**Tiago Daltro Duarte – Exercicío Linux**

1. Iniciar o cluster de Big Data

**Resposta:**

docker-compose up -d

2. Acessar o container do namenode.

**Resposta:**

docker.exe exec -it namenode bash

3. listar todos os diretórios de /input

**Resposta:**

ls

4. entrar na pasta /input e criar uma pasta “dados”

**Resposta:**

cd /input

mkdir dados

5. criar um arquivo dados\_cliente.csv e adicionar as seguintes linhas abaixo:

**Id;nome;idade**

**1;maria,35**

**2;joao;23**

**3;Paulo;15**

**Resposta:**

touch dados\_cliente.csv

digitar o texto dentro do arquivo

6. apresentar em tela o arquivo

**Resposta:**

cat dados\_cliente.csv

7. renomear o arquivo para dados\_alunos.csv

**Resposta:**

mv dados\_cliente.csv dados\_alunos.csv

8. criar um shellscript para criar a seguinte estrutura de pastas:

**/input/dados/dia01**

**/input/dados/dia02**

**/input/dados/dia03**

**Resposta:**

touch shellscript.sh

#!/bin/bash – Colocar no topo do arquivo

digitar o texto dentro do arquivo e salvar

mkdir -p /input/dados/dia01

mkdir -p /input/dados/dia02

mkdir -p /input/dados/dia03

chmod +x script.sh

9. mover o arquivo dados\_alunos.csv para a pasta /input/dados/dia01

**Resposta:**

mv arquivo dados\_alunos.csv /input/dados/dia01

10. adicionar mais um registro ao arquivo dados\_alunos.csv

**4;Pedro;27**

**Resposta:**

Digitar dentro do arquivo shellscript.sh (4; Pedro;27) e salvar

11. visualizar os 4 registros em tela do arquivo dados\_alunos.csv

**Resposta:**

cat dados\_alunos.csv

12. renomear a pasta /input/dados/dia03 para /input/dados/dia\_prova

**Resposta:**

mv /input/dados/dia03 /input/dados/dia\_prova

13 deletar a pasta /Input/dados

**Resposta:**

rm -rf /Input/dados

14. desligar o cluster

**TIAGO DALTRO DUARTE – Exercício de HDFS**

1. Iniciar o cluster de Big Data

$ docker-compose up -d

2. Acessar o container do namenode.

**Resposta:**

docker.exe exec -it namenode bash

3. Baixar os dados dos exercícios do treinamento na pasta home/input

**Efetuar o Download do csv em:**

**Curl –O https://raw.githubusercontent.com/caiuafranca/dados\_curso/main/alunos.csv**

4. Criar a estrutura de pastas no HDFS como apresentada a baixo pelo comando:

- /user/<SEU NOME>/data

**Resposta:**

hdfs dfs -mkdir -p /user/tiago/data

5. Enviar arquivo “/input/alunos.csv” para o diretório criado

**Resposta:**

**hdfs dfs -moveFromLocal** /input/alunos.csv /user/tiago/data/

6. Verificar as 5 primeiras linhas do arquivo “alunos.csv”

**Resposta:**

hdfs dfs -cat /user/tiago/data/alunos.csv | head -5

7. Mostrar ultimas linhas do arquivo “alunos.csv”

**Resposta:**

hdfs dfs -tail /user/tiago/data/alunos.csv

8. Contar linhas do arquivo “alunos.csv”

**Resposta:**

hdfs dfs -cat /user/tiago/data/alunos.csv

9. Criar um arquivo em branco com o nome de “teste.txt” em /user/<SEU NOME>/data

hadoop fs -touchz /user/tiago/data/teste.txt

10. Apagar arquivo “teste.txt”

**Resposta:**

**hdfs dfs -rm** /user/tiago/data/teste.txt

11. Exibir o espaço livre e o uso do disco

**Resposta:**

hdfs dfs -df -h /

**Tiago Daltro Duarte - Desafio 1**

**Arquivo rollout.sh**

**#!/bin/bash**

echo "Criando diretorio in"

hdfs dfs -mkdir -p /dados/indiana\_jones/in/

echo "Criando diretorio process"

hdfs dfs -mkdir -p /dados/indiana\_jones/process/

echo "Criando diretorio delete"

hdfs dfs -mkdir -p /dados/indiana\_jones/delete/

**Arquivo job**

**#!/bin/bash**

echo "foi considerado que o arquivo com maior quantiade de linhas é o mais recente"

echo "O arquivo que não possui a maior quantidade de linhas e nem a menor quantiade de linhas foi considerado que possui menos de 1 dia."

echo "foi considerado que o arquivo com menor quantiade de linhas possui mais de 1 dia."

hdfs dfs -moveFromLocal ./dados/3\_ingestao/dados\_cliente.txt /dados/indiana\_jones/in/

hdfs dfs -moveFromLocal ./dados/2\_ingestao/dados\_cliente.txt /dados/indiana\_jones/process/

hdfs dfs -moveFromLocal ./dados/1\_ingestao/dados\_cliente.txt /dados/indiana\_jones/delete/

**TIAGO DALTRO DUARTE – RESPOSTAS EXERCICIOS HIVE**

docker-compose up -d

1 – docker exec -it hive-server bash

Mkdir -p /home/input/dados

Curl -O <https://raw.githubusercontent.com/caiuafranca/dados_curso/main/cursos.csv>

2 – No diretório root@hive\_server:/opt# foi criado um diretório HDFS:

Hdfs dfs -mkdir -p user/aluno/dados/curso/

Depois ir ao diretório que está o arquivo cursos.csv e movê-lo para o diretório hdfs criado:

hdfs dfs -moveFromLocal curso.csv /user/aluno/dados/curso

3 – beeline -u jdbc:hive2://localhost:10000

show databases;

create database treinamento;

4 – CREATE EXTERNAL TABLE IF NOT EXISTS cursos\_stg(

`id\_curso` int,

`id\_unidade` int,

`codigo` int,

`nome` string,

`nivel` string,

`id\_modalidade\_educacao` int,

`id\_municipio` int,

`id\_tipo\_oferta\_curso` int,

`id\_area\_curso` int,

`id\_grau\_academico` int,

`id\_eixo\_conhecimento` int,

`ativo` string)

COMMENT 'Tabela Externa de Cursos'

row format delimited

FIELDS TERMINATED BY ','

STORED AS TEXTFILE

LOCATION '/user/aluno/dados/curso/'

5 – select \* from cursos\_stg; (DICIDIR NÃO PRINT DA TABELA PORQUE FICOU MUITO GRANDE).

6 – COMANDO PARA VER AS 5 PRIMEIRAS LINHAS:

SELECT \* FROM cursos\_stg LIMIT 5;

7 – select count(\*) from cursos\_stg;

8 – create table cursos (

`id\_curso` int,

`id\_unidade` int,

`codigo` int,

`nome` string,

`nivel` string,

`id\_modalidade\_educacao` int,

`id\_municipio` int,

`id\_tipo\_oferta\_curso` int,

`id\_area\_curso` int,

`id\_grau\_academico` int,

`id\_eixo\_conhecimento` int,

`ativo` string)

PARTITIONED BY (dt\_foto STRING)

ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.orc.OrcSerde'

STORED AS INPUTFORMAT 'org.apache.hadoop.hive.ql.io.orc.OrcInputFormat'

OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.orc.OrcOutputFormat'

TBLPROPERTIES ( 'orc.compress'='SNAPPY');

SET hive.exec.dynamic.partition=true;

SET hive.exec.dynamic.partition.mode=nonstrict;

INSERT OVERWRITE TABLE treinamento.cursos

PARTITION (dt\_foto)

SELECT

id\_curso,

id\_unidade,

codigo,

nome,

nivel,

id\_modalidade\_educacao,

id\_municipio,

id\_tipo\_oferta\_curso,

id\_area\_curso,

id\_grau\_academico,

id\_eixo\_conhecimento,

ativo,

'21052022' as dt\_foto

from treinamento.cursos\_stg;

9 – show partitions treinamento.cursos;

10 – select count(\*) from treinamento.cursos;

11 – select count(\*) from treinamento.cursos where ativo = 1;

**TIAGO DALTRO DUARTE – EXERCICÍO DE SPARK**

"Comando para entrar no container spark:"

docker.exe exec -it spark bash

"Carregar os packages com as classes de leitura de arquivos CSVs ao abrir o spark-shell:"

spark-shell –packages com.databricks:spark-csv\_2.10:1.5.0

"Comando para ler os arquivos do HDFS e salvar nos dataframes"

val dataframe\_clientes = spark.read.option("delimiter",";").option("header","true").option("inferSchema","true").csv("/dados\_processamento/dados/CLIENTES.csv")

val dataframe\_divisao = spark.read.option("delimiter",";").option("header","true").option("inferSchema","true").csv("/dados\_processamento/dados/DIVISAO.csv")

val dataframe\_endereco = spark.read.option("delimiter",";").option("header","true").option("inferSchema","true").csv("/dados\_processamento/dados/ENDERECO.csv")

val dataframe\_regiao = spark.read.option("delimiter",";").option("header","true").option("inferSchema","true").csv("/dados\_processamento/dados/REGIAO.csv")

val dataframe\_vendas = spark.read.option("delimiter",";").option("header","true").option("inferSchema","true").csv("/dados\_processamento/dados/VENDAS.csv")

"Questao 4"

"CRUZAMENTO DE DADOS PARA SUBSTITUIR 1 POR International E 2 POR Domestic"

dataframe\_clientes.withColumn("Division", when(dataframe\_clientes("Division") === "1","International").when(dataframe\_clientes("Division") === "2","Domestic").otherwise("Não Informado")).show()

"CRUZAMENTO DE DADOS SUBSTITUIR A NÚMERAÇÃO PELO NOME DA REGIÃO:"

dataframe\_clientes.withColumn("Region Code", when(dataframe\_clientes("Region Code") === "0","Canada")

.when(dataframe\_clientes("Region Code") === "1","Western")

.when(dataframe\_clientes("Region Code") === "2","Southern")

.when(dataframe\_clientes("Region Code") === "3","Northeast")

.when(dataframe\_clientes("Region Code") === "4","Central")

.when(dataframe\_clientes("Region Code") === "5","International")

.otherwise("Não Informado")).show()

"Questao 5"

"Quantos pedidos foram realizados:"

dataframe\_VENDAS.count()

"Quantos clientes tem em nossa base:"

dataframe\_clientes.count()

"Quantos clientes temos por Região:"

val clientes = dataframe\_clientes.withColumn("Region Code", when(dataframe\_clientes("Region Code") === "0","Canada")

.when(dataframe\_clientes("Region Code") === "1","Western")

.when(dataframe\_clientes("Region Code") === "2","Southern")

.when(dataframe\_clientes("Region Code") === "3","Northeast")

.when(dataframe\_clientes("Region Code") === "4","Central")

.when(dataframe\_clientes("Region Code") === "5","International")

.otherwise("Não Informado"))

"Canada"

clientes.filter(clientes("Region Code") === "Canada").count()

"Western"

clientes.filter(clientes("Region Code") === "Western").count()

"Southern"

clientes.filter(clientes("Region Code") === "Southern").count()

"Northeast"

clientes.filter(clientes("Region Code") === "Northeast").count()

"Central"

clientes.filter(clientes("Region Code") === "Central").count()

"International"

clientes.filter(clientes("Region Code") === "International").count()