

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 INTENSIVE COUPLING smell in the code?	Mention why (for instance, which code elements) you gave this rating.
27/04/2025 14:26:38	UFMG #12	SmartContentSelector.java		3	Com toda certeza tem um Long Method, mas não há exatamente Intensive Coupling com classes do sistema, só com a API de Java (StringBuilder) se ela for modificada ou deprecated.
28/04/2025 22:35:17	UFMG #22	ResizeController		2	The class is coupled to other classes. But, I do not think this is a problem since it is a controller class.
29/04/2025 12:46:19	UFMG #26	BatchExecutor		4	A classe avaliada me parece depender muito de diversos métodos de classes como StatementHandler e Configuration.
29/04/2025 14:03:52	UFMG #26	DictSegment		1	Não me parece ter problema de Intensive Coupling, uma vez que depende de poucas classes externas e todas elas da biblioteca java util.
01/05/2025 17:58:03	UFMG #23	Faker		5	Múltiplos campos de muitas outras classes são instanciados e armazenados nessa classe, o que torna a classe dependente de inúmeras outras.
01/05/2025 19:56:15	UFMG #23	QueueTest		5	Os métodos dessa classe frequentemente utilizam de outros dela mesma, gerando uma interdependência grande entre os métodos.
03/05/2025 14:00:38	UFMG #8	FlowControllerV1		3	Parece que possui um acoplamento moderado
03/05/2025 14:09:43	UFMG #8	ConstructorConstructor		4	Usa muito Reflection e Collection. Usa bastante também componentes Gson.
05/05/2025 16:54:28	UFMG #3	BytecodeInstruction		5	O tamanho do métodos.
05/05/2025 16:58:09	UFMG #3	AssemblyProcessor		4	Métodos grandes. A classe em si parece extensa também.
05/05/2025 17:03:41	UFMG #3	JavassistProxyFactory		4	O código apresenta algumas funções com vários inputs e os métodos são grandes.
06/05/2025 07:40:55	UFMG #3	JITDataManager		4	Várias responsabilidades para uma função, com excesso de entradas map<string, string>
06/05/2025 07:50:10	UFMG #18	PageResult		5	classe com atributos, construtor e get and set somente.
14/05/2025 17:41:25	UFMG #2	DefaultJedisSocketFactory		5	A classe tem mais responsabilidades que deveria. Assim, ela não segue o princípio da responsabilidade única.
14/05/2025 17:42:31	UFMG #8	JobGroupController		4	A classe JobGroupController é extensa possuindo mais de 100 linhas de código. Além disso, possui várias responsabilidades. Por isso, acredito que é um caso de Large Class
14/05/2025 17:43:24	UFMG #6	QueueTest		5	A classe é enorme.
14/05/2025 17:46:20	UFMG #6	GeometryViewerRegistry		5	Classes longas, métodos longos, muitos cheiros.
14/05/2025 17:48:12	UFMG #6	PoolStats		5	Data class, definitivamente.
14/05/2025 17:49:33	UFMG #1	PrefixPluginLogger		3	Talvez um feature envy ou métodos com muitos parâmetros.
14/05/2025 17:50:30	UFMG #10	ExcelWriteFillExecutor		4	Esse código possui alguns code smells como classe excessivamente grandes com métodos longos e acoplamentos elevados. Além disso, possui nomes ambíguos ou genéricos que podem dificultar um pouco na leitura dos códigos, o que me atrapalhou um pouco para entender o que a classe pretendia implementar.
14/05/2025 17:50:57	UFMG #20	CommandDecoder		4	Large Class: A classe lida com decodificação de comandos, construção de objetos, manipulação de buffer e controle de estado. São responsabilidades distintas concentradas em uma única classe.
14/05/2025 17:51:44	UFMG #28	ResizeController		5	O código implementa uma classe que contém diversos métodos que fazem diferentes coisas, atribuindo a ela muitas responsabilidades. Isso caracteriza um code smell do tipo Large Class.
14/05/2025 17:52:44	UFMG #7	ConstructorConstructor		4	O código tem muito acoplamento, principalmente no método get(), que verifica múltiplos tipos e cria objetos de muitas classes diferentes
14/05/2025 17:53:19	UFMG #32	AbstractSendMessageProcessor		3	O código tem um problema de large class, pois tem uma classe excessivamente longa.
14/05/2025 17:54:37	UFMG #30	ClusterConnectionManager		3	Além disso, a classe possui diversos métodos diferentes que fazem a mesma coisa ou coisas extremamente parecidas.
14/05/2025 17:55:54	UFMG #33	Finance		5	Métodos como buildInnerMsg, msgCheck, parseRequestHeader e decodeSendMessageHeaderV2 são muito longos e complexos.
14/05/2025 17:56:48	UFMG #35	DriverDataSource		5	A classe possui acoplamento intenso com muitas outras classes, como ClusterPartition e RedisURI. Manipula diretamente estados internos e lida com muitas responsabilidades.
14/05/2025 17:58:24	UFMG #39	MethodRoadie		5	O código apresenta uma única classe com mais de 150 linhas, caso de Large Class. Além disso, essa classe tem um método chamado createCountryCodeToBasicBankAccountNumberPatternMap, cujo nome é excessivamente longo e tem aproximadamente 70 linhas, fazendo um comando .put. Com isso, acredito que seja um caso de Large Method.
14/05/2025 17:59:08	UFMG #31	MethodRoadie.java		5	O código possui uma classe com extensão de código muito extensa, portanto possui o Code Smell: Large Class.
14/05/2025 17:59:42	UFMG #36	LexemePath		1	A classe MethodRoadie tem outras quatro classes como atributos dentro da classe, e elas são amplamente utilizadas nos métodos dessa classe, então qualquer mudança nessas quatro classes afetaria a classe MethodRoadie, o que causa um acoplamento severo.
14/05/2025 17:59:51	UFMG #17	DiagnosedStreamCorruptionException		5	Ele possui Intensive Coupling porque ele depende de várias classes e os métodos chamam métodos de outros objetos. Além disso a classe é muito longa (Large Class e Long Method) e os métodos são relativamente longos e fazem várias coisas. O próprio código já indica que está obsoleto (@Deprecated)
14/05/2025 18:00:45	UFMG #30	PacketEncoder		1	Não há Intensive coupling, pois só existe uma classe.
14/05/2025 18:01:35	UFMG #12	ConstructorConstructor		5	Método toString depende fortemente da outra classe criada, podendo criar vulnerabilidades no código.
14/05/2025 18:02:55	UFMG #38	NioClientTest		4	A classe depende de muitas APIs externas e classes específicas (ByteBuf, JsonSupport, Packet, etc.), indicando acoplamento excessivo.
14/05/2025 18:02:58	UFMG #13	Finance.java		5	The class ConstructorConstructor overly depends on the internal implementations of the class ObjectConstructor across its methods.
14/05/2025 18:04:51	UFMG #20	DubboServiceMetadataRepository		3	I don't think the code is Intensive Coupling because the coupling is more spread between the classes ByteBuffer, BufferUtil and Scanner. This looks more like Dispersed Coupling to me.
14/05/2025 18:05:34	UFMG #9	FlowControllerV1		2	Existe muito acoplamento na classe, ela é responsável por validar o iban, o bie e lidar com o cartão de crédito.
14/05/2025 18:07:03	UFMG #29	JITDataManager		5	No envio anterior não contei este code smell portanto estou reenviando.
14/05/2025 18:08:21	UFMG #25	StatisticSlot		5	Há um grande número de imports de outras partes do sistema
14/05/2025 18:09:03	UFMG #39	JavassistProxyFactory		5	Toda essa classe depende intensamente de outra classe e dos seus métodos, o que torna muito vulnerável caso algum erro ocorra.
14/05/2025 18:09:42	UFMG #30	ConnectionPageShellCommands		5	A classe faz muito uso dos métodos da 'Context context' e da 'DefaultNode node', algumas das várias vezes que acontecem no código:
14/05/2025 18:09:48	UFMG #20	JavaBeanDeserializer		4	node.increaseThreadNum(); node.addPassRequest(count);
14/05/2025 18:10:29	UFMG #4	BeanCopier.java		2	Por ser uma factory, não me parece que o fato de utilizar outras classes seja um problema, me parece que a classPool é somente uma classe auxiliar.
14/05/2025 18:11:55	UFMG #14	BytecodeInstruction		5	Classe altamente acoplada a múltiplas classes e componentes da UI SWT e modelo do DBeaver
14/05/2025 18:15:03	UFMG #34	BytecodeInstruction		5	No envio anterior não contei este code smell portanto estou reenviando.
14/05/2025 18:15:22	UFMG #24	Finance		5	A classe interage intensamente com diversas classes do framework de serialização/deserialização. Além disso, depende de estruturas internas e detalhes específicos do funcionamento dessas classes. Essa forte acoplagem com múltiplos componentes internos torna o código rígido e difícil de alterar sem impactos em cadeia.
14/05/2025 18:15:33	UFMG #29	JavassistProxyFactory		3	The class has a lot of imports which are being used, so it's a good indicator of intensive coupling.
14/05/2025 18:15:34	UFMG #21	JavassistProxyFactory		3	A função é extremamente grande mas não acredito estar usando de mais de outros métodos, ela de certa forma possui muitos objetos parecendo ser coesos com o que a classe propõem. PS: Acredito que na hora de enviar houve erro então estou enviando novamente este forms.
14/05/2025 18:16:05	UFMG #27	JsonAdapterAnnotationTypeAdapterFactory		4	Minha confiança é 4. A classe BytecodeInstruction demonstra Intensive Coupling especialmente com as classes Opcode e as implementações de IBytecodeParam (como BCParamSwitch). Isso é evidente nos métodos toStringSwitch e toStringNonSwitch, onde a lógica de formatação da string de saída depende fortemente de chamar métodos de Opcode (e.g., opcode.isSwitch()), opcode.getMnemonic()) e de acessar e converter os parâmetros para tipos específicos para obter seus dados, indicando um forte conhecimento e dependência da estrutura interna desses outros objetos.
14/05/2025 18:16:28	UFMG #41	SmsFlashPromotionProductRelationServiceImpl		3	Tem alguns métodos longos, poderiam estar em outro encapsulamento
14/05/2025 18:17:52	UFMG #39	JUnitCommandLineParseResult		4	Como alguém que não tem um entendimento completo da função acredito que a classe possui muitos métodos mas não sei dizer se é um overfitting ou não. Além disso a classe é muito grande o que aumenta a possibilidade deste overfitting.
14/05/2025 18:18:17	UFMG #29	ConnectionPageShellCommands		5	Minha confiança é 5. A classe JavassistProxyFactory exibe um claro Intensive Coupling com a biblioteca Javassist, utilizando intensivamente suas classes como CtClass, CtMethod, ClassPool, e CtNewMethod para gerar dinamicamente e modificar o bytecode de classes proxy. A funcionalidade central desta classe é totalmente dependente das capacidades e da API detalhada fornecida pela Javassist, manipulando diretamente estruturas de baixo nível do bytecode através dela.
14/05/2025 18:19:02	UFMG #5	PageResult		5	A classe tem dependências necessárias para funcionar, mas não apresenta intensive coupling.
14/05/2025 18:20:04	UFMG #33	NacosWatch		1	Ela é uma peça simples que faz a ligação entre annotation e criação do adaptador.
14/05/2025 18:20:45	UFMG #15	JavaBeanDeserializer		4	O acoplamento que existe é natural e esperado para o padrão usado.
14/05/2025 18:21:07	UFMG #40	BatchExecutor		1	Me parece que a maioria das funcionalidades dependem de chamar outras classes, então a classe está extremamente acoplada a essas classes auxiliares.
14/05/2025 18:21:14	UFMG #35	InPacketHandler		4	Essa classe depende muito dos métodos de uma outra específica e qualquer alteração nessa outra poderia gerar vários erros devido a essa dependência. Além de ser uma classe muito grande.
14/05/2025 18:21:39	UFMG #29	QueueTest		5	A classe PageResult não parece depender fortemente de outras classes nos seus métodos ou objetos, com métodos que modificam somente seus objetos. Parece ter mais problemas de Data Class.
14/05/2025 18:21:42	UFMG #22	BytecodeInstruction.java		5	O código tem muitos casos de Override, chamando diversos métodos de outra classe, caso de Intensive Coupling. Além disso, tem um método start, com longos IFs, um dentro do outro, caso de Long Method.
14/05/2025 18:22:51	UFMG #40	DefaultJedisSocketFactory		5	Não tem Intensive Coupling mas tem Large Class, uma vez que a classe tem mais de 1000 linhas.
14/05/2025 18:23:50	UFMG #43	DictSegment		5	A classe instancia muitas outras e realiza operações com elas. Há vários code smells além de Intensive Coupling, como Long Method, violação do princípio de responsabilidade única etc
14/05/2025 18:25:04	UFMG #21	AssertUtil		1	A classe InPacketHandler possui diversas dependências então não possui dependência excessiva em nenhuma outra única classe em específico.
				1	A classe é extremamente grande mas não vejo ela usando desnecessariamente outros métodos e não acho muito pelo trânsito da classe
				5	Este não é o codesmell principal, pois as classes e métodos deste código são todos muito grandes, sendo este o principal problema aqui, mas há sim certo acoplamento
				5	Muita dependência de outras classes, além de possuir muitas responsabilidades e ser muito grande.
				4	Depende muito de Character, indicando Intensive Coupling
				1	A classe AssertUtil não sofre de Intensive Coupling porque sua lógica é coesa, utiliza poucas chamadas por classe e apenas interage superficialmente com alguns utilitários

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 INTENSIVE COUPLING smell in the code?	Mention why (for instance, which code elements) you gave this rating.
14/05/2025 18:26:25	UFMG #43	DictSegment			A presença de diversos método para implementar a ideia de match caracteriza um Shotgun Surgery já que caso a ideia precise ser reformulada, todos estes métodos necessitariam de receber pequenas mudanças.
14/05/2025 18:26:28	UFMG #27	SmartContentSelector		3	Definitivamente há presença de long methods também, principalmente ao avaliar a complexidade de algumas de suas funções.
14/05/2025 18:26:52	UFMG #9	PacketEncoder		4	Não consegui avaliar o nível de acoplamento entre ela e outras funções devido a claridade dos comentários (que estão em chines eu acho(?)) e a minha inexperiência com java.
14/05/2025 18:28:16	UFMG #25	TestProxies		3	Sim, tem Intensive Coupling porque depende e usa muitas classes externas, centralizando muita lógica e gerando forte ligação com várias partes do sistema.
14/05/2025 18:28:51	UFMG #19	NacosWatch		1	Não apresenta Intensive Coupling.
14/05/2025 18:32:31	UFMG #38	MethodRoadie		1	A classe tem poucas dependências focadas, e não tem ligações excessivas ou uso intenso de outras partes do sistema.
14/05/2025 18:32:39	UFMG #42	FireBirdMetaModel.java		4	Não foi encontrado nenhum indicio de Intensive Coupling.
14/05/2025 18:33:18	UFMG #43	PassphraseEntryController		4	The class shows intensive coupling. It makes numerous method calls to TestMethod, like isIgnored, getTimeout, invoke, expectsException, getExpectedException, isUnexpected, getBefores, and getAfters
14/05/2025 18:35:40	UFMG #16	DriverDataSource		2	Código muito grande, com vários métodos
14/05/2025 18:35:40	UFMG #16	DriverDataSource		1	Não apresenta ter Intensive Coupling e depender diretamente de outra classe pois os tipos de variáveis para coisas semelhantes são diversos.
14/05/2025 18:36:22	UFMG #43	ResizeController		1	Intensive Coupling - Não encontrado
14/05/2025 18:36:57	UFMG #37	ConnectionPageShellCommands		1	Long Method - O método "public DriverDataSource(String jdbcUrl, String driverClassName, Properties properties, String username, String password)" possui muitas linhas, com diversas condicionais e várias operações
14/05/2025 18:43:57	UFMG #37	BeanCopier		4	Refused Bequest - Muitos métodos herdados sofreram override
14/05/2025 18:44:00	UFMG #22	DirectoryBrowserSupport.java		5	Depende muito de Region e seus métodos utilizam muito MouseEvent, sem outras possibilidades
14/05/2025 18:49:18	UFMG #12	DriverDataSource		5	Há o código smell Intensive Coupling na classe, pois ela depende de oito parâmetros advindos de outras classes para ser implementado.
14/05/2025 18:51:37	UFMG #41	SmartContentSelector		1	Eu não sei dizer com certeza se há algum code smell no código, pois não consegui entendê-lo muito bem. Porém, acredito que não, pois a classe depende apenas de dois parâmetros vindos de outras classes na sua implementação.
14/05/2025 18:55:45	UFMG #11	ServiceLoader		5	A classe DriverDataSource é bastante dependente da classe jdbcUrl. Se o tipo de URL precisar mudar ou se for utilizado outro tipo de url, teria que haver diversas alterações no código.
27/05/2025 15:14:49	UFMG #14	ClassLoaderCommand		1	A única classe SmartContentSelector não utiliza dados de outra classe específica que poderiam indicar um problema de Intensive Coupling
27/05/2025 15:21:02	UFMG #7	Tokeniser		4	A classe é bastante dependente de outras classes em seus métodos.
04/06/2025 19:39:39	UFOP #7	SimpleWebServer		5	O método process centraliza muitas responsabilidades e ramificações atuando como um dispatcher violando o princípio da responsabilidade única
04/06/2025 19:45:05	UFOP #22	EsProductServiceImpl.java		4	A classe Tokeniser possui acoplamento excessivo com diversas classes externas e internas, tornando dependente de muitos pontos de implementação
04/06/2025 19:46:53	UFOP #21	ClusterConnectionManager		5	a manipulação direta de NanoHTTPD, WebServerPlugin e estruturas como mimeTypeHandlers indica um acoplamento bem forte, mas não é suficiente para uma nota máxima.
04/06/2025 19:48:41	UFOP #1	SimpleWebServer		2	Não há Intensive Coupling porque não há uma classe que possui um método com detalhes dependentes de outra classe.
04/06/2025 19:48:45	UFOP #17	GeneralAppIdDecoder.java		2	Acredito que o código têm poucas chances de representar um code smell do tipo Intensive Coupling, pois ele depende de várias outros objetos e também por estar mais parecido com um Dispersed Coupling.
04/06/2025 19:50:44	UFOP #24	EsProductServiceImpl		4	O método public Response serve(IHttpSession session) depende muito dos detalhes internos da classe IHttpSession session, de forma que qualquer mudança realizada nessa classe resultará em um provável não funcionamento do método citado.
04/06/2025 19:53:21	UFOP #26	PacketEncoder		2	há dependências de outras classes, mas nada intensivo que possa justificar esse code smells
04/06/2025 19:55:11	UFOP #1	StpLogic		5	Esse sistema sobrescreve todos os métodos da classe que implementa, também não instancia um objeto de fora dela para ficar dependente dele.
04/06/2025 19:57:21	UFOP #5	AbstractSendMessageProcessor.java		5	Os métodos encodeJsonP e encodePackets são dependentes de encodePacket.
04/06/2025 19:57:51	UFOP #2	HostAndPort		1	Não consegui detectar com certeza um Intensive Coupling, porque existem sim métodos e classes nesse código que dependem muito de detalhes de outros, entretanto, pelo contexto do código, é possível argumentar que essa dependência seja necessária para a estrutura do sistema e seu objetivo.
04/06/2025 20:01:40	UFOP #19	DubboServiceMetadataRepository		1	Não existem dependências notáveis de objetos, métodos ou atributos de outra classe em específico.
04/06/2025 20:01:52	UFOP #13	ClusterConnectionManager.java		4	Me pareceu relativamente alto o número de métodos utilizados provenientes do alibaba, o que me parece ser um caso de acoplamento forte.
04/06/2025 20:04:38	UFOP #31	StpLogic.java		1	A classe ClusterConnectionManager.java não sofre de Intensive Coupling, mas sim de Dispersed Coupling, pois depende de muitas outras classes (vide declaração de atributos).
04/06/2025 20:05:57	UFOP #21	EsProductServiceImpl		5	5 muitas classes dependendo uma da outra...
04/06/2025 20:06:38	UFOP #3	HostAndPort		2	Acredito que o código está mais voltado para a classificação inversa do que foi sugerido, pois está mais parecido com o Dispersed Coupling por depender de diversos outros serviços para além do código que podem algum dia parar de funcionar.
04/06/2025 20:06:48	UFOP #27	ClassLoaderCommand		2	Achei o código ok, tem um método fromString que tem muitas responsabilidades, mas o restante do código aparenta ser tranquilo caso haja necessidade de manutenção e evoluções futuras.
04/06/2025 20:07:36	UFOP #2	WriteWorkbookHolder		5	A classe apresenta fortes indicativos de alto acoplamento, como:
04/06/2025 20:07:54	UFOP #19	CronExpression		1	- Importa de vários pacotes na mesma classe, o que acaba por tornar a classe altamente dependente de outras;
04/06/2025 20:08:02	UFOP #13	SimpleWebServer.java		4	O funcionamento geral do código depende bastante de um objeto da classe Workbook, qualquer mudança nela poderá impactar a classe WriteWorkbookHolder profundamente.
04/06/2025 20:09:37	UFOP #23	WriteWorkbookHolder		2	Não estou muito certo sobre esse code smell, devido ao fato de que não consegui notar nada muito alarmante.
04/06/2025 20:11:15	UFOP #18	GeneralAppIdDecoder.java		1	A classe não depende profundamente de outros métodos/detalhes de outra classe. Ao menos, não de classes específicas, apenas de classes consolidadas do Java, como o uso do equalsIgnoreCase().
04/06/2025 20:14:06	UFOP #21	FlowControllerV1		4	Não consegui identificar
04/06/2025 20:14:24	UFOP #24	DubboServiceMetadataRepository		4	Sim, código possui muitas funcionalidades antigas que foram desativadas mas que não foram retiradas do código, o que pode causar confusão caso outro desenvolvedor venha dar manutenção nesse código.
04/06/2025 20:16:30	UFOP #4	GeneralAppIdDecoder		3	Acredito que o código está mais parecido com a classificação como Long Method, pois temos métodos bastante grandes ao longo do código, mas, também, tem chances de ser classificado como Intensive Coupling, pois depende de recursos do alibaba, por exemplo.
04/06/2025 20:20:50	UFOP #33	EsProductServiceImpl.java		4	Por causa de trechos de códigos semelhantes a esse: private final ObjectMapper objectMapper = new ObjectMapper();
04/06/2025 20:27:41	UFOP #13	CronExpression.java		4	Atribui nota 4, pois também foi um pouco difícil de analisar este código, mas pelo o que eu pude entender. O número é decodificado por um método inicial e depois são feitas novas validações para checar se ele ainda precisa ser decodificado. Isso cria uma certa dependência de várias outros métodos.
04/06/2025 20:28:23	UFOP #6	WriteWorkbookHolder		3	A maioria das funcionalidades utiliza das dependências do SpringFramework. Não sei se seria considerado um Intensive Coupling
04/06/2025 20:28:55	UFOP #22	WriteWorkbookHolder.java		1	A classe não depende profundamente de nenhuma outra classe e faz uso mais intenso apenas de classes consolidadas do Java.
04/06/2025 20:29:18	UFOP #28	StpLogic		4	A classe analisada depende muito do conteúdo de outra classe, o que acaba tornando sua manutenção bastante demorada/complicada.
04/06/2025 20:30:40	UFOP #6	Tokeniser		3	3 Não há, porém tenho dúvida. Não acredito que a classe seja bem dependente de detalhes de outras classes
04/06/2025 20:31:05	UFOP #29	DirectoryBrowserSupport		5	5 O método possui Intensive Coupling devido ao vasto número de importações de outras classes, o que indica grande dependência.
04/06/2025 20:33:44	UFOP #2	StringUtil		1	A classe analisada utiliza métodos próprios, sem depender do conteúdo de outra classe, o que facilita a manutenção de seu código, uma vez que não irá depender do conteúdo externo.
04/06/2025 20:33:58	UFOP #27	ClassWriter		4	Acredito que exista intensive coupling, pois a classe importa e utiliza diretamente várias classes, se caso algumas dessas classes mudar ela precisa ser alterada.
04/06/2025 20:39:47	UFOP #4	HostAndPort		1	1 Em geral a classe não apresenta dependências destacadas de outra classe em específico.
04/06/2025 20:41:48	UFOP #30	HostAndPort		4	1 Não identifiquei muitos aspectos que pudessem indicar alto acoplamento.
04/06/2025 20:44:38	UFOP #16	StringUtil		5	4 Atribui a nota 4, visto que quase todos os métodos dependem das variáveis port e host, que são manipuladas em outros métodos.
04/06/2025 20:44:38	UFOP #16	StringUtil		2	5 A classe HostAndPort depende quase que exclusivamente de checkArgument().
04/06/2025 20:44:38	UFOP #16	StringUtil		2	2 O código não apresentou uma utilização excessiva de detalhes internos de outra classe/módulo, logo, acredito que não haja o code smell INTENSIVE COUPLING