

# A Importância da Refatoração de Código no Desenvolvimento de Software

## The Importance of Code Refactoring in Software Development

Raul A. V. Ribeiro, Victor M. Torres, Vinicius L. Poloniato.

Centro Universitário de Anápolis UniEVANGÉLICA

Av. Universitária, 75075-010, Anápolis - GO, Brasil

{raulvagmacker17, viniciuslp07060}@gmail.com, victoorres@icloud.com

**Abstract:** *In this article, it turns out that writing good quality code is a very important requirement for programmers. But being specific, what does this mean? A definition of quality code can vary widely and depend on several factors that are related to software development, and the programming language chosen for such a project is very important in this discussion. Developing code anyway can affect application processing and even confuse another project developer to maintain it. And as a proposal, we will demonstrate the necessary steps to build a quality code, refactoring and naming it correctly.*

**Keywords:** *Code Quality, Software, Developing.*

**Resumo:** *Neste artigo, constata-se que escrever código de boa qualidade é um requisito muito importante para os programadores. Mas sendo específico, o que isto quer dizer? Uma definição de um código de qualidade pode variar bastante e depender de diversos fatores que são relacionados ao desenvolvimento de um software, além de que a linguagem de programação escolhida para tal projeto é importantíssimo nesta discussão. Desenvolver um código de qualquer maneira pode afetar o processamento da aplicação e até mesmo confundir outro colaborador do projeto a fazer manutenções do mesmo. E como proposta do mesmo, demonstraremos os passos necessários para construir um código de qualidade, refatorando e o nomeando corretamente.*

**Palavras-chave:** *Qualidade de Código, Software, Desenvolvimento.*

### 1. Introdução

No cenário atual do desenvolvimento de software é importante adotar um padrão para codificação porque a qualidade do código de um software influencia de modo geral na segurança e na confiabilidade do mesmo. Sendo assim, a alta qualidade na codificação é imprescindível para a maior parte das equipes de desenvolvimento.

No interior de uma indústria de produção de software, há de existir certos padrões de codificação indispensáveis para a redução de falhas de algum programa específico. Isso

acontece porque, para encorajar a ampliação, organizações precisam de algumas ferramentas que auxiliem na codificação, aumentando a legibilidade do código, e conseqüentemente a manutenibilidade. Garantir que o código seja de alta qualidade e se mantenha assim até a sua implantação não é um serviço fácil. Necessita de constantes revisões e atualizações para que estes padrões sejam mantidos em todos os aspectos.

## 2. Referencial Teórico

É comum ouvirmos falar sobre qualidade de produtos, serviços e afins. Também sabemos que não é uma realidade distante quando falamos de desenvolvimento de software, pois usufruirmos de processos e metodologias, testes de aceitação, funcionais e stress a fim de garantir a qualidade e excelência do que foi ou está sendo desenvolvido. Dentro do estudo da engenharia de software se percebem diversos problemas que, enquanto desenvolvedores, devemos procurar evitar, a fim de chegar à um código fonte de qualidade. É preciso observar duplicidade de código, métodos e classes com responsabilidades relevantes, complexidade ciclomática elevada, ausência de testes de unidade e outros pontos (Queiróz, 2017).

Para escrever um bom código, todo o desenvolvedor Java necessariamente deve ter bem claros os conceitos de orientação a objetos(classes, polimorfismo, etc) e da API do Java. Em Software, uma métrica se refere a aplicação de uma medida objetiva para determinação de certas características dos artefatos desenvolvidos. A qualidade de uma arquitetura pode ser medida em parte pela quantificação de seus graus de extensibilidade, reusabilidade e manutenibilidade. Essas qualidades são influenciadas pela interdependência dos artefatos (Martins, 2019).

## 3. Desenvolvimento

Elaborar um código de boa qualidade é uma condição fundamental para os programadores. O conceito claro do que é código de qualidade pode modificar excepcionalmente e necessitar de variados fatores referentes ao desenvolvimento de software, além disso, a linguagem de programação utilizada no projeto é fator importante nessa discussão. Algumas dicas para implementar na padronização:

- **Indentação:** Procedimento de separar os graus de codificação por espaços em branco. No caso da Linguagem JavaScript não é essencial utilizar ponto e vírgula para isolar os métodos, mas como padrão utiliza-se para melhor compreensão do código. Há formataadores de código que fazem isso e outras boas práticas automaticamente, como é por exemplo o Prettier e o ESLint.
- **Facilitar testes booleanos complexos:** Testar as validações necessárias para realizar alguma finalidade ajuda muito na compreensão do código.
- **Reduzir dimensão:** Isso torna o código menos complexo, pois sendo assim, a leitura, a compreensão e a manutenção são bem mais acessíveis e compreendidas.
- **Uso da generalização:** Pressupor o código repetido em várias partes do sistema é uma forma de como pode ser incompreensível caso outro desenvolvedor venha à codificar.

Sendo assim, evitar a duplicação de código e ainda economizar o tamanho do sistema é um dos pilares da programação orientada a objetos.

- **Melhorar desempenho:** Na manutenção é muito mais compreensível mudar em um único lugar e resolver o problema que mudar em diversas outras partes e ainda ter a chance de esquecer algum local.
- **Convenção de nomenclatura:** Como padrão utilizamos o seguinte exemplo de nomenclatura de variáveis: nomeVariavel, e para classes, componentes ou outros arquivos seguimos o padrão CamelCase.

“Não está bagunçado, só está arrumado do meu jeito que eu entendo”, (Tessarini, 2019). Abaixo tem-se um exemplo de código claramente despadrão, onde o nome do método não condiz com o algoritmo, o código não está limpo o bastante para ser explicativo, e onde há falta de refatoração.

```
// Busca todas as avaliações pelo id da série, disciplina e turma
deletar_avaliacoes() {
  this.avaliacaoService
    .findAvaliacoesBySDTId(this.serieDisciplinaTurmaId)
    .subscribe(avalacoes =>{
      if (avalacoes.length){
        avalacoes.forEach((
          ) => {
            this.
              addItem();
          });
        this.
          set_items(avalacoes);
      } else
        this.
          add_item();
    });
}
```

Fonte: Autores.

Uma equipe pode montar sua melhor forma de padronização de código, por exemplo, seu respectivo estilo de indentação. Pode ser definido um determinado padrão de variáveis e componentes na linguagem de programação utilizada pela equipe de desenvolvimento, mas juntamente dessa definição deve ser avaliada a acessibilidade desse recurso pelos integrantes atuais e futuros. Abaixo vemos um exemplo de padronização de código a ser realizado indicando a nomenclatura corretamente, explicativo e refatorado.

```

findAvaliacoes() {
  this.avaliacaoService
    .findAvaliacoesBySerieDisciplinaTurmaId(this.serieDisciplinaTurmaId)
    .subscribe(avaliacoes => {
      if (avaliacoes.length) {
        avaliacoes.forEach(() => {
          this.addItem();
        });
        this.setItems(avaliacoes);
      } else {
        this.addItem();
      }
    });
}
}

```

Fonte: Autores.

“Nem vou mexer porque pode ser que eu não entenda o código dele”. Essa citação é uma das piores frases que podem ser ditas na equipe, principalmente em emergências de um projeto. Para uma equipe conseguir manter a boa continuidade dos projetos, é preciso que não só a maneira de deixar seu código seja padronizada e entendida, mas também sua forma de raciocinar e trabalhar seja igual e compreensível para todos (Tessarini, 2019).

Para a padronização é importante lembrar sobre a utilização do Clean Code. Clean Code é basicamente uma filosofia de desenvolvimento cuja o principal objetivo é aplicar técnicas simples que visam facilitar a escrita e leitura de um código.

#### 4. Conclusão

Certos programadores se consideram mais superiores quando o seu código fica mais complexo e complicado de entender, algo que é totalmente o oposto do que se deve pensar. Se todos seguirem um padrão específico, tudo fica mais compreensível, pois outro leitor ou desenvolvedor/programador consegue ler o código com uma maior facilidade, seja um colega de trabalho ou até mesmo um programador do outro lado do mundo. É evidente que codificar algo ou um método que seja legível visualmente para todos não é uma tarefa fácil e isso requer muita prática, porém isso vem quase inconscientemente com o tempo. Sendo assim, ler bastante

o código de outras pessoas é um tipo infalível de boa prática.



Fonte: SlideShare

## 5. Referências:

Batista, Stephany. **Qualidade de Software: dicas para escrever um código de qualidade.** Disponível em: <<http://www.linhadecodigo.com.br/artigo/3460/qualidade-de-software-dicas-para-escrever-um-codigo-de-qualidade.aspx/>> Acesso em: 26 de novembro de 2019.

Martins, Rodrigo. **Qualidade de Código | Atitude Reflexiva.** Disponível em: <<https://atitudereflexiva.wordpress.com/2009/08/15/qualidade-de-codigo/>> Acesso em: 12 de dezembro de 2019.

Queiróz, Kamilla. **DevQA: Como medir qualidade de código.** Disponível em: <<https://medium.com/@kamillaqueiroz/devqa-como-medir-qualidade-de-c%C3%B3digo-6149fada1e/>> Acesso em: 12 de dezembro de 2019.

Santos, Lucas. **Qualidade além do código – Organização e comunicação.** Disponível em: <<https://imasters.com.br/desenvolvimento/qualidade-alem-codigo-organizacao-e-comunicacao>> Acesso em: 26 de novembro de 2019.

Tessarini, Gabriel. **Padronização no código e na equipe.** Disponível em: <<https://imasters.com.br/carreira-dev/padronizacao-no-codigo-e-na-equipe/>> Acesso em: 12 de dezembro de 2019.

Valente, Wander. **Artigo Engenharia de Software 12 - Ferramentas de Qualidade de Código.** Disponível em: <<https://www.devmedia.com.br/artigo-engenharia-de-software-12-ferramentas-de-qualidade-de-codigo/12567/>> Acesso em: 26 de Novembro de 2019.

Rufino, Kete. **Qualidade de Código.** Disponível em:  
<<https://pt.slideshare.net/ketemartinsrufino/qualidade-de-codigo-78826563>> Acesso em: 16  
de Dezembro de 2019.