Carimbo de data/hora	Pontuação	Name and Student ID	What is the name of the file you evaluated?	How would you rate your confidence that there is a LONG METHOD smell in the code?	Mention why (for instance, which code elements) you gave this rating.
27/04/2025 14:30:49		UFMG #1	NodeTraversor.java	4	NodeTraversor.filter(NodeFilter filter, Node root) pode ser simplificado quebrando em vários métodos pequenos, mas dada a natureza de otimização de caminhamento em árvore (recursividade), embora ganhe em "qualidade" se uma refatoração for feita, a performance pode piorar significantemente ao adicionar mais "processos" a pilha de funcionalidade.
28/04/2025 22:24:07		UFMG #34	CodeCacheStage	4	Method redraw() is long and it has several branches.
28/04/2025 22:26:14		UFMG #34	OrderController	3	Method order is a bit long, but it does not seem to be a big problem.
29/04/2025 12:37:30		UFMG #29	VCardResultParser	5	Método matchVCardPrefixedField(CharSequence prefix, String rawText, boolean trim, boolean parseFieldDivider) é um forte candidato. Muitas linhas de código. Dois loops (separados) dentro de outro loop indicam mais do que uma responsabilidade, também. Consigo imaginar a separação desses loops em 2 métodos menores.
29/04/2025 13:58:59		UFMG #29	MarkdownWebServerPlugin	1	Único método maior é o readSource, mas talvez ele passe a sensação de ser grande por usar try, catch e finally.
06/05/2025 21:13:38		UFMG #4	XxlJobExecutor	3	Apesar de não ser tão longo, initEmbedServer poderia ser considerado Long Method, já que poderia ser quebrado em outros menores.
14/05/2025 17:37:53		UFMG #20	DefaultMQAdminExtImpl	5	Large Class: A classe realiza várias operações administrativas: envio de mensagens, controle de tópicos, ACL, entre outros. Muito acúmulo de funções. Long Method: Métodos como createAndUpdateSubscriptionGroupConfig fazem várias coisas sem divisão. Isso dificulta leitura e testes. Shotgun Surgery: Pequenas mudanças exigem alterações em muitos pontos. Isso fragiliza o código.
14/05/2025 17:44:23		UFMG #33	AndroidMusic	5	O código apresenta uma única classe, com quase duzentas linhas, o que eu considero muito longo para os meus padrões. Além disso, a classe tem um número muito alto de métodos e responsabilidades. Na minha opinião se trata de um caso de Large Class.
14/05/2025 17:45:53		UFMG #3	WebLogAspect	2	the bigger method has almost 40 lines, i would say its medium, and its not doing things that are alien to each other, its enclosed in a scope, its simple all in all ok to me.
14/05/2025 17:51:13		UFMG #5	ComparisonCriteria	4	the method is long and theres easy noticible breaking points, for example the for loop could be extracted in it`s own method.
14/05/2025 17:51:34		UFMG #20	ClassLoaderCommand	3	Long Method: O método main executa diversas ações (parsing de argumentos, execução de comandos, controle de erro) em sequência. Poderia ser dividido em métodos auxiliares para maior clareza, embora não seja excessivamente longo. Large Class: A classe é focada na execução de um comando via classloader. Apesar de centralizar o fluxo, não acumula muitas responsabilidades distintas.
14/05/2025 17:51:47		UFMG #9	CodeCacheStage	5	O método redraw é absurdamente longo.
14/05/2025 17:53:27		UFMG #39	NodeTraversor	5	Há uso excessivo de blocos if-else
14/05/2025 17:54:36		UFMG #6	CodeCacheStage	5	Definitivamente o smell de long method está presente.
14/05/2025 17:55:30		UFMG #37	ArgumentTokenizer	5	Há a presença do code smell Long Method no código, especificamente da linha 82 a 170. Isso porque, dentro do bloco for, há um if/else muito profundo, que contém dois blocos de switch/case e instâncias de if/else dentro de alguns cases, tornando o código difícil de entender e manter.
14/05/2025 17:55:43		UFMG #15	ExcelWriter	1	O código não apresenta métodos muito grandes, o maior deles tem 20 linhas.
14/05/2025 17:56:03		UFMG #23	NodeTraversor	5	Métodos traverse e filter são longos, com múltiplas responsabilidades e difícil leitura.
14/05/2025 17:56:31		UFMG #24	InPacketHandler	3	Metodo muito longo que faz muitas ações que poderiam ser encapsuladas, os ifs poderiam ser funções com um nome claro do que está sendo feito.
14/05/2025 17:57:35		UFMG #33	ComparisonCriteria	5	Esse código tem dois bad smells, tanto o fato da classe ser muito grande, quanto os métodos serem muito grandes, o que gera dificuldade de teste e de compreensão do código.
14/05/2025 17:57:52		UFMG #1	AttributeNameHistoWalker	4	Caso exista de fato um smell de long method, provavelmente está relacionado com a presença de mais de um if case no método visit()
14/05/2025 17:58:02		UFMG #10	CN_QuantifierSegmenter	5	O código apresenta no minimo 3 métodos que podem se enquadrar no badsmell de Long Method. Contendo muitos blocos IF, sendo necessário usar o scroll para ver todo o método, etc.
14/05/2025 17:59:02		UFMG #15	InPacketHandler	4	o método `channelRead0` tem quase 50 linhas de código, com vários `if` e exceções.
14/05/2025 18:00:01		UFMG #1	OrderController	4	O método order() assume muitas funções, este poderia ter sido quebrado em métodos menores para que fosse reduzida sua complexidade, caracterizando um smell de long method
14/05/2025 18:02:05		UFMG #1	CN_QuantifierSegmenter	5	O método processCount() possui diversos if cases aninhados, o que caracteriza um code smell de long method em função da sua complexidade resultante.
14/05/2025 18:03:05		UFMG #2	Faker	5	A classe possui uma quantidade excessiva de responsabilidades. Essa concentração prejudica a manutenibilidade e dificulta os testes.
14/05/2025 18:03:53		UFMG #35	PassphraseEntryController	5	Para mim este código com certeza tem o code smell de "Long method form", principalmente no método initialize, que usa várias linhas para fazer a inicialização individual de ínumeros objetos da classe. Isso poderia ser facilmente evitado criando um método diferente que trata de inicializar cada objeto.
14/05/2025 18:04:12		UFMG #23	ExcelWriter	4	Há uma forte dependência de uma única classe realizando múltiplas tarefas, e muitos métodos estão fazendo coisas muito semelhantes de uma forma ligeiramente variada, o que leva à duplicação de código e à falta de clareza.
14/05/2025 18:08:25		UFMG #16	TestProxies	5	A classe `TestProxies` apresenta um **cheiro de código** no uso de **métodos longos**. Cada método de teste tem muitas verificações, tornando o código difícil de manter. É recomendado dividir os métodos em partes menores, cada uma com uma única responsabilidade, como separar a criação de declarações, preparação de comandos e verificações de `unwrap`. Além disso, a configuração do `HikariDataSource` poderia ser extraída para um método de setup, evitando repetição e tornando o código mais organizado e fácil de manter.
14/05/2025 18:09:37		UFMG #44	DefaultMQAdminExtImpl	5	Com relação a long method, não acredito que exista, no entanto, a classe é extremamente grande, tendo o code smell large class.
14/05/2025 18:11:30		UFMG #7	DefaultMQAdminExtImpl	5	O arquivo tem 1131 linhas, um valor muito alto para um método
14/05/2025 18:11:47		UFMG #10	TestProxies	5	Todos os métodos da classe se enquadram no badsmell Long Method. Todos tem mais de 50 linhas, em todos é necessario o uso do scroll e existe o uso excessivo de blocos try/catch em um mesmo metodo.
14/05/2025 18:12:27		UFMG #36	AndroidMusic	3	Coloquei 3 porque acredito que não há muito o problema de Long Method. Os maiores métodos tem 20 linhas, o que não acredito que seja muito. O problema é que, por exemplo, o método play() tem vários ifs e vários trys, então fiquei em dúvida. Porém, percebi que pode ter o problema de Intensive Coupling, porque a classe AndroidMusic é fortemente acoplada à classe MediaPlayer do Android, já que praticamente toda a sua funcionalidade principal depende diretamente das chamadas a player.somemethod().
14/05/2025 18:13:15		UFMG #27	SaJwtUtil	1	O arquivo não apresenta o code smell Long Method. Os métodos são bem organizados, curtos e cumprem uma única responsabilidade.
14/05/2025 18:13:41		UFMG #41	SaJwtUtil	2	Os métodos GetTimeout e ParseToken talvez poderiam ser mais simplificados, mas não é algo muito problemático.
14/05/2025 18:14:14		UFMG #5	ComparisonCriteria	5	O método arrayEquals implementa checagens de igualdade diferentes, se tornando grande.
14/05/2025 18:14:33		UFMG #44	PassphraseEntryController	5	Existe um método extenso, no entanto, o code smell mais evidente é long parameter list, devido ao método PassphraseEntryController.
14/05/2025 18:15:45		UFMG #7	WebLogAspect	3	O método doAround tem muitas responsabilidades agrupadas em um único bloco de código
14/05/2025 18:16:41		UFMG #39	CardResultParser	4	Métodos com vários blocos if-else, muitas vezes com mais de 50 linhas
14/05/2025 18:17:25		UFMG #7	MarkdownWebServerPlugin	2	Acredito que nao possuí Long Method pois não parecem fazer coisas demais ou ser desnecessariamente e exessivamente longo visto as funcionalidades.
14/05/2025 18:19:31		UFMG #11	Spider	5	O método run() da classe Spider é um Long Method, pois ele possui as três características chaves desse code smell: precisamos scrollar para ver sua implementação completa, possui múltiplos aninhamentos condicionais, e é dotado de vários comentários explicando seu funcionamento ao longo do código, indicando possíveis modularizações daquele método.
14/05/2025 18:20:20		UFMG #28	ExpandedProductParsedResult	2	Por mais que seja uma classe com muitos parâmetros/membros, não acredito que os métodos estejam longos por conta de muitos processamentos sendo feitos no mesmo método, todos ali são do mesmo tipo, exemplo: equals, hashCode, etc. Talvez é necessário rever se há necessidade de tanto parâmetros, porém o uso deles no método não está indevida.
14/05/2025 18:21:57		UFMG #21	SaJwtUtil	4	Sim, a classe SaJwtUtil contém Long Methods, especialmente em parseToken e getTimeout, pois acumulam múltiplas responsabilidades
14/05/2025 18:22:32		UFMG #30	AndroidMusic	4	O código apresenta code smells como duplicação de tratamento de exceções (blocos try-catch repetitivos), checagens redundantes de player == null (violando Fail-Fast), uso de printStackTrace() (em vez de logging adequado), estado exposto desnecessariamente (protected wasPlaying), lógica de preparação espalhada (isPrepared em múltiplos métodos), violação do Single Responsibility em setPan() (gerencia volume e pan), callback sem tratamento de erros em onCompletion(), e magic numbers (divisões por 1000f). Esses problemas comprometem robustez e manutenibilidade.
14/05/2025 18:24:32		UFMG #44	CN_QuantifierSegmenter	5	O método processCount tem mais de 60 linhas com inúmeros IF, FOR e ELSE, caso de Long Method. Além disso é uma classe com quase 200 linhas, uma Large Class.
14/05/2025 18:24:42		UFMG #27	XxlJobExecutor	5	Sim, tem Long Method no start() porque ele faz muitas coisas seguidas num único método grande, dificultando leitura e manutenção.
14/05/2025 18:26:05		UFMG #35	PassphraseEntryController	4	A classe possui mais de 200 linhas, diversos imports e métodos. Creio que seja maior do que o ideal, por isso creio se tratar desse code smell.
14/05/2025 18:27:43		UFMG #40	VCardResultParser	5	O método matchVCardPrefixedField é excessivamente longo, podendo ser modularizado de forma a dividir suas responsabilidades.
14/05/2025 18:27:51		UFMG #25	eval-llms-code-smells /WriteWorkbookHolder	5	O código possui long method, pois podemos perceber que ele executa o a ação de pegar um arquivo , verifica qual o tipo de arquivo Excel e verifica se poderemos escrever nele, além, de lançar exceções quando há algo errado. São muitas funções para um único método.
14/05/2025 18:28:13		UFMG #45	InPacketHandler	4	O método channelRead0 é excessivamente longo, indicando que poderia ser dividido em mais de um método.
14/05/2025 18:30:35		UFMG #16	WebLogAspect	5	Long Method - O método "private Object getParameter(Method method, Object[] args)" possuiu muitas linhas e varias condicionais aninhadas
14/05/2025 18:38:01		UFMG #6	eval-llms-code-smells /OrderController	1	Não acredito que o código tenha o code smell long method, pois os métodos apresentados não são muito longos a ponto de precisarmos usar o scrollbar por exemplo e não necessitam de vários comentários explicativos, sendo fácil de entender.
14/05/2025 18:38:12		UFMG #45	Faker	3	Trata-se de um long class, não long method. O único método excessivamente longo é o construtor, devido aos atributos da classe.
14/05/2025 18:38:55		UFMG #28	MarkdownWebServerPlugin	1	Para mim este código não apresenta o code smell de long method, porque ele faz o processamento necessário que arquivos normalmente precisam em apenas um método, processamento este que parece ser todo interligado, sendo difícil de separar em outros métodos sem perder o entendimento do código. As vezes há código que realmente são feios de implementar, mas o importante é que a parte de leitura e serve do file está bem separada e desacoplada neste caso.
14/05/2025 18:39:47		UFMG #43	TestProxies	4	Apresenta muitos métodos que fazem muitas coisas e são inclusive parecidas como as de testar exceções
14/05/2025 18:40:04		UFMG #8	Faker	5	Por precisar de scrollar para ver seu funcionamento se caracteriza um long method. A função Faker tem como parâmetro de entrada 3 parâmetros, logo é um long parameter list. Também possui inúmeras dependências (dipersed coupling) E a classe possui aproximadamente 700 linhas, portanto é uma large class.
14/05/2025 18:43:41		UFMG #19	ClassLoaderCommand	5	Vários dos métodos da classe são realmente longos e poderiam ser quebrados em outros métodos.
27/05/2025 11:57:07		UFMG #4	ClassReader	5	O método accept é extenso, com lógica muito encadeada, múltiplos loops aninhados, manipulação direta de bytes e responsabilidades misturadas, ferindo os princípios de coesão e legibilidade.

Carimbo de data/hora	Pontuação	Name and Student ID	What is the name of the file you evaluated?	How would you rate your confidence that there is a LONG METHOD smell in the code?	Mention why (for instance, which code elements) you gave this rating.
27/05/2025 12:02:33		UFMG #17	Spider	5	O método run é extenso e contém muitas responsabilidades violando o princípio da responsabilidade única
04/06/2025 19:39:40		UFOP #1	JedisSentinelPool	5	O método JedisSentinelPool possui em torno de 400 linhas de código, se mostrando excessivamente grande, além de realizar diversas funções por todo o escopo do método, como funções para exclusão e diversos operadores if, while, for e afins.
04/06/2025 19:42:28		UFOP #7	ClassReader	5	metodo muito longo, mais de 334 linhas e 7 funções com os mais variados objetivos.
04/06/2025 19:46:28		UFOP #11	RestMethodInfoTest	5	Certamente caracteriza-se um long method já que a classe conta com 1322 linhas de código.
04/06/2025 19:46:41		UFOP #7	ClassLoaderCommand	5	Apresenta bastante code smell, com varios metodos sendo gigantes e tendo multiplas responsabilidades. Um exemplo é o metodo process.
04/06/2025 19:47:49		UFOP #6	XMLConfigBuilder	5	Certos métodos da classe contém inúmeros if/else, try e alguns requerem rolar a barra da página para se ler todo o método.
04/06/2025 19:48:40		UFOP #28	CompareToBuilder	4	O método possui muitas condicionais que podem ser encurtados para somente uma condicional em uma única linha.
04/06/2025 19:48:54		UFOP #12	CompareToBuilder	1	O código não apresenta code smells do tipo Long Method, pois não possui métodos execivamente grandes ou que realizem várias funções. Ele divide bem o código em diversos métodos que são utilizados ao longo do mesmo.
04/06/2025 19:50:21		UFOP #22	JedisSentinelPool.java	5	Pois há métodos longos que necessitam de rolagem para ler, que tem um bloco try catch dentro de um if que por sua vez está dentro de um for. Um exemplo de método assim desse código é o: private HostAndPort initSentinels(Set<HostAndPort> sentinels, final String masterName)
04/06/2025 19:51:54		UFOP #19	AbstractSentinelAspectSupport	4	O método 'extractDefaultFallbackMethod' me pareceu consideravelmente maior que os outros presentes na classe.
04/06/2025 19:53:17		UFOP #28	RestMethodInfoTest	4	O método possui vários parâmetros não utilizados e uma excessiva quantidade de linhas.
04/06/2025 19:53:24		UFOP #6	ExcelWriter	2	Embora hajam métodos contendo if/else e try, não chega a ser um uso excessivo e não tornam as classes grandes a ponto de ser necessário rolar a barra lateral.
04/06/2025 19:53:35		UFOP #14	eval-llms-code-smells /FrameworkField.java	5	O catch não faz nada, o comentário dentro do catch indica o futuro erro
04/06/2025 19:56:16		UFOP #12	JedisSentinelPool	3	Acredito que possa existir sim um code smell do tipo Long Method na classe run(). Acredito que o tamanho possa ser considerado um pouco grande. Ademais, acredito que ela tente realizar varias ações em si.
04/06/2025 19:56:18		UFOP #4	ClassReader	5	Atribui a nota 5, visto que o código mais de 15 loops for e diversas estruturas condicionais. Além disso, também é possível observar for dentro de if, if dentro de for, case switch dentro de for, entre outros. Todos contribuem para más práticas de programação e possíveis bugs.
04/06/2025 20:00:58		UFOP #6	GsonBuilder	2	Os métodos apresentados não possuem um número elevado ou exagerado de if/else, e os inúmeros comentários feitos foram entre um método e outro, e não dentro dos mesmos.
04/06/2025 20:02:38		UFOP #11	QuotedStringTokenizer	5	Se enquadra na descrição do code smells. Exemplo: o método hasMoreTokens() conta com 135 linha de código.
04/06/2025 20:04:22		UFOP #5	ArgumentTokenizer.java	5	Método de token gigantesco, e não possui responsabilidade única. Só nele tem umas 7 responsabilidades diferentes.
04/06/2025 20:05:59		UFOP #7	RedissonBloomFilter	2	TEm um pouco do code smell long method nas classes add e contains por elas possuirem uma logica complexa e manipular muitas coisas, porem é controlado e nao acredito que seja o suficiente.
04/06/2025 20:06:29		UFOP #10	RedissonBloomFilter	4	O código possui alguns métodos longos(que não são a maioria) e que fazem bastante coisa, o que indicaria um smell Long Mehotd
04/06/2025 20:06:45		UFOP #9	AbstractSentinelAspectSupport	5	Com certeza este código possui o code smell Long Method, notasse ao ver o tamanho dele, tendo que arrastar para baixo para ler o código inteiro, tendo comentários explicando seções também caracteriza este code smell.
04/06/2025 20:09:37		UFOP #9	QuotedStringTokenizer	2	Apesar de ser um código muito longo, ele é bem organizado e estruturado. Ao analisar, não parece conter o code smell Long Method, pois o código ser grande somente, não significa que possui o code smell. O código é grande, mas compreensível e robusto.
04/06/2025 20:16:04		UFOP #31	RedissonBloomFilter	5	É necessário usar a barra de rolagem para ler todo o método. Possui vários níveis de aninhamento, blocos profundos de if e try
04/06/2025 20:17:20		UFOP #13	DirectedGraphConnections.java	4	Métodos como o DirectedGraphConnections e incidentEdgeIterator são relativamente grandes (especialmente o primeiro mencionado).
04/06/2025 20:20:41		UFOP #4	XMLConfigBuilder	5	Atribui a nota 4, visto que o código possui diversas estruturas condicionais, além de um bloco try muito extenso.
04/06/2025 20:23:19		UFOP #20	/blob/main/GsonBuilder	1	Ambos os metodos exercem somente uma função
04/06/2025 20:24:05		UFOP #27	ClassReader	5	Praticamente todos os métodos dessa classe possuem mais de 50 linhas de código por função. Acredito que, ao refatorar seria possível observar métodos que podem ser repartidos em outros, isso ajudaria significativamente na legibilidade e entendimento de lógica do código.
04/06/2025 20:24:23		UFOP #8	DirectedGraphConnections	4	Existe o smell de long method principalmente no método ofImmutable(), que utiliza de estruturas condicionais dentro de outras que poderiam ser refatoradas e extraídas em outros métodos.
04/06/2025 20:27:41		UFOP #17	XMLConfigBuilder.java	2	Não há metodos com tamanhos excessivos dentro da classe, não há muitos comentários e nem alinhamentos profundos nos métodos
04/06/2025 20:27:44		UFOP #3	DirectedGraphConnections	3	Metodos longos e complexos, pode dificultar para encontrar bugs e para fazer evoluções futuras
04/06/2025 20:29:07		UFOP #5	CompareToBuilder.java	5	Muitos if encadeados e muito comentário explicando
04/06/2025 20:29:33		UFOP #25	XMLConfigBuilder	5	O código possui o code smell long method, ele faz muitas funções em uma só classe o deixando bem extenso e de difícil compreensão
04/06/2025 20:32:37		UFOP #31	JedisSentinelPool	5	Muita repetição de métodos (JedisSentinelPool), código longo faz coisas demais dificil de entender
04/06/2025 20:38:29		UFOP #25	AbstractSentinelAspectSupport	5	o código aparenta ter o smell, ele os métodos são extensos e tem muitos comentários em determinadas partes.
04/06/2025 20:41:29		UFOP #29	DirectedGraphConnections	5	Acredito que exista long method, pois os metodos addSuccessor e addPredecessor e successors tem um tamanho excessivo
04/06/2025 20:42:27		UFOP #3	CompareToBuilder	3	Os principais problemas são que alguns métodos são muito longos e cheios de if/else para tratar diferentes tipos de dados (parecendo um "switch" gigante), e a classe em si é grande por ter muitas funções parecidas.
04/06/2025 20:44:53		UFOP #3	ArgumentTokenizer	4	O principal problema é o método tokenize, que é muito longo e complexo, usando "switches" um dentro do outro e uma lógica de estados difícil de acompanhar
04/06/2025 20:45:13		UFOP #24	RedissonBloomFilter	2	Pois esse código parece ter métodos pequenos ou que não fazem coisas de mais