

ANO
2024



UNINTER

ATIVIDADE PRÁTICA

BIG DATA

Roteiro Elaborado por:
Prof. MSc. Guilherme Ditzel Patriota



INTRODUÇÃO

Olá a todos.

Sejam todos muito bem-vindos!

Esta avaliação foi planejada e preparada para as disciplinas de Big Data dos Cursos de Tecnologia em Ciência de Dados e Engenharia da Computação do Centro Universitário Internacional Uninter.

O objetivo desta atividade é fazer com que você, aluno, desenvolva os conhecimentos teóricos aprendidos na rota de maneira práticas e aplicável no mercado de trabalho. Para tanto, será necessário o uso da instalação do Hadoop e Spark em máquina virtual, criada nas aulas práticas 1 e 2, ou instalação do Spark em seu sistema, ou uso da biblioteca PySpark em conjunto com o Google Colab. Esta prática é baseada na aula prática 5, sobre o Spark com Hadoop.

Ao longo desse roteiro serão passadas as orientações gerais para realização da avaliação bem como os seus critérios de correção. Na sequência, apresenta-se um exemplo comentado de como se deve ser entregue uma questão. Seguindo o roteiro estarão as práticas a serem realizadas, cada uma delas possui uma explicação de como deve ser feita e como será cobrada e algumas dicas. Por fim, apresento uma seção com as respostas das dúvidas mais frequentes realizadas por vocês. Bons estudos!

*No mais, desejo-lhe boa atividade prática em nome dos professores
da disciplina de Big Data.*



LISTA DE FIGURAS

Figura 1: Resultado do somatório de todos os valores de cada pedido do banco de dados feito no Hadoop com MariaDB. _____

12



LISTA DE TABELAS

<i>Tabela 1: Possíveis notas no formato de apresentação</i>	<i>7</i>
<i>Tabela 2: Possíveis notas critério de Identificação Pessoal</i>	<i>8</i>
<i>Tabela 3: Possíveis notas na apresentação do código</i>	<i>9</i>
<i>Tabela 4: Possíveis notas na apresentação das imagens/fotos</i>	<i>10</i>
<i>Tabela 5: Possíveis notas na apresentação das respostas</i>	<i>11</i>



SUMÁRIO

INTRODUÇÃO	1
LISTA DE FIGURAS	2
LISTA DE TABELAS	3
ORIENTAÇÕES GERAIS	5
FORMATO DE ENTREGA	5
CRITÉRIOS DE AVALIAÇÃO	6
FORMATO DA APRESENTAÇÃO	7
IDENTIFICAÇÃO PESSOAL	8
CÓDIGO	9
IMAGENS/PRINTS	10
EXEMPLO DE APRESENTAÇÃO DE QUESTÃO	12
PRÁTICAS	13
MOTIVAÇÃO DO TRABALHO	13
DESCRIÇÃO DO PROJETO	13
DESCRIÇÃO DO CONJUNTO DE DADOS	13
OBJETIVO DO PROJETO	14
PASSOS INICIAIS	15
PRÁTICA 01 - SOMATÓRIO DE IDS	22
PRÁTICA 02 – DIFERENÇA NA SOMA DE PALAVRAS	23
RESPOSTAS AS DÚVIDAS MAIS FREQUÊNTES	25

ORIENTAÇÕES GERAIS

FORMATO DE ENTREGA

A entrega desta atividade prática deverá ser realizada pela área de “Trabalhos”, contendo os prints das duas resoluções no software, com o código usado, o resultado da questão e mais o seu RU digitado na linha de comandos no Hadoop/Spark/MariaDB e no seu código (sua IP = Identificação Pessoal).

O formato de entrega desejável dos prints das práticas desse roteiro, deve estar de acordo com o que é visto na seção “EXEMPLO DE APRESENTAÇÃO DE PRÁTICA”.

Recomenda-se que os trabalhos sejam enviados no formato .pdf. Uma vez que formatos .doc ou .docx podem apresentar falhas do tipo na codificação, carregamento ou apresentação de imagens. Sendo assim, fica **por conta e risco do estudante** se houver problemas com o documento enviados no formato doc ou docx ou outro formato editável.

Trabalhos feitos em outra forma que não seja utilizando SPARK HADOOP ou o PySpark perderão metade da nota total, referente aos prints de execução!

CRITÉRIOS DE AVALIAÇÃO

Os critérios de avaliação desse trabalho visam deixar a avaliação o mais justa e transparente possível. Nessa avaliação, cada questão valerá 20,00 pontos, sendo um total de 40 pontos de trabalho.

Cada questão será composta por print do código, print da tela do Hadoop/Spark/MariaDB/Colab com o resultado e resposta da questão. As questões serão avaliadas e corrigidas individualmente conforme a seguinte equação:

$$N = (FE) \cdot (IP) \frac{COD + IMG + RESP}{15}$$

Em que:

N (Nota da Questão): Nota total da questão, podendo variar de 0 até 20.

FE (Formato da Entrega): Nota do Formato de Entrega, podendo variar de 0 até 1.

IP (Identificação Pessoal): Nota Identificação Pessoal, podendo variar de 0 até 1.

COD (Código): Nota do Código usado, podendo variar de 0 até 100.

IMG (Imagens): Nota da Imagem com resultado correto, podendo variar de 0 até 100.

RESP (Resposta): Nota da Resposta com resultado correto, podendo ser 0 ou 100.

Cada um dos itens/critérios que compõe a equação acima será detalhado nas subseções a seguir. **Se mesmo assim houver dúvidas, não hesite em perguntar. O desconhecimento dos critérios não será aceito como desculpa!**

FORMATO DA APRESENTAÇÃO

O formato da apresentação é um dos critérios de avaliação, pois um profissional deve ser capaz de seguir normas no momento de elaboração de relatórios técnicos, manuais e outros documentos afins, bem como ser capaz de apresentar seus dados de forma limpa e compreensível.

As possíveis notas desse critério são apresentadas na tabela a seguir:

Tabela 1: Possíveis notas no formato de apresentação

NOTA	DESCRIÇÃO NA DEVOLUTIVA	COMENTÁRIOS
1,00	Formato da apresentação está correto	Está de acordo com o exemplo (ver a seção “EXEMPLO DE APRESENTAÇÃO DE PRÁTICA” para maiores detalhes)
0,70	Formato da apresentação está parcialmente correto	Está muito próximo do exemplo, mas apresenta alguns erros
0,50	Formato da apresentação está incorreto	Não seguiu o exemplo.

IDENTIFICAÇÃO PESSOAL

Todas as questões deverão apresentar um identificador pessoal nas seguintes partes:

- No código deve haver ao menos uma variável cujo nome seja composto pelo seu RU (e.g. contadorxxxxxx – onde o “x” s deve ser substituído pelo seu RU), mesmo que esta variável não seja utilizada em nenhuma parte do código.
- Nas imagens/prints do Hadoop, onde deverá conter seu RU escrito na linha de comandos do HADOOP/SPARK/MariaDB/Google Colab.

As possíveis notas para esse critério são apresentadas na tabela a seguir:

Tabela 2: Possíveis notas critério de Identificação Pessoal

NOTA	DESCRIÇÃO NA DEVOLUTIVA	COMENTÁRIOS
1,00	Apresentou o identificador pessoal no código e nas imagens/fotos.	Está de acordo com o exemplo (ver a seção “EXEMPLO DE APRESENTAÇÃO DE QUESTÃO” para maiores detalhes).
0,80	Apresentou identificador pessoal na imagem, mas não no código.	Não apresentou um identificador no código (e.g. o RU como parte do nome de uma variável)
0,70	Apresentou o identificador pessoal no código, mas não nas imagens/prints.	Não apresentou um identificador na imagem (exemplo: Linha de comandos do MariaDB com o RU do aluno.
0,50	Não apresentou identificador pessoal no código e nem nas imagens/prints.	Questão sem nenhuma identificação de autoria.
0,00	Apresentou o identificador de outra pessoa nas prints e/ou no código.	A questão veio com identificador pessoal de outra pessoa.

CÓDIGO

A apresentação dos códigos compõe um terço da nota total das questões. Este será avaliado conforme a tabela a seguir:

As possíveis notas para esse critério são apresentadas na tabela a seguir:

Tabela 3: Possíveis notas na apresentação do código

NOTA	DESCRIÇÃO NA DEVOLUTIVA	COMENTÁRIOS
100	Código coerente com a resposta encontrada e apresentado no formato imagem .	Está de acordo com o exemplo (ver a seção “EXEMPLO DE APRESENTAÇÃO DE QUESTÃO” para maiores detalhes)
70	Código coerente com a resposta encontrada e apresentado no formato texto .	Acertou o código, mas copiou o texto do código ao invés de tirar <i>print</i>
60	Código parcialmente correto e apresentado no formato imagem .	Errou um pouco código, mas colocou no trabalho no formato imagem
40	Código parcialmente correto e apresentado no formato texto .	Errou um pouco código e copiou o texto do código ao invés de tirar <i>print</i>
0	Sem código ou com código incorreto	A questão não apresentou código ou o código estava errado.

OBS. 1: NÃO ESQUECER DO IDENTIFICADOR PESSOAL (Ex.: COLOCAR SEU RU NO NOME DE UMA VARIÁVEL DO PROGRAMA).

OBS. 2: CÓDIGOS EXECUTADOS SEM USO DO SPARK OU PYSPARK TERÃO 100% DA NOTA REDUZIDA!!

IMAGENS/PRINTS

As imagens compõem um terço da nota total de cada questão. Essas, normalmente, são prints da tela com o código ou os softwares em execução. Cada prática/questão dessa atividade prática virá com instruções de como devem ser esses prints.

Entende-se que a **legenda faz parte de uma imagem**. Sendo assim, as **legendas serão avaliadas**.

As possíveis notas para esse critério são apresentadas na tabela a seguir:

Tabela 4: Possíveis notas na apresentação das imagens/fotos

NOTA	DESCRIÇÃO NA DEVOLUTIVA	COMENTÁRIOS
100	Imagens corretas e com legenda adequada .	Está de acordo com o exemplo (ver a seção “EXEMPLO DE APRESENTAÇÃO DE QUESTÃO” para maiores detalhes)
90	Imagens correta , mas com legenda superficial .	Ex. de legenda superficial: “Figura 1: Código em Python”.
80	Imagens corretas , mas com legenda precária .	Ex. de legenda precária: “Figura 1: Código”
70	Imagens correta , mas sem legenda.	Apresentou imagens corretas, mas não colocou legenda.
60	Imagens parcialmente corretas, mas com legenda adequada .	Imagem que não consiga identificar o que esteja acontecendo ou a falta de uma das imagens se encaixam nesse grupo.
50	Imagens parcialmente correta, e com legenda superficial .	Similar ao segundo item de cima para baixo dessa tabela, mas com pelo menos uma das imagens com problemas.
40	Imagens parcialmente corretas, e com legenda precária .	Similar ao terceiro item de cima para baixo dessa tabela, mas com pelo menos uma das imagens com problemas.
30	Imagens parcialmente correta, e sem legenda.	Similar ao quarto item de cima para baixo dessa tabela, mas com pelo menos uma das imagens com problemas.
0	Sem imagens ou com imagens incorretas	A questão veio sem imagens ou com imagens erradas

OBS. 1: NÃO ESQUECER DO IDENTIFICADOR PESSOAL (Ex.: DIGITAR SEU RU NA LINHA DE COMANDO QUE APAREÇA NO PRINT).



RESPOSTA

A apresentação da resposta correta será avaliada de forma booleana:

As possíveis notas para esse critério são 0 ou 100:

Tabela 5: Possíveis notas na apresentação das respostas

NOTA	DESCRIÇÃO NA DEVOLUTIVA	COMENTÁRIOS
100	Resposta correta	Está de acordo com o exemplo (ver a seção “EXEMPLO DE APRESENTAÇÃO DE QUESTÃO” para maiores detalhes)
0	Resposta incorreta	A questão não apresentou a resposta correta para a pergunta.

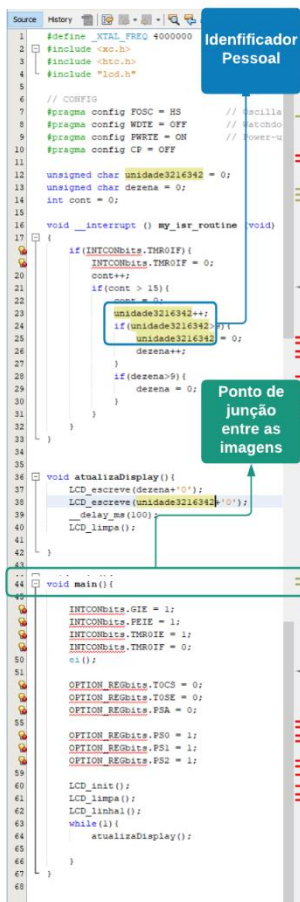
EXEMPLO DE APRESENTAÇÃO DE QUESTÃO

Prática XX – Pedidos de clientes com valor

Questão XX – Consolidação dos valores de cada pedido

Enunciado: Encontre o valor total de cada pedido feito.

I. Apresentação do Código (não esquecer do identificador pessoal):

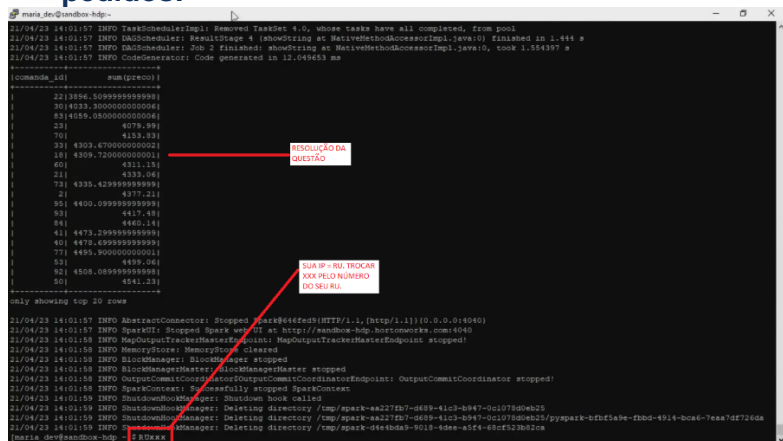


Identificador Pessoal

Ponto de junção entre as imagens

II. Apresentação das Imagens/Prints (não esquecer do identificador pessoal):

a. Print do resultado para soma do valor total de pedidos:



CONSOLIDAÇÃO DA QUESTÃO

SQL SP - RU, TROCAR O SEU RUI

Figura 1: Resultado do somatório de todos os valores de cada pedido do banco de dados feito no Hadoop com MariaDB.

III. Resposta à pergunta: Qual o maior valor de pedido?

Resposta: O maior valor encontrado foi o da comanda de id 50, no valor de 4.541,23.

OBS1: Nas suas imagens não precisa circular e apontar o identificador pessoal.

OBS2: Perceba que toda a atividade está contida numa só página.

OBS3: Optou-se por juntar as imagens do código. No entanto, não houve perda de clareza e organização.

PRÁTICAS

As práticas desse roteiro utilizam a máquina virtual com Hadoop, Spark e MariaDB criados nas aulas práticas da rota. Sugerimos o uso do HDP Cloudera 2.6.5 por ser mais leve do que o mais atual. O processo de download pode ser encontrado nos documentos adicionais de nossa rota de estudos.

MOTIVAÇÃO DO TRABALHO

Saber criar uma estrutura de Big Data é imprescindível para profissionais Cientistas de Dados. Os conhecimentos necessários vão de criação de servidores dedicados, clusters ou máquinas virtuais a mineração, representação e interação com grandes quantidades de dados.

Muitas empresas utilizam miniprojetos, como este que faremos, para filtrar seus candidatos e buscar os melhores talentos no mercado.

Sabendo disso, idealizamos esta atividade para que você possa treinar e aplicar seus conhecimentos adquiridos nas aulas teóricas e práticas.

Este projeto será feito com utilização dos servidores em máquinas virtuais criados nas aulas práticas para que um conjunto grande de dados seja adquirido, estruturado, filtrado e minerado.

DESCRIÇÃO DO PROJETO

Um famoso conjunto de dados, também chamado de dataset, é a muito utilizado para criação de modelos de processamento de linguagem natural e de clusterização (classificação em grupos) para análise de textos que apresentem algum tipo de sentimento, seja ele positivo ou negativo.

Estamos falando sobre o dataset IMDB, um conjunto de textos com reviews de filmes diversos.

Vamos utilizar neste projeto a versão traduzida para português, destes dados.

É importante sabermos que neste trabalho não faremos nenhum tipo de processamento de linguagem natural, apenas utilizaremos o dataset IMDB como exemplo e realizaremos coletas de informações simples sobre mesmo.

DESCRIÇÃO DO CONJUNTO DE DADOS

O conjunto de dados IMDB PT-BR é uma tradução para português do famoso conjunto de dados IMDB, usado para criação de classificadores automatizados de texto em textos positivos e negativos.

Traduzido por Luís Fred e publicado na plataforma Kaggle (<https://www.kaggle.com/luisfredgs/imdb-ptbr>) de forma livre e gratuita, estes dados foram muito usados em pesquisas científicas nas áreas de Ciência de Dados e Big Data.

O arquivo postado por Fred está em formato CSV, possui aproximadamente 125 MB, com pouco menos de 50 mil registros e conta com 4 colunas de dados referentes a resenhas de filmes com classificação em positivo ou negativo conforme o sentimento expressado pelo texto da resenha.

Cada registro de resenha possui seu texto original em inglês e uma tradução em português, sendo duas das quatro colunas existentes, “text_en” e “text_pt”, respectivamente. As outras duas colunas de dados são “id” (um número sequencial de identificação de cada resenha) e “sentiment” (uma coluna de com informação binária “pos” com aproximadamente 50% dos dados e “neg” com os outros 50% das resenhas).

OBJETIVO DO PROJETO

Nosso objetivo com este dataset é responder, as duas questões que se encontram neste documento, criar o relatório da atividade prática, conforme modelo apresentado anteriormente e publicar em “Trabalhos”.

Para que as questões possam ser respondidas será necessário que você utilize o Spark Hadoop em conjunto com outras ferramentas aprendidas em aula.

De forma resumida, você deverá cumprir os seguintes passos mínimos, para finalizar esta tarefa:

1. **AQUISIÇÃO:** Fazer o download do dataset no link fornecido abaixo.
2. **ESTRUTURAÇÃO:** Carregar o dataset no HDFS do hadoop em sua máquina (virtual ou local) ou imagem Docker ou Google Colab e carregar uma visualização formatada dos dados, um Spark DataFrame (o uso da biblioteca **Pandas** para criação do DataFrame **não será aceito** neste trabalho). Aqui você deverá importar o dataset para um Spark DataFrame. Certifique-se de que seus dados foram importados corretamente. Este dataset em particular pode gerar diversos erros de importação por conta de a língua portuguesa conter diversos símbolos menos comuns em inglês, como vírgulas e acentuações.
3. **FILTRAGEM:** Informações desnecessárias, espaços duplicados e demais artefatos irrelevantes ao projeto deverão ser filtrados, para obtenção do dado corretamente analisado. Outras técnicas poderão ser necessárias nesta etapa, para um resultado mais preciso.
4. **MINERAÇÃO:** Agrupar os dados de forma a obter as respostas solicitadas nas perguntas com uso ou de algoritmo em python (spark-submit xyz...py) ou com uso de linguagem Scala. O uso da biblioteca PySpark no Google Colab também será aceito nesta etapa, mas não o uso da biblioteca Pandas.
5. **REPRESENTAÇÃO:** Criar uma forma visual de analisar os dados pode ser muito útil para garantir a precisão das suas respostas. Nesta etapa, é importante que você revise seus passos e verifique se o seu resultado é coerente. Lembre-se, Mineração de dados só será útil se você souber o que está buscando. Confira cada etapa do seu processamento antes de aceitar qualquer resultado.
6. **REFINAMENTO:** Encontrou alguma divergência em seus resultados que pode ser solucionada? Agora é o momento de voltar no seu processo e melhorá-lo.
7. **INTERAÇÃO:** Esta etapa é o fruto de nosso trabalho em Big Data, mas para este trabalho, utilize as questões aqui descritas para garantir sua resposta o mais precisa possível e depois vá ao UNIVIRTUS, na área de trabalho da nossa matéria e poste seu relatório com as respostas e prints das questões lá.



Para dar um pontapé inicial em nosso projeto, seguem alguns passos iniciais, levando em consideração que você já tem todas as etapas das aulas práticas replicadas em sua máquina local.

PASSOS INICIAIS

Neste projeto serão utilizados os servidores feitos em máquinas virtuais durante as aulas práticas, que exigem ao menos 8GB de memória RAM instalados em seu computador e um processador de ao menos 2 núcleos, mas você poderá realizar esta prática de diversas outras formas mostradas abaixo:

- **Uso do Google Colab com Spark e biblioteca PySpark (esta é a opção **mais indicada** e barata e leve de todas, porém você não terá o contato completo com um servidor SPARK ou HADOOP como sugerido por este trabalho.**
- **Utilização de containers Docker para criação dos servidores necessários**
- **Uso de serviços como AWS para criação dos servidores e instalação do Hadoop e Spark**
- **Uso de sua máquina local para instalar o Spark (necessário possuir Linux ou WSL instalado)**

Caso você queira se desafiar e ao mesmo tempo aprender a utilizar uma nova ferramenta muito cobrada em entrevistas de emprego e em startups, sugiro a utilização do Docker como ferramenta de virtualização dos servidores, porém esta é a opção que mais exigirá recursos de seu computador.

Além do exemplo usado abaixo, você também pode realizar os mesmos procedimentos feitos na aula prática 5, de Spark, para carregar pelo Ambari e filtrar os dados com uso do PySpark e linguagem Python, em conjunto com o MariaDB.

Exemplo 1 (o **mais indicado) – Importação e carregamento do arquivo .CSV no Google Colab com PySpark e linguagem Python:**

1. Acesse: https://colab.research.google.com/github/N-CPUinter/Big_Data/blob/main/trabalho_big_data.ipynb
2. Este arquivo já possui todos os comandos de criação do Spark Session, download e importação correta dos dados CSV.
3. Preencha as 6 células faltantes:
 - a. QUESTÃO 1:
 - i. **Criar funções de MAP**
 - ii. **Cria funções de REDUCE**
 - iii. **Aplicação do map/reduce e visualização do resultado**
 - b. QUESTÃO 2:
 - i. **Criar funções de MAP**
 - ii. **Cria funções de REDUCE**

iii. Aplicação do map/reduce e visualização do resultado

É importante destacar que a é possível importar os dados de um arquivo CSV com o uso de uma função de parse manual que recupere cada linha do arquivo CSV e coloque no local certo, porém esta solução é desnecessariamente complexa, por necessitar tratamento diferente entre a primeira linha e as demais linhas e ainda não ser trivial a interpretação das separações de colunas de uso em frase de vírgulas, ponto e vírgulas e aspas.

Exemplo 2 – Importação e carregamento do arquivo .CSV no Hadoop com apache Spark em linguagem SCALA:

1. Para este projeto, você precisará realizar o download do dataset “IMDB PT-BR”, disponível no repositório de dados “kaggle.com”: <https://www.kaggle.com/luisfredgs/imdb-ptbr/download>
2. Fazer o upload do arquivo para o HDFS do seu servidor Hadoop criado durante as aulas práticas.
 - a. Explicações iniciais:
 - i. Quando logado como maria_dev no sandbox-hdp, você está dentro do linux, e não dentro do sistema de arquivos HDFS do Hadoop, que é o sistema usado no Ambari pelo navegador (**Ambari Files View**).
 - ii. Para colocar um arquivo no HDFS, basta abrir as pastas no **Ambari Files View** e colocar seus arquivos na pasta /user/maria_dev.
 - iii. Temos aqui dois sistemas de arquivos:
 1. Ambari Files View = Sistema de arquivos do HDFS (todos os arquivos aqui já estão com armazenamento distribuído)
 2. Linux CentOS (acessado pelo putty) = Sistema de arquivos normais do Linux e que não estão no HDFS ainda.
 - b. Fazendo o acesso pelo PuTTY com o usuário **maria_dev**, você poderá fazer os comandos linux para ver os arquivos locais e mudar de pastas da seguinte forma (apenas exemplos que não precisam ser executados):
 - i. ls
 - ii. cd /
 - iii. cd ~
 - c. Para ver os arquivos que estão dentro do HDFS do Hadoop, temos que usar o comando (execute um dos dois para saber se seu arquivo imdb-reviews-pt-br.csv está listado corretamente):
 - i. **hadoop fs -ls**
(ou)
 - ii. **hdfs dfs -ls**
 - d. Estes dois comandos fazem a mesma coisa e te mostrarão os arquivos que estarão dentro da pasta "/user/maria_dev" vista no **Ambari Files View**.
 - i. Por padrão, a pasta que o comando hadoop fs percebe é a pasta de usuário do maria_dev no **Ambari Files View**.
 - e. Para copiar arquivos do hadoop para o linux faça o seguinte comando (execute este comando para copiar do HDFS para o Linux o seu arquivo CSV):

- i. **hdfs dfs -copyToLocal /user/maria_dev/imdb-reviews-pt-br.csv .**
(O ponto no final faz parte do comando)
3. Em seu prompt de comandos, faça a inclusão do pacote spark-csv ao abrir o spark-shell:
<https://spark-packages.org/package/databricks/spark-csv>
 - a. Comando:
spark-shell --packages com.databricks:spark-csv_2.10:1.5.0
ou
spark-shell --packages com.databricks:spark-csv_2.11:1.5.0
ou
spark-shell
 - b. Os próximos passos utilizarão o Spark com códigos em Scala no spark-shell.
4. Com o spark-shell aberto, importe o SQLContext, para permitir a leitura do arquivo CSV:
 - a. Comando:
import org.apache.spark.sql.SQLContext
5. Para realizarmos a interpretação dos dados no arquivo “imdb-reviews-pt-br.csv”, é necessário criarmos uma instância do SQLContext com o seguinte comando (dependendo da versão do HDP pode ser que a seção já exista com nome de spark em substituição ao sqlC):
 - a. Comando:
val sqlC = new SQLContext(sc)
6. Por fim, vamos importar os dados do arquivo “imdb-reviews-pt-br.csv” e criar um Spark DataFrame com ele, para que possamos dar sequência na mineração de dados. Para tal, utilize o seguinte comando:
 - a. No comando abaixo é importante que você substitua os seguintes dados:
 - i. **<HOST>** Substitua pelo endereço IP do seu host do HDFS da máquina virtual criada em aula prática. Provavelmente **localhost** ou **sandbox-hdp.hortonworks.com**. Para conferir, acesse o hadoop pelo PuTTY (maria_dev@127.0.0.1:2222) e veja dentro do arquivo core-site.xml o valor da propriedade fs.defaultFS. para abrir este arquivo, instale o nano (sudo yum nano) e depois digite “nano /etc/hadoop/conf/core-site.xml”
 - ii. **<PORT>** Substitua pela porta do seu host da máquina virtual, provavelmente 8020. Veja a porta no mesmo arquivo core-site.xml descrito anteriormente.
 - iii. **<CAMINHO DA PASTA ONDE VOCÊ COLOCOU O ARQUIVO CSV>**
Substitua pelo caminho da pasta na qual você colocou o arquivo “imdb-reviews-pt-br.csv” na sua máquina virtual do HADOOP.
 - b. Comando:
val imdbDf = sqlC.read.option("header", "true").option("quote", "\""").option("escape", "\\").csv("imdb-reviews-pt-br.csv")
Perceba as duas opções de “quote” e “escape”. Estas opções ajudarão a importação dos dados em português e podem ser usadas nos comandos anteriores também.
7. Após carregado o arquivo no DataFrame criado, você poderá manipulá-lo da forma que achar necessário para encontrar as respostas das nossas perguntas deste trabalho, mas

antes é muito importante garantir que a importação não causou erros de parse, como colunas com dados incorretos.

8. Para visualizar os dados carregados, utilize o comando:

- a. Comando:

```
imdbDf.show()
```

Caso esteja utilizando o Hadoop Cloudera, sugiro seguir os passos mostrados nas aulas práticas (carga pelo Ambari).

Exemplo 3 – Importação e carregamento do arquivo .CSV no Hadoop com apache Spark em linguagem Python:

1. Para este projeto, você precisará realizar o download do dataset “IMDB PT-BR”, disponível no repositório de dados “kaggle.com”: <https://www.kaggle.com/luisfredgs/imdb-ptbr/download>
2. Fazer o upload do arquivo para o HDFS do seu servidor Hadoop criado durante as aulas práticas (ver exemplo 1 - item 2).
3. Em seu prompt de comandos, faça:

- a. Comando:

```
pyspark
```

- b. Os próximos passos utilizarão o Spark com códigos em Python no pyspark shell.

4. Com o pyspark shell aberto pelo servidor SPARK ou HADOOP, a seção spark já é criada automaticamente e possui o nome `spark`, não sendo necessário abrir nova seção.
5. Para importarmos os dados do arquivo “imdb-reviews-pt-br.csv” e criar um Spark DataFrame com ele e então darmos sequência na mineração de dados, utilize o seguinte comando:

- a. No comando abaixo é importante que você substitua os seguintes dados:

- i. `<HOST>` Substitua pelo endereço IP do seu host do HDFS da máquina virtual criada em aula prática. Provavelmente `localhost` ou `sandbox-hdp.hortonworks.com`. Para conferir, acesse o hadoop pelo PuTTY (`maria_dev@127.0.0.1:2222`) e veja dentro do arquivo `core-site.xml` o valor da propriedade `fs.defaultFS`. para abrir este arquivo, instale o nano (sudo yum nano) e depois digite “nano /etc/hadoop/conf/core-site.xml”
- ii. `<PORT>` Substitua pela porta do seu host da máquina virtual, provavelmente 8020. Veja a porta no mesmo arquivo `core-site.xml` descrito anteriormente.
- iii. `<CAMINHO DA PASTA ONDE VOCÊ COLOCOU O ARQUIVO CSV>`
Substitua pelo caminho da pasta na qual você colocou o arquivo “imdb-reviews-pt-br.csv” na sua máquina virtual do HADOOP.

- b. Comando:

```
imdbDf = spark.read.csv('imdb-reviews-pt-br.csv', header=True, quote="\"", escape="\"", encoding="UTF-8")
```



- c. Perceba as opções de “quote”, “escape”, “header” e “encoding”. Estas opções ajudarão a importação dos dados em português e podem ser usadas nos comandos anteriores também.
6. Após carregado o arquivo no Spark DataFrame criado, você poderá manipulá-lo da forma que achar necessário para encontrar as respostas das nossas perguntas deste trabalho, mas antes é muito importante garantir que a importação não causou erros de parse, como colunas com dados incorretos ou caracteres errados.
7. Para visualizar os dados carregados, utilize o comando:
 - a. Comando:
`imdbDf.show()`

Exemplo 4 – Outras opções:

Além da VM do Cloudera (HDP Sandbox), também é possível realizar esta tarefa pelo AWS, Google Cloud Platform ou Azure, porém existe o risco de se ultrapassar os limites gratuitos destas plataformas neste trabalho. Para estes casos, você deverá criar um cluster de 3 ou mais máquinas, instalar o hadoop em uma delas (NameNode) e indicar as demais como DataNode1 até 3. No caso da AWS:

1. Crie 4 servidores EC2 Ubuntu no maior nível gratuito possível (provavelmente t2.micro) (number of instances = 4).
2. Renomear as instâncias para NameNode, DataNode1, DataNode2 e DataNode3.
3. Pelo PuTTYGen, crie a chave privada SSH-2 RSA usando o arquivo .pem gerado na criação das máquinas pelo AWS.
4. Nas configurações do PuTTY, em SSH->AUTH e Escolha seu arquivo .ppk gerado, como chave privada para autenticação.
5. No Putty configure o endereço do “public DNS” de cada máquina dado nas descrições de cada uma delas na porta 22. (usuário: ubuntu)
6. Salve as configurações de acesso criadas de cada máquina.
7. Teste os acessos (perceba que suas máquinas no AWS devem estar rodando/ativas para isso).
8. Configure o WinSCP para envio de arquivos para os servidores. Mesmos hosts, porta 22, mesmo arquivo de autenticação gerado pelo PuTTYGen, usuário ubuntu, sem senha e protocolo SFPT.
9. Teste os acessos remotos às pastas. Aqui você pode enviar os arquivos para os servidores.
10. Instale o openjdk-7-jdk, copie os arquivos do hadoop, descompacte para /usr/local e mova todos os arquivos para a pasta /usr/local/hadoop, em todas as máquinas.
11. Altere as permissões para a nova pasta do hadoop em todas as máquinas: `sudo chown -R ubuntu /usr/local/hadoop`
12. Configure as variáveis de ambiente em todas as máquinas (em /home/ubuntu/.profile):
 - a. Incluir no final deste arquivo:
`export JAVA_HOME=/usr`
`export PATH=$PATH:$JAVA_HOME/bin`
`export HADOOP_HOME=/usr/local/hadoop`
`export PATH=$PATH:$HADOOP_HOME/bin`

```
export HADOOP_CONF_DIR /usr/local/hadoop/etc/hadoop
```

13. Recarregue as variáveis de ambiente em todas as máquinas: `./profile`
14. Altere os arquivos `core-site.xml` de todas as máquinas para conter a chave `fs.defaultFS`. Exemplo:

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://<seu_DNS_de_cada_máquina>:9000</value>
  </property>
</configuration>
```
15. Inclua no arquivo de `/etc/hosts` de seu NameNode todos os IPs dos DataNodes e do NameNode, antes do localhost.
16. Inclua, no seu NameNode, a configuração do cluster no arquivo `/usr/local/hadoop/etc/hadoop/hdfs-site.xml`:

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>3</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:///usr/local/hadoop/hadoop_data/hdfs/namenode</value>
  </property>
</configuration>
```
17. Crie no seu NameNode a pasta `/usr/local/hadoop/hadoop_data/hdfs/namenode`
18. Crie no seu NameNode o arquivo `/usr/local/hadoop/etc/hadoop/masters`
19. Apague o conteúdo do arquivo `/usr/local/hadoop/etc/hadoop/slaves` e adicione os endereços DNS públicos de cada um dos seus DataNodes.
20. Inclua, em cada um dos DataNodes, a configuração do cluster no arquivo `/usr/local/hadoop/etc/hadoop/hdfs-site.xml`:

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>3</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>file:///usr/local/hadoop/hadoop_data/hdfs/datanode</value>
  </property>
</configuration>
```
21. Crie em cada DataNode a pasta `/usr/local/hadoop/hadoop_data/hdfs/datanode`
22. Formate o HDFS no NameNode: `hdfs namenode -format`
23. Inicie o serviço do hadoop no NameNode: `/usr/local/hadoop/sbin/start-dfs.sh`
24. Teste o acesso em seu navegador: Seu_DNS_público_do_NameNode:50070



-
25. Inicie, no NameNode, o serviço do YARN:
`/usr/local/hadoop/sbin/start-yarn.sh`
`/usr/local/hadoop/sbin/mr-jobhistory-daemon.sh start historyserver`
 26. Inicie o JPS no NameNode: `jps`
 27. Instale o Spark no NameNode: Baixe os arquivos do spark e descompacte na pasta
`/usr/local/spark`
 28. Configure as variáveis de ambiente no NameNode (arquivo `~/.profile` anteriormente visto):
`export YARN_CONF_DIR=$HADOOP_HOME/etc/hadoop`
`export SPARK_HOME=/usr/local/spark`
`export PATH=$PATH:$SPARK_HOME/bin`
 29. Demais etapas são as mesmas do começo ao fim do exemplo 1 ou 2.
 30. Fim.

A seguir são explicadas as práticas a serem desenvolvida a fim de preencher corretamente o caderno de resolução (explicado na seção Formato de Entrega).



PRÁTICA 01- SOMA DE ID

Essa primeira prática foi idealizada como primeiro contato do aluno com o Spark/HADOOP. Sendo assim, a dificuldade aumenta conforme a progressão dos itens pedidos.

Nessa prática você deverá descobrir, utilizando sua máquina virtual com o Hadoop ou qualquer outra forma descrita anteriormente, qual o valor da **soma (não contagem)** de todos os campos “id” dos filmes classificados como **negativos** para o banco de dados “imdb-reviews-pt-br.csv”.

Sugestão de realização da tarefa:

1. Escreva um script em Python ou Scala para somar a coluna “id” das entradas baseadas na coluna “sentiment” (“sentiment” será a chave da dupla chave-valor para a etapa do map no map-reduce).
 - a. Se o problema for de somar os ids, adicione ao script uma função do map e outra do reduce para somar os ids
 - b. Se o problema for de contar palavras, crie uma função de map que já retorne o número de palavras de cada entrada e crie uma função reduce que some todas as quantidades de todas as entradas. (Em Python, sugiro usar o método .split() de strings, para separar as frases em palavras e descartar os espaços em branco. Usar .split(“ ”) gerará erro na contagem! Use a função len() para contar as palavras).
 - c. Sugiro que você crie apenas um script com todos os dados que precisará, para que a execução seja feita toda de uma vez. Não esqueça de criar uma saída com prints para que você possa entender o resultado apresentado.
2. Considerando que os passos de carga do banco de dados já foram executados, carregue o Script no servidor pelo Ambari
3. Copie o script para o HDFS pela interface do MariaDB (hadoop fs -copyToLocal seuscript.py)
4. Rode o script no Hadoop com seu usuário maria_dev (pelo PuTTY ou web-shell), pelo spark-submit (spark-submit seuscript.py)

OBS1: Utilizar ferramentas que não sejam de Big Data podem causar truncamento do banco de dados e perda de informações em sua análise.

OBS2: A execução dos passos sugeridos poderá ser diferente, dependendo de sua instalação e ferramentas preferenciais.

OBS3: É possível executar esta atividade inteiramente pelo console do Spark e sem o uso do MariaDB ou pelo Google Colab.

OBS3: O uso da biblioteca Pandas resultará na perda total de nota na questão.

PRÁTICA 02 – DIFERENÇA NA CONTAGEM DE PALAVRAS

Nessa prática você deverá contar todas as palavras existentes nos textos negativos em português e inglês e então informar quantas palavras a mais, no total, os textos em português possuem (total de palavras dos textos negativos em português menos o total de palavras dos textos negativos em inglês).

Para tal, crie um script em Python ou Scala e rode-o com sua máquina virtual Hadoop, como feito na prática 1.

Apesar de parecer simples, esta é a primeira etapa para que possamos aplicar técnicas de Processamento de Linguagem Natural (NLP), que se inicia com o processo de tokenização de textos para encontrarmos dados estatísticos de cada palavra no texto, como frequência em que aparecem.

Entretanto, nesta tarefa apenas contaremos todas as palavras dos textos em inglês e português com sentimentos negativos e faremos uma subtração entre os valores. Caso haja palavra repetida, contaremos quantas vezes forem necessárias.

É necessário se preocupar em filtrar corretamente as avaliações de filmes para que apenas os textos marcados como negativos sejam contabilizados.

Se sua escolha for Python, não será permitido o uso de bibliotecas como Pandas ou NLTK para esta tarefa, apenas a biblioteca PySpark é permitida.

Sugestão de realização da tarefa:

1. Escreva um script em Python ou Scala para contar as palavras dos textos em português e inglês onde a coluna “sentiment” seja igual a “neg” (“sentiment” será a chave da dupla chave-valor para a etapa do map no map-reduce).
 - a. Se o problema for de contar palavras, crie uma função de map que já retorne o número de palavras de cada entrada e crie uma função reduce que some todas as quantidades de todas as entradas. (Em Python, sugiro usar o método `.split()` de strings, para separar as frases em palavras e descartar os espaços em branco. Usar `.split(" ")` com aspas e espaço entre as aspas gerará erro na contagem! Use a função `len()` para contar as palavras. Em Python com o método `split` do pyspark, usar com o regex `"[]+"` ou `"\s+"`).
 - b. Sugiro que você crie apenas um script com todas os dados que precisará, para que a execução seja feita toda de uma vez, para as duas práticas. Não esqueça de criar uma saída com prints para que você possa entender o resultado apresentado.
2. Considerando que os passos de carga do banco de dados já foram executados, carregue o Script no servidor pelo Ambari



3. Copie o script para o HDFS pela interface do MariaDB (`hadoop fs -copyToLocal seuscript.py`)
4. Rode o script no Hadoop com seu usuário `maria_dev` (pelo PuTTY ou web-shell), pelo `spark-submit` (`spark-submit seuscript.py`)

OBS1: Utilizar ferramentas que não sejam de Big Data podem causar truncamento do banco de dados e perda de informações em sua análise.

OBS2: A execução dos passos sugeridos poderá ser diferente, dependendo de sua instalação e ferramentas preferenciais.

OBS3: É possível executar esta atividade inteiramente pelo console do Spark e sem o uso do MariaDB ou pelo Google Colab.

OBS3: O uso da biblioteca Pandas resultará na perda total de nota na questão.

RESPOSTAS AS DÚVIDAS MAIS FREQUÊNTES

1. Eu não tenho a máquina virtual com o Hadoop instalado em minha máquina. Como farei a atividade prática?

R: Siga as aulas práticas da Rota da aula prática 1 a 5. Lembre-se de configurar a máquina virtual para usar apenas metade da memória do seu computador e metade dos processadores. O indicado para esta tarefa é um computador dual core com 8GB de memória RAM e 90GB de HD livre. Caso você não possua acesso a um equipamento como este, você poderá desenvolver a atividade em seu polo, que possui computador para tal, ou usar alguma das soluções dadas na explicação, como serviços de servidor em nuvem (AWS, AZURE ou GCP), Instalar o Spark em sua máquina local (necessário Linux ou WSL no Windows) ou ainda usar o Google Colab.

2. Onde baixo os softwares criar os scripts das atividades?

R: Você poderá utilizar o Google Colab (<https://colab.research.google.com/>) de forma online ou instalar, caso você tenha o Windows, o Anaconda (<https://www.anaconda.com/products/individual>) para gerenciamento dos pacotes e ambientes virtuais do Python. Para criação do script, qualquer editor de texto servirá, porém sugiro a utilização do VSCode ou do PyCharm para esta tarefa.

Você poderá testar seus scripts com uso da biblioteca PySpark.

3. Poderia me explicar, resumidamente, como fazer o trabalho?

R: Lembre-se de que este trabalho pode ser feito pelo Google Colab!

De forma geral, aqui estão algumas linhas de código iniciais para começar o trabalho pelo Google Colab:

- Código inicial:

!pip install pyspark

```
from pyspark.sql import SparkSession  
spark = SparkSession.builder.getOrCreate()  
df = spark.read.csv('imdb-reviews-pt-br.csv', header=True, quote='"', escape='\\',  
encoding='UTF-8')
```

A partir daqui é necessário criar o comando para somar todos os IDs que possuam a coluna "sentiment" igual a "neg" (Primeira questão) e criar o comando Que calcula a diferença da

quantidade de palavras entre os textos português e inglês que possuam a coluna "sentiment" igual a "neg" (segunda questão).

Dicas:

- Com apenas 1 linha de comando é possível resolver a questão 1, mas não será um linha simples. A solução mais encontrada é:

1 - Criar uma função map que receba cada linha do dataframe como uma lista (padrão do método rdd.map do pyspark) e que retorna uma tupla com o primeiro elemento sendo a coluna sentiment (índice [3] da lista) e a conversão para int (int(string)) da primeira coluna (índice [0] da lista):

def map(x):

Apenas 1 linha de código com o Return da função map para retornar a tupla corretamente

2 - Criar uma função reduceByKey que recebe dois valores e retorna a soma destes dois valores:

def reduceByKey(x,y):

Apenas 1 linha de código com o Return da função reduceByKey para retornar a soma corretamente

3 - Aplicar no map/reduce do dataframe spark usando as duas funções criadas:

df.rdd.map(map**).reduceByKey(**reduceByKey**).collect()**

ou

df.rdd.map(map**).reduceByKey(**reduceByKey**).collect()[0][1]**

- Para a segunda questão, também é possível resolver com apenas 1 linha de código, porém o mais comum é:

1 - Cria a função map que receba cada linha do dataframe como uma lista (padrão do método rdd.map do pyspark) e que retorna uma tupla com o primeiro elemento sendo o dado em "sentiment" (índice [3] da lista) e o segundo elemento outra tupla com o



primeiro elemento sendo o comprimento da lista formada pelas palavras da coluna text_en (índice [1] da lista) (len(text_en.split())) e o segundo elemento é o mesmo, mas com a coluna text_pt (índice [2] da lista).

def map(x):

Apenas 1 linha de código com o Return da função map para retornar a tupla corretamente

2 - Criar uma função reduceByKey que recebe dois valores da segunda posição da tupla do retorno do map com a mesma key ("sentiment" neg ou pos) e retorna uma tupla com a soma dos dois elementos da segunda tupla dentro do segundo elemento da tupla de retorno do map:

def reduceByKey(x,y):

Apenas 1 linha de código com o Return da função reduceByKey para retornar a tupla da soma das quantidades de palavras corretamente

3 - Aplicar no map/reduce do dataframe spark usando as duas funções criadas:

df.rdd.map(map).reduceByKey(reduceByKey).collect()

ou

df.rdd.map(map).reduceByKey(reduceByKey).collect()[0]1

Para a resposta da questão 2, ficará faltando apenas **subtrair** o menor do maior valor.

4. Estou terminando o curso, tem como fazer um questionário para atividade prática?

R: Não.

5. Eu não possuo máquina para realizar esta atividade. Como devo proceder?

R: Você poderá usar um computador em seu polo. Nossas atividades são pensadas para execução em computador disponível nos polos, porém você pode optar por usar sua própria máquina.

6. Além de minha máquina ou a do polo, tenho outra opção?



R: Você poderá utilizar um dos serviços de servidor em nuvem como AWS, GCP ou AZURE, porém existe o risco de cobrança caso ultrapasse os limites de uso gratuito. Cuidado! Além disso, em último caso, você poderá usar o google colab apenas com a biblioteca PySpark para finalizar a tarefa, porém esta opção limitará seu aprendizado sobre as ferramentas Hadoop e Spark.

7. Preciso realizar algum tipo de análise ou processamento de linguagem natural nesta atividade?

R: Não, esta é uma atividade pensada apenas para colocar em prática seus conhecimentos de linguagem de programação, lógica e Big Data (Hadoop e Spark).

8. Como pode a faculdade solicitar um trabalho que exija equipamentos que o aluno pode não possuir?

R: Muitos dos conhecimentos que você deve aprender para trabalhar no mercado de trabalho são com base em materiais e equipamentos que você não possui ou não tem acesso fácil, como servidores, grandes bancos de dados, roteadores e switches de grande porte, equipamentos de grande porte e muitos outros. Independentemente desta dificuldade você deve aprender a utilizar estas ferramentas para poder mostrar um desempenho melhor do que os demais no mercado de trabalho. A qualidade do profissional que você será só depende de você e isto resultará em trabalhos mais bem remunerados no futuro.

Sendo assim, desejo a todos um ótimo aprendizado e nos vemos na tutoria.

Atenciosamente.

Professor Guilherme D Patriota.