Carimbo de data/hora	Endereço de e-mail	Formação	Anos de termos	Na quanto tempo você trabalha como programador	Você já trabalhou com alguma dessas linguagens	Diariamente, utilizamos o comando "grep" para	Em uma situação de desenvolvimento, qual dos ch	Com base no seu resposta anterior, por favor, expl	Você achou que alguma das imagens era confusa	A ação exemplificada na imagem (movimento p	Em uma situação de desenvolvimento, qual dos ch	Com base no seu resposta anterior, por favor, explique o que influenciou sua escolha. Se você marcou um dos dois sem	Você achou que alguma das imagens era confusa	A ação exemplificada na imagem (mudança no tex	Em uma situação de desenvolvimento, qual dos ch	Com base no seu resposta anterior, por favor, expl	Você achou que alguma das imagens era confusa	A ação exemplificada na imagem (alteração de ver	Em uma situação de desenvolvimento, qual dos ch	Com base no seu resposta anterior, por favor, expl	Você achou que alguma das imagens era confusa	A ação exemplificada na imagem (alteração de ver	Após analisar os casos, você acredita que o uso de	Você estaria interessado em abrir ferramenta
09/01/2025 17:13:57	Participante_1		Ll no termos e concordo	2 anos - 5 anos	C	Não	Resolução 2	A segunda opção é visualmente menos confusa	Sim	A imagem de Resolução 1	Sim	Resolução 3	O resultado 3 é melhor pois ao refatorar apenas uma trecho de código alterados e nao ha linha	Sim	Resolução 3	Sim	Sim	Resolução 3	Sim	Resolução 3	Sim	Sim	Sim	Gostei bastante dessa nova abordagem! Acredito q Sim
09/01/2025 18:11:31	Participante_2		Ll no termos e concordo	5 anos e mais	Java	Não	Resolução 2	Considero visualmente 2 como bastante superior, pois ela não apenas mostra o que foi alterado, m a recontextualização do um pedaço de código	Sim	A imagem de Resolução 1	Sim	Os dois resultados são equivalentes	Nesse exemplo específico, acredito que o único e o embaralhamento das alterações realmente é bem simples e deixa em antes alterdiagrm, não fazendo grande diferença no tempo gasto para compreender as alterações realizadas.	Sim	Resolução 3	Sim	Sim	Resolução 3	Sim	Resolução 3	Sim	Sim	Sim	Sim, acredito que a ferramenta seria magnífica para Tabea
09/01/2025 21:29:12	Participante_3		Ll no termos e concordo	5 anos e mais	Java, C	Não	Resolução 2	Visualmente mais rápido de ler	Sim	A imagem de Resolução 1	Não	Resolução 3	Para menos motivo, parecia uma leitura mais rápida	Sim	Resolução 3	Não	Sim	Resolução 3	Não	Resolução 3	Não	Não	Sim	Se a diff contém o a comparação visual antes seja Sim
09/01/2025 21:44:05	Participante_4		Ll no termos e concordo	2 anos - 5 anos	Java	Não	Resolução 2	3 respostas 2 e 3 não está mais legível no meio do texto	Sim	A imagem de Resolução 1	Não	Os dois resultados são equivalentes	Dadíquer meio tempo para analisar o resultado 3 e 4. Ambos ainda claro.	Sim	Resolução 3	Sim	Sim	Resolução 3	Sim	Resolução 3	Sim	Sim	Sim	Acredito que sim, se tornar o código mais legível. Sim
09/01/2025 14:07:03	Participante_5		Ll no termos e concordo	2 anos - 5 anos	Java, C	Não	Resolução 2	Ambos são claros, porém em 2 ha mais opções	Sim	A imagem de Resolução 1	Sim	Resolução 2	É o menos e penso ser claro	Sim	Resolução 2	Sim	Sim	Resolução 2	Sim	Resolução 2	Sim	Sim	Sim	Sim, principalmente em linguagens fracamente tipa Sim
09/01/2025 09:42:06	Participante_6		Ll no termos e concordo	2 anos - 5 anos	Java, C	Não	Resolução 2	Fácil ler e entendendo	Sim	A imagem de Resolução 1	Não	Os dois são claros de entender	Os dois são claros de entender, mas a primeira ainda que é uma ação melhor para visualizarmos "juntar" um bug	Sim	Resolução 3	Sim	Sim	Os dois resultados são equivalentes	Sim	Os dois resultados são equivalentes	Sim	Sim	Sim	Acredito que sim! Diff requer ou mais complexo. Sim
09/01/2025 09:56:54	Participante_7		Ll no termos e concordo	5 anos e mais	Java	Sim	Resolução 2	O segundo é o que eu utilizo no trabalho	Sim	A imagem de Resolução 1	Sim	Resolução 3	ele melhor de visualizar	Não	Resolução 3	Sim	Sim	Resolução 3	Sim	Resolução 3	Sim	Sim	Sim	sim, ao p utilizo extensões do vs code que formam Sim
09/01/2025 09:57:45	Participante_8		Ll no termos e concordo	5 anos e mais	Java, C	Não	Resolução 2	Foi o mais fácil de identificar visualmente	Sim	A imagem de Resolução 1	Sim	Resolução 3	visibilidade	Não	Resolução 3	Sim	Sim	Resolução 3	Sim	Resolução 3	Sim	Sim	Sim	Sim,uma forma de algum meu trabalho Sim
07/01/2025 15:02:41	Participante_9		Ll no termos e concordo	2 anos - 5 anos	Python	Não	Resolução 1	Ne 1 é mais claro o que foi alterado	Sim	A imagem de Resolução 2	Sim	Resolução 3	Agora dá para ver a diferença entre o antes/dep e o curto. Ele deixa mais claro o que foi alterado	Sim	Resolução 3	Sim	Sim	Resolução 3	Sim	Resolução 3	Sim	Sim	Sim	Acho que importante, sim, mas depende de caso. Sr Tabea
08/01/2025 23:14:06	Participante_10		Ll no termos e concordo	5 anos e mais	Java Typescript, PHP	Não	Resolução 1	mais simples de ser entendido e a comparação ao	Sim	A imagem de Resolução 1	Sim	Resolução 3	mais claro de se entender	Sim	Resolução 3	Sim	Sim	Resolução 3	Sim	Resolução 3	Sim	Sim	Sim	creio que seria algo trivial, como costumo ambos ok Tabea	
09/01/2025 07:14:54	Participante_11		Ll no termos e concordo	2 anos - 5 anos	Java, Python	Não	Os dois resultados são equivalentes	Apenas a ordem das mudanças no código também.	Sim	A imagem de Resolução 1	Sim	Resolução 3	Menos código	Sim	Resolução 3	Não	Os dois resultados são equivalentes	Sim	Resolução 3	Não	Não	Sim	Acredito que sim, sim, mas posicionamento Tabea	
09/01/2025 12:08:04	Participante_12		Ll no termos e concordo	5 anos e mais	Java, C,C#,Javascript, delphi, Vba	Sim	Resolução 2	Comparação lado a lado visualmente é melhor	Sim	A imagem de Resolução 1	Sim	Resolução 3	Visualmente fica mais fácil a comparação	Sim	Resolução 3	Sim	Sim	Resolução 3	Sim	Resolução 3	Sim	Sim	Sim	Sim, 2 uma situação muito recorrente Sim	
09/01/2025 14:00:47	Participante_13		Ll no termos e concordo	5 anos e mais	Java, C, Go	Sim	Os dois resultados são equivalentes	Entender não é a melhor expressão. Resolução 1 é mais fácil de ler, mas a versão Resolução 2 mais claro. Ou ao ou dois.	Sim	A imagem de Resolução 1	Não	Resolução 4	Resolução	Sim	Resolução 3	Sim	Sim	Resolução 3	Sim	Resolução 3	Sim	Sim	Sim	Provavelmente. Pode ser útil para identificar alguns Sim	
09/01/2025 14:02:04	Participante_14		Ll no termos e concordo	5 anos e mais	Java, C	Não	Resolução 2	Entender uma alteração mesmo de deslocamento de um pedaço de código é mais difícil. Ela envolve um conjunto de ações de remoções e inserções que de fato, não existiram.	Sim	A imagem de Resolução 1	Sim	Os dois resultados são equivalentes	É possível visualizar o código anterior e a alteração no mesmo imagem.	Sim	Resolução 3	Sim	Sim	Resolução 3	Sim	Resolução 3	Sim	Sim	Sim	Sim. Parecem uma evolução. Causa forma é para Sim	
09/01/2025 14:05:02	Participante_15		Ll no termos e concordo	5 anos e mais	Java	Não	Resolução 2	O segundo é o que eu utilizo no trabalho	Sim	A imagem de Resolução 1	Não	Resolução 4	o resultado 4 é mais claro pois sabemos exatamente onde	Sim	Resolução 3	Sim	Sim	Resolução 3	Sim	Resolução 3	Sim	Sim	Sim	Sim, fica mais claro e rápido de detectar e analisar. Sim	
09/01/2025 19:14:07	Participante_16		Ll no termos e concordo	2 anos - 5 anos	Java Typescript	Sim	Resolução 2	Escolhi o segundo resultado, pois ainda que fica melhor a visualização e entendimento do código	Não	A imagem de Resolução 1	Sim	Resolução 3	Como eu falava anteriormente, na outra pergunta, este resultado é o que mais me ajuda a ver que realmente	Sim	Resolução 3	Sim	Sim	Resolução 3	Sim	Resolução 3	Sim	Sim	Sim	Acredito que é extremamente útil, sim, sim de ser prático Sim	
09/01/2025 15:00:27	Participante_17		Ll no termos e concordo	5 anos e mais	Java	Não	Resolução 2	Mesma coisa para o que eu falei na questão anterior	Sim	A imagem de Resolução 1	Sim	Resolução 3	Mesma coisa	Sim	Resolução 3	Sim	Sim	Resolução 3	Sim	Resolução 3	Sim	Sim	Sim	Pode ser mais claro e simples a utilização Sim	
09/01/2025 15:01:03	Participante_18		5 anos e mais	Java	Sim	Sim	Resolução 2	O resultado 2 é mais claro, visualmente, uma identificação rápida das mudanças do código	Não	A imagem de Resolução 1	Sim	Os dois são operandos permitem rápida identificação da alteração realizada	Os dois são operandos permitem rápida identificação da alteração realizada	Sim	Resolução 3	Sim	Sim	Resolução 3	Sim	Resolução 3	Sim	Sim	Sim	Sim. Além de ser rápida percepção do que está sendo Sim	
09/01/2025 19:41:33	Participante_19		Ll no termos e concordo	5 anos e mais	Java Angular	Não	Resolução 2	Fica mais claro e menos ambíguo	Sim	A imagem de Resolução 1	Sim	Resolução 3	Ver o arquivo antigo e o novo separados, ve gir ele ocorreu ou duas linhas juntas no mesmo arquivo, fica por que compara	Sim	A imagem de Resolução 3	Sim	Sim	Resolução 3	Sim	Resolução 3	Sim	Sim	Sim	Muito mais fácil de ser as mudanças no Resolução 1	Sim
09/01/2025 17:41:37	Participante_20		Ll no termos e concordo	5 anos e mais	Java Python	Não	Resolução 2	Acredito que o resultado 2 é superior pq é mais fácil de ler	Sim	A imagem de Resolução 1	Sim	Resolução 3	Da mesmaira semelhante ao resultado 2, o resultado 3 é superior pq é mais correto. Ao olhar para ele, posso atenção sem	Sim	Resolução 3	Sim	Sim	Resolução 3	Sim	Resolução 3	Sim	Sim	Sim	O resultado 3 é superior pq é mais correto. Sim	
10/01/2025 01:05:06	Participante_21		Ll no termos e concordo	5 anos e mais	Java Typescript	Não	Resolução 2	Um dos problemas com PR's no mundo typescript é o uso de ferramentas de lint, ou prettier, que acabam mudando a ordem das coisas e acabam movimentos feitos ao longo do tempo aparecem como alterados no PR (ao ter uma ofuscação da diferença por ex.)	Sim	A imagem de Resolução 1	Sim	Resolução 3	Ed pra identificar mais rápido onde está a mudança	Sim	Resolução 3	Sim	Sim	Resolução 3	Sim	Resolução 3	Sim	Sim	Sim	Sim a adição de código licenças não feita é ótima Sim	
10/01/2025 16:44:01	Participante_22		Ll no termos e concordo	5 anos e mais	Java, C, C++, python, php	Não	Resolução 1	A escolha foi tendenciosa por uma melhor legibilidade e mais familiaridade	Sim	A imagem de Resolução 2	Sim	Resolução 4	A legibilidade de ambos no nosso não tinha, para ampuxos com muitas modificações, preferiria o resultado 3 (lado a lado	Sim	Resolução 3	Sim	Sim	Resolução 3	Sim	Resolução 3	Sim	Sim	Sim	Sim, principalmente em revisões de PR e as ameiras Tabea	
10/01/2025 17:42:22	Participante_23		Ll no termos e concordo	2 anos - 5 anos	Java	Sim	Resolução 2	Com aqui avaliação não creio nova amarela (não desconsiderei esta opção, creio ser só simplificação a revisão do uma dada alteração, mas ainda é válida a ressalva de que ainda não avaliado 1, infelizmente a cargo cognitivo para compreender alguma mudanças	Sim	A imagem de Resolução 1	Sim	Resolução 3	No caso da uma mudança isolada, o resultado 3 desenha ajuda que 1/o ofendeu. Sugerindo uma alteração mais sutil, o resultado 2 ajuda muito a ver o que foi alterado. Sugerindo uma alteração mais sutil, o resultado 3 desenharia um artifício maior para entender o que foi alterado.	Sim	Resolução 3	Sim	Sim	Resolução 3	Sim	Resolução 3	Sim	Sim	Sim	Sim	Um impacto positivo, ao reduzir o cognitivo cognitivo Sim	
10/01/2025 11:40:19	Participante_24		Ll no termos e concordo	5 anos e mais	Java, C, Python	Sim	Os dois resultados são equivalentes	Ao duas são recontextuaiss do código	Sim	A imagem de Resolução 1	Não	Os dois resultados são equivalentes	Mostram claramente a linha em que houve alteração	Sim	Resolução 3	Não	Os dois resultados são equivalentes	Sim	Resolução 3	Não	Não	Sim	Pode comprovar o uso de varias ferramentas de com Não	
10/01/2025 12:50:01	Participante_25		Ll no termos e concordo	5 anos e mais	Java, C,C#	Não	Resolução 2	É difícil identificar a recontextização do código no resultado 1 e 3 quando ocorre muitas alterações em um mesmo arquivo.	Sim	A imagem de Resolução 1	Não	Resolução 3	Troca de menos é uma operação de remoção e ao menos a linha inteira como modificada, isso no mais tempo pesou	Sim	A imagem de Resolução 3	Não	Os dois resultados são equivalentes	Sim	Resolução 3	Não	Não	Sim	Para os exemplos citados, o diff sintático ajudou em Sim	
10/01/2025 13:47:14	Participante_26		Ll no termos e concordo	5 anos e mais	Java, C, Python, C++, e outros	Não	Resolução 2	Acostumei a ver menos ambíguo mesmo para entender o que mudou.	Não	A imagem de Resolução 1	Não	Resolução 3	Fica claro o que mudou e com certeza é superior em relação a mudança	Não	A imagem de Resolução 3	Não	Os dois resultados são equivalentes	Não	Resolução 3	Não	Não	Sim	Facilitaria para entender, menos ambíguo mesmo Sim	
10/01/2025 14:25:06	Participante_27		Ll no termos e concordo	2 anos - 5 anos	JavaScript, TypeScript, Kotlin	Sim	Resolução 2	Código agrio e diff mais correto	Sim	A imagem de Resolução 1	Sim	Resolução 3	Código mais correto	Sim	Resolução 3	Sim	Sim	Resolução 3	Sim	Resolução 3	Sim	Sim	Sim	Sim, o diff sintático ajuda sim de forma positiva c Sim	
10/01/2025 15:24:41	Participante_28		Ll no termos e concordo	5 anos e mais	Java, C	Não	Resolução 2	Acredito que para código mais complexo, a capacidade de identificar movimentos do código é muito boa.	Sim	A imagem de Resolução 1	Não	Resolução 4	Nesse caso, por estar acostumado a fazer o diff no linha do comando, acho ele mais prático. Mas o Resolução 3 não é ruim	Sim	Resolução 3	Não	Sim	Resolução 3	Não	Resolução 3	Não	Não	Sim	Sim, acho que poderia facilitar a identificação de Sim	
10/01/2025 16:45:24	Participante_29		Ll no termos e concordo	5 anos e mais	Java, Javascript	Sim	Resolução 2	É demonstrando identificar melhor que não movimentos para um outro ponto do arquivo.	Não	A imagem de Resolução 1	Não	Resolução 4	Quando não tem muitas sentenças movidas, a versão linha por linha é mais vantajosa	Não	Resolução 3	Não	Sim	Resolução 3	Não	Resolução 3	Não	Não	Sim	Sim, Quidoro	
10/01/2025 17:15:09	Participante_30		Ll no termos e concordo	5 anos e mais	Java, C, Javascript	Não	Resolução 2	Acho mais fácil de identificar o que foi alterado	Não	A imagem de Resolução 1	Não	Resolução 4	Deixa ser ou como deixaram mais claro e mudança	Não	Os dois resultados são equivalentes	Não	Os dois resultados são equivalentes	Não	Resolução 3	Não	Não	Sim	Acredito que pessoa acolhe provavelmente me mais Tabea	
11/01/2025 22:11:50	Participante_31		Ll no termos e concordo	5 anos e mais	Java, C	Não	Resolução 2	Ao primeiro caso o acostumei muito pelo diff do linux, não marcando como mudas ainda(mas mantendo, enquanto no segundo caso(o 2) a cabeça que a mudança foi feita e o que foi alterado ficou mais evidente	Não	A imagem de Resolução 1	Não	Resolução 3	Ao contrário 3 o possível ver especificamente o que foi alterado no código, em vez de deixar a linha inteira	Não	Resolução 3	Não	Sim	Resolução 3	Não	Resolução 3	Não	Não	Sim	Uma ferramenta para comparação de código varios Sim	
10/01/2025 08:41:17	Participante_32		Ll no termos e concordo	5 anos e mais	Java, C, Ruby, no Rails	Não	Resolução 2	Acho mais fácil de identificar o que foi alterado, simples. Como não tanto costume de comparar sequencias de código, acho difícil de identificar o que foi alterado no resultado 1 e 3 quando ocorre muitas coisas	Não	A imagem de Resolução 1	Não	Resolução 3	Menos perguntas sem respostas. Sem certeza real, sem muitas ampuxos cada um em muitas modifica	Não	Resolução 3	Não	Sim	Resolução 3	Não	Resolução 3	Não	Não	Sim	Acho que cada desenvolvedor vai ser endeusado e Não	
10/01/2025 08:42:15	Participante_33		Ll no termos e concordo	5 anos e mais	Java, C#, JavaScript, Ruby	Sim	Resolução 2	Acho que a recontextualização dos arquivos, lados lado, fica visualmente mais fácil identificar o que foi removido, inserido, alterado ou movido. Sem muitas mudanças, acho que a recontextização do código acaba a comparação lado a lado	Não	A imagem de Resolução 1	Sim	Os dois resultados são equivalentes	Como as alterações foram pontuais e mínimas, ambas as formas do comparar atendem bem para este tipo de tarefa.	Não	Resolução 3	Não	Sim	Resolução 3	Não	Resolução 3	Não	Não	Sim	Sim	O uso dessas ferramentas já faz parte da minha rot Sim	
10/01/2025 08:42:01	Participante_34		Ll no termos e concordo	2 anos - 5 anos	vdi	Não	Resolução 2	o resultado seleciona(e mas) linhas e claro.	Não	A imagem de Resolução 1	Sim	Resolução 3	visualmente mais explicativo e respectiva.	Não	A imagem de Resolução 3	Não	Sim	Resolução 3	Não	Resolução 3	Não	Não	Sim	eu mudanças são claras e detalhadas, sim Sim, c Sim	
10/01/2025 08:50:08	Participante_35		Ll no termos e concordo	2 anos - 5 anos	Java, C#, Python	Não	Resolução 1	As cores e o modo de visualizacao do primeiro resultado são mais atrativos	Não	A imagem de Resolução 2	Sim	Resolução 3	Ressaltamos o quanto da cor e atrativo visual	Não	A imagem de Resolução 4	Sim	Sim	Resolução 3	Não	Resolução 3	Não	Não	Sim	Mesma justificativa. A medida que a ferramenta de Sim	
10/01/2025 08:07:55	Participante_36		Ll no termos e concordo	5 anos e mais	Java, C#, JavaScript	Não	Resolução 2	o 2 ficou mais claro e mais fácil de identificar o que foi alterado.	Não	A imagem de Resolução 1	Não	Resolução 4	A justificativa é parecida com a anterior, ou seja,entendo tanto nesse quanto no anterior, a diferenciação do resultado 3	Não	Resolução 3	Não	Sim	Resolução 3	Não	Resolução 3	Não	Não	Sim	absolutamente acredito que não, mas ainda carece Sim	
10/01/2025 09:05:29	Participante_37		Ll no termos e concordo	5 anos e mais	Java, C#, Javascript, Kotlin	Sim	Resolução 1	Familiaridade com o gitlab	Não	A imagem de Resolução 1	Não	Resolução 4	Familiaridade com o gitlab	Não	Resolução 3	Não	Sim	Resolução 3	Não	Resolução 3	Não	Não	Sim	Ao meu caso não, porém acredito q sim acredito Sim	
10/01/2025 09:15:29	Participante_38		Ll no termos e concordo	2 anos - 5 anos	Java, C	Não	Resolução 2	Acho mais fácil de identificar o que foi alterado	Não	A imagem de Resolução 1	Não	Resolução 3	Nesse caso, não vejo uma diferença muito grande. Prefiro ainda ter um argumento de diff sintático	Não	Resolução 3	Não	Sim	Resolução 3	Não	Resolução 3	Não	Não	Sim	O resultado 3 ainda ser mesmo suporte para mim, pois, fica mais fácil de entender que algo de antigo código ainda é aproveitado	Sim	Resolução 3	Não	Não	Sim	Sim, adoraria colaborar e analisar de qual requisito p Sim	
10/01/2025 13:56:50	Participante_39		Ll no termos e concordo	5 anos e mais	Java, PHP, Python, JavaScript	Não	Resolução 2	O resultado 2 apresenta a vantagem de indicar trechos de código que não se baseiam em uma mudança específica, mas que foram alterados em uma das vezes, nos facilitou a analisar para onde o código está caminhando. Ao ver o trecho de código que foi movido e quais mudanças foram feitas, nos ajudou a aprender(ou ao melhorar)de código anterior.	Não	A imagem de Resolução 1	Não	Resolução 3	A facilidade de identificar o que de fato foi alterado, sem a necessidade de analisar todos o linha de código.	Não	Resolução 3	Não	Sim	Resolução 3	Não	Resolução 3	Não	Não	Sim	A simplicidade em identificar as mudanças que, de	Sim	Resolução 3	Não	Não	Sim	Certo que o uso do diff sintático poderia simplificar o Sim	
10/01/2025 14:56:01	Participante_40		Ll no termos e concordo	2 anos - 5 anos	C#	Não	Resolução 2	Melhor visualização, comparação lado a lado	Não	A imagem de Resolução 1	Não	Resolução 3	Comparação lado a lado não mais fácil de visualizar, ou mesmo ajuda	Não	A imagem de Resolução 3	Não	Sim	Resolução 3	Não	Resolução 3	Não	Não	Sim	Por ramos muitos pequenos, a alternação é fácil de ser visualizada nos dois exemplos	Não	Resolução 3	Não	Não	Sim	Certo que facilitaria a verificação das alterações. Sim	
10/01/2025 14:40:01	Participante_41		Ll no termos e concordo	5 anos e mais	Java, C,javascript, go,,python, etc	Não	Resolução 2	É mais fácil de entender o que mudou	Não	A imagem de Resolução 1	Não	Os dois resultados são equivalentes	Para esse tipo de mudança, a interface web de git (github, gitlab, etc) já destaca a palavra que alterou, e não vai a linha inteira	Não	Resolução 3	Não	Sim	Resolução 3	Não	Resolução 3	Não	Não	Sim	Para esse tipo de mudança, a interface web do git,	Não	Resolução 3	Não	Não	Sim	Poderia facilitar o entendimento do diff em casos e Sim	
10/01/2025 19:13:07	Participante_42		Ll no termos e concordo	5 anos e mais	Java, C++, Basic, Python, Perl, IB,Delphi, Basic X	Não	Resolução 1	Legibilidade de ler	Não	A imagem de Resolução 1	Não	Resolução 3	Nesse caso a visualização do 3 é mais clara.	Não	Resolução 3	Não	Sim	Resolução 3	Não	Resolução 3	Não	Não	Sim	Claramente, o cinema não carregamos de 2	Não	Resolução 3	Não	Não	Sim	Acho que poderia ajudar em poucos casos, sim, e Tabea	
10/01/2025 19:50:10	Participante_43		Ll no termos e concordo	5 anos e mais	Java, Javascript, Typescript, Python, Pkg	Não	Resolução 2	No resultado 3 fica mais claro para comparar o que o código fazia antes, para o que ele depois. Ao ver o resultado 2 fica difícil de ver o que tenho recontextualizar do código	Não	A imagem de Resolução 1	Não	Resolução 3	O Resolução 3 mostra explicitamente o que foi modificado, já o 2 está mostrando a linha toda	Não	Resolução 3	Não	Sim	Resolução 3	Não	Resolução 3	Não	Não	Sim	Resolução 3 mostra que é claramente antes e depois	Não	Resolução 3	Não	Não	Sim	Sim resolução 3 mostra que claramente que foi apar	Sim	Resolução 3	Não	Não	Sim	Quando fazem pull requests no meu trabalho, apro Sim	
10/01/2025 20:51:15	Participante_44		Ll no termos e concordo	5 anos e mais	Java, C, JS, TS, Python, Ruby, Rust, R	Não	Resolução 2	Acho que o mais costume de fazer não o caso tempo a em diferentes contextos (desenvolvimento local vs Pull Request	Não	A imagem de Resolução 1	Não	Os dois resultados são equivalentes	Acho que o mais costume de fazer não o caso tempo a em diferentes contextos (desenvolvimento local vs Pull Request	Não	Os dois resultados são equivalentes	Não	Os dois resultados são equivalentes	Não	Resolução 3	Não	Resolução 3	Não	Não	Sim	Quando o time e simples o suficiente, positivo ver as	Não	Resolução 3	Não	Não	Sim	Kute me sinto "perguntado" por falar com diffs, mas e Tabea	
10/01/2025 20:20:09	Participante_45		Ll no termos e concordo	5 anos e mais	Java, python, js, scala	Sim	Resolução 2	As mudanças são mais claras com o texto anterior/atual	Não	A imagem de Resolução 1	Não	Resolução 3	A mudança do diagrm é ainda mais determinística	Não	Resolução 3	Não	Sim	Resolução 3	Não	Resolução 3	Não	Não	Sim	Sirjuan, ser automatizado e prático Sim	
10/01/2025 22:24:02	Participante_46		Ll no termos e concordo	5 anos e mais	Java, C, C++, Python, asm,vi,vi	Sim	Resolução 1	Ao ler o texto, sem antes sentagem técnica que é mais fácil de ser percebido o linguagem com ferramentas (juntamente, Nesse caso, ao ler o resultado 2 ficou difícil de ver as palavras gerou um resultado melhor. Acredito que ter duas versões de arquivo ajudou muito a ver o que foi feito e a compreender as mudanças com menos cognitivo.	Não	A imagem de Resolução 2	Não	Resolução 3	Acho ser a alternativa em contexto local é mais fácil de que mudar o foco.	Não	A imagem de Resolução 3	Não	Sim	Resolução 3	Não	Resolução 3	Não	Não	Sim	Os dois são simples de entender.	Não	Resolução 3	Não	Não	Sim	Talvez, ser bem integrado diretamente nos forkers Tabea	
10/01/2025 17:47:09	Participante_47		Ll no termos e concordo	5 anos e mais	Java, PHP, JavaScript, C++	Sim	Resolução 2	Ter as versões do arquivo separadas em dois lados de tela facilita a compreensão do código como um todo e das alterac	Não	A imagem de Resolução 1	Não	Resolução 3	Ter as versões do arquivo separadas em dois lados de tela facilita a compreensão do código como um todo e das alterac	Não	A imagem de Resolução 3	Não	Sim	Resolução 3	Não	Resolução 3	Não	Não	Sim	Acredito que a separação em lados da tela facilita a compreensão.	Não	Resolução 3	Não	Não	Sim	Acredito que a separação em lados da tela facilita a Sim	
20/01/2025 19:27:55	Participante_48		Ll no termos e concordo	5 anos e mais	Java, Ruby, Clojure, JavaScript	Não	Resolução 2	Maior facilidade de entendimento do que foi alterado no código	Não	A imagem de Resolução 1	Sim	Resolução 3	Maior facilidade de entendimento do que foi alterado no código	Não	A imagem de Resolução 3	Não	Sim	Resolução 3	Não	Resolução 3	Não	Não	Sim	Acredito que poderia impactar sim, principalmente Tabea	
20/01/2025 17:50:44	Participante_49		Ll no termos e concordo	5 anos e mais	Java, C,C#, Javascript	Não	Resolução 2	Legibilidade e maior clareza de informações ao receber as alterações nas versões do código	Não	A imagem de Resolução 1	Não	Resolução 3	Clareza nas mudanças e menos poluição visual	Não	Resolução 3	Não	Sim	Resolução 3	Não	Resolução 3	Não	Não	Sim	Quanto menos informação para explicar o diff, mel Sim	
21/01/2025 08:42:02	Participante_50		Ll no termos e concordo	2 anos - 5 anos	Java, C, Profissionalmente, apenas Python, C++,ca	Sim	Os dois resultados são equivalentes	Nenhum dos dois oferece um ganho significativo	Não	A imagem de Resolução 1	Não	Resolução 3	Acho que o 3 está um pouco mais claro, em relação aos separados	Não	A imagem de Resolução 3	Não	Sim	Resolução 3	Não	Resolução 3	Não	Não	Sim	Sim, entendem mais claro	Não	Resolução 3	Não	Não	Sim	Sim. Mas o VS Code já oferece extensões pra isso Tabea	
21/01/2025 19:27:14	Participante_51		Ll no termos e concordo	5 anos e mais	Java, C#, JS, Python	Não	Resolução 2	Nenhum dos dois oferece um ganho significativo. O segundo resultado traz informações mais estruturas. O fato de ter/adicionar a recontextização do código, ao meu ver, ajuda a ver as mudanças de uma ação (ação) acontecendo em alterações(mas linha por linha), ao invés de ver as mudanças de forma isolada. Ao ver as mudanças de arquivo/linear diferença entre versões	A imagem de Resolução 1	Não	Resolução 3	Mesma lógica da compreensão anterior, não acrescentar qualquer mudança a delação e adição, e ainda trazer semantiamente, o que acontece de diferença no código e muito relevante	A imagem de Resolução 3	Não	Os dois resultados são equivalentes	Não	Resolução 3	Não	Resolução 3	Não	Não	Sim	Resultado mais relevante, explicando o que acontec	Não	Resolução 3	Não	Não	Sim	Sim, acredito que teria um impacto positivo em Sim	
29/01/2025 08:41:55	Participante_52		Ll no termos e concordo	5 anos e mais	Basic	Não	Resolução 2	Equivalentes	Não	A imagem de Resolução 1	Não	Os dois resultados são equivalentes	Cada que o ramo/ poss desenvolvimento é criado que qualquer diferenciação	Não	A imagem de Resolução 3	Não	Os dois resultados são equivalentes	Não	Os dois resultados são equivalentes	Não	Que deus proteja o Santos todos para que fiquem	Não	Os dois resultados são equivalentes	Não	Não	Sim	Sabe que não é um diff cu, nada que eu dominar o Não	
29/01/2025 11:20:09	Participante_53		Ll no termos e concordo	2 anos - 5 anos	Java	Sim	Resolução 2	Em relação ao código o meio convem que o diff mexeu não tem a ideia de ser mesmo. Ao ver o resultado 2 fica difícil de ver o que foi confundido mudas vezes o desenvolvedor precisa ter a ideia de ver o que foi feito e ao abordagem mais linguo o resultado 3 consegue visalisar a metade do código que garantemos que o que foi feito e o que foi alterado.	Não	A imagem de Resolução 1	Sim	Resolução 3	Pelo mesmo motivo da comparação de resultado 1 como 2.	Não	Resolução 3	Não	Sim	Resolução 3	Não	Resolução 3	Não	Não	Sim	Mesmo motivo dos questionamentos anteriores.	Não	Resolução 3	Não	Não	Sim	Sim, afinal, jurno utilizo dele através do interface do Sim	
29/01/2025 18:01:03	Participante_54		6 meses - 1 ano	Javascript	Não	Não	Resolução 1	Apenas ter a ideia de qt diff em mais pois mesmo a leitura ficou que ocorreu a mudança não acredito que influenciaria a	Não	A imagem de Resolução 1	Não	Os dois resultados são equivalentes	Apenas ter a ideia de qt diff em mais pois mesmo a leitura ficou que ocorreu a mudança não acredito que influenciaria a	Não	Resolução 3	Não	Os dois resultados são equivalentes	Não	Resolução 3	Não	Não	Sim	Não sei dizer pois não tenho clara ocorrencia da mudança, pelo fato do código de cada não ainda estar relacionado com o contexto	Não	A imagem de Resolução 3	Não	Não	Sim	Pelo fato de ser uma mudança sutil e de uma linha	Não	Os dois resultados são equivalentes	Não	Não	Sim	Sim