

ARTIGO

Vibe coding: o que é, como surgiu, riscos e mais!

**Francine Ribeiro**

5 de Agosto

Atualizado em 4 de Agosto

COMPARTILHE



Nos últimos anos, o desenvolvimento de software tem passado por transformações impulsionadas pela [Inteligência Artificial](#), e um dos conceitos que mais tem ganhado relevância nesse cenário é o **vibe coding**.

Introduzido por Andrej Karpathy em 2025, o termo descreve uma nova abordagem à programação, em que as pessoas desenvolvedoras deixam de se concentrar na escrita do código e passam a expressar suas intenções em linguagem natural, enquanto [agentes de IA](#) se encarregam da implementação técnica.

Essa mudança de paradigma está sendo rapidamente adotada em escala global: segundo o [Vibe Coding Dashboard 2025 da Innobu](#), **a IA já é responsável por gerar 41% de todo novo código, com 256 bilhões de linhas escritas somente em 2024**. Esses dados evidenciam o impacto e a velocidade dessa evolução, além de redefinir o papel da pessoa programadora na era da IA.



Continue a leitura para entender mais sobre o conceito de *vibe coding*, como essa abordagem pode impactar tanto profissionais como empresas, suas principais ferramentas e como introduzi-la nas organizações.

O que é *vibe coding*?

Basicamente, *vibe coding* é uma nova forma de programar que coloca a ideia de quem está desenvolvendo no centro do processo.

Assim, em vez de uma pessoa escrever cada linha de código manualmente, ela descreve o que deseja criar (ou seja, a “vibe” do projeto) e a [Inteligência Artificial](#) cuida da parte técnica, sugerindo códigos prontos, montando estruturas básicas e [automatizando tarefas repetitivas](#).

VEJA TAMBÉM:

[Multiagentes de IA: o que são, como funcionam e aplicações práticas nos negócios](#)

[IA para devs: como a Inteligência Artificial pode aumentar a produtividade do setor de tecnologia?](#)

[Inteligência artificial generativa e o Centauro IA](#)

Essa abordagem permite que a pessoa desenvolvedora foque mais na criatividade e no design da solução, enquanto a IA trata dos detalhes operacionais.

Na prática, o *vibe coding* também torna o desenvolvimento de software mais rápido, intuitivo e acessível, representando uma grande mudança na forma como a programação é feita e como é criada.

O que significa *vibe coding*?

A expressão *vibe coding*, que pode ser traduzida como “codificação por intenção” ou “programar pela vibe”, representa muito mais do que uma técnica. Trata-se de uma mudança na forma de pensar o desenvolvimento de software.

Nesse modelo, como vimos, o fluxo é iterativo: **a pessoa explica, a IA gera o código, a pessoa desenvolvedora testa e ajusta**, e esse ciclo se repete até que o resultado desejado esteja satisfatório.

Isso favorece a experimentação rápida e permite a refinação da solução com base em testes contínuos, alinhando-se aos princípios das [metodologias ágeis](#).

Vale ressaltar também que, mesmo ao codar com IA, o papel do(a) profissional de desenvolvimento continua essencial, por ser quem garante que o código faça sentido, esteja alinhado aos objetivos do projeto e seja seguro.

Por isso, podemos concluir que o sucesso do *vibe coding* depende da capacidade humana de orientar a IA com clareza, revisar com senso crítico e tomar decisões estratégicas com base no contexto do negócio.

4 ferramentas de *vibe coding*

Para experimentar o *vibe coding*, na prática, basta utilizar uma das diversas ferramentas capazes de gerar *AI coding* (código com IA) a partir de comandos em linguagem natural.

Atualmente, existem várias soluções no mercado que viabilizam essa abordagem de forma eficiente. A seguir, apresentamos quatro das ferramentas mais populares.

1. GitHub Copilot

Desenvolvido em parceria com a OpenAI, o [GitHub Copilot](#) é uma das ferramentas mais populares de codificação assistida por IA. Ela se integra diretamente ao VS Code, Neovim, JetBrains e outros editores, oferecendo sugestões em tempo real com base em comentários e no próprio contexto do código.

Melhor uso: ideal para acelerar tarefas rotineiras de programação, gerar trechos de código repetitivos, prototipar funcionalidades rapidamente e explorar novas APIs.

2. Cursor

O [Cursor](#) é um editor de código com IA baseado no VS Code, mas otimizado desde o início para colaboração com IA. Ele permite conversar com o código e realizar buscas contextuais, correções, testes e refatorações usando comandos em linguagem natural.

Melhor uso: fluxos de trabalho iterativos, onde a pessoa desenvolvedora precisa testar ideias para prototipação, exploração de novos projetos ou refatorações em grande escala.

3. Ghostwriter

Integrado à plataforma online Replit, o [Ghostwriter](#) permite a codificação assistida em tempo real no navegador, com suporte para dezenas de [linguagens](#). Seu sistema oferece sugestões completas de código, depuração assistida por IA e até mesmo explicações em linguagem simples do que cada trecho faz.

Melhor uso: perfeito para pessoas desenvolvedoras iniciantes ou para fins educativos. Também é útil em contextos colaborativos, onde múltiplos membros de uma equipe compartilham um mesmo projeto.

4. Amazon Q Developer

Voltado principalmente para pessoas desenvolvedoras que usam o ecossistema AWS, o [Amazon Q](#) oferece sugestões de código, snippets, integração com [serviços da nuvem](#) e suporte para diversas linguagens de programação, como Python, Java e JavaScript.

Melhor uso: indicado para equipes que desenvolvem em ambientes de produção usando infraestrutura da AWS. Também é uma escolha sólida para empresas preocupadas com conformidade e segurança em aplicações corporativas.

Leia também: [10 ferramentas de IA para aumentar a produtividade da sua empresa](#)

Desafio ético por trás do *vibe coding*

Embora o *vibe coding* esteja frequentemente associado ao [uso de IA na programação](#), ele possui uma definição específica (e controversa). Em sua forma original, a abordagem significa aceitar o código gerado pela IA sem necessariamente compreendê-lo por completo.

Essa característica é o que mais diferencia essa abordagem de uma prática mais tradicional de desenvolvimento assistido por IA, onde a pessoa programadora continua sendo responsável por revisar, testar e entender tudo o que está sendo implementado. Essa diferença foi destacada pelo pesquisador [Simon Willison](#), que afirmou:

“Se um [LLM](#) escreveu cada linha do seu código, mas você revisou, testou e entendeu tudo, isso não é vibe coding na minha opinião — isso é usar um LLM como um assistente de digitação.”

Essa distinção é crucial, especialmente porque o termo *vibe coding* passou a ser usado amplamente no mercado, muitas vezes por empresas que promovem [ferramentas de IA](#) como soluções profissionais sob esse rótulo.

Ao fazer isso, existe o risco de mudar o significado original do termo e, pior, de promover práticas potencialmente perigosas.

Por trás do nome atrativo, algumas organizações podem acabar incentivando a ideia de que é aceitável “esquecer que o código existe”. Ou seja, adotar o código com IA sem questionamento, mesmo em sistemas críticos de produção.

Por este motivo, mais do que uma discussão técnica, o debate em torno do *vibe coding* revela um choque ético e cultural.

De um lado, estão os valores clássicos da engenharia de software: **rigoriedade, responsabilidade técnica e compreensão profunda do sistema**. Do outro, surge uma nova cultura, baseada na **velocidade, nos prompts e na experimentação rápida**, impulsionada pela acessibilidade das [ferramentas de IA](#).

Essa discussão já provocou reações fortes dentro da comunidade das pessoas desenvolvedoras. Inclusive, o termo *vibe coding* já foi duramente criticado, chegando a ser classificado por muitos(as) profissionais como uma prática “descuidada”, “preguiçosa” ou até mesmo como um “insulto à engenharia de software”.

Diante disso, o verdadeiro desafio não está somente em adotar novas ferramentas, mas em equilibrar dois mundos: de um lado, a [inovação](#) e agilidade oferecidas pela [Inteligência Artificial](#); de outro, os fundamentos sólidos da programação.

Leia também: [Ética e Inteligência Artificial — desenvolver a tecnologia com consciência](#)

Quais os riscos de codar com IA?

Complementando as questões éticas que o *vibe coding* possui, existem outros riscos que, se ignorados, podem comprometer seriamente a qualidade e a segurança de diferentes projetos. Abaixo, estão os principais desafios associados à prática de codificação com IA — especialmente em sua forma mais automatizada, como no *vibe coding puro*.

Compreensão superficial do código

Ao aceitar um *IA coding* sem revisar cuidadosamente sua lógica, as pessoas desenvolvedoras correm o risco de implementar soluções que funcionam momentaneamente, mas que não são totalmente compreendidas. Isso afeta diretamente a capacidade de manter, escalar ou adaptar o sistema criado no futuro.

Introdução de falhas e vulnerabilidades

Modelos de IA, como [LLMs](#), aprendem com grandes volumes de código disponível na internet, incluindo padrões inseguros, desatualizados ou errôneos. Isso significa que o código gerado pode conter:

- erros lógicos difíceis de identificar em testes superficiais;

- vulnerabilidades de [segurança](#), como falhas em autenticação e problemas de validação de entrada;

- práticas obsoletas, como o uso de bibliotecas descontinuadas ou abordagens ultrapassadas de arquitetura.

Se o código for colocado em produção sem validação adequada, esses riscos podem resultar em falhas sistêmicas, vazamentos de dados ou exploração de vulnerabilidades por ciberataques.

Leia também: [A arquitetura de segurança como uma ferramenta para proteger sua empresa](#)

Dependência da IA

O uso frequente de ferramentas de codificação assistida sem reflexão crítica pode levar à perda de autonomia técnica. Isso significa que as pessoas desenvolvedoras podem deixar de praticar habilidades fundamentais como:

- estruturação lógica de algoritmos;

- escrita e interpretação de código limpo;

- análise de desempenho;

- compreensão de documentação e bibliotecas.

Problemas de propriedade intelectual

A IA pode gerar código com base em exemplos extraídos de repositórios públicos, como o GitHub, durante seu treinamento.

Embora os modelos não copiem código diretamente, há o risco de reprodução parcial de trechos protegidos por licenças específicas (como GPL ou MIT). Isso pode gerar implicações legais, especialmente em produtos comerciais que exigem direitos autorais e licenças de software.

Leia também: [Os 10 maiores riscos da inteligência artificial](#)

Como usar o *vibe coding* nas empresas?

Apesar dos desafios envolvidos, a abordagem *vibe coding*, como vimos ao longo do texto, representa uma verdadeira faca de dois gumes.

Por um lado, há riscos que não podem ser ignorados, como a falta de compreensão total do código com IA gerado.

Por outro lado, também existem benefícios concretos, como [ganho de produtividade](#), aceleração na prototipação e mais fluidez nos processos criativos de desenvolvimento.

O que ninguém pode contestar é que essa prática já é uma realidade no mercado. Sendo assim, a questão que devemos evidenciar é: como as empresas podem usar o *vibe coding* de forma ética em seus projetos?

Para isso, separamos algumas etapas abaixo que, para empresas que querem adotar essa abordagem, serão essenciais para enfrentar os desafios e usar essa tecnologia a favor da organização.

1ª etapa: fale com sua equipe e escolha a ferramenta

Antes de tudo, entenda como o *vibe coding* pode ser introduzido na sua empresa.

Para isso, converse com seu time de desenvolvimento juntamente com a sua [liderança tech](#) para entender como essa abordagem pode ser benéfica para o setor.

Nesta conversa, alinhe também os objetivos e, a partir deles, busque a ferramenta de *vibe coding* que mais se alinhe à equipe, como as opções que apresentamos anteriormente. A escolha deve considerar:

nível de [autonomia](#) desejado;

[linguagens de programação utilizadas;](#)

integração com plataformas já existentes (como AWS, Git, CI/CD).

2ª etapa: estabeleça diretrizes claras de uso

Para evitar usos irresponsáveis, as empresas devem criar políticas internas de governança de IA.

Isso inclui padrões mínimos de validação de código e definições sobre onde e quando a IA pode ser usada, como nas fases de produção e prototipagem.

Essas diretrizes equilibram criatividade e controle para promover responsabilidade e segurança, tanto para os projetos da empresa quanto para a equipe, pois mostra que o trabalho de seus colaboradores e colaboradoras é essencial e insubstituível no processo.

Leia também: [Governança de dados — o que é e como implementar](#)

3ª etapa: capacite a equipe para trabalhar com IA

Implementar *vibe coding* exige uma mudança de processos, mas também de cultura. Para que as pessoas envolvidoras da sua empresa usem essa abordagem com eficiência, é preciso que toda a equipe seja treinada em [letramento em IA](#).

Assim, todas as pessoas podem conseguir formular prompts eficientes, avaliar criticamente as respostas da IA e melhorar os resultados gerados.

Leia também: [Como a IA pode facilitar o trabalho das pessoas colaboradoras dentro de uma empresa](#)

4ª etapa: acompanhe o uso e os resultados do *vibe coding*

Após implementar ferramentas e integrar o *vibe coding* ao fluxo de trabalho do seu time, é essencial monitorar continuamente seu uso e os impactos gerados.

Essa etapa garante que a prática esteja realmente agregando valor ao desenvolvimento, além de ajudar a identificar pontos de melhoria e evitar desvios que comprometam a qualidade do código.

Para concluir, podemos resumir que o *vibe coding* representa uma oportunidade concreta de transformar a maneira como o software é construído dentro das empresas.

Com uso responsável, processos bem definidos e [capacitação das equipes](#), a IA pode se tornar uma aliada estratégica poderosa.

Destacamos que o futuro do desenvolvimento corporativo não está em substituir pessoas desenvolvedoras por máquinas, mas sim em potencializar suas capacidades humanas com o apoio inteligente da tecnologia.

Para colocar isso em prática, sua empresa pode contar com a parceria educacional da **Alura + FIAP Para Empresas**. Com formações e treinamentos específicos para IA e outras tecnologias, sua equipe terá o conhecimento necessário para integrar o *vibe coding* com ética e eficiência.

Leia também: [Human-centered AI — como criar soluções de inteligência artificial com foco nas pessoas](#)

Confira neste artigo:

[O que é vibe coding?](#)

[O que significa vibe coding?](#)

[4 ferramentas de vibe coding.](#)

[Desafio ético por trás do vibe coding.](#)

[Quais os riscos de codar com IA?](#)

[Como usar o vibe coding nas empresas?](#)



Francine Ribeiro

Analista de Conteúdo da Alura +FIAP Para Empresas. Jornalista de formação, com MBA em Comunicação Corporativa pela Universidade Tuiuti do Paraná (UTP) e MBA em Business Strategy e Transformation pela FIAP. Atua com produção de conteúdo para empresas desde 2009 e com marketing digital desde 2016.

ARTIGO ANTERIOR

[Motivação no trabalho: estratégias para manter sua equipe motivada no dia a dia](#)

PRÓXIMO ARTIGO

[Equipe multicultural: como gerenciar times de diversos países de forma eficaz](#)

A Alura Para Empresas é a organização que engloba as soluções corporativas da Alura — a maior escola online de tecnologia do Brasil, voltadas a empresas, órgãos governamentais e instituições educacionais.

Uma empresa do grupo Alun



Inscriva-se em nossa newsletter

Email corporativo*

INSCREVA-SE

EMPRESA

Sobre nós

Contato

Planos

Fale com especialistas

CLIENTES

Por segmento

Financeiro

Tech

Gov

Multi

NOSSOS PRODUTOS

Áreas e habilidades

Plataforma

Programas personalizados

Formação continuada

Programas de impacto social

CONTEÚDOS

Artigos

Materiais

Podcasts

Cases

Na Mídia

Por porte

Newsletter

Pequenas e médias

Grandes e enterprise

