WBA0753_v1.0

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

Introdução e instalação

Definição de *Python*, *Spark* e *Hadoop*. Funcionamento e relações. Preparação do ambiente.

Bloco 1

Yuri Sá

Python

Python é uma linguagem de programação livre, flexível e com uma curva de aprendizado suave e acessível.

- Surgiu nos anos 1990, já com código aberto.
- Scriptável.
- Amplamente extensível.
- *Cross-platform.*
- Intuitiva.
- Conectividade e integração.

Figura 1 - Logo *Python*

Fonte: https://upload.wikimedia.org/wikipedia/commons/f/f8/Python_logo_and_wordmark.svg. Acesso em: 22 mar. 2021.

Problemas do *Python*

Python é maravilhosa, mas existem deficiências que devem ser notadas:

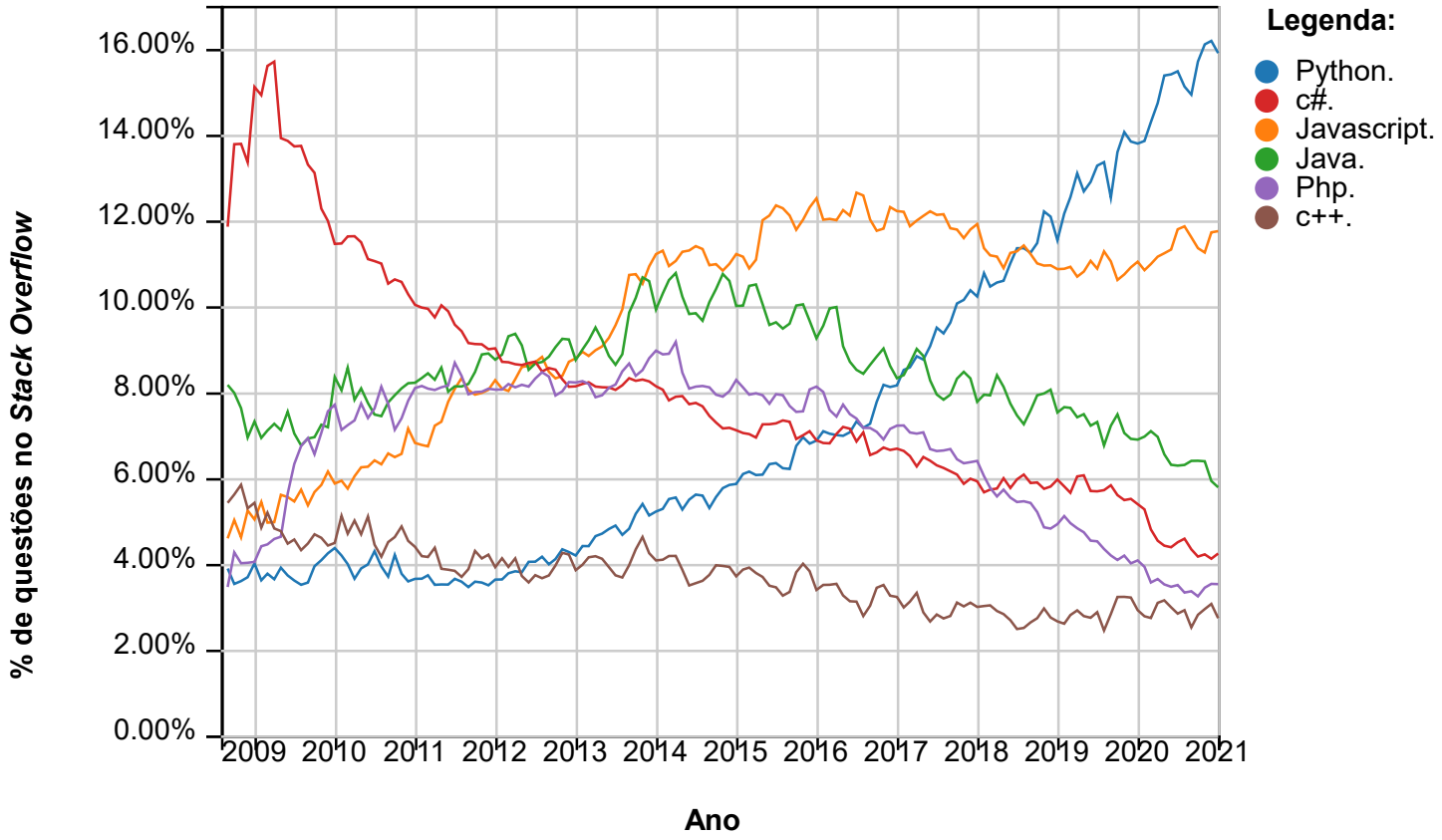
- Fracamente tipada.
- Falta de suporte nativo a *multithreading real*.
- Otimizações funcionais não são suportadas.
- Desempenho baixo em plataformas *mobile*.
- Sintaxe baseada em indentação.

Figura 1 - Logo *Python*

Fonte: https://upload.wikimedia.org/wikipedia/commons/f/f8/Python_logo_and_wordmark.svg. Acesso em: 22 mar. 2021.

Crescimento e evolução do *Python*

Figura 2 - Crescimento e evolução das linguagens



Fonte: adaptado de print de tela de *Stack Overflow.*

Spark

O *Spark* é um *framework* para processamento de *Big Data*, que disponibiliza ferramentas de alto nível para análises e processamento de grandes volumes de dados com foco em velocidade e simplicidade.

- Surgiu no final dos anos 2000.
- *OpenSource.*
- Simplicidade.
- Múltiplas linguagens de programação.
- Implementação em infraestruturas variadas.
- Utiliza os dados em memória.

Figura 3 - Logo *Spark*

Fonte: https://commons.wikimedia.org/wiki/File:Apache_Spark_logo.svg. Acesso em: 22 mar. 2021.

Hadoop

Hadoop é uma plataforma de software para processamento de grandes volumes dados em lotes, de forma confiável, escalável e distribuída.

- Surgiu nos anos 2000.
- *OpenSource.*
- Escalabilidade.
- Simplicidade.
- Confiabilidade.
- Baixo custo.
- Utiliza os dados em disco.

Figura 4 - Logo *Hadoop*

Fonte: https://commons.wikimedia.org/wiki/File:Hadoop_logo_new.svg. Acesso em: 22 mar. 2021.

Introdução e instalação

Definição de *Python*, *Spark* e *Hadoop*. Funcionamento e relações. Preparação do ambiente.

Bloco 2

Yuri Sá

Instalação *Python/ Spark/Hadoop*

Guia de instalação: demonstração prática.

1. *Java.*
2. *Python.*
3. *Spark.*
4. *Hadoop.*

Teoria em Prática

Bloco 3

Yuri Sá

Reflita sobre a seguinte situação

Imagine uma empresa que capta todos dados de vendas e de seus clientes, porém, ainda não os utiliza para nada.

Querem implementar uma solução de *Big Data* para melhorar a performance de vendas e/ou planejamento.

Que solução você recomendaria?

Norte para a resolução...

- Avalie a criticidade e volume das informações.
- Busque sempre opções de menor custo, seja monetário ou computacional.
- Tempo é dinheiro, então, o caminho mais curto tende a ser o melhor.
- Transforme todas essas informações em ativos de um projeto para elaborar sua resposta.

Dica do (a) Professor (a)

Bloco 4

Yuri Sá

Pacotes, *containers* e *VMs*

Para ajudar na instalação, e até mesmo na montagem de um servidor próprio, procure por pacotes e *containers* com as ferramentas pré-instaladas.

Essas ferramentas podem acelerar o processo de *setup* e ajudam no momento de escalar o trabalho para colocar em produção.

Referências

O'MALLEY, O. Terabyte sort on apache hadoop. **Yahoo**, maio de dois mil e treze. Disponível em: https://sortbenchmark.org/Yahoo2013Sort.pdf. Acesso em: 22 mar. 2021.

ROBINSON, D. The incredible growth of Python. **The Overflow**, quatorze de setembro de dois mil e dezessete. Disponível em: https://stackoverflow.blog/2017/09/14/python-growing-quickly/. Acesso em: 22 mar. 2021.

VAN ROSSUM, G.; DRAKE JR, F. L. **Python reference manual**. Amsterdam: Centrum voor Wiskunde en Informatica, 1995.

ZAHARIA, M. *et al*. Apache spark: a unified engine for big data processing. **Communications of the ACM**, v. 59, n. 11, p. 56-65, 2016.

Bons estudos!