



Universidade Federal de Pernambuco  
Centro de Informática

Graduação em Ciência da Computação

## **Extensão e análise de performance da biblioteca CEPlin**

Proposta de Trabalho de Graduação

**Aluno:** Dicksson Rammon Oliveira de Almeida (droa@cin.ufpe.br)

**Orientador:** Kiev Santos da Gama (kiev@cin.ufpe.br)

Recife, 10 de setembro de 2018

## **Sumário**

<b>1. Contextualização</b>	<b>3</b>
<b>2. Objetivos</b>	<b>4</b>
<b>3. Cronograma</b>	<b>5</b>
<b>4. Possíveis Avaliadores</b>	<b>6</b>
<b>5. Referências</b>	<b>7</b>

## 1. Contextualização

A necessidade de processamento de fluxos de informação cresceu significativamente nos últimos anos, com sistemas, ferramentas e aplicações surgindo em grande número. A pesquisa e desenvolvimento nessa área produziu uma variedade de conceitos, desde a programação e arquiteturas orientadas a eventos à processamento de eventos complexos (*Complex Event Processing - CEP*) [3].

Para entender o funcionamento de sistemas de processamento de eventos complexos, também é útil entender os conceitos de programação reativa, um paradigma de programação declarativa; tais conceitos são aplicados por linguagens reativas (*Reactive Languages - RLs*). Similar ao padrão de software *Observer*, linguagens reativas interessam-se em propagação de mudanças no sistema e com fluxos de valores de várias fontes que mudam ao longo do tempo, notificando as partes interessadas de forma implícita.

Sistemas baseados em CEP, por sua vez, lidam com fluxos de eventos, acionados através de mudanças no ambiente. Tais sistemas realizam operações sobre esses eventos, como a filtragem, combinação e agregação, a fim de detectar eventos de mais alto nível; similar à RLs, esses sistemas também realizam a notificação às partes interessadas das ocorrências dos eventos.

Pelo exposto acima, existe um relacionamento entre CEP e RLs; todavia, esse relacionamento foi pouco explorado pelas comunidades científica e desenvolvedora. Sistemas baseados em CEP normalmente utilizam uma linguagem de consulta - similar à *SQL* e suas extensões - o que é pouco intuitivo e propício a erros [3]. As semelhanças entre CEP e RLs permitem que sistemas baseados no primeiro lidem com o fluxo de eventos de forma declarativa, algo proporcionado pelo segundo.

Pesquisas recentes acerca do relacionamento entre os dois conceitos resultaram na implementação de ferramentas baseadas em CEP a partir de uma abordagem reativa, usando RLs como complemento. As bibliotecas CEPSwift [7] e CEPlin [8], para uso em aplicações escritas em Swift e Kotlin, respectivamente, foram construídas com esse propósito.

## **2. Objetivos**

Este trabalho pretende dar continuação à biblioteca CEPLin, uma das bibliotecas citadas anteriormente. Entre os trabalhos futuros elencados em [8], estão a implementação de operadores adicionais, além de uma análise de performance da biblioteca. Portanto, este trabalho propõe-se a aumentar a quantidade de operadores suportados pela biblioteca, a fim de expandir a expressividade da mesma, além de fazer uma análise comparativa de performance de diferentes implementações dos operadores, baseando-se em métricas de análise de performance existentes.

### 3. Cronograma

Atividades	Mês															
	Agosto		Setembro			Outubro			Novembro			Dezembro				
Definição do escopo	█	█														
Formulação da proposta		█	█													
Revisão bibliográfica			█	█	█											
Estudo e análise das ferramentas				█	█	█	█									
Implementação						█	█	█	█	█	█					
Escrita do documento								█	█	█	█	█				
Preparação da defesa											█	█	█			
Defesa														█		

#### **4. Possíveis Avaliadores**

Os seguintes professores são considerados como possíveis avaliadores do trabalho proposto:

- Márcio Lopes Cornélio (mlc2@cin.ufpe.br)
- Ricardo Massa Ferreira Lima (rmfl@cin.ufpe.br)

## 5. Referências

- [1] PASCHKE, Adrian; KOZLENKOV, Alexander; BOLEY, Harold. A homogeneous reaction rule language for complex event processing. arXiv preprint arXiv:1008.0823, 2010.
- [2] AGUILERA, Marcos K. et al. Matching events in a content-based subscription system. In: Proceedings of the eighteenth annual ACM symposium on Principles of distributed computing. ACM, 1999.
- [3] CUGOLA, Gianpaolo; MARGARA, Alessandro. Processing flows of information: From data stream to complex event processing. ACM Computing Surveys (CSUR), v. 44, n. 3, 2012.
- [4] MARGARA, Alessandro; SALVANESCHI, Guido. Ways to react: Comparing reactive languages and complex event processing. REM, 2013.
- [5] ReactiveX, RxKotlin Project. Disponível em <<https://github.com/ReactiveX/RxKotlin>>. Acessado em 27 de março de 2018.
- [6] TEYMOURIAN, Kia; PASCHKE, Adrian. Enabling knowledge-based complex event processing. In: Proceedings of the 2010 EDBT/ICDT Workshops. ACM, 2010.
- [7] BELO, George. CEPsSwift: Complex Event Processing Framework for Swift. TG. 2017.
- [8] LINS, Jonas. CEPLin: A Complex Event Processing Framework for Kotlin. TG. 2018.

## **Assinaturas**

---

Dicksson Rammon Oliveira de Almeida  
(Aluno)

---

Kiev Santos da Gama  
(Orientador)