

## Exercício Hands-on com Map Reduce usando Python

1. Considere como *dataset* o arquivo *movies.data*, disponível na pasta *data* do Box compartilhado com sua turma. Esse arquivo apresenta o seguinte layout (user\_id, movie\_id, rating, timestamp). Com base nisso, implemente o MapReduce em Python para fornecer uma lista de todos os filmes, ordenados de forma ascendente pela popularidade de cada um deles.

### Objetivo(s):

- Ordenar os filmes de forma ascendente pela popularidade.

### Estratégia:

- Map usando pares (movie\_id, 1);
- Reduce com saída: (rating\_count, movie\_id)
- Considere enviar a saída para um segundo Reducer que irá ordenar os filmes pelo rating\_count.

### Perguntas(s):

- a) Qual o filme de maior popularidade?
- b) Qual o filme menos popular?

2. Considere como *dataset* o arquivo *Marvel-graph.txt*, disponível na pasta *data* do Box compartilhado com sua turma. Esse arquivo apresenta em cada linha o ID de um dado herói, seguido dos IDs de seus amigos heróis. Com base nisso, implemente o MapReduce em Python para encontrar quem é o herói mais popular, isto é, quem deles tem mais amigos heróis.

### Objetivo(s):

- Encontrar o herói mais popular da Marvel.

### Estratégia:

- Map: contar o número de amigos de cada herói, por linha;
- Reduce: somar amigos por herói;
- Map: agregar o herói ao seu número de amigos;
- Reduce: encontrar o herói com máximo contador de amigos.

3. Considere como *dataset* o arquivo *movies.data*, disponível na pasta *data* do Box compartilhado com sua turma. Esse arquivo apresenta o seguinte layout (user\_id, movie\_id, rating, timestamp). Com base nisso, implemente o MapReduce em Python para fornecer uma lista dos DEZ filmes mais populares, ordenados de forma decrescente.

### Objetivo(s):

- Encontrar os DEZ filmes mais populares.

### Estratégia:

- Map usando pares (movie\_id, 1);
- Reduce com saída: (rating\_count, movie\_id)
- Considere enviar a saída para um segundo Reducer que irá ordenar os filmes e limitar-se aos mais populares.

