

# Uma breve apresentação do Haskell

Daniel Elias dos Santos<sup>1</sup>

<sup>1</sup>Departamento de Informática – Instituto Federal de Educação,  
Ciência e Tecnologia de Minas Gerais (IFMG)  
Sabará – MG – Brasil

danielias.santos@gmail.com

## 1. Introdução

### 1.1. Breve histórico

Haskell é uma linguagem de programação puramente funcional, polimórfica, estaticamente tipada e de avaliação tardia (*lazy*)<sup>[1]</sup>. Seu nome foi dado em homenagem ao lógico Haskell Brooks Curry, cujo trabalho em lógica matemática serve como base para linguagens funcionais. A linguagem foi projetada por uma comissão, formada em 1987 a partir da conferência “*Functional Programming Languages and Computer Architecture*” realizada em Portland, Oregon (EUA), para criação de uma linguagem funcional unificada, que estabelecesse uma base estável para o desenvolvimento de aplicações, e assim, popularizasse o paradigma funcional de programação<sup>[2]</sup>. A primeira versão da linguagem foi apresentada em 1990, e a primeira versão considerada estável foi apresentada em 1999. Haskell foi influenciada pelas linguagens Miranda e ML.

A linguagem, que utiliza o paradigma funcional, difere em muito do paradigma imperativo das linguagens mais conhecidas, tais como Cpp, Java ou CSharp, tanto em semântica quanto em sintaxe, o que pode surpreender desenvolvedores acostumados com o modelo imperativo. Dentre os diversos tutoriais disponíveis na *Web* Miran Lipovača<sup>[3]</sup> criou um indicado para iniciantes, que foi traduzido para português brasileiro por Tailor Fontela<sup>[4]</sup>.

### 1.2. Características principais

Haskell é uma linguagem estaticamente tipada, o que significa que erros são capturados em tempo de compilação. Os seus algoritmos não são expressos através de listas de instruções, como ocorre nas linguagens supracitadas: Haskell tem seus algoritmos totalmente baseados no conceito matemático de função. As estruturas de repetição que utiliza são por recursividade; outras estruturas como *for* ou *while* não estão disponíveis.

Em Haskell, funções são tratadas como tipos de primeira classe, que podem ser retornadas por outras funções ou passadas como parâmetro para outras funções. Alguns dos outros tipos primitivos principais são **Bool**, que pode assumir *true* ou *false*; **Char**, que se refere a caracteres isolados ('a', 'b', 'c', '1', '2', '3', etc.); **String**, como uma sequência de caracteres (“*Hello World!*”); **Int**, que são valores inteiros do intervalo  $-2^{31}$  e  $2^{31}$ ; **Integer**, que tem maior capacidade que o tipo Int; **Float**, que é um número real em ponto flutuante de precisão simples; e **Double**, que é um número real em ponto flutuante com o dobro de precisão. As estruturas de dados mais utilizadas em Haskell são listas e tuplas, sendo que cada lista pode conter outras listas, tornando-se matrizes

## 2. Metodologia

### 2.1. Ambiente

A execução dos testes foi realizada em ambiente Linux, utilizando a distribuição Debian GNU/Linux 9.4 (Stretch) 64bits, com o kernel em sua versão 4.9.88-1+deb9u1. Além disso, o processador da máquina utilizada foi fabricado pela Intel, modelo i7 4510U 2GHz, com 8GB DDR3 1600MHz de memória RAM instalada. O compilador da linguagem foi o *The Glasgow Haskell Compiler* (GHCi), em sua versão 8.0.1. Além disso, foi utilizada a IDE *Leksah Haskell*, que facilita a codificação e compilação da linguagem.

Para execução dos códigos, basta executar o compilador (num terminal, ou prompt de comando), com o comando *ghci*, e executar o módulo de interesse (Fibonacci.hs, Mandel.hs, Newton.hs, Parse.hs, Pi.hs, Quicksort.hs). Para execução de todos os algoritmos é necessário que as bibliotecas Graphics.UI.GLUT e Numeric.AD sejam instaladas no ambiente. Em Linux podem ser instaladas por meio do gerenciador de dependências “Cabal”, com os comandos *cabal install glut* e *cabal install ad*. Após, basta chamar o nome do módulo com os parâmetros.

## 3. Conclusão

Haskell é uma poderosa linguagem de programação altamente baseada em conceitos matemáticos e de uso geral, que pode ser usada em praticamente qualquer tipo de programa e problema. Na internet podemos encontrar vários programas e algoritmos feitos em Haskell; destaca-se aqui o site oficial da linguagem <http://www.haskell.org/> que possui vasto conteúdo abrangendo tutoriais, documentação, implementações, bibliotecas, etc. Os tutoriais citados anteriormente poderão ser utilizados como uma boa base de conhecimento para quem se interessar, pois são bem didáticos.

## Referências

- [1] WANSBROUGH, K.; ERISSON, S. *Haskell Wiki - The Haskell Programming Language*. Acesso em: 02/07/2018. Disponível em: <<https://wiki.haskell.org/Haskell>>.
- [2] HUDAK, P. et al. *A history of haskell: Being lazy with class*. 2007.
- [3] LIPOVAČA, M. *Learn You A Haskell*. Acesso em 02/07/2018. Disponível em: <<http://learnyouahaskell.com/chapters>>.
- [4] FONTELA, T. *Aprenda Haskell*. Acesso em 02/07/2018. Disponível em: <<http://haskell.tailorfontela.com.br/chapters>>.