FIAP GRADUAÇÃO

FIAP

DATA SCIENCE

BIG DATA ARCHITECTURING & DATA INTEGRATION

Prof. Dr. Renê de Ávila Mendes

Objetivos da disciplina

DISCIPLINA: Big Data Architecturing & Data Integration

OBJETIVOS: Entenda as principais **arquiteturas** para ingestão, processamento e análise de grandes volumes de dados. Conheça as principais **ferramentas** open-source de Big Data como Hadoop, MapReduce, Spark, Sqoop, NiFi, Flume, Kafka, Zookeeper, HBase, Hive e as **integre** com as ferramentas de **extração, transformação e carga** de dados em modelos dimensionais. Entenda conceitos sobre **computação paralela e distribuída**, aplicação do Hadoop e bases Apache e arquiteturas *serverless* e desacopladas. Veja como **visualizar os dados** estruturados ou não estruturados com ferramentas de Self-Service Business Intelligence como PowerBI, utilizando as melhores práticas de **visualização de dados**.

Assuntos – 2º Semestre

- Arquiteturas para Big Data
- Data Pipelines
- Conceito de Data Lake
- Knime, HIVE, Spark, Data Streaming

Próximas aulas

- 11/09 - Revisão de conteúdo
- 18/09 - Checkpoint 6
- 25/09 - Aula para preparação para Challenge Sprint 4
- **02/10 - Introdução à análise de dados com Spark**
- 09/10 - Introdução a streaming
- 16/10 - Pipeline de dados - Da extração à visualização - Aula 1
- 23/10 - Pipeline de dados - Da extração à visualização - Aula 2
- 30/10 - Pipeline de dados - Da extração à visualização - Aula 3
- 06/11 - Pipeline de dados - Da extração à visualização - Aula 4
- 13/11 - Kick-off Global Solutions
- 20/11 - Feriado
- 27/11 - Global Solutions
- 04/12 - Substitutiva

Objetivo dessa aula

OBJETIVOS: Entender como funciona a análise de dados com o Spark

SPARK

FIAP

Apache Spark

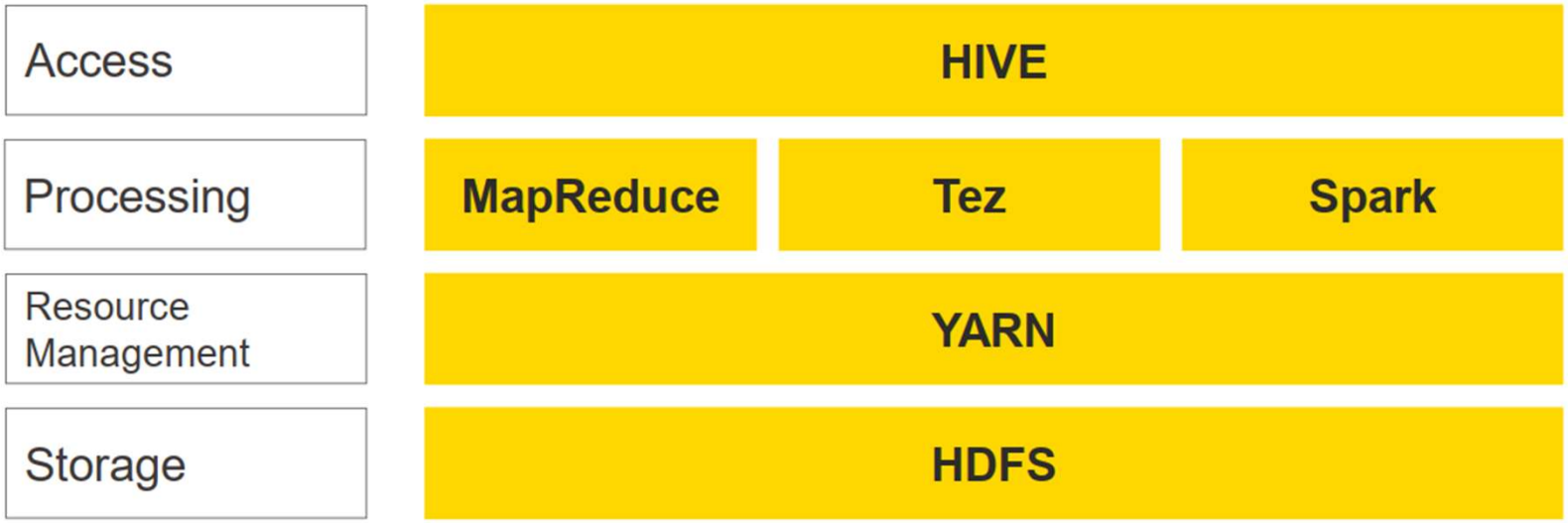
- Apache Spark é uma plataforma de computação em cluster que roda no topo de uma camada de armazenamento.

- Ele estende o MapReduce com suporte para mais tipos de componentes, tais como *streaming* e análises interativas.

- Spark oferece a capacidade de executar cálculos na memória, mas também é mais eficiente que o MapReduce quando precisa processar dados armazenados no disco.

FIAP

Apache Spark



FIAP

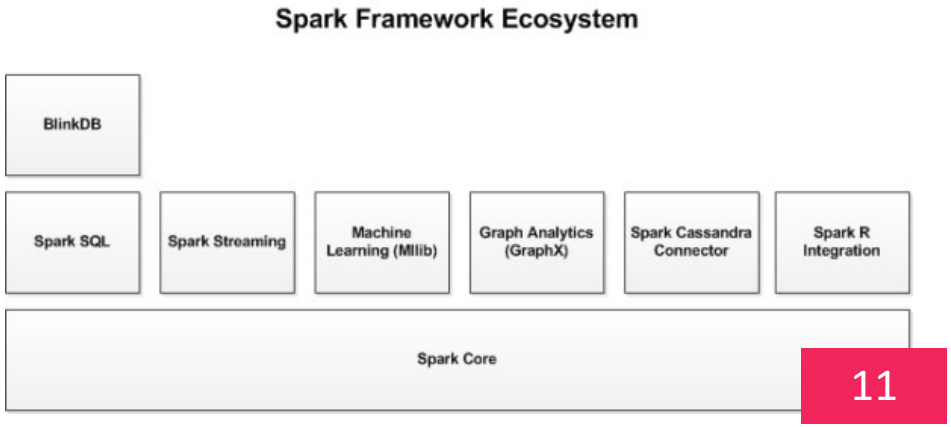
Apache Spark

- Spark processa os dados 10 vezes mais rápido que o MapReduce no disco e 100 vezes mais rápido na memória.
- Spark permite construir muito rapidamente algoritmos complexos para processamento de dados.
- Ele fornece suporte para muitos mais operadores que MapReduce como Joins, reduceByKey, combineByKey.
- Spark também fornece a capacidade de escrever programas interativamente usando o spark shell disponível para Scala e Python.

FIAP

Apache Spark

- Spark é totalmente compatível com Hadoop.
- Ele roda com YARN e acessa os dados no HDFS, Hbase e Hive.
- Além disso, ele permite a utilização do Apache Mesos, um gerenciador de recursos mais geral.



FIAP

Apache Spark

- Spark possui um framework integrado para análises avançadas como processamento de grafos, consultas avançadas, processamento de fluxo e machine learning.

- É possível combinar essas bibliotecas na mesma aplicação e usar uma única linguagem de programação.

- As aplicações podem ser desenvolvidas em Scala, Python, Java e a partir da versão 1.4.1, SparkR.

Spark e Databricks

- Em 2012 os pesquisadores da UC Berkley que projetaram do Spark fundaram uma empresa chamada Databricks;
- Seus principais produtos são uma plataforma web para programação para Spark e uma implementação de data lake para machine learning chamada Delta Lake;
- A empresa recebeu U$S 1,3 bilhões de aporte em sua existência, tendo seu valor de mercado estimado em US$ 38 bilhões, e mantendo cerca de 4.000 funcionários em escritórios espalhados em 15 países, incluindo o Brasil.

FIAP

MapReduce versus Spark

	Hadoop MapReduce	Spark
Armazenamento	Apenas no disco	in-Memory ou disco
Operações	Map e Reduce	Muitas transformações e ações, incluindo Map e Reduce
Modelo de execução	Batch	Batch, iterativo, streaming
Linguagens suportadas nativamente	Java	Scala, Java, R e Python

FIAP

Apache Spark

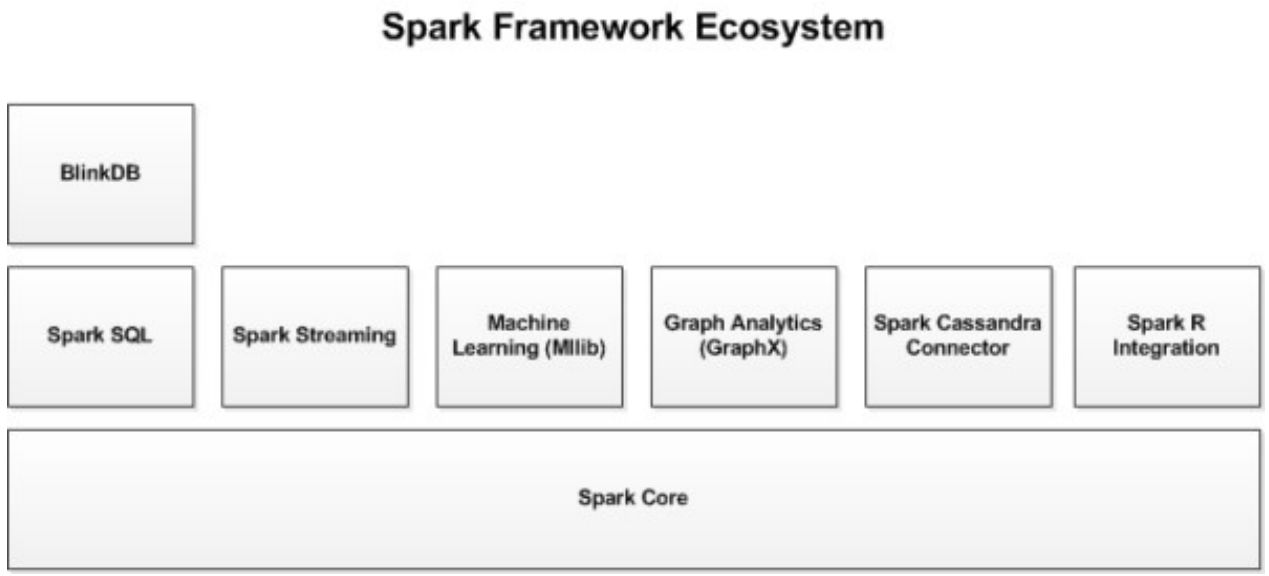
	Hadoop MR Record	Spark Record	Spark 1 PB
Data Size	102.5 TB	100 TB	1000 TB
Elapsed Time	72 mins	23 mins	234 mins
# Nodes	2100	206	190
# Cores	50400 physical	6592 virtualized	6080 virtualized

https://databricks.com/blog/2014/11/05/spark-officially-sets-a-new-record-in-large-scale-sorting.html

Ecossistema Apache Spark

FIAP

- Além da API do Spark, existem bibliotecas adicionais que fazem parte do seu ecossistema e fornecem capacidades adicionais para as áreas de análise de Big Data e aprendizado de máquina



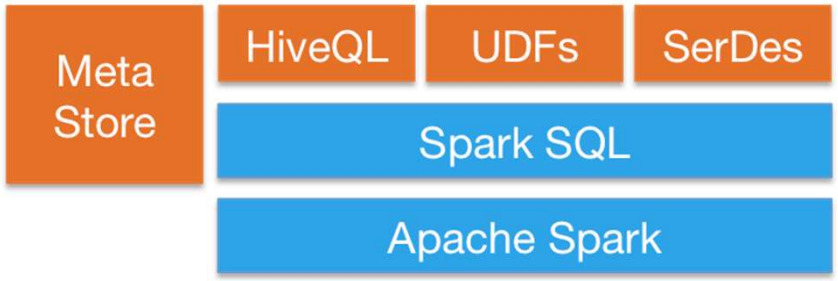
Ecossistema Apache Spark

FIAP

- Spark Streaming:
  - Usado para processar dados de streaming em tempo real baseado na computação de microbatch.
  - Para isso é utilizado o DStream que é basicamente uma série de RDD para processar os dados em tempo real;

Ecossistema Apache Spark

- Spark SQL:
  - Fornece a capacidade de expor os conjuntos de dados Spark através de uma API JDBC.
  - Isso permite executar consultas no estilo SQL sobre esses dados usando ferramentas tradicionais de BI e de visualização.
  - Além disso, também permite que os usuários usem ETL para extrair seus dados em diferentes formatos (como JSON, Parquet, ou um banco de dados), transformá-los e expô-los para consultas ad-hoc;



Ecossistema Apache Spark

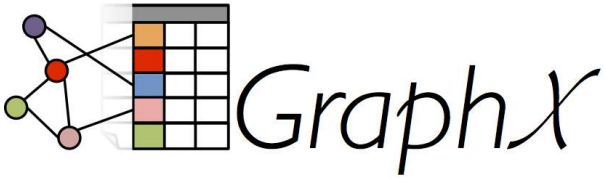
FIAP

- Spark MLlib:
  - Biblioteca de aprendizado de máquina do Spark, que consiste em algoritmos de aprendizagem, incluindo a classificação, regressão, clustering, filtragem colaborativa e redução de dimensionalidade;

Ecossistema Apache Spark

FIAP

- Spark GraphX:
  - GraphX é uma nova API do Spark para grafos e computação paralela.
  - Em alto nível, o GraphX estende o Spark RDD para grafos. Para apoiar a computação de grafos, o GraphX expõe um conjunto de operadores fundamentais (por exemplo, subgrafos e vértices adjacentes), bem como uma variante optimizada do Pregel.
  - Além disso, o GraphX inclui uma crescente coleção de algoritmos para simplificar tarefas de análise de grafos



Ecossistema Apache Spark

FIAP

- BlinkDB
  - Engine SQL para consultas por amostragem e pode ser usado para a execução de consultas interativas em grandes volumes de dados.
  - Permite que os usuários equilibrem a precisão de consulta com o tempo de resposta.
  - Além disso, o BlinkDB funciona em grandes conjuntos de dados, através de amostragem de dados e apresentação de resultados anotados com os valores de erros.

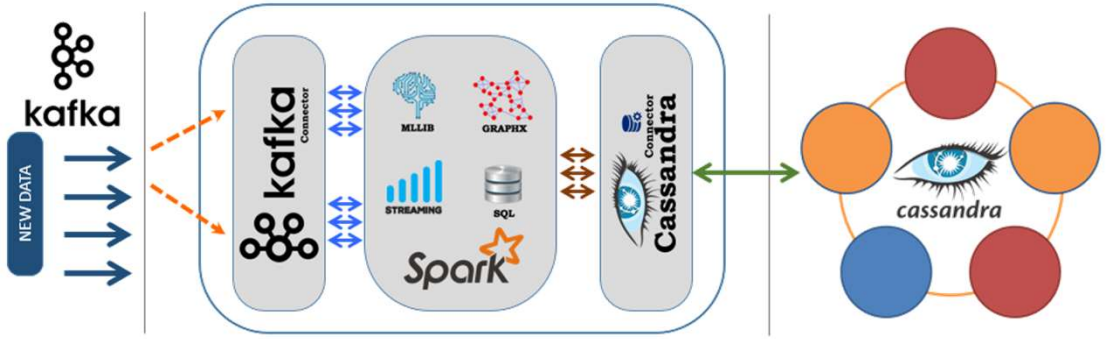
Ecossistema Apache Spark

FIAP

- Tachyon
  - Sistema de arquivos distribuídos em memória que permite o compartilhamento de arquivos de forma confiável e rápida através de frameworks de cluster, como Spark e MapReduce.
  - Também armazena em cache os arquivos que estão sendo trabalhados, permitindo que a existência de diferentes processamentos / consultas e enquadramentos para acessar arquivos em cache na velocidade de memória.

Ecossistema Apache Spark

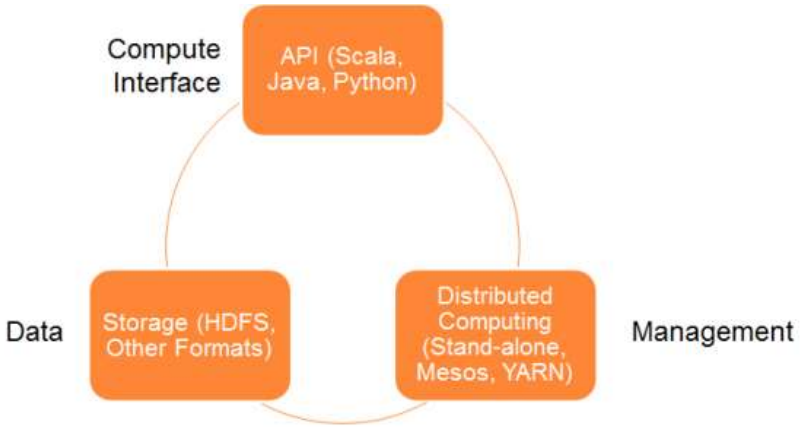
- Finalmente, há também adaptadores de integração com outros produtos, como Cassandra (Cassandra Spark Connector) e R (SparkR).
- Com o Cassandra Connector, é possível usar o Spark para acessar dados armazenados no banco de dados Cassandra e realizar com o R análises estatísticas.



FIAP

Arquitetura Spark

- A arquitetura Spark inclui os seguintes componentes:
  - Armazenamento de dados;
  - API;
  - Framework de gerenciamento.



FIAP

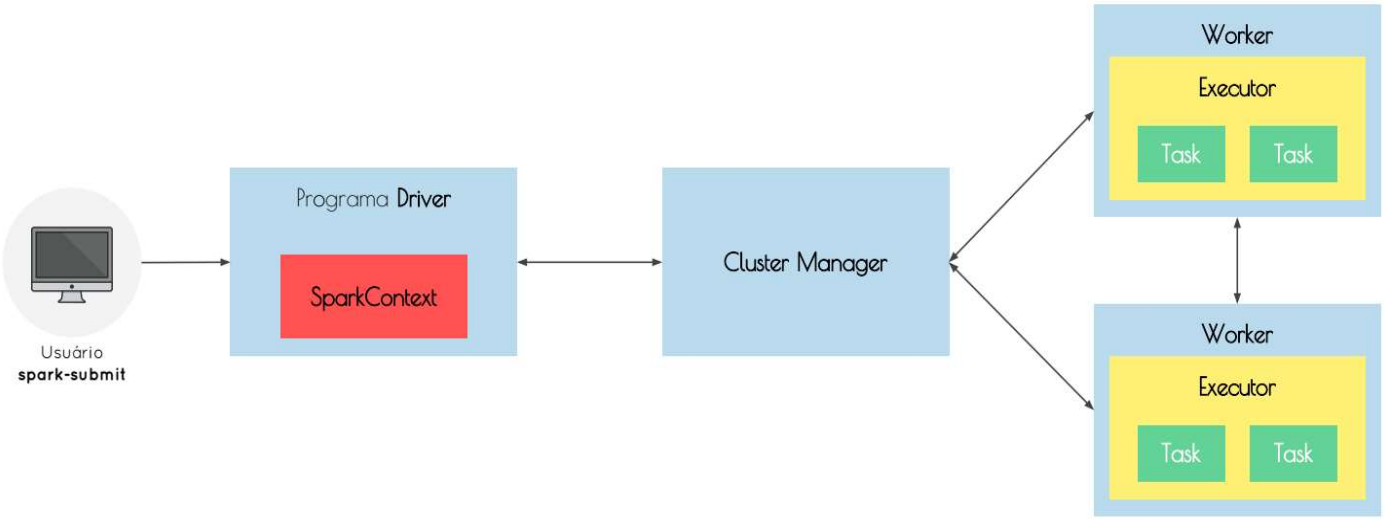
Arquitetura Spark

- **Armazenamento de dados:**
  - O Spark usa sistema de arquivos HDFS para armazenamento de dados.
  - Funciona com qualquer fonte de dados compatível com Hadoop, incluindo o próprio HDFS, HBase, Cassandra, etc.
- **API:**
  - A API permite que os desenvolvedores de aplicações criem aplicações baseadas no Spark usando uma interface de API padrão para Scala, Java e Python.
- **Gestão de recursos:**
  - O Spark pode ser implantado como um servidor autônomo ou em uma estrutura de computação distribuída como o Mesos ou o YARN.

FIAP

Arquitetura Spark

- **Driver** inicia o método main que vai criar um SparkContext.
- **Driver** entra em contato com o Cluster Manager solicitando recursos.
- Cluster Manager inicia os executores para o Driver.



FIAP

Representação de dados no Spark

DataFrame:

- *Table-like*: Collection of rows, organized in columns with names and types

Name	Surname	Age
John	Doe	35
Jane	Roe	29

- *Immutable*:
  - Data manipulation = creating new DataFrame from an existing one by applying a *function* on it
- *Lazily evaluated*:
  - Functions are not executed until an *action* is triggered, that requests to actually see the row data
- *Distributed:*
  - Each row belongs to exactly one *partition*
  - Each partition is held by a Spark Executor

FIAP

Resilient Distributed Datasets (RDD)

- Resilient distributed datasets (RDD) é a principal abstração no Spark.
- Eles são uma coleção de objetos que são distribuídos entre os nós do cluster.
- Um RDD pode ser criado a partir de uma coleção existente ou a partir de fontes de dados externas.
- Quando os dados são lidos no Spark, cria-se um RDD base (o primeiro RDD).

Resilient Distributed Datasets (RDD)

- Uma vez criado, RDD são imutáveis.
- Pode-se persistir ou fazer cache de RDDs na memória ou no disco.
- RDDs Spark são tolerantes a falhas. Se um determinado nó ou tarefa falhar, o RDD pode ser construído automaticamente em nós restantes e a tarefa será finalizada.

FIAP

Resilient Distributed Datasets (RDD)

- Existem dois tipos de operações de dados que podem ser executadas em um RDD:
  - transformações e
  - ações.

FIAP

Resilient Distributed Datasets (RDD)

- A transformação irá retornar um RDD; mas como RDDs são imutáveis, a transformação irá retornar um novo RDD.
- Transformações sofrem uma avaliação tardia (lazy evaluation).
  - RDDs são processados apenas quando uma ação é executada.
  - Exemplos de transformação incluem filter e map.
- Uma ação retorna um valor para o programa driver após executar alguma operação no dataset.
  - Exemplos de ação incluem count() e reduce().

FIAP

Lista de transformações mais comuns

Transformação	Descrição
map	Retorna um novo RDD aplicando uma função a todos os elementos.
filter	Retorna um novo RDD que consiste de elementos onde uma função é verdadeira.
groupByKey	Retorna um dataset de pares (k, Iterable<V>) a partir de um dataset (K, V)
reduceByKey	Retorna um dataset de pares (K, V) onde os valores da mesma chave são agregadas usando a função de redução.
flatMap	Similar ao map. No entanto, a função deve retornar uma sequência ao invés de um item único.
distinct	Retorna um novo dataset contendo apenas elementos distintos.

FIAP

Lista de ações mais comuns

Ação	Descrição
count()	Retorna o número de elementos no dataset.
reduce(func)	Agrega elementos de um dataset utilizando a função func.
collect()	Retorna todos os elementos do dataset como um array.
take(n)	Retorna um array com os primeiros n elementos.
first()	Retorna o primeiro elemento do dataset.
takeOrdered(n, func)	Retorna os primeiros n elementos do RDD em ordem ascendente ou especificado pela função func.

Formas de executar o Spark

FIAP

- Local
  - Roda na mesma JVM. Neste modo o programa Driver e os workers estão na mesma JVM.
  - Tudo fica no mesmo espaço de memória. Ele é util para prototipação, debug e teste.
- Standalone
  - Simples gerenciador de cluster pode ser executado manualmente, executando o mestre e os workers ou por meio de um script disponibilizado pelo Apache Spark.
  - Para instalar o Spark no modo Standalone, basta apenas adicionar uma versão compilada do Spark em cada nó no cluster.

Formas de executar o Spark

FIAP

- Hadoop Yarn
  - Existem dois modos de implementação que podem ser usados para lançar aplicações Spark no YARN.
  - No modo de cluster, o driver executado dentro de um processo mestre, que é gerido pelo YARN no cluster e o cliente pode sair sem esperar pelo fim da aplicação.
  - No modo cliente, o driver é executado no processo do cliente, e a aplicação mestre é apenas utilizada para solicitar recursos ao YARN.
  - É vantajoso executar Spark com YARN caso já exista um cluster Hadoop. Assim, pode-se usar o mesmo cluster Hadoop sem ter que manter um cluster separado.

Formas de executar o Spark

FIAP

- Apache Mesos
  - Ao usar Mesos, o Mesos master substitui o Cluster Manager.
  - As vantagens da utilização de Spark com Mesos são:
    - particionamento dinâmico entre o Spark e outras estruturas,
    - particionamento escalável entre várias instâncias do Spark.

FIAP

DataFrames

- Um DataFrame é uma coleção distribuída de dados organizados em colunas nomeadas.
- É conceitualmente equivalente a uma tabela em um banco de dados relacional, com muitas otimizações.
- DataFrames podem ser construídos a partir de uma ampla variedade de fontes, tais como: arquivos de dados estruturados, tabelas em Hive, bancos de dados externos, ou RDDs já existentes.