



Universidade do Minho
Escola de Engenharia

Relatório de Laboratórios de Informática III

Guião 1

2ºano - 1ºsemestre

Trabalho realizado por:

Joana Branco (A96584)

Joana Pereira (A97588)

Marta Sá (A97158)

Braga, 8 de novembro de 2021

Exercício 1

O exercício 1 consiste na seleção dos registos que contêm diversos parâmetros, como nome de usuário, data, entre outros, que se encontrem válidos. Segundo o enunciado a data deve respeitar um mínimo e as listas de seguidores e pessoas que segue devem ser correspondidas.

A estratégia adotada para a resolução deste exercício partiu inicialmente pela validação do parâmetro de entrada (`validPar`). Esta função verifica se um número é inteiro positivo e é utilizada como auxiliar para muitas outras funções. De seguida, comparou-se os tipos de `Type` possíveis e válidos (`User`, `Bot` e `Organization`) com o que é apresentado (`validType`). A data tem um formato a respeitar e com a função, já pré-definida, `strptime` verificou-se se esta era válida e correspondia aos limites mínimos dados pelo enunciado do guião, tendo em conta se o ano é bissexto ou não (`validDate`). Prosseguiu-se para a validação do login tendo em conta que só se pode receber algo do tipo `string` (`validString`). A função `validBool` irá servir de ajuda para validar um `Type`. Já a `validList` verifica se todos os elementos de uma lista são números inteiros positivos. A função `init_user` basicamente, junta todas as funções anteriormente explicadas e inicializa o usuário já validado em todos os parâmetros. A `init_repos` tem um desenvolvimento semelhante, mas com ainda mais casos de verificação. À semelhança destas duas últimas, a `init_commits` inicializa o commit. De seguida, irão ser avaliados todos os users inseridos no input. E, da mesma maneira, para os commits e repositórios.

Neste exercício foram sentidas algumas limitações na comparação das listas que comparavam os campos `following` e `followers` com a `following_list` e `follower_list` mas foram ultrapassadas e no fim conseguimos eliminar as 300 linhas inválidas como o suposto.

Exercício 2

O exercício 2 compara os dados entre os ficheiros produzidos no exercício anterior, filtrando os que são inválidos. De acordo com o enunciado, devem ser removidas linhas que apresentem commits de usuários e repositórios inexistentes e, com a mesma semelhança, devem ser removidas linhas que apresentem repositórios de utilizados inexistentes e que não inclua nenhum commit.

A estratégia adotada para a resolução deste exercício partiu pela inicialização do repositório, tal como se fez para o exercício 1 (initRepos). À semelhança do init_commits do exercício anterior, criou-se a função init_commits2. Partiu-se depois para a eliminação de commits com usuários inexistentes e para a eliminação de commits com repositórios inexistentes (comp_id e comp_rep). A função get_size permite saber qual o tamanho do ficheiro que está a ser avaliado. De seguida, a função doUser2 escreve os usuários num outro ficheiro de saída e devolve o tamanho da lista de todos os usuários. Tal como se fez no exercício 1, a função doRepos2 é responsável por avaliar se os repositórios são ou não válidos. O mesmo se repete para a doCommits2.

Neste exercício, apesar de não termos sentido tantas limitações quanto no primeiro exercício, o código tornou-se pouco eficiente e demorado na compilação, problema este que é do nosso interesse tentar melhorar para ser implementado nos próximos guiões.