

MEMORIA PROYECTO BUSCADOR PAG 1

MEMORIA PROYECTO BUSCADOR
AED1 Nov, 2024
Ángel Ruiz Fernandez
Carla Ramos
G2.2 B117

MEMORIA PROYECTO BUSCADOR PAG 2

MEMORIA 1. Analisis del programa 1.1. Clases 1.2. Modulos 1.3. Makefile 1.4. Normalización 1.5. Tabla de dispersión 1.5.1. Tipo 1.5.2. Función de dispersión - comparativa 1.5.3. Liberar (spoiler: está en el stack, RAI) 1.6. Arbol 1.6.1. Tipo 1.6.2. Definición de arbol y nodo 1.6.3. Referencia a paginas 1.6.3. (AVL - no) 1.6.5. Liberar 1.7. Globales (no) 1.8. ChatGPT (no) 2. Listado del código (adjunto - documento siguiente) 3. Informe de desarrollo 4. Conclusiones y valoraciones personales

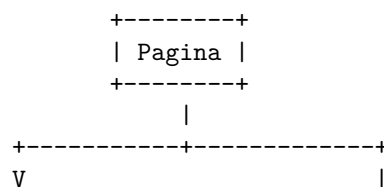
MEMORIA PROYECTO BUSCADOR PAG 3

1. Analisis del programa

1.1. Clases

- class Pagina
Representa una pagina, almacena su url, titulo, relevancia y contenido.
- class PagListIt : public std::list::iterator Usa Pagina Iterador heredado que implementa el operador '<' para poder ser ordenado en un contenedor ordenado. Representa una referencia a elemento de std::list
- struct nodo_trie_t Usa PagListIt Representa un nodo del arbol trie de palabras. De este cuelgan hijos en un diccionario <char, nodo_trie_t>. Relaciona con una lista de referencias a pagina (PagListIt).
- class Arbol Usa PagListIt y nodo_trie_t Contiene la estructura arbol oculta, y posibles operaciones sobre el.
- class Diccionario Usa Pagina, PagListIt y Arbol Contiene la estructura de tabla de dispersión (std::list[N]), su función de hash, y una instancia de Arbol. Expone las posibles operaciones sobre la tabla, además de pasar las operaciones del arbol.

Diagrama de clases



```

+-----+ | | PagListIt | | +-----+ | | | | | | +---+ | | V V | | |
+-----+ + | | | | nodo_ trie_ t | | | | +-----+ + | | | | | | | +---+ +---+
+---+ | V | +---+ + | | Arbol | | +---+ + | | | | +-----+ +-----+ V
+-----+ + + Diccionario | +-----+ +

```

MEMORIA PROYECTO BUSCADOR PAG 4

1.2. Modulos

Cada modulo (menos main) tiene un header asociado

- diccionario.hpp Contiene la declaración de todas las clases y structs.
diccionario.cpp Contiene la definición de todos los metodos de las clases.
- interprete.hpp Contiene la declaración de las funciones que se encargan de interpretar los comandos de la entrada.
interprete.cpp Contiene la definición de las funciones de interpretación de comandos, que llaman a operaciones sobre el diccionario, que se le es pasado por referencia.
- main.cpp Contiene el bucle principal del programa, que lee comandos y llama al interprete. Es propietario de la instanciación del diccionario, donde se almacenan todos los datos de la aplicación.