Universidade Federal de Pernambuco
Centro de Informática
Graduação em Ciência da Computação

Swicity: Visualizando Sistemas de Software em Swift como cidades

Proposta de Trabalho de Graduação

Aluno: **Rafael Nunes Galdino da Silveira**

Orientador: **Fernando José Castor de Lima Filho**

Recife
Setembro de 2016

Resumo

Este trabalho tem por objetivo estudar a implementação e desenvolvimento de sistemas na linguagem de programação Swift. Seu resultado mais concreto é uma plataforma de visualização de software que explore métricas relativas a elementos de código Swift. Também vista possibilitar uma visão diferenciada do código, podendo gerar *insights* e observações das consequências práticas, comportamentais e teóricas da implementação dos sistemas. Para isso será criada uma ferramenta de visualização de software 3D baseada na metáfora de cidades.

palavras-chave: swift, metáfora de cidades, visualização de software, *city metaphor*.

Sumário

1. Introdução

Swift[1] teve sua primeira versão lançada em 2014 e é uma linguagem de programação multiparadigma, pois orquestra diferentes paradigmas como funcional, orientado a protocolo, orientado a objetos e imperativo. Swift, atualmente na versão 2.2.1, vem sendo desenvolvida pela Apple para substituir Objective-C como a linguagem de desenvolvimento para suas plataformas iOS, OS X, watchOS, tvOS, além de ter também uma versão disponível para o sistema operacional Linux.

Apesar de nova, Swift tem uma curva de adoção bastante significante segundo o GitHub[2] e segundo pesquisas recentes sobre a utilização de linguagens de programação. Diversas aplicações *iOS* que são *open-source* já foram desenvolvidas em Swift e estão disponíveis no GitHub. Em uma lista[3] que contém cerca de 400 aplicativos *iOS*, com mais de 1,000 *commits* e 88 contribuidores, 162 desses aplicativos foram desenvolvidos em Swift.

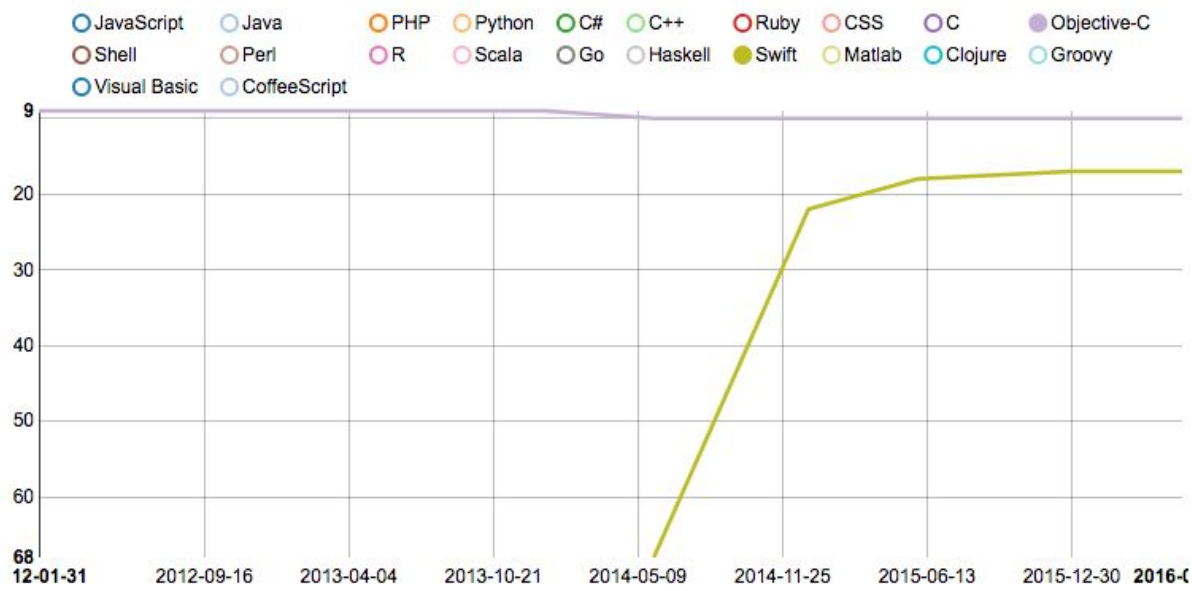

Figura 1. *Programming Language Rank: Position vs. Dates over the years.*
The RedMonk Programming Language Rankings: June 2016.

Na Figura 1, podemos observar alguns resultados da pesquisa feita por O'Grady[4] em Junho de 2016. É notória a crescente adoção de Swift que partiu em 2014 da 68º posição, já aparece no top 20 linguagens mais utilizadas do GitHub, agora na 17º posição. Enquanto isso, Objective-C manteve-se estável e isso condiz com a proposta da Apple de dar suporte aos programas desenvolvidos em suas antigas plataformas, assim como o crescimento de Swift mostra que cada vez mais a linguagem cumpre seu papel de substituir a utilização de Objective-C.

[1] https://swift.org
[2] http://githut.info
[3] https://github.com/dkhamsing/open-source-ios-apps
[4] http://redmonk.com/sogrady/2016/07/20/language-rankings-6-16/

2. Motivação

Apesar da crescente e notória adoção de Swift, poucos iniciativas e estudos trouxeram até agora alternativas para análise de código Swift. Além disso, sabemos que o crescimento da complexidade do código e da complexidade em sistemas de software modernos sobrecarregam a capacidade natural dos engenheiros e times de desenvolvimento de compreender, gerenciar e identificar o comportamento do sistema apenas olhando para o código [5]. Para tentar mitigar esse problema, técnicas de visualização de código e software são normalmente utilizadas. Técnicas de visualização 2D, por exemplo, já são amplamente utilizadas e difundidas como em [1, 2, 3], mas também algumas iniciativas novas de visualização vêm surgindo, como as técnicas em visualização 3D [4, 5, 6].

Wettel propõe uma visualização de software 3D seguindo uma nova abordagem que ele chama de "metáfora de cidades", onde os elementos do código são mapeados e visualizados no contexto de vizinhança, blocos, prédios [6]. Tal metáfora já está sendo utilizada e adotada para o estudo e visualização de sistemas e aspectos de código como pode ser visto em [4, 5].

Apesar da visualização de software ser uma ótima ferramenta para o entendimento do desenvolvimento e evolução do código, pouco disso foi aplicado e explorado no contexto da análise de código Swift, em particular, e no ecossistema da Apple, em geral. Sendo assim, é interessante explorar uma abordagem de visualização tal como a "metáfora de cidades" pondo Swift como contexto de estudo.

3. Objetivo

O objetivo deste trabalho é fornecer uma plataforma que possibilite uma visão diferenciada do código, podendo gerar insights e observações das consequências práticas, comportamentais e teóricas da implementação de sistemas em Swift. Sendo assim será desenvolvida uma ferramenta de visualização de software 3D que construa uma visualização baseada na metáfora de cidades [6].

Para isso será necessário estudar as peculiaridades dos elementos e código de Swift para fazer um mapeamento dos mesmos para o conceito da metáfora, estudar também os aspectos relevantes e interessantes para o contexto da visualização e estudar uma maneira de implementá-los numa tecnologia que torne a plataforma acessível, interessante e útil.

4. Cronograma

Atividade	**Agosto**		**Setembro**		**Outubro**		**Novembro**		**Dezembro**	
Estudo de *Swift* e de sua utilização.										
Elaboração da proposta.										
Estudo e análise dos elementos do código Swift.										
Implementação da ferramenta.										
Avaliação da implementação e possíveis melhorias.										
Escrever o relatório final.										
Apresentação.										

5. Possíveis Examinadores

Kiev Gama
Francisco Soares Neto
Henrique Rebêlo
Paulo Borba

6. Referências Bibliográficas

[1] L. Voinea, A. Telea, and J. J. van Wijk. CVSscan: visualization of code evolution. In Proceedings of 2005 ACM Symposium on Software Visualization (Softviz 2005), pages 47–56, St. Louis, Missouri, USA, May 2005.

[2] M. Wilhelm and S. Diehl. Dependencyviewer - a tool for visualizing package design quality metrics. In VISSOFT, 2005

[3] S. Ducasse and M. Lanza. The class blueprint: Visually supporting the understanding of classes. Transactions on Software Engineering (TSE), 31(1):75–90, Jan. 2005.

[4] Viana, M., Moraes, E., Barbosa, G., Hora, A., & Valente, M. JSCity – Visualização de Sistemas JavaScript em 3D. Congresso Brasileiro de Software (CBSOFT), Set. 2015.

[5] Waller, J., Wulf, C., Fittkau, F., Dohring, P., & Hasselbring, W. (2013). Synchrovis: 3D visualization of monitoring traces in the city metaphor for analyzing concurrency. 2013 First IEEE Working Conference on Software Visualization (VISSOFT). doi:10.1109/vissoft.2013.6650520

[6] Wettel, R. and Lanza, M. (2007). Visualizing Software Systems as Cities. 2007 4th IEEE International Workshop on Visualizing Software for Understanding and Analysis. doi:10.1109/vissof.2007.4290706

7. Assinaturas

_____

Aluno: **Rafael Nunes Galdino da Silveira**

_____

Orientador: **Fernando José Castor de Lima Filho**