

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 LARGE CLASS smell in the code?	Mention why (for instance, which code elements) you gave this rating.
27/04/2025 14:21:12		UFMG #7	RequestLimiter.java	4	Classe RequestLimiter.java trata de uma lista de array baseada em uma janela de tempo, então todos os métodos estão relacionados a esta janela. Contudo, temos métodos específicos para manipular a lista, que poderia estar em uma outra classe, e não sobre a limitação desta lista em específico. Poderia ser quebrada? Sim. Mas compensaria quebrar a classe? Acredito que não, já que o código é muito simples e faz sentido estar lá, não compensaria criar um objeto só pra ter manipulação de lista.
28/04/2025 22:21:59		UFMG #10	ClassWriter	4	This class has too many attributes and several methods. It also has almost 400 loc.
01/05/2025 19:51:34		UFMG #16	ConstrainableInputStream	3	A classe aparenta ter muitas responsabilidades, como gerenciar o buffer de input, lidar com timeout, etc.
03/05/2025 14:04:17		UFMG #20	HazelcastPubSubStore	1	Parece coesa. Implementa publish and subscribe.
05/05/2025 17:06:05		UFMG #43	MigrationRunController	5	Muitas responsabilidades para uma única classe
06/05/2025 18:10:31		UFMG #6	SymbolInfo	4	Era melhor nessa parte do código usar uma lista imutável: private static SymbolInfo[] symbols = PROD_SYMBOLS;
06/05/2025 21:32:08		UFMG #33	ContentType	2	Nem o tamanho é tão grande, nem possui tantos métodos. As constantes parecem necessárias.
06/05/2025 21:46:35		UFMG #33	GeometryViewerRegistry	5	Classe com mais de 200 linhas com métodos complexos e muitos atributos.
06/05/2025 21:49:29		UFMG #33	RequestLimiter	2	Só não consegui analisar se a classe possui mais de um papel. Não possui as demais características para ser code smell.
14/05/2025 17:43:11		UFMG #23	HomeController	4	Código contém constantes duplicadas e lógica duplicada entre métodos rest() e feign().
14/05/2025 17:44:02		UFMG #9	ClassLoaderModel	5	A classe possui muitos métodos: 20 no total.
14/05/2025 17:49:30		UFMG #17	HistoryPageFilter	5	A classe criada possui muitas responsabilidades e funções.
14/05/2025 17:51:08		UFMG #11	EmbedServer	5	Essa classe é definitivamente um bad smell, pois é uma classe muito grande, que possui muitas funções e tratamentos, o que torna difícil a compreensão e a refatoração.
14/05/2025 17:54:18		UFMG #29	AcclException	3	Apesar de não ser uma classe muito grande, tem funções muito redundantes, que poderiam ser melhor definidas, então ela acaba sendo grande para seu propósito.
14/05/2025 17:54:38		UFMG #11	JITDataModel	1	Por mais que o código da classe pareça verboso a princípio e tenhamos muitos métodos, as responsabilidades implementadas são típicas de uma classe que representa um modelo de dados, que naturalmente possui mais responsabilidades mesmo.
14/05/2025 17:54:54		UFMG #3	ConstrainableInputStream	1	the class has only 120 lines, not even 10 method the limit for the code smell in php is 22 its very far from a big class
14/05/2025 17:54:56		UFMG #18	SaFoxUtil	4	Eu classificaria minha confiança em 4 de 5 de que essa classe SaFoxUtil sofre do code smell Large Class, pois ela agrega muitas responsabilidades distintas em um único lugar: geração de strings aleatórias, formatação de datas, manipulação de URLs, buscas em coleções, conversões de tipos e outros utilitários diversos Luzkan San Tuon . Além disso, a classe contém um volume elevado de métodos estáticos heterogêneos sem coesão clara, ultrapassando facilmente o limiar de responsabilidade única recomendado para manutenibilidade Refactoring Guru Code Smells . Segundo práticas reconhecidas, uma Large Class geralmente excede limites de tamanho (por exemplo, mais de 100 métodos ou milhares de linhas de código), o que torna o código difícil de navegar e refatorar Wikipédia Refactoring Guru . A mistura de funcionalidades (como encoding de URLs, regex de busca e formatação de listas) viola o Single Responsibility Principle , e tende a criar acoplamentos acidentais e aumentar o risco de bugs quando qualquer parte precisar de mudança Medium San Tuon . Por esses motivos, tenho alta confiança de que este é um caso clássico de Large Class que se beneficiaria de decomposição em componentes menores, seguindo recomendações de refactoring como as de "Clean Code" e estudos empíricos sobre a co-ocorrência de smells em classes inchadas.
14/05/2025 17:55:10		UFMG #25	eval-llms-code-smells /ClassWriter	3	A classe parece possuir code smell de Large Class, principalmente por possuir muitos atributos (mais de 10) e muitas linhas de código, uma indicação de que a classe está fazendo muitas coisas. Contudo, os seus métodos são coerentes com a função da classe de criar novas classes, ou seja, os métodos estão conforme a funcionalidade da classe, por isso marquei 3. Acredito que definitivamente temos code smells no código, principalmente o Long Method e o Large Class. Para o primeiro, temos alguns métodos de 30+ linhas, com múltiplos blocos condicionais, ifs, por exemplo. Por exemplo, o método Heap.crossOverUp(...) contém uma série de verificações e manipulações de estado para ajustar a posição de um elemento no heap. Outros exemplos incluem MinMaxPriorityQueue.removeAt(...), Heap.tryCrossOverAndBubbleUp(...) e QueueIterator.remove(). Para o segundo, o código tem quase 1000 linhas, com várias sub-classes dentro da classe principal. A classe principal MinMaxPriorityQueue é extensa, gerenciando diversas responsabilidades como a lógica da fila de prioridade dupla, o armazenamento em array, o crescimento da capacidade, a coordenação de duas sub-estruturas de heap (através da classe interna Heap, que por si só também é complexa e grande) e um iterador customizado (QueueIterator). Essa concentração de responsabilidades em classes volumosas aumenta a complexidade e dificulta a manutenção
14/05/2025 17:56:34		UFMG #36	MinMaxPriorityQueue	5	
14/05/2025 17:59:57		UFMG #23	ReplyMessageProcessorTest	5	Classe grande com muitos mocks, dados e métodos auxiliares; mistura lógica de teste e preparação excessiva.
14/05/2025 18:01:27		UFMG #17	ClassLoaderModel	5	ClassLoaderModel possui diversas responsabilidades e funções.
14/05/2025 18:01:33		UFMG #39	ConstrainableInputStream	3	Não me parece uma classe muito grande, apesar de ser subjetivo o tamanho a classe possui um objetivo específico, e o tamanho do código condiz com o que ela necessita de fazer. Um outro code smell que pode ser identificado é o excesso de comentários explicando o código, o que indica que a lógica muito complexa.
14/05/2025 18:02:04		UFMG #11	MigrationRunController	5	Só de vemos o tanto de parâmetros no construtor da classe, percebemos que ele está com muitas responsabilidades, além do simples e necessário. Não faz muito sentido uma única classe lidar com as senhas ao mesmo tempo que lida com a verificação de botões e outras coisas mais.
14/05/2025 18:02:21		UFMG #23	SingleRoomBroadcastOperations	4	Muitos métodos públicos com lógica repetida; classe mistura envio, filtragem e callbacks.
14/05/2025 18:02:48		UFMG #30	TestConnections	5	A classe TestConnections contém mais de 30 métodos de teste e centenas de linhas, concentrando diversos cenários de verificação do HikariCP.
14/05/2025 18:06:02		UFMG #13	EmbedServer	5	A classe é longa, em termos de linhas, mas também possui diferentes responsabilidades, necessitando até de comentários em forma de separadores para delimitar os comportamentos
14/05/2025 18:06:36		UFMG #31	AcessControlLogEntry	4	Ele não possui Large Class porque o construtor apenas inicializa os campos a partir de um Map, mas tem Data Class.
14/05/2025 18:06:43		UFMG #27	ArgumentTokenizer	2	Essa classe tem um tamanho um pouco maior do que o normal, mas suas responsabilidades parecem bem definidas, e a estrutura geral não apresenta os sinais típicos de "Large Class" como responsabilidades excessivas, métodos muito grandes ou dificuldades de entendimento e testes.
14/05/2025 18:07:04		UFMG #24	CodahaleHealthChecker	1	Todas as 3 classes indicadas possuem poucas responsabilidades.
14/05/2025 18:08:36		UFMG #29	RequestLimiter	1	Essa classe não é muito grande, o que torna esse bad smell não aplicável ao meu ver.
14/05/2025 18:09:14		UFMG #42	HomeController	3	Não me pareceu uma classe muito grande. Apesar de ter muitos atributos, tem poucos métodos. Não entendi exatamente o que essa classe faz, mas me deu a impressão que existem muitas strings literais usadas dentro do método "rest", o que pode gerar problemas para outras classes dependentes caso esses endereços sejam alterados de alguma forma.
14/05/2025 18:10:16		UFMG #5	SymbolInfo	4	A classe SymbolInfo tem muitos elementos dentro do objeto symbol, e cada symbol tem muita informação, além dos métodos só serem getters e setters.
14/05/2025 18:11:44		UFMG #8	ScriptProcessorBuilder	4	A classe começa na linha 13 e finaliza na linha 71, possuindo diversas funcionalidades em sua declaração. Logo, se caracteriza uma Large Class.
14/05/2025 18:11:49		UFMG #21	JobTriggerPoolHelper	3	Estou dando nota 3 ,pois: 1) O método addTrigger ficou bem grande, Long Method 2) Ele e o wrapper estático trigger recebem seis parâmetros cada. A gente viu que lista longa de parâmetros é um Long Parameter List: fácil trocar a ordem e ruim de testar. 3)Vários números mágicos (10 timeouts, 1000/2000 de fila, 500 ms de limite) aparecem crus. Se alguém quiser ajustar, precisa garimpar no código. Não é desastroso; dá para entender e manter. Mas esses pontos já nos criam atenção depois da nossa aula sobre bad smells.
14/05/2025 18:12:12		UFMG #15	CodahaleHealthChecker	3	A classe não é muito grande em linhas de código, mas ela estende muitos métodos de outra classe "HealthCheck" . Então eu diria que ela usurpa algumas funções de "HealthCheck"
14/05/2025 18:12:22		UFMG #32	IKArbitrator	3	O código apresenta os code smells Long Method (métodos com muitas responsabilidades, como process e judge) e Feature Envy (acesso frequente a métodos de outras classes, como LexemePath e QuickSortSet). Não há evidência de Large Class, pois a classe possui poucas funções e responsabilidade bem definida.
14/05/2025 18:12:23		UFMG #30	ScriptProcessorBuilder	3	Classe tem múltiplos métodos de construção com lógica repetida, mas ainda é coesa e pequena
14/05/2025 18:12:46		UFMG #34	ClassLoaderModel	5	Minha confiança é 5. A classe ClassLoaderModel é um exemplo de Large Class devido à grande quantidade de atributos distintos que ela possui, como classSet, resources, loadClass, urls, classLoaders, classLoaderStats e outros. Cada um desses atributos representa uma faceta diferente dos dados que podem ser retornados por um comando de classloader, resultando em uma classe com muitas variáveis de instância e seus respectivos métodos getters e setters, tornando-a extensa
14/05/2025 18:13:25		UFMG #34	JobTriggerPoolHelper	4	Minha confiança é 4. A classe JobTriggerPoolHelper pode ser considerada uma Large Class porque acumula diversas responsabilidades: gerencia dois ThreadPoolExecutor distintos (fastTriggerPool e slowTriggerPool), mantém a lógica de contagem de timeouts de jobs (jobTimeoutCountMap e minTim), e ainda expõe uma interface de controle estática (toStart, toStop, trigger). Essa concentração de diferentes funcionalidades em uma única classe a torna extensa e complexa
14/05/2025 18:13:26		UFMG #42	IKArbitrator	4	Não me parece ter code smells de Large Class. Apesar disso, o código está demasiadamente comentado. Isso gera uma quebra de raciocínio para entender o que o código faz.
14/05/2025 18:15:01		UFMG #4	SimpleEchoSocket.java	2	Well , in my opinion there is no "Large class" code smell , since it has few methods and attributes.
14/05/2025 18:15:33		UFMG #7	RandomService	4	O código contém várias funções utilitárias, mas poderia ser mais modular e organizado, seguindo o princípio de "Responsabilidade Única". Algumas funções, como joinParam, poderiam ser mais flexíveis e evitar duplicação de lógica. A segurança pode ser melhorada na manipulação de URLs para prevenir vulnerabilidades. A função getValueByType pode ser otimizada para evitar verificações desnecessárias. Também, nomes de métodos como vagueMatch podem ser mais claros. Refatorar o código traria mais clareza e manutenção facilitada.
14/05/2025 18:15:34		UFMG #39	ClassWriter	2	Boa parte do código está preenchido por comentários, então talvez a classe em si não seja tão extensa, e a implementação dela parece de acordo com seu objetivo. Creio que o code smell seria de excesso de comentários.
14/05/2025 18:16:21		UFMG #29	SymbolInfo	5	Essa classe é muito grande, o que se encaixa nesse bad smell, podendo ser resolvido se subdividisse em classes menores, para facilitar a manutenção de códigos e a depuração.
14/05/2025 18:16:36		UFMG #28	JITDataModel	5	A classe JITDataModel possui diversos métodos que tratam de diferentes responsabilidades, como getEventListCopy e findMetaMember, assim como outros. Isso é uma Large Class.
14/05/2025 18:16:37		UFMG #2	AztecCode	2	A classe não está realizando cálculos complexos nem assumindo muitas responsabilidades, mas se suas funcionalidades se expandirem no futuro, ela poderá se tornar uma classe grande com múltiplas responsabilidades.
14/05/2025 18:16:38		UFMG #10	SaFoxUtil	4	O código contém várias funções utilitárias, mas poderia ser mais modular e organizado, seguindo o princípio de "Responsabilidade Única". Algumas funções, como joinParam, poderiam ser mais flexíveis e evitar duplicação de lógica. A segurança pode ser melhorada na manipulação de URLs para prevenir vulnerabilidades. A função getValueByType pode ser otimizada para evitar verificações desnecessárias. Também, nomes de métodos como vagueMatch podem ser mais claros. Refatorar o código traria mais clareza e manutenção facilitada.
14/05/2025 18:16:52		UFMG #12	AutoLocker	1	The AutoLocker class doesn't have many responsibilities and fields. As seen in the code, the class only has four methods with clear and specific responsibilities each one.
14/05/2025 18:19:21		UFMG #24	MinMaxPriorityQueue	5	A classe tem quase mil linhas, o que já tornou difícil para fazer a leitura e realizar uma avaliação mais detalhada. Além disso, encontrei muitos métodos similares entre si e ainda há declarações de outras classes dentro da classe principalmente que fariam mais sentido serem implementadas em arquivos distintos.
14/05/2025 18:20:00		UFMG #14	GeometryViewerRegistry	5	A função é grande pelo motivo de tentar fazer muitas coisas, portanto existe code smells
14/05/2025 18:20:41		UFMG #42	PreparedStatementHandler	3	A classe não me pareceu ter code smells. Apesar disso, parece que há um acoplamento disperso em alguns métodos e também feature envy, pois chamam outros métodos de outras classes e fazem pouco uso de métodos próprios.
14/05/2025 18:21:00		UFMG #27	ScriptProcessorBuilder	1	Não é uma Large Class. Ele é uma classe pequena, simples, e com responsabilidade clara.
14/05/2025 18:23:08		UFMG #38	SingleRoomBroadcastOperations	2	I don't think it has the Large Class code smell. Although it has 100+ lines, most of its methods are just variations of send and sendEvent
14/05/2025 18:25:04		UFMG #49	DataFormatter	5	A classe possui muitas classes filhas que poderiam ser implementadas em outros arquivos e evitar que o arquivo da classe DataFormatter seja tão grande com mais de 900 linhas. Isso para manutenção se torna algo desafiador.
14/05/2025 18:25:10		UFMG #5	JITDataMode	4	A classe JITDataModel faz muitas coisas: gerencia pacotes, estatísticas, eventos, compiladores, compilations, e ainda realiza buscas e atualizações. Isso quebra o princípio de responsabilidade única (SRP).
14/05/2025 18:25:33		UFMG #42	ReplyMessageProcessorTest	1	O método updateStats(...) é extenso e faz múltiplas tarefas: invocações por reflection, atualiza contadores, interpreta atributos, etc. É uma classe de testes. Não vejo motivos de ter code smell. A lógica do teste requer que sejam feitos processos em etapas para verificar a lógica a ser testada.
14/05/2025 18:29:27		UFMG #40	ResultLoaderMap	4	Essa classe tem muitas responsabilidades
14/05/2025 18:29:50		UFMG #1	GeometryViewerRegistry	5	Além da complexidade de populateFromConfig() indicar definitivamente um smell de long method, o tamanho e a quantidade de métodos da classe parece indicar um smell de long class.
14/05/2025 18:30:37		UFMG #36	AccessControlLogEntry	3	A classe em questao nao se caracteriza como Large Class, pois tem proposito bem claro, não se estende muito, e tem sua responsabilidade de ser um objeto de transferencia de dados bem definido. Porem, fiquei na duvida se podemos identificar um Data Class, porque seu comportamento se limita a agregar e fornecer dados, e não manipula-los so com metodos de get e set.
14/05/2025 18:34:36		UFMG #14	TestConnections	5	Sinceramente nem precisei ler o que a classe faz para identificar, uma passada de olho rápido mostra que essa classe não era para ter 1000 linhas. Talvez ela possa ser separada em classes menores.
14/05/2025 18:36:15		UFMG #28	HistoryPageFilter	5	A classe HistoryPageFilter implementa muitos métodos, tornando-a difícil de entender e manter. O método "setSearchString" é implementado e nem é usado por outros métodos, por exemplo.
14/05/2025 18:38:29		UFMG #40	ReplyMessageProcessorTest	2	Essa classe tem várias responsabilidades, mas todas na mesma linha
14/05/2025 18:39:06		UFMG #22	FakeValuesService.java	5	Classes e métodos gigantescos
14/05/2025 18:39:10		UFMG #15	HttpRequest	5	A classe possui mais de 1000 linhas de código. Mesmo grande parte sendo comentários de tradução, acredito que mesmo retirando ainda seria uma classe muito grande.

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 LARGE CLASS smell in the code?	Mention why (for instance, which code elements) you gave this rating.
14/05/2025 18:40:55		UFMG #40	CN_QuantifierSegmenter	4	A classe acumulada várias responsabilidades, como get's, processCount, processCNnumber, etc.
14/05/2025 18:43:28		UFMG #42	SimpleEchoSocket	1	Não me parece ter code smell algum. A classe é simples como o nome diz. Métodos curtos.
14/05/2025 18:48:39		UFMG #35	SaFoxUtil	5	A classe apresentada possui muitos métodos, tratando-se de uma classe longa.
14/05/2025 18:52:00		UFMG #35	HomeController	2	A classe não é muito longa para ser considerada uma large class, pois possui poucos métodos.
27/05/2025 11:53:45		UFMG #2	HttpRequest	5	Concentra responsabilidades excessivas, violando claramente o princípio da responsabilidade única.
27/05/2025 15:18:26		UFMG #1	ConstrainableInputStream	4	Apesar da classe encapsular logicamente ela limita leituras de um InputStream com restrições de tamanho e tempo, além de conter muitas responsabilidades:
04/06/2025 19:38:10		UFOP #20	JedisClusterTest.java	5	Analisando o arquivo é bem notório o problema de Large Class: muitos métodos são implementadosa em uma única classe
04/06/2025 19:42:28		UFOP #4	RedissonSessionRepository.java	5	Atribui a nota 5, visto que a classe possui mais de 25 métodos, resultando em mais de 400 linhas de código. Além disso, são importadas diversas bibliotecas java e outras dependências.
04/06/2025 19:50:08		UFOP #32	BillboardParticleBatch	5	A classe possui mais de 1000 linhas com múltiplas responsabilidades – como alocação de shaders, renderização, configuração de atributos, controle de textura, construção de malhas, entre outras. Viola o princípio de responsabilidade única.
04/06/2025 19:51:49		UFOP #9	RedissonSessionRepository	5	Com certeza tem o code smell Large Class identificado neste código, pois é possível identificar classes com muitas responsabilidades, métodos, ou campos tornando difícil de entender, testar e manter.
04/06/2025 19:52:37		UFOP #4	LinkedTreeMap	5	Atribui a nota 5, visto que a classe é responsável por diversos métodos, como remover um nó, encontrar, criar, substituir, rebalancear a árvore, entre outros. No total, são mais de 600 linhas de código.
04/06/2025 19:57:13		UFOP #11	LinkedTreeMap	5	Apresenta uma quantidade excessiva de métodos.
04/06/2025 19:58:16		UFOP #21	JedisClusterTest	4	Acredito que o código está classificado de forma satisfatória como Large Class por ser uma classe que faz muita coisa em um só lugar. Porém há alguns indícios de que o código está parecido com o Shotgun Surgery, pois existem várias funções que fazem algo parecido, mas de várias formas diferentes, dificultando a manutenção.
04/06/2025 19:58:40		UFOP #1	CmsSubjectCategoryExample	5	A classe CmsSubjectCategoryExample possui diversas responsabilidades vinculadas a ela, o que comprova o smell de Large Class, de forma que isso torna a classe extremamente sensível a mudanças e que haja a necessidade de refatoração para que a mesma não carregue tantos métodos e outras demais responsabilidades.
04/06/2025 19:58:44		UFOP #27	RedissonSessionRepository	5	A classe de repository tem a classe da entidade base dentro dela, o que torna a classe geral mais longa e fora da resposabilidade de um repository apenas. Métodos como createSession() poderiam estar em um service, por exemplo.
04/06/2025 19:58:45		UFOP #14	eval-llms-code-smells / CompareObjectsSettings.ja	1	o código está bem estruturado
04/06/2025 19:59:34		UFOP #3	FakeValuesService	4	- código parece ter muitas tarefas em um só lugar, tornando grande e confuso - tem alguns métodos longos que fazem várias coisas ao mesmo tempo e se eu precisar mudar algo terei que mexer em muitos pontos - a forma como ele lida com outras partes do sistema e usa muitos parâmetros em seus métodos indica que pode ser difícil de manter.
04/06/2025 20:07:58		UFOP #17	Dictionary.java	3	A classe Dictionary tem muitos métodos, mas ao mesmo tempo são necessários e de fácil manutenção
04/06/2025 20:11:44		UFOP #6	ResultLoaderMap	4	A classe apresentada contém um grande número de responsabilidades apresentadas, de forma que deixa o código extenso para ler e difícil de interpretar.
04/06/2025 20:13:14		UFOP #26	SingleRoomBroadcastOperations	1	Acredito que não seja um caso de Large Class pois os métodos criados tem o mesmo sentido de finalidade, não fugindo de sua responsabilidade, e a classe também não é muito grande.
04/06/2025 20:15:28		UFOP #33	CmsSubjectCategoryExample.java	5	A classe GeneratedCriteria possui mais de 80 métodos presentes dentro dela. Eles poderiam ser separados e sendo aplicados em uma ou mais classes diferentes
04/06/2025 20:16:32		UFOP #5	RestAdapter.java	2	As classes são grandes, mas lidam com funcionalidades do seu contexto. Apenas tem linhas demais.
04/06/2025 20:16:48		UFOP #10	ParentRunner	5	A classe ParentRunner é demasiada extensa e possui muitas responsabilidades e métodos, muitas vezes distintos, o que indica um smell Large Class
04/06/2025 20:17:53		UFOP #26	ParentRunner	5	Se trata de um caso de Large Class pois a classe contem muitas responsabilidades como por exemplo: getDescription,run,filter,sort,order, runLeaf,getName,getTestClass. Muitos dos métodos fariam sentido estar em outras classes, para melhor organização.
04/06/2025 20:19:53		UFOP #32	CmsSubjectCategoryExample	4	A classe contém diversas responsabilidades agrupadas em uma estrutura única
04/06/2025 20:21:12		UFOP #16	JedisClusterTest	5	A classe é muito extensa e possui diversos métodos. Os métodos aparentam serem necessários, mas pelo tamanho da classe acredito ser um code smell. O código tem vários métodos de testes como testInvalidStartNodeNotAdded e testJedisClusterTimeoutWithConfig, no entanto possui alguns métodos no final que podem não serem tão necessário ali, como o georadiusStoreBinary.
04/06/2025 20:24:17		UFOP #28	RestAdapter	2	Por mais que o método tenha várias funções com funcionalidades diferentes, ainda sim, as funções fazem parte de um mesmo escopo.
04/06/2025 20:24:47		UFOP #13	ParentRunner.java	5	A classe possui muitos métodos e muitas linhas.
04/06/2025 20:24:58		UFOP #25	LinkedTreeMap	4	algumas das classes são bem extensas, são elas, a LinkedTreeMap e Node
04/06/2025 20:26:50		UFOP #30	ResultLoaderMap	4	A classe ResultLoaderMap possui uma quantidade consideravelmente grande de métodos.
04/06/2025 20:28:59		UFOP #30	FakeValuesService	5	A classe FakeValuesService possui uma quantidade muito grande de métodos.
04/06/2025 20:29:37		UFOP #31	JedisClusterTest	5	long method e Large class (muitos métodos e atributos)
04/06/2025 20:30:44		UFOP #18	BillboardParticleBatch.java	4	Metodos utilizam conjunto repetidos de dado na classe BillboardParticleBatch
04/06/2025 20:36:09		UFOP #24	RestAdapter	3	Me parece que as classes tem muitas responsabilidades
04/06/2025 20:38:46		UFOP #3	DataFormatter	4	O principal é que ela é muito grande, e algumas partes dela, tipo a função createDateFormat, são enormes e cheias de lógica difícil, pode ser resolvido com uma boa organizada, para se tornar mais fácil de ler.
04/06/2025 20:42:15		UFOP #24	RedissonSessionRepository	3	Essa classe me parece ter muitos métodos
04/06/2025 20:50:19		UFOP #29	ResultLoaderMap	5	Acredito que o tamanhos das classes excedem um tamanho recomendável entao a presença de large class