**ATP – Fundamentos de Big Data**

**Aluno: Stephan Kotarski**

**Professor: Adilson Galiano Filho**

**Código Fonte e Resultados Obtidos.**

1. **País com a maior quantidade de transações comerciais efetuadas.**

**Código Fonte:**

**package com.atp.atp;**

**import java.io.IOException;**

**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;**

**import org.apache.hadoop.mapreduce.Reducer;**

**import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;**

**import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;**

**/\*\***

**\***

**\* @author stephan.kotarsky**

**\*/**

**public class Informacao1 {**

**public static class MapperInformacao1 extends Mapper<Object, Text, Text, IntWritable>{**

**@Override**

**public void map(Object chave, Text valor, Context context) throws IOException, InterruptedException{**

**String linha = valor.toString();**

**String[] campos = linha.split(";");**

**if(campos.length == 10){**

**String pais = campos[0];**

**int quantidade = 1;**

**Text chaveMap = new Text(pais);**

**IntWritable valorMap = new IntWritable(quantidade);**

**context.write(chaveMap, valorMap);**

**}**

**}**

**}**

**public static class ReducerInformacao1 extends Reducer<Text, IntWritable, Text, IntWritable>{**

**@Override**

**public void reduce(Text chave, Iterable<IntWritable> valores, Context context) throws IOException, InterruptedException{**

**int soma = 0;**

**for(IntWritable val : valores){**

**soma += val.get();**

**}**

**IntWritable valorSaida = new IntWritable(soma);**

**context.write(chave, valorSaida);**

**}**

**}**

**public static void main(String[] args) throws IOException, InterruptedException, ClassNotFoundException{**

**String arquivoEntrada = "/home2/ead2021/SEM1/stephan.kotarsky/Desktop/stephan.kotarski/base\_inteira.csv";**

**String arquivoSaida = "/home2/ead2021/SEM1/stephan.kotarsky/Desktop/stephan.kotarski/Informacao1";**

**if(args.length == 2){**

**arquivoEntrada = args[0];**

**arquivoSaida = args[1];**

**}**

**Configuration conf = new Configuration();**

**Job job = Job.getInstance(conf, "atividade1");**

**job.setJarByClass(Informacao1.class);**

**job.setMapperClass(MapperInformacao1.class);**

**job.setReducerClass(ReducerInformacao1.class);**

**job.setOutputKeyClass(Text.class);**

**job.setOutputValueClass(IntWritable.class);**

**FileInputFormat.addInputPath(job, new Path(arquivoEntrada));**

**FileOutputFormat.setOutputPath(job, new Path(arquivoSaida));**

**job.waitForCompletion(true);**

**}**

**}**

**Resposta: Australia 307522**

1. **Mercadoria com a maior quantidade de transações comerciais no Brasil (como a base de dados está em inglês, utilize Brazil)**

**Código Fonte:**

**package com.atp.atp;**

**import java.io.IOException;**

**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;**

**import org.apache.hadoop.mapreduce.Reducer;**

**import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;**

**import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;**

**/\*\***

**\***

**\* @author stephan.kotarsky**

**\*/**

**public class Informacao2 {**

**public static class MapperInformacao2 extends Mapper<Object, Text, Text, IntWritable>{**

**@Override**

**public void map(Object chave, Text valor, Context context) throws IOException, InterruptedException{**

**String linha = valor.toString();**

**String[] campos = linha.split(";");**

**if(campos.length == 10){**

**if(campos[0].equals("Brazil")){**

**String mercadoria = campos[3];**

**int quantidade = 1;**

**Text chaveMap = new Text(mercadoria);**

**IntWritable valorMap = new IntWritable(quantidade);**

**context.write(chaveMap, valorMap);**

**}**

**}**

**}**

**}**

**public static class ReducerInformacao2 extends Reducer<Text, IntWritable, Text, IntWritable>{**

**@Override**

**public void reduce(Text chave, Iterable<IntWritable> valores, Context context) throws IOException, InterruptedException{**

**int soma = 0;**

**for(IntWritable val : valores){**

**soma += val.get();**

**}**

**IntWritable valorSaida = new IntWritable(soma);**

**context.write(chave, valorSaida);**

**}**

**}**

**public static void main(String[] args) throws IOException, InterruptedException, ClassNotFoundException{**

**String arquivoEntrada = "/home2/ead2021/SEM1/stephan.kotarsky/Desktop/stephan.kotarski/base\_inteira.csv";**

**String arquivoSaida = "/home2/ead2021/SEM1/stephan.kotarsky/Desktop/stephan.kotarski/Informacao2";**

**if(args.length == 2){**

**arquivoEntrada = args[0];**

**arquivoSaida = args[1];**

**}**

**Configuration conf = new Configuration();**

**Job job = Job.getInstance(conf, "atividade2");**

**job.setJarByClass(Informacao2.class);**

**job.setMapperClass(MapperInformacao2.class);**

**job.setReducerClass(ReducerInformacao2.class);**

**job.setOutputKeyClass(Text.class);**

**job.setOutputValueClass(IntWritable.class);**

**FileInputFormat.addInputPath(job, new Path(arquivoEntrada));**

**FileOutputFormat.setOutputPath(job, new Path(arquivoSaida));**

**job.waitForCompletion(true);**

**}**

**}**

**Resposta: Dentifrices 59**

**3. Quantidade de transações comerciais realizadas por ano.**

**Código Fonte:**

**package com.atp.atp;**

**import java.io.IOException;**

**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;**

**import org.apache.hadoop.mapreduce.Reducer;**

**import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;**

**import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;**

**/\*\***

**\***

**\* @author stephan.kotarsky**

**\*/**

**public class Informacao3 {**

**public static class MapperInformacao3 extends Mapper<Object, Text, Text, IntWritable>{**

**@Override**

**public void map(Object chave, Text valor, Context context) throws IOException, InterruptedException{**

**String linha = valor.toString();**

**String[] campos = linha.split(";");**

**if(campos.length == 10){**

**String ano = campos[1];**

**int quantidade = 1;**

**Text chaveMap = new Text(ano);**

**IntWritable valorMap = new IntWritable(quantidade);**

**context.write(chaveMap, valorMap);**

**}**

**}**

**}**

**public static class ReducerInformacao3 extends Reducer<Text, IntWritable, Text, IntWritable>{**

**@Override**

**public void reduce(Text chave, Iterable<IntWritable> valores, Context context) throws IOException, InterruptedException{**

**int soma = 0;**

**for(IntWritable val : valores){**

**soma += val.get();**

**}**

**IntWritable valorSaida = new IntWritable(soma);**

**context.write(chave, valorSaida);**

**}**

**}**

**public static void main(String[] args) throws IOException, InterruptedException, ClassNotFoundException{**

**String arquivoEntrada = "/home2/ead2021/SEM1/stephan.kotarsky/Desktop/stephan.kotarski/base\_inteira.csv";**

**String arquivoSaida = "/home2/ead2021/SEM1/stephan.kotarsky/Desktop/stephan.kotarski/Informacao3";**

**if(args.length == 2){**

**arquivoEntrada = args[0];**

**arquivoSaida = args[1];**

**}**

**Configuration conf = new Configuration();**

**Job job = Job.getInstance(conf, "atividade3");**

**job.setJarByClass(Informacao3.class);**

**job.setMapperClass(MapperInformacao3.class);**

**job.setReducerClass(ReducerInformacao3.class);**

**job.setOutputKeyClass(Text.class);**

**job.setOutputValueClass(IntWritable.class);**

**FileInputFormat.addInputPath(job, new Path(arquivoEntrada));**

**FileOutputFormat.setOutputPath(job, new Path(arquivoSaida));**

**job.waitForCompletion(true);**

**}**

**}**

**Resposta: 2009 373939**

**2007 374327**

**2011 374596**

**2012 377343**

**2006 378065**

**4. Mercadoria com maior quantidade de transações financeiras.**

**Código Fonte:**

package com.atp.atp;

import java.io.IOException;

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;

import org.apache.hadoop.mapreduce.Reducer;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

/\*\*

\*

\* @author stephan.kotarsky

\*/

public class Informacao4 {

public static class MapperInformacao4 extends Mapper<Object, Text, Text, IntWritable>{

@Override

public void map(Object chave, Text valor, Context context) throws IOException, InterruptedException{

String linha = valor.toString();

String[] campos = linha.split(";");

if(campos.length == 10){

String mercadoria = campos[3];

int quantidade = 1;

Text chaveMap = new Text(mercadoria);

IntWritable valorMap = new IntWritable(quantidade);

context.write(chaveMap, valorMap);

}

}

}

public static class ReducerInformacao4 extends Reducer<Text, IntWritable, Text, IntWritable>{

@Override

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

int soma = 0;

for(IntWritable val : valores){

soma += val.get();

}

IntWritable valorSaida = new IntWritable(soma);

context.write(chave, valorSaida);

}

}

public static void main(String[] args) throws IOException, InterruptedException, ClassNotFoundException{

String arquivoEntrada = "/home2/ead2021/SEM1/stephan.kotarsky/Desktop/stephan.kotarski/base\_inteira.csv";

String arquivoSaida = "/home2/ead2021/SEM1/stephan.kotarsky/Desktop/stephan.kotarski/Informacao4";

if(args.length == 2){

arquivoEntrada = args[0];

arquivoSaida = args[1];

}

Configuration conf = new Configuration();

Job job = Job.getInstance(conf, "atividade4");

job.setJarByClass(Informacao4.class);

job.setMapperClass(MapperInformacao4.class);

job.setReducerClass(ReducerInformacao4.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

FileInputFormat.addInputPath(job, new Path(arquivoEntrada));

FileOutputFormat.setOutputPath(job, new Path(arquivoSaida));

job.waitForCompletion(true);

}

}

**RESPOSTA: Matches 6239**

**5. Mercadoria com maior quantidade de transações financeiras em 2016.**

**Código Fonte:**

package com.atp.atp;

import java.io.IOException;

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;

import org.apache.hadoop.mapreduce.Reducer;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

/\*\*

\*

\* @author stephan.kotarsky

\*/

public class Informacao5 {

public static class MapperInformacao5 extends Mapper<Object, Text, Text, IntWritable>{

@Override

public void map(Object chave, Text valor, Context context) throws IOException, InterruptedException{

String linha = valor.toString();

String[] campos = linha.split(";");

if(campos.length == 10){

if(campos[1].equals("2016")){

String mercadoria = campos[3];

int quantidade = 1;

Text chaveMap = new Text(mercadoria);

IntWritable valorMap = new IntWritable(quantidade);

context.write(chaveMap, valorMap);

}

}

}

}

public static class ReducerInformacao5 extends Reducer<Text, IntWritable, Text, IntWritable>{

@Override

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

int soma = 0;

for(IntWritable val : valores){

soma += val.get();

}

IntWritable valorSaida = new IntWritable(soma);

context.write(chave, valorSaida);

}

}

public static void main(String[] args) throws IOException, InterruptedException, ClassNotFoundException{

String arquivoEntrada = "/home2/ead2021/SEM1/stephan.kotarsky/Desktop/stephan.kotarski/base\_inteira.csv";

String arquivoSaida = "/home2/ead2021/SEM1/stephan.kotarsky/Desktop/stephan.kotarski/Informacao5";

if(args.length == 2){

arquivoEntrada = args[0];

arquivoSaida = args[1];

}

Configuration conf = new Configuration();

Job job = Job.getInstance(conf, "atividade5");

job.setJarByClass(Informacao5.class);

job.setMapperClass(MapperInformacao5.class);

job.setReducerClass(ReducerInformacao5.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

FileInputFormat.addInputPath(job, new Path(arquivoEntrada));

FileOutputFormat.setOutputPath(job, new Path(arquivoSaida));

job.waitForCompletion(true);

}

}

**Resposta: Matches 201**

**6. Mercadoria com maior quantidade de transações financeiras em 2016, no Brasil (como a base de dados está em inglês, utilize Brazil).**

**Código Fonte:**

package com.atp.atp;

import java.io.IOException;

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;

import org.apache.hadoop.mapreduce.Reducer;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

/\*\*

\*

\* @author stephan.kotarsky

\*/

public class Informacao6 {

public static class MapperInformacao6 extends Mapper<Object, Text, Text, IntWritable>{

@Override

public void map(Object chave, Text valor, Context context) throws IOException, InterruptedException{

String linha = valor.toString();

String[] campos = linha.split(";");

if(campos.length == 10){

if(campos[1].equals("2016") && campos[0].equals("Brazil")){

String mercadoria = campos[3];

int quantidade = 1;

Text chaveMap = new Text(mercadoria);

IntWritable valorMap = new IntWritable(quantidade);

context.write(chaveMap, valorMap);

}

}

}

}

public static class ReducerInformacao6 extends Reducer<Text, IntWritable, Text, IntWritable>{

@Override

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

int soma = 0;

for(IntWritable val : valores){

soma += val.get();

}

IntWritable valorSaida = new IntWritable(soma);

context.write(chave, valorSaida);

}

}

public static void main(String[] args) throws IOException, InterruptedException, ClassNotFoundException{

String arquivoEntrada = "/home2/ead2021/SEM1/stephan.kotarsky/Desktop/stephan.kotarski/base\_inteira.csv";

String arquivoSaida = "/home2/ead2021/SEM1/stephan.kotarsky/Desktop/stephan.kotarski/Informacao6";

if(args.length == 2){

arquivoEntrada = args[0];

arquivoSaida = args[1];

}

Configuration conf = new Configuration();

Job job = Job.getInstance(conf, "atividade6");

job.setJarByClass(Informacao6.class);

job.setMapperClass(MapperInformacao6.class);

job.setReducerClass(ReducerInformacao6.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

FileInputFormat.addInputPath(job, new Path(arquivoEntrada));

FileOutputFormat.setOutputPath(job, new Path(arquivoSaida));

job.waitForCompletion(true);

}

}

**Resposta: Sodium 2**

**Tankers 2**

**Whey 2**

**Whiskies 2**

**Yogurt 2**

**7. Mercadoria com maior total de peso, de acordo com todas as transações comerciais.**

**Código Fonte:**

package com.atp.atp;

import java.io.IOException;

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;

import org.apache.hadoop.mapreduce.Reducer;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

/\*\*

\*

\* @author stephan.kotarsky

\*/

public class Informacao7 {

public static class MapperInformacao7 extends Mapper<Object, Text, Text, IntWritable>{

@Override

public void map(Object chave, Text valor, Context context) throws IOException, InterruptedException{

String linha = valor.toString();

String[] campos = linha.split(";");

if(campos.length == 10){

String mercadoria = campos[3];

String peso = campos[6];

IntWritable valorMap = new IntWritable(0);

Text chaveMap = new Text(mercadoria);

try {

valorMap = new IntWritable(Integer.parseInt(peso));

} catch(NumberFormatException e){

} finally{

}

context.write(chaveMap, valorMap);

}

}

}

public static class ReducerInformacao7 extends Reducer<Text, IntWritable, Text, IntWritable>{

@Override

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

long soma = 0;

for(IntWritable val : valores){

soma += val.get();

}

IntWritable valorSaida = new IntWritable(0);

try {

valorSaida.set(Integer.parseInt(String.valueOf(soma)));

} catch(NumberFormatException e){

} finally {

}

context.write(chave, valorSaida);

}

}

public static void main(String[] args) throws IOException, InterruptedException, ClassNotFoundException{

String arquivoEntrada = "/home2/ead2021/SEM1/stephan.kotarsky/Desktop/stephan.kotarski/base\_inteira.csv";

String arquivoSaida = "/home2/ead2021/SEM1/stephan.kotarsky/Desktop/stephan.kotarski/Informacao7";

if(args.length == 2){

arquivoEntrada = args[0];

arquivoSaida = args[1];

}

Configuration conf = new Configuration();

Job job = Job.getInstance(conf, "atividade7");

job.setJarByClass(Informacao7.class);

job.setMapperClass(MapperInformacao7.class);

job.setReducerClass(ReducerInformacao7.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

FileInputFormat.addInputPath(job, new Path(arquivoEntrada));

FileOutputFormat.setOutputPath(job, new Path(arquivoSaida));

job.waitForCompletion(true);

}

}

Resposta: Mate 2134533884

**8. Mercadoria com maior total de peso, de acordo com todas as transações comerciais, separadas por ano**

**Código Fonte:**

package com.atp.atp;

import java.io.IOException;

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;

import org.apache.hadoop.mapreduce.Reducer;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

/\*\*

\*

\* @author stephan.kotarsky

\*/

public class Informacao8 {

public static class MapperInformacao8 extends Mapper<Object, Text, Text, IntWritable>{

@Override

public void map(Object chave, Text valor, Context context) throws IOException, InterruptedException{

String linha = valor.toString();

String[] campos = linha.split(";");

if(campos.length == 10){

String mercadoria = campos[3];

String peso = campos[6];

String ano = campos[1];

IntWritable valorMap = new IntWritable(0);

Text chaveMap = new Text(mercadoria + " - " + ano);

try {

valorMap = new IntWritable(Integer.parseInt(peso));

} catch(NumberFormatException e){

} finally{

}

context.write(chaveMap, valorMap);

}

}

}

public static class ReducerInformacao8 extends Reducer<Text, IntWritable, Text, IntWritable>{

@Override

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

long soma = 0;

for(IntWritable val : valores){

soma += val.get();

}

IntWritable valorSaida = new IntWritable(0);

try {

valorSaida.set(Integer.parseInt(String.valueOf(soma)));

} catch(NumberFormatException e){

} finally {

}

context.write(chave, valorSaida);

}

}

public static void main(String[] args) throws IOException, InterruptedException, ClassNotFoundException{

String arquivoEntrada = "/home2/ead2021/SEM1/stephan.kotarsky/Desktop/stephan.kotarski/base\_inteira.csv";

String arquivoSaida = "/home2/ead2021/SEM1/stephan.kotarsky/Desktop/stephan.kotarski/Informacao8";

if(args.length == 2){

arquivoEntrada = args[0];

arquivoSaida = args[1];

}

Configuration conf = new Configuration();

Job job = Job.getInstance(conf, "atividade8");

job.setJarByClass(Informacao8.class);

job.setMapperClass(MapperInformacao8.class);

job.setReducerClass(ReducerInformacao8.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

FileInputFormat.addInputPath(job, new Path(arquivoEntrada));

FileOutputFormat.setOutputPath(job, new Path(arquivoSaida));

job.waitForCompletion(true);

}

}

**Resposta: Buta-1, 3-diene and isoprene - 2012 251046281**

**Buta-1, 3-diene and isoprene - 2013 222247320**

**Buta-1, 3-diene and isoprene - 2014 240882757**

**Buta-1, 3-diene and isoprene - 2015 242902667**

**Buta-1, 3-diene and isoprene - 2016 334957626**