

Dev. ID	1st term	2nd term	3rd term	4th term	5th term	Most relevant term (and why)
1	identação	padronização de nome de variáveis	-	-	-	padronizacao de nome de variaveis
2	documentação	estrutura	linguagem	-	-	documentação
3	modularização	legibilidade	nome de variáveis/funções	-	-	Modularização
4	sem padrão de nomenclatura (nomes que não são auto-explicativos, uso de acrônimos)	vários comandos numa única linha	falta do uso de indentação das linhas	arquivos com muitas linhas de código	sentenças condicionais muito longas	nomenclatura indentação
5	denso	macarrão	desorganizado	não-linear	-	denso
6	nome de variavel não condiz com o que ela faz	métodos grandes	classes não coesas	-	-	nome de variavel que não condiz com o que ela faz
7	nomes de variáveis e funções	-	-	-	-	funções
8	sem indentação	muito verboso	-	-	-	sem indentação, porque a legibilidade do código fica comprometida, o que dificulta o entendimento.
9	problema	confusão	tempo perdido	irritação	-	Tempo Perdido, porque vai tomar muito tempo para identificar problemas simples
10	descomentado	linhas muito longas	muitos blocos aninhados	-	-	linhas muito longas, pois dificulta a leitura.
11	code smell	falta de padrão	desorganização	-	-	Desorganização. Escolhi essa palavra porque vejo uma grande dificuldade em manter as pessoas dentro do time de desenvolvimento alinhados e seguindo um determinado padrão de codificação.
12	grande	complexo	longo	nomes	identação	longo
13	comprimento	quebra	desordem	erros	complexidade	Quebra, uma vez que, na minha opinião, o que me causa maior confusão no entendimento de um código-fonte são quebras de raciocínio, tais que, para um código X, a sequência lógica de ações Y a serem tomadas é quebrada para a adição de alguma funcionalidade extra ou quando esta é feita de uma maneira "estranha".
14	variables com nomes não informativos	não está comentando as linhas de código	funções grandes	-	-	variables

15	amontoado	longo	demorado	-	-	amontoado - porque a falta de indentação e nomes mal formulados fazem a leitura do código mais lenta
16	identação	variáveis	-	-	-	identação: a falta de indentação dificulta a leitura do código e a identificação das hierarquias entre as funções
17	variáveis não utilizadas	não formatação do código	não padronização do código	-	-	formatação e padronização
18	desmodularizado	sem comentários	nome de variáveis e funções não intuitivos	pouca legibilidade	-	desmodularizado
19	comentários	constantes	nomes curtos	profundidade de loop	recursão	profundidade de loop. Acredito que está relacionado a uma questão mais de organização geral, que acabou não vindo a minha cabeça de primeira.
20	bagunça	desgraça	tristeza	trabalho	tempo	Tempo. Código confuso leva tempo para ser entendido.
21	métodos muito grandes, com muitas linhas	códigos sem indentação	linhas de código muito grandes que precisem ser quebradas	nomes de variáveis egnções com genéricas (i, x)	arquivos muito grandes, com 200 ou 300 linhas	Métodos com muitas linhas. Acho muito confuso pois fica difícil fazer o rastreio para entender a lógica do método.
22	poderia ter tido um maior esforço para elaborar	-	-	-	-	melhorar
23	funções muito grandes	uso de muitos condicionais	funções/objetos desempenhando várias tarefas	pouca documentação em trechos importantes	-	Funções muito grandes, pois me parece que o código foi feito sem a preocupação de que outras pessoas irão lê-lo. Ou mais que a própria pessoa precisará lê-lo no futuro.
24	trechos muito longos	repetição dos trechos	nome das entidades	-	-	longos: não uso de locais
25	desleixo*	complexidade	mudanças requisitos	dificuldade (S)	-	Desleixo, pois na minha opinião escrever a solução para um problema é só o início do trabalho. Você tem que considerar que a sua solução poderá ser usada por anos em muitos casos diferentes, será lida e precisará ser compreendida por muitas pessoas diferentes com uma maior ou menor compreensão do contexto e precisarão evoluir aquela solução. Logo, faz parte do seu trabalho não somente resolver o problema, mas resolvê-lo da forma mais simples e compreensível para facilitar o trabalho dos que vierem a dar manutenção no seu código posteriormente.
26	espaguete	variáveis ilegíveis	código ilegível	linhas densas	-	Linhas densas, porque um código denso faz com que o programador precise se concentrar demais para entender um trecho menor de código e a leitura não foi naturalmente
27	tamanho: funções com muitas linhas de código	nomes: variáveis com nomes estranhos	aninhamento: vários aninhamentos de estruturas de controle	-	-	tamanho Porque a quantidade de linhas dentro de uma mesma função ou classe torna mais difícil entender o objetivo principal da abstração correspondente
28	smell	confusão	desordem	-	-	Code Smell e falta de organização. São as que observo mais utilizada para justificar a falta de entendimento de um código.

29	complexidade ciclomática	muitas estruturas aninhadas	trecho muito grande de código	-	-	complexidade ciclomática.
30	code smells	variáveis genéricas	métodos genéricos	comparação binária	regex	code smells, porque são estruturas de código que normalmente o tornam confuso, ineficiente e problemático
31	grande	aninhado	complexo	-	-	Aninhado. Muitos aninhamentos tendem a ser, para mim, bastante difíceis de ler e compreender (especialmente quando há muitas chamadas de métodos no trecho de código que estou analisando).
32	complexidade	refactoring	métodos longos	classes	code smells	code smell
33	métodos longos	linha longa de código	código muito aninhado	-	-	Código muito aninhado. Dificulta o entendimento do código visto que há uma maior dificuldade em entender o fluxo que o código está seguindo.