**Qual​ ​o​ ​objetivo​ ​do​ ​comando​ ​​cache​ ​​em​ ​Spark?**

*Esse comando permite incluir datasets dentro de um cache de um cluster em memória para que possa ser utilizado por outras ações.*

**O​ ​mesmo​ ​código​ ​implementado​ ​em​ ​Spark​ ​é​ ​normalmente​ ​mais​ ​rápido​ ​que​ ​a​ ​implementação​ ​equivalente​ ​em MapReduce. ​ ​Por​ ​quê?**

*Por que todo o acesso e trafego dos dados é feito em memória. Diferente do MapReduce que faz acesso ao disco. Além disso o código no Spark é executado uma única fez e quando necessário invés de a cada iteração.*

**Qual​ ​é​ ​a​ ​função​ ​do​ ​​SparkContext​?**

*Liga o python ao kernel do spark permitindo a execução de todo o código pyspark quando solicitado.*

**Explique​ ​com​ ​suas​ ​palavras​ ​​ ​o​ ​que​ ​é​ ​​Resilient​ ​Distributed​ ​Datasets​​ ​ (RDD).**

*Se trata de um dataset divididos em várias partições e distribuídos entre os nós de um cluster.*

**GroupByKey​ ​​é​ ​menos​ ​eficiente​ ​que​ ​​reduceByKey​ ​​em​ ​grandes​ ​dataset. ​ ​Por​ ​quê?**

*O GroupByKey distribui as chaves e valores por toda rede e os workers disponíveis enquanto o reduceByKey combina uma saída com uma chave única em cada partição antes de distribuir os dados.*

**Explique​ ​o​ ​que​ ​o​ ​código​ ​Scala​ ​abaixo​ ​faz:**

*val​​ ​​textFile​​ ​​=​​ ​​sc​.​textFile​(​"hdfs://..."​) 🡪 Carrega arquivo do HDFS*

*val​​ ​​counts​​ ​​=​​ ​​textFile​.​flatMap​(​line​​ ​​=>​​ ​​line​.​split​(​"​ ​"​)) ​ 🡪 faz a quebra da ​​linha pelos espaço ​​ ​​ ​​ ​​ ​​ ​​ ​​ ​​ ​​ ​​ ​​ ​​ ​​ ​​ ​​ ​​.​map​(​word​​ ​​=>​​ ​​(​word​,​​ ​​1​)) 🡪 atribui 1 para cada palavra encontrada ​ ​​ ​​ ​​ ​​ ​​ ​​ ​​ ​​ ​​ ​​ ​​ ​​ ​​ ​​ ​​ ​​ ​​.​reduceByKey​(​\_​​ ​​+​​ ​​\_​) 🡪 realiza a redução p/ computar a quantidade de cada palavra*

*counts​.​saveAsTextFile​(​"hdfs://..."​) 🡪 grava o resultado em um arquivo do hdfs*