



RELATÓRIO

GUIÃO 1

Laboratórios de Informática III

2021/2022

Licenciatura em Engenharia Informática | 2ºAno

Diana Ribeiro Mateus | A95985

Pedro Marcelo Bogas Oliveira | A95076

Rodrigo José Teixeira Freitas | A96547

Índice

Introdução.....	2
Desenvolvimento	3
Conclusão.....	4

Introdução

No âmbito da Unidade Curricular de Laboratórios de Informática III, foi proposta a realização de um trabalho que inclui vários guiões, sendo que este relatório incide sobre o Guião 1.

O Guião-1 consiste na utilização da linguagem C para o tratamento de ficheiros, nomeadamente ficheiros do tipo CSV (*comma-separated values*), incluindo a sua verificação, geração e comparação. A verificação dos ficheiros inclui a apuração de linhas bem estruturadas, segundo as *guidelines* fornecidas no que concerne ao tipo de campos, isto é, se os dados são do tipo inteiro não negativo, *string*, etc, e posteriormente, a geração de ficheiros com as linhas corretas. Para além disso, a comparação de ficheiros possibilitou a filtração das linhas corretamente relacionadas entre ficheiros, com, novamente, uma geração final de ficheiros.

Os objetivos principais deste trabalho passam pela utilização da linguagem C e de todas as suas funcionalidades (operações sobre ficheiros, *parsing* de dados, memória dinâmica, estruturas complexas de dados), para além da consolidação de todos os conhecimentos essenciais ao desenvolvimento de um projeto, nomeadamente a compilação, linkagem, depuração de erros e gestão dos respetivos repositórios.

Desenvolvimento

A estratégia utilizada para a execução do exercício 1 passou pela utilização de listas ligadas para cada documento. Este método permite garantir a salvaguarda dos dados dos ficheiros, através da utilização de memória dinâmica, ou seja, através da utilização de uma estrutura de dados linear e dinâmica que torna a manipulação dos ficheiros relativamente eficiente. Para além disso, foi utilizada a função *fgets* para guardar o cabeçalho dos ficheiros.

De forma a manipular os ficheiros, foram criadas funções respetivas que permitem não só percorrer todas as linhas, como também verificar os campos e imprimir no novo ficheiro o cabeçalho e as linhas corretas. Existem ainda, algumas funções universais a todos os ficheiros, nomeadamente, *testInt*, *testString*, *testDateHour*, isto é, funções que testam o tipo dos campos comuns, e outras que testam os campos exclusivos de cada ficheiro, por exemplo, *testFollowersList (Users)* e *testBool (Repos)*.

Foram criados ficheiros do tipo “.c” com as funções relativas a cada um dos CSV’s fornecidos, com funções que permitiam percorrer cada ficheiro e guardar os respetivos campos em variáveis distintas da estrutura de dados utilizada. Estas funções apenas guarda na lista, os campos e as linhas corretamente identificados e através de uma outra função auxiliar imprime no novo ficheiro as linhas corretas, que aquando da verificação foram gravadas numa variável específica da lista ligada.

No exercício 2, foram utilizadas árvores binárias de procura para guardar os *ID’s* de todos os ficheiros e permitir a comparação entre eles. Esta estratégia permite a utilização de um algoritmo de pesquisa dinâmico e muito eficiente no que concerne ao tempo de procura, considerando que esta se encontra razoavelmente balanceada. Como auxílio, foram criadas funções de *insert* e *search* em árvores, que possibilitam a inserção de elementos na árvore binária e a procura desses mesmos elementos. Para além disso, foi utilizada uma estratégia iterativa para o desenvolvimento destas funções.

De forma a comparar os três ficheiros, foram utilizadas duas funções que permitiam percorrer cada um dos ficheiros que seriam necessários comparar com os *ID’s*. Estas funções facilitam a passagem pelos campos dos ficheiros por ordem em que aparecem, e viabilizam a sua comparação com os *ID’s* necessários. Foram criadas *flags* que permitem ao longo do documento a verificação de um ou mais *ID’s* e, por último, a impressão das linhas comuns nos respetivos ficheiros finais.

Além disso, foi criado um ficheiro *Makefile*, que define um *set* de tarefas a ser executado, essencialmente, na execução e compilação dos ficheiros “.c” e “.h”, criando também ficheiros “.o” que foram utilizados para a depuração de erros.

Conclusão

A realização deste trabalho foi importante na consolidação dos conhecimentos da linguagem C, através da aplicação dos conceitos de memória dinâmica e estrutura de dados, bem como extração de dados e manipulação de ficheiros. Sendo assim, podemos concluir que este objetivo foi cumprido.

Para além disso, a utilização do *Makefile* possibilitou a utilização de ferramentas de compilação, nomeadamente a utilização de comandos de otimizações, bem como linkagem e a criação de ficheiros que permitiam a depuração de erros. A gestão dos repositórios ficou um pouco aquém do esperado. Na generalidade, estes objetivos foram cumpridos.

De uma forma geral, o trabalho foi realizado de forma fluída, sem grandes dificuldades ou erros. Contudo, um impedimento que encontramos ao longo deste processo foram os erros de compilação entre diferentes máquinas. Estes obstáculos foram ultrapassados, permitindo-nos perceber que os diferentes compiladores utilizados, nomeadamente o *gcc* no *Ubuntu* e o *Clang* no *macOS*, funcionam de forma diferente. Estas diferenças na compilação permitiram encontrar algumas incongruências no código desenvolvido, tendo sido, de certa forma, uma experiência muito educativa.

Em conclusão, este trabalho foi realizado cumprindo os objetivos, tendo não só os testes realizados ao programa comprovado esta mesma permissa, mas também o conhecimento que conseguimos obter ao longo do processo, no que concerne ao desenvolvimento de *Software* em equipa.