UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
INF01203 – Estruturas de Dados
Profas. Renata Galante e Viviane Moreira
{galante,viviane}@inf.ufrgs.br

TRABALHO FINAL - *Twitter Analytics*

1 **Objetivo**

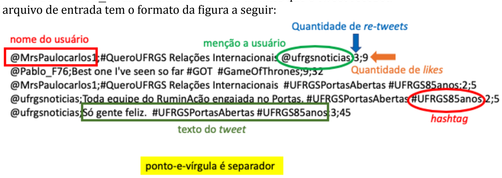
O objetivo do trabalho é definir uma estrutura de dados para avaliar métricas de usuários do *Twitter*, utilizando estruturas de dados vistas na disciplina.

2 **Especificação da Aplicação**

A entrada da aplicação é um arquivo do tipo texto CSV (veja Figura a seguir) no qual cada linha armazena dados de um *tweet*. As seguintes informações são representadas no arquivo, separadas por ponto-e-vírgula, onde:

- `user` – é o nome do usuário que escreveu o tweet.
- `text` – é texto do *tweet*. O texto pode conter:
  - o `menções`, representa os usuários que foram mencionados no *tweet*. Cada menção começa com o símbolo @
  - o `hashtags`, cada hashtag começa com o símbolo #
- `RT_count` – indica o número de *retweets* que o tweet recebeu
- `favorite_count` – indica o número de *likes* que o tweet recebeu

O arquivo de entrada tem o formato da figura a seguir:



ponto-e-vírgula é separador

Você deve propor uma estrutura de dados apropriada para armazenar o arquivo de entrada e executar as seguintes consultas:

Operação **a**. *Top hashtags*. Listar as *hastags* mais citadas em todo arquivo.

Operação **b**. *Top Ativos*. Listar os usuários que mais postam tweets.

Operação **c**. *Top retweets*. Listar os *tweets* com maior número de *retweets*.

Operação **d**. *Top menções*. Listar os usuários mais mencionados nos *tweets*.

Operação **e**. *Influenciadores*. Listar os usuários com maior número de retweets.

Operação **f**. *Engajamento*. Listar os usuários mais engajados da rede. O engajamento é contabilizado pela quantidade de interações dos tweets de um usuário, que inclui: gostar do *tweet* (*like*), quantidade de retweets e menções.

Operação **g**. *Termos Associados*. Listar as principais *hashtags* associadas a uma *hashtag* específica.

Tempo. Contabilizar o tempo de criação da estrutura de dados e da geração do arquivo de saídas.

A figura a seguir ilustra o fluxo de execução da aplicação.

Detalhamento das Operações

`OPERAÇÃO a.` Listar as *hastags* mais citadas em toda rede. A operação recebe como entrada um `número` que indica quantas *hashtags* devem ser listadas e quantidade de citações. Se o número for zero, todas as *hashtags* devem ser listadas. A saída da função é uma lista em ordem decrescente das *hastags* mais citadas. Restrições: *(i)* listar um resultado por linha; *(ii) hashtags* com a mesma frequência devem ser listadas em ordem alfabética.

`OPERAÇÃO b.` Listar os usuários que mais postam tweets. A operação recebe como entrada um `número` que indica quantos usuários devem ser listados. Se o número for zero, todos os usuários devem ser listados. A saída da função é uma lista em ordem decrescente de usuários mais ativos e o número de postagens. Restrições: *(i)* listar um resultado por linha.

`OPERAÇÃO c.` Listar os *tweets* com maior número de retweets. A operação recebe como entrada um `número` que indica quantos *tweets* devem ser listados. Se o número for zero, todos os *tweets* devem ser listados. A saída da função é uma lista em ordem decrescente de *tweets* mais "retuitados" e o número de *retweets*. Restrições: *(i)* listar um resultado por linha; *(ii) tweets* com a mesma frequência devem ser listados em ordem alfabética.

`OPERAÇÃO d.` Listar os usuários mais mencionados nos *tweets*. A operação recebe como entrada um `número` que indica quantos usuários devem ser listados. Se o número for zero, todos os usuários devem ser listados. A saída da função é uma lista em ordem decrescente de usuários mais mencionados. Restrições: *(i)* listar um resultado por linha; *(ii)* usuários com a mesma frequência devem ser listados em ordem alfabética.

`OPERAÇÃO e.` Listar os usuários mais influentes. Um usuário influente é aquele que possui o maior número de *retweets*. A operação recebe como entrada um `número` que indica quantos usuários devem ser listados. Se o número for zero, todos os usuários devem ser listados. A saída da função é uma lista em ordem decrescente pelo número de influência e o total de retweets. Restrições: *(i)* listar um resultado por linha; *(ii)* usuários com a mesma frequência devem ser listados em ordem alfabética.

`OPERAÇÃO f.` Listar os usuários mais engajados da rede. Engajamento é a interação do público com as postagens do usuário e determina o alcance das postagens. O engajamento é contabilizado pela somatório da quantidade de interações dos *tweets* de um usuário, que inclui: gostar do tweet (*like*), quantidade de *retweets* e quantidade de menções ao usuário. A operação recebe como entrada um `número` que indica quantos usuários devem ser listados. Se o número for zero, todos os usuários devem ser listados. A saída da função é uma lista em ordem decrescente de engajamento que deve exibir o usuário e o número de engajamento. Restrições: *(i)* listar um resultado por linha; *(ii)* usuários com a mesma frequência devem ser listados em ordem alfabética.

`OPERAÇÃO g.` Listar os termos associados a uma *hashtag*. A operação recebe como entrada uma *hashtag* e um `número` que indica quantas *hashtags* associadas devem ser listadas. Se o número for zero, todos as *hashtags* associadas devem ser listadas. A saída da função é uma lista em ordem decrescente do número de vezes que a *hashtag* aparece associada, seguida do número de vezes.. Restrições: *(i)* listar um resultado por linha; *(ii) hashtags* com a mesma frequência devem ser listadas em ordem alfabética.

`Tempo.` Deve ser contabilizado o tempo do carregamento da base de dados na estrutura e o tempo da geração do arquivo de saídas.

A seguir, um exemplo do arquivo de entrada 1 que contém os dados dos *tweets*.

```
@MrsPaulocarlos1;#QueroUFRGS Relações Internacionais @ufrgsnoticias;3;9
@Pablo_F76;Best one I've seen so far #GOT  #GameOfThrones;9;32
@MrsPaulocarlos1;#QueroUFRGS Relações Internacionais  #UFRGSPortasAbertas #UFRGS85anos;2;5
@ufrgsnoticias;Toda equipe do RuminAção engajada no Portas. #UFRGSPortasAbertas #UFRGS85anos;2;5
@ufrgsnoticias;Só gente feliz.  #UFRGSPortasAbertas #UFRGS85anos;3;45
```

A seguir, um exemplo do arquivo de entrada 2 que contém as operações que devem ser executadas.

```
a;0
b;0
c;2
d;2
e;3
f;2
g; #UFRGSPortasAbertas
```

A seguir, um exemplo do arquivo de saída tendo como entrada os dois arquivos anteriores.

```
--- OP A
#UFRGS85anos, 3
#UFRGSPortasAbertas, 3
#GameOfThrones, 2
#GOT, 2
--- OP B
@ufrgsnoticias, 2
@danibunny5, 1
@MrsPaulocarlos1, 1
@Pablo_F76, 1
--- OP C
Best one I've seen so far #GOT  #GameOfThrones, 9
this made my day #MadQueen  #GOT #GameOfThrones @Pablo_F76, 9
--- OP D
@ufrgsnoticias, 2
@Pablo_F76, 1
--- OP E
@danibunny5, 9
@Pablo_F76, 9
@ufrgsnoticias, 5
--- OP F
@ufrgsnoticias, 56
@danibunny5, 42
--- OP G
#UFRGS85anos, 3
#QueroUFRGS, 1
```

A aplicação **não é case sensitive.** Caracteres acentuados devem ser transformados para sua forma não acentuada (por exemplo: "Á" deve ser compreendido como "A").

Seu programa deverá ser chamado **a partir da linha de comando** (passando parâmetros para o `main`).

As entradas e saídas da sua aplicação são:

Entradas:
(i) o nome do arquivo de entrada
(ii) o nome arquivo com as operações

Saídas:
(i) arquivo com os resultados da execução do arquivo de operações sobre o arquivo das operações.

Exemplo de chamada: `C:\minhaaplicacao entrada.txt operacoes.txt saida.txt`

O Moodle contém um exemplo de arquivo de entrada, um arquivo de operações e sua saída. No dia da apresentação, **um novo arquivo** será fornecido.

3 **Requisitos**

- É necessário elaborar um relatório **detalhado.** Estrutura do relatório está no final desse documento.
- O trabalho deve ser feito, preferencialmente, em duplas. Também aceitaremos trabalhos feitos individualmente de duplas cujos integrantes sejam de turmas diferentes.
- A linguagem de programação aceita é C (Não é C++ nem C#).

6. Entrega e Apresentação

- apresentação (no horário da aula) e entrega pelo Moodle

7. Critérios de Avaliação

O trabalho deve ser realizado em duplas e deverá ser apresentado e defendido na data prevista.
Para a avaliação serão adotados diversos critérios:
O trabalho deve ser realizado em duplas e deverá ser apresentado e defendido na data prevista.
Para a avaliação serão adotados diversos critérios:

- funcionamento (Peso: 40%);
- organização e documentação do código (Peso: 30%); e

relatório (Peso: 30%).

8. Estrutura do Relatório

A documentação do programa é como um pequeno artigo que explica o que o programa faz, como faz, e apresenta conclusões obtidas sobre o trabalho. A documentação é um documento à parte e não deve ser escrita no programa fonte.

A documentação a ser entregue deve conter pelo menos:

- Descrição sucinta sobre o desenvolvimento do trabalho.
- Uma explicação sobre as decisões de implementação tomadas, uma visão geral do funcionamento do programa, comentários sobre os testes executados, etc.
- Descrição dos módulos e sua inter-dependência. Uma breve descrição de cada módulo bem como um diagrama, por exemplo, mostrando a relação de dependência entre eles. Note que esta parte certamente estará relacionada com os TADs.
- Descrição dos TADs e as estruturas de dados utilizadas.
- Uma explicação sobre os TADs definidos, as operações disponíveis e como os TADs são implementados. Você pode fazer essa descrição utilizando desenhos ou escrevendo.
- Descrição do formato de entrada e saída dos dados.
- Descrição sucinta dos testes realizados.

Importante:
Este trabalho deverá representar a solução da dupla para o problema proposto. O plágio é terminantemente proibido e a sua detecção incorrerá na divisão da nota obtida pelo número de alunos envolvidos. Para detectar o plágio, usaremos o software MOSS (http://theory.stanford.edu/~aiken/moss/).