

**Artigo escolhido:**

<https://www.devmedia.com.br/fatores-de-qualidade-na-programacao/29780>

## **Fatores de qualidade na programação**

O texto destaca que desenvolvedores vivenciam várias regras de qualidade de software e em vários casos essas regras são ignoradas, porém como basicamente tudo atualmente utiliza ou depende de software é extremamente importante que haja qualidade no seu desenvolvimento.

Entretanto, o conceito de qualidade é algo extremamente abstrato nesse meio, o que pode ser definido como isso em alguns times pode não ser em outros. Em seguida o autor destaca alguns pontos importantes que podem definir a qualidade de um projeto.

A importância da linguagem é um fator que pode definir a qualidade de um software, por mais trivial e simplório que isso pareça. Há o destaque de que quanto mais avançada a linguagem mais dificuldade de processamento a máquina terá, podendo gerar alguma perda de eficiência, há o contraponto dos usuários desejarem cada vez mais complexidade e experiência completas, é interessante moderar desempenho e funcionalidades.

O autor destaca que o principal elemento de qualidade de um software é uma boa documentação, ressalta que pouco importa se técnicas boas e avançadas são utilizadas se a posteriori ninguém entenderá aquilo que foi feito. É pontuado que a melhor forma de se criar uma documentação é através da UML e seus diagramas. Ademais, outra forma de documentar é descrevendo bem os métodos do código.

Outrossim, escrever código legível e reaproveitar esse são duas boas qualidades na construção de um projeto, uma vez que “a garantia de uma leitura mais fácil do programa pode gerar uma série de recompensas para o desenvolvedor, entre as quais se destacam o maior controle sobre a complexidade do código e a grande facilidade de depuração de código”, logo é importante estar atento com fatores simples, como espaçamento e recuo das linhas, bons nomes para declarar variáveis e métodos e padrões gerais da equipe.

Por fim, os dois últimos pontos abordados são os testes e a segurança na programação. Existem vários tipos de teste, a classificação mais genérica dos mesmos é a que divide os testes em dois tipos: testes de caixa-preta e testes de caixa-branca. O primeiro é também chamado de teste funcional, e é utilizado para analisar simplesmente o comportamento do objeto, ignorando sua construção interna. Já o segundo, ao contrário, analisa a estrutura do objeto, e é conhecido como teste estrutural.

Para garantir que o código que está sendo criado é seguro, é interessante seguir algumas técnicas de programação defensiva. É como se fosse uma atitude de desconfiança com relação ao código, para prevenir o que possa vir a dar errado. Basicamente, essa técnica consiste em assumir um maior controle sobre a execução do código, incluindo um maior número de verificações no mesmo.

O único fator de novo para mim foi que quanto mais alto nível de linguagem menor tende a ser seu desempenho para a máquina. Todo o restante do artigo já havia sido visto/estudado em algum momento da minha jornada como estudante e desenvolvedor.

O artigo pontua bons pontos a respeito de qualidade de software, desde questões mais básicas e às vezes ignoradas pelas partes que o desenvolvem, como documentação e a escolha da linguagem até pontuações a respeito de construção de código fonte, como testes, escrever código legível através de recuo correto, espaçamento, bom nome de variáveis, regras de padrão para a equipe, e uma programação mais defensiva utilizando blocos como try/catch ou até mesmo if/else. Portanto, eu entendo que é um bom artigo que ressalta bons pontos a respeito de práticas que ajudam na qualidade de projeto e construção de software.