




# Big Data e IoT




Componentes e arquitetura  
Aula 003  
Ara0168



# Introdução ao Hadoop


Fonte: <https://hadoop.apache.org/>



 Apache Hadoop

[Download](#) [Documentation](#) [Community](#) [Development](#) [Help](#)

Apache Software Foundation

 Apache Hadoop

The Apache™ Hadoop® project develops open-source software for reliable, scalable, distributed computing.

The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures.

[Learn more »](#) [Download »](#) [Getting started »](#)

### Latest news

Release 3.3.6 available2023 Jun 23

This is a release of Apache Hadoop 3.3 line. It contains 117 bug fixes, improvements and enhancements since 3.3.5. Users of Apache Hadoop 3.3.5 and earlier should upgrade to this release.

Feature highlights:

**SBOM artifacts**

Starting from this release, Hadoop publishes Software Bill of Materials (SBOM) using CycloneDX Maven plugin. For more information on SBOM, please go to [SBOM](#).

**HDFS RBF: RDBMS based token**

### Modules

The project includes these modules:

- **Hadoop Common:** The common utilities that support the other Hadoop modules.
- **Hadoop Distributed File System (HDFS™):** A distributed file system that provides high-throughput access to application data.
- **Hadoop YARN:** A framework for job scheduling and cluster resource management.
- **Hadoop MapReduce:** A YARN-based system for parallel processing of large data sets.

### Who Uses Hadoop?

A wide variety of companies and organizations use Hadoop for both research and production. Users are encouraged to add themselves to the Hadoop [PoweredBy](#) [wiki page](#).

### Related projects

Other Hadoop-related projects at Apache include:

- **Ambari™:** A web-based tool for provisioning, managing, and monitoring Apache Hadoop clusters which includes support for Hadoop HDFS, Hadoop MapReduce, Hive, HCatalog, HBase, ZooKeeper, Oozie, Pig and Sqoop. Ambari also provides a dashboard for viewing cluster health such as heatmaps and ability to view MapReduce, Pig and Hive applications visually alongwith features to diagnose their performance characteristics in a user-friendly manner.
- **Avro™:** A data serialization system.
- **Cassandra™:** A scalable multi-master database with no single points of failure.
- **Chukwa™:** A data collection system for managing large distributed systems.
- **HBase™:** A scalable, distributed database that supports structured data storage for large tables.

# Hadoop - Intro

---

## Introdução e contextualização

Atualmente, muitas aplicações envolvem grandes volumes de dados, como as transações financeiras on-line, a produção e o compartilhamento de conteúdo nas redes sociais e os estudos nas áreas da biologia genética.

Esses são apenas alguns exemplos que nos ajudam a ilustrar como situações semelhantes a essas estão inseridas no nosso cotidiano. Essas aplicações fazem parte do que conhecemos como Big Data.

Esse termo da língua inglesa foi incorporada ao nosso dia a dia para descrever um conjunto de tecnologias que gerencia aplicações complexas associadas à expressão **5Vs**, que descreve as características fundamentais das aplicações de Big Data (ISHWARAPPA; ANURADHA, 2015):

# Hadoop - Intro

## Volume

## Variedade

## Velocidade

## Veracidade

## Valor

Trata da quantidade de dados gerada e coletada pelas aplicações. Normalmente, uma aplicação é classificada como Big Data quando trabalha com um volume de dados da ordem de Petabytes (PB), sendo que um 1 PB corresponde a 1.024 Terabytes.

## Volume

## Variedade

## Velocidade

## Veracidade

## Valor

Os dados são encontrados em diversos formatos, podendo ser estruturados e não estruturados. É bastante comum trabalhar nesse tipo de aplicação com dados disponíveis em tabelas, arquivos texto e JSON, por exemplo.

# Hadoop - Intro

Volume	Variedade	<u>Velocidade</u>	Veracidade	Valor
--------	-----------	-------------------	------------	-------

Essa característica está relacionada tanto com a velocidade com a qual os dados são gerados quanto com a que são processados.

Volume	Variedade	Velocidade	<u>Veracidade</u>	Valor
--------	-----------	------------	-------------------	-------

Trata da questão fundamental da qualidade dos dados. Em especial, nesse tipo de aplicação, com tantas variáveis para se controlar, é muito importante aplicar técnicas e usar ferramentas para garantir a integridade e a qualidade dos dados e evitar processamentos desnecessários.

# Hadoop - Intro

Volume

Variedade

Velocidade

Veracidade

Valor

Essa característica está relacionada à recompensa que se espera obter ao trabalhar com aplicações de Big Data. Dados em grandes volumes são muito úteis em estudos estatísticos para descobrir padrões e adquirir conhecimento.

# Hadoop - Arquitetura

---

O Hadoop é uma tecnologia de framework de software livre desenvolvida pela Apache Foundation, sendo aplicado no armazenamento e no processamento de dados de grandes volumes, ou seja, em Big Data. Além da distribuição livre da Apache, o Hadoop possui outras distribuições, como:

- Cloudera;
- Hortonworks;
- MapR;
- IBM;
- Microsoft Azure;
- Amazon Web Services Elastic MapReduce Hadoop Distribution.

# Hadoop - Arquitetura

---

As grandes empresas da internet, como Facebook, Yahoo, Google, Twitter e LinkedIn, entre outras, usam o Hadoop pela natureza de suas aplicações, ou seja, pelos diferentes tipos de dados. Esses dados podem ser:

- Estruturados, como tabelas e planilhas;
- Não estruturados, como logs, corpo de e-mails e texto de blogs;
- Semiestruturados, como metadados de arquivos de mídia, XML e HTML.





# Hadoop - Arquitetura



MapReduce (modelo de programação paralela)



HDFS (sistema de arquivos distribuídos do Hadoop)

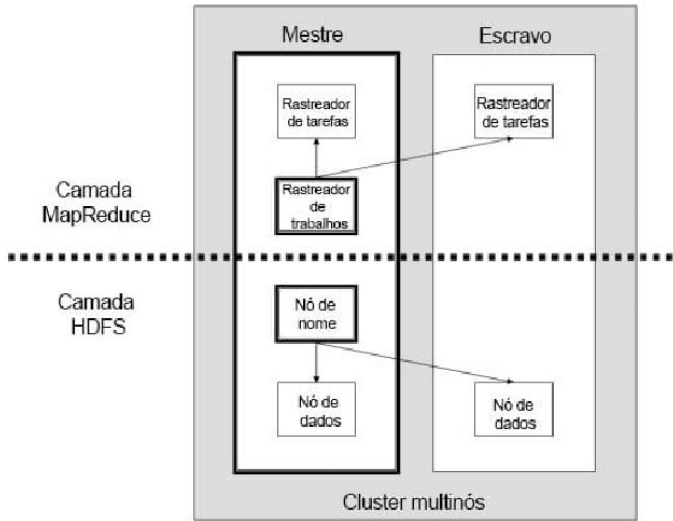


YARN (Yet Another Resource Negotiator)



Utilitários comuns do Hadoop (Hadoop Common)

# Hadoop - Arquitetura (Visão geral)



# Hadoop - Arquitetura - MapReduce

---

Para realizar o processamento paralelo dos dados, o Hadoop utiliza um mecanismo chamado de MapReduce. Trata-se de uma estrutura que trabalha com duas etapas distintas.

## Mapeamento dos dados

Essa etapa coleta os dados de entrada e os converte em um conjunto de dados que pode ser processado como um par do tipo valor e chave.

## Redução dos dados

O resultado da fase de mapeamento é consumido pela tarefa de redução. Em seguida, os dados são processados até se atingir o objetivo da aplicação.

# Hadoop - Arquitetura - MapReduce - Mapeamento

---

A entrada da fase de mapeamento é um conjunto de dados passado para a função Map. Essa função divide esses blocos de dados em tuplas, que são pares do tipo valor e chave. Tais pares são enviados para a fase de redução após a de mapeamento, que é dividida nas seguintes etapas:

## Leitor de registros (*record reader*)

Divide os dados de entrada em pares de chave e valor para serem enviados como entrada para o mapeador. A chave a que nos referimos possui informações de localização, enquanto o valor são os dados associados a ela.

# Hadoop - Arquitetura - MapReduce - Mapeamento

## Mapeador

Função definida pelo usuário que faz o processamento das tuplas obtidas do leitor de registros. Essa função não gera nenhum par do tipo chave-valor.

## Combinador (*combiner*)

É usado entre o mapeador e o redutor para reduzir o volume de transferência de dados entre ambos.

## Particionador

Tem duas funções principais: buscar pares de chave-valor gerados nas fases do mapeador e gerar os fragmentos correspondentes a cada redutor.

# Hadoop - Arquitetura - MapReduce - Redução

A função *reduce* combina as tuplas geradas pela fase de mapeamento e aplica as operações necessárias para processar os dados, que, em seguida, são enviados para o nó de saída final.

O processamento dos dados é sempre feito na fase de redução. Ela é dividida nas seguintes etapas:

## Embaralhamento e classificação



O processo em que o mapeador gera os pares intermediários chave-valor e os transfere para a fase de redução é conhecido como embaralhamento (é bastante comum usar o termo em inglês *shuffle*). Por meio da aplicação do processo de embaralhamento, o sistema pode classificar os dados com o uso de seus pares chave-valor.

Perceba aqui o benefício desse processo: à medida que algumas tarefas de mapeamento são concluídas, o embaralhamento já começa, ou seja, o processamento dos dados não espera pela conclusão da tarefa feita pelo mapeador.

# Hadoop - Arquitetura - MapReduce - Redução

---

## Redução



Essa tarefa agrupa as tuplas geradas a partir do mapeamento e, em seguida, aplica os processos de classificação e agregação nesses pares de chaves e valores.

# Hadoop - Arquitetura - MapReduce - Redução

---

## Gravação da saída



Quando todas as operações são executadas, os pares de chaves e valores são gravados em arquivo. Cada registro é gravado em uma nova linha, e a chave e o valor são separados por espaço.



# Hadoop - Arquitetura - HDFS (sistema de arquivos distribuídos Hadoop)

---

O sistema de arquivos distribuídos do Hadoop, mais conhecido como HDFS (*Hadoop Distributed File System*), é responsável pelo armazenamento de dados em um cluster do Hadoop. O HDFS foi projetado para trabalhar com grandes volumes de dados em hardware comum, isto é, em dispositivos baratos, como computadores pessoais.

**Trata-se de um sistema que tem tolerância a falhas e, além disso, fornece alta disponibilidade para a camada de armazenamento e os outros dispositivos presentes nesse cluster do Hadoop.**

A arquitetura do HDFS, por meio dos componentes NameNode e DataNode, utiliza um sistema de arquivos distribuídos que proporciona alto desempenho no acesso aos dados em clusters Hadoop. Esses arquivos podem ser expandidos, ou seja, são altamente escaláveis. Vamos entender os componentes do HDFS de maneira mais detalhada:

# Hadoop - Arquitetura - HDFS - Namenode

---

## NameNode

Desempenha o papel de “mestre” em um cluster Hadoop que gerencia o Datanode (nós “escravos”). Ele tem como objetivo o armazenamento dos dados sobre os dados, ou seja, os metadados. Os logs de transações que servem para rastrear a atividade do usuário em um cluster Hadoop são um exemplo de metadados. Os NameNodes gerenciam os DataNodes por meio das operações de abrir, excluir, criar, replicar, renomear e fechar arquivos.

# Hadoop - Arquitetura - HDFS - Metadados

---

## Metadados

Contêm informações sobre os arquivos, como nomes, tamanhos e informações sobre a localização (número ou ids do bloco) do DataNode em que o NameNode faz o armazenamento. Essas informações são úteis para encontrar o DataNode mais próximo. Como consequência, as operações de comunicação ficam mais rápidas.

# Hadoop - Arquitetura - HDFS - Datanode

---

## DataNode

Desempenha o papel de “escravo”. A principal utilização dos DataNodes é armazenar os dados em um cluster Hadoop. A quantidade de DataNodes pode ser muito grande, aumentando, assim, a capacidade de armazenamento que o cluster Hadoop pode realizar.

# Hadoop - Arquitetura - Componentes

---

## YARN (Yet Another Resource Negotiator)

YARN é o componente estrutural sobre o qual funciona o MapReduce. Ele realiza duas operações distintas:

### Agendamento de tarefas

O objetivo é dividir uma grande tarefa em tarefas menores para que elas possam ser atribuídas a vários nós escravos em um cluster do Hadoop. Dessa forma, o desempenho do processamento será maximizado. É o agendador de tarefas que faz o controle das prioridades de execução das tarefas, considerando aspectos, como sua importância e dependências em relação às demais tarefas e quaisquer outras informações, como o tempo para conclusão do trabalho, por exemplo.

# Hadoop - Arquitetura - Componentes

---

## Gerenciamento de recursos - ResourceManager

Faz o controle de todos os recursos disponibilizados para a execução de uma tarefa em um cluster Hadoop.

Relacionado ao gerenciador de recursos está o gerenciador de dados (NodeManager), que é o agente de estrutura por máquina responsável pelos containers por meio do monitoramento do uso de recursos (CPU, memória, disco e rede) e do envio de relatórios de uso para o gerenciador de recursos.

# Hadoop - Arquitetura - Utilitários

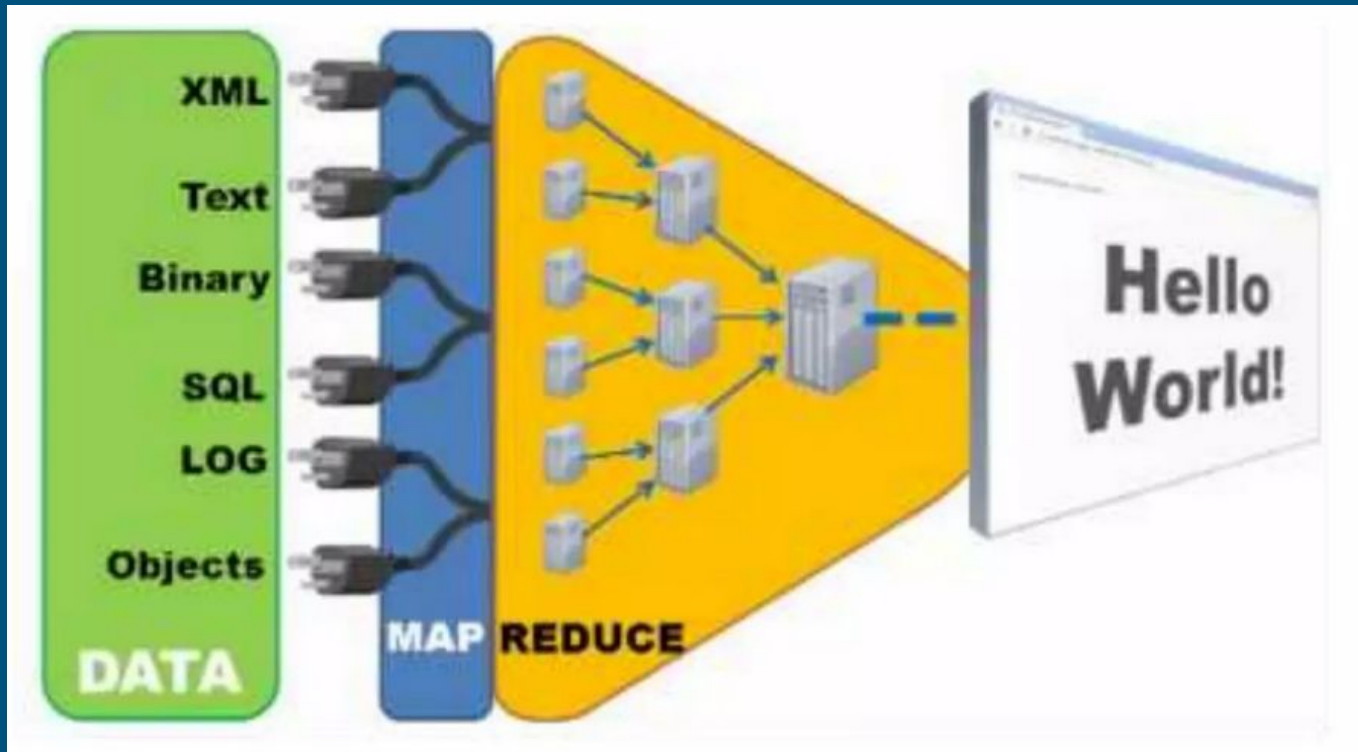
---

## Utilitários comuns do Hadoop (Hadoop Common)

Os utilitários comuns do Hadoop são as bibliotecas e aplicações que oferecem suporte para ele. Eles são usados para executar as aplicações no cluster Hadoop pelos componentes HDFS, YARN e MapReduce. De forma semelhante à que ocorre nos demais módulos do Hadoop, os utilitários assumem que as falhas de hardware são comuns e que devem ser tratadas automaticamente no software pelo Hadoop Framework.

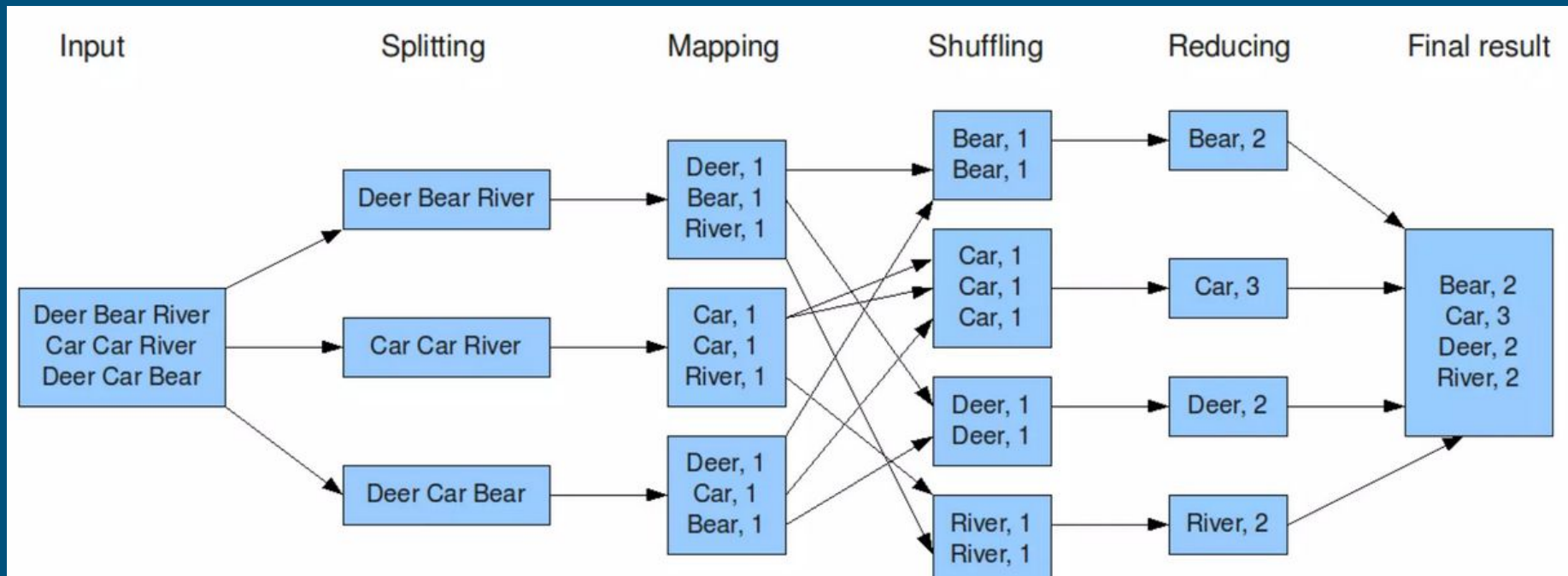
# Hadoop - Arquitetura - Me ajuda???

---





# Hadoop - Arquitetura - Me ajuda II ???



# Hadoop - Vantagens

---

## Escalabilidade

O Hadoop foi projetado desde o início para trabalhar com grandes volumes de dados. Para isso, os componentes da sua arquitetura lidam com diferentes aspectos do armazenamento e do processamento de dados distribuídos em diferentes nós da infraestrutura que aplicamos na solução.

## Redução de custos

A distribuição Apache do Hadoop é de um software livre. Além disso, ele não requer uma infraestrutura de hardware especial, podendo utilizar equipamentos comuns.

## Flexibilidade

O Hadoop é capacitado para trabalhar com diferentes tipos de dados tanto estruturados como não estruturados. Dessa forma, as empresas podem aplicar suas estratégias para gerar valor a partir da composição e da análise desses dados.

# Hadoop - Vantagens

---

## **Velocidade**

Os componentes da arquitetura do Hadoop, como o HDFS e o MapReduce, são projetados respectivamente para gerenciar e processar dados com a aplicação de estruturas de dados e estratégias de algoritmos que otimizam a operação dos processos.

## **Tolerância a falhas**

O Hadoop utiliza um processo de replicação dos dados entre os nós do cluster de modo que, se houver falha em algum nó, haverá outra cópia disponível para uso.

# Hadoop - desvantagens

---

## Preocupações de segurança

Devido à complexidade das aplicações de Big Data de modo geral, os aspectos relacionados à segurança são um grande desafio. No caso do Hadoop, esse desafio está longe de ser trivial. Por exemplo, o modelo de segurança dele é desabilitado por padrão. Portanto, é da responsabilidade de quem vai gerenciar a infraestrutura da plataforma fazer a habilitação do módulo de segurança; caso contrário, os dados correrão um grande risco. Também é necessário tratar explicitamente de aspectos de criptografia dos dados.

## Vulnerabilidade intrínseca

O Hadoop foi desenvolvido na linguagem de programação Java. Existem diversos casos já catalogados de quebra de segurança do Hadoop, como escalonamento de privilégios e acesso não autorizado a senhas. Tudo isso ocorre devido à complexidade das aplicações de Big Data. O profissional que trabalha com os aspectos de segurança e controle de vulnerabilidades precisa conhecer muito bem a arquitetura do Hadoop e estudar constantemente os fóruns oficiais sobre esse tema, que é bastante dinâmico.

# Hadoop - desvantagens

---

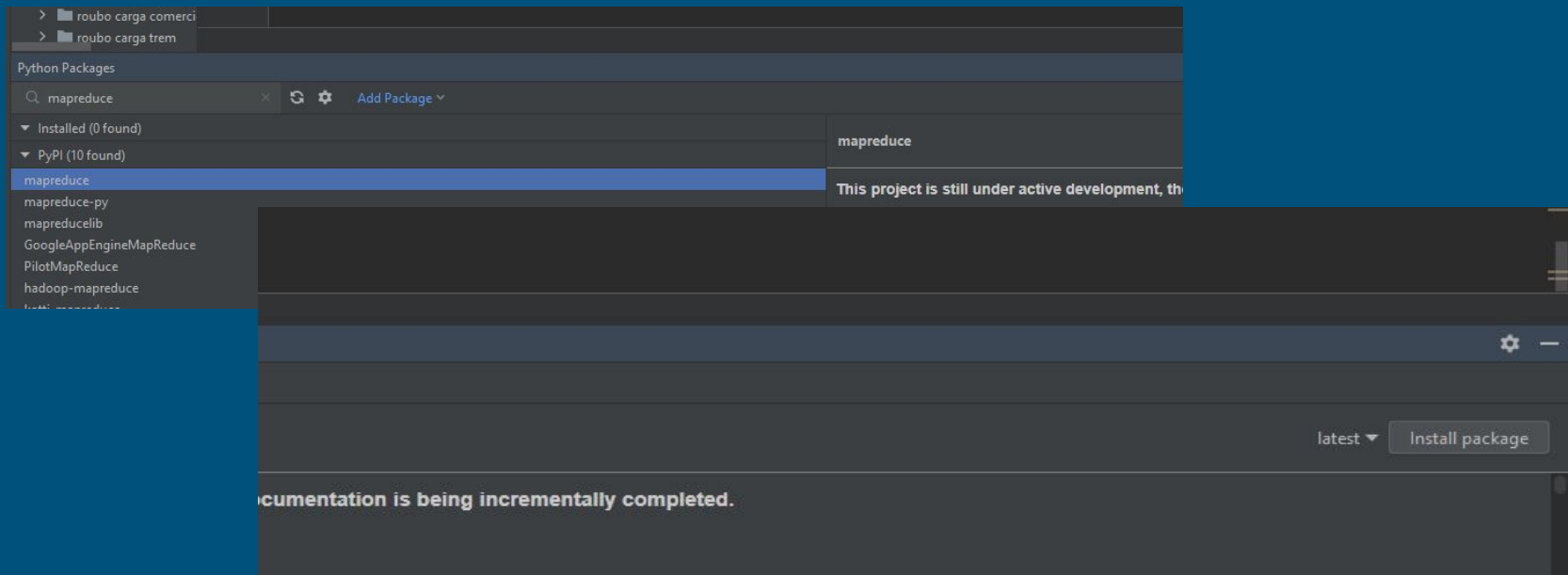
## Não é adequado para dados pequenos

O Hadoop foi projetado para trabalhar com grandes volumes de dados. Infelizmente, isso significa que ele não é uma boa opção para trabalhar com pequenos volumes. Isso parece ser contraditório, mas, na verdade, não é. Os componentes HDFS e MapReduce utilizam técnicas eficientes para manipular muitos dados. Isso implica que as estruturas de dados e os algoritmos são dimensionados com essa finalidade e que, se essas técnicas forem usadas para trabalhar com pequenos volumes, serão ineficientes em relação a soluções mais simples.

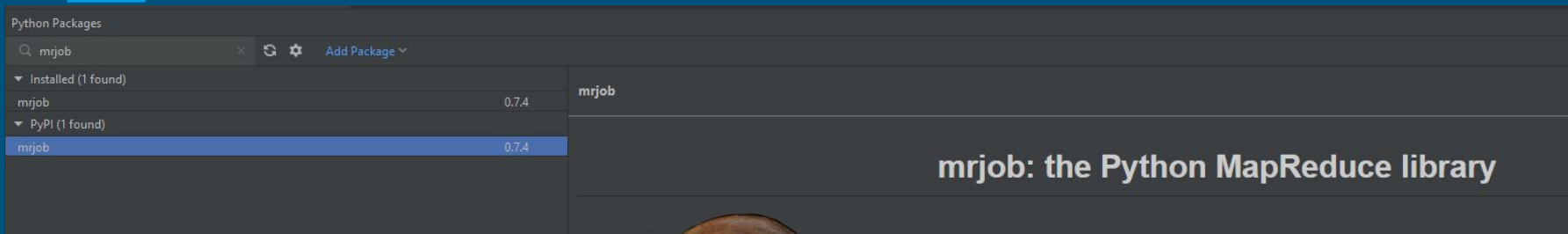
## Problemas de estabilidade

O Hadoop está em constante evolução e tem versões distribuídas por vários fornecedores. Por isso, não é raro que ocorram problemas relacionados à estabilidade da plataforma. Mais uma vez, isso reforça a necessidade de ter um profissional focado em aspectos da arquitetura e infraestrutura do Hadoop e que esteja se atualizando constantemente nos canais oficiais e fóruns de usuários e analistas.

# Hadoop - Mãos a obra - Instalar MapReduce



# Hadoop - Mãos a obra - Instalar MapReduce



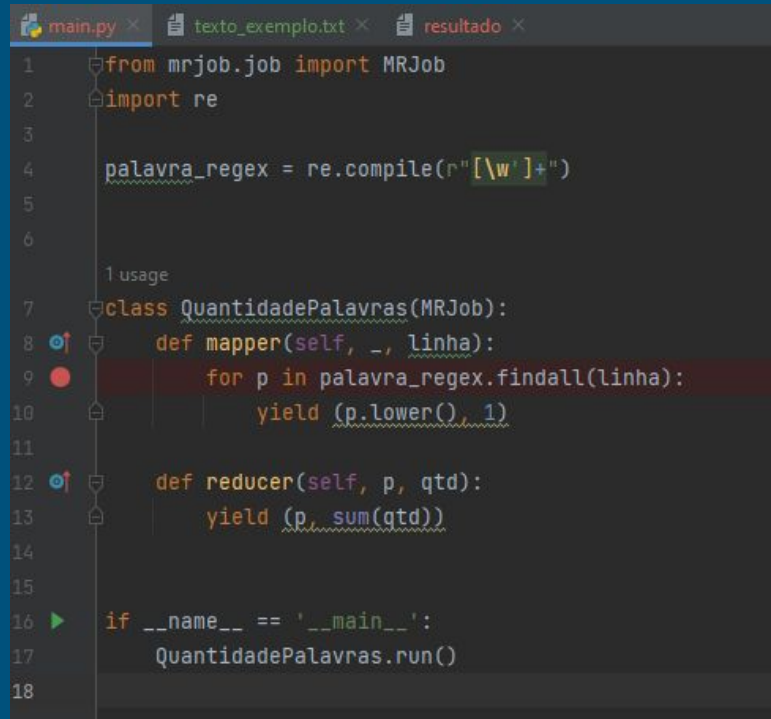
# Hadoop - Mãos a obra - Instalar MapReduce

---

```
(venv) PS C:\Users\fcarr\PyscharmProjects\hadoop> python -m pip install setuptools
Requirement already satisfied: setuptools in c:\users\fcarr\pyscharmprojects\hadoop\venv\lib\site-packages (60.2.0)
WARNING: You are using pip version 21.3.1; however, version 24.2 is available.
```



# Hadoop - Mãos a obra - Codificar

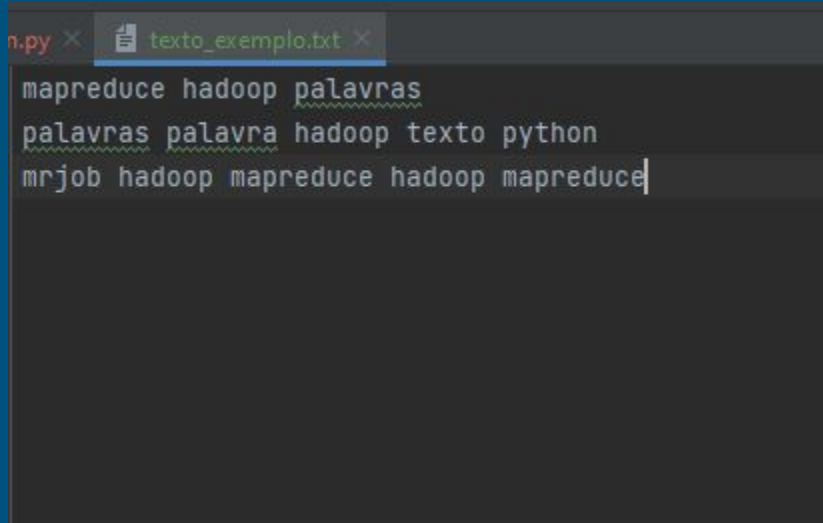


The screenshot shows a code editor with three tabs: `main.py`, `texto_exemplo.txt`, and `resultado`. The `main.py` file contains the following Python code:

```
1 from mrjob.job import MRJob
2 import re
3
4 palavra_regex = re.compile(r"[\w^st]+" )
5
6
7 1 usage
8 class QuantidadePalavras(MRJob):
9     def mapper(self, _, linha):
10         for p in palavra_regex.findall(linha):
11             yield (p.lower(), 1)
12
13     def reducer(self, p, qtd):
14         yield (p, sum(qtd))
15
16 if __name__ == '__main__':
17     QuantidadePalavras.run()
18
```

# Hadoop - Mãos a obra - Criar “base de dados”

---

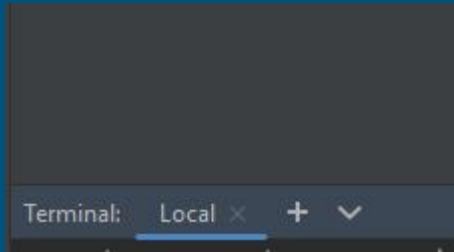


A screenshot of a code editor window with a dark background. The editor has two tabs at the top: 'n.py' and 'texto\_exemplo.txt'. The 'texto\_exemplo.txt' tab is active. The code in the editor is as follows:

```
mapreduce hadoop palavras  
palavras palavra hadoop texto python  
mrjob hadoop mapreduce hadoop mapreduce|
```

# Hadoop - Mãos a obra - Executar

---



```
python .\main.py .\texto_exemplo.txt > resultado.clear
```

```
Running Step 1 of 1...  
Job output is in C:\Users\fc card\AppData\Local\Temp\main.fc card.20230821.205914.032766\output  
Streaming final output from C:\Users\fc card\AppData\Local\Temp\main.fc card.20230821.205914.032766\output...  
Removing temp directory C:\Users\fc card\AppData\Local\Temp\main.fc card.20230821.205914.032766...
```

# Hadoop - Mãos a obra - Resultado

