

## **Notas sobre o capítulo 2 de Clean Code – by Lucas Trevizan**

### **Use nomes que revelem seu propósito**

O capítulo fala sobre a importância de nomes, e isso é óbvio, afinal de contas nós damos nomes para tudo, é assim que reconhecemos as coisas.

Parece óbvio, como de fato é. O nome de uma variável, classe e função tem que descrever com exatidão o que ela faz, se você achar um nome melhor depois de um tempo, então mude, se precisar de um comentário para explicar, então está ruim.

### **Evite informações erradas**

Abreviar nomes não é o melhor caminho a se seguir, o que parece óbvio para você em uma abreviação pode ter um significado dubio para outro. Se eu falar "hp" por exemplo, você pensa na marca de impressoras, no Harry Potter, na hipotenusa... acho que já deu para entender.

Use nomes completos, cuidado ao usar o "list" nos nomes, sabemos que list é algo específico para programadores e cuidado com nomes parecidos. Alguns caracteres podem gerar confusão, um "l" minúsculo se assemelha ao 1 e "o" também minúsculo se assemelha a um 0.

### **Faça distinções significativas**

Se os nomes precisam ser diferentes, também devem ter significados distintos. Resumidamente falando, não use palavras genéricas tipo Data, Info e etc. para complementar uma classe ou método.

Então pensando de novo no conceito de ser um autor de um código, escreva as coisas com o pensamento de que o leitor saiba o que você está dizendo, e, portanto, entenda a diferença.

### **Use nomes pronunciáveis**

Quanto mais abreviada sua variável for, mais um idiota você vai parecer ao pronunciar ela durante uma discussão. Então pense em nomes inteligentes para que surjam conversas inteligentes ao discutir situações que envolvam essa variável, eu programo em português então não me faltam palavras.

### **Use nomes passíveis de busca**

É muito mais fácil você procurar no código um nome que faça sentido do que procurar um número ou uma letra comum como "e". Bob fala que variáveis com uma única letra devem ser usadas apenas em escopo local dentro de métodos pequenos, eu acho que não devem ser usadas nunca.

### **Interfaces e implementações**

Não enfeite interfaces com nomes que dão informações excessivas ou distrativas,

## **Evite o mapeamento mental**

Variáveis de letra única possuem certos padrões que um programador exibido gosta de mostrar, então ele faz um método pequeno com "a", "b", "i", "j" todas guardando pequenos valores que você constantemente precisa lembrar e mapear durante a leitura.

Eu particularmente detesto essa abordagem e como já tive o desprazer de topar com elas em alguns momentos, e eu sempre sinto vontade de gritar "ESCREVA CÓDIGOS QUE OS OUTROS POSSAM ENTENDER!!!!".

## **Nome de classes**

Nomes de classe e objetos precisam ser substantivos (que evidencia a substância, a essência). Evite palavras como Info, Dados etc.

## **Nome de métodos**

Métodos devem ser verbos que descrevam com exatidão o que aquele método faz.

## **Parâmetros lógicos 2.2**

Esses parâmetros explicitam o fato de que a função faz mais de uma coisa, portanto devem ser evitados.

## **Não de uma de espertinho**

As vezes pode parecer divertido colocar um nome de método como uma piada interna, NÃO FAÇA ISSO! Mantenha sempre a clareza no nome, expresse o que você quer de fato expressar, não de margem ao erro.

## **Selecione uma palavra por conceito**

Escolher uma palavra por cada conceito abstrato ajuda a manter a coesão do código, ter métodos com palavras semelhantes como "pegar" "recuperar" e "manter" gera confusão ao ter que decidir o que de fato você precisa utilizar. Da mesma forma acontece com as classes, então mantenha sempre um "dicionário" consistente.

## **Adicione um contexto significativo**

Poucos nomes são significativos por si só, por isso usamos o contexto para deixar mais legível. No livro temos um exemplo grande de código mostrando como melhorar um método que à primeira vista não é tão claro

## **Conclusão**

Escolher bons nomes está diretamente ligado a uma boa habilidade de descrição, e isso é mais uma questão de prática do que técnica. Se precisar renomear variáveis por saber que isso

melhorará a legibilidade do código, faça. Inclusive essa prática eu faço regularmente, sempre que eu leio um nome e penso que ele pode melhorar eu o renomeio.