

## Linguagens de programação para ciência de dados (Python com Spark)



## Porque Python, Spark e Hadoop? Preparação do ambiente em Spark e Python


**Bloco 1**

Anderson Paulo Ávila Santos






## ► Objetivos

- Apresentar o Python, vantagens e desvantagens.
  - Apresentar o Spark e seu funcionamento.
  - Apresentar o Hadoop.
  - Configurar ambiente de desenvolvimento.
- 



## ► Python: o que é? Por que usar?

- Python é uma linguagem de alta produtividade e de fácil legibilidade.
  - Várias características positivas, entre elas, a legibilidade facilitada.
  - Pouca verbosidade.
  - Grande quantidade de bibliotecas.
  - Paradigmas de programação.
- 



## ► O *Hadoop*

- O *Hadoop* existe desde julho de 2008 e tem boas soluções para análise de grandes volumes de dados.
- Utiliza técnica de *Map* e *Reduce* para processamento.
- O *Hadoop* possui *clusters* que apresentam dificuldades para sua configuração e gerenciamento.
- Ao utilizar o *Spark* com o *Hadoop*, é possível evitar alguns desses problemas.



## ► O Spark

- Spark é um *framework* com bom desempenho para análise de dados de maneira sofisticada para *Big Data*.
- Atualmente, é um projeto da Apache Foundation.
- Utilizando o Spark, as aplicações têm a possibilidade de serem executadas em *clusters Hadoop*.
- Possibilita o desenvolvimento em Python, Scala ou Java.



## Porque Python, Spark e Hadoop? Preparação do ambiente em Spark e Python

Bloco 2

Anderson Paulo Ávila Santos





## ► Spark e Hadoop

- O Spark possibilita o compartilhamento de dados em memória através de grafos.
- Diferentes tarefas trabalham com os mesmos dados.
- A infraestrutura utilizada é o *Hadoop Distributed File System* (HDFS), aperfeiçoada e com ferramentas adicionais.
- Disponibiliza suporte para validar consultas sob demanda para *Big Data*.





## ► Spark e Hadoop

- Uma API de alto nível para aprimoramento da produtividade no desenvolvimento para *Big Data*.
- O resultados intermediários são obtidos ainda em memória antes de escrever-se em disco.
- O Spark é feito em linguagem Scala, que, por sua vez, é executado na máquina virtual do Java.

# PÓS-GRADUAÇÃO

## Teoria em prática

### Bloco 3

Anderson Paulo Ávila Santos





## ► Teoria em prática

A geração de muitos dados e a não utilização desses é algo recorrente no dia a dia das empresas. A maior parte gera uma quantidade enorme de dados e não analisa de forma a utilizá-los para a melhoria, aumento das vendas, entre outras vantagens que podem ser obtidas. Nesse contexto, uma dessas empresas que realiza vendas em varejo e armazena todos os dados das compras de todos seus clientes, decidiu que utilizará alguma tecnologia de *Big Data* para analisar o perfil de seus clientes, entre outras possibilidades que serão levantadas após a análise.



## ► Teoria em prática

Para isso, estão realizando um estudo para decidir qual das ferramentas existentes no mercado será a melhor opção. Sendo assim, o Spark está entre essas possibilidades de escolha. Outra escolha a ser realizada é com relação a linguagem de programação.



## ► Exercício 01

Levando em consideração o contexto apresentado, qual das alternativas apresenta uma das principais vantagens da utilização da linguagem Python?

- a) Linguagem que desobriga a indentação.
- b) Grande verbosidade.
- c) Documentação ampla.
- d) Suporta apenas orientação a objetos.
- e) Bibliotecas pouco atualizadas.

## ► Exercício 02

Com relação ao Spark, o que não caracteriza o *framework*?

- a) Suporte apenas para função de *Map* e *Reduce*.
- b) Operação em grafos otimizada.
- c) Avaliação de consultas de *Big Data* sob demanda.
- d) APIs consistentes para Python, Java e Scala.
- e) Shell interativo.

## ► Exercício 03

Com relação as características do Spark, seu ganho de desempenho em memória pode ser quantas vezes as tecnologias tradicionais?

- a) Duas vezes.
- b) Quatro vezes.
- c) Duzentas vezes.
- d) Cem vezes.
- e) Trinta vezes.



## Dica do professor

### Bloco 4

Anderson Paulo Avila Santos





## ► Dicas de leitura

- A documentação do Spark.
- Toda documentação do Spark e das API's.
- API para desenvolvimento em Python.



## ► Bibliografia

ZAHARIA, M. *et al.* Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing. *In: Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation. **USENIX Association**, 2012.*

OLIPHANT, T. E. Python for scientific computing. **Computing in Science & Engineering**, v. 9, n. 3, 2007.

DA SILVA, R. O.; SILVA, I. R. S. Linguagem de programação Python. **Tecnologias em Projeção**, v. 10, n. 1, 2019.

