

Memorial Descritivo

Francisco Sant’Anna

Rio de Janeiro, 3 de março de 2015

Introdução

Brasileiro, nascido no Rio de Janeiro em 1979, Doutor em Ciência da Computação pela PUC–Rio em 2014. Estudei do primário até o vestibular no CAp–UERJ. Em 1997, ingressei na turma de Engenharia da Computação da UFRJ e cursei os dois primeiros anos, me transferindo em 1999 para a PUC–Rio, onde me formei em 2003. Em paralelo à universidade trabalhei em diversas empresas, nas áreas de segurança de redes, desenvolvimento Web, e sistemas embarcados. Após terminar a faculdade, atuei por três anos na área de jogos e aplicativos para celular. Em 2007 retornei à PUC–Rio, onde fiz o mestrado e doutorado na área de linguagens de programação com o professor Roberto Ierusalimsky.

Meu interesse por linguagens de programação iniciou-se na graduação, cursando disciplinas específicas com o professor Roberto, e seguindo com o projeto final, uma linguagem para simular circuitos digitais. No mestrado desenvolvi uma extensão que faz a linguagem Lua funcionar de forma reativa, com aplicações guiadas exclusivamente por estímulos externos. Além da pesquisa da dissertação, atuei no desenvolvimento do padrão de TV Digital brasileiro, especificamente na integração entre as linguagens NCL e Lua. O doutorado continuou na linha de sistemas reativos, onde projetei a linguagem CÉU com foco em verificações estáticas para sistemas concorrentes de tempo real. Durante esse período, fiz doutorado sanduíche na Suécia onde aplicamos CÉU à área de redes de sensores sem fio. Após o doutorado, iniciei o pós doutorado na PUC–Rio em 2013, onde continuo com o desenvolvimento de CÉU.

Fui professor do Instituto de Tecnologia ORT (2º grau e escola técnica) por dois anos, tendo lecionado as disciplinas de Introdução à Programação (1º ano), Tópicos Especiais em Programação (Jogos e Redes de Sensores sem Fio, 3º ano) e Projeto Final (3º ano). Na PUC–Rio, planejei em parceria com a Professora Noemi Rodriguez a disciplina Sistemas Reativos, que está sendo lecionada por mim pelo segundo semestre consecutivo.

Pesquisa

O foco de minha pesquisa é em linguagens reativas e concorrentes. Em sistemas reativos, o controle de aplicações é predominantemente guiado por interações contínuas e em tempo real com o mundo externo. Interações com o ambiente têm natureza reativa: um estímulo externo (*input*) gera uma reação no programa que pode gerar um estímulo de volta ao ambiente (*output*). Por exemplo, em um sistema de janelas, o movimento do mouse (*input*) altera instantaneamente a posição do ponteiro na tela (*output*). Os principais desafios dentro desse contexto são o de manter o sistema “responsivo” (i.e., com resposta idealmente imediata) e o de tratar estímulos simultâneos/concorrentes de forma correta.

A área de sistemas reativos me acompanhou (e me desafiou) durante a trajetória como programador e pesquisador: no estágio com sistemas embarcados, no projeto final com sistemas digitais,

na indústria de jogos, na pesquisa com TV Digital, e finalmente, como objeto fim na pesquisa em linguagens reativas durante o mestrado e doutorado.

Como pesquisador, vivenciei três projetos bem delimitados em escopo e experiências: o padrão de TV Digital brasileiro, a linguagem *LuaGravity*, desenvolvida no mestrado, e a linguagem CÉU, desenvolvida no doutorado.

Entre 2007 e 2009, fui pesquisador do Laboratório Telemídia, coordenado pelo professor Luiz Fernando Gomes Soares, idealizador do Sistema Brasileiro de TV Digital (conhecido como Ginga). Através do Ginga, emissoras podem distribuir aplicações que executam em sincronia com programas de TV, permitindo que telespectadores interajam em tempo real. A linguagem declarativa NCL, desenvolvida para esse fim, pode ser estendida com scripts escritos em Lua (de maneira análoga a como JavaScript estende HTML). Durante o período no Telemídia, fui o responsável pelo design, documentação e implementação da ponte *NCLua* entre as linguagens NCL e Lua. *NCLua* é usado para auxiliar NCL em tarefas difíceis de realizar declarativamente, tais como comunicação por TCP/IP e interações mais complexas (e.g., jogos). O principal artigo referente a esse período foi publicado no *WebMedia* em 2008 [1], simpósio nacional onde trabalhos relacionados ao Ginga são geralmente publicados. Além do artigo, fui co-autor do capítulo sobre *NCLua* do livro de NCL [6]. Também escrevi a parte da norma ABNT referente ao *NCLua* [7]. Finalmente, escrevemos um artigo com viés mais teórico para a conferência *DocEng* em 2009 [8].

O mesmo período entre 2007 e 2009 constituiu o mestrado sob a orientação do professor Roberto. O mestrado foi uma experiência de “assimilação” do modo de operação da academia. A abrangência e o rigor da literatura acadêmica foram fundamentais para a formulação e reflexão sobre os problemas encontrados em sistemas reativos, assim como para o levantamento do estado da arte da área. Com o decorrer da pesquisa, identificamos duas abordagens complementares que visavam facilitar o desenvolvimento e verificação de sistemas reativos: o estilo imperativo da linguagem Esterel e o estilo declarativo das linguagens funcionais reativas. A dissertação de mestrado discutiu essas duas abordagens com base no design da linguagem *LuaGravity*, uma extensão em tempo de execução para a linguagem Lua. Com *LuaGravity* é possível escrever, de maneira imperativa, comandos reativos tais como `await E` para bloquear uma linha de execução até que o evento `E` ocorra, ou ainda, de maneira declarativa, expressões tais como `a=b+c` em que mudanças em `b` ou `c` atualizam o valor de `a` automaticamente. O trabalho com o *LuaGravity* foi publicado no *SBLP* de 2009 [5] e apresentado no *Lua workshop* de 2009.

O período entre 2009 e os dias de hoje compreende o doutorado e o desenvolvimento da linguagem CÉU. CÉU nasceu com requisitos mais rigorosos que *LuaGravity*: eficiência no uso de recursos, garantias estáticas para concorrência e apoio em uma semântica reativa precisa. Como consequência, também direcionei os casos de uso para aplicações nas quais eficiência e garantias estáticas tornam-se diferenciais, mesmo com alguma limitação em expressividade: Redes de sensores sem fio (RSSF) executam em hardware bastante restrito e dependem de parcimônia no uso de bateria, já que são instalados em locais de acesso físico difícil (florestas, vulcões, etc.). Utilizamos CÉU para desenvolver drivers e protocolos já existentes para RSSFs, com resultados equivalentes aos desenvolvidos originalmente em C (e.g., uso de CPU e memória). Como melhorias, as implementações em CÉU têm garantias teóricas de responsividade e de ausência de condições de corrida em variáveis compartilhadas. O design de CÉU foi publicado em 2013 no *SenSys* [3], a conferência de maior prestígio na área de RSSFs. Com o fim do doutorado, o foco foi em expandir o poder de expressividade de CÉU, em particular no suporte a aplicações com alocação dinâmica. Como resultado, a nova abstração de *organismos* unifica objetos e *threads* em uma mesma construção, sem abrir mão das garantias estáticas do design original de CÉU. Esse conceito foi apresentado no *Future Programming Workshop* e *Workshop on Reactive and Event-based Languages & Systems* que aconteceram dentro do *OOPSLA/SPLASH* em 2014. O artigo completo será publicado na conferência *Modularity/AOSD* em 2015 [4].

CÉU é um projeto de longo prazo, com diversas opções de continuidade em pesquisa, ensino e

também para o mercado de programadores. Esse ano, fui convidado a apresentar CÉU no encontro anual do *Working Group on Language Design*, grupo de trabalho vinculado ao *IFIP (International Federation for Information Processing)*. No ano passado apresentei CÉU no *StrangeLoop*, conferência não acadêmica voltada para programadores profissionais. Com relação a ensino, CÉU faz parte do programa de algum curso ou disciplina desde 2011: lecionei RSSF para o 3º ano técnico da escola ORT por dois anos; na PUC–Rio, planejei a disciplina eletiva *Sistemas Reativos*, que esse ano está sendo dada pela segunda vez. Fui informado (por e-mail, de maneira informal) sobre o uso de CÉU na disciplina de RSSF de Stanford¹ e de Compiladores em Berkeley². CÉU participou do *Google Summer of Code* em 2014 com um estudante da Índia em parceria com o LabLua³. Finalmente, estamos estabelecendo uma comunidade de CÉU, com website, documentação, wiki, vídeos, lista de discussão, e implementação open source para Arduino, redes de sensores, smartphones e desktops.⁴⁵

¹<http://web.stanford.edu/class/cs240e/>

²<https://www.youtube.com/watch?v=hC5i-Wr4FuA>

³<https://www.google-melange.com/gsoc/org2/google/gsoc2014/lablua>

⁴<http://www.ceu-lang.org/>

⁵<http://github.com/fsantanna/ceu/>

Títulos e Trabalhos (para análise da banca)

Qualificação Acadêmica (QUADRO 1a)

- Graduação em Engenharia de Computação, 2003, PUC-Rio. — [Anexo-1 (diploma)]
- Mestrado em Informática, 2009, PUC-Rio. — [Anexo-2 (diploma)]
- Doutorado em Informática, 2013, PUC-Rio. — [Anexo-3 (diploma)]
- Pós-Doutorado em Informática, 2013-2018 (2º ano em andamento), PUC-Rio. — [Anexo-4 (comprovação do departamento)]

Publicações e Produção Científica (QUADRO 2)

- Modularity 2015 (AOSD até 2011, Qualis-A1, **a ser publicado**) “Structured Synchronous Reactive Programming with Céu” [4] — [Anexo-5 (primeira página, programa)]
- ACM SenSys 2013 (Qualis-A1) “Safe System-Level Concurrency on Resource-Constrained nodes” [3] — [Anexo-6 (primeira página, página na biblioteca digital da ACM)]
- ACM DocEng 2009 (Qualis-B1) “Relating Declarative Hypermedia Objects and Imperative Objects through the NCL Glue Language” [8] — [Anexo-7 (primeira página, página na biblioteca digital da ACM)]
- SBLP 2009 (Qualis-B3) “LuaGravity, a Reactive Language Based on Implicit Invocation” [5] — [Anexo-8 (primeira página, publicações sobre a linguagem Lua no site oficial)]
- WebMedia 2008 (Qualis-B3) “NCLua: Objetos Imperativos Lua na Linguagem Declarativa NCL” [1] — [Anexo-9 (primeira página, página na biblioteca digital da ACM)]
- Capítulo de livro 2009 “Programando com Objetos NCLua. Programando em NCL” [6] — [Anexo-10 (capa do livro, primeira página do capítulo)]
- Capítulo de livro 2009 (minicurso) “Desenvolvimento de Aplicações Híbridas para TV Digital Interativa no Middleware Ginga” [2] — [Anexo-11 (primeira página, site)]

Sobre os critérios de pontuação para produção científica do edital (Anexo II, Quadro 2, página 17), sugiro analisar a separação entre publicações em periódicos e conferências acadêmicas, uma vez que, segundo o documento de área da Capes para Ciência da Computação, os dois meios de publicação são considerados como equivalentes na avaliação trienal de departamentos⁶:

“Durante os três últimos períodos de Avaliação Trienal (2004, 2007 e 2010) a área de Ciência da Computação trabalhou com publicações em conferências e periódicos. Na área, as publicações submetidas a conferências tradicionais passam por um rigoroso processo de avaliação por pares e os artigos publicados, disponíveis em bases de dados internacionais, são hoje tão importantes para o avanço da área como os melhores artigos em veículos classificados de periódicos. Qualquer pesquisador da área de Ciência da Computação sabe que há conferências de enorme prestígio e que os artigos publicados nos anais dessas conferências são levados em alta conta em avaliações de pesquisa. Há documentos, inclusive do IEEE, enfatizando a importância das conferências para a área. Como o conjunto de publicações em conferências e periódicos é essencial para

⁶http://www.capes.gov.br/images/stories/download/avaliacaotrienal/Docs_de_area/, Ciência da Computação, Seção IV, páginas 15-16.

a avaliação da área em qualquer parte do mundo, a avaliação da produção bibliográfica compreende os veículos chamados tradicionalmente de periódicos e de anais de conferências.” (— [Anexo-12])

O critério do edital pontua periódicos e congressos de maneira desproporcional (máximo de 60 e 10 pontos, respectivamente) e desconsidera por completo a qualidade dos congressos. Por exemplo, artigos completos de 12 páginas em congressos Qualis-A valem 2 vezes menos que artigos em periódicos Qualis-B5. Minha produção se concentra em conferências de qualidade, por exemplo, a ACM SenSys e Modularity são ambas Qualis-A1.

Docência e Atividade Profissional (QUADRO 4)

- Professor do Instituto de Tecnologia ORT, 2010–2012 (2 anos) — [Anexo-13 (carteira de trabalho)]
- Professor da disciplina *Sistemas Reativos* na PUC-Rio, 2014–2015 (1 semestre) — [Anexo-14 (e-mail para os alunos da graduação)]
- Programador de jogos para celular, Wiz Technologies, 2004–2007 (2 anos) — [Anexo-13 (carteira de trabalho)]

Referências

- [1] SANT'ANNA, F., ET AL. NCLua - Objetos Imperativos Lua na Linguagem Declarativa NCL. In *WebMedia* (2008), ACM. In portuguese.
- [2] SANT'ANNA, F., ET AL. *Desenvolvimento de Aplicações Declarativas para TV Digital no Middleware Ginga com Objetos Imperativos Lua*. 2009, p. 40.
- [3] SANT'ANNA, F., ET AL. Safe System-level Concurrency on Resource-Constrained Nodes. In *Proceedings of SenSys'13* (2013), ACM.
- [4] SANT'ANNA, F., ET AL. Structured Synchronous Reactive Programming with Céu. In *Proceedings of Modularity'15* (2015). to appear.
- [5] SANT'ANNA, F., AND IERUSALIMSKY, R. LuaGravity, a Reactive Language Based on Implicit Invocation. In *Proceedings of SBLP'09* (2009), pp. 89–102.
- [6] SANT'ANNA, F., AND SALLES, C. *Programando em NCL 3.0*. 2009, ch. Programando com Objetos NCLua.
- [7] SOARES, L. F. G., ET AL. *Digital terrestrial television – Data coding and transmission specification for digital broadcasting – Part 2: Ginga-NCL for fixed and mobile receivers – XML application language for application coding*. SBTVD Forum, 2008. Norma ABNT NBR 15606-2:2007.
- [8] SOARES, L. F. G., ET AL. Relating Objects with Imperative Code and Objects with Declarative Code through NCL Glue Language. In *DocEng* (2009), ACM.