

Tutorial de implementação de aplicação MapReduce

PROFESSORA

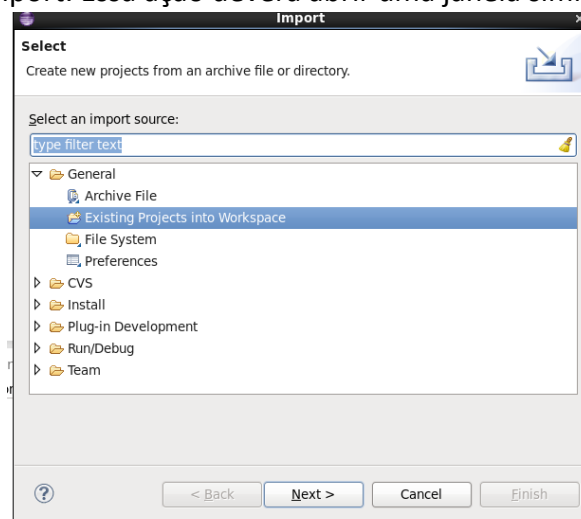
Rosângela de Fátima Pereira – rpereira@larc.usp.br

DESCRIÇÃO

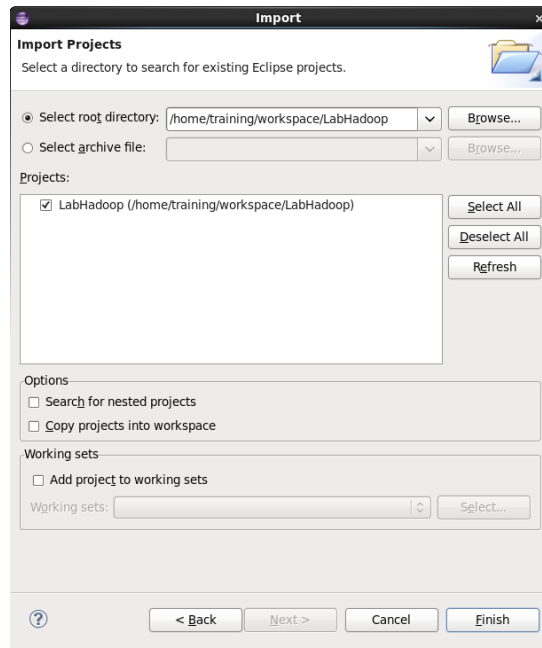
Esse documento descreve os passos necessários para a implementação e execução de uma aplicação MapReduce em um ambiente Hadoop

ATIVIDADES

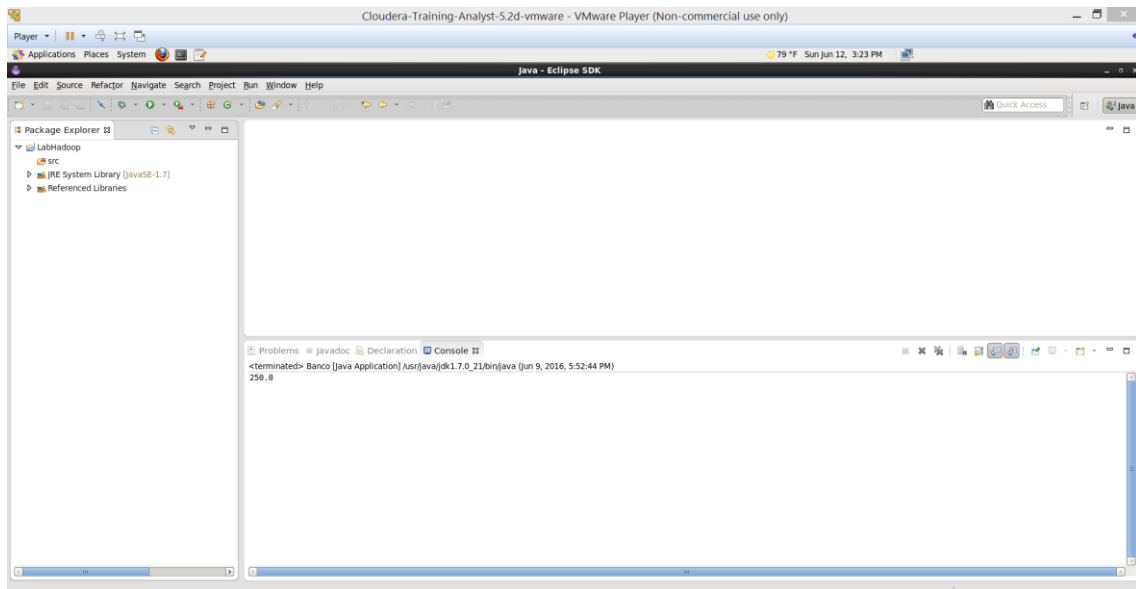
1. Abra a IDE Eclipse
2. Selecione a opção File>>Import. Essa ação deverá abrir uma janela similar à Figura a seguir:



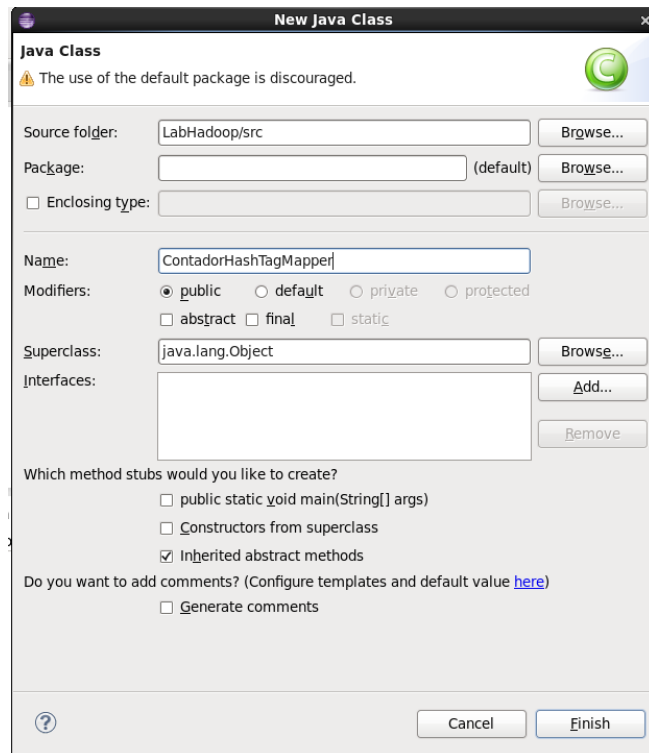
3. Selecione a opção “Existing Projects into Workspace” e pressione o botão Next. Deverá aparecer a seguinte janela:



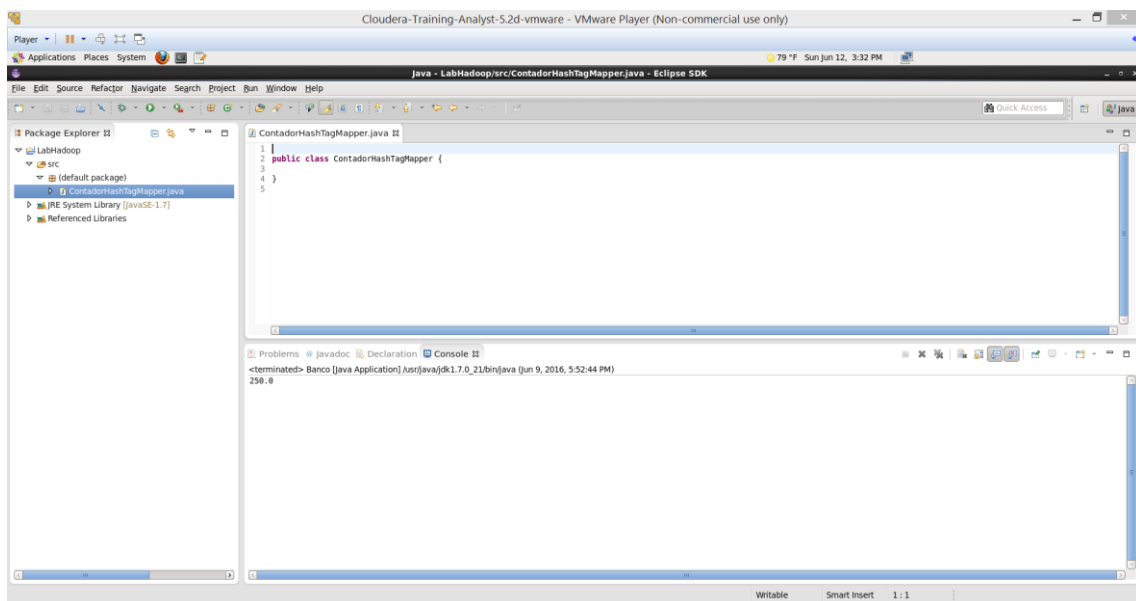
4. No campo “Select root directory” indique o seguinte caminho:
/home/training/workspace/LabHadoop
Feito isso, pressione o botão Finish. O projeto deverá aparecer conforme imagem a seguir:



5. Agora deverá ser criada a classe ContadorHashTagMapper. Para isso, clique com o botão direito do mouse sobre o nome do projeto e selecione a opção: New >> Class. Deverá aparecer a janela similar à imagem a seguir:



6. No campo “Name”, indique a classe ContadorHashTagMapper. Após isso, pressione o botão Finish. Deverá aparecer a seguinte imagem:



7. Na classe ContadorHashTagMapper, inserir o código a seguir:

```
import java.io.*;
import org.apache.hadoop.conf.Configured;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
```

```

import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;

public class ContadorHashTagMapper extends Mapper<Object, Text, Text, IntWritable> {

    private final static IntWritable numeroUm = new IntWritable(1);
    private Text palavra = new Text();
    private String hashTagExpRegular = "(?:\\s|\\A|^)[##]+([A-Za-z0-9- _]+)";

    public void map(Object key, Text value, Context context) throws IOException, InterruptedException {

        String[] palavras = value.toString().split(" ");

        for (String str : palavras) {
            if (str.matches(hashTagExpRegular)) {
                palavra.set(str);
                context.write(palavra, numeroUm);
            }
        }
    }
}

```

8. Repita o mesmo procedimento da Etapa 5, para criar a classe ContadorHashTagReducer, inserindo o código a seguir:

```

import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class ContadorHashTagReducer extends Reducer<Text,IntWritable,Text,IntWritable> {

    public void reduce(Text key, Iterable<IntWritable> values, Context context
        ) throws IOException, InterruptedException {

        int soma = 0;
        for (IntWritable val : values) {
            soma += val.get();
        }
        context.write(key, new IntWritable(soma));
    }
}

```

9. Crie agora a classe ContadorHashTagDriver, inserindo o código a seguir:

```

import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.conf.Configured;
import org.apache.hadoop.conf.Configuration;

```

```

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class ContadorHashTagDriver {

    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf);

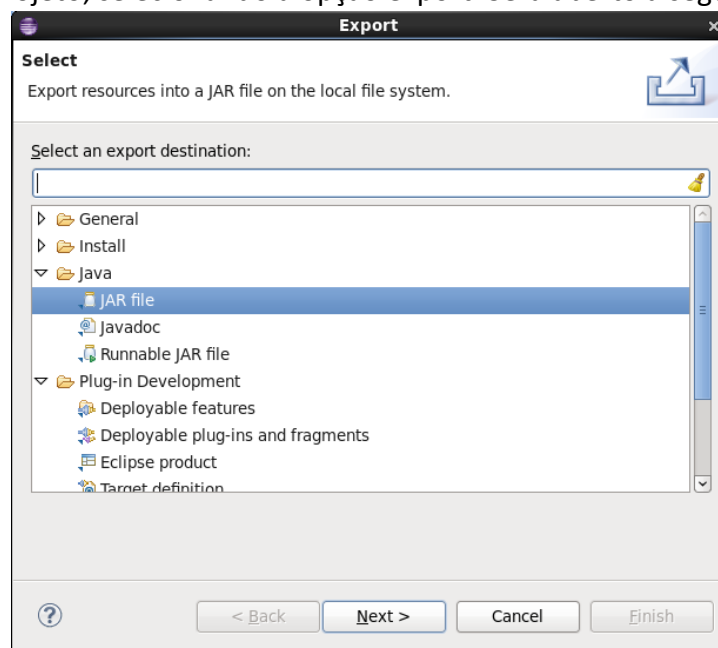
        job.setJarByClass(ContadorHashTagDriver.class);
        job.setMapperClass(ContadorHashTagMapper.class);
        job.setReducerClass(ContadorHashTagReducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);

        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));

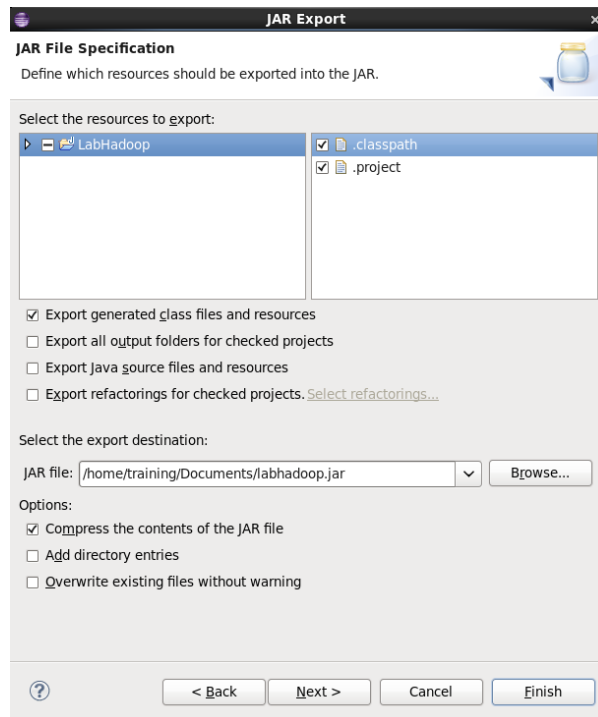
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}

```

10. Tendo implementado todas as classes, gere um arquivo JAR clicando com o botão direito do mouse no nome do projeto, selecionando a opção export. Será aberto a seguinte janela:



11. Selecione a opção JAR file e pressione o botão “Next”. Deverá aparecer a seguinte janela:



12. No campo JAR file, indique o seguinte caminho: /home/training/Documents/labhadoop.jar e pressione o botão Finish. Essa ação criará o jar do job implementado.

13. Para executar o job, abra um terminal e execute o seguinte comando:

```
$ hadoop jar ~/Documents/labhadoop.jar ContadorHashTagDriver input_cp output_contador
```

14. Após o job ter sido executado, verifique os arquivos de saída gerados por meio do seguinte comando:

```
$ hadoop fs -ls output_contador
```

15. Após o job ter sido executado, verifique os arquivos de saída gerados por meio do seguinte comando:

```
$ hadoop fs -cat output_contador/part-r-00000
```

Parabéns! Você concluiu as etapas de implementação e execução de uma aplicação MapReduce.