

O imposto dos tokens: por que os agentes de IA autônomos estão corroendo justamente as margens que deveriam proteger

Um guia para COOs, CXOs
e vice-presidentes de
operações



A oportunidade é real

A IA transformou o desenvolvimento de software como o conhecíamos. Ele está mais rápido, mais acessível e mais capaz do que nunca.

Os desenvolvedores conseguem criar e publicar em velocidades que pareciam impossíveis há apenas um ano. 84% dos desenvolvedores já recorrem à IA para acelerar o próprio trabalho ([Stack Overflow Developer Survey, 2025](#)). Protótipos inteiros de aplicativos que antes levavam semanas estão sendo estruturados em dias. A democratização do desenvolvimento de software é real e vem acelerando. Os CIOs enxergam para onde isso está indo. Até 2030, a Gartner projeta que 75% do trabalho de TI será feito por humanos apoiados pela IA e 25% será feito apenas pela IA — sem nenhum trabalho de TI intocado pela inteligência artificial ([Gartner, novembro de 2025](#)). Não se trata de uma previsão distante. É o horizonte de planejamento que a maioria das organizações de TI corporativa já considera hoje.

O investimento se justifica. O rumo está certo. Mas entregar mais ainda não se converteu nos resultados de negócio transformadores que o conselho de administração espera. E são os líderes de TI que estão sendo chamados a explicar o porquê.

Essa distância não é um problema de tecnologia. É um problema de arquitetura. Três perguntas ocupam o centro dessa questão, e a maioria das empresas ainda não tem resposta clara para nenhuma delas.

1. ***Desenvolver com IA ficou realmente mais caro do que simplesmente contratar mais engenheiros?***
2. ***Dá para confiar no código gerado por IA para sustentar operações corporativas em escala?***
3. ***Como manter visibilidade e controle sobre aquilo que a IA está construindo e implantando?***



A questão do custo: você está pagando mais para acumular mais dívida?

A promessa do desenvolvimento assistido por IA era clara: reduzir o custo e o tempo de desenvolvimento de software. A realidade, para muitas empresas, é que a estrutura de custos mudou de maneiras que não constavam no business case original.

Escrever código não é a parte cara do desenvolvimento de software. Isso responde por cerca de 10% do ciclo de vida total do aplicativo. Os 90% restantes são revisão, testes, correções de segurança, depuração, refatoração e manutenção contínua à medida que os requisitos de negócio mudam. Os assistentes de código com IA aceleraram esses 10%. O que os dados mostram cada vez mais é que eles também encareceram os 90% restantes.

[A análise de 2025 da GitClear](#), que examinou 211 milhões de linhas de código em repositórios do Google, da Microsoft, da Meta e de organizações corporativas, identificou um aumento de oito vezes nos blocos de código duplicados em projetos assistidos por IA na comparação com dois anos antes. O churn de código – linhas revisadas ou descartadas em até duas semanas depois de escritas – dobrou entre 2020 e 2024. Nas palavras de Bill Harding, fundador da GitClear, o código gerado por IA funciona como "um cartão de crédito novinho em folha" para acumular dívida técnica: na hora, o gasto parece produtivo; a fatura chega depois. [O relatório The State of Software Delivery 2025, da Harness](#), constatou que a maioria dos desenvolvedores hoje gasta mais tempo depurando código gerado por IA do que criando novos recursos.

O imposto dos tokens piora esse quadro. Cada linha de código escrita por uma IA custa tokens para ser produzida. Diferentemente de um token gasto em uma interação com o cliente, o token gasto para gerar código de aplicativo cria um ativo que exige investimento contínuo em manutenção. Se esse ativo acumula dívida mais rápido do que gera valor, a conta se acumula na direção errada desde o primeiro dia.

E a estrutura de custos das próprias ferramentas de desenvolvimento com IA deixou de ser previsível. Em junho de 2026, o GitHub Copilot migrou para a cobrança por consumo, trocando o modelo de unidades de solicitação premium com tarifa fixa por créditos de IA baseados em tokens. A explicação do próprio GitHub admitiu que, no modelo anterior, "uma pergunta rápida no chat e uma sessão autônoma de programação de várias horas podem custar o mesmo ao usuário", e que absorver custos crescentes de inferência "já não é sustentável" ([Blog do GitHub, abril de 2026](#)). As empresas que montaram o orçamento com base na tarifa fixa do desenvolvimento com IA passaram a operar em um ambiente de custo variável, com menos visibilidade sobre o gasto por fluxo de trabalho.

A questão de fato não é se as ferramentas de desenvolvimento com IA valem a pena. Elas valem. A questão é se a abordagem atual – usar IA para gerar mais código em menos tempo – realmente reduz o custo total de desenvolvimento ou apenas antecipa a velocidade de entrega, acumulando um passivo de manutenção que vai aparecer no orçamento do próximo ano.

A questão de desempenho e segurança: dá para confiar no que a IA constrói?

Entregar mais rápido só gera valor se houver confiança de que o entregue vai rodar em escala corporativa. Nessa frente, as evidências vêm dando aos líderes de TI razões legítimas para cautela.

A análise do CodeRabbit sobre 470 pull requests reais constatou que o código gerado por IA introduz 1,7 vez mais problemas no total do que o código escrito por humanos e, especificamente, 2,74 vezes mais vulnerabilidades de segurança ([CodeRabbit State of AI vs. Human Code Generation, dezembro de 2025](#)). [Uma pesquisa do IEEE apresentada no ISSRE 2025](#) mostrou que o código gerado por IA apresenta taxas mais altas de vulnerabilidades de segurança de alto risco, mesmo produzindo resultados estruturalmente mais simples – o que sugere que o risco está na forma como a lógica é construída, e não apenas na sua complexidade.

As ferramentas de IA geram soluções para problemas imediatos e localizados. Elas não enxergam por completo a arquitetura mais ampla do seu sistema, as suas abstrações internas, os seus utilitários compartilhados nem as restrições de segurança que determinam como os dados circulam pelo seu ambiente. O resultado é um código que passa nos testes unitários e supera a revisão inicial, mas traz vulnerabilidades que só se manifestam em casos extremos ou sob carga de produção.

76% dos desenvolvedores que recorrem a ferramentas de programação com IA afirmam ter gerado, ao menos algumas vezes, código que não compreendiam por completo ([Stack Overflow, 2026](#)). Esse dado tem peso operacional real. Quando um regulador, o jurídico ou o conselho de administração pergunta ao CIO por que determinado comportamento do sistema aconteceu, responder "a IA gerou isso e o desenvolvedor não sabia bem como funcionava" não é aceitável. E, no entanto, é exatamente essa a posição implícita de responsabilização de qualquer organização que leva código gerado por IA para produção sem uma camada estrutural de governança.

As principais previsões da Gartner para 2026 incluem a estimativa de que as ações judiciais por "morte causada por IA" passarão de 2.000 no mundo todo, motivadas pela governança insuficiente sobre o que os aplicativos com IA geram ([Gartner, outubro de 2025](#)). Não se trata de riscos futuros hipotéticos. É exposição ativa para as empresas que hoje já concederam ao código gerado por IA autoridade relevante sobre fluxos de trabalho regulados ou voltados ao cliente.

A questão da confiabilidade: você sabe o que a IA está construindo?

Velocidade e segurança são preocupações operacionais. A auditabilidade é estrutural – e, para CIOs e arquitetos corporativos, talvez seja a mais difícil de incorporar depois que tudo já está pronto.

Quando um aplicativo toma uma decisão – encaminhar um caso, determinar elegibilidade, conceder acesso ou gerar um conteúdo voltado ao cliente –, quem responde por ela é a TI, não a IA que escreveu o código. Foi a área de TI que implantou o aplicativo, e foram os líderes que o aprovaram.

Em um ambiente de desenvolvimento tradicional, essa responsabilização é administrável, porque a lógica pode ser inspecionada. Quando um auditor pergunta por que determinada decisão foi tomada, o desenvolvedor rastreia o caminho de código correspondente e encontra a resposta. Em um ambiente no qual a lógica do aplicativo foi gerada por IA, essa rastreabilidade deixa de ser garantida. O código gerado por IA entrega um resultado que funciona, mas o raciocínio por trás de escolhas específicas de implementação não fica documentado como acontece na engenharia humana intencional. Quando a auditoria pergunta "por que este sistema tomou essa decisão neste caso extremo", responder "porque a IA escreveu assim" não se sustenta.

Esse é o problema de confiabilidade que mais pesa no nível corporativo. A questão não é se o código funciona bem na média. A questão é se você consegue explicar qualquer resultado específico a qualquer pessoa, em qualquer momento futuro, numa linguagem que permita a ela agir.

A lógica de aplicativo visual e declarativa resolve esse problema na estrutura. Quando as regras de negócio, as tabelas de decisão e o roteamento dos fluxos de trabalho são expressos como configuração inspecionável, em vez de código gerado, cada caminho de decisão fica rastreável sem análise forense. A lógica passa a existir em uma camada legível para as equipes de compliance, as áreas jurídicas e os stakeholders do negócio, não só para os engenheiros. A auditoria vira uma revisão de configuração, e não um exercício de arqueologia de código.

A arquitetura que responde às três questões

A mudança arquitetural que resolve custo, segurança e confiabilidade ao mesmo tempo não impõe restrição ao uso da IA. É um redirecionamento sobre em que ponto da pilha de desenvolvimento a IA é aplicada.

A IA é de fato eficaz para acelerar a geração de estruturas de aplicativo de alto nível: modelos de dados, definições de fluxo de trabalho, tabelas de decisão e lógica de roteamento expressa de forma declarativa. São artefatos que descrevem o que um aplicativo faz sem embutir a implementação em código imperativo. Eles são legíveis, auditáveis e fáceis de manter de um modo que o código gerado não é. Quando os requisitos de negócio mudam, o modelo muda e o comportamento do aplicativo acompanha, sem as alterações de código em cascata e o risco de regressão que elas trazem.

É essa a abordagem de desenvolvimento orientado a modelos na prática. A IA faz aquilo em que realmente é boa: traduzir intenção em estrutura, gerar configuração a partir de linguagem natural e acelerar a definição da arquitetura. O resultado é um modelo, não código. A geração de código, quando necessária, ocorre por baixo dele, como detalhe de implementação, e não como o artefato principal.

As implicações operacionais são concretas. As equipes de desenvolvimento dedicam menos tempo à manutenção, porque a superfície de manutenção passa a ser o modelo, e não a base de código. A revisão de segurança fica mais viável, porque a lógica de negócio é expressa de forma declarativa. As auditorias de compliance ficam mais rápidas, porque a camada de decisão pode ser inspecionada sem apoio da engenharia. E o gasto de tokens fica concentrado na parte mais valiosa do desenvolvimento: definir o que o aplicativo deve fazer, em vez de gerar a mecânica de como ele faz.

Apenas 28% dos casos de uso de infraestrutura de IA atendem plenamente às expectativas de ROI, de acordo com a pesquisa de I&O da [Gartner realizada no fim de 2025](#). A distância entre expectativa e realidade não é uma falha de capacidade. É uma falha da arquitetura. As empresas que obtêm ROI da IA no desenvolvimento não estão necessariamente recorrendo a modelos mais avançados. Elas usam a IA para construir artefatos que não se depreciam no ritmo em que o código gerado se deprecia.



A decisão de arquitetura

A escolha que cabe aos líderes de TI não é entre velocidade e segurança. É justamente esse enquadramento que faz a situação atual parecer um beco sem saída. A escolha real está entre duas arquiteturas distintas, com perfis de custo e risco de longo prazo muito diferentes.

Uma arquitetura centrada em código gerado por IA entrega resultado inicial rápido, mas traz uma cauda de manutenção que se acumula, auditabilidade limitada e exposição de segurança que cresce junto com a base de código. Uma arquitetura centrada no desenvolvimento orientado a modelos com IA incorporada entrega um resultado inicial auditável por design, sustentável na camada do modelo, e concentra o gasto de tokens na parte do desenvolvimento que realmente gera valor duradouro.

A liderança de TI não precisa negar a demanda do negócio por entregas aceleradas pela IA. A resposta não é desacelerar. A resposta está em outra filosofia de desenvolvimento, que o negócio consegue sustentar e a TI consegue defender.

O investimento certo em IA não está em escrever código mais rápido. Está em desenvolver modelos de aplicativo que não precisem ser reescritos.





Obrigado

Sobre a Pega

A Pega fornece a principal plataforma impulsionada por AI para transformação empresarial. As organizações mais influentes do mundo confiam na nossa tecnologia para repensar como o trabalho é realizado, automatizando fluxos de trabalho, personalizando a experiência do cliente e modernizando sistemas legados. Desde 1983, nossa arquitetura escalável e flexível vem alimentando uma inovação constante, ajudando clientes a acelerar a jornada rumo à Empresa Autônoma.

[PEGA.COM/PT-BR](https://www.pega.com/pt-br)