# Análise da linguagem Julia

Integrantes: Ana Beatriz Silva Zerati e William D'Abruzzo Martins

Julia é uma linguagem de programação científica usada para ciência de dados, machine learning, data mining, programação paralela e distribuída, entre outros. Segundo seus criadores, surgiu para ser uma linguagem com "a velocidade de C, o dinamismo de Ruby, as notações matemáticas de MatLab, de fácil uso como Python, fácil de manipular estatísticas como R e rápida, novamente, como C".
Por ser uma linguagem voltada para o âmbito científico, mais especificamente matemático, sua sintaxe é matematicamente orientada, possuindo uma vasta existência de tipos de dados, adequados para a aplicação, e suporte a operações matematicamente complexas e a notações matemáticas.

É uma linguagem altamente ortogonal, dando maior flexibilidade de combinação entre tipos, e com grande suporte à abstração, permitindo tipos abstratos e funções genéricas que podem realizar diversas operações com um único operador. Ainda sobre variáveis e tipagem, Julia é fraca e dinamicamente tipada, possuindo tipagem inferida e tipos definidos em tempo de execução e podendo tratar um tipo de dado de mais de uma maneira.

Com relação ao mecanismo de implementação, é híbrida, ou seja, compilada e interpretada. Possui compilador JIT (Just In Time), que compila o código em código de máquina otimizado durante a execução do programa, ao invés de compilação prévia à execução.

Além disso, Julia possui um interpretador usado para executar um código ainda não compilado.

Com todas essas características, a linguagem possui uma boa legibilidade para aqueles já familiarizados com o contexto matemático, em razão da sintaxe matematicamente orientada. Para os novos programadores, ou aqueles que não conhecem tanto da matemática, porém, o código pode se tornar confuso pela diversidade de tipos e alta ortogonalidade, flexibilidade em operações.

Por motivos similares, pode se apresentar como de fácil escrita e adaptação, porém com a possibilidade de confusão quando ainda não se está acostumado, devido à alta expressividade, com os operadores matemáticos, e às múltiplas possibilidades de resolução de problemas. Além disso, a escritabilidade é, em partes, abatida pela falta de guia do compilador, que não especifica erros, dificultando o seu tratamento e a correção de bugs.

No mesmo âmbito, pela tipagem fraca e dinâmica e, com isso, alta possibilidade de bugs, porém falta de apoio por parte do compilador, a linguagem sofre no quesito confiabilidade. Somado a isso, por seu ecossistema jovem, muitas das funções prontas diretamente de Julia ou estendidas de outras linguagens (R e Python, majoritariamente) não foram suficientemente testadas, podendo causar bugs no programa.  Para tal, Julia apresenta um mecanismo de tratamento de erros semelhante ao de Java.

Por conta desses problemas, seu custo de manutenção pode ser alto. Ainda por ser uma linguagem nova, também não há muito avanço no seu estado da arte, faltando ainda experiências e relatos do seu uso.

Conclui-se, então, que a principal vantagem da linguagem é seu alto desempenho misturado com sua alta produtividade, sendo uma linguagem com uma escritabilidade muito produtiva. Em compensação, sua confiabilidade é baixa por conta da sua idade e tipagem.