Trabalho 1 - Gabriel Teixeira Júlio

As linguagens de programação escolhidas foram Julia e Python, e o motivo principal que eu escolhi estas linguagens foi eu querer trabalhar com linguagens que eu tenho pouco ou nenhum conhecimento sobre elas.

Julia: é uma linguagem de programação dinâmica de alto nível projetada para atender os requisitos da computação de alto desempenho numérico e científico, sendo também eficaz para a programação de propósito geral. Em 2020, um estudo com 20.000 pessoas feita pela empresa tchéca JetBrains apontou que apenas 1% dos entrevistados usava Julia em seu ecossistema. No entanto, o número de usuários tem crescido de forma expressiva, assim como os pacotes desenvolvidos em Julia, de tal forma que, entre janeiro de 2020 e janeiro de 2021, a linguagem saiu da posição 47 para a posição 23 no ranking do Índice TIOBE.

O motivo da escolha do Julia foi ela ser uma boa linguagem para trabalhar com problemas matemáticos e ser uma linguagem tão rápida quanto C.

Confiabilidade: Julia possui muitos mecanismos para garantir a confiabilidade de um código. Dentre os mesmo podemos destacar no exemplo abaixo a verificação dos índices de vetores em tempo de compilação, além do tratamento de exceções.

Paradigma: Julia se apresenta como uma linguagem multiparadigma tendo características presentes em paradigmas imperativos, funcionais e orientado a objetos, e pode ser considerada uma linguagem de programação de propósito geral.

Classe: embora seja definida Julia como compilada, utilizar uma máquina virtual e um conceito similar ao de código intermediário, o mais correto é classificar a linguagem como híbrida.

Tempo de execução: 31,128581 segundos

Tratamento de erros: qualquer erro que aparecer durante a execução do programa vai parar a execução, mostrar uma mensagem de qual é o erro, porque ele aconteceu e onde ele aconteceu. Os erros não podem ser tratados durante a execução com o try/catch.

Tratamento de exceções: funcionam como os erros, só que elas podem ser tratadas durante a execução com o try/catch. Para tratar algum erro específico basta colocar o nome dele na frente da declaração do exception, o nome pode ser pego na mensagem de erro dele sem o try/catch.

Verificação de tipo: só se é necessário a verificação de dados quando usado em estruturas com tipos setados, como Numpy Matrix, que precisam que os dados sejam do mesmo tipo, não aceitando dados de outros tipos.

Legibilidade: Julia possui muitos mecanismos de forma a facilitar o entendimento de um código.No exemplo abaixo podemos citar o fechamento de blocos com o termo reservado *end* é uso do termo *function*, indicando que está sendo criado uma função

Facilidade de escrita: A linguagem não possui muitas formas de se realizar o mesmo procedimento, facilitando assim o aprendizado do programador.

Python: é uma linguagem de programação de alto nível, interpretada de script, imperativa, orientada a objetos, funcional, de tipagem dinâmica e forte. Foi lançada por Guido van Rossum em 1991. Atualmente, possui um modelo de desenvolvimento comunitário, aberto e gerenciado pela organização sem fins lucrativos Python Software Foundation. Apesar de várias partes da linguagem possuírem padrões e especificações formais, a linguagem, como um todo, não é formalmente especificada.

O motivo da escolha do Python foi ela ser uma boa linguagem para trabalhar com problemas matemáticos e processos computacionais complexas, e eu quero melhorar meu conhecimento de Python.

Confiabilidade: É uma linguagem dinamicamente tipada e ao mesmo tempo fortemente tipada e também possui inferência a tipos. O sistema de verificação de tipos não realiza a conversão implícita de um tipo de dados para outro, levantando uma exceção quando tipos inconsistentes são utilizados.

Paradigma: uma linguagem de propósito geral de alto nível, multiparadigma, suporta o paradigma orientado a objetos, imperativo, funcional e procedural. Possui tipagem dinâmica e uma de suas principais características é permitir a fácil leitura do código e exigir poucas linhas de código se comparado ao mesmo programa em outras linguagens.

Classe: é uma linguagem interpretada mas, assim como Java, passa por um processo de compilação. Um código fonte Java é primeiramente compilado para um bytecode e depois interpretado por uma máquina virtual.

Tempo de execução: 64,09697842597961 segundos

Tratamento de erros: qualquer erro que aparecer durante a execução do programa vai parar a execução, mostrar uma mensagem de qual é o erro, porque ele aconteceu e onde ele aconteceu. Os erros não podem ser tratados durante a execução com o try/exception.

Tratamento de exceções: funcionam como os erros, só que elas podem ser tratadas durante a execução com o try/exception. Para tratar algum erro específico basta colocar o nome dele na frente da declaração do exception, o nome pode ser pego na mensagem de erro dele sem o try/exception.

Verificação de tipo: só se é necessário a verificação de dados quando usado em estruturas com tipos setados, como Numpy Matrix, que precisam que os dados sejam do mesmo tipo, não aceitando dados de outros tipos.

Legibilidade: é uma linguagem bastante legível, o que não é de espantar, dado que descende do ABC, uma linguagem desenhada exclusivamente com fins didáticos. O Python usa o nível de indentação no início de cada linha para definir blocos de código, dispensando a utilização de chavetas ou de outros indicadores de início e fim de bloco.

Simplicidade: Python tem por objetivo ser minimalista e possui menos instruções básicas e palavras reservadas, o que contribui para o seu fácil aprendizado. Python procura implementar apenas um modo correto de se executar uma determinada operação, isso fortalece a simplicidade e consequentemente a legibilidade da linguagem.