Relatório nº 2 Laboratórios de Informática 3 Grupo 58

Nuno Miguel Leite da Costa A96897
Daniel Jose Silva Furtado A97327

A97327 A96897

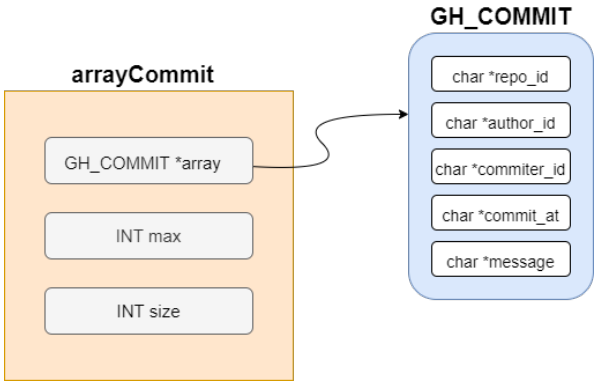
1 Resumo

Para este segundo guião foi nos proposto a implementação de 10 queries responsáveis por a interação do utilizador com o programa. Este relatório tem como objetivo documentar as estratégias que utilizamos e as limitacações que encontramos com este segundo guião e também comentar a medição de aspetos de desempenho.

2 Tipos e Estruturas de Dados

2.1 Estruturas de Dados

De maneira a melhorar o funcionamento das queries distribuimos os dados de cada ficheiro em uma struct .



Exemplo da estrutura arrayCommit

2.2 Organizção das Estruturas de dados

A estrutura de dados que implementa cada um parse é constituída por : um array dinâmico, com o objetivo de armazenar todos os dados respetivos para

cada parametro (GH-USER,GH-COMMITS,GH-REPOS). Um inteiro que representa o tamanho máximo do array dinâmico. Um inteiro que representa o tamanho atual do array.

3 Queries

Nos tópicos seguintes, iremos apresentar as queries e as estratégias escolhidas para responder as estas.

3.1 Query 1 - Quantidade de bots, organizações e utilizadores.

Nesta querie, é carregado o ficheiro users.cv, onde filtramos e contamos o número de cada paramêtro de Type. Esta querie está encarregue pela inicialização do documento especifico (commands X.txt) com o output do número de Bot,User e Organization, respetivamente.

```c
void query1 (int x) {

    //determina o nome do ficheiro
    char text[100];
    sprintf(text, "../guiao-2/saida/command_%d.txt", x);
    FILE *fp = fopen (text, "w+");
    if (!fp) printf("Não abre");
    INFO_USER users = data_User();

    int bot = 0, user = 0, organization = 0;

    for(int i = 0; i< users->size; i++){

        char * s = get_Type(users->array[i]);

        if (!strcmp(s, "Organization")) organization++;
        if (!strcmp(s, "Bot")) bot++;
        if (!strcmp(s, "User")) user++;
    }
    fprintf (fp, "Bot: %d\nOrganization: %d\nUser: %d\n",bot,organization,user);
    fclose(fp);
}
```

Função query 1 utilizada

3.2 Query 2 - Número médio de colaboradores por repositório

Para responder a esta querie, iniciamos a nossa estratégia com a criação de funções de ordenar segundo o author-id e commiter-id(quicksort). De seguida, realizamos uma procura de cada id proveniente do users.cv, no array ordenado pelo quicksort de author-id. Caso encontrasse o id, uma variavel "colaboradores"incrementava, caso contrário, esses ids eram acrescentados a um array novo (inicialmenete sem informacões). Após este processo, verificavamos estes ids que estão no array novo e procurá-los nos array de commiter-id ordenado. Caso encontrasse incrementava os "colaboradores". Por fim, obtemos o número tolal de repositórios através do size presente na struct associada aos repos.csv e calulamos a média entre estes dois valores.

3.3 Query 3 - Quantidade de repositórios com bots

Nesta querie optamos por utilizar uma estratégia semelhante á query 2. Calculamos o número de colaboradores da mesma forma, aplicando restrições ao Type, contando deste modo apenas os colaboradores que fossem Bot.

3.4 Query 4 - Qual a quantidade média de commits por utilizador

Esta quarie tem como objetivo calcular a média entre o número total de utilizadores e o número total de commits. A estratégia utilizada foi bastante simples, apenas tivemos de utilizar os valores users-¿size e commits-¿size provenientes das structs respetivas.

```
void query4 (int x) {
    char text[100];
    sprintf(text, "../guiao-2/saida/command_%d.txt", x);
    FILE *fp = fopen (text, "w+");
    if (!fp) printf("Não abre");
    INFO_USER users = data_User();
    int n_user = users->size;
    INFO_COMMIT commits = data_Commit();
    int n_commit = commits->size;

    float i = (float)n_user / (float)n_commit;

    fprintf(fp, "%0.2f", i);
    fclose(fp);

}
```

Função querie 4 utilizada

3.5 Query 5 - Top N de utilizadores mais ativos num determinado intervalo de datas

Em primeiro lugar verificamos se a data proveniente dos commits está entre os intervalos das datas presentes nos argumentos da query. Aos isso, inserimos num array inicializado inicialmente , o commiter-id correspondente e procuramos o mais frequente para encontrar o TOP. Por fim, retiramos todas as ocorrências deste mesmo elemento e repetimos o processo de encontrar novamente o mais frequente de modo a encontrar o TOP N.

3.6 Query 6 - Top N de utilizadores com mais commits em repositórios de uma determinada linguagem

Começamos por inserir num array todos os repo-id dos repositórios que apresentassem a linguagem passada nos argumentos da query. De seguida, verificamos quais os commiter-id associado a cada repo-id para um array novo. Por fim, vemos qual o commiter-id mais frequente e documentamos no ficheiro inicializado, removendo-o de seguida e repetindo este processo para encontrar o TOP n.

3.7 Query 7 - Top N de utilizadores mais ativos num determinado intervalo de datas

Para implementar esta query começamos por comparar a data do input e a data de cada commit-at, caso esta seja menor que a data do input, é guardado o repo-id num array. Posteriormente, ordenadamos o array e eliminamos os repetidos. De seguida, ordenamos o ficheiro repos segundo o valor id e fazemos uma procura binária por cada valor do array pelo ids dos repos. Por fim, obtemos a message.

4 Limitações

Ao resolver este guião, as principais limitações que encontramos pelo o caminho foram: primeiramente como tratar do parsing dos dados de forma fácil de entender e de utilizar. Segundamente, como trabalhar com o armazenamento da memória, utilizando posteriormente memória dinâmica. Por fim, encontramos limitações ao idealizar as estratégias a utilizar em cada query de forma eficiente e simples.

5 Custo computacional

Na nossa opinião,concluímos que os tempos estão razoáveis. Parte das razões que justificam estes tempos devem-se ao facto de as estratégias não serem as mais eficientes. Contudo, houve claras melhorias visto que no ínicio o nosso programa demorava mais de 10 minutos a percorrer os ficheiros do Guião2 na BlackBoard.