

Proposta de TCC 2016 - IME-USP

Fellipe souto Sampaio (fellipe.sampaio@usp.br) - BCC IME USP

Introdução

Programação reativa é um paradigma de programação orientado pelo do **fluxo de dados** e e pela **propagação de mudanças**, enquanto que a programação funcional reativa é um paradigma de desenvolvimento de aplicações reativas que utiliza os conceitos fundamentais de programação funcional (map, reduce, filter...).

Motivação

Aplicações altamente concorrentes, como interfaces de usuários complexas, tem tradicionalmente manipulado os estados do programa através de variáveis globais. Muitas ações são coordenadas através da manipulação de eventos, algo que é naturalmente procedural. Porém esse modelo se tornou impraticável a medida que as complexidades do sistema aumentaram.

*A programação funcional se apresentou como um poderoso aliado na construção de sistemas confiáveis, eliminando estados mutáveis e permitindo que as aplicações fossem escritas de uma forma declarativa e segmentada. Esses princípios deram origem a **Programação Funcional Reativa**, paradigma excepcional para construir aplicações concorrentes e assíncronas. Ele permite a modelagem de estados mutáveis de forma funcional **

***Tradução livre do prefácio do livro Clojure Reactive Programming**

Nos últimos anos, a programação reativa tem sido tema central de muitos livros, artigos e softwares. Exemplo disso é a criação do **Manifesto Reativo**, um documento que tenta definir o que seria um sistema reativo; e também temos o popular framework **React**, idealizado pelo Facebook para criação de interfaces de usuários.

Atualmente a palavra **reatividade** virou uma *buzzword* dentro da comunidade de desenvolvimento de software, o uso dos seus mais variados conceitos e princípios, nas mais variadas situações, tende a crescer consideravelmente nos próximos anos.

O objetivo deste trabalho de conclusão de curso é estudar os princípios da programação reativa, em que pontos ela se apresenta mais viável em relação a outros paradigmas e como ela pode ser utilizada para construir aplicações reais. Pretendo utilizar inicialmente **Clojure** como linguagem de programação, por se tratar de uma linguagem funcional, derivada do LISP,

ela é muito eficiente em criar aplicações concorrentes e assíncronas. Em segundo plano pretendo estudar como o **Javascript e seus frameworks** podem ser utilizados para criar programação reativa.

Por fim pretendo estudar como uma aplicação reativa pode se integrar de forma harmoniosa com outros tipos de aplicações escritas no paradigma de orientação a objeto, utilizando **Java e Ruby**.

O estudo de **Scala e de suas ferramentas reativas** ficará para um momento posterior.

Objetivos

- Estudar e aplicar conceitos de programação funcional
- Estudar e aplicar conceitos de programação reativa
- Estudar e aplicar conceitos de design patterns e refatoração
- Estudar e aplicar conceitos de teste de unidade e aceitação
- Estudar e aplicar conceitos de desenvolvimento web

Tecnologias

- Clojure
- Javascript
- Ruby
- Java
- Scala

Principais Referências

- Clojure Reactive Programming - Leonardo Borges [Packt]
- Reactive Programming with JavaScript - Jonathan Hayward [Packt]
- Design Patterns: Elements of Reusable Object-Oriented Software - Erich Gamma, Richard Helm, ... [Addison-Wesley Professional]
- Refactoring: Improving the Design of Existing Code - Martin Fowler [Addison-Wesley Professional]
- Structure and Interpretation of Computer Programs - Harold Abelson, Gerald Jay Sussman
- Dataflow and Reactive Programming Systems: A Practical Guide - Matt Carkci [CreateSpace Independent Publishing Platform]