Prezado candidato.

Gostaríamos de fazer um teste que será usado para sabermos a sua proficiência nas habilidades para a vaga. O teste consiste em algumas perguntas e exercícios práticos sobre Spark e as respostas e códigos implementados devem ser armazenados no GitHub. O link do seu repositório deve ser compartilhado conosco ao final do teste.

Quando usar alguma referência ou biblioteca externa, informe no arquivo README do seu projeto. Se tiver alguma dúvida, use o bom senso e se precisar deixe isso registrado na documentação do projeto.

Qual o objetivo do comando **cache** em Spark?
É efetivo, para pequenos conjuntos de dados, onde é necessários armazená-los temporariamente, e reutilizá-los várias vezes, além de, melhorar a eficiência do código.

O mesmo código implementado em Spark é normalmente mais rápido que a implementação equivalente em MapReduce. Por quê?
Porque, em Spark a JVM permanece em constante execução em cada nó.

Qual é a função do **SparkContext**?
Opera como um cliente de execução, executar jobs.

Explique com suas palavras o que é **Resilient Distributed Datasets** (RDD).
É uma abstração de dados do Spark, além de de serem tolerante a falhas e imutáveis.

GroupByKey é menos eficiente que **reduceByKey** em grandes dataset. Por quê?
Porque com a reduceByKey, é realizada a operação passada como parâmetro de todos os elementos da mesma chave passada, obtendo um resultado parcial, gerando assim em um conjunto menor de dados a ser transferido.

Explique o que o código Scala abaixo faz.

```scala
val textFile = sc.textFile("hdfs://...")
val counts = textFile.flatMap(line => line.split(" "))
             .map(word => (word, 1))
             .reduceByKey(_ + _)
counts.saveAsTextFile("hdfs://...")
```

Linha: 1- Leitura do arquivo texto.
Linha: 2- Cada linha é quebrada em uma sequência de palavras, transformadas em uma única coleção de palavras.
Linha: 3- Cada palavra é transformada em um mapeamento de chave - valor.
Linha: 4- Valores agregados por chaves (+).
Linha: 5- Contagem de cada palavra é salvo em um arquivo de texto.

HTTP requests to the NASA Kennedy Space Center WWW server

Fonte oficial do dateset: http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html

Dados:

- Jul 01 to Jul 31, ASCII format, 20.7 MB gzip compressed, 205.2 MB.
- Aug 04 to Aug 31, ASCII format, 21.8 MB gzip compressed, 167.8 MB.

Sobre o dataset: Esses dois conjuntos de dados possuem todas as requisições HTTP para o servidor da NASA Kennedy Space Center WWW na Flórida para um período específico.

Os logs estão em arquivos ASCII com uma linha por requisição com as seguintes colunas:

- **Host fazendo a requisição**. Um hostname quando possível, caso contrário o endereço de internet se o nome não puder ser identificado.
- **Timestamp** no formato "DIA/MÊS/ANO:HH:MM:SS TIMEZONE"
- **Requisição (entre aspas)**
- **Código do retorno HTTP**
- **Total de bytes retornados**

Questões

Responda as seguintes questões devem ser desenvolvidas em Spark utilizando a sua linguagem de preferência.

1. Número de hosts únicos.
2. O total de erros 404.
3. Os 5 URLs que mais causaram erro 404.
4. Quantidade de erros 404 por dia.
5. O total de bytes retornados.