se adjunta un diagrama que describe el flujo real del programa inverted_index.c, desde la inicialización hasta la liberación de memoria, pasando por las fases de carga, procesamiento de archivos, consulta del usuario y presentación de resultados. Este diagrama está alineado paso a paso con la estructura del código fuente entregado.

Es largo. Lo adjuntaré junto al resto del proyecto, pero se entrega a continuación una idea general.

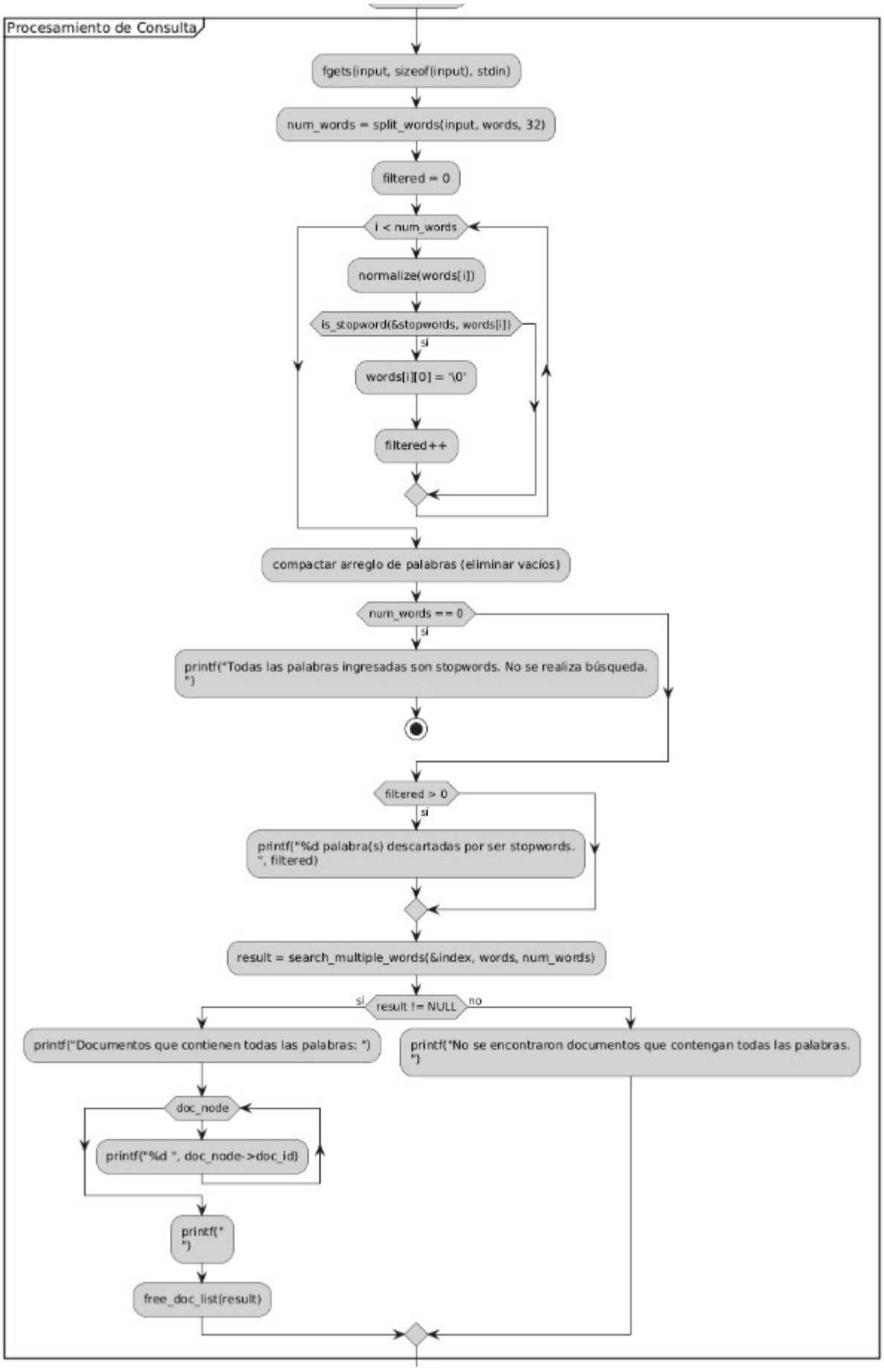
1. **Carga de Stopwords:**



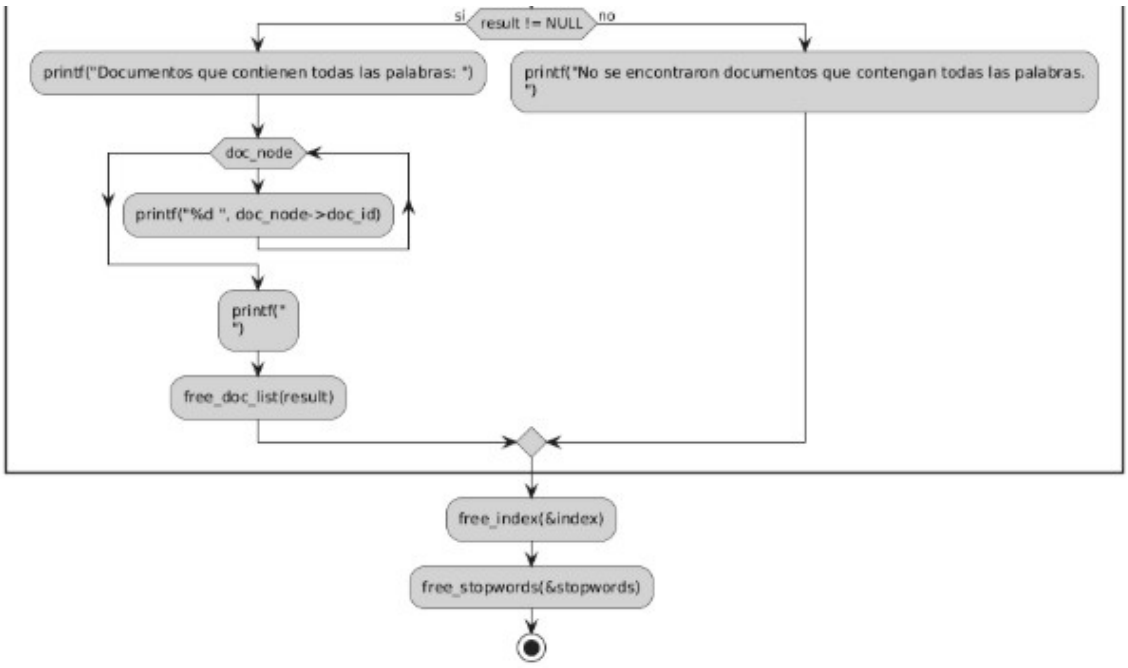
2. Continuando con el flujo del procesamiento de archivos:



3. **Finalizando Procesamiento Consulta.**


Procesamiento de Consulta
fgets(input, sizeof(input), stdin)
num_words = split_words(input, words, 32)
filtered = 0
i < num_words
normalize(words[i])
is_stopword(&stopwords, words[i])
si
words[i][0] = '\0'
filtered++
compactar arreglo de palabras (eliminar vacíos)
num_words == 0
si
printf("Todas las palabras ingresadas son stopwords. No se realiza búsqueda.
")
filtered > 0
si
printf("%d palabra(s) descartadas por ser stopwords.
", filtered)
result = search_multiple_words(&index, words, num_words)
si
result != NULL
no
printf("Documentos que contienen todas las palabras: ")
printf("No se encontraron documentos que contengan todas las palabras.
")
doc_node
printf("%d ", doc_node->doc_id)
printf("
")
free_doc_list(result)

4. **Fin Diagrama de flujo**



Ejemplo de Ejecución (Disponible script)

```
$ ./inverted_index stopwords_english.dat.txt texto1.txt texto2.txt
Procesando archivos...
Procesando archivo: texto1.txt (doc_id: 1)
Procesando archivo: texto2.txt (doc_id: 2)
Procesamiento completado. Listo para buscar.
Ingrese las palabras a buscar: information retrieval systems
Documentos que contienen todas las palabras: 1
```

Conclusión

El presente trabajo constituye un primer acercamiento funcional a la implementación de motores de búsqueda simples utilizando índices invertidos y estructuras de datos manuales. La modularidad del sistema y el control de memoria dinámica permiten la implementacino de futuras extensiones. Se cumplieron los requisitos mínimos funcionales definidos para el proyecto, incluyendo intersección de listas, carga desde archivos externos y procesamiento de consultas en tiempo de ejecución, más no los opcionales.