Guião-2

Luís Vilas, João Silva e João Brito

University of Minho, Department of Informatics, 4710-057 Braga, Portugal
e-mail: a95641, a91671, a91697@alunos.uminho.pt

Resumo: Trabalho desenvolvido com base no “guião-2”, com conhecimentos essenciais da linguagem C e de Engenharia de Software, onde o foco principal do guião esteve em modularidade e encapsulamento, estruturas dinâmicas dedados, e medição de desempenho.

Keywords: Catálogo de Utilizadores, Catálogo de Commits, Catálogo de Repositórios, Queries estatísticas, Queries parametrizáveis

Relatório

Num primeiro contacto com o guião, reparou-se que iria ser preciso utilizar a glib pois ia se ter que guardar os registos numa estrutura de dados de eficiente procura e armazenamento. A glib cria árvores binárias de procura balanceadas, sendo a procura nesta logarítmica.

Já de seguida, começou-se por desenvolver o catálogo de utilizadores, onde é guardado todos os utilizadores provenientes de registos válidos do ficheiro users.csv, a partir da função void loadUsers(char *fileName, GTree *t), que chama a função void buildUsers(char* line, GTree *t) insere-se os utilizadores conforme o seu id.

Depois, criou-se o catálogo de commits, onde se armazena convenientemente os commits recolhidos do ficheiro commits.csv, a partir da função void loadCommits(char* filename, GTree *t), que chama a função void buildCommits(char* line, int lineNumber, GTree *t) que insere os commits conforme o nº da linha do ficheiro.

De seguida, desenvolveu-se o catálogo de repositórios, no qual se guarda todos os repositórios e respetiva informação contida no ficheiro repos.csv, a partir da função void loadRepos(char* filename, GTree *t), que chama a função void buildRepos (char* line, GTree *t) que insere os repositórios conforme o seu id.

Já mais tarde, começou-se a realizar as queries, na querie de id 1 criou-se a função gboolean countType_internal(gpointer key, gpointer value, gpointer data) para ser utilizada na g_tree_foreach da função char* numTypes(GTree *t) que retorna a quantidade de bots, organizações e utilizadores, com o output desejado.

Na querie de id 2, a gboolean conta_total_por_repo(gpointer key, gpointer value, gpointer data), que calcula o nº de colaboradores por repositório e irá ser utilizado numa g_tree_foreach para calcular o nº total de colaboradores na função float numOfCommitters(GTree *commits, GTree *repos) que irá retornar a média de colaboradores por repositório.

Na querie de id 3, criou-se a função gboolean conta_bots (gpointer key, gpointer value, gpointer data) que conta o nº de bots por repositório sendo utilizado numa g_tree_foreach, depois na função int quantidadeReposWithBots(GTree* users, GTree* commits) é se calculado o tamanho da hash table que corresponde à quantidade de repositórios com bots, como se pretende.

Na querie de id 4, criou-se a função int commitsPerUser(GTree *commitTree, GTree *userTree) que conta os nodos da árvore dos commits e da árvore dos useres a partir da função g_tree_nodes, por sua vez, a média é a divisão destes 2 valores, pelo que, se retorna esta arredonda a duas casas decimais, como é pedido no guião.

Na querie de id 5, criou-se a gboolean g_func_StrartDate (gpointer key, gpointer value, gpointer data) que verifica se a data de um dado commit está entre o intervalo e coloca-o numa GTree se sim. A função gboolean g_func_findMoreOccur (gpointer key, gpointer value, gpointer data) é utilizada para saber o número de ocorrências de cada comitter. A função gboolean g_func_UserFromNumCommits (gpointer key, gpointer

value, gpointer data) é usada para retornar o user id e o número de commits realizados pelo mesmo. Por fim, a função char* numOfUsersActive(GTree *treeCommits, GTree *users, int nTop, char* dateStrat, char* dateEnd) que irá retornar o output desejado conforme o guião 2.

Na querie id 6, utilizou-se a função gboolean find_repos_with_language (gpointer key, gpointer value, gpointer data) para identificar os repositórios com uma dada linguagem. Criou-se também a função gboolean count_per_user_num_commits_repos (gpointer key, gpointer value, gpointer data) que conta para um dado utilizador o nº de commits deste para um repositório. Estas funções serão utilizadas em g_tree_foreach na função char* topNlanguage_language(GTree *users, GTree *commits, GTree *repos, int nTop, char* lang) que irá retornar o output desejado de N linhas em que cada uma indica o id, login e nº de commits do user, conforme a quantidade de commits dos users para uma dada linguagem.

Na querie id 7, criou-se a função gboolean adiciona_repos_sem_commmits (gpointer key, gpointer value, gpointer data) de forma a adicionar a uma hash table os repositórios sem nenhum commit, depois criou-se a gboolean gera_hashtable_repos_c_commits (gpointer key, gpointer value, gpointer data) que irá armazenar os repositórios de acordo com a data do seu último commit. Estas funções serão utilizadas na função char* inactiveRepos(GTree *commits, GTree *repos, char* dateStart) de forma a perceber quais são os repositórios sem commits, ou seja inativos, a partir de uma certa data.