XI CONGRESSO DE INICIAÇÃO CIENTÍFICA DA UNIVERSIDADE FEDERAL DE CAMPINA GRANDE

PIBIC/CNPq/UFCG-2014

# *CARACTERIZAÇÃO DO VOCABULÁRIO DE COMMITS AO LONGO DA EVOLUÇÃO DE SISTEMAS*

**João Pedro Ferreira de Melo Leôncio[1], Jorge César Abrantes de Figueiredo[2]**

## RESUMO

As questões que envolvem o vocabulário de commits são essenciais para o estudo da evolução de sistemas. Essas questões caracterizam, muitas vezes, a forma como os sistemas envelhecem e, portanto, uma caracterização detalhada permite a identificação de aspectos importantes de sua evolução por meio de detecção de padrões. O desenvolvimento de ferramentas para a experimentação foi essencial para por conseguinte, com o objetivo de investigar melhor a variação do vocabulário de commits ao longo da evolução de sistemas, serem realizados experimentos que conseguiram captar aspectos importantes inerentes as métricas definidas inicialmente.

**Palavras-chave:** vocabulário, commit, evolução.

## CHARACTERIZATION OF THE COMMIT VOCABULARY ALONG SOFTWARE EVOLUTION

## ABSTRACT

The questions which involve the vocabulary of commits are essential for the study of systems evolution. These questions often caracterize the manner how the systems get old and, therefore, a detailed caracterization allows the identification of important aspects of its evolution by pattern detection. The development of tools for the experimentation was essential to consequently, in order to better investigate the variation of commits vocabulary over the systems evolution, perform experiments which captured important aspects connected to the metrics originally defined.

**Keywords: vocabulary, commit, evolution**

[1]Aluno do Curso de Ciência da Computação, Departamento de Sistemas e Computação, UFCG, Campina Grande, PB, e-mail: joao.leoncio@ccc.ufcg.edu.br
[2]Ciência da Computação, Professor Doutor, Departamento de Sistemas e Computação, UFCG, Campina Grande, PB, e-mail: abrantes@computacao.ufcg.edu.br

## INTRODUÇÃO

Nos cenários modernos de desenvolvimento de software, novos paradigmas de produção baseados em software livre ou em metodologias ágeis, não necessariamente respeitam todas Leis de Lehman (LEHMAN, 1980) que mostram tendências, limitações e relações na evolução de um software. Neste sentido, nos anos recentes, em decorrência de uma melhoria na qualidade de identificadores de um código fonte, o vocabulário de software, que é formado pelos termos distintos extraídos dos seus identificadores, passou a ser tema de interesse de pesquisadores da engenharia de software.

Durante a evolução do software, as mudanças contínuas sobre um sistema são normalmente registradas em sistemas de controle de versão (***VCS - Version Control Systems***), tais como Mercurial, Git, Subversion ou CVS, na forma de *commits*. Um *commit* é o termo usado para representar a operação de transferência das mais recentes alterações do código para o repositório de código, fazendo com que essas alterações passem a ser a versão corrente do código.

Cada mudança registrada no repositório contém, além de outras informações, dados sobre identificadores, comentários e JavaDocs[3] e, por meio disso, o vocabulário de software vai sendo consistituido e incrementado (KOKOL, 1999) (KUHN, 2007). Minerar repositórios na tentativa de caracterizar commits com relação ao vocabulário que compõe cada uma das mudanças contínuas pode revelar aspectos importantes que contribuam para entender e explicar o fenômeno do envelhecimento em sistemas em desenvolvimento.

## REVISÃO BIBLIOGRÁFICA

A revisão bibliográfica foi definida como primeira fase do trabalho em conjunto com um treinamento na condução de experimentos de extração de vocabulário de commits que é foco do estudo de evolução de sistemas sendo realizado. O objetivo era se familiarizar com as principais técnicas, o ferramental em geral e fazer um levantamento dos principais mecanismos para a extração dos commits.

Após o aprendizado sobre extração de vocabulário, a definição de métricas se deu com base nas métricas estabelecidas em (SANTOS, GUERRERO e FIGUEIREDO, 2012). As métricas definidas são o Total de Termos do Commit e Total de Entidades por commit.

## MATERIAIS E MÉTODOS

Os materiais utilizados foram associados a utilização do **DeveloperVocabularyExtractor**, um *script* responsável por automatizar o processo de identificação e captura do vocabulário das modificações realizadas das entidades Java em um *commit*.

Durante o processo faz-se necessário o uso de duas ferramentas desenvolvidas pela equipe de Evolução de Software do SPLab[4], o **VocabularyTools** e o **CommitVocabularyIdentify,** cujas informações encontram-se descritas a seguir.

O **VocabularyTools** consiste em um conjunto de soluções para extração, manipulação e análise de vocabulário de projetos Java. Foram desenvolvidas as ferramentas: VocabularyExtractor e TermsCounter (composto do VxlReader, IdentifierFilter, IR, MeasuresDispersion); e definido um formato de armazenamento: Schema XSD para VXL (Vocabulary XML). A Figura 2 apresenta uma visão geral desses artefatos de software e, como se dá o processamento sobre um código fonte Java.

[3] Documentação das classes Java do repositório

[4] Laboratório de Práticas de Software, localizado na UFCG

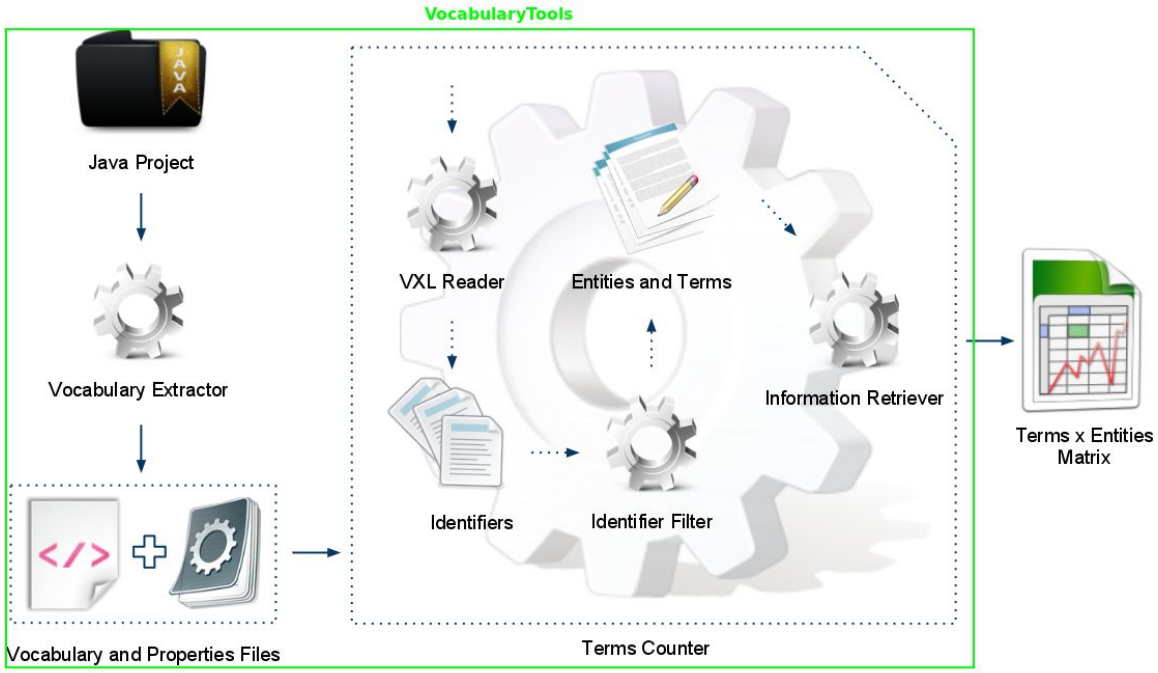


Figura 2. Representação do VocabularyTools

O **CommitVocabularyIdentify** é uma ferramenta responsável por identificar o vocabulário das alterações realizadas no código-fonte do projeto. Para isso, é necessário indicar o vocabulário do projeto anterior e posterior a realização do *commit*. De posse desses dados, a ferramenta gera um arquivo em formato *csv*[5] contendo apenas o vocabulário do que foi manipulado no *commit*.

Na Figura 1 podemos ver o fluxo de execução do **DeveloperMatrixExtractor** para o processo de identificação e captura do vocabulário dos desenvolvedores.

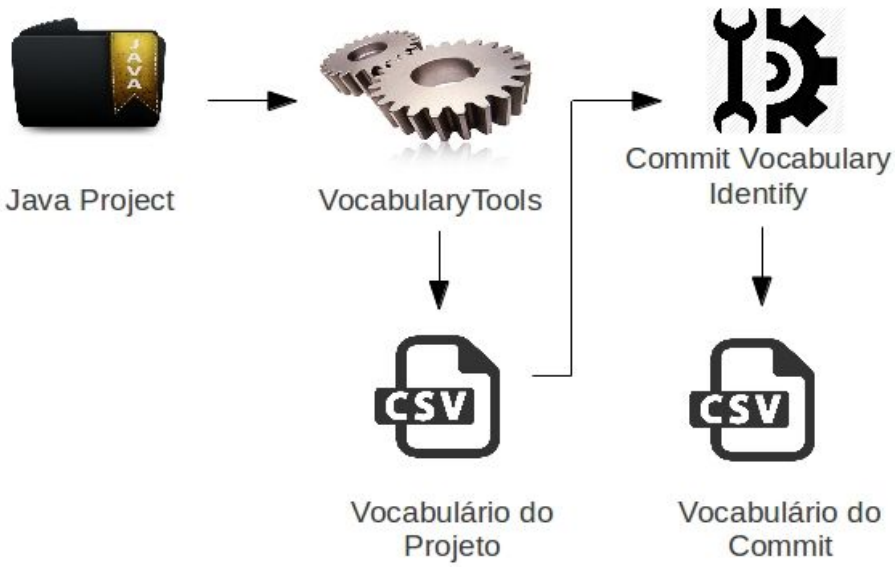


Figura 1. Representação do fluxo de execução do DeveloperMatrixExtractor

[5] Comma-separeted values: arquivo que armazena os dados em formato tabular

A execução destes experimentos foi precedida de uma fase de planejamento onde foram definidos com cuidado as métricas e parâmetros de configuração a fim de que o processo de avaliação estivesse bem fundamentado. Esse planejamento requer o estudo de trabalhos relacionados e embasamento teórico sobre extração de vocabulário de commits baseados em medição de algumas características ao longo da evolução do sistema, como também, entender o funcionamento do ferramental de extração; tal embasamento consistiu em atividade inicial do projeto. Após a execução dos experimentos foi conduzida a análise dos dados a partir de métodos estatísticos bem conhecidos. Seguindo essa metodologia, a realização desse projeto compreendeu as seguintes etapas:

- Embasamento teórico
- Treinamento para entender o funcionamento do ferramental de apoio
- Planejamento das métricas a serem aplicadas nos experimentos
- Condução dos experimentos
- Análise dos resultados dos experimentos
- Documentação e divulgação dos resultados

## RESULTADOS E DISCUSSÕES

A condução dos experimentos se deu a partir da utilização das métricas definidas para o projeto e utilização do ferramental para extrair os dados do estudo de caso.

### *O Estudo de Caso*

Para que fosse possível a execução dos experimentos foi necessário utilizar um estudo de caso real em que se pudesse ter a visualização dos resultados com base em um tempo significativo amostral.

O repositório utilizado para se realizar a extração é o do projeto e-Pol, um sistema informação gerencial que está sendo desenvolvido no Laboratório de Práticas de Software da Universidade Federal de Campina Grande em conjunto com a Polícia Federal. Foram coletados dados de commits de 4 meses de desenvolvimento, entre Outubro de 2013 e Fevereiro de 2014 e, a partir disso, foram geradas tabelas e gráficos que facilitaram a visualização e definição de resultados.

### *Os Resultados*

Para fins de análise de resultados, foram criadas mais dois parâmetros os quais depentem do Total de Termos e Total de Entidades, são eles Termos por Linha e Termos por Entidade, que apenas caracterizam melhor a evolução do sistema sob o um aspecto mais importante, o Total de Termos.

O Gráfico 1 trata duas variáveis do experimento em duas cores. A primeira, em azul, corresponde ao Total de Termos por commit ao longo da evolução do sistema. É possível visualizar que há momentos em que o Total de Termos cresce e, após isso, decresce. Isso aponta para grandes adições de termos em determinados períodos. O crescimento do Total de Termos mais destacado é o que ocorre no último mês de análise e mais perto do fim, quando a uma queda substancial no Total de Termos por commit. A segunda, em vermelho, corresponde ao Total de Entidades por commit durante o período total de execução. É possível visualizar que há momento em que não há um crescimento muito grande no Total de Entidades, tais qual é possível ver durante o segundo mês do repositório e picos de crescimento em periodos constantes, provavelmente semanais.

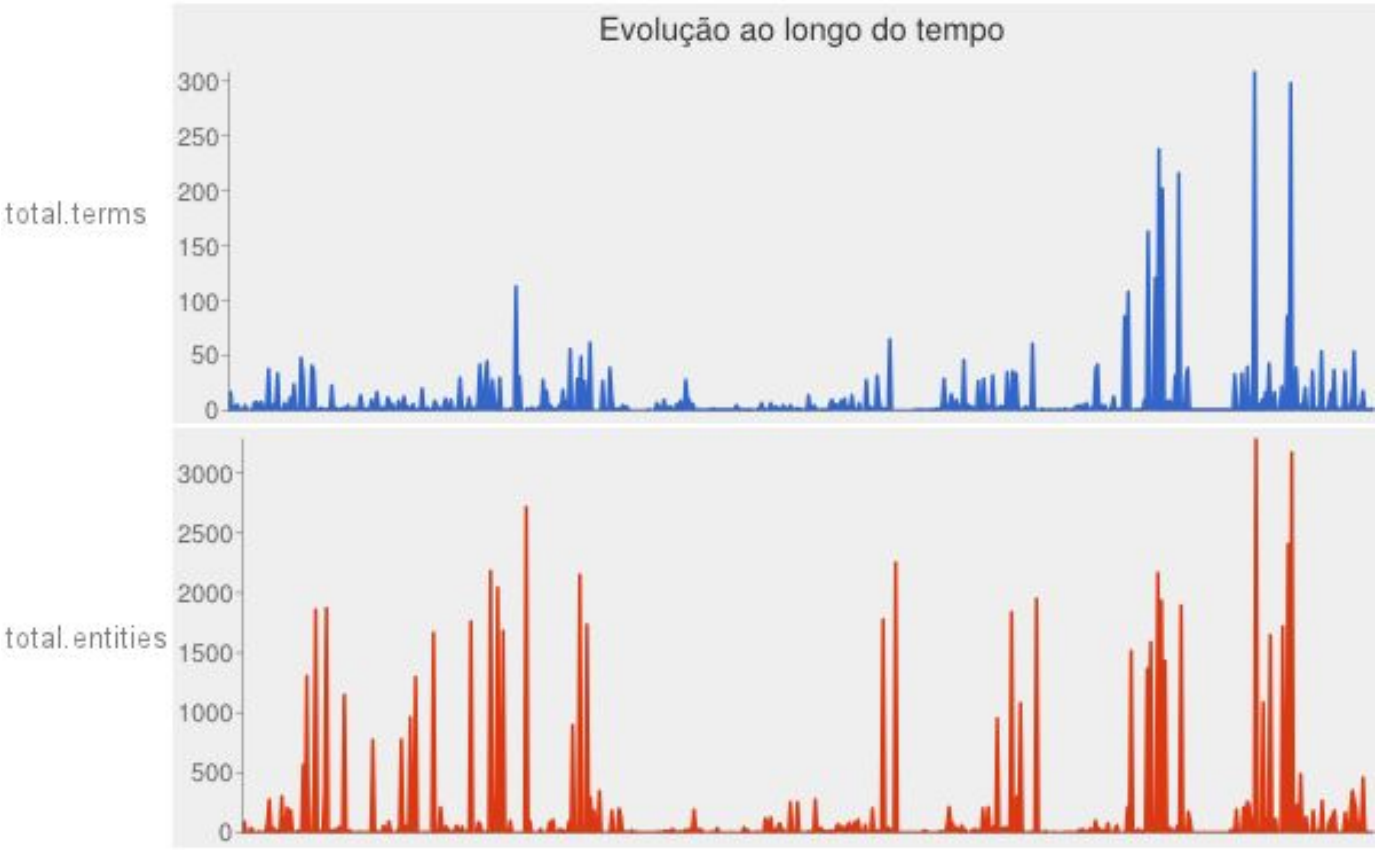


Gráfico 1: Total de entidades e Total de Termos ao longo do periodo do caso de teste por commit

Já os Gráfico 2 e 3 apontam para dois parâmetros criados para fins de análise em duas cores. A primeira, em azul, aponta o Total de Termos por Entidade durante o período extraído. É possível visualizar picos negativos porque há momentos em que o Total de Termos será negativo e sendo a quantidade de Entidades pequena, os picos se tornam maiores. No mais, há poucos momentos em que há mais adição de termos que entidades. O Gráfico 3, em vermelho, aponta que nos primeiros meses de coleta houve mais Termos por Linha que nos 2 últimos e no máximo houve a adição de 6 termos por linha.

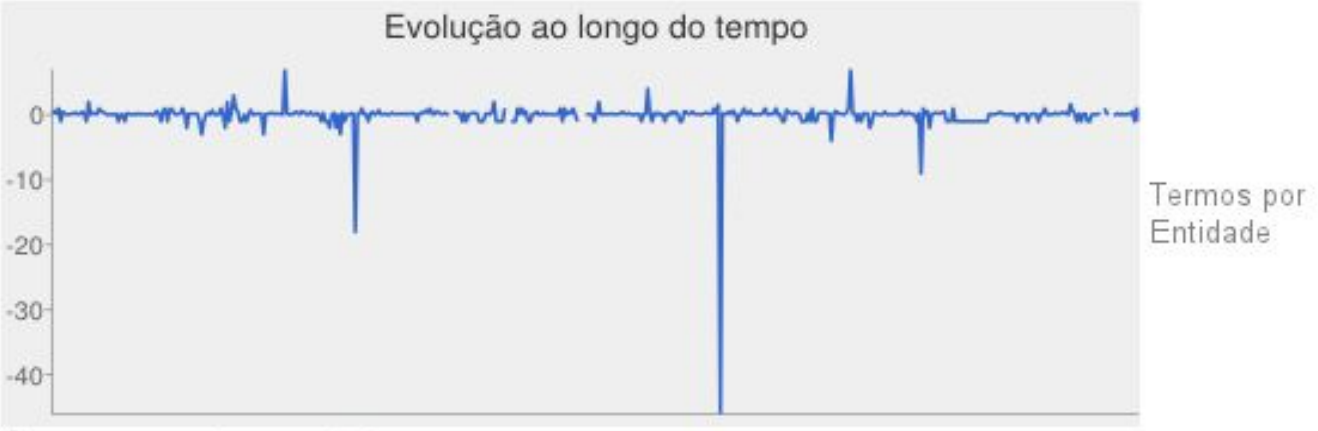


Gráfico 2: Termos por Entidade ao longo do período do caso de teste

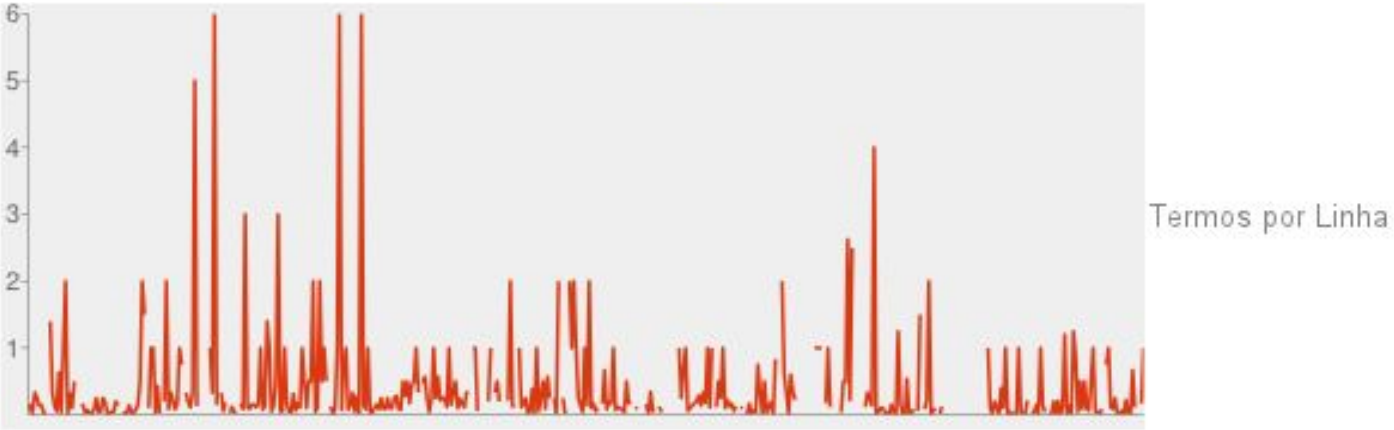


Gráfico 3: Termos por Linha ao longo do período do caso de teste

Alguns outros dados interessantes coletados na extração e que não podem ser visualizados em forma gráfica serão mostrados na Tabela 1, a seguir.

| | Total de Termos | Total de Linhas | Total de Entidades | Termos por Entidade | Termos por Linha | Tempo de Extração (segundos) |
|---|---|---|---|---|---|---|
| Média | 8.464566929 | 463.6708661 | 127.288189 | -0.09692101014 | 0.4197585508 | 9.801574803 |
| Máximo | 309 | 33069 | 3290 | 7 | 6 | 120 |
| Mínimo | 0 | 0 | -1 | -46 | 0.003263308179 | 0 |

O tempo de extração máximo foi de cerca de 2 minutos para todos os commits. Em média o Total de Termos adicionados por commit é 8.4, o Total de Linhas, 463, o Total de Entidades, 127.2.

## CONCLUSÕES

Com base no estudo de caso e na avaliação dos resultados da extração de vocabulário de commits, o vocabulário de commits pode ser descrito para o Total de Termos como diretamente proporcional ao Total de Entidades, com base na alta correlação encontrada nos dois. Esse resultado já era esperado como descrito em (SANTOS, GUERRERO e FIGUEIREDO, 2012), porém validado neste estudo de caso. Também como resultado, foi possível definir que o Total de Termos por Entidade a cada novo commit ocorre de no máximo 7 e o Total de Linhas por Entidade, 6. Sendo possível descrever o comportamento de commits de forma válida.

## AGRADECIMENTOS

O presente trabalho foi realizado com apoio do CNPq, Conselho Nacional de Desenvolvimento Científico e Tecnológico – Brasil
Aos colegas do Laboratório de Práticas de Software (SPLab) do Departamento de Sistemas e Computação, onde este trabalho foi desenvolvido, por todo o apoio e contribuição.

## REFERÊNCIAS BIBLIOGRÁFICAS

LEHMAN, M. ,**Programs, life cycles, and laws of software evolution**. Proceedings of the IEEE. Vol. 68. Issue 9, 1980.

KOKOL, P, PODGORELEC, Vili, ZORMAN, Milan . C**omputer and Natural Language Texts - A Comparison Based on Long-Range Correlations**. Journal of the American Society forInformation Science, John Wiley & Sons, vol. 50, num. 14, pp. 1295-1301, December 1999.

KUHN, Adrian, DUCASSE, Stéphane, GIRBA, Tudor. **Semantic clustering: Identifying topics in source code**. Information and Software Technology, 49(3):230–243, March 2007.

SANTOS, Katyusco de Farias, GUERRERO, Dalton D. S., FIGUEIREDO, Jorge C. A. de Bittencourt, Roberto A. **Towards a Prediction Model for Source Code Vocabulary**. 1st Workshop on the Next Five Years of Text Analysis in Software Maintenace - ICSM2012.