

## Subprojeto de Iniciação Científica

<b>Edital:</b>	Edital PIIC 2020/2021
<b>Título do Projeto:</b>	A rede de ontologias SFWON ( <i>Software Frameworks Ontology Network</i> ) e sua aplicação no método <i>FrameWeb</i> ( <i>Framework-based Design Method for Web Engineering</i> )
<b>Título do Subprojeto:</b>	<i>Plug-in</i> para análise de código fonte
<b>Candidato a Orientador:</b>	Vítor Estêvão Silva Souza
<b>Candidato a Bolsista:</b>	(A definir)
<b>Membros Equipe do Projeto:</b>	Camila Zacche de Aguiar (Doutorado), Cleisson Santos Guterres (Mestrado)

### 1 Resumo

---

Dentro do contexto da Engenharia Web, o método Frameweb foi proposto a fim de auxiliar o projeto de sistemas de informação baseados na Web que utilizam *frameworks* em sua arquitetura. O método define uma arquitetura padrão para facilitar a integração com tais *frameworks* e propõe um conjunto de modelos que traz para o projeto arquitetural do sistema conceitos inerentes a estes *frameworks*. Uma vez que códigos fonte são gerados a partir desses modelos, este subprojeto tem por objetivo constuir um *plug-in* para analisar semanticamente esses códigos fonte.

*Palavras-chave:* Engenharia de Software. Análise de Código. *Code Smells*. Ontologias

### 2 Introdução

---

O método FrameWeb (Souza, 2007; Souza et al., 2009) define uma arquitetura padrão para facilitar a integração com *frameworks* utilizados no desenvolvimento de software. O método propõe um conjunto de modelos baseados na UML que trazem para o projeto arquitetural do sistema conceitos inerentes a estes *frameworks* e, adicionalmente, gera código fonte a partir desses modelos a fim de facilitar o trabalho do programador.

No entanto, más escolhas podem ser feitas durante o processo de desenvolvimento de modo a produzir *bad smells*, soluções comuns que não necessariamente falham ou erram, mas mesmo assim apresentam sintomas que indicam um problema (Beck et al., 1999). Portanto, *Code Smells* emergem de problemas de código fonte local de estruturas de programação de baixo nível e podem causar problemas no desenvolvimento, manutenção e evolução do software (de Paulo Sobrinho et al., 2018). A fim de auxiliar o desenvolvimento de software e contribuir para a sua

qualidade, ferramentas para detecção de *smells* tem sido elaboradas.

Nesse contexto, o método O-SCAN (Aguiar et al., 2020) propõe uma técnica de detecção de *smells* baseada em análise sintática e semântica de código fonte para múltiplas linguagens de programação. O método propõe uma sequência de atividades agrupadas em fases que produzem um conjunto de artefatos que serão posteriormente reunidos para definir uma técnica de detecção de *smells*. As fases são organizadas em: (i) Abstração Sintática para analisar o código fonte em sua forma elementar e representar suas construções em sintaxe estruturada seguindo a gramática da linguagem de programação; (ii) Abstração Semântica para analisar a estrutura sintática gerada na fase anterior e representar as informações do código fonte em um modelo semântico, isto é, uma ontologia; e (iii) Definição de *Smells* para analisar a estrutura semântica gerada na fase anterior e produzir regras capazes de identificar *smells* no código fonte.

Este subprojeto tem por objetivo desenvolver um *plug-in* de análise de código fonte baseado no método O-SCAN a fim de possibilitar a fácil aplicação do método pelos desenvolvedores.

### 3 Objetivos

---

O objetivo geral desse subprojeto é construir um *plug-in* de análise de código fonte para uma IDE de modo que seja possível analisar o código fonte de projetos de software em diferentes linguagens de programação. O *plug-in* será construído sob uma arquitetura genérica de modo que novos insumos relacionados ao domínio de código fonte, a linguagem de programação e ao tipo de smell possam ser adicionados de forma evolutiva. No primeiro momento, o *plug-in* cobrirá os conceitos do domínio de orientação a objetos, da linguagem Java e um *smell* específico, chamado *Long Parameter List*.

São objetivos específicos deste subprojeto:

- Conhecer análise de código e o método O-SCAN por meio da descrição do método e da sua prova de conceito;
- Conhecer os conceitos e a sintaxe da linguagem Java, conforme a especificação da linguagem;
- Conhecer os recursos tecnológicos disponíveis para o desenvolvimento do *plug-in* na IDE escolhida;
- Aprender sobre *parser* sintático e como utilizá-lo de maneira genérica no *plug-in*;
- Aprender sobre ontologia operacional e como utilizá-la de maneira genérica no *plug-in*;
- Desenvolver o *plug-in* de análise de código para a IDE escolhida, de modo que desenvolvedores possam analisar o código fonte do projeto desenvolvido na IDE escolhida.

No contexto do projeto de pesquisa “A rede de ontologias SFWON (*Software Frameworks Ontology Network*) e sua aplicação no método *FrameWeb* (*Framework-based Design Method for Web Engineering*)”, este subprojeto visa contribuir com o objetivo “O4. Ferramentas: criação de novas ferramentas e aprimoramento das ferramentas existentes de modo a permitir a apropriação do método por parte da indústria e da própria academia”.

## 4 Metodologia

---

Este subprojeto será feito no contexto de um trabalho de pós-graduação, envolvendo, portanto, além do(a) aluno(a) de Iniciação Científica (IC) e do professor orientador, também estudantes de pós-graduação que possuem trabalhos relacionados ao tema. Em particular, a aluna de doutorado Camila Zacche de Aguiar atuará diretamente na orientação do(a) aluno(a) de IC, por ser a proponente do método O-SCAN que motivou este subprojeto.

O trabalho será iniciado com leitura de artigos sobre análise de código fonte e do método O-SCAN (Aguiar et al., 2020). Em seguida, o(a) aluno(a) de IC irá conhecer e praticar tecnologias que poderão auxiliar o desenvolvimento do plug-in, tais como, ANTLR,<sup>1</sup> Jena<sup>2</sup> e Visual Studio Code.<sup>3</sup> Esta última, a princípio, será a IDE escolhida para o desenvolvimento do plug-in.

Após estas fases preparatórias, o(a) aluno(a) se concentrará nas atividades principais deste subprojeto: o desenvolvimento do *plug-in* de análise de código fonte para a IDE escolhida. Para isso, o(a) aluno(a) deverá aprofundar os estudos nas tecnologias escolhidas, modelar e implementar a solução. Neste contexto, contará com o apoio direto dos estudantes de pós-graduação que fazem parte do projeto.

Ao final do período de Iniciação Científica, espera-se conseguir utilizar o plug-in sobre um projeto Java a fim de analisar seu código fonte. Além disso, prevê-se a participação do(a) aluno(a) na escrita de um artigo científico junto aos demais membros do projeto sobre o desenvolvimento do *plug-in*.

## 5 Plano de trabalho/Cronograma

---

Esta seção descreve as atividades que serão desenvolvidas pelo(a) aluno(a) de IC e seu cronograma de execução para que o objetivo deste subprojeto possa ser alcançado.

---

<sup>1</sup><https://www.antlr.org>

<sup>2</sup><https://jena.apache.org>

<sup>3</sup><https://code.visualstudio.com>

## ATIVIDADES

Lista de atividades
1 - Estudo sobre análise de código fonte e método O-SCAN com leitura de artigos e reuniões de alinhamento com os integrantes do grupo para entendimento do método;
2 - Estudo sobre <i>parsing</i> sintático e sobre os conceitos e sintaxe da linguagem de programação Java;
3 - Estudo sobre ontologia e sobre os conceitos da orientação a objetos;
4 - Estudo sobre os recursos tecnológicos disponíveis para o desenvolvimento do <i>plug-in</i> na IDE escolhida;
5 - Implementação de um <i>plug-in</i> na IDE escolhida;
6 - Teste do <i>plug-in</i> desenvolvido sobre projetos Java;
7 - Elaboração dos relatórios técnicos obrigatórios do PIIC e de um artigo científico relatando os resultados alcançados durante a Iniciação Científica.

## CRONOGRAMA (Ago/2019 a Jul/2020)

Atividade	ago	set	out	nov	dez	jan	fev	mar	abr	mai	jun	jul
1	X	X										
2		X	X									
3			X	X								
4		X	X	X	X	X	X					
5						X	X	X	X	X		
6									X	X	X	X
7						X					X	X

## Referências

- Aguiar, C. Z., Zanetti, F. L., & Souza, V. E. S. (2020). An Ontology-based Method for Code Smells Detection. In *Submitted to the 34th Brazilian Symposium on Software Engineering (under review)*.
- Beck, K., Fowler, M., & Beck, G. (1999). Bad smells in code. *Refactoring: Improving the design of existing code*, 1:75–88.
- de Paulo Sobrinho, E. V., De Lucia, A., & de Almeida Maia, M. (2018). A systematic literature review on bad smells—5 w's: which, when, what, who, where. *IEEE Transactions on Software Engineering*.
- Souza, V. E. S. (2007). FrameWeb: um Método baseado em Frameworks para o Projeto de

Sistemas de Informação Web. Technical report, Universidade Federal do Espírito Santo, Vitória, ES.

Souza, V. E. S., Falbo, R. A., & Guizzardi, G. (2009). Designing Web Information Systems for a Framework-based Construction. In Halpin, T., Proper, E., & Krogstie, J., editors, *Innovations in Information Systems Modeling: Methods and Best Practices*, chapter 11, pages 203–237. IGI Global, 1 edition.