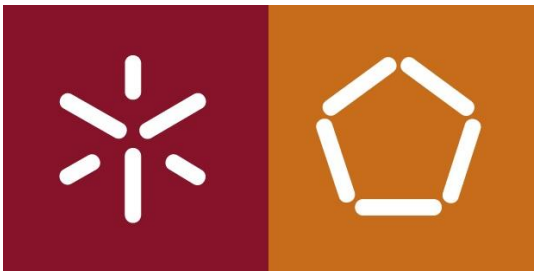
Universidade do Minho

Mestrado Integrado em Engenharia Informática



Stack Overflow - Relatório

Laboratórios de Informática III

Grupo 032

Trabalho realizado por:	*Número*
Ana Filipa Pereira	A81712
Helena Martins	A82500
Pedro Medeiros	A80580

12 de Junho de 2018

Conteúdo

1 Introdução

No âmbito da unidade curricular de Laboratórios de Informática III do 2º ano do Mestrado Integrado em Engenharia Informática, foi proposto pelos docentes o desenvolvimento de um projeto em linguagem Java capaz de processar ficheiros XML que armazenam as várias informações utilizadas pelo Stack Overflow, sendo assim possível responder a um conjunto de interrogações na forma mais eficiente. Outro objetivo deste projeto será a consolidação de conhecimentos adquiridos em outras unidades curriculares sendo a mais relevante nesta fase Programação Orientada a Objetos. O grande desafio deste projeto será a programação em larga escala, devido à elevada quantidade de dados e assim, com maior complexidade a nível algorítmico e de estruturas.

2 Descrição das Estruturas

A nível de estruturas foram usadas duas HashMaps, para armazenar os Users e as Tags, e uma TreeMap para armazenar os Posts. A TreeMap que armazena os posts está organizada pelos ID's dos posts, a HashMap dos Users está organizada pelos ID's dos Users e a HashMap das Tags pelo nome da tag. Decidimos usar estas estruturas de dados uma vez que o acesso aos dados é bastante mais rápido. Para conseguirmos fazer parse dos ficheiros que estão no formato XML foi necessário criar uma classe para o parser dos ficheiros. O parser é do tipo SAX, uma vez que uma das suas características é a sua elevada eficiência, pois não necessita de importar os documentos para a memória.

3 Interrogações

3.1 Interrogação 1

Esta interrogação procura o título do post e o nome de utilizador do autor, dado o identificador de um post. Caso o post for uma resposta, devem ser retornados o título e utilizador da pergunta correspondente.

Estratégia: Na árvore dos posts procuramos o post com o id fornecido. Verificamos se é pergunta ou resposta e, caso seja pergunta, buscamos o titulo e vamos à tabela de hash dos users obter o nome do autor do post. Caso seja resposta, buscamos o post ao qual está a responder e fazemos o mesmo.

3.2 Interrogação 2

Esta interrogação procura o top N de utilizadores com maior número de posts de sempre (tanto perguntas como respostas).

Estratégia: Na tabela dos users, percorremos os utilizadores e usamos um comparator para ordenar os utilizadores pelo número de posts, em ordem decrescente, pegando depois nos N primeiros. A esses N utilizadores com mais posts aplicamos o map de forma a obter os seus ids, retornando no fim como uma lista.

3.3 Interrogação 3

A interrogação número 3 pretende obter o número total de posts, durante um determinado intervalo de tempo, identificando perguntas e respostas separadamente.

Estratégia: Na árvore dos posts procuramos os posts que estão entre as datas dadas, percorrendo-os e incrementando os contadores das respostas e das perguntas consoante o tipo do post.

3.4 Interrogação 4

Esta interrogação procura todas as perguntas que contenham uma determinada tag, dado um intervalo de tempo arbitrário. Deve ser retornada uma lista com os IDs das perguntas ordenadas em cronologia inversa.

Estratégia: Na árvore dos posts vamos buscar os posts que estão entre as datas fornecidas que contém a tag em questão e aplicamos um sort utilizando um comparator que ordena pela data, da mais recente para a mais antiga. De seguida aplicamos um map para obtermos os ids dos posts, sendo retornados numa lista.

3.5 Interrogação 5

Nesta interrogação, dado o ID de um utilizador, pretende-se obter a informação do seu perfil e os IDs dos seus 10 últimos posts (perguntas ou respostas), ordenadas por cronologia inversa.

Estratégia: Através do id dado, obtemos o utilizador e com isso vamos buscar

a sua biografia. No Set que guarda os seus posts utilizamos o comparator que ordena pela data para buscar os 10 posts mais recentes, obtendo uma lista com os ids desses tais posts. Criamos depois um par com a biografia e a lista dos ids dos últimos 10 posts.

3.6 Interrogação 6

A interrogação número 6 procura os IDs das N respostas com mais votos, em ordem decrescente do número de votos, num dado intervalo de tempo arbitrário.

Estratégia: Na árvore dos posts procuramos as respostas que estão entre as datas fornecidas, ordenamos essas perguntas usando um comparator que ordena os posts pelos votos em ordem decrescente, ficando com os N primeiros, ou seja as N respostas com mais votos. Retornamos depois a lista dos ids dessas N respostas.

3.7 Interrogação 7

Nesta interrogação pretende-se devolver os IDs das N perguntas com mais respostas, em ordem decrescente do número de respostas, dado um intervalo de tempo arbitrário.

Estratégia: Na árvore dos posts vamos buscar as perguntas que estão entre as datas fornecidas ordenando-as usando um comparator que ordena os posts pelo número de respostas em ordem decrescente, ficando com os N primeiros. Assim é retornada a lista dos ids dessas N perguntas.

3.8 Interrogação 8

A interrogação número 8 pede o retorno de uma lista com os IDs de N perguntas cujos títulos contenham uma dada palavra, ordenados por cronologia inversa.

Estratégia: Percorremos os posts e buscamos as perguntas que contém a palavra dada no titulo. Aplicamos um sort utilizando um comparator que ordena pela data em cronologia inversa e retornamos a lista com os N primeiros ids.

3.9 Interrogação 9

Esta interrogação procura, dados os IDs de dois utilizadores, as últimas N perguntas(perguntas e respostas) em que participaram os dois utilizadores.

Estratégia: Vamos buscar as perguntas e respostas de cada utilizador que são guardadas em Sets de Posts. Para o caso do utilizador 1 responder ao utilizador 2, percorremos as respostas do utilizador 1, procuramos a pergunta correspondente e verificamos se esse post pertence às perguntas do utilizador 2. Para o caso contrário, utilizamos o mesmo método, mas agora com as repostas do utilizador 2 e as perguntas do utilizador 1. Para o caso de ambos responderem ao mesmo post, vamos buscar os ids das perguntas correspondentes às respostas dos dois utilizadores. Depois percorremos uma das listas com esses ids e vemos se existem ids em comum. Sempre que encontramos um post em que os dois utilizadores tenham participado adicionamos a pergunta a um TreeSet auxiliar

ordenando pela data e no fim obtemos os ids correspondentes, retornando a lista com os ids dos N posts mais recentes em que os dois participaram.

3.10 Interrogação 10

Nesta interrogação é pedido a melhor resposta dado o ID de um pergunta, usando-se a seguinte função de média : (Scr * 0.45) + (Rep * 0.25) + (Vot * 0.2) + (Comt * 0.1) onde Scr é o score da resposta, Rep é a reputação do utilizador, Vot é o número de votos recebidos pela resposta e Comt é o número de comentários recebidos pela resposta.

Estratégia: Vamos buscar o post em questão e criamos a lista com as respostas a essa pergunta. Para cada pergunta obtemos o seu valor, buscando a reputação do autor na tabela de hash, guardando sempre o id da resposta com o melhor valor. No fim é retornado esse id.

3.11 Interrogação 11

A interrogação 11 pede os identificadores das N tags mais usadas pelos N utilizadores com melhor reputação, dado um intervalo arbitrário de tempo.

Estratégia: Ordenamos os utilizadores usando um comparator que ordena pela reputação em valor decrescente e pegamos nos N utilizadores com maior reputação. Depois para cada utilizador percorremos os seus posts entre as datas dadas e guardamos as tags por eles utilizadas numa lista. Usamos outra lista para guardar as tags usadas sem repetidos(fazendo distinct()). Para cada tag na lista com as tags sem repetidos, contamos o número de ocorrências na lista que tem todas as tags, criando para cada tag um par com o seu nome e a sua contagem. Armazenamos esses pares noutra lista que é ordenada utilizando um comparator que ordena pela contagem do par em ordem decrescente, buscando assim as N tags mais usadas. Para essas N tags vamos buscar o seu id à tabela de hash onde guardamos as tags e os seu ids correspondentes.

4 Conclusão

Nesta fase do trabalho achamos que o nível de dificuldade foi menor, uma vez que as estruturas usadas foram muito semelhantes às usadas na fase de C e também porque tinhamos ao nosso dispor diversas API's de estruturas como Maps e Sets que nos pouparam bastante tempo e tornaram o processo de realização do projeto bastante mais acessível. Devido a alguns erros, a interrogação 10 não devolve os resultados esperados e nas interrogações 7 e 11 os dados resultantes não estão ordenados como era suposto. Com este projeto pudemos aplicar os conhecimentos obtidos na unidade curricular de Programação Orientada aos Objetos e ganhamos mais experiência no que toca a esta linguagem de programação.