

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 FEATURE ENVY smell in the code?	Mention why (for instance, which code elements) you gave this rating.
01/05/2025 19:40:53	UFMG #17	SentinelDubboConsumerFilter		4	Muitos métodos simplesmente passam parâmetros para um objeto da classe DubboUtils, algo que poderia, então, ser feito por essa mesma classe.
01/05/2025 19:54:08	UFMG #17	ReedSolomonDecoder		4	As funções da classe referenciam múltiplas vezes as funcionalidades da classe GenericGFPoly, o que indica que parte dessa lógica poderia ser transferida para esta outra classe.
03/05/2025 14:06:35	UFMG #27	SentinelDubboConsumerFilter		4	Seus métodos estão excessivamente envolvidos com detalhes de outras classes (principalmente do Sentinel)
06/05/2025 21:29:58	UFMG #12	PrefixPluginLogger		1	Não vi "envy" desta classe com a classe Marker, sendo que this.marker.getName aparece uma única vez.
14/05/2025 17:34:50	UFMG #39	BaseSwaggerConfig		5	A classe BaseSwaggerConfig não tem nenhum objeto próprio e só modifica e instancia os objetos de outras classes dentro de todos os seus métodos.
14/05/2025 17:37:08	UFMG #1	JobGroupController		5	Métodos Longos.
14/05/2025 17:41:17	UFMG #9	SimpleEchoSocket		2	Apesar de chamar alguns métodos da classe CountDownLatch, não é algo excessivo que pode ser considerado um Feature Envy.
14/05/2025 17:43:09	UFMG #3	Code		1	There's no other classes involving that context to begin with, looks like a closed scope for me, the class it's following the single responsibility principal.
14/05/2025 17:44:41	UFMG #32	AwtCodec		5	Dispersed Coupling. O código depende de muitas outras classes
14/05/2025 17:45:42	UFMG #15	ReedSolomonDecoder		1	O código implementa uma espécie de algoritmo de criptografia/detecção de erros, então eu creio que o código deve ter uma complexidade maior e também métodos um pouco maiores. Ao meu ver a classe não faz chamadas excessivas a outras classes, uma vez que as chamadas são para métodos matemáticos essenciais para o algoritmo.
14/05/2025 17:45:53	UFMG #4	ClientHandler.java		4	I think that the finally scope calls the function safeClose too many times . Therefore , this could be encapsulated within a void function , because we could debug errors easier . In addition to it , I think that the code might fit into the "feature envy", because the method close for example is pulling data out of another object to process it.
14/05/2025 17:46:46	UFMG #23	AutoLocker		3	Uso de TODO não resolvido e possível acesso fora da FX Thread. A classe também acessa muitos detalhes internos de Vault. (sem querer responder esse forms em outro, esse é o certo)
14/05/2025 17:47:00	UFMG #34	HazelcastPubSubStore		4	Eu dei nota 4 , pois ao analisar o Método subscribe verifiquei que ele concentra por volta de 28 linhas e 3 responsabilidades: 1.Obter/registrar topic no Hazelcast; 2.Criar listener anônimo com lógica de filtragem por nodeId; 3.Gerenciar cache local (ConcurrentMap) com um bloco de "double-check" manual.
14/05/2025 17:51:13	UFMG #13	NacosWatch.java		5	Pelo que entendi a profundidade de aninhamento chega a três níveis (método -> Listener->if) que analisando os slides é um sinal de Long Method.
14/05/2025 17:52:31	UFMG #7	MigrationRunController		5	Dessa maneira decidi dar nota 4, pois o tamanho da classe não chega a ser enorme.
14/05/2025 17:54:17	UFMG #23	FastByteArrayOutputStream		2	O código utiliza muito o método compare and set da propriedade running da classe AtomicBoolean, isso torna o código verboso e ilegível
14/05/2025 17:55:33	UFMG #14	CachingExecutor		5	Porque o método migrate() é longo demais e trata várias responsabilidades (como lógica de migração, manipulação de UI e tratamento de erros, tudo junto). Acaba dificultando a leitura e manutenção
14/05/2025 17:57:22	UFMG #24	CachingExecutor		2	Código claro e objetivo, mas há leve repetição na conversão de charset para string.
14/05/2025 17:58:42	UFMG #3	SaFoxUtil		5	Este código possui evidentemente o Feature Envy, todos os métodos e classes na maioria do tempo estão usando métodos de ou objetos de outras classes que foram importados.
14/05/2025 18:00:12	UFMG #24	CodeCacheEventWalker		5	Na minha análise não consegui achar bad smells muito claros , mas verifiquei que apenas um método (query com CacheKey) apresenta ramificações internas que podem confundir a leitura, por isso a nota 2.
14/05/2025 18:00:30	UFMG #33	JedisClusterCommand		4	the class is huge, theres a bunch different things being done, from url encoding, json handling, description ...
14/05/2025 18:00:38	UFMG #5	FastByteArrayOutputStream		4	É um código muito longo com espaçamentos desnecessários, e com muitos desvios condicionais alinhados
14/05/2025 18:01:05	UFMG #20	DokanyVolume		5	Única classe com mais de duzentas linhas, com muitos métodos, caso de Large Class. Método dessa classe chamado handleConnectionProblem tem muitos comentários explicando suas seções, caso de Long Method. Método runWithRetries com mais de 50 linhas, também Long Method.
14/05/2025 18:02:05	UFMG #40	FrameworkField		1	A classe FastByteArrayOutputStream só implementa métodos para o seu objeto FastByteBuffer buffer, sem modificar nenhuma outra ou usar objetos de outras, embora dê overrides em métodos da biblioteca Utils.
14/05/2025 18:02:22	UFMG #15	InvokeCountOperation		5	No envio anterior não contei exatamente este code smell portanto estou reenviando.
14/05/2025 18:03:39	UFMG #20	FxApplication		4	No envio anterior não contei exatamente este code smell portanto estou reenviando.
14/05/2025 18:03:39	UFMG #10	JsonAdapterAnnotationTypeAdapterFactory		1	A classe parece estar bem estruturada. O code smell Feature Envy é possível de ser identificado, mas também pode ser justificado pois se trata de uma classe Framework, que atua como "intermediário" e manipula apenas um tipo de objeto. Portanto, é preciso de mais contexto para determinar se é um code smell de fato. Além disso, a utilização de um catch silencioso pode ser problemático; o ideal seria pelo menos loggar sua ocorrência.
14/05/2025 18:06:16	UFMG #32	ReedSolomonDecoder		5	Praticamente todo método do AutoLocker não somente acessa os dados de Vault por meio de métodos getters, como também realiza operações voltadas para o funcionamento interno de Vault.
14/05/2025 18:06:48	UFMG #7	WebLogAspect		4	A classe do exemplo possui métodos que sempre referenciam a classe TypeAdapter muito mais que métodos próprios.
14/05/2025 18:07:01	UFMG #41	AutoLocker		1	Esse código até possui feature envy, mas na prática ele só depende de outra classe pois seu objetivo é manipular os dados de outra classe.
14/05/2025 18:07:44	UFMG #8	NacosWatch		5	Além disso, o código possui problema de long method, que é mais claro, dado o exemplo do método imei.
14/05/2025 18:08:06	UFMG #18	CachingExecutor		1	A classe accessa muitos detalhes internos de XxlJobGroup, como validações e manipulações de atributos. Assim, ela demonstra inveja de funcionalidades que deveriam estar encapsuladas na própria entidade.
14/05/2025 18:09:30	UFMG #11	AutoLocker		4	Além disso, o código possui problema de long method, que é mais claro, dado o exemplo do método imei.
14/05/2025 18:09:41	UFMG #35	JsonAdapterAnnotationTypeAdapterFactory		2	A classe accessa muitos detalhes internos de XxlJobGroup, como validações e manipulações de atributos. Assim, ela demonstra inveja de funcionalidades que deveriam estar encapsuladas na própria entidade.
14/05/2025 18:09:51	UFMG #32	Code		4	Além disso, o código possui problema de long method, que é mais claro, dado o exemplo do método imei.
14/05/2025 18:09:57	UFMG #2	JobGroupController		4	A classe accessa muitos detalhes internos de XxlJobGroup, como validações e manipulações de atributos. Assim, ela demonstra inveja de funcionalidades que deveriam estar encapsuladas na própria entidade.

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 FEATURE ENVY smell in the code?	Mention why (for instance, which code elements) you gave this rating.
14/05/2025 18:09:59	UFMG #25	CodeCacheEventWalker		5	O código da classe CodeCacheEventWalker apresenta o cheiro de código chamado Feature Envy. Esse cheiro ocorre quando um método parece ser mais interessado em manipular os dados de um objeto que não pertence à sua classe. No caso, o método visit contém lógica de manipulação e verificação de compilação e endereços nativos, que são de responsabilidade da classe Compilation e não da classe CodeCacheEventWalker.
14/05/2025 18:10:52	UFMG #33	DuplicateRemovedScheduler		3	Para melhorar o código, seria interessante mover parte dessa lógica para métodos dentro da classe Compilation ou outra classe específica para manipulação de endereços, tornando o código mais coeso e alinhado à responsabilidade única de cada classe.
14/05/2025 18:11:10	UFMG #34	CachingExecutor		3	Acho que o método pushWhenNoDuplicate, por não fazer nada (código vazio) apresenta um caso de RefusedBequest. Além disso, o método push faz um override e usa métodos de outra classe, que pode ser um Feature Envy.
14/05/2025 18:11:32	UFMG #31	ClientHandler		4	Minha confiança é 3. O método query da classe CachingExecutor exibe Feature Envy principalmente em relação ao objeto MappedStatement (ms) e ao TransactionalCacheManager (tcm). Ele acessa repetidamente ms para obter informações de cache (ms.getCache(), ms.isUseCache()) e o BoundSql, e também manipula diretamente o tcm para obter e colocar objetos no cache. Essas operações demonstram um interesse maior nos detalhes internos dessas outras classes do que na própria lógica de execução que delega
14/05/2025 18:11:36	UFMG #21	BaseSwaggerConfig		4	Na minha visão não há uso descabido de "features" de outras classes, porém com certeza o uso dos métodos estáticos da class NanoHTTPD poderia ser muito melhorado. O próprio método close só redireciona a chamada, e a chamada ao logger da classe NanoHTTPD não deveria ser utilizado deste jeito, a própria classe NanoHTTPD deve gerenciar seus logs.
14/05/2025 18:14:22	UFMG #7	SentinelDubboConsumerFilter		4	Ocorrem 5 chamadas de getter seguidas no objeto swaggerProperties. Toda sua lógica depende exclusivamente dos dados dessa outra classe.
14/05/2025 18:14:27	UFMG #22	IKArbitrator		3	Porque depende demais da classe DubboUtils para obter nomes de métodos e interfaces, em vez de fazer isso por conta própria.
14/05/2025 18:14:59	UFMG #16	NacosWatch		2	A classe está fortemente acoplada a e manipula dados de outras classes como AnalyzeContext, LexemePath e QuickSortSet de maneiras que sugerem que as responsabilidades, acredito eu que poderiam ser delegadas ou encapsuladas em outro lugar. Eu dei nota 3, pois pode ser uma decisão de design intencional.
14/05/2025 18:16:12	UFMG #21	URLConnectionClient		5	Ao meu ver, o código não inveja outra classe, pois há pouquíssimas chamadas de métodos de objetos de outras classes no código.
14/05/2025 18:19:44	UFMG #21	InvokeCountOperation		5	Sim, pois, repetitivamente, métodos de request são chamados
14/05/2025 18:20:04	UFMG #36	FrameworkField		2	Sim, pois o método count(String className, BytecodeInstruction instruction) acessa vários dados e comportamentos do objeto instruction
14/05/2025 18:20:05	UFMG #26	ExpandedProductParsedResult		5	Mesmo com vários métodos da classe FrameworkField usarem métodos de Field, com field., descobri que a classe em questão é um "wrapper", então a utilização desses métodos não configura bad smell, e sim exatamente o trabalho de muitos métodos em um wrapper. O FrameworkField não está tentando "roubar" a funcionalidade do Fiel, ele está expondo essa funcionalidade de uma maneira controlada e adaptada para o JUnit.
14/05/2025 18:22:36	UFMG #35	AndroidMusic		5	Utiliza de forma demasiada um método passado como parametro para um método da classe. Existe também o long parameter list, já que alguns métodos possuem muitos parâmetros,
14/05/2025 18:23:08	UFMG #19	AwtCodec		4	Usa vários métodos de outra classe
14/05/2025 18:24:00	UFMG #41	JedisClusterCommand		3	A maioria dos métodos faz uso pesado dos métodos da classe DefaultJSONParser, o que pode indicar Feature Envy e o ponto da classe AwtCodec parece ser integrar as classes Point, Rectangle, Color e Font com estes métodos. Poderia ser melhor fazer com que o parser fosse um método de cada uma dessas classes separadamente ao invés de criar uma nova classe para isso.
14/05/2025 18:24:47	UFMG #6	DuplicateRemovedScheduler		5	3 A classe utiliza de vários métodos, entretanto não sei se eles são importantes o suficiente para ela para estarem constantemente sendo utilizados ou não.
14/05/2025 18:25:17	UFMG #8	Code		5	5 A função push faz uso excessivo da classe logger, indicando que esta não deveria estar ali.
14/05/2025 18:26:29	UFMG #28	VCardResultParser		5	Na minha opinião este código da classe Code está muito acoplado com os métodos da classe Faker, o que além de poder trazer dores de cabeça na hora de fazer uma manutenção em ambas as classes, faz com que o código dependa muito desta outra classe. Com certeza poderia ser refatorada para tratar diferente.
14/05/2025 18:26:38	UFMG #41	AsserUtil		1	Nesse código muitos métodos de outras classes são usados, como equalsIgnoreCas, getSchemeSpecificPart e outros.
14/05/2025 18:27:13	UFMG #38	JobGroupController		1	A função não utiliza de vários métodos auxiliares, na verdade não vi utilizando métodos de outras funções então não acredito ter este problema.
14/05/2025 18:36:13	UFMG #40	IKArbitrator		4	4 This class makes a lot of pulling out of XxlJobGroup to make processing. This is very clear in the save method
14/05/2025 18:37:35	UFMG #31	URLConnectionClient		3	A classe possui diversas chamadas de métodos de outras entidades, atuando quase como service que realiza alguma ação. A documentação presente no arquivo não contribui ativamente para identificar suas responsabilidades, portanto a definição de code smells não foi tão assertiva quanto poderia.
14/05/2025 18:38:04	UFMG #36	BaseSwaggerConfig		3	3 Os métodos não dependem muito de outros objetos
14/05/2025 18:38:23	UFMG #22	JedisClusterCommand.java		5	Percebi um certo nível de Feature envy da classe Base em relação a Swagger. Ela realiza várias chamadas a métodos GET de Swagger para extrair dados e usa-los para construir um objeto. Tem uma forte concentração de chamadas para Swagger dentro de apilInfo, o que sugere que a lógica para transformar SwaggerProperties em ApilInfo está intimamente ligada aos dados de SwaggerProperties.
14/05/2025 18:42:49	UFMG #43	URLConnectionClient		4	4 A função não utiliza de vários métodos auxiliares, na verdade não vi utilizando métodos de outras funções então não acredito ter este problema.
14/05/2025 18:46:01	UFMG #36	SocketIOChannelInitializer.java		5	5 Há vários parâmetros em muitas funções, logo várias funções estão sendo herdados também
04/06/2025 19:41:28	UFOP #14	eval-lms-code-smells /AllMembersSupplier.java		5	5 large class faz muita coisa, ela lida com campos, arrays, gerações de parâmetros .
04/06/2025 19:47:45	UFOP #13	HikariDataSource.java		5	5 A classe HikariDataSource frequentemente chama métodos da função HikariPool.
04/06/2025 19:47:53	UFOP #15	QueryParser		1	1 Um método em uma classe não usa métodos ou dados de outra classe em excesso.
04/06/2025 19:49:46	UFOP #7	DuplicateRemovedScheduler		1	1 alguns métodos da classe fazem uso da classe logger mas não o suficiente para ser considerado um code smell.
04/06/2025 19:54:51	UFOP #10	TextureAtlas		3	3 Nesse trecho de código alguns métodos chamam mais outras classes do que a classe do mesmo, o que indicaria que talvez esse método deveria estar em outro lugar
04/06/2025 19:55:00	UFOP #15	ConnectionPool		2	2 Nesse caso alguns métodos em uma classe usam métodos ou dados de outra classe, mas nada considerado em excesso, no meu ver está dentro do aceitável.
04/06/2025 19:56:42	UFOP #23	SaOAuth2Template		1	1 Código muito longo
04/06/2025 19:56:43	UFOP #22	HikariDataSource.java		4	4 Acho que sim pois o método HikariDataSource usa mais métodos/dados de outra classe (HikariConfig) do que os da sua própria classe.
04/06/2025 19:58:10	UFOP #20	ClassicPluginStrategy		4	4 Acredito que seja o caso de feature envy pelo elevado número de importações e também um alto uso de chamadas de métodos pertencentes a outras classes.
04/06/2025 19:58:29	UFOP #33	SaOAuth2Template.java		3	3 No método CodeModel há um objeto de mesmo tipo que depende das várias presentes no objeto tipo RequestAuthModel criada na mesma class. Ele utiliza diversos das variáveis presentes no objeto, sendo assim um método que requisita vários atributos de outro objeto.
04/06/2025 20:02:06	UFOP #15	AbstractWriteHolder		2	2 Alguns métodos em classes usam métodos ou dados de outra classe, mas não ao ponto de ser considerado um "Feature Envy"
04/06/2025 20:02:49	UFOP #18	AbstractWriteHolder.java		4	4 Método como initAnnotationConfig possui muito código, violando o princípio da responsabilidade única
04/06/2025 20:03:00	UFOP #33	AdminBrokerProcessor.java		4	4 No método RemotingCommand que trata de exceções, abresenta diversos switch case que utilizam do retorno de outros métodos da classe
04/06/2025 20:03:37	UFOP #19	SocketIOChannelInitializer		4	4 Houve um alto número no uso de métodos de classes externas. Além disso, pode haver também uma questão de acoplamento forte, já que muitos métodos pertenciam a mesma classe.
04/06/2025 20:07:06	UFOP #15	ProjectExportWizard		2	2 Outros code smells identificados, mas não "FEATURE ENVY" em si. Algumas referências de métodos extra classe mas nada exagerado.

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 FEATURE ENVY smell in the code?	Mention why (for instance, which code elements) you gave this rating.
04/06/2025 20:07:27	UFOP #33		QueryParser.java	4	No método privado QueryParser - o qual gera uma nova query, utiliza um objeto do tipo TokenQueue em que seus valores são retornos de condicionais que verificam tipos de retorno de outros métodos. Há também o método FindElements que sofre do mesmo problema, o uso de diversas condicionais verificando variáveis de outros métodos.
04/06/2025 20:11:36	UFOP #20		TableView	1	Acredito que não tenha nenhum Feature Envy, pois a classe não tem nenhum método que deveria pertencer a outra classe.
04/06/2025 20:12:12	UFOP #32		ConnectionPool	4	dependência excessiva de ClientConnectionsEntry e MasterSlaveEntry dentro da classe
04/06/2025 20:12:15	UFOP #31		QueryParser	4	O método faz mais chamadas a outra classe do que à sua própria, ex.: Class QueryParser
04/06/2025 20:15:07	UFOP #28		ClassicPluginStrategy	2	O método não parece haver Feature Envy.
04/06/2025 20:16:53	UFOP #22		AbstractWriteHolder.java	2	Não há, pois não há classes que usem excessivamente métodos ou dados de outra classe.
04/06/2025 20:18:11	UFOP #6		AdminBrokerProcessor	2	Enquanto possui um código bastante extenso, não notei o uso excessivo de um método de outra classe presente no código, uma vez que só havia uma instanciada no mesmo.
04/06/2025 20:19:03	UFOP #32		HttpServerTest	3	A classe acessa bastante os dados de outras classes como IHHTPSession e TestServer, especialmente no método serve()
04/06/2025 20:19:06	UFOP #1		guava-30.1.1_Monitor	3	Não consegui identificar com clareza a presença ou não do smell em questão.
04/06/2025 20:20:21	UFOP #10		ClassicPluginStrategy	2	O código não apresenta sinais de Feature Envy significativos para classificar o código como um todo como possuidor de um smell de mesmo tipo
04/06/2025 20:22:56	UFOP #20		ClassicPluginStrategy	1	long method, Possui vários níveis de aninhamento (blocos profundos de if, for, try)
04/06/2025 20:23:51	UFOP #17		IKArbitrator.java	5	O método process faz uso intensivo de outros métodos e dados de outras classes
04/06/2025 20:25:37	UFOP #1		ConnectionPool	4	A classe ConnectionPool<T> possui métodos e dados de outras classes, isso é um forte indício de Future Envy, indicando também que provavelmente algumas das suas funcionalidades deveriam ser realocadas para outro lugar dentro do código.
04/06/2025 20:25:48	UFOP #31		HikariDataSource	5	faz mais chamadas a outra classe do que à sua própria, HikariDataSource.
04/06/2025 20:29:42	UFOP #16		ApolloDataSourceFactoryBean	5	Ela possui ocode smell do tipo DATA CLASS, é uma classe que apenas faz armazenamento de dados, possui apenas getters e setter como métodos.
04/06/2025 20:31:36	UFOP #3		TextureAtlas	4	O código é uma grande classe onde possui muitas responsabilidades e subclasses. Tem um "Método Longo" para carregar dados do arquivo, que é bem complicado. No geral, ele parece funcionar bem, mas é um pouco denso.
04/06/2025 20:32:32	UFOP #18		TextureAtlas.java	2	Acho que nao possee code smells, apesar de metodo definindo muitas regioes.
04/06/2025 20:35:10	UFOP #29		SaOAuth2Template	2	Não tem feature envy, pois todos os metodos creio que deveriam estar nessa classe.
04/06/2025 20:38:46	UFOP #16		ConnectionPool	2	O código não apresenta uma FEATURE ENVY, não utiliza-se excessivamente de métodos de outras classes.
04/06/2025 20:43:03	UFOP #27		ProjectExportWizard	4	A classe possui alguns métodos que manipulam mais dados de outras classes do que de sua própria classe.
04/06/2025 20:43:07	UFOP #25		AdminBrokerProcessor	4	o código aparenta ter o smell, pois faz muitas chamadas para outra classe
04/06/2025 20:47:29	UFOP #30		AllMembersSupplier	3	A classe AllMembersSupplier possui moderado número de chamadas à outras classes.
04/06/2025 20:48:02	UFOP #25		SaOAuth2Template	3	Não consegui identificar com certeza se ele realmente possui o smell indicado