

LLISTA EXERCICIS (OBLIGATORIS I ADDICIONALS)

Paola Gonzalez, Mia Insa i Anna Vilacís

LAB 1:

- ☒ ~~* Implement the links linked list~~
- ☒ ~~* Initialize Document from file path~~
- ☒ ~~* Implement the documents linked list and load all documents from the dataset~~
- ☒ ~~* Print the documents to the CLI and allow the user to select and view one by index~~
- ☒ ~~* Format your code with clang-format: "make f"~~
- ☐ *** Write at least 1 unit test for the document parsing functionality
- ☐ *** Write at least 3 unit tests for the documents linked list
- ☐ *** Write at least 3 unit tests for the links linked list
- ☒ ~~*** Fix any memory leaks detected by Valgrind: "make v"~~

LAB 2:

- ☒ ~~* Implement the query linked list~~
- ☒ ~~* Initialize Query from string~~
- ☒ ~~* Linear search through the documents and print the results to CLI~~
- ☒ ~~** Show the last 3 queries using a queue~~
- ☒ ~~** Allow excluding keywords~~
- ☐ *** Write at least 3 unit tests for the query linked list
- ☐ *** Write at least 1 unit test for the linear search functionality
- ☐ **** Allow adding or conditions to the search query
- ☒ ~~*** Fix any memory leaks detected by Valgrind: "make v"~~

LAB 3:

- ☒ ~~* Implement a hashmap~~
- ☒ ~~* Add all (word, documentids) to the reverse index hashmap~~
- ☒ ~~* Use the reverse index to efficiently search through the documents~~
- ☒ ~~** Normalize words in the parser (uppercase, punctuation, etc.)~~
- ☒ ~~*** Write at least 3 unit tests for the hashmap~~
- ☒ ~~*** Write at least 1 unit test for the search functionality using hashmap~~
- ☒ ~~*** Fix any memory leaks detected by Valgrind: "make v"~~
- ☐ **** Suggest and implement one improvement to lower the search runtime complexity
- ☐ *** Allow serializing/deserializing the reverse index from/to a file
- ☐ **** Implement short and full barrels (match keywords in the title first)
- ☐ **** Show document snippets of the text surrounding the matched word

LAB 4:

- ☒ ~~* Implement a directed document graph~~
- ☒ ~~* Get indegree of a document in the graph and print it as the relevance score~~
- ☒ ~~** Sort search results according to the relevance score (choose an adequate algorithm)~~
- ☐ *** Write at least 1 unit test for the graph
- ☐ *** Fix any memory leaks detected by Valgrind: "make v"
- ☐ **** Implement the page rank algorithm for computing the relevance score
- ☐ *** Precalculate and cache relevance scores
- ☐ *** Serialize/deserialize cache relevance scores from/to a file

ÚS DE LA IA:

En aquest treball hem fet servir la IA per complementar la nostra feina i assegurar que el projecte avançava de manera clara i ben estructurada. Ens ha ajudat en diversos aspectes:

- Primer de tot, ens ha servit per aclarir quins passos havíem de seguir, distingint les parts que ja teníem fetes i les que faltaven. Això ens ha ajudat a tenir un ordre de feina més lògic.
- També ens ha permès desglossar millor els requisits i entendre exactament què calia fer amb funcions com el càlcul de la rellevància amb l'indegree o la ordenació dels documents.
- A més, ens ha donat exemples i explicacions que ens han servit de guia per implementar aquestes funcions bàsiques i integrar-les al projecte.
- Igualment, ens ha ajudat a resoldre conflictes a l'hora de compilar ja que el procés era una mica diferent al que havíem utilitzat fins a dia d'avui.
- Finalment, la IA ens ha resultat molt útil per corregir errors que teníem al codi, ja que ens ha indicat possibles problemes o maneres de millorar la implementació i assegurar-nos que funcionés correctament.