

Atividade de Reposição da Segunda Aula - Módulo 3

Entrega 24 ago em 23:59 **Pontos** 10 **Perguntas** 7

Disponível 19 ago em 17:00 - 24 ago em 23:59 5 dias

Limite de tempo Nenhum

Histórico de tentativas

	Tentativa	Tempo	Pontuação
MAIS RECENTE	Tentativa 1	450 minutos	8,58 de 10

⚠ As respostas corretas estarão disponíveis em 24 ago em 23:59.

Pontuação deste teste: **8,58** de 10

Enviado 20 ago em 21:18

Esta tentativa levou 450 minutos.

Pergunta 1

1,42 / 1,42 pts

Sobre os métodos da classe *JobConf*, podemos afirmar:

☐

O método `setJobName` permite que o usuário defina um nome para as classes Mapper e Reducer.

☒

O método `setJobName` permite que o usuário defina um nome para o job MapReduce/Hadoop que será executado.

☐

O método `setJobName` permite que o usuário defina um nome para a instância da classe principal.

☐

O método `setJobName` permite que o usuário defina um nome para a instância da classe *JobConf*.

Incorreta

Pergunta 2

0 / 1,42 pts

Sobre o uso da classe *MapReduceBase* podemos afirmar, EXCETO:

- ☒ Possui apenas dois métodos: Configure e Close.
- ☐ A classe *MapReduceBase* não possui métodos específicos para as fases Map e Reduce.
- ☐ Possui quatro métodos: Configure, Close, Map e Reduce.
- ☐ É a classe base para implementação dos Mappers e Reducers.

Esse assunto está disponível na Aula 4.2.

Pergunta 3

1,42 / 1,42 pts

A classe *JobConf* é a interface primária onde o usuário descreve um *jobMapReduce* / Hadoop. Sobre o *JobConf*, podemos afirmar:

- ☐ O *JobConf* serve apenas para definir o nome do job.
- ☐ O *JobConf* é extremamente importante para a execução do job *MapReduce/Hadoop*, entretanto ele não pode ser acessado dentro das classes *Mapper* e *Reducer*.
- ☒ O *JobConf* é um objeto que une todas as informações necessárias para a execução do job, tais como: quais as classes *Mapper* e *Reducer* que foram definidas pelo desenvolvedor, o nome do job etc.



O JobConf pode receber a atribuição das classes Mapper e Reducer, porém não pode atribuir valores a nenhum parâmetro de configuração do Hadoop.

Pergunta 4

1,42 / 1,42 pts

Sobre a estrutura de um programa Apache Hadoop em Java, podemos afirmar:



A estrutura básica de um programa Apache Hadoop deve ser formada obrigatoriamente por uma classe principal, uma classe Mapper, uma classe Reducer e uma classe Combiner.



A estrutura básica de um programa Apache Hadoop pode ser formada por uma classe principal, uma classe Mapper e uma classe Reducer.



O método main(...) está presente na classe Mapper.



A estrutura básica de um programa Apache Hadoop deve obrigatoriamente conter uma classe Combiner.

Pergunta 5

1,42 / 1,42 pts

As interfaces *Mapper* e *Reducer* são de fundamental importância para a construção dos programas no Hadoop. Sobre essas interfaces é CORRETO afirmar:



A interface Mapper é responsável por fornecer o método `map()`, onde os desenvolvedores poderão criar sua função personalizada.



A interface Mapper é responsável por fornecer o método `map()`, porém os desenvolvedores não têm liberdade para escrever sua própria função personalizada.



A interface Reducer é responsável por fornecer tanto a função Map quanto a função Reduce. É facultado ao desenvolvedor implementar suas próprias funções Map e Reduce personalizadas.



As funções Map e Reduce não estão disponíveis nas interfaces Mapper e Reducer, e sim na classe `MapReduceBase`.

Pergunta 6

1,42 / 1,42 pts

Os métodos *`setMapperClass`*, *`setReducerClass`* e *`setCombinerClass`* fazem parte da classe *`JobConf`* e são muito importantes para a execução do *job* MapReduce/Hadoop. Sobre esses métodos, podemos afirmar:



Os três métodos são utilizados para informar ao job quais classes foram definidas como responsáveis pelas funções Map, Reduce e eventual Combiner.



Os três métodos não fazem parte da classe *`JobConf`*.



Para o efetivo funcionamento de qualquer job MapReduce/Hadoop, os três métodos devem ser obrigatoriamente utilizados.



O método `setCombinerClass` deve possuir obrigatoriamente uma classe diferente de `setReducerClass`.

Pergunta 7

1,48 / 1,48 pts

Sobre a manipulação de diretórios no sistema de arquivos distribuído do Hadoop (HDFS) podemos afirmar:



A classe `FileSystem` é específica para implementação das funções Map e Reduce, e não fornece nenhum método para manipulação de diretórios no HDFS.



Podemos manipular os diretórios do HDFS somente usando linhas de comando no sistema operacional.



Podemos manipular os diretórios usando linhas de comando no sistema operacional ou pela nossa aplicação, por meio da classe `FileSystem`.



Podemos manipular os diretórios do HDFS apenas através dos métodos da classe `FileSystemInOut`.

Pontuação do teste: **8,58** de 10