UNIVERSIDADE FEDERAL DE PERNAMBUCO
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
CENTRO DE INFORMÁTICA

Estudo sobre linguagens funcionais em alta demanda no mercado atual

---

Proposta de Trabalho de Graduação

Aluno: João Lucas Gomes de Miranda
Orientador: Marcio Lopes Cornelio

Sumário

1. Contexto

A programação funcional é um paradigma de programação onde a *computação* é tratada como a avaliação de expressões matemáticas. Tem esse nome justamente por se tratar basicamente de aplicação de funções a argumentos [1]. Dispõe de benefícios como a imutabilidade e funções de primeira ordem – funções que sempre produzem o mesmo resultado dadas as mesmas entradas.

A origem da programação funcional é datada de 1930 no *lambda calculus* – um sistema formal desenvolvido para o estudo de computabilidade, definição e aplicação de funções, recursão etc. [2] Muitas linguagens de programação funcional podem ser vistas como elaborações do *lambda calculus*. [2]

As linguagens de programação funcional são muito usadas na academia, mas também – e em especial recentemente – temos linguagens como Scheme [3], Common Lisp, Clojure [4, 12, 13], F# [5], Erlang [6],, Racket [7], Elixir [8], Haskell [9] etc. que tem sido muito usadas em aplicações industriais e comerciais [11]. Muito do porquê dessa "renascença" se deve ao fato de que linguagens funcionais oferecem, como consequência da imutabilidade, muita utilidade a sistemas distribuídos. São, portanto, tipicamente utilizadas em sistemas onde performance e integridade são pontos cruciais – isso é, quando o programa precisa fazer exatamente (e sempre) o que é esperado e opera em ambiente onde tarefas são compartilhadas por centenas ou milhares de devices. [10]

Clojure, por exemplo, é utilizada no projeto Akamai [12], uma rede de entrega de recursos (ou *cdn - content delivery network*) utilizada por gigantes como Facebook, e o Nubank, case de sucesso brasileiro, que também utiliza Clojure em seus sistemas [13]. Twitter, por sua vez, utiliza Scala em seus componentes de performance mais intensiva [14].

Muitas das linguagens modernas já implementam recursos de Programação Funcional. JavaScript – uma das linguagens mais populares da atualidade [15] – possui mecanismos de programação funcional como *maps, reducers, filters* etc. em adição aos paradigmas imperativo e orientado a objetos. Outras linguagens não funcionais, como Python, Java, C++ e C#, são também exemplos de linguagens que implementam recursos funcionais.

2. Objetivo

O objetivo deste trabalho é estudar as linguagens funcionais, dando especial ênfase às linguagens que estão em alta demanda profissional hoje em dia – linguagens como Clojure, F#, Elixir, Erlang, Scala são alguns exemplos. A idéia é investigar quais benefícios essas tecnologias trazem em oposição às tecnologias de paradigmas como Orientado a Objetos (OO) para os problemas que a indústria tecnológica atual enfrenta.

Serão, assim, estudados os conceitos fundamentais dessas tecnologias, bem como serão analisadas estatísticas relacionadas a: variedade de bibliotecas, *frameworks*, curva de aprendizado, satisfação do programador, segurança, performance etc.

Por fim, será feito um comparativo performático a partir de uma implementação simples de API em cada uma das linguagens estudadas.

3. Cronograma

Atividade	**Agosto**		**Setembro**		**Outubro**		**Novembro**	
Elaboração da Proposta		x	x					
Pesquisa			x	x	x	x		
Experimentos				x	x	x		
Elaboração do Relatório					x	x	x	x
Preparação da Apresentação							x	x

Tabela 1: cronograma de atividades

4. Referências

[1] John Hughes. “Why Functional Programming Matters”. Em: “Research Topics in Functional Programming” ed. D. Turner, Addison-Wesley, 1990, pp 17–42. URL: https://www.cs.kent.ac.uk/people/staff/dat/miranda/whyfp90.pdf

[2] Paul Hudak. “Conception, Evolution, and Application of Functional Programming Languages”. Em: ACM Computing Surveys, Vol. 21, No. 3, September 1989. URL: http://www.dbnet.ece.ntua.gr/~adamo/languages/books/p359-hudak.pdf

[3] Will Clinger (1987). "MultiTasking and MacScheme". Em: MacTech. 3 (12). URL: http://preserve.mactech.com/articles/mactech/Vol.03/03.12/Multitasking/index.html

[4] Nubank. Nubank Repositories. URL: https://github.com/nubank

[5] Howard Mansell (2008). “Quantitative Finance in F#”. Em: CUFP 2008. URL: http://galois.com2008/abstracts.html#MansellHoward

[6] "Who uses Erlang for product development?". Em: Frequently asked questions about Erlang. URL: http://erlang.org/faq/introduction.html#idp32582608

[7] “The Racket Language”. URL: https://racket-lang.org/

[8] Steve Cohen. “Introducing new open source tools for the Elixir community”. URL: https://medium.com/@Pinterest_Engineering/introducing-new-open-source-tools-for-the-elixir-community-2f7bb0bb7d8c

[9] “Haskell in Industry”. URL: https://wiki.haskell.org/Haskell_in_industry

[10] Jonathon Morgan. “Don’t be scared of Functional Programming”. URL: https://www.smashingmagazine.com/2014/07/dont-be-scared-of-functional-programming/

[11] “Functional Programming in the Real World”. URL: http://homepages.inf.ed.ac.uk/wadler/realworld/

[12] “The Akamai project”. URL: https://www.akamai.com/br/pt/

[13] “O nubank mudou de postura ao recrutar seus engenheiros”. URL: https://exame.abril.com.br/carreira/o-nubank-mudou-de-postura-ao-recrutar-seus-engenheiros/

[14] Charles Humble. “Twitter Shifting More Code to JVM, Citing Performance and Encapsulation As Primary Drivers”. URL: https://www.infoq.com/articles/twitter-java-use

[15] Matt Weinberger. “The 15 most popular computer languages, according to the Facebook for programmers”. URL: https://www.businessinsider.com/github-most-popular-coding-languages-2016-9/

5. Assinaturas

Recife, 10 de Setembro de 2018.

_____

Marcio Lopes Cornelio
(Orientador)

_____

João Lucas Gomes de Miranda
(Aluno)