

Vantagens da Programação Funcional (PF)

Daniel K. de Souza Rodrigues¹

¹Departamento de Computação - Universidade Federal de Ouro Preto (UFOP)
Ouro Preto – MG – Brasil

{daniel.rodrigues}@aluno.ufop.edu.br

Abstract. *This article aims to show advantages of functional programming, also presenting some important concepts about it.*

Resumo. *Este artigo tem como objetivo mostrar as vantagens da Programação Funcional, apresentando também, alguns conceitos importantes sobre a mesma.*

1. Introdução

Em 1930 Alonso Church desenvolveu o cálculo *lambda*, que nada mais é do que um modo muito formal de descrever um cálculo de uma função. Já em 1950 John McCarthy criou a linguagem de programação *Lisp* baseada em *lambda*, essa linguagem foi a primeira linguagem de programação funcional. Mesmo não sendo uma linguagem de programação puramente funcional, *Lisp* introduziu a maioria das características hoje encontradas nas modernas linguagens de programação funcional [Luís 2018]. Em 1980 começou o desenvolvimento da linguagem de programação *Haskell*, uma linguagem de programação *lazy* e *open source*.

Programação funcional é tido como um paradigma que trata a computação como uma avaliação de funções matemáticas e que evita estados ou dados mutáveis. Ela enfatiza a aplicação de funções, em contraste da programação imperativa, que enfatiza mudanças no estado do programa [Souza 2017]. O paradigma de uma linguagem de programação é a sua identidade. Corresponde a um conjunto de características que, juntas, definem como ela opera e resolve os problemas [Tedesco 2016].

O uso da programação funcional é adequado para computações matemáticas, inteligência artificial e processamento paralelo [Informant 2014]. Além de que, diferente da linguagem imperativa, a linguagem funcional é mais legível e gerenciável. O que é feito imperativamente, é possível fazer de forma mais clara, elegante e gerenciável, no sentido de dar suporte com programação funcional. Assim com a programação funcional a imperativa é um paradigma de programação, porém, esta descreve a computação como ações, enunciados ou comandos que mudam o estado ou variáveis.

Este trabalho está dividido em três Seções. Primeiramente temos a Seção 1, 'Introdução', que é falado um pouco do surgimento da programação funcional e o que ela é, a Seção 2, 'Vantagens da Programação Funcional' é reservada para apresentarmos algumas vantagens e conceitos importantes sobre a mesma, além de alguns *softwares* utilizados na indústria. Por fim, temos a Seção 3, que são as 'Conclusões' sobre o trabalho em questão.

2. Vantagens da Programação Funcional

Esta seção tem como objetivo listar algumas das vantagens da programação funcional. Para [Moreira et al. 2018] o uso da programação funcional é frequente no ambiente acadêmico, porém, algumas linguagens de programação já são utilizadas na indústria e no comércio, como: *Haskell*, *F*, *R* (estatística), *Erlang* (aplicações concorrentes), *Mathematica* (matemática simbólica) *J* e *K* (análise financeira) e *XSLT*.

Na programação funcional o desenvolvedor deve escrever funções e o software funcionar com base na interação entre essas expressões matemáticas, ou seja, ele deve descrever o que espera que aconteça, desta forma, o computador resolve o problema descrito pelo programador da melhor maneira possível. Este tipo de programação gera um código completo e autossuficiente que pode trazer diversos benefícios ao trabalho dos desenvolvedores. Além de contar com algumas outras vantagens listadas.

1. Visualização dos programas como funções uniformes;
2. Notação concisa;
3. Facilidade nos testes e na busca por *bugs*;
4. Uso de gerenciamento de memória automático;
5. Tratamento das funções como dados;
6. Possui grande flexibilidade;
7. Semântica simples.

Como as funções são mais fáceis de serem avaliadas de forma isolada, os testes podem ser direcionados para expressões específicas, possibilitando programas mais concisos e imunes a erros [Souza 2017]. Neste tipo de programação a o risco de erro na codificação é menor pois cada componente é completamente isolado. Outro ponto importante, segundo [Moreira et al. 2018] a recursividade usado em funções de primeira-classe permite que as provas para correção sejam mais simples do que normalmente espelhar a estrutura do código.

Alguns conceitos sobre programação funcional que devem ser utilizados para desenvolvimento, conceitos estes muito importantes, como literais e construtores. [Malaquias 2014] os literais são expressões com sintaxe especial para escrever alguns valores. Já construtores constantes são identificadores começando com letra maiúscula. Além de vermos também um pouco sobre expressões *lambdas*, que são funções anônimas que podem ser usadas como qualquer outro valor de primeira classe (números, caracteres, tuplas, listas e funções). Vimos também recursividade, que nada mais é, que um mecanismo de programação no qual uma definição de função ou de outro objeto refere-se ao próprio objeto sendo definido. Desta forma, função recursiva é uma função que é definida em termos de si mesma. Foram apresentados estes e outros conceitos, como casamento de padrão, expressões condicionais, polimorfismo, processamento de listas, etc.

Ainda sobre conceitos, temos casamento de padrão que é o ato de verificação da presença de um padrão em um dado ou em um conjunto de dados, tal é utilizado para testar se o objeto em questão possui a estrutura desejada. Já expressões condicionais permitem a escolha entre duas alternativas na obtenção do valor da expressão, com base em uma condição. Outro muito importante é o polimorfismo, que o *Haskell* incorpora uma forma deste que é a sobrecarga de funções, variáveis e literais. Por ultimo, mas não menos importante, processamento de listas, onde a lista é uma estrutura de dados formada por uma sequência de elementos do mesmo tipo.

3. Conclusão

Como conclusão, tem-se que, as linguagens funcionais são bastante fáceis de aprender. Além de que, uma boa ideia é iniciar por uma linguagem puramente funcional para se familiarizar com conceitos como funções de ordem superior, avaliação preguiçosa, recursividade, funções lambda e imaginar soluções de forma diferente da que está acostumado *estilson2018*. Temos também que, a programação funcional é um conceito bem antigo, que vem ganhando espaço, porém ela ainda é muito pouco utilizado.

Referências

Informant (2014). Conheça a programação funcional!

luís, J. (2018). Começando com programação funcional.

Malaquias, J. R. (2014). Programação funcional em haskell.

Moreira, R. A. F., Passos, A. V. C., Neto, O. R., and Júnior, J. A. (2018). Trabalho de programação funcional linguagem de programação i.

Souza, E. (2017). Programação funcional para iniciantes.

Tedesco, K. (2016). Linguagens e paradigmas de programação.