Entrega Projeto UT2

Alunas: Laryssa Finizola Costa da Silva
Ludmila Maria Pereira da Silva

Nesta etapa de melhoria, os requisitos solicitados foram atendidos. Foi criado um pacote com nome de *estruturas* onde foi colocado as estruturas de dados utilizadas, sendo elas *ListaEncadeada<T>, Fila<T>, HashMapSimples.* As estruturas foram usadas da seguinte forma:

- ListaEncadeada:
  - Substituiu o uso direto de array durante a leitura do arquivo tweets_mentioned_persons.csv.
  - Foi implementada na classe ListaEncadeada<Tweet> e usada no método lerCSVComLista(String caminho), esse método está na classe *CSVUtils*.

```
public static ListaEncadeada<Tweet> lerCSVComLista(String caminho) {
    ListaEncadeada<Tweet> lista = new ListaEncadeada<>();

    try (BufferedReader reader = new BufferedReader(new FileReader(caminho))) {
        String linha = reader.readLine();
        while ((linha = reader.readLine()) != null) {
            String[] partes = linha.split(",(?=([^\"]*\"[^\"]*\")*[^\"]*$)", -1);
            if (partes.length < 8) {
                continue;
            }

            Tweet tweet = new Tweet(
                    partes[0], partes[1], partes[2], partes[3], partes[4], partes[5],
                    partes[6], Integer.parseInt(partes[7])
            );
            lista.adicionar(tweet);
        }
    } catch (IOException e) {
        e.printStackTrace();
    }

    return lista;
}
```

  - A lista encadeada vai resolver o problema da necessidade de saber previamente o tamanho da entrada para fazer o armazenamento dinamico

dos dados. Outra melhoria é que agora ela simula um ambiente real, onde normalmente os dados poderiam chegar de forma indefinida.

- Como justificativa, a complexidade de inserção foi reduzida de O(n) para O(1) com a implementação de um ponteiro para o último nó (o nome do nó é 'ultimo'):

```
public void adicionar(T valor) {
    No novo = new No(valor);
    if (inicio == null) {
        inicio = novo;
        ultimo = novo;
    } else {
        ultimo.proximo = novo;
        ultimo = novo;
    }
    tamanho++;
}
```

- Fila
  - Substituiu o laço *for* aninhado na classe Main, foi usada para processar e armazenar os pares de algoritmo e campo de ordenação, como 'jobs'.
  - O uso da Fila organiza e modulariza o processamento dos algoritmos de ordenação e facilita uma futura extensão para execução paralela. Abaixo segue uma chamada para a classe Fila:

```
// Fila de jobs de ordenação
Fila<OrdenacaoJob> fila = new Fila<>();

for (String campo : campos) {
    for (String algoritmo : algoritmos) {
        if (algoritmo.equals("countingSort") && !campo.equals("count")) continue; // Algor
        fila.enfileirar(new OrdenacaoJob(campo, algoritmo));
    }
}

while (!fila.estaVazia()) {
    System.out.println("Fila não está vazia");
    OrdenacaoJob job = fila.desenfileirar();
    System.out.println("Executando: " + job.getCampo() + " - " + job.getAlgoritmo());
    processarOrdenacao(base, job.getCampo(), job.getAlgoritmo());
}


System.out.println("Processamento finalizado!");
```

- A justificativa para o uso da fila é a implementação simples com operações de enfileiramento e desenfileiramento eficientes O(1). O uso de fila apoia o uso de estruturas clássicas de controle de fluxo.

- HashMap Simples
  - Foi implementado para ser usado no pré processamento do array de tweets na classe Main, para contar o número de tweets por usuário:

```
// Contagem de tweets por usuário usando HashMapSimples
HashMapSimples contagemUsuarios = new HashMapSimples();
for (Tweet tweet : base) {
    String user = tweet.getUser();
    Integer atual = contagemUsuarios.get(user);
    contagemUsuarios.put(user, atual == null ? 1 : atual + 1);
}
```

  - O uso do hashmap simples resolve o problema de contagem de tweets, tornando ela eficiente e com agrupamento de dados baseado em uma chave, no caso usuário. Simula uma análise de frequência, caso seja um trabalho futuro na disciplina o uso em relatórios ou filtros.
  - A justificativa para o hashmap simples é que ela contorna o uso do HashMap nativo do Java, usando hashing manual. Também é simples e eficiente, a média de acesso é O(1) em condições normais ou favoráveis.