# Impactos de Práticas de Codificação sobre a Legibilidade: Um Estudo Preliminar

**Rodrigo Magalhães dos Santos**[1], **Marco Aurélio Gerosa**[2]

[1]Instituto de Pesquisas Tecnológicas de São Paulo (IPT-SP)
Av Prof. Almeida Prado, 532, Cid. Universitária – Butantã
CEP 05508-901 – São Paulo – SP – Brasil

[2]Northern Arizona University
S San Francisco St, Flagstaff, AZ 86011, EUA

rodrigo.magalhaes@ipt.br, gerosa@ime.usp.br

***Abstract.*** *Although many different coding conventions and standards are available in the software development industry and communities, low code readability levels perceived by developers still represent a barrier. In this paper, it is described a preliminary study that assessed the impact of a set of Java coding practices on the readability perceived by a group of software developers. The survey reveals a varying level of impact produced by each practice and the results can provide some guidance to coding standards creators.*

***Resumo.*** *Embora haja diferentes padrões e convenções de codificação disponíveis na indústria e nas comunidades de desenvolvimento de software, os baixos níveis de legibilidade de código percebidos pelos desenvolvedores ainda representam uma barreira. Neste artigo, é descrito um estudo preliminar que avaliou o impacto de um conjunto de práticas de codificação Java sobre a legibilidade percebida por um grupo de desenvolvedores de software. A pesquisa de opinião revelou níveis variados de impacto produzido por cada prática e os resultados podem auxiliar criadores de padrões de codificação.*

## 1. Introdução

Padrões ou convenções de codificação são comuns na indústria do desenvolvimento de software. Porém, apesar das diferentes convenções disponíveis, diversos relatos de problemas ligados à baixa legibilidade são encontrados na literatura, com impactos significativos sobre os projetos. No âmbito dos projetos de software livre, por exemplo, há evidências de que dificuldades de compreensão do código levam à redução nas contribuições recebidas [Midha et al. 2010] e a atrasos na realização da primeira contribuição de um autor ao projeto [Bird et al. 2007].

Neste artigo, descrevemos um levantamento de opiniões com o objetivo de obter subsídios para uma visão mais detalhada acerca de quais práticas interferem positiva ou negativamente na legibilidade percebida pelos desenvolvedores.

O levantamento de opiniões envolveu dois grupos: alunos de graduação em Ciência da Computação e programadores profissionais de uma grande empresa brasileira de desenvolvimento de software. Um conjunto de 11 práticas de codificação foi derivado

de modelos de legibilidade de código [Buse and Weimer 2010, Scalabrino et al. 2016] e, para cada prática que desejamos avaliar, foi definido um par de trechos de código, no qual um código aderia à prática avaliada e o outro, a violava. Conjuntos aleatórios de pares foram apresentados aos participantes, solicitando a indicação do trecho de código mais legível em cada par.

Procuraremos responder a seguinte questão de pesquisa: *Como as práticas de codificação influenciam a legibilidade percebida pelos leitores do código?*

Os resultados obtidos foram variados: dentre as 11 práticas avaliadas, 7 apresentaram melhora estatisticamente significativa na legibilidade percebida e 1 apresentou piora. As outras 3 não apresentaram efeitos estatisticamente relevantes.

## 2. Trabalhos Relacionados

Pesquisas de opiniões de desenvolvedores acerca de trechos de código já foram realizadas em outras ocasiões [Buse and Weimer 2010, Dorn 2012, Scalabrino et al. 2016], mas com objetivos diferentes do deste trabalho: os trabalhos referidos visavam quantificar relações entre atributos do código e a legibilidade percebida para permitir a construção de modelos gerais para estimativa da legibilidade de código.

Por outro lado, o presente estudo visa quantificar a relação entre a legibilidade e práticas de codificação empregadas, procurando compreender o efeito das práticas em si, e não a construção de modelos.

Os efeitos da aplicação de um padrão sobre um projeto são discutidos em diferentes trabalhos [Boogerd and Moonen 2008, Li and Prasad 2005, Li 2006, Boogerd and Moonen 2009, Smit et al. 2011], mas, até onde a pesquisa bibliográfica realizada pôde alcançar, não há estudos avaliando o efeito de práticas de codificação específicas sobre a legibilidade percebida pelos desenvolvedores.

A importância das práticas de codificação foi evidenciada em trabalhos anteriores [Hellendoorn et al. 2015], indicando que a noção de convenções de codificação emerge naturalmente nos projetos de software livre. Do ponto de vista da presente pesquisa, dentre os resultados de maior interesse obtidos pelos autores, podemos destacar:

1. contribuições contendo código mais aderente às convenções já adotadas no projeto têm maiores chances de serem aceitas;
2. as contribuições fornecidas vão ficando cada vez mais aderentes às convenções estabelecidas à medida em que o desenvolvedor vai ganhando mais experiência com o projeto.

## 3. Método

Nesta seção, apresentaremos dados sobre os participantes convidados e sobre a preparação dos recursos utilizados, bem como o método de análise dos resultados visando responder a questão de pesquisa proposta neste estudo.

### 3.1. Participantes Convidados

Os participantes convidados para a pesquisa foram divididos em dois grupos. O primeiro e mais numeroso é composto de alunos da disciplina de Engenharia de Software do curso de

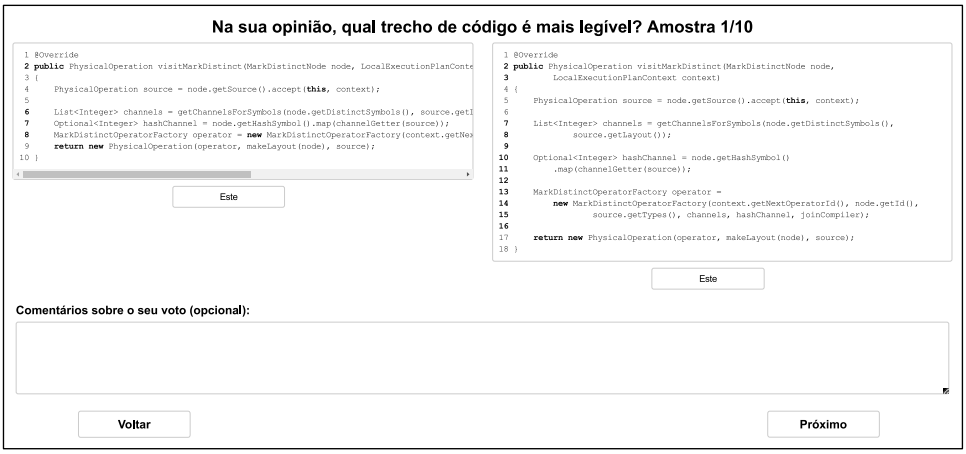


**Figura 1. Exemplo de tela para escolha do trecho de código mais legível.**

Ciência da Computação de uma universidade americana. Esse grupo realizou a pesquisa como parte de atividades extra classe atribuídas.

O segundo e menor grupo é composto de programadores profissionais, funcionários de uma grande companhia de desenvolvimento de software no Brasil. Esse grupo participou da pesquisa mediante um convite realizado a todos os programadores da fábrica de software da empresa.

Ao todo, 62 pessoas participaram do estudo, sendo 55 alunos de graduação e 7 programadores profissionais.

### 3.2. O Processo de Coleta de Opiniões sobre a Legibilidade

Para cada prática de codificação avaliada, foi confeccionado um par de trechos de código – um dos trechos seguindo a prática de codificação avaliada e o outro a violando. Os trechos de código foram apresentados lado a lado aos participantes e, para cada par, solicitou-se que ele selecionasse o trecho de código que considerasse o mais legível. A interface de apresentação dos pares de códigos e coleta das opiniões dos participantes é representada esquematicamente na Figura 1.

### 3.3. Trechos de Código Utilizados na Pesquisa

O levantamento de trechos de código foi realizado utilizando-se o ambiente Google BigQuery, na base de dados de demonstração do GitHub. A consulta levantou nomes de repositórios e caminhos de arquivos e levou em conta os seguintes predicados: a) arquivos com extensão `.java`; b) arquivos contendo cláusulas `import`.

Os arquivos foram organizados em ordem decrescente de quantidade de `imports`, de modo a obtermos classes com um maior número de dependências – essa foi a maneira adotada para evitar classes de código trivial. As classes retornadas foram avaliadas manualmente em busca de trechos úteis para ilustrar as práticas de codificação que desejávamos exercitar.

### 3.4. Práticas de Codificação Avaliadas

As práticas de codificação avaliadas na pesquisa foram derivadas de atributos medidos nos modelos de Buse e Weimer e também de um dos atributos avaliados no modelo de

Scalabrino et al. [Buse and Weimer 2010, Scalabrino et al. 2016].

A seguinte lista contém as práticas de codificação avaliadas nesta pesquisa. Cada prática está identificada com um código, de modo a facilitar referências no decorrer do trabalho:

(P.1) Após colchetes de abertura e fechamento de blocos ("`{`" e "`}`") deverá haver uma linha em branco; **Exceções:** colchetes fechando blocos de código seguidos de cláusulas representando uma continuação do bloco anterior, por exemplo, `elses` e `else ifs` (que representam continuações de `ifs` e `catches` que representam continuações de blocos `try`);

(P.2) Blocos de codificação seguirão a convenção mais usada em Java, isto é, colchete de abertura na mesma linha e colchete de fechamento numa linha individual;

(P.3) Utilizar linhas em branco para separar sequências de instruções relacionadas;

(P.4) Deve-se procurar manter o comprimento das linhas dentro do limite de 80 caracteres;

(P.5) Utilizar tabulações com tamanho de 4 caracteres;

(P.6) Evitar aninhamento de blocos em mais de três níveis;

(P.7) Evitar escrever múltiplas instruções em uma única linha, separadas por "`;`". Procurar descrever cada instrução em sua linha;

(P.8) Evitar, sempre que possível, referenciar classes por seus nomes totalmente qualificados. Usar a cláusula `import` no início do bloco;

(P.9) Caso haja o uso frequente de campos em um nível muito profundo da hierarquia da classe, armazenar a referência ao objeto comum mais alto na hierarquia e realizar os acessos posteriores usando tal referência;

(P.10) Evitar a escrita de longas cadeias de operações lógicas numa mesma instrução: sempre que possível, computar uma expressão lógica em partes, armazenar os resultados das partes em variáveis com nomes alusivos e compor expressões usando os resultados armazenados em tais variáveis, até chegar ao resultado final.

(P.11) Nomes de identificadores deverão fazer uso de palavras em inglês, incluindo variáveis usadas em computações intermediárias e que representem valores não diretamente mapeáveis a conceitos no domínio do problema resolvido pelo programa;

### 3.5. Análise dos Resultados

Cada par de trechos de código exercita uma das práticas de codificação mencionadas. O trecho de código eleito como mais legível pelo participante pode ser o trecho *aderente* ou o trecho *violador* da prática exercitada no par. Neste estudo, deu-se o nome de *voto divergente* aos votos destinados ao trecho de código *violador* da prática exercitada no par do qual ele faz parte.

Com base nos votos dos participantes, cada prática de codificação foi dividida em dois grupos:

1. práticas cujos pares de código tiveram um percentual de votos divergentes *inferior a 50%*, ou seja, neste grupo estão os pares que receberam mais votos no código *aderente* à prática de codificação relacionada ao par;
2. práticas cujos pares receberam *50% ou mais* de votos divergentes, ou seja, pares em que houve um empate ou vitória dos códigos que *violam* a prática de codificação relacionada ao par.

**Tabela 1. Votos recebidos por prática avaliada**

| Prática | Votos no Aderente | Votos no Violador | % de Votos Divergentes | Votos Totais | Prática | Votos no Aderente | Votos no Violador | % de Votos Divergentes | Votos Totais |
|---|---|---|---|---|---|---|---|---|---|
| P.1 | 45 | 7 | 13,46% ▲ | 52 | P.7 | 48 | 10 | 17,24% ▲ | 58 |
| P.2 | 16 | 37 | 69,81% ▼ | 53 | P.8 | 52 | 6 | 10,34% ▲ | 58 |
| P.3 | 28 | 28 | 50,00% ♦ | 56 | P.9 | 45 | 9 | 16,67% ▲ | 54 |
| P.4 | 44 | 15 | 25,42% ▲ | 59 | P.10 | 21 | 34 | 61,82% ▼ | 55 |
| P.5 | 21 | 29 | 58,00% ▼ | 50 | P.11 | 42 | 12 | 22,22% ▲ | 54 |
| P.6 | 34 | 19 | 35,85% ▲ | 53 | | | | | |

Em ambos os grupos foi realizado um teste de proporção unilateral [Barbetta et al. 2004, p. 212] da estatística *proporção de votos divergentes* com o objetivo de verificar o quão representativa é a nossa amostra em relação à população de desenvolvedores.

A hipótese alternativa de cada teste depende do grupo ao qual o par de códigos faz parte:

1. no caso do primeiro grupo: a proporção de votos divergentes é *inferior a 50%*;
2. no segundo: a proporção de votos divergentes é *superior a 50%*;

Os testes fizeram uso da abordagem do *valor-p* e foi empregado o nível de significância $\alpha = 0,05$. Os pares de códigos cujos testes apresentaram valor-p inferior ao nível de significância foram considerados como tendo apresentado efeito sobre a legibilidade.

## 4. Resultados

As opiniões dos participantes revelaram impactos variados das práticas de codificação sobre a legibilidade e, a seguir, os resultados obtidos são apresentados e discutidos.

### 4.1. Quantidades de Votos

A Tabela 1 sumariza os votos recebidos em cada par de trechos de código (cada par de códigos é identificado pelo mesmo código da prática de codificação associada a ele). Ao lado do percentual de votos divergentes há um símbolo indicando a prevalência de votos no trecho de código aderente ou violador referente à prática avaliada: o símbolo ▲ indica que os votos no trecho aderente foram superiores; o símbolo ▼ indica prevalência de votos no violador e o símbolo ♦ indica empate de votos;

### 4.2. Influência das Práticas sobre a Legibilidade

Conforme mencionado na Seção 3.5, o primeiro teste visou avaliar as práticas que supostamente apresentaram melhora na legibilidade. De acordo com o apresentado na Tabela 1, essas práticas são: P.1, P.4, P.6, P.7, P.8, P.9 e P.11. O próximo teste estatístico foi realizado para validar se as demais práticas (P.2, P.3, P.5 e P.10) provocaram a piora na legibilidade percebida pelos participantes. As Tabelas 2 e 3 sumarizam, respectivamente os valores-p obtidos em cada um dos conjuntos de testes.

A lista a seguir apresenta de forma resumida as práticas agrupadas de acordo com os resultados apresentados nos testes de proporção:

- **Melhoraram a legibilidade**: P.1, P.4, P.6, P.7, P.8, P.9 e P.11;
- **Pioraram a legibilidade**: P.2;
- **Não apresentaram diferença tangível**: P.3, P.5 e P.10;

**Tabela 2. Valores-p do teste unilateral de proporção ($H_1 : p < 0.5$ com $\alpha = 0,05$)**

| Prática | Valor-p |
|---|---|
| P.1 | 0,0000 |
| P.4 | 0,0001 |
| P.6 | 0,0272 |
| P.7 | 0,0000 |
| P.8 | 0,0000 |
| P.9 | 0,0000 |
| P.11 | 0,0000 |

**Tabela 3. Valores-p do teste unilateral de proporção ($H_1 : p > 0.5$ com $\alpha = 0,05$)**

| Prática | Valor-p |
|---|---|
| P.2 | **0,0030** |
| P.3 | 0,5000 |
| P.5 | 0,1611 |
| P.10 | 0,0528 |

### 4.3. Impacto das Práticas e os Modelos de Legibilidade

Buse e Weimer ressaltam que seu modelo, embora não possa ser usado para a derivação direta de práticas de codificação, é capaz de destacar características do código que impactam a legibilidade e, assim, merecem maior atenção [Buse and Weimer 2010]. Dada a natureza semelhante do modelo de Scalabrino et al. [Scalabrino et al. 2016], essa mesma característica se aplica a ele[1].

Os resultados obtidos nesta pesquisa condizem com os modelos. De acordo com as opiniões dos participantes, a maioria das práticas promoveu um aumento na legibilidade. Dentre as práticas que apresentaram um percentual de votos divergentes menor que 50% (ou seja, os códigos aderentes foram escolhidos com mais frequência que os códigos violando a prática), todas apresentaram evidência estatística de que a legibilidade foi de fato aprimorada. Dentre as práticas que apresentaram empate ou maioria de votos de divergentes, apenas uma, a P.2, apresentou evidência estatística de piora da legibilidade.

### 4.4. Práticas com Resultados Inconclusivos

Acerca das práticas apresentando resultados inconclusivos, cabe notar que todas elas fazem parte do grupo que com prevalência de votos divergentes. Esse fato indica que as práticas que pioraram a legibilidade não o fizeram com a mesma intensidade do que as que promoveram ganhos. Em outras palavras, as práticas que ganharam mais votos divergentes encontram menos sustentação nas opiniões dos participantes consultados.

Além disso, deve-se considerar também que prescrições de práticas de codificação estão atreladas a diferentes custos: esforços para aprendizado das práticas pelos programadores, esforços de validação pelos revisores, codificação das práticas em ferramentas de validação automática entre outros. O fato de algumas práticas apresentarem resultados inconclusivos indicam baixos ganhos de legibilidade e, assim, poderiam ser eliminadas de padrões de codificação sem maiores prejuízos à sua efetividade.

## 5. Ameaças à Validade

Nesta seção, são descritas ameaças à validade interna e externa, identificadas durante a fase de planejamento e após a análise dos resultados obtidos.

[1]Um exemplo de derivação direta fornecido por Buse e Weimer é o de prescrever a inserção de um número arbitrário de linhas em branco para aumentar a legibilidade, já que o modelo atribui um peso positivo à quantidade de linhas em branco – uma prática de codificação que não faria qualquer sentido.

É importante notar que as práticas de codificação apontadas nesta pesquisa não foram fruto de tal derivação direta, mas sim de uma interpretação dos atributos medidos pelo modelo e seus respectivos pesos.

### 5.1. Validade Interna

O processo de seleção manual dos trechos de código usados para ilustrar as diferentes práticas de codificação pode ter introduzido algum viés nos resultados, já que não foram empregados métodos formais na seleção desses trechos de código. O emprego de ferramentas de análise estática de código poderia atenuar esse problema, já que seus resultados permitiram quantificar o nível de aderência a uma prática de codificação.

Embora não haja indicações de múltiplas participações de um mesmo indivíduo, os mecanismos implementados na aplicação são incapazes de prevenir toda e qualquer tentativa nesse sentido. Assim, deve-se considerar a possibilidade de alguns participantes terem opinado mais de uma vez, algo que distorceria os dados, já que a opinião desses participantes ganharia maior peso e prejudicaria a qualidade de nossa amostra.

### 5.2. Validade Externa

Os participantes da pesquisa pertencem a grupos relativamente restritos – estudantes e programadores profissionais de uma empresa brasileira. Embora eles sejam representativos de grupos de interesse grandes do mercado do desenvolvimento de software, não há uma amostra grande o suficiente para a extensão das conclusões obtidas aqui para grupos mais gerais de programadores.

## 6. Conclusão

Os resultados obtidos possibilitaram responder a questão de pesquisa do estudo, revelando, em 8 das 11 práticas avaliadas, evidências de que as diferenças de formatação influenciaram as opiniões dos participantes acerca da legibilidade. Em 3 das 11 práticas (P.3, P.5 e P.10) não foi possível refutar a hipótese nula e, consequentemente, não se pode afirmar que tais práticas manifestam efeito sobre a legibilidade.

Os casos inconclusivos indicam, como oportunidade de trabalhos futuros, a extensão desta pesquisa a um grupo maior de participantes. Uma reavaliação do caráter opcional de alguns itens da pesquisa pode ser útil, já que identificamos oportunidades de uso para os comentários dos participantes sobre suas escolhas. É necessário considerar, porém, que a obrigatoriedade dos campos pode levar ao aumento da taxa de abandono dos participantes antes do término da pesquisa.

Os dados gerados por esta pesquisa poderão ser usados para a busca de correlações entre características do código e os votos recebidos, o que poderá revelar relações inesperadas entre atributos do código e a preferência dos leitores, fornecendo novas perspectivas sobre os dados já disponíveis.

## Referências


Barbetta, P. A., Reis, M. M., and Bornia, A. C. (2004). *Estatística: para cursos de engenharia e informática*, volume 3. Atlas São Paulo.

Bird, C., Gourley, A., Devanbu, P., Swaminathan, A., and Hsu, G. (2007). Open borders? immigration in open source projects. In *Mining Software Repositories, 2007. ICSE Workshops MSR'07. Fourth International Workshop on*, pages 6–6. IEEE.

Boogerd, C. and Moonen, L. (2008). Assessing the value of coding standards: An empirical study. In *Software Maintenance, 2008. ICSM 2008. IEEE International Conference on*, pages 277–286.

Boogerd, C. and Moonen, L. (2009). Evaluating the relation between coding standard violations and faults within and across software versions. In *Mining Software Repositories, 2009. MSR'09. 6th IEEE International Working Conference on*, pages 41–50. IEEE.

Buse, R. P. and Weimer, W. R. (2010). Learning a metric for code readability. *Software Engineering, IEEE Transactions on*, 36(4):546–558.

Dorn, J. (2012). A general software readability model. *MSC Thesis avaiable from (http://www.cs.virginia.edu/weimer/students/dorn-mcs-paper.pdf)*.

Hellendoorn, V., Devanbu, P., and Bacchelli, A. (2015). Will they like this? evaluating code contributions with language models. In *Mining Software Repositories (MSR), 2015 IEEE/ACM 12th Working Conference on*, pages 157–167.

Li, X. (2006). Using peer review to assess coding standards-a case study. In *Frontiers in education conference, 36th annual*, pages 9–14. IEEE.

Li, X. and Prasad, C. (2005). Effectively teaching coding standards in programming. In *Proceedings of the 6th conference on Information technology education*, pages 239–244. ACM.

Midha, V., Singh, R., Palvia, P., and Kshetri, N. (2010). Improving open source software maintenance. *Journal of Computer Information Systems*, 50(3):81–90.

Scalabrino, S., Linares-Vásquez, M., Poshyvanyk, D., and Oliveto, R. (2016). Improving code readability models with textual features. In *2016 IEEE 24th International Conference on Program Comprehension (ICPC)*, pages 1–10.

Smit, M., Gergel, B., Hoover, H. J., and Stroulia, E. (2011). Maintainability and source code conventions: An analysis of open source projects. *University of Alberta, Department of Computing Science, Tech. Rep. TR11-06*.