

# Entendendo o Big Data

(baseado em Guller (2015) - Big Data Analytics with Spark A Practitioner's Guide to Using Spark for Large-Scale Data Processing, Machine Learning, and Graph Analytics, and High-Velocity Data Stream Processing)



# Falando sobre o Cenário de Big Data

Estamos na era do big data. Os dados não só se tornaram a força vital de qualquer organização, mas também estão crescendo exponencialmente.

Os dados gerados hoje são várias magnitudes maiores do que os gerados há apenas alguns anos. O desafio é como extrair valor comercial desses dados.

Esse é o problema que as tecnologias relacionadas a big data visam resolver.

Portanto, big data se tornou uma das tendências de tecnologia mais recentes nos últimos anos.

Alguns dos projetos de código aberto mais ativos estão relacionados a big data, e o número desses projetos está crescendo rapidamente.

O número de startups focadas em big data explodiu nos últimos anos.

Grandes empresas estabelecidas estão fazendo investimentos significativos em tecnologias de big data

# E o que é Big Data mesmo ?

Embora o termo “big data” seja popular, sua definição é vaga.

Uma definição está relacionada ao volume de dados; outra definição se refere à riqueza dos dados.

Alguns definem big data como dados “muito grandes” para os padrões tradicionais; enquanto outros definem big data como dados que capturam mais nuances sobre a entidade que representam.

Um exemplo do primeiro seria um conjunto de dados cujo volume excede petabytes ou vários terabytes. Se esses dados fossem armazenados em uma tabela de banco de dados relacional tradicional (RDBMS), eles teriam bilhões de linhas.

Um exemplo da última definição é um conjunto de dados com linhas extremamente largas. Se esses dados fossem armazenados em uma tabela de banco de dados relacional, eles teriam milhares de colunas.

Outra definição popular de big data são dados caracterizados por três Vs: volume, velocidade e variedade.

- Já discutimos o volume.
- Velocidade significa que os dados são gerados em uma taxa rápida.
- Variedade se refere ao fato de que os dados podem ser não estruturados, semiestruturados ou multiestruturados.

# Big Data x Bancos de Dados

Os bancos de dados relacionais padrão não podiam lidar facilmente com big data.

A tecnologia central para esses bancos de dados foi projetada há várias décadas, quando poucas organizações tinham petabytes ou mesmo terabytes de dados.

Hoje, não é incomum que algumas organizações gerem terabytes de dados todos os dias.

Não apenas o volume de dados, mas também a taxa em que estão sendo gerados está explodindo.

Portanto, havia a necessidade de novas tecnologias que pudessem não apenas processar e analisar um grande volume de dados, mas também ingerir um grande volume de dados em um ritmo rápido.

# Ainda falando de Big Data

Outros fatores importantes para as tecnologias de big data incluem escalabilidade, alta disponibilidade e tolerância a falhas a baixo custo.

A tecnologia para processar e analisar grandes conjuntos de dados foi amplamente pesquisada e está disponível na forma de produtos comerciais proprietários há muito tempo.

Por exemplo, bancos de dados MPP (processamento paralelo massivo) já existem há algum tempo.

Os bancos de dados MPP usam uma arquitetura “sharednothing”, onde os dados são armazenados e processados em um cluster de nós. Cada nó vem com seu próprio conjunto de CPUs, memória e discos.

Eles se comunicam por meio de uma interconexão de rede. Os dados são particionados em um cluster de nós. Não há contenção entre os nós, portanto, todos podem processar dados em paralelo.

Exemplos de tais bancos de dados incluem Teradata, Netezza, Greenplum, ParAccel e Vertica.

O Teradata foi inventado no final da década de 1970 e, na década de 1990, era capaz de processar terabytes de dados. No entanto, os produtos MPP proprietários são caros. Nem todo mundo pode comprá-los.

# Uma Tecnologia de Big Data Hadoop

Hadoop foi uma das primeiras tecnologias populares de big data de código aberto.

É um sistema tolerante a falhas escalonável para processar grandes conjuntos de dados em um cluster de servidores.

Ele fornece uma estrutura de programação simples para processamento de dados em grande escala usando os recursos disponíveis em um cluster de computadores.

O Hadoop é inspirado em um sistema inventado no Google para criar um índice invertido para seu produto de pesquisa.

Jeffrey Dean e Sanjay Ghemawat publicaram artigos em 2004 descrevendo o sistema que criaram para o Google. O primeiro, intitulado “MapReduce: Processamento de Dados Simplificado em Grandes Clusters” está disponível na pesquisa [neste link](https://research.google.com/archive/mapreduce.html) [. \(google.com/archive/mapreduce.html\)](https://research.google.com/archive/mapreduce.html).

O segundo, intitulado “O sistema de arquivos do Google” está disponível neste [link](https://research.google.com/archive/gfs.html) [. \(research.google.com/archive/gfs.html\)](https://research.google.com/archive/gfs.html).

Inspirados por esses documentos, Doug Cutting e Mike Cafarella desenvolveram uma implementação de código aberto, que mais tarde se tornou Hadoop.

# Mais Hadoop

Muitas organizações substituíram produtos comerciais proprietários caros pelo Hadoop para processar grandes conjuntos de dados. Um dos motivos é o custo.

O Hadoop é um software livre e executado em um cluster de hardware comum. Você pode escalá-lo facilmente adicionando servidores baratos.

Alta disponibilidade e tolerância a falhas são fornecidas pelo Hadoop, então você não precisa comprar hardware caro.

Em segundo lugar, é mais adequado para certos tipos de tarefas de processamento de dados, como processamento em lote e ETL (carga de transformação de extração) de dados em grande escala.



# Idéias e Conceitos por trás do Hadoop

O Hadoop é baseado em algumas ideias importantes.

Em primeiro lugar, é mais barato usar um cluster de servidores de commodity para armazenar e processar grandes quantidades de dados do que usar servidores poderosos de ponta. Em outras palavras, o Hadoop usa arquitetura scale-out em vez de arquitetura scale-up. Em segundo lugar, implementar tolerância a falhas por meio de software é mais barato do que implementá-la em hardware. Os servidores tolerantes a falhas são caros.

O Hadoop não depende de servidores tolerantes a falhas. Ele pressupõe que os servidores falharão e trata as falhas do servidor de maneira transparente. Um desenvolvedor de aplicativos não precisa se preocupar em lidar com falhas de hardware. Esses detalhes confusos podem ser deixados para o Hadoop lidar.

# Idéias e Conceitos por trás do Hadoop

Terceiro, mover o código de um computador para outro em uma rede é muito mais eficiente e rápido do que mover um grande conjunto de dados na mesma rede.

Por exemplo, suponha que você tenha um cluster de 100 computadores com um terabyte de dados em cada computador. Uma opção para processar esses dados seria movê-los para um servidor muito poderoso que pode processar 100 terabytes de dados.

No entanto, mover 100 terabytes de dados levará muito tempo, mesmo em uma rede muito rápida. Além disso, você precisará de hardware muito caro para processar dados com essa abordagem.

Outra opção é mover o código que processa esses dados para cada computador em seu cluster de 100 nós; é muito mais rápido e eficiente do que a primeira opção. Além disso, você não precisa de servidores de última geração, que são caros.

# Idéias e Conceitos por trás do Hadoop

Quarto, escrever um aplicativo distribuído pode ser fácil separando a lógica de processamento de dados principais da lógica de computação distribuída.

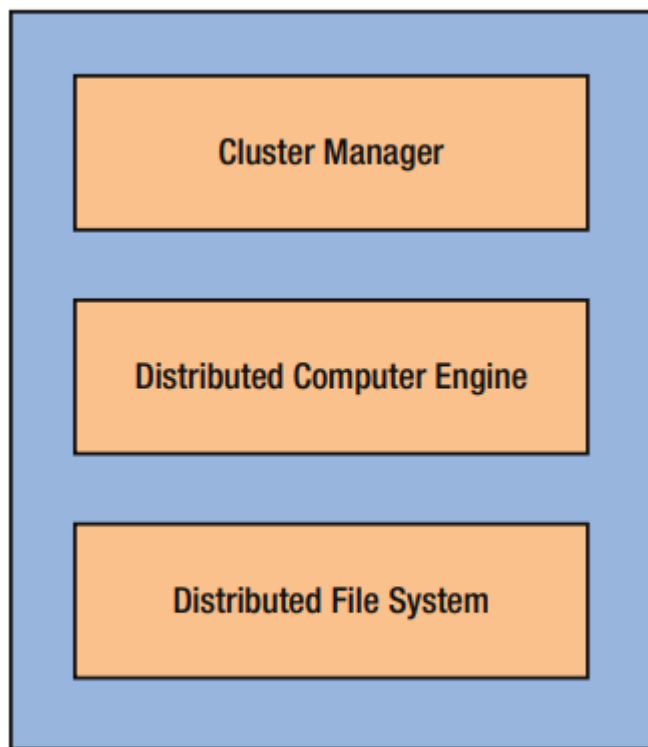
O desenvolvimento de um aplicativo que aproveita os recursos disponíveis em um cluster de computadores é muito mais difícil do que o desenvolvimento de um aplicativo executado em um único computador.

O conjunto de desenvolvedores que podem escrever aplicativos que rodam em uma única máquina é muito maior do que aqueles que podem escrever aplicativos distribuídos.

O Hadoop fornece uma estrutura que esconde as complexidades de escrever aplicativos distribuídos. Assim, permite que as organizações acessem um grupo muito maior de desenvolvedores de aplicativos.

# Idéias e Conceitos por trás do Hadoop

Embora as pessoas falem sobre o Hadoop como um único produto, não é realmente um único produto. Ele consiste em três componentes principais: um gerenciador de cluster, um mecanismo de computação distribuído e um sistema de arquivos distribuído (consulte a Figura 1) !

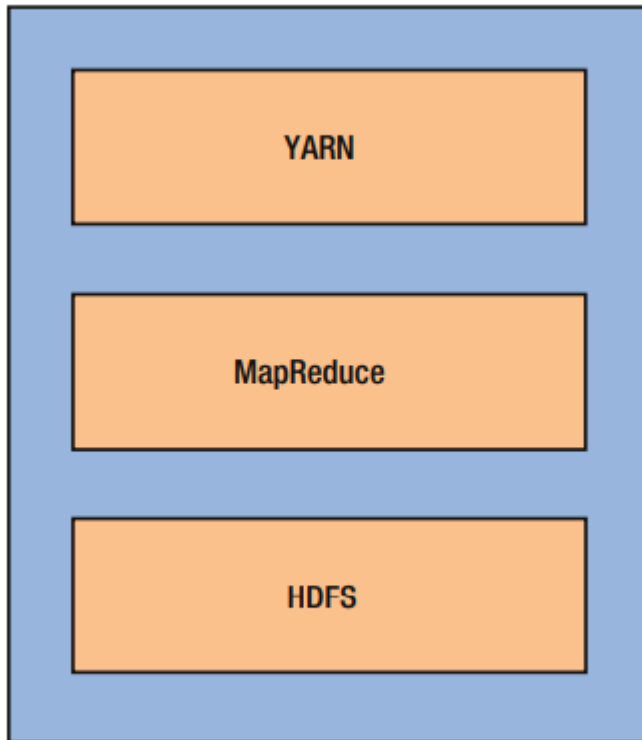


**Figure 1-1.** Key conceptual Hadoop components

Até a versão 2.0, a arquitetura do Hadoop era monolítica.

Todos os componentes foram fortemente acoplados e agrupados. A partir da versão 2.0, o Hadoop adotou uma arquitetura modular, que permite combinar e combinar componentes do Hadoop com tecnologias não-Hadoop.

As implementações concretas dos três componentes conceituais mostrados na Figura 1-1 são HDFS, MapReduce e YARN (consulte a Figura 2).



*Figure 1-2. Key Hadoop components*

**Na próxima aula continuamos com explicando  
conceitos e e mais componente do cenário Big Data**



## Introdução ao Spark

Este curso vai trabalhar bastante com o *Spark*, portanto faz todo o sentido começar examinando um pouco da história do Spark e seus diferentes componentes.

Esta aula introdutório está dividida em três seções.

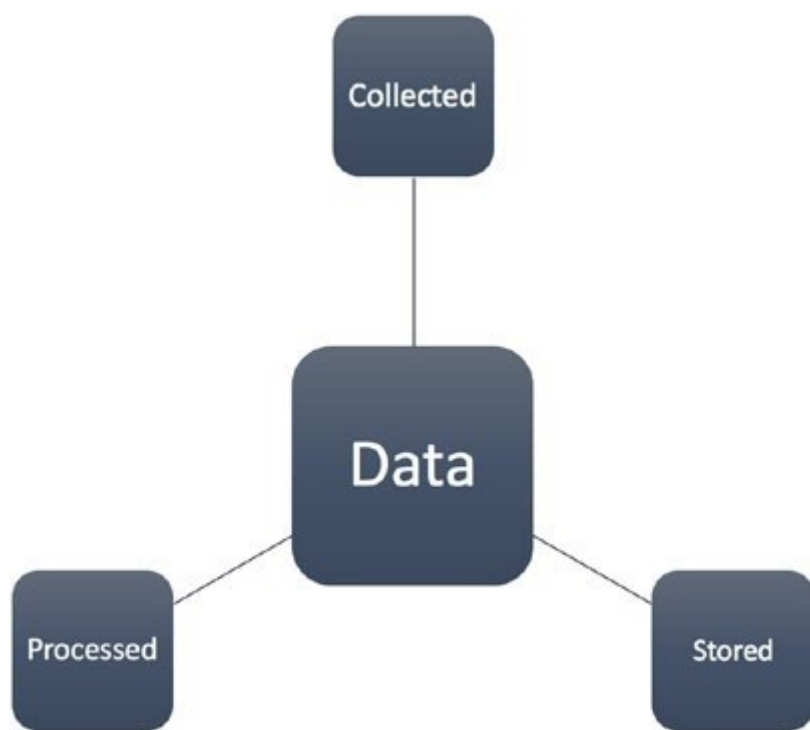
- Na primeira, examinaremos a evolução dos dados e como eles chegaram até onde chegaram, em termos de tamanho. Vamos abordar os três aspectos principais dos dados.
- Na segunda seção, vamos nos aprofundar na parte interna do *Spark* e examinar os detalhes de seus diferentes componentes, incluindo sua arquitetura e *modus operandi*.
- A terceira e última seção desta aula enfoca como usar o *Spark* em um ambiente de nuvem.

# História

O nascimento do projeto *Spark* ocorreu no Laboratório de Algoritmos, Máquina e Pessoas (AMP) da Universidade da Califórnia, Berkeley.

O projeto foi iniciado para resolver os possíveis problemas na estrutura **Hadoop MapReduce**. Embora o **Hadoop MapReduce** fosse um inovador *framework* para lidar com o processamento de **big data**, na realidade, ainda tinha muitas limitações em termos de velocidade. O *Spark* era novo e capaz de fazer cálculos na memória, o que o tornou quase 100 vezes mais rápido do que qualquer outra estrutura de processamento de **big data**. Desde então, tem havido um aumento contínuo na adoção do *Spark* em todo o mundo para aplicativos de **big data**. Mas antes de entrar nos detalhes do Spark, vamos considerar alguns aspectos dos dados em si. Os dados podem ser vistos de três ângulos diferentes: a maneira como são coletados, armazenados e processados, conforme mostrado na Figura abaixo:





# Coleta de dados

Uma grande mudança na maneira como os dados são coletados ocorreu nos últimos anos.

Desde comprar uma maçã em uma mercearia até excluir um aplicativo em seu telefone celular, cada ponto de dados agora é capturado no *back-end* e coletado por meio de vários aplicativos integrados.

Dispositivos diferentes da Internet das coisas (*IoT*) capturam uma ampla gama de sinais visuais e sensoriais a cada milissegundo.

Tornou-se relativamente conveniente para as empresas coletar esses dados de várias fontes e usá-los posteriormente para melhorar a tomada de decisões.

# Armazenamento de dados

Em anos anteriores, ninguém jamais imaginou que os dados residiriam em algum local remoto ou que o custo para armazenar dados seria tão barato quanto é. As empresas adotaram o armazenamento em nuvem e começaram a ver seus benefícios em relação às abordagens locais.

No entanto, algumas empresas ainda optam pelo armazenamento local, por vários motivos.

Sabe-se que o armazenamento de dados começou com o uso de fitas magnéticas. Então, a introdução inovadora dos disquetes tornou possível mover dados de um lugar para outro.

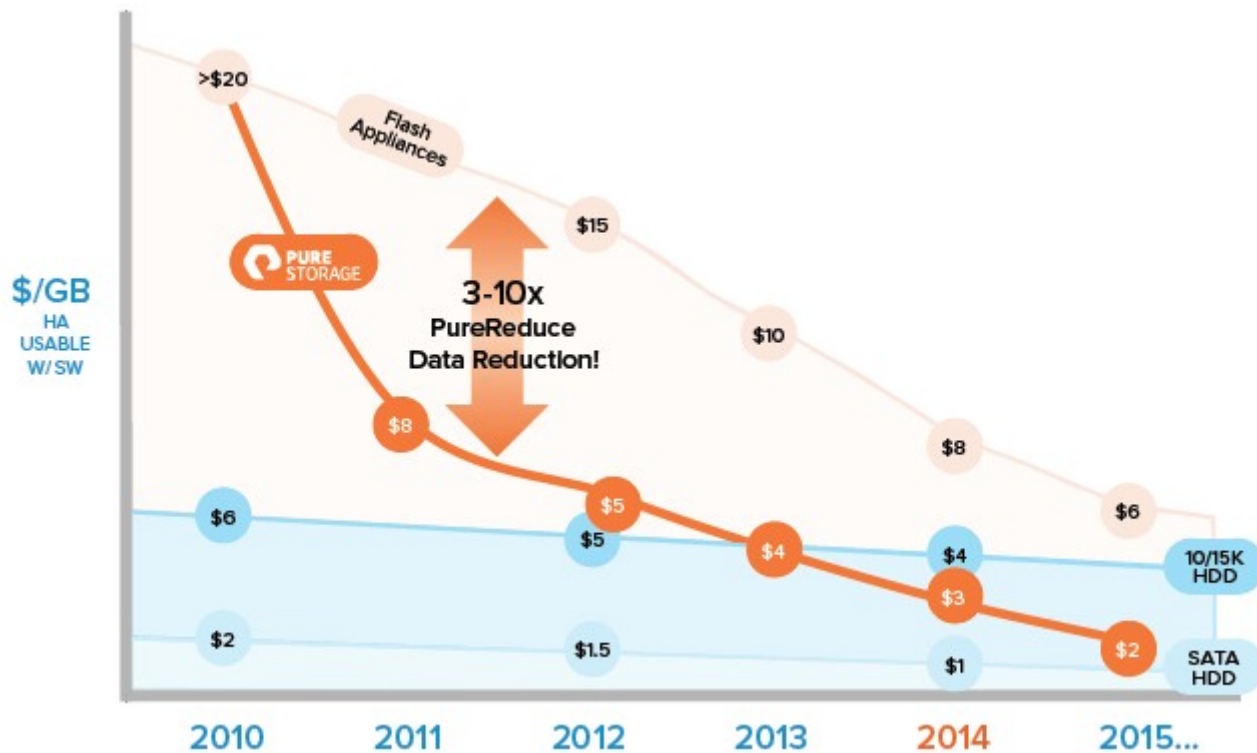
No entanto, o tamanho dos dados ainda era uma grande limitação. Unidades flash e discos rígidos tornaram ainda mais fácil armazenar e transferir grandes quantidades de dados a um custo reduzido. (Veja a Figura abaixo.)

A última tendência no avanço dos dispositivos de armazenamento resultou em drives flash capazes de armazenar dados de até 2 TBs, a um preço descartável



Esta tendência indica claramente que o custo para armazenar dados foi reduzido significativamente ao longo dos anos e continua a diminuir.

Como resultado, as empresas não se intimidam em armazenar grandes quantidades de dados, independentemente do tipo. De registros a transações financeiras e operacionais a comentários simples de funcionários, tudo é armazenado.



Custos de Armazenamento de dados Fonte: [www.enterpriseai.news>]

# Processamento de dados

O aspecto final dos dados é usar dados armazenados e processá-los para alguma análise ou para executar um aplicativo. Testemunhamos como os computadores se tornaram eficientes nos últimos 20 anos.

O que costumava levar cinco minutos para ser executado provavelmente leva menos de um segundo usando as máquinas de hoje com unidades de processamento avançadas.

Portanto, nem é preciso dizer que as máquinas podem processar dados com muito mais rapidez e facilidade. No entanto, ainda há um limite para a quantidade de dados que uma única máquina pode processar, independentemente de seu poder de processamento.

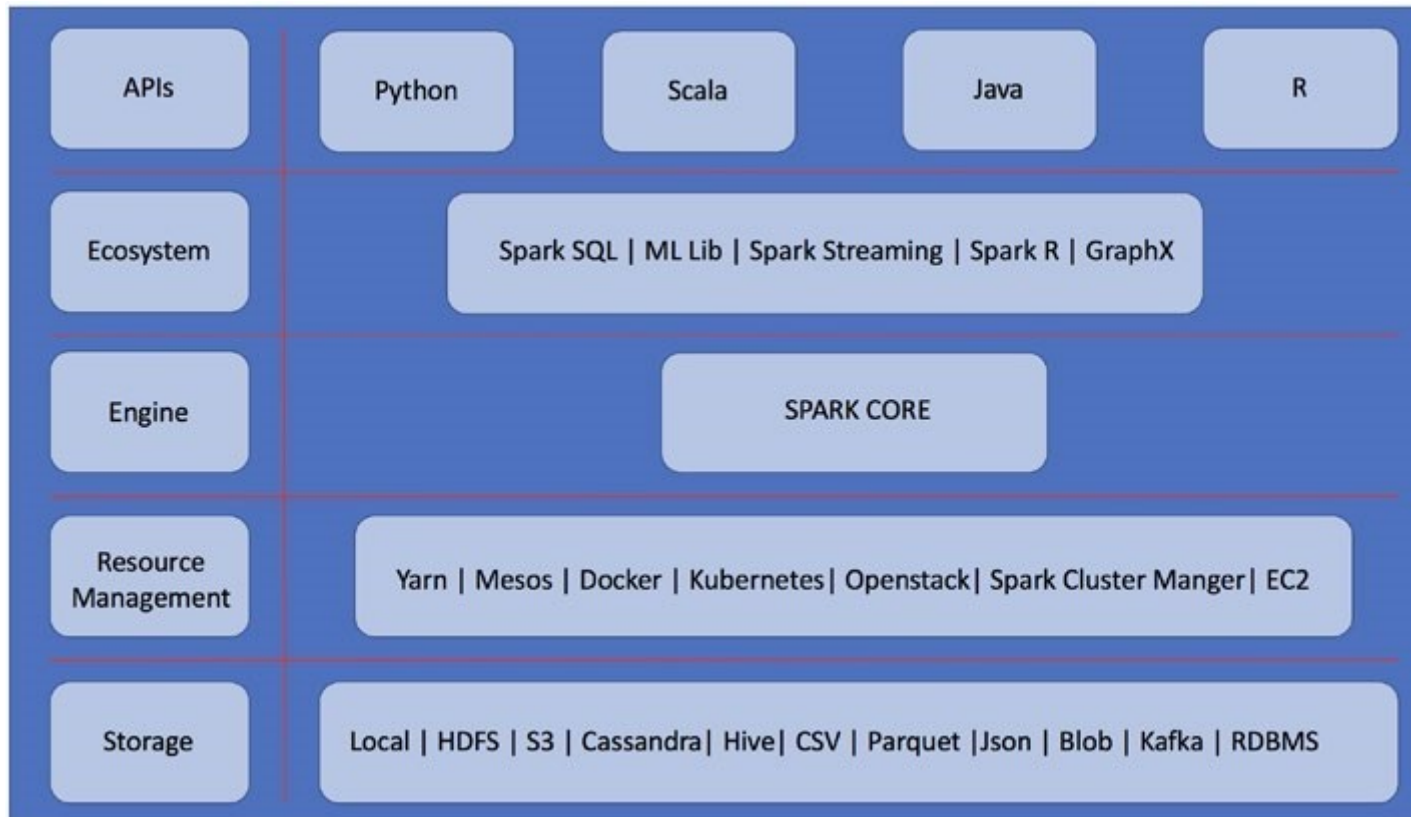
Portanto, a ideia subjacente ao *Spark* é:

*Usar uma coleção (cluster) de máquinas e um mecanismo de processamento unificado (Spark) para processar e manipular grandes quantidades de dados, sem comprometer a velocidade e a segurança. Este foi o objetivo final que resultou no nascimento de Spark.*

# Arquitetura *Spark*

Existem cinco componentes principais que tornam o Spark tão poderoso e fácil de usar. A arquitetura principal do Spark consiste nas seguintes camadas, conforme mostrado na Figura abaixo:

- Armazenamento (*Storage*)
- Gestão de recursos (*Resource management*)
- *Engine*
- Ecossistema (*Ecosystem*)
- *APIs*



Core componentes do *Spark* Fonte: [www.enterpriseai.news>]

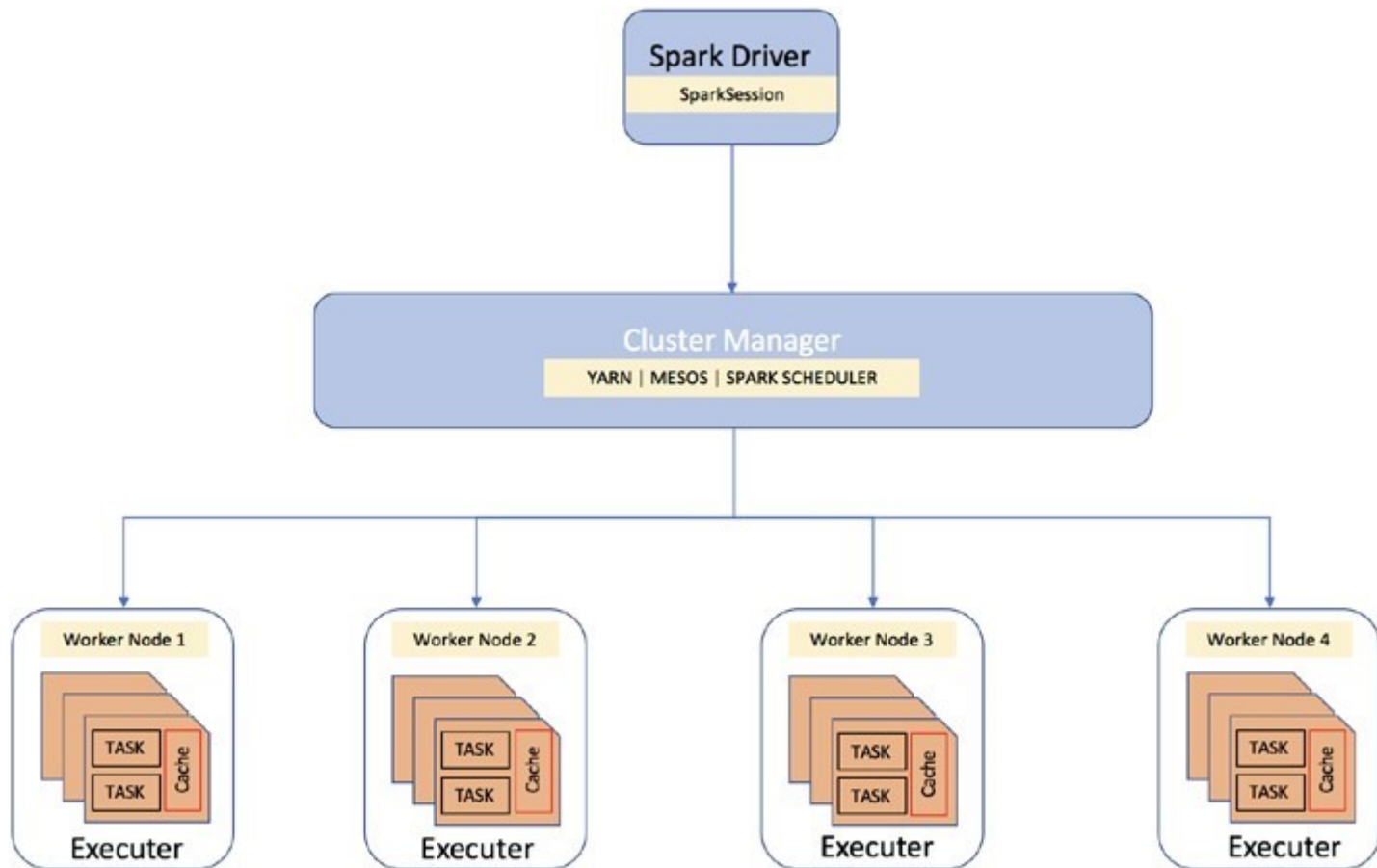


## Armazenamento

Antes de usar o Spark, os dados devem ser disponibilizados para serem processados. Esses dados podem residir em qualquer tipo de banco de dados. O Spark oferece várias opções para usar diferentes categorias de fontes de dados, para poder processá-los em grande escala. O Spark permite que você use bancos de dados relacionais tradicionais, bem como NoSQL, como Cassandra e MongoDB.

# Gestão de recursos

A próxima camada consiste em um gerenciador de recursos. Como o Spark funciona em um conjunto de máquinas (também pode funcionar em uma única máquina com vários núcleos), é conhecido como cluster Spark. Normalmente, há um gerenciador de recursos em qualquer cluster que trata com eficiência a carga de trabalho entre esses recursos. Os dois gerenciadores de recursos mais amplamente usados são YARN e Mesos. O gerenciador de recursos tem dois componentes principais internamente:



As tarefas que são fornecidas aos nós de trabalho são geralmente as partes individuais do aplicativo Spark geral. O aplicativo Spark contém duas partes:

1. Tarefa
2. Driver Spark A tarefa é a lógica de processamento de dados que foi escrita no código PySpark ou Spark R.

Pode ser tão simples quanto calcular a frequência total de palavras para um conjunto muito complexo de instruções em um conjunto de dados não estruturado.

O segundo componente é o driver Spark, o controlador principal de um aplicativo Spark, que interage de forma consistente com um gerenciador de cluster para descobrir quais nós de trabalho podem ser usados para executar a solicitação.

A função do driver Spark é solicitar que o gerenciador de cluster inicie o executor Spark para cada nó de trabalho

## Engine e ecossistema

A base da arquitetura do Spark é seu núcleo, que é construído sobre RDDs (conjuntos de dados distribuídos resilientes) e oferece várias APIs para a construção de outras bibliotecas e ecossistemas por colaboradores do Spark. Ele contém duas partes: a infraestrutura de computação distribuída e a abstração de programação RDD. As bibliotecas padrão no kit de ferramentas Spark vêm como quatro ofertas diferentes.

## Spark SQL

O SQL usado pela maioria dos operadores de ETL em todo o mundo torna uma escolha lógica fazer parte das ofertas do Spark. Ele permite que os usuários do Spark executem o processamento de dados estruturados executando consultas SQL. Na verdade, o Spark SQL aproveita o otimizador de catalisador para realizar as otimizações durante a execução de consultas SQL. Outra vantagem de usar o Spark SQL é que ele pode lidar facilmente com vários arquivos de banco de dados e sistemas de armazenamento, como SQL, NoSQL, Parquet, etc.

## **MLlib**

O treinamento de modelos de aprendizado de máquina em grandes conjuntos de dados estava começando a se tornar um grande desafio, até que o MLlib (biblioteca de aprendizado de máquina) do Spark passou a existir. MLlib oferece a capacidade de treinar modelos de aprendizado de máquina em grandes conjuntos de dados, usando clusters do Spark. Ele permite que você crie sistemas supervisionados, não supervisionados e de recomendação; Modelos baseados em PNL; e aprendizado profundo, bem como na biblioteca Spark ML.

## Streaming Estruturado

A biblioteca Spark Streaming fornece a funcionalidade de ler e processar dados de streaming em tempo real. Os dados recebidos podem ser dados em lote ou dados quase em tempo real de diferentes fontes. O fluxo estruturado é capaz de ingerir dados em tempo real de fontes como Flume, Kafka, Twitter, etc. Há um capítulo dedicado a esse componente posteriormente neste livro (consulte o Capítulo 3).

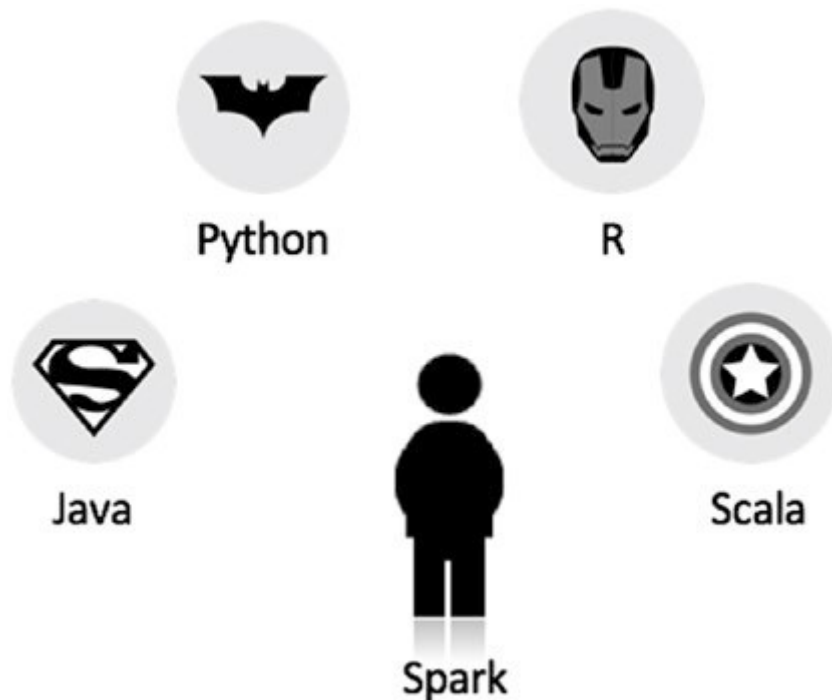


## Graph X

Esta é uma biblioteca que fica em cima do núcleo do Spark e permite que os usuários processem tipos específicos de dados (dataframes de grafos), que consiste em nós e vértices. Um grafo típico é usado para modelar a relação entre os diferentes objetos envolvidos. Os nós representam o objeto, e o vértice entre os nós representa o relacionamento entre eles. Os dataframes de grafo são usados principalmente na análise de rede, e o GraphX torna possível ter o processamento distribuído de tais frames de dados de grafos.

## APIs de linguagem de programação

O Spark está disponível em quatro linguagens. Como o Spark é construído usando Scala, ele se torna o idioma nativo. Além do Scala, também podemos usar Python, Java e R, conforme mostrado na Figura abaixo.



# Montando seu ambiente de trabalho para esta disciplina

- Você pode montar um ambiente para acompanhar as aulas e realizar seus próprios estudos na área de Big Data.
- Neste curso vamos trabalhar com o *PySpark* que é a versão em *Python* do *Spark*. Para construir este modle você precisará instalar no seu computador algumas ferramentas para usar *Python*, *PySpark* e assim poder entra no mundo do Big Data.

# Jupyter e Anaconda



As duas ferramentas essenciais na nossa jornada são o Anaconda e o Jupyter Notebook vamos conhece-los um pouco mais

# Anaconda

Anaconda é uma distribuição gratuita e de código aberto das linguagens de programação Python e R para computação científica (ciência de dados, aplicativos de aprendizado de máquina, processamento de dados em grande escala, análise preditiva, etc.), que visa simplificar o gerenciamento de pacotes e desdobramento, desenvolvimento. A distribuição inclui pacotes de ciência de dados adequados para Windows, Linux e macOS.

É desenvolvido e mantido pela Anaconda, Inc., fundada por Peter Wang e Travis Oliphant em 2012.

Como um produto Anaconda, Inc., ele também é conhecido como Anaconda Distribution ou Anaconda Individual Edition, enquanto outros produtos da empresa são Anaconda Team Edition e Anaconda Enterprise Edition, ambos os quais não são gratuitos.

Versões de pacotes no Anaconda são gerenciadas pelo sistema de gerenciamento de pacotes conda.

Este gerenciador de pacotes foi desenvolvido como um pacote de código aberto separado, pois acabou sendo útil por si só e para outras coisas além do Python.

Também existe uma pequena versão de bootstrap do Anaconda chamada Miniconda, que inclui apenas conda, Python, os pacotes dos quais eles dependem e um pequeno número de outros pacotes.

o Anaconda pode se baixado netse [link](http://www.anaconda.com/prod)  
([www.anaconda.com/prod](http://www.anaconda.com/prod)).

# Jupyter Notebook

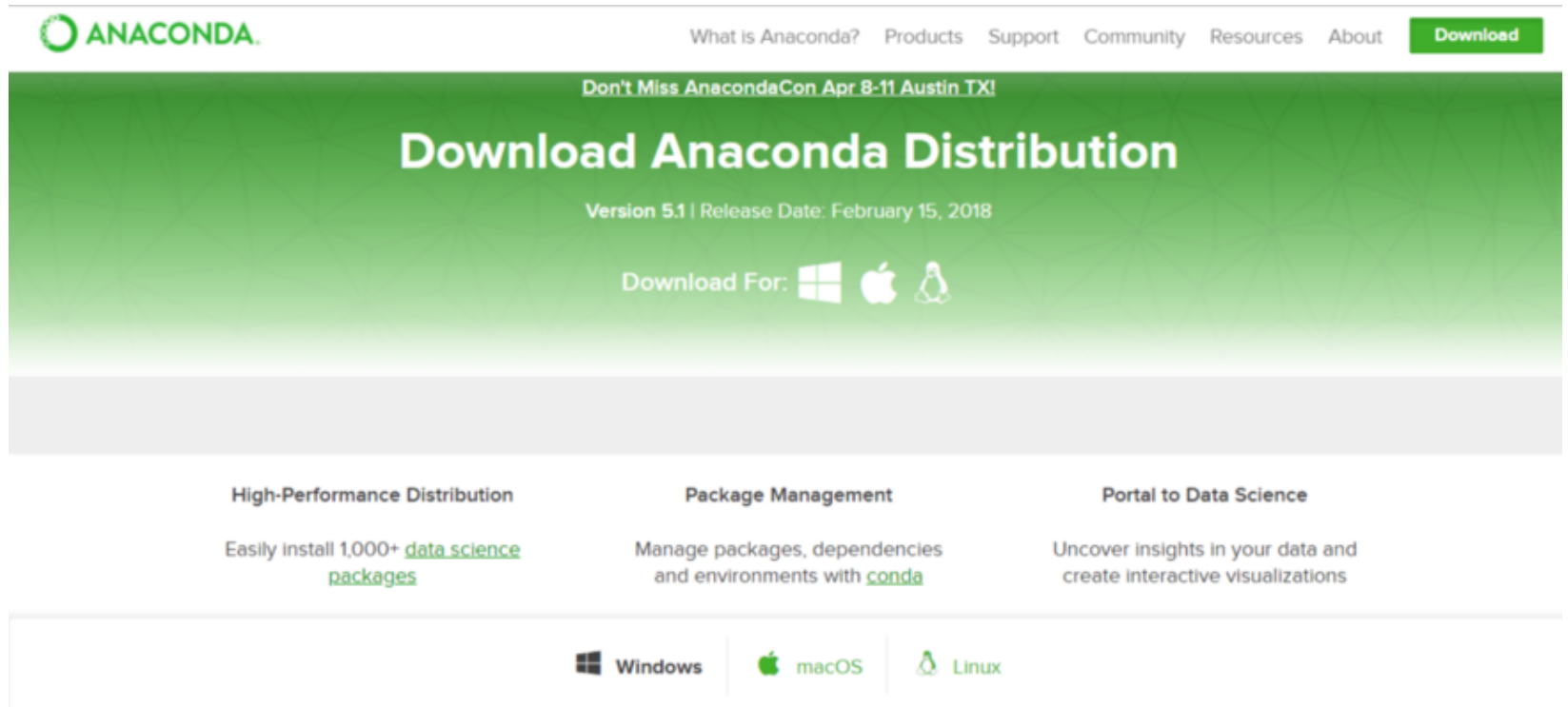
O Projeto Jupyter é uma organização sem fins lucrativos criada para "desenvolver software de código aberto, padrões abertos e serviços para computação interativa em dezenas de linguagens de programação". Originado do IPython em 2014, o Projeto Jupyter suporta ambientes de execução em dezenas de linguagens de programação. O nome do projeto é uma referência às três principais linguagens de programação suportadas por Jupyter, Julia, Python e R, e também uma homenagem aos cadernos de anotações de Galileu que registram a descoberta das luas de Júpiter.

Um Notebook Jupyter é um ambiente computacional web para a internet rica para criação de documentos para a plataforma Jupyter. O termo "notebook" pode, dependendo do contexto, fazer referência a entidades distintas como Jupyter (aplicativo Web), Jupyter Python (servidor Web) ou ao formato de documento para a plataforma. Um documento Jupyter Notebook é estruturado formato JSON, contendo uma lista ordenada de células de entrada / saída que podem conter código, texto (usando Markdown), matemática, gráficos e texto enriquecido, geralmente terminando com a extensão `.ipynb`.

**Não é necessário baixar e instalar o Jupyter isso é feito durante a instalação do Anaconda.**

# Baixando e Instalando o Anaconda

Procure por esse link e faça o download da versão para o seu sistema operacional



The screenshot shows the Anaconda website's download page. At the top, the Anaconda logo is on the left, and navigation links for 'What is Anaconda?', 'Products', 'Support', 'Community', 'Resources', and 'About' are on the right, followed by a green 'Download' button. Below the navigation bar is a green banner with the text 'Don't Miss AnacondaCon Apr 8-11 Austin TX!' and 'Download Anaconda Distribution'. Underneath the banner, it says 'Version 5.1 | Release Date: February 15, 2018' and 'Download For:' followed by icons for Windows, macOS, and Linux. The main content area is divided into three columns: 'High-Performance Distribution' (with the text 'Easily install 1,000+ [data science packages](#)'), 'Package Management' (with the text 'Manage packages, dependencies and environments with [conda](#)'), and 'Portal to Data Science' (with the text 'Uncover insights in your data and create interactive visualizations'). At the bottom, there is a row of three buttons: 'Windows' (with a Windows icon), 'macOS' (with an Apple icon), and 'Linux' (with a Linux icon).




ANACONDA.

What is Anaconda? Products Support Community Resources About [Download](#)

Don't Miss AnacondaCon Apr 8-11 Austin TX!

## Download Anaconda Distribution

Version 5.1 | Release Date: February 15, 2018

Download For:   

**High-Performance Distribution**




Easily install 1,000+ [data science packages](#)

**Package Management**

Manage packages, dependencies and environments with [conda](#)

**Portal to Data Science**

Uncover insights in your data and create interactive visualizations

 Windows  macOS  Linux

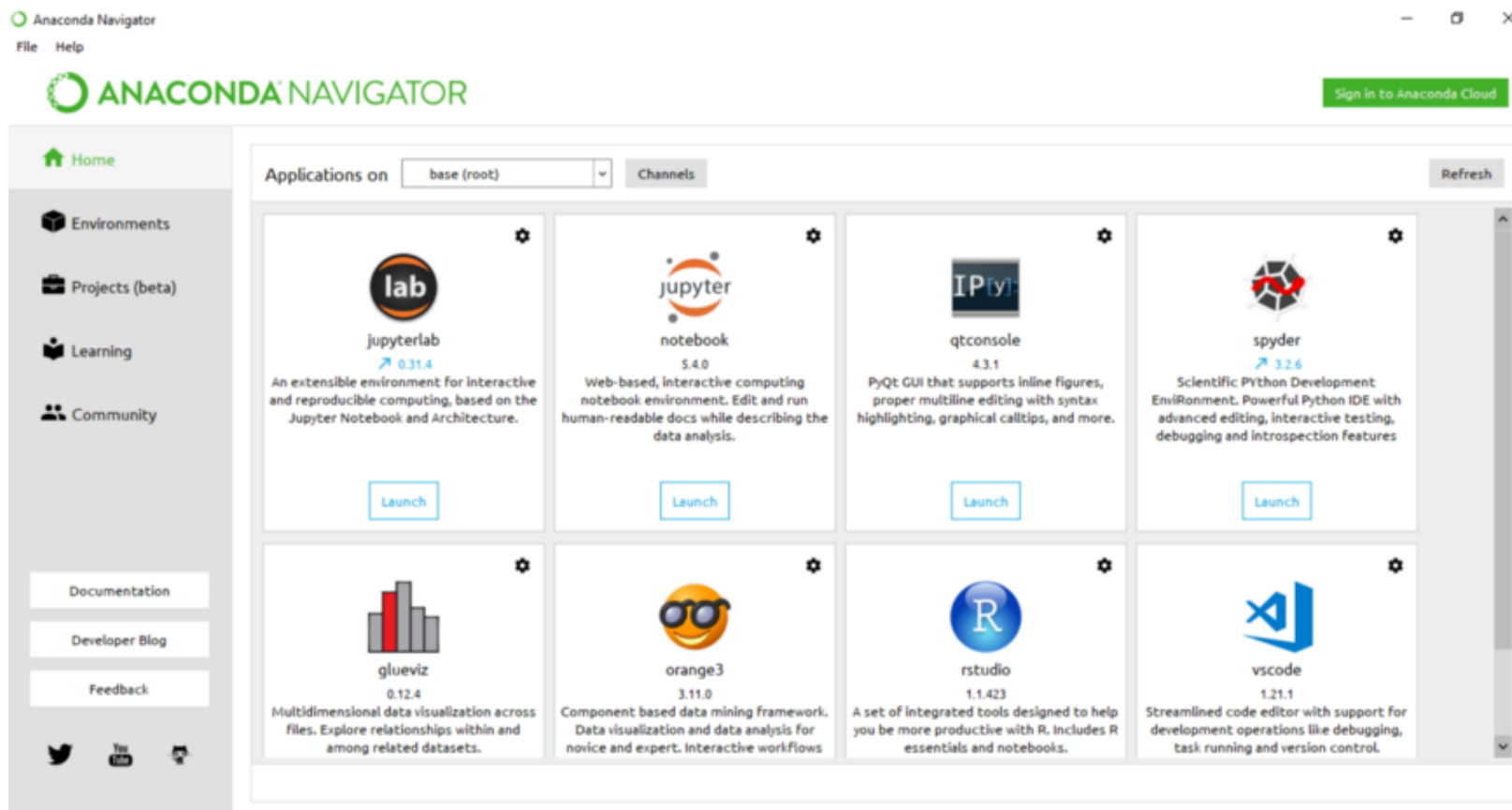


Na sequencia basta executar o programa abixado

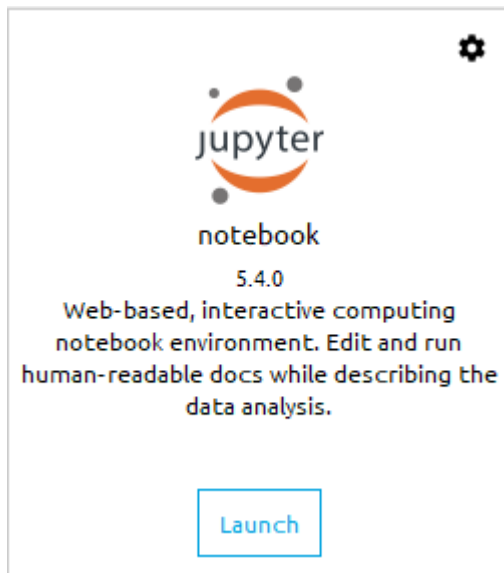


# Abra o Anaconda Navigator

Se seu Anaconda Navigator foi aberto, você está vendo uma tela com vários programas disponíveis. Estas são as plataformas que o pacote Anaconda oferece. Fique a vontade para não clicar em nada por enquanto. Confira se a tela que abriu se parece com essa.



Apesar de possuir várias plataformas de trabalho, a que utilizaremos hoje é o Jupyter Notebook, então, abra-o. Para isso, basta clicar no botão Launch (ou Install) abaixo da descrição do Jupyter.



# Iniciando no Databricks



# O que é Databricks

Databricks é uma plataforma que roda em cima do Apache Spark. Ele possui convenientemente uma configuração de sistemas de Notebook.

É possível provisionar clusters facilmente na nuvem e também incorpora um espaço de trabalho integrado para exploração e visualização.

Você também pode agendar qualquer notebook existente ou código Spark desenvolvido localmente para ir do protótipo à produção sem reengenharia.

**Passo 1 - Cadastre sua conta neste [link](https://databricks.com/try-databricks?utm_medium=cpc&utm_source=google&utm_campaign=8984002507&utm_offer=try-databricks&utm_content=trial&utm_term=databricks%20community&gclid=Cj0KQCQiAhs79BRD0ARIsAC6XpaUW0NH3WbNt7qirc1TiMfBRHmp1KuyZ2xty1dJJxV53bJwkB49aS3waAgC6EALw_wcB)**  
**([https://databricks.com/try-databricks?](https://databricks.com/try-databricks?utm_medium=cpc&utm_source=google&utm_campaign=8984002507&utm_offer=try-databricks&utm_content=trial&utm_term=databricks%20community&gclid=Cj0KQCQiAhs79BRD0ARIsAC6XpaUW0NH3WbNt7qirc1TiMfBRHmp1KuyZ2xty1dJJxV53bJwkB49aS3waAgC6EALw_wcB)**  
**[utm\\_medium=cpc&utm\\_source=google&utm\\_campaign=8984002507&utm\\_offer=try-](https://databricks.com/try-databricks?utm_medium=cpc&utm_source=google&utm_campaign=8984002507&utm_offer=try-databricks&utm_content=trial&utm_term=databricks%20community&gclid=Cj0KQCQiAhs79BRD0ARIsAC6XpaUW0NH3WbNt7qirc1TiMfBRHmp1KuyZ2xty1dJJxV53bJwkB49aS3waAgC6EALw_wcB)**  
**[databricks&utm\\_content=trial&utm\\_term=databricks](https://databricks.com/try-databricks?utm_medium=cpc&utm_source=google&utm_campaign=8984002507&utm_offer=try-databricks&utm_content=trial&utm_term=databricks%20community&gclid=Cj0KQCQiAhs79BRD0ARIsAC6XpaUW0NH3WbNt7qirc1TiMfBRHmp1KuyZ2xty1dJJxV53bJwkB49aS3waAgC6EALw_wcB)**  
**[%20community&gclid=Cj0KQCQiAhs79BRD0ARIsAC6](https://databricks.com/try-databricks?utm_medium=cpc&utm_source=google&utm_campaign=8984002507&utm_offer=try-databricks&utm_content=trial&utm_term=databricks%20community&gclid=Cj0KQCQiAhs79BRD0ARIsAC6XpaUW0NH3WbNt7qirc1TiMfBRHmp1KuyZ2xty1dJJxV53bJwkB49aS3waAgC6EALw_wcB)**  
**[XpaUW0NH3WbNt7qirc1TiMfBRHmp1KuyZ2xty1dJJx](https://databricks.com/try-databricks?utm_medium=cpc&utm_source=google&utm_campaign=8984002507&utm_offer=try-databricks&utm_content=trial&utm_term=databricks%20community&gclid=Cj0KQCQiAhs79BRD0ARIsAC6XpaUW0NH3WbNt7qirc1TiMfBRHmp1KuyZ2xty1dJJxV53bJwkB49aS3waAgC6EALw_wcB)**  
**[V53bJwkB49aS3waAgC6EALw\\_wcB](https://databricks.com/try-databricks?utm_medium=cpc&utm_source=google&utm_campaign=8984002507&utm_offer=try-databricks&utm_content=trial&utm_term=databricks%20community&gclid=Cj0KQCQiAhs79BRD0ARIsAC6XpaUW0NH3WbNt7qirc1TiMfBRHmp1KuyZ2xty1dJJxV53bJwkB49aS3waAgC6EALw_wcB)**

**Passo 2 - Acesse o Login link**

**([https://accounts.cloud.databricks.com/registration.html?](https://accounts.cloud.databricks.com/registration.html?_gl=1*av32nu*_gcl_aw*R0NMLjE2MDU2MzIxNzEuQ2owSONRaUFocz5QlJEMEFSSXNBQzZYcGFVVzBOSDNXYk50N3FpcmMxVGINZkJSSG1wMUt1eVoyeHR5MWRKSnhWNTNiSndrQjQ5YVMzd2FBZ0M2RUFMd193Y0I.#login)**

**[\\_gl=1\\*av32nu\\*\\_gcl\\_aw\\*R0NMLjE2MDU2MzIxNzEuQ2owSONRaUFocz5QlJEMEFSSXNBQzZYcGFVVzBOSDNXYk50N3FpcmMxVGINZkJSSG1wMUt1eVoyeHR5MWRKSnhWNTNiSndrQjQ5YVMzd2FBZ0M2RUFMd193Y0I.#login](https://accounts.cloud.databricks.com/registration.html?_gl=1*av32nu*_gcl_aw*R0NMLjE2MDU2MzIxNzEuQ2owSONRaUFocz5QlJEMEFSSXNBQzZYcGFVVzBOSDNXYk50N3FpcmMxVGINZkJSSG1wMUt1eVoyeHR5MWRKSnhWNTNiSndrQjQ5YVMzd2FBZ0M2RUFMd193Y0I.#login)**

## Passo 3 - Criação de um novo cluster

Começamos criando um novo cluster para executar nossos programas. Clique em “Cluster” na página principal e digite um novo nome para o cluster.

Em seguida, você precisa selecionar a versão “Databricks Runtime”. O Databricks Runtime é um conjunto de componentes principais executados em clusters gerenciados por Databricks. Inclui o Apache Spark, mas também adiciona uma série de componentes e atualizações para melhorar a usabilidade e o desempenho da ferramenta.

Você pode selecionar qualquer versão do Databricks Runtime - selecionei 3.5 LTS (inclui Apache Spark 2.2.1, Scala 2.11). Você também pode escolher entre Python 2 e 3 ..



## Create Cluster

### New Cluster

Cancel

Create Cluster

0 Workers: 0.0 GB Memory, 0 Cores, 0 DBU  
1 Driver: 6.0 GB Memory, 0.88 Cores, 1 DBU

Cluster Name

Cluster-1

Databricks Runtime Version

3.5 LTS (includes Apache Spark 2.2.1, Scala 2.11)

Python Version

2

Instance

Free 6GB Memory: As a Community Edition user, your cluster will automatically terminate after an idle period of two hours.  
For [more configuration options](#), please [upgrade your Databricks subscription](#).

Instances Spark

Availability Zone

us-west-2c

A criação do cluster levará alguns minutos. Depois de algum tempo, você deve conseguir ver um cluster ativo no painel.

#### ▼ Interactive Clusters

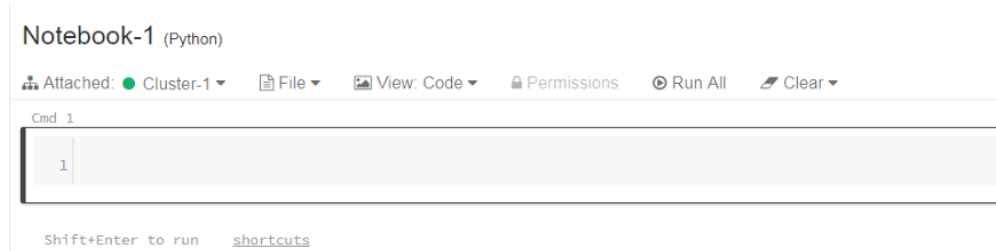
Name	State	Nodes	Driver	Worker	Runti
Cluster-1	Running	1 (0 spot)	Community Opti...	Community Opti...	3.5 L1

# Passo 4 - Criando um novo Notebook

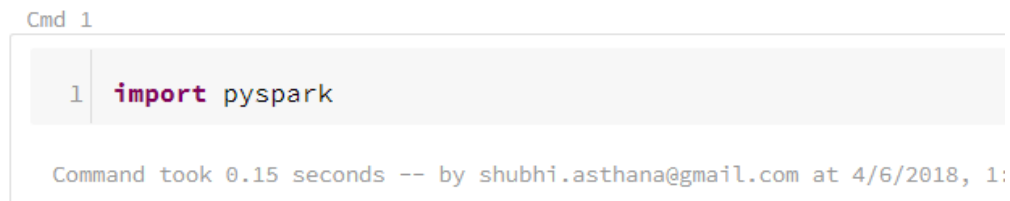
Vamos criar um novo Notebook no qual você pode executar seu programa.

Na página principal, clique em “Novo Notebook” e digite um nome para o Notebook.

Selecione o idioma de sua escolha - eu escolhi Python aqui. Você pode ver que o Databricks suporta vários idiomas, incluindo Scala, R e SQL.



Depois que os detalhes forem inseridos, você observará que o layout do notebook é muito semelhante ao do Jupyter. Para testar o notebook, vamos importar o pyspark.



O comando foi executado em 0,15 segundos e também fornece o nome do cluster no qual está sendo executado. Se houver algum erro no código, ele aparecerá abaixo da caixa do cmd.

Você pode clicar no ícone do teclado no canto superior direito da página para ver os atalhos específicos do sistema operacional.

Os atalhos mais importantes aqui são:

- Shift + Enter para executar uma célula
- Ctrl + Enter continua executando a mesma célula sem mover para a próxima célula

Observe que esses atalhos são para Windows. Você pode verificar os atalhos específicos do sistema operacional para o seu sistema operacional no ícone do teclado.

# Passo 5 - Carregando dados para Databricks

Vá até a seção "Tables" na barra esquerda e clique em "Criar table".

Você pode carregar um arquivo ou se conectar a uma fonte de dados Spark ou algum outro banco de dados.

Vamos fazer o upload do arquivo de conjunto de dados da íris comumente usado aqui (se você não tiver o conjunto de dados, use este [link \(https://archive.ics.uci.edu/ml/machine-learning-databases/iris/\)](https://archive.ics.uci.edu/ml/machine-learning-databases/iris/))

Depois de fazer upload dos dados, crie a tabela com uma IU para que possa visualizar a tabela e visualizá-la em seu cluster. Como você pode ver, você pode observar os atributos da tabela. O Spark tentará detectar o tipo de dados de cada uma das colunas e também permitirá que você edite.

## Select a Cluster to Preview the Table

Choose a cluster with which you will read and preview the data.

Cluster

Cluster-1 (6 GB, Running, 3.5 LTS (includes Apache Spark 2...)

Preview Table

## Specify Table Attributes

Specify the Table Name, Database and Schema to add this to the data UI for other users to access

Table Name

iris\_data

Create in Database

default

File Type

CSV

Column Delimiter

,

First row is header

Create Table

Table Preview

_c0	_c1	_c2	_c3	_c4
4.6	3.1	1.5	0.2	Iris-setosa
5.0	3.6	1.4	0.2	Iris-setosa
5.4	3.9	1.7	0.4	Iris-setosa
4.6	3.4	1.4	0.3	Iris-setosa
5.0	3.4	1.5	0.2	Iris-setosa
4.4	2.9	1.4	0.2	Iris-setosa

# Passo 5 - Carregando dados para Databricks II

Agora precisamos colocar cabeçalhos para as colunas, para que possa identificar cada coluna por seu cabeçalho em vez de c0, c1 e assim por diante.

Coloque seus cabeçalhos como comprimento da sépala, largura da sépala, comprimento da pétala, largura da pétala e classe.

Aqui, o Spark detectou o tipo de dados das primeiras quatro colunas incorretamente como String, então mude para o tipo de dados desejado - Float

## Specify Table Attributes

Specify the Table Name, Database and Schema to add this to the data UI for other users to access

Table Name ⓘ	Table Preview				
iris_data	Sepal Length	Sepal Width	Petal Length	Petal Width	Class
Create in Database ⓘ	FLOAT ▼	FLOAT ▼	FLOAT ▼	FLOAT ▼	STRING ▼
default	5.4	3.7	1.5	0.2	Iris-setosa
File Type ⓘ	4.8	3.4	1.6	0.2	Iris-setosa
CSV	4.8	3.0	1.4	0.1	Iris-setosa
Column Delimiter ⓘ	4.3	3.0	1.1	0.1	Iris-setosa
.					
<input checked="" type="checkbox"/> First row is header ⓘ					

# Passo 6 - Como acessar dados do Notebook

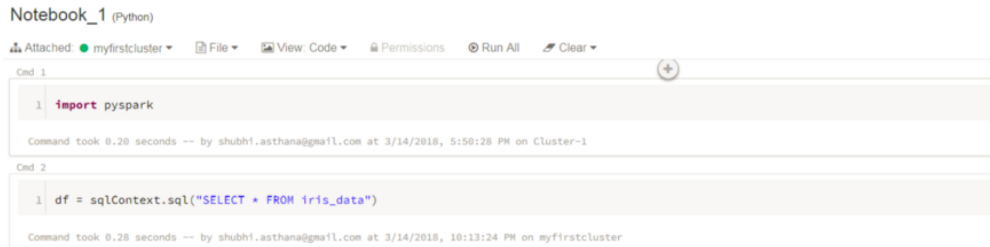
Spark é uma estrutura que pode ser usada para analisar big data usando SQL, aprendizado de máquina, processamento de gráfico ou análise de streaming em tempo real.

Estaremos trabalhando com SparkSQL e Dataframes nesta aula.

Vamos começar a trabalhar com os dados do Notebook.

Os dados que carregamos agora são colocados em formato tabular. Exigimos uma consulta SQL para ler os dados e colocá-los em um dataframe.

Digite `df = sqlContext.sql ("SELECT * FROM iris_data")` para ler os dados da íris em um dataframe.



Notebook\_1 (Python)

Attached: myfirstcluster | File | View: Code | Permissions | Run All | Clear

Cmd 1

```
1 import pyspark
```

Command took 0.20 seconds -- by shubhl.asthana@gmail.com at 3/14/2018, 5:50:28 PM on Cluster-1

Cmd 2

```
1 df = sqlContext.sql("SELECT * FROM iris_data")
```

Command took 0.28 seconds -- by shubhl.asthana@gmail.com at 3/14/2018, 10:13:24 PM on myfirstcluster

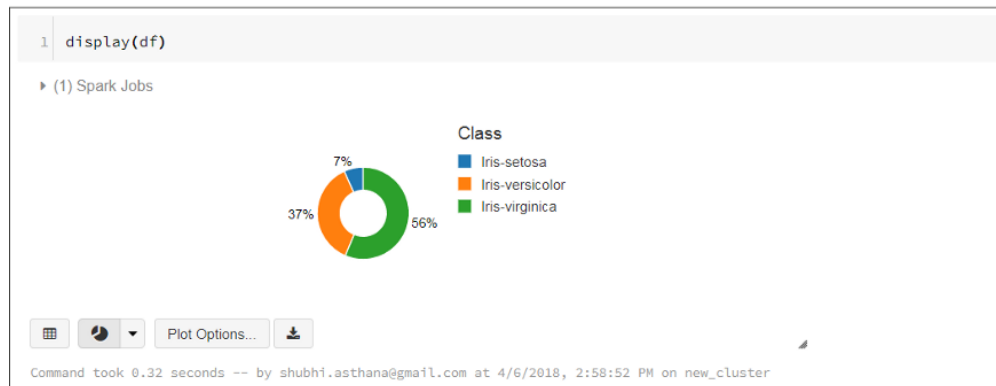
Para visualizar as primeiras cinco linhas no dataframe, posso simplesmente executar o comando: `display(df.limit(5))`

```
1 display(df.limit(5))
```

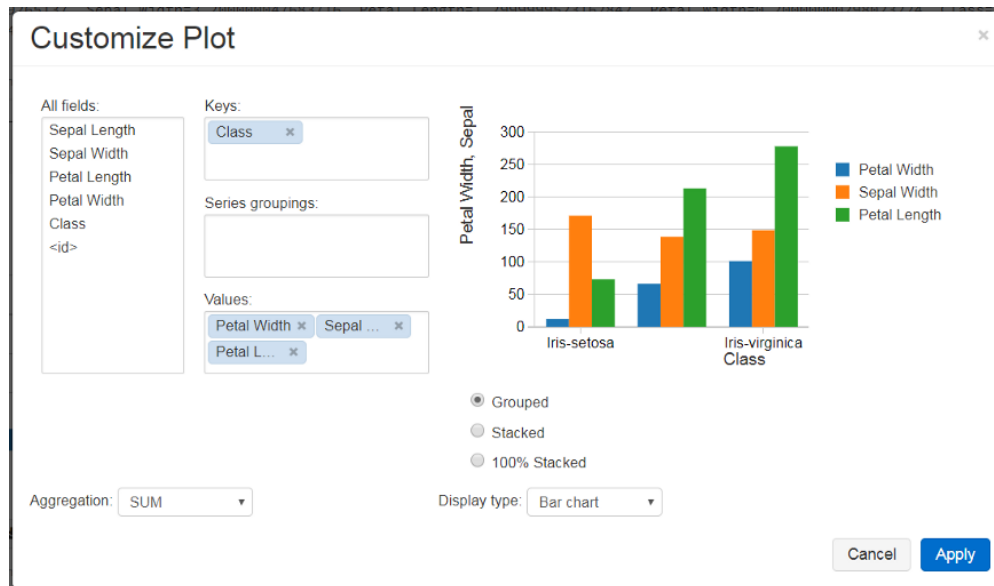
▶ (1) Spark Jobs

Sepal Length	Sepal Width	Petal Length	Petal Width	Class
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
5	3.6	1.4	0.2	Iris-setosa

Observe um ícone de gráfico de barras na parte inferior. Depois de clicar, você pode visualizar os dados que importou para o Databricks. Para visualizar o gráfico de barras de dados completos, execute `display(df)` em vez de `display(df.limit (5))`.



O botão suspenso permite que você visualize os dados em diferentes gráficos, como barra, pizza, dispersão e assim por diante. Ele também oferece opções de gráfico para personalizar o gráfico e visualizar apenas colunas específicas.





Para visualizar todas as colunas dos dados, basta digitar `df.columns`

---

Cmd 4

```
1 df.columns
```

```
Out[4]: ['Sepal Length', 'Sepal Width', 'Petal Length', 'Petal Width', 'Class']
```

Para contar quantas linhas no total existem no Dataframe (e ver quanto tempo leva para uma varredura completa do disco remoto / S3), execute `df.count()`.

Cmd 5

```
1 df.count()
```

► (1) Spark Jobs

Out[5]: 150

# Passo 7 - Convertendo um dataframe Spark em um dataframe Pandas.

Agora, se você se sentir confortável com o uso de dataframes do pandas e quiser converter seu dataframe do Spark em pandas, pode fazer isso colocando o comando

Digite

```
import pandas as pd  
pandas_df=df.to_pandas()
```

Agora você pode usar as operações do pandas no dataframe `pandas_df`.

```
1 import pandas as pd  
2
```

Command took 0.02 seconds -- by shubhi.asthana@gmail.com at 4/6/2018, 4:06:42 PM on new\_cluster

Cmd 10

```
1 pandas_df = df.toPandas()
```

► (1) Spark Jobs

Command took 0.39 seconds -- by shubhi.asthana@gmail.com at 4/6/2018, 4:06:56 PM on new\_cluster

Cmd 11

```
1 pandas_df.head()
```

Out[28]:

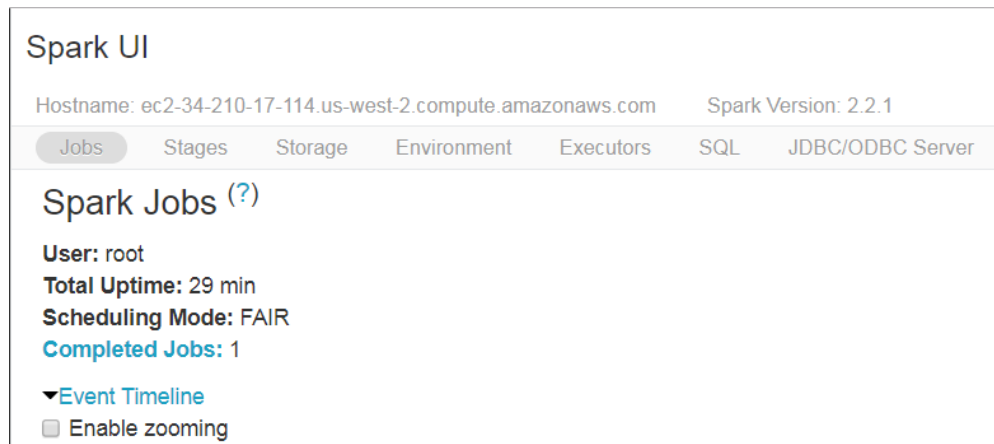
	Sepal Length	Sepal Width	Petal Length	Petal Width	Class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

## Passo 8 - Visualizando a IU do Spark

A UI do Spark contém muitas informações necessárias para depurar jobs do Spark. Há um monte de ótimas visualizações, então vamos vê-las em uma essência.

Para ir para o Spark UI, você precisa ir para o topo da página, onde há algumas opções de menu como “Arquivo”, “Exibir”, “Código”, “Permissões” e outros. Você encontrará o nome do cluster no topo ao lado de “Anexado” e um botão suspenso próximo a ele.

Clique no botão suspenso e selecione “Exibir interface do usuário do Spark”. Uma nova guia será aberta com muitas informações em seu Notebook.



Este tutorial pode ser acessado no seguinte link: <https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/58885268>  
(<https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/58885268>)

---