Impacto das Inteligências Artificiais Generativas no Ciclo de Vida do Desenvolvimento de Software: um estudo sobre as percepções dos profissionais em empresas de Paranavaí

Claudir Fantuci Junior[1], **Ayslan Trevizan Possebom**[1]

[1]Instituto Federal do Paraná – Campus Paranavaí
Paranavaí – PR – Brasil

claudirfantucijunior@gmail.com, ayslan.possebom@ifpr.edu.br

Abstract. *This paper investigates the adoption and perceived impact of Generative Artificial Intelligence (GenAI) in software development in Paranavaí, Brazil. A mixed-method survey with 32 software professionals examined use across requirements, design, implementation, and testing. Implementation showed the strongest adoption, with 90.6% reporting frequent use. Perceived gains were concentrated on speed and productivity, while context, security, and reliability were among the concerns reported by participants. Claude and ChatGPT were the most frequently reported tools across stages. In addition, 90.6% of respondents reported reviewing AI-generated code always or most of the time. The findings indicate that GenAI is mainly perceived as a productivity accelerator whose effective use still depends on human validation and organizational governance.*

Resumo. *Este artigo investiga a adoção e o impacto percebido de Inteligências Artificiais Generativas (IAGs) no desenvolvimento de software em Paranavaí. Foi realizado um survey de método misto com 32 profissionais, abrangendo requisitos, design, implementação e testes. A implementação apresentou a adoção mais intensa, com 90,6% de uso frequente. Os principais ganhos percebidos concentram-se em velocidade e produtividade; contexto, segurança e confiabilidade estiveram entre as preocupações relatadas. Claude e ChatGPT foram as ferramentas mais citadas nas quatro etapas. Além disso, 90,6% dos respondentes afirmaram revisar o código gerado sempre ou na maioria das vezes. Os resultados indicam que as IAGs são percebidas principalmente como aceleradoras do trabalho, ainda dependentes de validação humana e governança organizacional.*

1. Introdução

A popularização das Inteligências Artificiais Generativas (IAGs), especialmente de ferramentas baseadas em Grandes Modelos de Linguagem (LLMs), ampliou o uso de Inteligência Artificial em atividades de Engenharia de Software. Essas ferramentas passaram a apoiar produção e análise de código, documentação, testes, requisitos e outras tarefas do ciclo de vida do desenvolvimento de software (do inglês *Software Development Life Cycle* – SDLC) [Hou et al. 2024, Nguyen-Duc et al. 2025].

A literatura recente apresenta evidências de ganhos de produtividade, mas também de limitações relacionadas à correção, segurança e adequação ao contexto. Liang,

Yang e Myers [Liang et al. 2024], em levantamento com 410 desenvolvedores, identificaram rapidez, redução de esforço mecânico e apoio à escrita de código entre os benefícios de assistentes de programação. Em contrapartida, os participantes também relataram dificuldades para obter resultados aderentes a requisitos específicos. Dakhel et al. [Dakhel et al. 2023] observaram que o GitHub Copilot pode produzir soluções eficientes, porém também gera respostas incorretas ou não reprodutíveis. Pearce et al. [Pearce et al. 2025] demonstraram adicionalmente que sugestões de código podem conter vulnerabilidades em cenários específicos de segurança.

O impacto dessas tecnologias não é exclusivamente técnico. Russo [Russo 2024] destaca que a adoção depende da compatibilidade entre a ferramenta e os fluxos de trabalho da organização. Assim, políticas de uso, capacitação, revisão e segurança tornam-se relevantes para transformar uso individual em adoção organizacional estruturada.

Embora estudos internacionais descrevam diferentes formas de utilização de IAGs, contextos regionais permanecem menos documentados. Este trabalho investiga profissionais vinculados ao ecossistema de software de Paranavaí – PR e busca responder à seguinte questão de pesquisa: *como as IAGs são utilizadas e percebidas nas diferentes etapas do desenvolvimento de software no contexto investigado?*

O objetivo é analisar a frequência de uso, os principais benefícios percebidos, as barreiras técnicas e aspectos organizacionais associados à adoção. A contribuição é descritiva: o estudo não busca demonstrar causalidade entre uso de IA e produtividade, mas produzir evidências empíricas locais que possam ser comparadas ao conhecimento existente na literatura.

O restante deste artigo está organizado da seguinte forma. A Seção 2 apresenta a fundamentação e os trabalhos relacionados. A Seção 3 descreve a metodologia. A Seção 4 apresenta e discute os resultados. A Seção 5 apresenta as limitações do estudo, e a Seção 6 apresenta as conclusões.

2. Fundamentação e Trabalhos Relacionados

O SDLC reúne atividades de concepção, especificação, projeto, implementação, teste, entrega e manutenção de sistemas. Mesmo antes da popularização dos LLMs, técnicas de IA já eram investigadas em atividades como geração de código, teste, manutenção e melhoria de processos [Barenkamp et al. 2020]. Com as IAGs, o conjunto de aplicações aumentou consideravelmente.

Hou et al. [Hou et al. 2024], em uma revisão sistemática de 395 estudos, organizaram aplicações de LLMs em seis grupos de atividades: requisitos, design, desenvolvimento, garantia da qualidade, manutenção e gestão. Nguyen-Duc et al. [Nguyen-Duc et al. 2025] também defendem uma visão integrada do SDLC e observam que a literatura se concentra especialmente em implementação, garantia da qualidade e manutenção, ao mesmo tempo em que áreas como requisitos e aspectos organizacionais ainda apresentam questões abertas.

Na implementação, assistentes de programação oferecem autocompletar, geração de funções, refatoração, explicação e documentação de código. Liang et al. [Liang et al. 2024] observaram que desenvolvedores valorizam principalmente rapidez e redução de tarefas mecânicas, mas também encontram dificuldades de controle sobre a

saída produzida. Dakhel et al. [Dakhel et al. 2023] reforçam essa dualidade ao mostrar que sugestões úteis coexistem com soluções incorretas, tornando a experiência do desenvolvedor relevante para avaliar os resultados.

A segurança representa outra limitação. Pearce et al. [Pearce et al. 2025] avaliaram código sugerido pelo GitHub Copilot em cenários relacionados a vulnerabilidades conhecidas e identificaram vulnerabilidades em aproximadamente 40% dos 1.689 programas gerados em 89 cenários do experimento. Esse resultado não representa uma taxa universal de falhas em código gerado por IA, mas demonstra que a geração automática não substitui práticas de revisão e segurança.

Em testes, a IA pode apoiar geração de casos de teste, detecção de defeitos, análise de cobertura e priorização. Escalante-Viteri e Mauricio [Escalante-Viteri e Mauricio 2025], em revisão sistemática de uma década de pesquisas, identificam avanço das aplicações de IA em testes, mas também apontam desafios de confiabilidade, avaliação e aproximação com ambientes industriais.

Por fim, a adoção precisa ser analisada também no nível organizacional. Russo [Russo 2024] encontrou evidências de que a compatibilidade com os processos existentes influencia a adoção de IA generativa em equipes de software. Martinović e Rozić [Martinović e Rozić 2025] identificaram percepção geral positiva sobre produtividade e satisfação no uso de ferramentas de IA, embora dimensões específicas de qualidade apresentem avaliações mais moderadas. Esses resultados motivam a análise conjunta de velocidade, qualidade, confiança, revisão humana e governança realizada neste trabalho.

3. Metodologia

Esta pesquisa caracteriza-se como descritiva e utiliza levantamento (*survey*) de método misto. O instrumento de coleta foi um questionário com questões fechadas, escalas Likert de cinco pontos e duas questões abertas. A coleta ocorreu por meio do Google Forms entre 16 de agosto e 3 de setembro de 2026 e resultou em 32 respostas.

A amostra é não probabilística, por conveniência. Foram contatadas diretamente, por e-mail, sete empresas de tecnologia e desenvolvimento de sistemas sediadas em Paranavaí – PR, às quais foi solicitado o encaminhamento do questionário aos desenvolvedores e suas equipes. A partir desses contatos, o formulário foi enviado a aproximadamente 54 profissionais envolvidos no desenvolvimento de software, resultando em 32 respostas válidas, o que corresponde a uma taxa de resposta estimada de 59%. A participação foi voluntária. O Google Forms estava configurado para não coletar e-mails automaticamente, e o questionário não solicitou nomes nem identificadores das empresas. Também não foi coletado um identificador organizacional; portanto, diferentes participantes podem atuar na mesma empresa. A unidade de análise é o respondente, e não a empresa.

Na abertura do formulário, os participantes foram informados sobre o objetivo da pesquisa, o caráter voluntário da participação, o anonimato declarado e o uso das respostas exclusivamente para fins acadêmicos. Considerando o caráter de pesquisa de opinião com participantes não identificados, os autores enquadram o estudo na hipótese de dispensa de registro e avaliação pelo sistema CEP/CONEP prevista no art. 1º, parágrafo único, inciso I, da Resolução CNS nº 510/2016. A finalidade e o tratamento informados aos participantes observaram a Lei Geral de Proteção de Dados Pessoais (Lei nº 13.709/2018) [Brasil. Conselho Nacional de Saúde 2016, Brasil 2018].

O instrumento foi organizado em blocos relativos ao perfil profissional e a quatro etapas do SDLC: levantamento e análise de requisitos, design e arquitetura, implementação e testes e qualidade. Em cada etapa foram investigados frequência de utilização, atividades apoiadas, impacto percebido e barreiras. Implantação/manutenção e gestão de projetos foram consideradas em uma questão geral sobre as etapas percebidas como mais beneficiadas, mas não receberam blocos específicos equivalentes.

As escalas de impacto utilizaram cinco categorias, de "Muito negativo" a "Muito positivo". Para síntese descritiva, essas categorias foram codificadas de 1 a 5. As dimensões avaliadas foram definidas diretamente no questionário: em requisitos, qualidade dos requisitos, velocidade/agilidade, redução de ambiguidades e comunicação com o cliente; em design, qualidade das decisões arquiteturais, velocidade de definição e qualidade da documentação; em implementação, velocidade, qualidade do código, redução de erros/bugs, aprendizado e confiança; em testes, cobertura, velocidade de criação, qualidade/confiabilidade e identificação de bugs; e, na avaliação geral, produtividade, qualidade das entregas, satisfação, cultura da equipe e competitividade.

As médias foram calculadas para cada dimensão utilizando somente as respostas efetivamente preenchidas naquele item, com respostas em branco desconsideradas e sem aplicação de filtro adicional pela frequência declarada de uso na etapa. Por esse motivo, o número de respostas válidas varia entre os blocos. Essas médias são empregadas apenas como síntese descritiva, e não como estatística inferencial. Nas questões de múltipla seleção, um mesmo participante podia indicar mais de uma alternativa; por isso, os resultados dessas questões são apresentados em número de indicações, e não como categorias mutuamente exclusivas.

As duas questões abertas abordaram limitações na geração de testes e identificação de bugs e desafios para ampliação do uso de IAGs. As respostas foram tratadas por análise de conteúdo inspirada em Bardin [Bardin 2011], organizando os relatos em temas exploratórios e comparando-os aos resultados quantitativos, sem pretensão de generalização regional.

Dos 32 participantes, 21 (65,6%) identificaram-se como desenvolvedores ou programadores e cinco (15,6%) como analistas de sistemas. Os demais atuavam em gestão, liderança técnica, qualidade, direção ou propriedade. Quanto ao nível declarado de uso de IAGs, 22 (68,8%) classificaram-se como avançados, nove (28,1%) como intermediários e um (3,1%) como iniciante. Em relação à experiência profissional, cinco (15,6%) tinham menos de um ano na área, 11 (34,4%) de um a dois anos, oito (25,0%) de três a cinco anos, quatro (12,5%) de seis a dez anos e quatro (12,5%) mais de dez anos. A faixa etária mais frequente foi de 21 a 30 anos (21; 65,6%), e 29 participantes (90,6%) estavam cursando ou haviam concluído graduação em TI. Sobre o tempo de atuação das empresas, 31 responderam: 14 (45,2%) indicaram de 11 a 20 anos, 12 (38,7%) mais de 20 anos, quatro (12,9%) de dois a dez anos e um não soube informar. O questionário não coletou o número de funcionários; portanto, não permite caracterizar o porte das empresas. A concentração de usuários intermediários e avançados deve ser considerada na interpretação dos resultados.

4. Resultados e Discussão

4.1. Adoção ao Longo das Etapas do SDLC

A Figura 1 apresenta a frequência de uso das IAGs nas quatro etapas investigadas em profundidade.

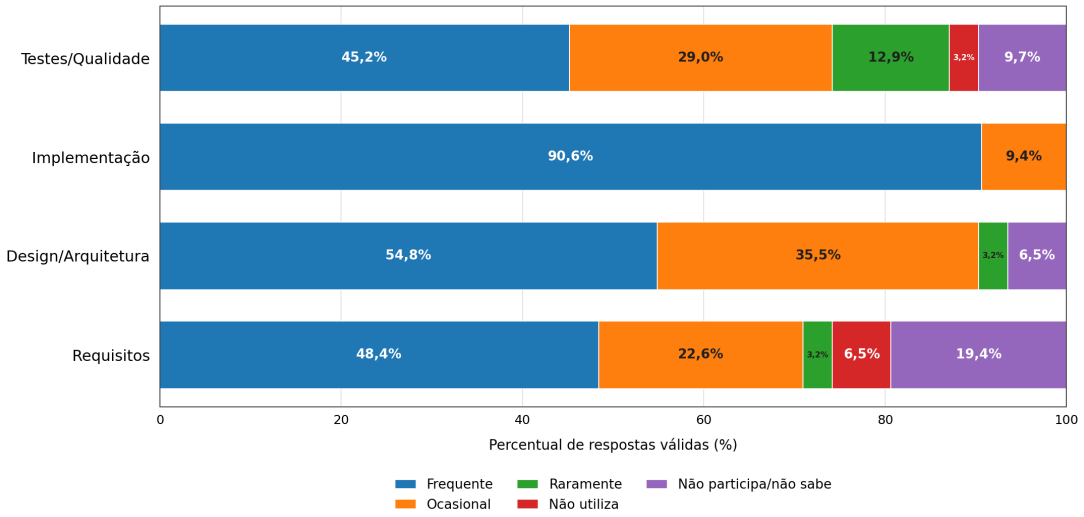


Figura 1. Frequência de utilização de IAGs nas etapas investigadas do SDLC.

A implementação é a etapa com adoção mais intensa. Todos os 32 respondentes declararam utilizar IAGs em codificação, sendo 29 (90,6%) com frequência e três (9,4%) ocasionalmente. Em design e arquitetura, 17 de 31 respostas válidas (54,8%) indicaram uso frequente e 11 (35,5%) uso ocasional. Em requisitos, 15 de 31 (48,4%) relataram uso frequente e sete (22,6%) ocasional. Em testes, 14 de 31 (45,2%) declararam uso frequente, nove (29,0%) ocasional e quatro (12,9%) raro.

Esse padrão é coerente com Nguyen-Duc et al. [Nguyen-Duc et al. 2025], que identificam forte concentração de pesquisas de IAGs em implementação e garantia da qualidade. Entretanto, os dados também mostram utilização relevante em requisitos e design, indicando que o fenômeno não se restringe à codificação.

A mesma tendência aparece na percepção de benefício. Como mostra a Figura 2, implementação/codificação foi indicada por 28 dos 30 participantes que responderam à questão, seguida por testes e qualidade (19), design e arquitetura (15), requisitos (12), implantação e manutenção (8) e gestão de projetos e equipes (6).

Nas questões de múltipla seleção sobre ferramentas, Claude foi a mais citada em cada etapa: 20 de 25 respostas válidas em requisitos, 24 de 29 em design, 26 de 32 em implementação e 21 de 28 em testes. ChatGPT apareceu em seguida, com 18, 21, 20 e 17 indicações, respectivamente. Gemini recebeu 8, 10, 6 e 5 indicações; Copilot, 5, 3, 7 e 6; e Cursor, 2, 2, 5 e 3. As respostas à opção “Outro” incluíram Codex em uma resposta por etapa e TestBooster.ai em uma resposta sobre testes. Como era possível indicar mais de uma ferramenta, as contagens não são mutuamente exclusivas.

A convergência entre frequência de uso e benefício percebido reforça a centralidade da implementação, mas também evidencia uma adoção distribuída por diferentes atividades do SDLC.

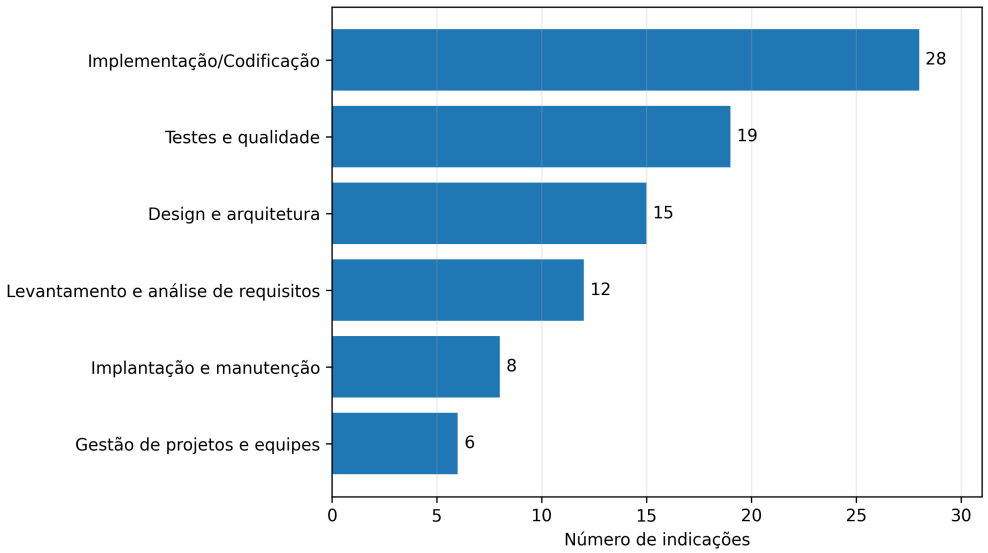


Figura 2. Etapas percebidas como mais beneficiadas pelas IAGs ($n = 30$; múltiplas respostas).

4.2. Velocidade, Qualidade e Confiança

A Tabela 1 apresenta as 21 dimensões de impacto avaliadas no questionário, o número de respostas válidas em cada item e a respectiva média. As médias utilizam a codificação de 1 (“Muito negativo”) a 5 (“Muito positivo”) e consideram apenas respostas preenchidas em cada dimensão.

O padrão mais consistente é a avaliação elevada da velocidade. Em requisitos, velocidade/agilidade alcançou média 4,72; em design, 4,48; em implementação, 4,75; e em testes, 4,33. Na percepção geral, produtividade individual obteve média 4,60, com 29 das 30 respostas válidas classificadas como positivas ou muito positivas. A qualidade das entregas apresentou média 4,37 e 28 avaliações positivas ou muito positivas. As demais dimensões também receberam médias acima do ponto neutro, incluindo qualidade do código (4,03), aprendizado profissional (3,88), satisfação no trabalho (4,40) e cultura/clima da equipe (4,13), embora a confiança no código (3,69) e a comunicação com o cliente (3,68) tenham apresentado os menores valores.

Esses resultados convergem com Martinović e Rozić [Martinović e Rozić 2025], que observaram percepção geral positiva sobre produtividade entre usuários de ferramentas baseadas em IA. No presente estudo, a percepção de ganho de velocidade aparece não apenas na implementação, mas também em requisitos, design e testes.

Ao mesmo tempo, velocidade não é acompanhada pelo mesmo nível de confiança. Na implementação, confiança no código produzido obteve média 3,69, enquanto redução de erros/bugs ficou em 3,91. Em testes, qualidade/confiabilidade dos testes gerados apresentou média 3,93, abaixo da velocidade (4,33) e da cobertura percebida (4,25). A combinação sugere que os respondentes distinguem produzir mais rápido de confiar automaticamente no artefato produzido.

Essa interpretação é reforçada pela prática de revisão: 18 participantes declararam revisar sempre o código gerado por IA e 11 afirmaram revisar na maioria das

Tabela 1. Dimensões de impacto percebido avaliadas no questionário

Contexto	Dimensão	n	Média
Requisitos	Qualidade dos requisitos levantados	25	4,00
	Velocidade/agilidade do processo	25	4,72
	Redução de ambiguidades	25	4,00
	Comunicação com o cliente	25	3,68
Design	Qualidade das decisões arquiteturais	29	4,10
	Velocidade na definição do design	29	4,48
	Qualidade da documentação gerada	29	4,17
Implementação	Velocidade de desenvolvimento	32	4,75
	Qualidade do código gerado	32	4,03
	Redução de erros/bugs	32	3,91
	Aprendizado e desenvolvimento profissional	32	3,88
	Confiança no código produzido	32	3,69
Testes	Cobertura dos testes realizados	28	4,25
	Velocidade na criação de testes	27	4,33
	Qualidade/confiabilidade dos testes gerados	28	3,93
	Identificação de bugs e falhas	28	4,18
Impacto geral	Produtividade individual	30	4,60
	Qualidade das entregas	30	4,37
	Satisfação no trabalho	30	4,40
	Cultura e clima da equipe	30	4,13
	Competitividade da empresa	30	4,07

vezes. Assim, 29 de 32 (90,6%) realizam revisão frequente ou sistemática. O comportamento é compatível com as limitações de correção observadas por Dakhel et al. [Dakhel et al. 2023] e com os riscos de segurança demonstrados por Pearce et al. [Pearce et al. 2025].

4.3. Barreiras Técnicas e Limitações de Contexto

As principais barreiras relatadas na implementação são apresentadas na Figura 3.

Sugestões inadequadas ao contexto do projeto foram indicadas por 24 participantes, preocupações com segurança e vulnerabilidades por 21, código com erros ou bugs por 16, custo das ferramentas por 14 e propriedade intelectual/licenciamento por 11. Apenas um participante declarou não perceber barreiras significativas.

É relevante que a inadequação ao contexto seja a barreira mais citada justamente na etapa de maior adoção. Esse resultado mostra que uso frequente não significa ausência de limitações. O benefício parece resultar de uma relação complementar: a IA acelera a produção, enquanto o profissional fornece contexto e valida os resultados.

As respostas abertas sobre testes oferecem indícios complementares. Onze participantes responderam à questão qualitativa específica. Seus relatos mencionaram dificuldades para representar regras de negócio e o contexto do projeto, cobrir fluxos de integração, evitar falsos positivos e contemplar casos críticos. Alguns participantes observaram que a ferramenta pode gerar um teste executável sem necessariamente produzir um *bom* teste para o comportamento real do sistema. Dado o número reduzido de respostas, esses relatos são tratados como indícios exploratórios da amostra, e não como padrões

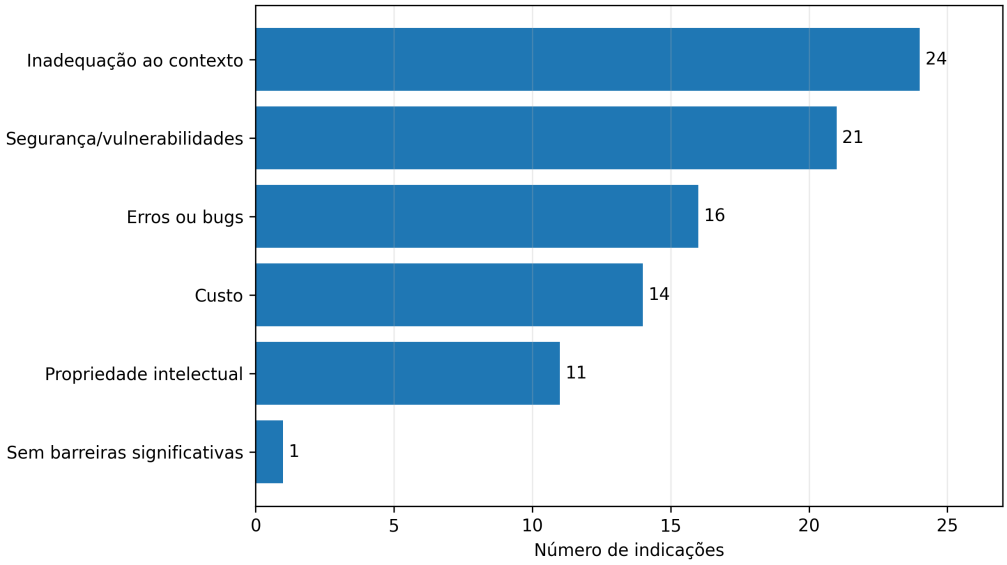


Figura 3. Barreiras percebidas na implementação ($n = 32$; múltiplas respostas).

regionais.

Esse resultado dialoga com Escalante-Viteri e Mauricio [Escalante-Viteri e Mauricio 2025], que apontam desafios de avaliação e confiabilidade na aplicação de IA a testes. Nos dados locais, a limitação não aparece como incapacidade de gerar artefatos, mas como dificuldade de representar completamente o contexto e as regras que definem se o artefato é adequado. Uma possível explicação seria a forma como o contexto e as instruções são fornecidos às ferramentas; entretanto, o questionário não mediu conhecimentos de engenharia de prompt ou de contexto. Assim, os dados não permitem atribuir essa dificuldade especificamente à ferramenta ou ao nível de domínio do profissional, sendo esse um aspecto a ser aprofundado em estudos futuros.

4.4. Governança e Adoção Organizacional

A adoção técnica não é acompanhada pelo mesmo grau de formalização organizacional. Entre 31 respostas válidas, 21 (67,7%) relataram que suas organizações possuem apenas orientações verbais ou informais para uso de IA. Cinco (16,1%) indicaram política formal e documentada e cinco (16,1%) declararam inexistência de orientação ou política, conforme a Figura 4.

A capacitação segue padrão semelhante. Entre 31 respostas válidas, 12 participantes relataram treinamentos pontuais ou esporádicos, oito indicaram incentivo informal ao aprendizado autônomo, cinco relataram treinamentos regulares, quatro informaram não haver iniciativa de treinamento e dois não souberam informar. Esses números descrevem respostas individuais sobre o ambiente de trabalho e não podem ser interpretados como quantidade de empresas, pois o instrumento não identificou organizações únicas.

Quatorze participantes responderam à questão aberta sobre desafios gerais de adoção. Os relatos mencionaram confiabilidade das respostas, segurança e confidencialidade, integração com sistemas legados, resistência cultural, dependência acrítica, necessidade de revisão, governança de dados e dificuldade de manter contexto suficiente sobre

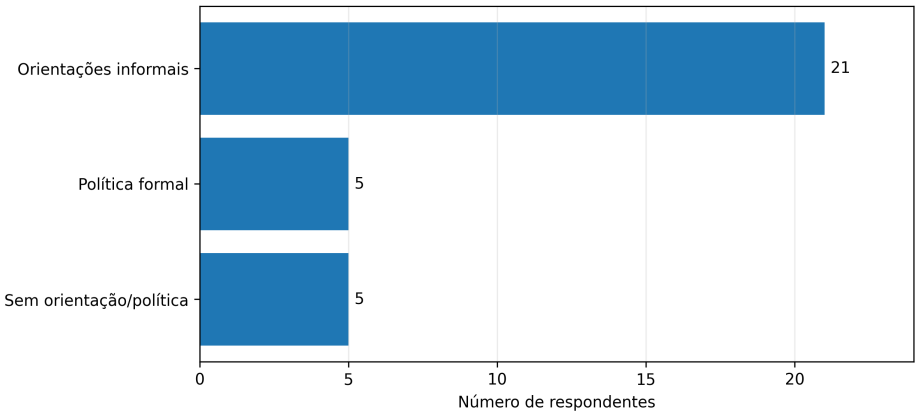


Figura 4. Formalização das orientações organizacionais para uso de IAGs ($n = 31$).

sistemas complexos. Esses elementos são indícios qualitativos das respostas obtidas e não permitem afirmar a existência de padrões regionais.

Os resultados sugerem um descompasso entre incorporação individual e institucionalização. Ferramentas já fazem parte do trabalho cotidiano de muitos participantes, mas políticas, capacitação e processos ainda são frequentemente informais. Esse resultado é consistente com Russo [Russo 2024], para quem a compatibilidade com práticas e fluxos existentes constitui fator relevante de adoção.

4.5. Síntese dos Achados

Em conjunto, os dados apontam três resultados principais. Primeiro, as IAGs apresentam adoção elevada, especialmente na implementação, mas também estão presentes em requisitos, design e testes. Segundo, o principal ganho percebido é a velocidade: as maiores médias em cada etapa estão relacionadas à agilidade, enquanto confiança, qualidade e adequação ao contexto recebem avaliações mais moderadas. Terceiro, a utilização intensa não elimina a supervisão humana nem garante maturidade organizacional: 90,6% revisam código gerado sempre ou na maioria das vezes e 67,7% das respostas válidas sobre políticas indicam apenas orientações informais.

Esses resultados sustentam a interpretação de que, no contexto investigado, as IAGs são percebidas principalmente como aceleradoras do trabalho de software, e não como substitutas da validação técnica. O valor obtido depende da capacidade dos profissionais de fornecer contexto, revisar artefatos e aplicar critérios de qualidade e segurança.

5. Limitações do Estudo

Os resultados apresentados devem ser interpretados considerando o recorte e as características metodológicas desta pesquisa. A coleta reuniu 32 respostas de profissionais vinculados ao contexto investigado, obtidas por meio de participação voluntária e amostragem por conveniência. Dessa forma, os resultados descrevem as percepções dos respondentes e não permitem estimar a adoção de IAGs entre todos os profissionais ou empresas de Paranavaí.

A composição da amostra também deve ser considerada na interpretação dos achados. Houve predominância de profissionais diretamente envolvidos com atividades de

desenvolvimento de software e de participantes que se classificaram como usuários intermediários ou avançados de ferramentas de IA Generativa. Essa característica favorece a análise de experiências de profissionais que já utilizam essas tecnologias em suas atividades, embora outras percepções possam ser observadas entre profissionais com menor contato com IAGs.

Como o questionário foi encaminhado pelos gestores das empresas, é possível que profissionais mais engajados com o uso de IAGs tenham sido convidados, ou tenham aceitado participar, com maior frequência. Esse viés de seleção pode ter contribuído para a predominância de usuários intermediários e avançados na amostra.

As informações relacionadas ao contexto das organizações foram analisadas a partir da percepção dos próprios respondentes. Como o questionário não coletou identificadores organizacionais, diferentes respostas podem representar a mesma empresa; por isso, os resultados sobre políticas, treinamentos, investimentos e práticas são apresentados no nível do respondente, sem serem interpretados como contagens de empresas. O instrumento perguntou o tempo de atuação das empresas, mas não coletou número de funcionários ou outra medida de porte, o que impede comparar organizações por tamanho. As questões abertas também tiveram participação limitada: 11 respostas sobre testes e 14 sobre desafios gerais. Seus temas devem ser lidos como indícios exploratórios, sem extrapolação para padrões regionais.

Outra característica do estudo está relacionada à natureza perceptiva das variáveis investigadas. Aspectos como produtividade, qualidade das entregas, satisfação e confiança foram avaliados com base na experiência relatada pelos participantes. Não foram utilizadas métricas objetivas de desempenho antes e depois da adoção das ferramentas. Assim, os resultados representam o impacto percebido das IAGs, que constitui justamente o foco definido para esta pesquisa.

Além disso, as etapas de levantamento e análise de requisitos, design e arquitetura, implementação e testes foram investigadas de maneira mais detalhada no questionário. Implantação, manutenção e gestão de projetos foram contempladas na avaliação geral sobre as etapas mais beneficiadas pelas IAGs, oferecendo uma visão complementar, porém menos detalhada dessas atividades.

Essas características delimitam o alcance das conclusões e, ao mesmo tempo, indicam possibilidades para investigações posteriores. Estudos futuros podem ampliar o número e a diversidade de participantes, aprofundar etapas menos exploradas do SDLC, coletar medidas organizacionais como porte e incorporar indicadores objetivos de produtividade e qualidade.

6. Conclusão

Este trabalho analisou como profissionais do contexto de software de Paranavaí utilizam e percebem IAGs ao longo do desenvolvimento de software. Os resultados indicam adoção em diferentes etapas do SDLC, com maior intensidade na implementação e presença também em requisitos, design e testes.

As conclusões descrevem as percepções dos 32 respondentes desta amostra por conveniência e não devem ser generalizadas para todo o ecossistema regional.

O principal benefício percebido está relacionado à velocidade e à produtividade.

Em contrapartida, confiança, adequação ao contexto, segurança e confiabilidade aparecem de forma mais moderada ou como barreiras, o que reforça a importância da revisão técnica. Assim, os achados sugerem que as IAGs atuam, no contexto investigado, como ferramentas de apoio à execução das atividades, sem eliminar a necessidade de conhecimento profissional e validação humana.

No âmbito organizacional, a predominância de orientações informais indica que a incorporação prática dessas ferramentas pode ocorrer antes da consolidação de políticas, capacitação e mecanismos mais estruturados de governança.

Como contribuição, o estudo apresenta evidências empíricas sobre a utilização de IAGs em um contexto regional ainda pouco explorado na literatura, relacionando aspectos técnicos do SDLC a elementos organizacionais como capacitação, políticas de uso e percepção de benefícios e barreiras. Os resultados também reforçam uma característica recorrente na literatura recente: os ganhos proporcionados pelas IAGs estão associados à capacidade de combiná-las com conhecimento técnico, revisão crítica e práticas adequadas de utilização.

Como continuidade, estudos futuros podem ampliar e diversificar a amostra, aprofundar etapas menos exploradas do SDLC e combinar medidas de percepção com indicadores objetivos de produtividade, qualidade e segurança. Também é pertinente investigar de forma específica como práticas de engenharia de prompt e de contexto influenciam a qualidade dos resultados produzidos pelas ferramentas.

Referências

Bardin, L. (2011). *Análise de Conteúdo*. Edições 70, São Paulo. Tradução de Luís Antero Reto e Augusto Pinheiro. 279 p.

Barenkamp, M., Rebstadt, J., e Thomas, O. (2020). Applications of AI in classical software engineering. *AI Perspectives*, 2(1):1–19.

Brasil (2018). Lei nº 13.709, de 14 de agosto de 2018. Diário Oficial da União. Lei Geral de Proteção de Dados Pessoais. Disponível em: `https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709compilado.htm`. Acesso em: 18 out. 2026.

Brasil. Conselho Nacional de Saúde (2016). Resolução nº 510, de 7 de abril de 2016. Diário Oficial da União. Disponível em: `https://www.gov.br/conselho-nacional-de-saude/pt-br/acesso-a-informacao/atos-normativos/resolucoes/2016/resolucao-no-510.pdf`. Acesso em: 18 out. 2026.

Dakhel, A. M., Majdinasab, V., Nikanjam, A., Khomh, F., Desmarais, M. C., e Jiang, Z. M. (2023). GitHub Copilot AI pair programmer: Asset or liability? *Journal of Systems and Software*, 203:111734.

Escalante-Viteri, A. e Mauricio, D. (2025). Artificial intelligence in software testing: A systematic review of a decade of evolution and taxonomy. *Algorithms*, 18(11):717.

Hou, X., Zhao, Y., Liu, Y., Yang, Z., Wang, K., Li, L., Luo, X., Lo, D., Grundy, J., e Wang, H. (2024). Large language models for software engineering: A systematic literature review. *ACM Transactions on Software Engineering and Methodology*, 33(8):1–79.

Liang, J. T., Yang, C., e Myers, B. A. (2024). A large-scale survey on the usability of AI programming assistants: Successes and challenges. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, ICSE '24, pages 616–628, New York, NY, USA. Association for Computing Machinery.

Martinović, B. e Rozić, R. (2025). Perceived impact of AI-based tooling on software development code quality. *SN Computer Science*, 6(1):63.

Nguyen-Duc, A., Cabrero-Daniel, B., Felderer, M., Mikkelsen, E., Melegati, J., Guerra, E., e Abrahamsson, P. (2025). Generative artificial intelligence for software engineering—a research agenda. *Software: Practice and Experience*, 55(11):1806–1843.

Pearce, H., Ahmad, B., Tan, B., Dolan-Gavitt, B., e Karri, R. (2025). Asleep at the keyboard? assessing the security of GitHub Copilot's code contributions. *Communications of the ACM*, 68(2):96–105.

Russo, D. (2024). Navigating the complexity of generative AI adoption in software engineering. *ACM Transactions on Software Engineering and Methodology*, 33(5):1–50.