



**Olá, aluno(a)!**  
**Seja bem-vindo(a) à aula interativa!**

Você entrará na reunião com a câmera e o microfone desligados.

Sua presença será computada através da enquete.  
Fique atento(a) e não deixe de respondê-la!



# Processamento de Fluxos Contínuos de Dados

Primeira Aula Interativa

Prof. Dr. Pedro Calais

# Quem sou eu?

- ❑ Doutor, Ciência da Computação, UFMG (2015)
- ❑ Engenheiro de software @ Stone
- ❑ Estudando economia da escola austríaca @ Instituto Mises Brasil
- ❑ Professor de ciência de dados @ XPE



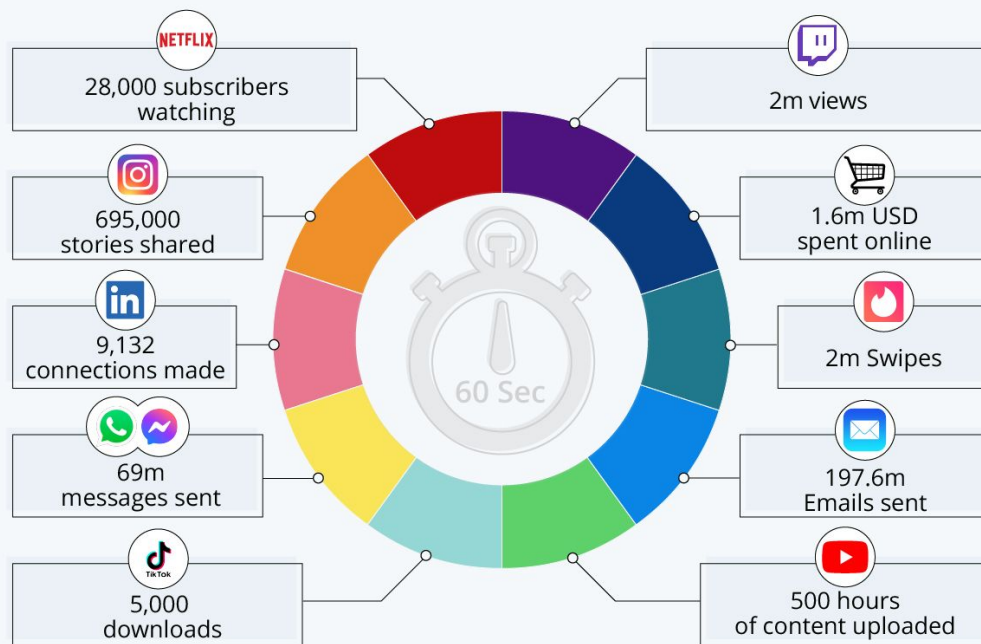
# Qual nosso objetivo hoje?

- Vamos revisar a **teoria** da primeira parte do curso
  - big data
  - processamento massivo de dados
  - engenharia de dados
  - Apache Spark
- **Vamos para a prática:** escrever código usando o Spark!

# Big, very big data!

## A Minute on the Internet in 2021

Estimated amount of data created on the internet in one minute



Source: Lori Lewis via AllAccess



statista

- Necessidade de ferramentas que extraem
  - valor dados segue crescendo
- Organizações usam dados para
  - melhorar serviços;
  - reduzir custos;
  - tomar melhores decisões.



# Aplicações de *Big Data*



# Aplicação: comércio eletrônico

## Benefits of Big Data in e-commerce sphere



Trend prediction



Price formation



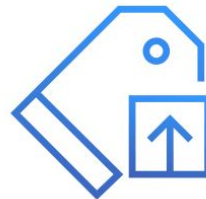
Demand forecast



Personalized approach

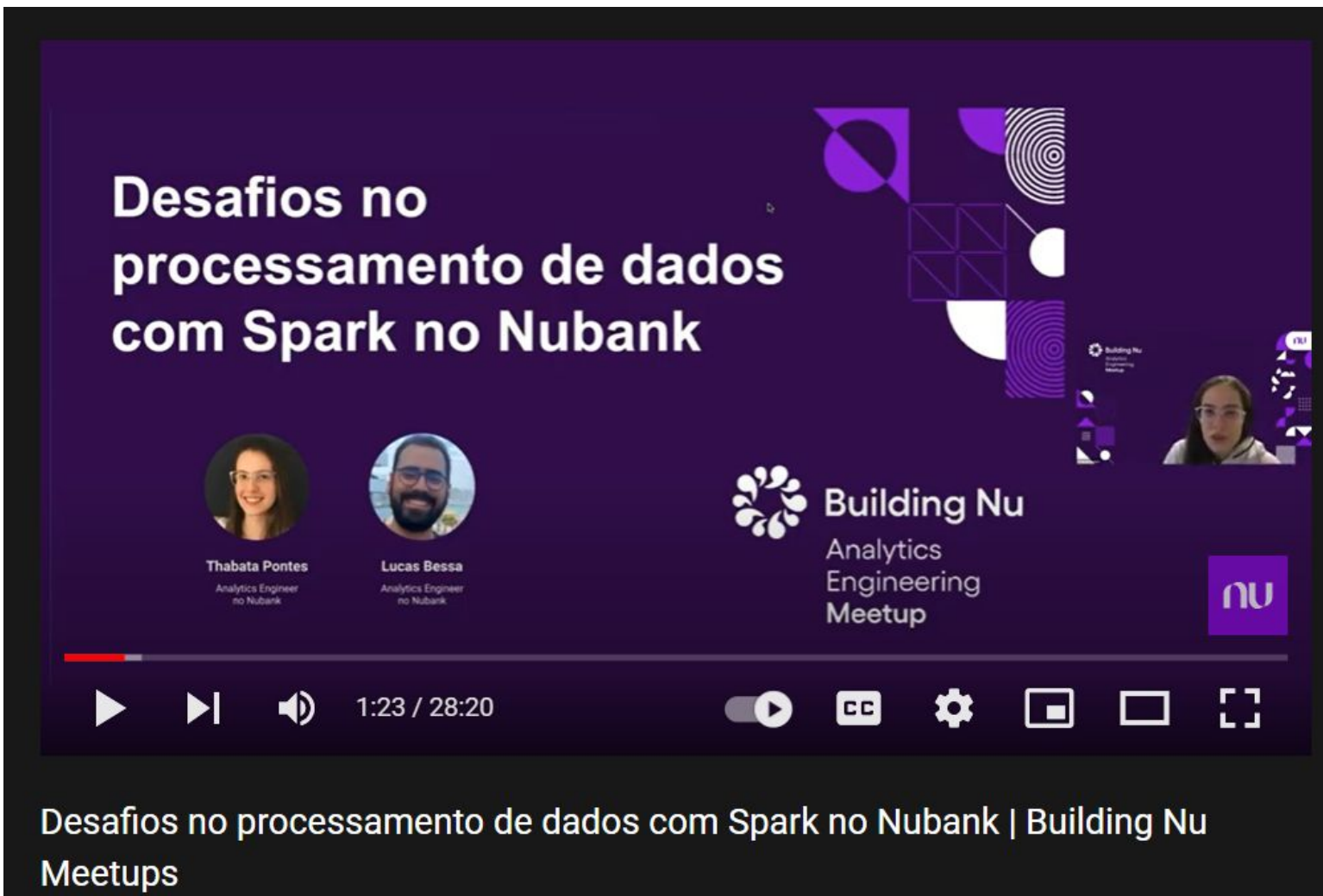


Service Improvement



Sales increase

# Aplicação: Nubank



The screenshot shows a video player interface with a dark purple background. The main content is a presentation slide with the title "Desafios no processamento de dados com Spark no Nubank" in white text. Below the title, there are two circular profile pictures of speakers: Thabata Pontes and Lucas Bessa, both identified as "Analytics Engineer no Nubank". To the right of the speakers, there is a logo for "Building Nu Analytics Engineering Meetup" and a small video feed of a person. The video player controls at the bottom include a progress bar, play/pause button, volume icon, and a timestamp of 1:23 / 28:20. The Nubank logo is visible in the bottom right corner of the slide.

**Desafios no processamento de dados com Spark no Nubank**

Thabata Pontes  
Analytics Engineer  
no Nubank

Lucas Bessa  
Analytics Engineer  
no Nubank

Building Nu  
Analytics Engineering  
Meetup

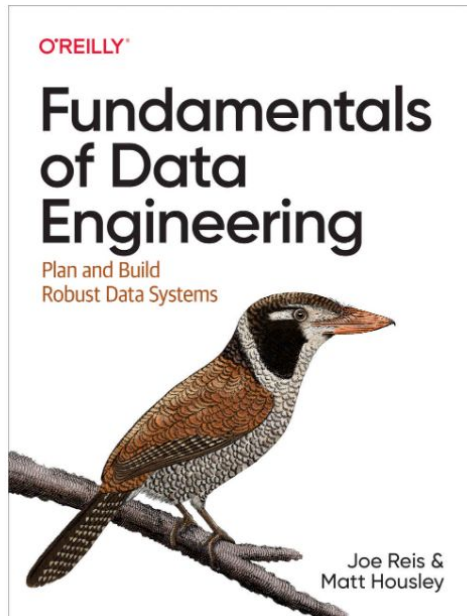
1:23 / 28:20

Desafios no processamento de dados com Spark no Nubank | Building Nu Meetups



# Os desafios técnicos

- Dados precisam ser coletados, preparados, mantidos e geridos.
- A confiabilidade e qualidade dos dados precisa ser mínima.
- Dados brutos precisam ser transformados em formatos úteis para análise.



# O desafio técnico

- Dados não cabem na memória de um único computador;
- Dados são gerados em uma taxa muito alta, e queremos interpretá-los e reagir a eles em tempo real;
- Sistemas tradicionais como banco de dados relacionais não escalam para o tamanho “Google”.

Como resolvemos os desafios?  
Com ferramentas!



igti

# A solução: computação distribuída

- Dividimos um conjunto de dados grandes em conjuntos menores;
- Processamos os dados de forma distribuída, em múltiplas máquinas;
- Google: 2,5 milhões de servidores.







# O que é o Apache Spark?

Uma ferramenta poderosa para engenharia e ciência de dados

# O que é o Apache Spark?



what is apache spark

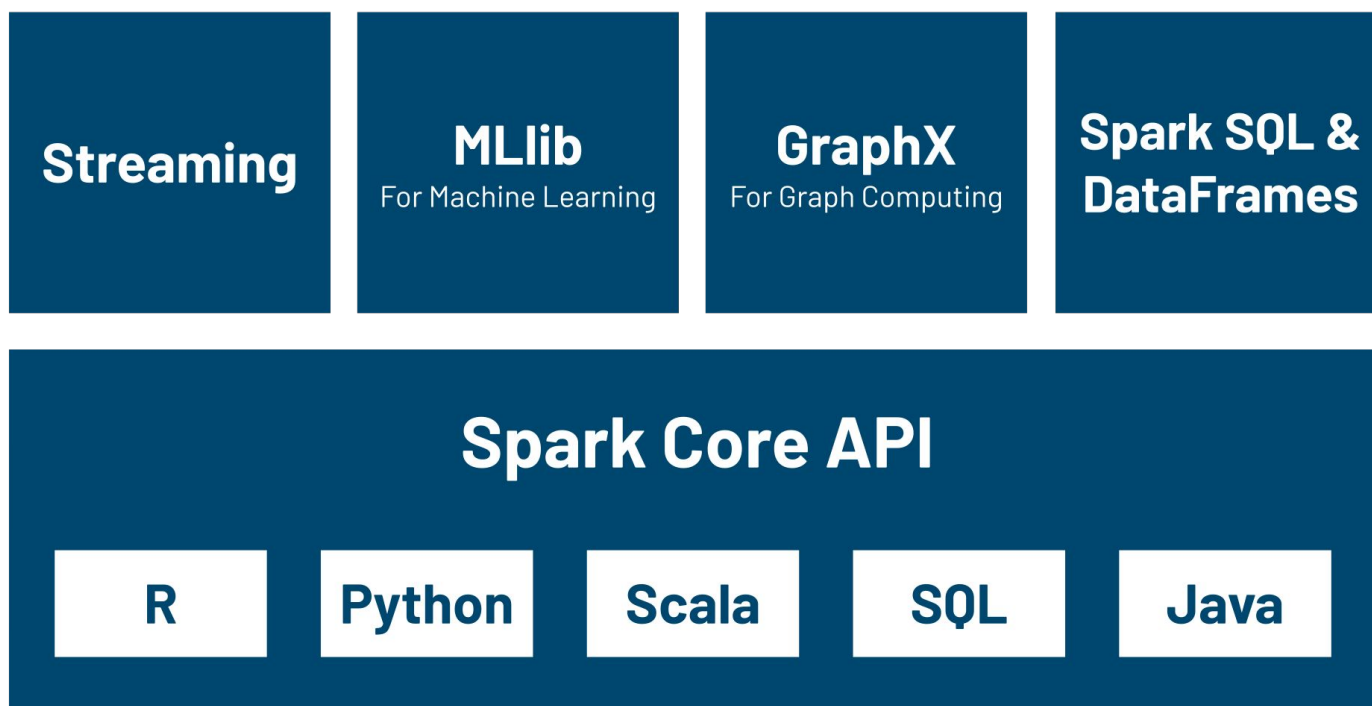
- Big data platform
- Data processing framework
- General-purpose distributed data processing engine
- Lightning-fast cluster computing technology
- Lightning-Fast Unified Analytics for Big Data and Machine Learning
- Unified engine for large-scale data analytics
- Open-source unified analytics engine for large-scale data processing
- Open-source, distributed processing system used for big data workloads

# O que o Spark entrega pra você?

- **Paralelismo implícito**
  - programador não precisa se preocupar em dividir e coordenar a computação entre as máquinas;
  - aumento de produtividade do programador.
- **Tolerância a falhas**
  - capacidade do sistema operar mesmo após falhas;
  - redundância e replicação.

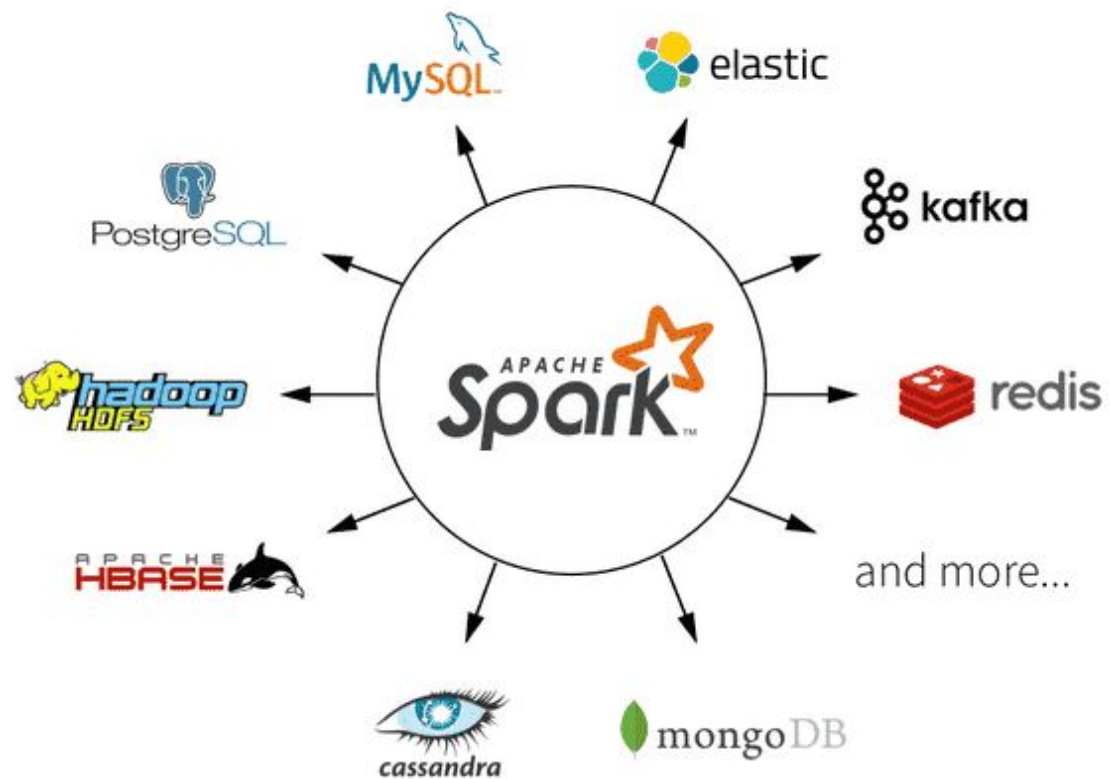


# O Spark é um ecossistema

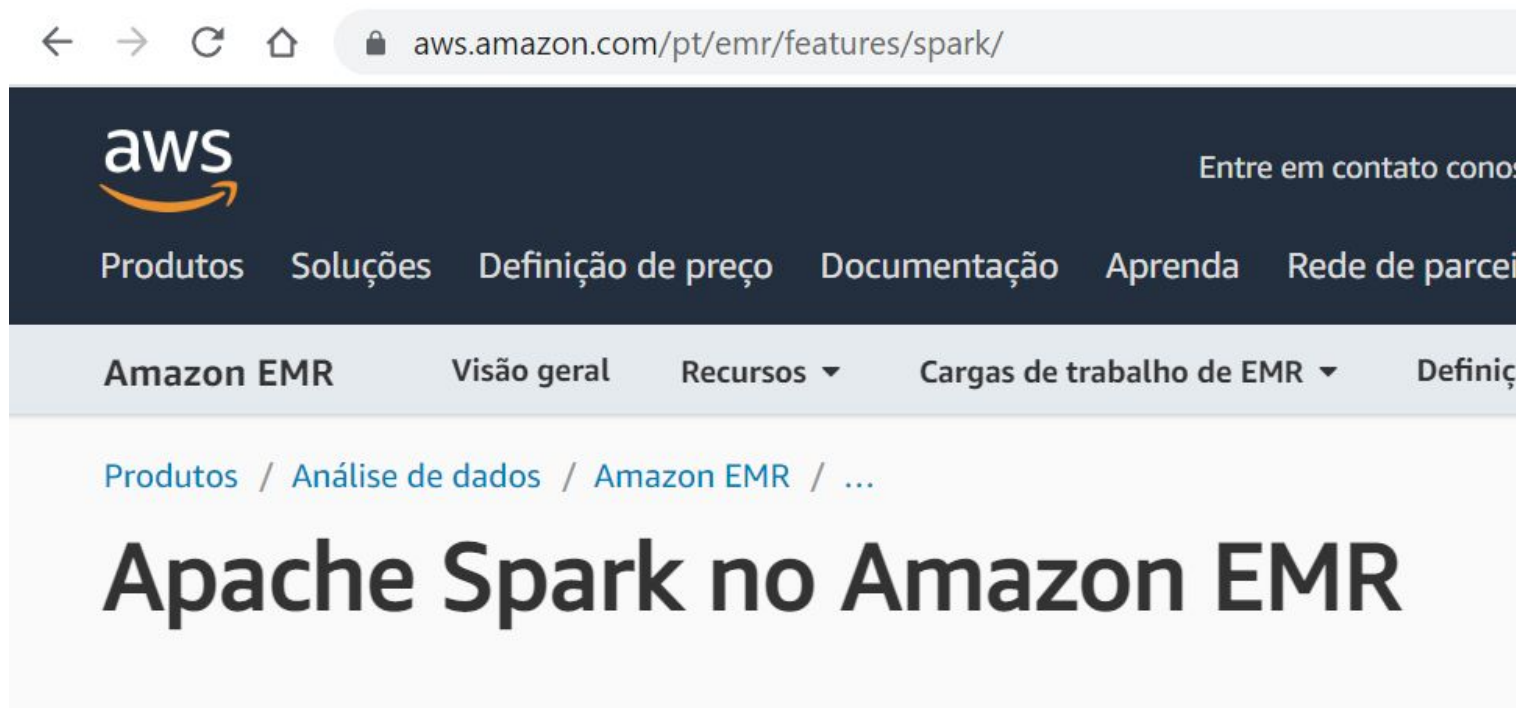




# O Spark é um ecossistema



# Spark na nuvem: AWS



# Spark na nuvem: Google Cloud

Google Cloud

Por Que O Google?

Soluções

Produtos

Preços >



Documentos

Suporte



Lan

[Entre em contato com nossa equipe](#)

Assista ao [Data Cloud Summit](#) sob demanda e saiba mais sobre as inovações mais recentes em análise, IA, BI e

## Spark no Google Cloud

Visão geral

Vantagens

Principais recursos

## Spark no Google Cloud

O primeiro Spark sem servidor e com escalonamento automático integrado ao melhor das ferramentas nativas e de código aberto do Google. Desenvolva e execute o Spark onde você precisar em todos os casos de uso, incluindo ETL, ciência de dados e exploração.

# Spark na nuvem: Azure



Azure

Explore ▾

Products ▾

Solutions ▾

Pricing ▾

Pa

[Home](#) / [Services](#) / Azure Databricks

## Azure Databricks

Design AI with Apache Spark™-based analytics

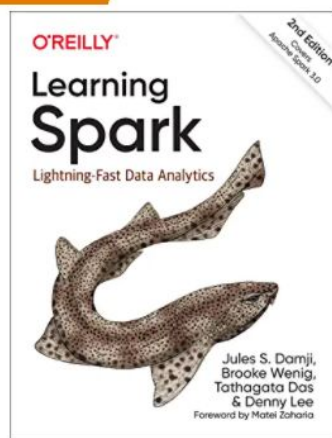
Start free

Already an Azure customer? [Get started >](#)



# Popularidade do Spark

Mais vendido



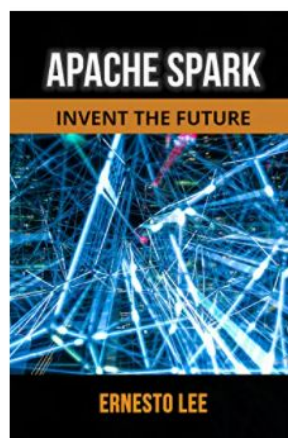
**Learning Spark: Lightning-Fast Data Analytics**

Edição Inglês  
por Jules S Damji, Brooke Wenig, e  
outros.

★★★★★ ~ 60

**Capa Comum**

R\$375<sup>90</sup>



**APACHE SPARK: INVENT THE FUTURE (English Edition)**

Edição Inglês  
por ERNESTO LEE

**Kindle**

R\$0<sup>00</sup> **kindleunlimited**

Gratuito com assinatura ilimitada do  
Kindle [Saiba mais](#)



**High Performance Spark: Best Practices for Scaling and...**

Edição Inglês  
por Holden Karau e Rachel Warren

★★★★★ ~ 42

**Capa Comum**

R\$180<sup>69</sup>

em até 6x de R\$ 30,14 sem juros



**Spark in Action: Covers Apache Spark 3 with Examl...**

Edição Inglês  
por Jean-Georges Perrin

★★★★★ ~ 17

**Capa Comum**

R\$439<sup>18</sup>

em até 10x de R\$ 43,99 sem juros

# O que torna o Spark especial?



desempenho



facilidade de usar



plataforma unificada

# Quem usa o Spark?



engenheiros de dados



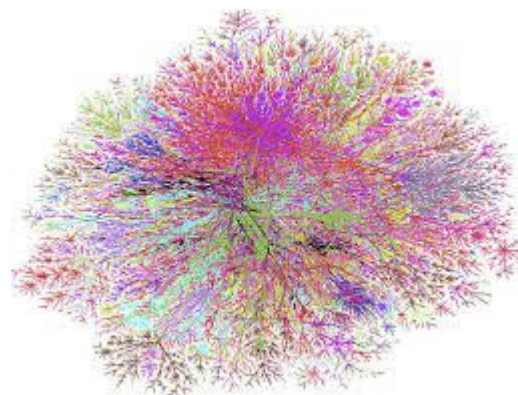
cientistas de dados



engenheiros de *machine learning*

# Estudo de Caso

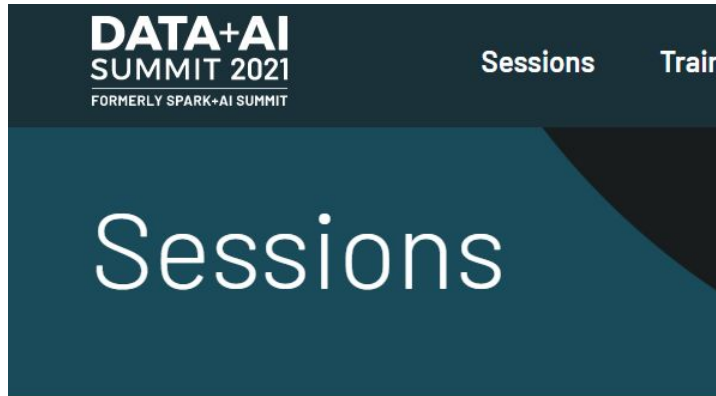
WORLDSENSE  
HYPER YOUR CONTENT



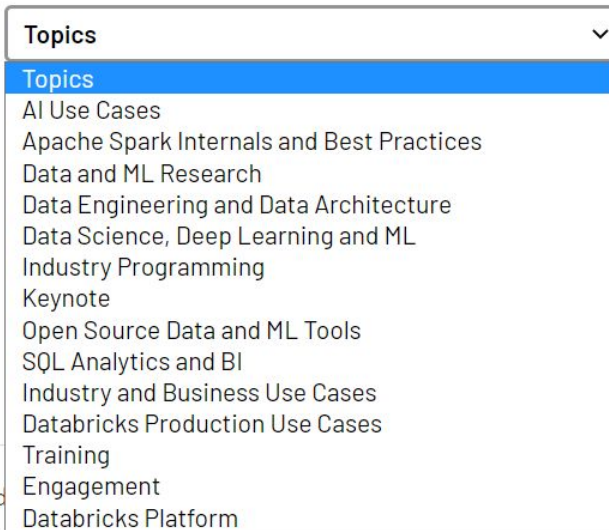
- 40 TB de dados;
- ~2 bilhões de documentos Web;
- ~ 10 bilhões de links;
- A WorldSense processava os dados em algumas horas em máquinas da AWS.



databricks.com



databricks



igti

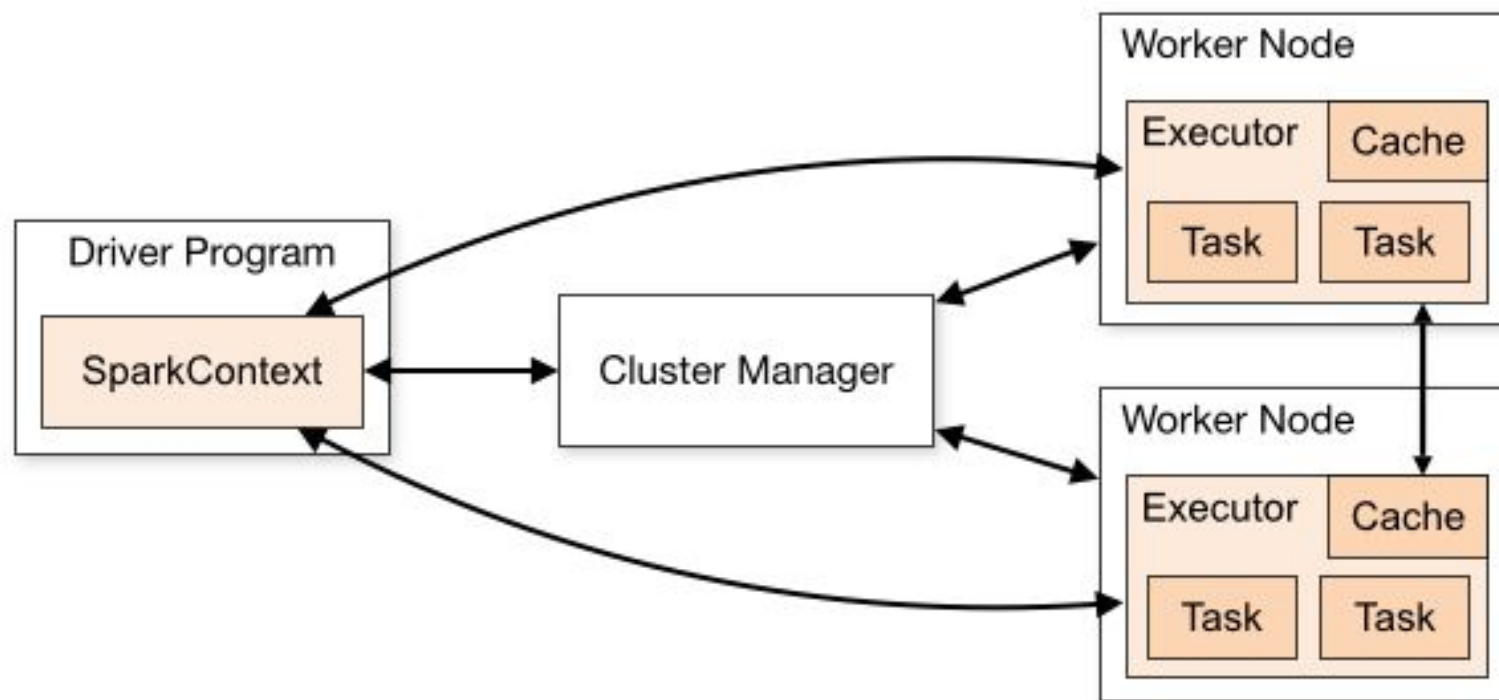
# Download e instalação do Spark

<https://spark.apache.org/downloads.html>

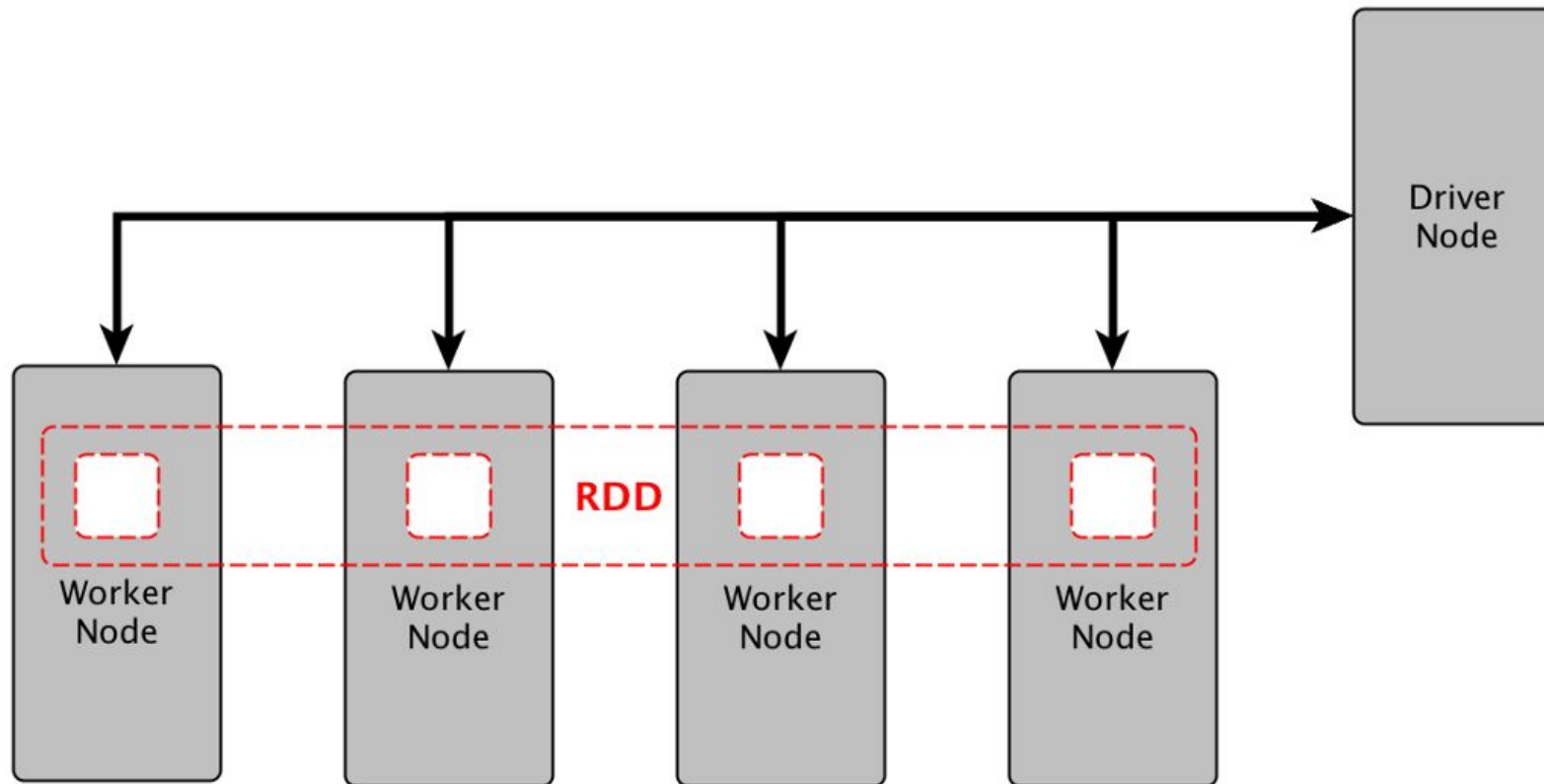
## Download Apache Spark™

1. Choose a Spark release:  ▼
2. Choose a package type:  ▼
3. Download Spark: [spark-3.2.1-bin-hadoop3.2.tgz](#)

# Arquitetura do Spark



# RDDs e operações distribuídas



fonte: <https://medium.com/@lavishj77/spark-fundamentals-part-2-a2d1a78eff73>

# Um programa Spark é um conjunto de transformações e ações

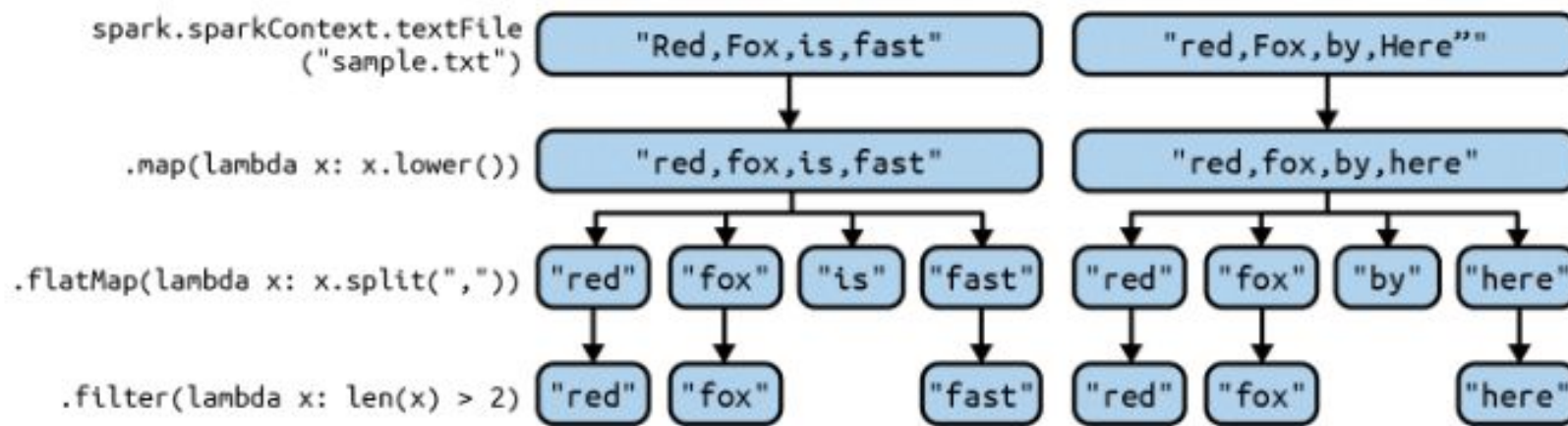
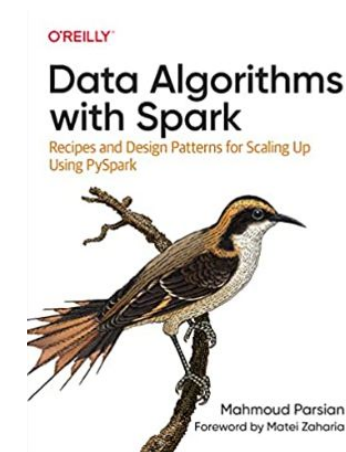


Figure 1-1. Simple RDD transformations



Um programa Spark é um conjunto de transformações e ações

```
// Count errors mentioning MySQL:  
errors.filter(_.contains("MySQL")).count()
```



# Um programa Spark é um conjunto de transformações e ações

## Transformations

map	join	union	distinct	repartition
mapPartitions	flatMap	intersection	pipe	coalesce
cartesian	cogroup	filter	sample	
sortByKey	groupByKey	reduceByKey	aggregateByKey	
mapPartitionsWithIndex		repartitionAndSortWithinPartitions		

## Actions

reduce	take	collect	takeSample	count
takeOrdered	countByKey	first	foreach	saveAsTextFile
saveAsSequenceFile		saveAsObjectFile		

# Código com RDD às vezes se torna complexo

```
In [16]: 1 # Agregate all ages by name and get the average name by age.
2
3 # Create an RDD of tuples (name, age)
4 dataRDD = spark.sparkContext.parallelize([("Pedro", 38), ("Maria", 20), ("Pedro", 40), ("Rafael", 10)])
5
6 agesRDD = (dataRDD
7             .map(lambda x: (x[0], (x[1], 1)))
8             .reduceByKey(lambda x, y: (x[0] + y[0], x[1] + y[1]))
9             .map(lambda x: (x[0], x[1][0]/x[1][1])))
10
11 agesRDD.collect()
```

```
Out[16]: [('Pedro', 39.0), ('Maria', 20.0), ('Rafael', 10.0)]
```

# Dataframe: quase uma tabela relacional

```
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|Direction|Year|      Date| Weekday|Country|Commodity|Transport_Mode|Measure|      Value|Cumulative|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|  Exports|2015|01/01/2015|Thursday|   All|    All|         All|   $|104000000|104000000|
|  Exports|2015|02/01/2015|  Friday|   All|    All|         All|   $| 96000000|200000000|
|  Exports|2015|03/01/2015|Saturday|   All|    All|         All|   $| 61000000|262000000|
|  Exports|2015|04/01/2015|  Sunday|   All|    All|         All|   $| 74000000|336000000|
|  Exports|2015|05/01/2015|  Monday|   All|    All|         All|   $|105000000|442000000|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
only showing top 5 rows
```

# Dataframe tem esquema!

---

```
root
|-- Direction: string (nullable = true)
|-- Year: string (nullable = true)
|-- Date: string (nullable = true)
|-- Weekday: string (nullable = true)
|-- Country: string (nullable = true)
|-- Commodity: string (nullable = true)
|-- Transport_Mode: string (nullable = true)
|-- Measure: string (nullable = true)
|-- Value: string (nullable = true)
|-- Cumulative: string (nullable = true)
```

# A API de Dataframes

```
# Create a new DataFrame that contains "young users" only
young = users.filter(users.age < 21)

# Alternatively, using Pandas-like syntax
young = users[users.age < 21]

# Increment everybody's age by 1
young.select(young.name, young.age + 1)

# Count the number of young users by gender
young.groupBy("gender").count()

# Join young users with another DataFrame called logs
young.join(logs, logs.userId == users.userId, "left_outer")
```





Vamos escrever um pouco de código?