

Resumen

El proyecto consiste en desarrollar un programa que permita a los usuarios buscar una palabra o varias palabras en el cual se implementarán varios algoritmos de búsqueda en diferentes estructuras de datos, como listas, árboles o grafos, para así poder encontrar de manera eficiente la ubicación de la palabra deseada, también el programa ofrecerá dos opciones para interactuar con el usuario: una interfaz gráfica o interfaz de consola de fácil manejo, en donde el usuario podrá ingresar la palabra que quiera y el programa le mostrará la ubicación de dicha palabra en el diccionario y para lograr esto se utilizarán distintos algoritmos de búsqueda, en los cuales se podrían incluir la búsqueda binaria, búsqueda por bfs, búsqueda por dfs y búsqueda lineal.

Requerimientos

Requerimientos Funcionales:

1. Implementar diferentes algoritmos de búsqueda en estructuras de datos como listas, árboles y grafos.
2. Permitir a los usuarios ingresar un elemento que desean buscar en la estructura de datos.
3. Mostrar la ubicación del elemento buscado en la estructura de datos.

Requerimientos No Funcionales:

2. Herramientas: Si se opta por una interfaz gráfica, se pueden utilizar herramientas como PyQt o Tkinter.
3. Rendimiento: El programa debe ser eficiente en términos de tiempo y recursos al ejecutar los algoritmos de búsqueda en las diferentes estructuras de datos.
4. Usabilidad: Tanto la interfaz gráfica como la versión por consola deben ser intuitivas y fáciles de usar para los usuarios.

Estos son algunos ejemplos de requerimientos funcionales y no funcionales extraídos del enunciado dado. Es importante analizar y documentar los requerimientos específicos del proyecto para asegurar su implementación exitosa.

Tecnología

Dentro de las tecnologías que hemos ocupado y vamos a utilizar tenemos a Miro (plataforma para la organización colaborativa, carta Gantt), draw.io (programa gratuito para

los bocetos), el lenguaje Python (para la programación), la interfaz gráfica, aún en evaluación, el usar PyQt o Tkinter (para el desarrollo de la aplicación) y GitHub como repositorio de desarrollo para el equipo de trabajo.

Ventajas y desventajas

Ventaja de búsqueda binaria:

Algoritmo eficiente para buscar un valor en específico, en este caso, el de una palabra en un diccionario, ya que gracias a este algoritmo la búsqueda binaria reduce el espacio de la búsqueda a la mitad en cada paso

Desventaja de búsqueda binaria:

La búsqueda binaria requiere que la estructura de datos esté ordenada para poder realizar la búsqueda.

Ventaja de BFS:

Tiene la ventaja de encontrar la solución más cercana a la raíz, es decir, que si nuestro proyecto implica buscar una palabra en el diccionario y se desea encontrar la primera ocurrencia más cercana a la posición inicial, BFS es el indicado para eso.

Desventaja de BFS:

Y la principal desventaja de este algoritmo es su uso intensivo en la memoria, ya que si el diccionario es muy grande o tiene una estructura de grafo compleja, la memoria requerida por BFS puede ser un factor a considerar.

Ventaja de DFS:

La ventaja que tiene el uso del DFS es el explorar profundamente un camino antes de retroceder, y esto puede ser beneficioso para nuestro proyecto si deseamos encontrar todas las ocurrencias de una palabra en el diccionario, en lugar de solo la primera ocurrencia en la raíz.

Desventaja de DFS:

La desventaja es que no garantiza encontrar la solución más cercana a la raíz, debido a que se adentra profundamente en un camino antes de retroceder y eso quiere decir que si el objetivo es encontrar la solución más cercana, DFS podrá requerir una exploración más extensa.

Ventaja de Python:

La ventaja principal de Python es que tiene una amplia disponibilidad de bibliotecas y frameworks, y esto nos ayudaría o facilita para tener un mejor desarrollo de nuestro código.

Desventaja de Python:

La desventaja de Python es el consumo de recursos, ya que si se requiere un proceso de grandes volúmenes de datos como en este caso de la biblioteca, es necesario que debemos optimizar el código.

Metodología de desarrollo de software

La manera en la que estamos trabajando para desarrollar el software requerido es con la metodología del **modelo de cascada**, puesto a que lo necesario a realizar para el proyecto es bastante puntual y no requiere de tantas reuniones para recapitular todo lo que se ha hecho. Básicamente es para un desarrollo más rápido.

Carta Gantt

Se muestra diapositiva con imagen, aclarando que hay que modificarlo, y se procede a explicar la metodología de desarrollo de software ocupado para realizar ésta.

Fase de

Requisitos: que se divide en

- Definir proyecto
- Investigación de algoritmos
- Análisis de los algoritmos
- Especificar el trabajo a realizar

Fase de

Diseño: que se divide en

- Realización de maqueta
- Periodo de diseño
- Resolución
- Avance 1 documentación

Fase de

Implementación: que se divide en

- Desarrollo frontend
- Desarrollo backend
- Implementación de entorno
- Avance 2 documentación

Fase de

Pruebas: que se divide en

- Primera fase de prueba
- Primera revisión
- Segunda fase de prueba
- Segunda revisión
- Avance 3 documentación

Fase de

Presentación: que se divide en

- Documentación
- Implementación de entorno completo
- Entrega del proyecto

Caso de uso

Use case, este es un tipo de diagrama en el que vemos a un actor, en este caso un usuario, interactuando con un sistema, el programa de algoritmos de búsqueda, y otro actor que ofrece el sistema, que seríamos nosotros, pero en este caso solo un dataset, y el incluir y extender, incluir para determinar que hay luego de realizar una acción y el extender para entender que puede conllevar una acción.

Entonces, el usuario puede iniciar la aplicación o cerrarlo, lo cuál conlleva abrir o cerrar la ventana principal, la ayuda o manual, el cual abre una ventana secundaria en el que podemos ver la documentación del programa o ejemplos de cómo se utiliza el programa, en conjunto de botones para avanzar o retroceder páginas. Por último tenemos la función principal que es la interfaz de búsqueda, donde debemos ingresar uno o un conjunto de palabras, seleccionar el algoritmo que se ocupara y luego buscar, lo cuál puede mostrar un error, en el caso de que no se haya ingresado palabras, o abrir una ventana secundaria para mostrar el resultado de la búsqueda, que puede ser ninguno y poder cerrar la ventana, o mostrar una tabla de las palabras y sus definiciones, con la opción de cerrar la ventana. Y lo último de lo último, que es que la definición de las palabras vienen de un diccionario que es entregado por un dataset a implementar en el programa.

Boceto

Y he aquí el mockup que refleja lo que deseamos realizar como interfaz gráfica para el programa. Primero tenemos la ventana principal, después la ventana secundaria de ayuda, la lista de algoritmos implementados y su botón, el de obligar al usuario a ingresar una palabra, cuando se abre la ventana secundaria y no se encuentran resultados, y cuando si hay un resultado. Y este sería el resultado "final".

Algoritmos

Búsqueda binaria

La búsqueda binaria es un algoritmo eficiente para encontrar un elemento en una lista ordenada de elementos. Funciona al dividir repetidamente a la mitad la porción de la lista que podría contener al elemento, hasta reducir las ubicaciones posibles a solo una.

BFS

Es un algoritmo de recorrido de graph que explora los vértices en el orden de su distancia desde el vértice de origen, donde la distancia es la longitud mínima de un camino desde el vértice de origen hasta el nodo, como se desprende del ejemplo anterior, en este caso el bfs es útil si deseamos encontrar el valor en el diccionario lo

más rápido posible y también se nos hizo útil en la búsqueda en el diccionario por el simple hecho de que utilizamos grafos.

DFS

Es un algoritmo para atravesar o buscar estructuras de datos de árboles o grafos. Uno comienza en la raíz (seleccionando algún nodo arbitrario como la raíz de un graph) y explora tanto como sea posible a lo largo de cada rama anterior y además se puede utilizar para la generación de laberintos y resolución de juegos como el ajedrez, y al igual que con el bfs, se nos hizo útil utilizar el dfs ya que nos ayuda a explorar más a fondo y se nos facilita para encontrar el valor más rápido.

Búsqueda lineal

Es un método para encontrar un valor objetivo dentro de una lista y ésta comprueba secuencialmente cada elemento de la lista para el valor objetivo hasta que es encontrado o hasta que todos los elementos hayan sido comparados, es decir, que cuantos más elementos contiene la lista, más tiempo tardará el proceso en encontrar un resultado válido.

Bibliografía

MegaPractical (2019, Enero 8). 5 Pasos para definir el Proceso de Desarrollo de Software correcto para tu empresa. [En línea]. Disponible en:
<https://megapractical.com/5-pasos-para-definir-el-proceso-de-desarrollo-de-software-correcto-para-tu-empresa/>

Python Software Foundation (2023, Mayo 23). Documentación de Python en Español (3.11.0). [En línea]. Disponible en: <https://python-docs-es.readthedocs.io/es/3.11/>

Qt Company Ltd. (2023). Qt for Python. [En línea]. Disponible en:
<https://doc.qt.io/qtforpython-6/>

F. Carazo, J. Amat (2023, Febrero). Introducción a grafos y redes con Python. [En línea]. Disponible en:
<https://www.cienciadedatos.net/documentos/pygml01-introduccion-grafos-redes-python.html>

M. López Mamani (2020, Mayo 25). Difference between Breadth Search (BFS) and Deep Search (DFS). [En línea]. Disponible en:
<https://www.encora.com/es/blog/dfs-vs-bfs#:~:text=Aplicaciones%20DFS,en%20un%20grafo%20de%20nodos>

Universidad de Pamplona (2015). Teoría de grafos. [En línea]. Disponible en:
https://www.unipamplona.edu.co/unipamplona/portallG/home_23/recursos/general/11072012/grafos3.pdf

ProgrammerClick (2023). Algoritmo de búsqueda en Python. [En línea]. Disponible en:
<https://programmerclick.com/article/1369731630/>

B. Page (2019, Noviembre 7). Ventajas y Desventajas de la Búsqueda Lineal. [En línea]. Disponible en: <https://ed.team/comunidad/ventajas-y-desventajas-de-la-busqueda-lineal>

L. Salcedo (2020, Diciembre 22). Algoritmos de Búsqueda en Python. [En línea]. Disponible en: <https://pythondiario.com/2017/10/algoritmos-de-busqueda-en-python.html>

A. Ali (2022, Enero 28). Sorting and Searching in Python. [En línea]. Disponible en: <https://code.tutsplus.com/tutorials/sorting-and-searching-in-python--cms-25668>

H. Kareem Shaik (2021, Septiembre 22). Implementaciones de algoritmos de búsqueda en Python. [En línea]. Disponible en: <https://geekflare.com/es/python-search-algorithms/>

M. L. (2018, Abril 13). Métodos A* y Best First Search en grafo de ciudades. [En línea]. Disponible en: <https://es.stackoverflow.com/questions/153777/m%C3%A9todos-a-y-best-first-search-en-graf-o-de-ciudades>

secnot (2016, Diciembre 30). Búsqueda en Profundidad en Python. [En línea]. Disponible en: <https://es.stackoverflow.com/questions/41215/b%C3%BAsqueda-en-profundidad-en-python>

Wikibooks (2014, Noviembre 1). Estructuras de datos dinámicas/Algoritmos de búsqueda. [En línea]. Disponible en: https://es.wikibooks.org/wiki/Estructuras_de_datos_din%C3%A1micas/Algoritmos_de_b%C3%BAsqueda

Equipo editorial, Etecé (2023, Enero 23). Algoritmo: ventajas, desventajas, ejemplos y características. [En línea]. Disponible en: <https://humanidades.com/algoritmo/>

Redacción KeepCoding (2022, Diciembre 13). Ventajas y Desventajas de Python. [En línea]. Disponible en: <https://keepcoding.io/blog/ventajas-y-desventajas-de-python/>

H. Jindal (2023, Enero 30). Búsqueda lineal. [En línea]. Disponible en: <https://www.delftstack.com/es/tutorial/algorithm/linear-search/>

P. Guillen (2011). Búsqueda lineal (secuencial). [En línea]. Disponible en: https://pier.guillen.com.mx/algorithms/03-ordenacion/03.6-busqueda_lineal.htm

T. Cormen, D. Balkcom (2015). Ciencias de la computación: Búsqueda binaria. [En línea]. Disponible en: <https://es.khanacademy.org/computing/computer-science/algorithms/binary-search/a/binary-search>

D. Gil (2020, Abril 20). Búsqueda binaria. [En línea]. Disponible en: <https://medium.com/@daniel.patrick.gil/b%C3%BAsqueda-binaria-dec386ad8525>