



**\*TRS**

**Tecnologia, Redes e Sociedade**

Informação ( gestão | espaços | ubíqua e inteligente )

## Relatório Interno TRS 01/2024

Título

Inteligência Artificial (IA) e IA Generativa  
para geração de código

Autor(es)

Luís Borges Gouveia, UFP

Mês, Ano

Dezembro, 2024

Local de presença Web <http://tecnologiaredesesociedade.wordpress.com>

Repositório de trabalho científico \*trs <http://bdigital.ufp.pt/handle/10284/3787>

Universidade Fernando Pessoa

Praça 9 de Abril, 349

4249-004 Porto, Portugal

## Tabela de Conteúdos

Resumo.....	3
1. Estado da Arte Atual .....	3
2. Principais tendências .....	4
3. Utilização da IA Generativa na Produção de Código.....	4
4. Exemplo de ferramentas disponíveis.....	5
5. Limitações das ferramentas de geração de código com IA atuais.....	6
6. Quais as áreas da programação que constituem ainda mais desafios para serem automatizadas? .....	8
7. Perspetivas futuras .....	9
8. Considerações finais.....	9
Lista de ferramentas IA para desenvolvimento de software .....	10

## Inteligência Artificial (IA) e IA Generativa para geração de código

Luís Borges Gouveia, [lmbg@ufp.edu.pt](mailto:lmbg@ufp.edu.pt)

### Resumo

O relatório interno apresenta a posição a 9 de dezembro de 2024, associada com as ferramentas de geração de código para auxiliar na prática de programação e para o desenvolvimento de software. Neste contexto, é apresentada de forma sintética uma discussão de como o recurso à Inteligência Artificial e, em particular, à Inteligência Artificial Generativa.

O uso de modelos de linguagem massivos e de sistemas associados com estes, permite a geração ou análise de código e existe já uma generalização de adoção de meios de suporte que estendem sistemas que são adotados no contexto da engenharia do software e no desenvolvimento e escrita de código.

O texto apresenta algumas das questões que importa aprofundar e termina com uma lista não exaustiva destas ferramentas com indicação de nome, endereço na Web onde podem ser obtidas (URL) e uma breve discussão do seu potencial e contexto de uso.

Verifica-se desta forma uma área ainda em maturação, com uma oferta de novas propostas significativas, mas que já tem os protagonistas e principais atores atuais, também envolvidos na oferta de ferramentas e serviços de geração de código. O relatório explora uma área que exige trabalho adicional para avaliar não apenas as ferramentas, mas o seu impacto nas competências e na própria indústria de desenvolvimento de software.

**Palavras-Chave:** geração de código; programação; inteligência artificial; inteligência artificial generativa; desafios.

### 1. Estado da Arte Atual

Em 2024, a IA generativa encontra-se num estágio avançado, com a adoção crescente em diversas indústrias. Ferramentas de IA generativa estão se torna-se cada vez mais sofisticadas, utilizando grandes modelos de linguagem (LLMs) e algoritmos de aprendizagem profunda para oferecer um

suporte abrangente aos programadores. Deste modo, a IA generativa tem proporcionado uma tecnologia que oferece ferramentas poderosas para auxiliar na produção de código, com suporte em diversas etapas do desenvolvimento de software.

Os avanços recentes incluem a capacidade de gerar código de alta qualidade, identificar e corrigir bugs, e até mesmo traduzir código de uma linguagem para outra. No entanto, ainda há desafios, como custos elevados de treino de modelos de IA de última geração e a necessidade de revisão humana para garantir a precisão do código gerado.

## 2. Principais tendências

Considerando a geração de código e o desenvolvimento de software, assistimos ao recrudescimento de práticas que envolvem o uso e exploração da inteligência artificial. Em especial, é possível destacar quatro possibilidades de explorar a tecnologia:

- *Uso de modelos de linguagem avançados*: ferramentas como os assistentes baseados em modelos de linguagem estão a tornar possível gerar código de alta qualidade com base em promptings (em especial, pela partilha de trechos de código);
- *Integração com editores de código*: IDEs como o VSCode e o JetBrains estão a incorporar funcionalidades de IA para sugerir código e para refatoração;
- *Automação de tarefas repetitivas*: ferramentas como Kite AI e Tabnine estão a usar a IA para automatizar tarefas como completar código, refatoração e sugestões de melhorias;
- *Geração de código a partir de descrições naturais*: modelos como LLMs estão a permitir gerar código a partir de descrições em linguagem natural.

## 3. Utilização da IA Generativa na Produção de Código

As ferramentas de IA podem ajudar a otimizar o processo de programação de várias maneiras, mesmo que não possam resolver todos os problemas sozinhas ou mesmo assegurar a obtenção de um código totalmente funcional (embora durante o último ano, tenha havido um desenvolvimento muito positivo na geração de código funcional à primeira tentativa):

1. *Sugestões e assistência em tempo real*: ferramentas como o GitHub Copilot e o Cursor AI oferecem sugestões de código em tempo real, ajudando os programadores a escrever código mais rápido e com maior eficiência;
2. *Deteção automática de erros e vulnerabilidades*: a IA pode identificar bugs e falhas de segurança que podem ter sido esquecidos ou não detetados por programadores humanos, tornando o processo de revisão de código muito mais eficiente, rápido e com maior probabilidade de cobertura de eventuais problemas existentes;
3. *Geração de testes automatizados*: as ferramentas como o Codium AI podem gerar testes com significado e válidos, a partir de descrições em linguagem natural, facilitando a revisão e a transformação dessas descrições em código de teste;
4. *Análise preditiva e otimização de projetos*: a IA pode ser baseada em dados históricos para estimar cronogramas, requisitos de recursos e identificar riscos em projetos, ajudando os programadores a se adaptarem às necessidades de cada projeto em particular, de forma mais eficaz;
5. *Automatização de tarefas repetitivas*: a IA pode ajudar a automatizar tarefas como documentação, análise de código e refatoração, libertando tempo dos programadores para estes se focarem em tarefas mais complexas e criativas, tornando-os mais produtivos e acelerando o desenvolvimento de software (o que reduz os custos associados);
6. *Aprendizagem assistida*: as ferramentas de IA podem ajudar os programadores menos experientes a aprender novas linguagens e técnicas, sugerindo exemplos e explicações sobre partes do código;
7. *Integração com sistemas existentes*: a IA pode facilitar a integração entre plataformas e sistemas legados, simplificando a migração e a interoperabilidade;
8. *Melhoria da qualidade do código*: alguns sistemas de IA podem analisar o código e sugerir melhorias na estrutura, modularidade e eficiência, tornando-o mais fácil de entender e manter, sendo que a manutenção é um dos aspetos críticos associados com o desenvolvimento de software.

#### **4. Exemplo de ferramentas disponíveis**

Entre as muitas ferramentas que atualmente é possível considerar, as seguintes sete, apresentam uma seleção que se propõe partilhar a diversidade de soluções e oferta que existe:

1. GitHub Copilot: um assistente de codificação que sugere linhas de código e funções completas com base no contexto do que o programador está a escrever;

2. ChatGPT: pode ser usado para gerar código a partir de descrições em linguagem natural, além de ajudar na depuração e otimização de código;
3. Tabnine: oferece sugestões de código baseadas em IA para várias linguagens de programação, ajudando a acelerar o processo de desenvolvimento;
4. Codiga: focado na análise e revisão de código, sugere melhorias e otimizações para tornar o código mais eficiente;
5. Amazon CodeWhisperer: um assistente de codificação que oferece sugestões de código e ajuda na depuração, integrado com os serviços da AWS;
6. Kite AI: oferece sugestões de código baseadas em IA, incluindo geração de código e refatoração;
7. Codota: plataforma que usa a IA para gerar código a partir de descrições naturais.

## 5. Limitações das ferramentas de geração de código com IA atuais

Embora seja reconhecido um grande potencial e já uma capacidade efetiva para a produção de código, verifica-se que existem limitações para o que pode, de facto, ser obtido com os sistemas atuais. Considerando o contexto atual, podemos afirmar que ocorrem as seguintes limitações:

- *Falta de contexto completo*: as IA podem gerar código baseado em padrões e exemplos, mas não compreendem o contexto completo do projeto como um ser humano o faria. Podem sugerir soluções que parecem corretas à primeira vista, mas que não funcionam adequadamente dentro do contexto específico de um dado projeto;
- *Dependência em dados e exemplos anteriores*: as IA não têm a capacidade de inovar ou criar soluções inéditas que vão além do que já foi programado ou alimentado no seu treino. Tal pode limitar a criatividade e a capacidade de propor soluções verdadeiramente originais;
- *Limitações em interpretar e resolver problemas complexos*: questões como a otimização de desempenho, manipulação de grandes volumes de dados ou integração com bibliotecas específicas exigem um entendimento profundo de como uma determinada linguagem funciona. A IA pode ajudar com exemplos ou abordagens gerais, mas o ajuste fino e a solução definitiva muitas vezes dependem da competência do programador de aplicar os princípios da linguagem de forma criativa e eficiente;
- *Falta de domínio dos fundamentos*: confiar cegamente na IA pode levar a erros subtis que, sem o domínio das bases quer da programação, quer da linguagem em uso, podem passar despercebidos e comprometer todo o projeto. Esta situação é mais problemática para

aqueles que iniciam agora a sua atividade como programadores e que não viveram uma época em que formaram a sua experiência ainda sem este tipo de ferramentas;

- *Risco de aceitar sugestões sem questionar*: programadores menos experientes podem acabar por aceitar sugestões da IA sem questionar, o que pode levar a resultados inesperados ou até mesmo a bugs difíceis de rastrear;
- *Limitações na integração com sistemas existentes*: muitas empresas operam sistemas legados que podem ser rígidos e não suportar as APIs modernas necessárias. Esta realidade implica que não seja possível uma integração suave das funcionalidades de IA com o contexto específico e as necessidades de desenvolvimento de software em certos ambientes;
- *Desafios éticos e de conformidade*: o desenvolvimento de IA envolve considerações éticas complexas, como privacidade, preconceito e potencial uso indevido da tecnologia. Este último, facilitando o aparecimento de atores com maior sofisticação e risco de mau uso e criação de tecnologia e software malicioso;
- *Complexidade em gerir expectativas*: os utilizadores finais e partes interessadas têm frequentemente grandes expectativas em relação às ferramentas de IA, antecipando soluções quase mágicas para problemas complexos;
- *Dificuldade de adaptação à evolução rápida do campo*: o campo da IA está em constante mudança, o que cria um desafio para os programadores que devem manter-se atualizados com os avanços mais recentes e incorporar métodos e práticas do estado da arte, nas suas ferramentas;
- *Recursos computacionais intensivos*: o treino de modelos de IA e de aprendizagem máquina, especialmente redes neuronais e aprendizagem profunda, é exigente no uso intensivo de recursos computacionais, o que pode ser caro, demorado de obter e exclusivo para alguns.

É importante notar que, apesar de todas estas limitações, as ferramentas de IA estão em constante melhoria e podem ser de grande valia quando usadas de forma consciente e em conjunto com o conhecimento e julgamento humano.

## 6. Quais as áreas da programação que constituem ainda mais desafios para serem automatizadas?

É importante notar que, apesar das limitações enumeradas, a IA está em constante evolução e melhora o seu desempenho e capacidades de forma rápida, muito significativa e de forma otimizada, o que proporciona menores custos, maior disponibilidade e generalização, alargando a praticamente todos os programadores que o pretendam, a geradores de código.

Em muitas áreas da programação, já existem ofertas de valor de ferramentas de assistência e automação, mas é crítico entender como estas novas capacidades e as suas limitações podem condicionar uma integração de forma efetiva nos fluxos de trabalho.

Importa assim considerar quais as principais áreas da programação onde a IA tem menos capacidade de auxiliar ou cujo uso se revela mais desafiante:

- *Tarefas altamente criativas e inovadoras*: a IA tem dificuldade em gerar soluções verdadeiramente originais ou inovadoras, especialmente em áreas que requerem pensamento lateral e criatividade única;
- *Contexto e compreensão holística*: a IA pode ter dificuldade em entender o contexto completo de um projeto ou sistema, o que pode levar a sugestões ou soluções que não funcionam adequadamente dentro do contexto específico do problema a resolver;
- *Problemas complexos e não estruturados*: questões que envolvem múltiplos fatores interdependentes ou que não podem ser facilmente quantificados ou estruturados podem ser desafiadoras para a IA;
- *Domínios especializados com conhecimento profissional*: áreas que exigem um entendimento profundo e específico de um domínio particular podem ser difíceis para a IA replicar sem treino extensivo (experiência real) e atualização contínua;
- *Tarefas que requerem julgamento ético*: decisões que envolvem questões éticas complexas ou considerações morais, podem ser desafiadoras para a IA, pois dependem de valores e princípios que não são facilmente codificáveis;
- *Interação com sistemas legados*: a integração com sistemas existentes que podem ter restrições técnicas ou requisitos específicos podem também constituir um desafio para a IA;
- *Análise de dados em tempo real com muitas variáveis*: situações que exigem análise rápida de grandes volumes de dados em tempo real podem ser complexas para a IA processar adequadamente, em especial se conterem múltiplas dimensões e um carácter dinâmico;

- *Tarefas que requerem flexibilidade e adaptação constante*: áreas que exijam uma capacidade de adaptação rápida a mudanças ou novos cenários podem ser desafiadoras para sistemas baseados em IA.

## 7. Perspetivas futuras

O recurso a este tipo de ferramentas para a geração de código está em rápido crescimento, com expectativas de avanços significativos já para os próximos anos e não nas próximas décadas. Um esforço de investigação e desenvolvimento está em curso para explorar formas de melhorar a qualidade do código gerado, aumentar a capacidade de adaptação às preferências individuais dos programadores e integrar estas tecnologias de forma mais profunda nos processos de desenvolvimento e nos ambientes e práticas de desenvolvimento de software existentes.

É importante notar que, embora essas ferramentas estejam a revolucionar a forma como escrevemos código, elas não substituem completamente o papel do programador humano. Pelo contrário, elas estão a ser projetadas para aumentar a produtividade e qualidade do trabalho dos programadores, permitindo que eles se concentrem em tarefas mais complexas e criativas – reforçando a importância de considerar também a aprendizagem, o uso e a exploração destas ferramentas.

## 8. Considerações finais

Os geradores de código para suporte ao desenvolvimento de software foram introduzidas e o caminho realizado não tem retorno. De facto, tudo parece indicar que o recurso à IA para o desenvolvimento e produção de código será incrementado. Neste contexto, parece ser adequado considerar os seguintes pontos a ter em consideração:

- *Ética e transparência*: há debates sobre o uso ético da IA em geração de código e possíveis questões de autoria;
- *Aprendizagem contínua*: as ferramentas precisam de se adaptar continuamente para se manterem atualizadas com as melhores práticas de programação;
- *Integração com fluxos de trabalho existentes*: as ferramentas mais eficazes são aquelas que estão integradas com os fluxos de trabalho dos programadores;

- *Personalização*: de futuro, será eventualmente possível ter ferramentas capazes de aprender com o estilo do programador individual.

Deste modo, verifica-se uma tendência para uma maior sofisticação e integração de facilidade de IA e IA Generativa no contexto do desenvolvimento de software, o que aconselha ao estudo destas práticas e à integração de ferramentas deste tipo, no ensino e aprendizagem de programação e do estudo da engenharia de software.

### Lista de ferramentas IA para desenvolvimento de software

Uma lista não completa com algumas das ferramentas disponíveis para uso e exploração.

Nome da Ferramenta	URL	Descrição
GitHub Copilot	<a href="https://github.com/features/copilot">https://github.com/features/copilot</a>	Assistente de codificação que sugere linhas de código e funções completas
ChatGPT	<a href="https://chat.openai.com/">https://chat.openai.com/</a>	Gera código a partir de descrições em linguagem natural e ajuda na depuração
Tabnine	<a href="https://www.tabnine.com/">https://www.tabnine.com/</a>	Oferece sugestões de código baseadas em IA para várias linguagens de programação
Codiga	<a href="https://www.codiga.io/">https://www.codiga.io/</a>	Analisa e otimiza código, sugerindo melhorias e correções
Amazon CodeWhisperer	<a href="https://aws.amazon.com/codewhisperer/">https://aws.amazon.com/codewhisperer/</a>	Assistente de codificação integrado com serviços da AWS
IBM Watson	<a href="https://www.ibm.com/watson">https://www.ibm.com/watson</a>	Plataforma de IA que oferece ferramentas e APIs para integrar IA em aplicações
DeepCode	<a href="https://www.deepcode.ai/">https://www.deepcode.ai/</a>	Analisa código para identificar e corrigir bugs e vulnerabilidades
Kite	<a href="https://www.kite.com/">https://www.kite.com/</a>	Autocompletar de código com IA para Python, JavaScript e outras linguagens
Repl.it Ghostwriter	<a href="https://replit.com/site/ghostwriter">https://replit.com/site/ghostwriter</a>	Assistente de codificação que ajuda a escrever e depurar código
Snyk	<a href="https://snyk.io/">https://snyk.io/</a>	Identifica e corrige vulnerabilidades de segurança no código
Codex	<a href="https://openai.com/research/codex">https://openai.com/research/codex</a>	Modelo de IA da OpenAI que traduz linguagem natural em código

<b>CodeGuru</b>	<a href="https://aws.amazon.com/codeguru/">https://aws.amazon.com/codeguru/</a>	Ferramenta da AWS que faz a revisão do código e sugere melhorias
<b>Ponicode</b>	<a href="https://www.ponicode.com/">https://www.ponicode.com/</a>	Gera testes unitários automaticamente para garantir a qualidade do código
<b>SourceAI</b>	<a href="https://www.sourceai.dev/">https://www.sourceai.dev/</a>	Gera código a partir de descrições em linguagem natural
<b>CodeT5</b>	<a href="https://github.com/salesforce/CodeT5">https://github.com/salesforce/CodeT5</a>	Modelo de IA para geração e tradução de código
<b>Codota</b>	<a href="https://www.codota.com/">https://www.codota.com/</a>	Sugestões de código baseadas em IA para Java e Kotlin
<b>IntelliCode</b>	<a href="https://visualstudio.microsoft.com/services/intellicode/">https://visualstudio.microsoft.com/services/intellicode/</a>	Sugestões de código inteligentes integradas ao Visual Studio
<b>Jupyter Notebooks</b>	<a href="https://jupyter.org/">https://jupyter.org/</a>	Ferramenta interativa para desenvolvimento de código e análise de dados
<b>PyCharm</b>	<a href="https://www.jetbrains.com/pycharm/">https://www.jetbrains.com/pycharm/</a>	IDE com recursos de IA para desenvolvimento em Python
<b>AlphaCode</b>	<a href="https://deepmind.com/research/highlighted-research/alphacode">https://deepmind.com/research/highlighted-research/alphacode</a>	Modelo de IA da DeepMind para geração de código e resolução de problemas
<b>CodeWP</b>	<a href="https://codewp.ai">https://codewp.ai</a>	Gera código WordPress a partir de descrições em linguagem natural.
<b>Codex CLI</b>	<a href="https://github.com/openai/openai-codex-cli">https://github.com/openai/openai-codex-cli</a>	Interface de linha de comando para interagir com o modelo Codex da OpenAI
<b>Codeium</b>	<a href="https://www.codeium.com">https://www.codeium.com</a>	Auto completar de código com IA para várias linguagens de programação
<b>Sourcery</b>	<a href="https://sourcery.ai">https://sourcery.ai</a>	Sugestões de refatoração de código para Python
<b>Tabby</b>	<a href="https://tabbyml.github.io/tabby">https://tabbyml.github.io/tabby</a>	Assistente de codificação <i>opensource</i> com suporte para várias linguagens
<b>Polycoder</b>	<a href="https://polycoder.dev">https://polycoder.dev</a>	Gera código a partir de descrições em linguagem natural
<b>CodeWP</b>	<a href="https://codewp.ai">https://codewp.ai</a>	Gera código WordPress a partir de descrições em linguagem natural