



Universidade do Minho
Escola de Engenharia

Guião-3

Laboratórios de Informática III

Luís Vilas, João Brito e João Silva

Grupo 68

e-mails: a91697, a95641, a91671 @alunos.uminho.pt

Universidade do Minho, Departamento de Informática - Braga, Portugal

2021/2022

Resumo: Trabalho desenvolvido com base no “guião-3”, com conhecimentos essenciais da linguagem C e de Engenharia de Software, onde o foco principal do guião esteve em modularidade e encapsulamento, estruturas dinâmicas de dados, medição de desempenho e otimização de consulta de grande volume de dados.

O presente guião pede para desenvolver um mecanismo de interação alternativo através da criação de um menu que permita interação com o programa e que torne mais fácil a interpretação do mesmo, sendo que este menu tem de ser visualmente apelativo para qualquer utilizador “padrão”. Este menu deve ser sempre apresentado quando o programa não recebe um ficheiro de comandos como argumento. Se o programa for invocado sem argumentos, este deve processar os ficheiros de dados “padrão”, mostrar, ao utilizador, um menu que liste as diferentes funcionalidades que dispõe o programa e pedir para o utilizador selecionar uma destas funcionalidades. De seguida, o programa deve retornar os resultados da funcionalidade escolhida anteriormente, mostrando estes num formato apelativo e que permita a consulta e navegação dos mesmos. Sendo assim, necessário retornar os resultados num submenu sob a forma de páginas e que seja possível navegar sobre as mesmas. Tendo em atenção que os ficheiros de entrada iniciais podem conter registos inválidos, ou seja, o programa tem que garantir que as *queries* são executadas sobre um conjunto de ficheiros válidos, logo estes ficheiros têm de passar por uma verificação, já definida no guião 1.

Para este guião, pretende-se também que sejam criados testes que validem e avaliem, simultaneamente, o funcionamento de cada *query* solicitada no guião anterior, verificando também se o tempo de execução desta é inferior a 5 segundos. Além disso, é também pedido que se implemente um mecanismo eficiente de gestão de dados sem recurso a base de dados e cujo funcionamento interno seja opção para quem

pretenda utilizar os módulos que implementam tal mecanismo, desta forma deverão ser aplicados os princípios do encapsulamento e modularidade.

Este guião tem como principais objetivos: consolidar os conceitos de modularidade e encapsulamento aplicados e referidos anteriormente, realizar suporte a diferentes mecanismos de interação entre programa e utilizador, manipular e otimizar a consulta de grande volume de dados, realizar testes funcionais que permitam avaliar o desempenho e os resultados obtidos a partir do programa.

Analisando o guião, tem-se que é necessário terminar o guião 2, pois as *queries* que foram abordadas neste são essenciais para o funcionamento do programa no guião 3, e que também é útil realizar algumas mudanças em relação ao guião anterior, tendo em atenção principalmente à gestão de dados, pois este guião tem uma dimensão bastante superior aos utilizados nos guiões anteriores. Assim, de forma a manter padrões razoáveis de usabilidade e responder em tempo útil aos pedidos do utilizador, é necessário organizar parte dos dados em ficheiro. Para isso começou por se implementar uma otimização da *glib* que consiste em utilizar a função *g_tree_lookup* com o objetivo de que a procura nas *gtrees* seja mais rápida e eficiente. Também se deixou de gravar os *char** com tamanho elevado dos ficheiros, tendo em vista fazer-se uma melhor utilização da memória tendo em conta as dimensões dos ficheiros, de forma a só se irem buscar estes quando necessários para o programa.

Tendo em conta que o grupo acabou por não realizar todas as queries propostas anteriormente e que algumas destas se encontravam com resultados errados ou não otimizados, acabou por se reestruturar as queries parametrizáveis. Passando a utilizar a struct Glist para as queries em vez de char*, uma vez que usando esta struct torna-se mais fácil utilizar os “top N”, permitiu criar uma lista ligada que mostra este top conforme a página onde se encontra.

As alterações efetuadas nas queries em relação aos guiões anteriores estão enunciadas abaixo:

- ***Query 1:***

Durante a leitura do ficheiro dos users, armazenou-se na struct dos catálogos a quantidade de users, de bots e organizações para posteriormente poupar tempo na execução da query 1.

- ***Query 5:***

Como se removeram todos os char* das structs, o username do utilizador é obtido, quando necessário, através da procura da linha (armazenada na struct) no ficheiro original.

- ***Query 6:***

Como se removeram todos os char* das structs, o username do utilizador é obtido, quando necessário, através da procura da linha (armazenada na struct) no ficheiro original, tal como na query anterior.

- **Query 7:**

Quando é necessário obter a descrição do repositório, passou-se a abrir o ficheiro respetivo e procurar pelo nº de linha correspondente. Deste modo, poupou-se espaço em memória, apesar de aumentar o tempo de execução da mesma.

- **Query 8:**

Percorreu-se o repositório dos commits para se poder obter os mesmos a partir de uma determinada data e assim, filtrar os repositórios com os quais se estava a trabalhar, obtendo também o nº de commits a partir dessa mesma data. Por fim, ordena-se e insere-se numa lista ligada as linguagens mais usadas por ordem decrescente.

Tendo em conta a falta de tempo, não foram realizadas as queries 9 e 10. É de notar, por fim, que a realização deste projeto proporcionou mais conhecimento nesta linguagem ao nível do encapsulamento e na gestão de memória.