

Análise da Produtividade no Desenvolvimento com Assistência de Modelos de Linguagem de Grande Escala

Luryan Delevati Dorneles

Fabio Santana Linhares

Hans Ponfick de Aragão

Universidade Federal de Alagoas

27 de maio de 2025

Resumo

Este trabalho apresenta uma reflexão crítica sobre a integração de Modelos de Linguagem de Grande Escala (LLMs) no processo de desenvolvimento de software, baseando-se na experiência prática com o GitHub Copilot durante a implementação de uma ferramenta de análise probabilística de desmatamento na Amazônia. A análise foca nos ganhos de produtividade, na qualidade do código gerado e na transformação do fluxo de trabalho do desenvolvedor quando assistido por inteligência artificial. As observações indicam que a tecnologia proporcionou redução significativa no tempo de desenvolvimento, melhorou a qualidade dos códigos e textos e permitiu maior foco em aspectos conceituais das soluções. São discutidas também as limitações observadas e as implicações para o futuro da prática profissional.

Palavras-chave: Modelos de Linguagem de Grande Escala. GitHub Copilot. Engenharia de Software. Inteligência Artificial. Programação Assistida.

Abstract: This paper presents a critical reflection on the integration of Large Language Models (LLMs) in the software development process, based on practical experience with GitHub Copilot during the implementation of a probabilistic analysis tool for Amazon deforestation. The analysis focuses on productivity gains, code quality improvements, and transformations in the developer's workflow when assisted by artificial intelligence. Observations indicate that the technology provided significant reduction in implementation time, improved code quality, and allowed greater focus on conceptual aspects of solutions. The study reveals a qualitative reconfiguration of the development process, with the human developer shifting from syntax-focused coding to higher-level architectural design. We also discuss observed limitations, particularly in domain-specific understanding, and implications for future professional practice in software engineering, where LLMs serve not as replacements but as cognitive amplifiers of human creativity.

Keywords: Large Language Models. GitHub Copilot. Software Engineering. Artificial Intelligence. Assisted Programming.

1 Introdução

A emergência dos Modelos de Linguagem de Grande Escala (LLMs) aplicados ao desenvolvimento de software representa um momento transformador para a prática. Ferramentas como GitHub Copilot, fundamentadas em modelos de linguagem treinados em vastos repositórios de código-fonte, têm demonstrado capacidade impressionante de compreender contexto e gerar implementações funcionais a partir de descrições em linguagem natural ou fragmentos iniciais de código. Este fenômeno suscita questões importantes sobre como estas tecnologias impactam a produtividade, a qualidade do código e, fundamentalmente, o próprio processo criativo de programação.

O presente trabalho analisa criticamente a experiência de desenvolvimento de uma solução computacional com assistência do GitHub Copilot: uma implementação de modelos probabilísticos para análise de dados de desmatamento na Amazônia brasileira, combinando Modelos Ocultos de Markov e Redes Bayesianas.

Esta análise não se propõe a comparar diretamente implementações com e sem assistência de LLM, mas sim a examinar qualitativamente como a integração desta tecnologia moldou o processo de desenvolvimento, identificando padrões de interação, contribuições específicas do assistente e transformações no fluxo de trabalho. Busca-se compreender em que medida estes assistentes baseados em IA podem potencializar as capacidades do desenvolvedor e como esta relação homem-máquina reconfigura a natureza do trabalho.

2 Desenvolvimento Assistido por LLM: Contexto e Metodologia

O desenvolvimento da solução computacional ocorreu em ambiente integrado Visual Studio Code com extensão GitHub Copilot ativada, proporcionando sugestões contextuais em tempo real durante o processo de codificação. A implementação utilizou Python como linguagem principal, complementada por bibliotecas específicas de análise de dados e aprendizado de máquina (pandas, numpy, hmmlearn, pgmpy) para o sistema de análise de desmatamento.

A dinâmica de interação com o assistente seguiu um padrão recorrente: inicialmente, elaborava-se uma descrição do problema ou componente a ser implementado, frequentemente na forma de comentários ou docstrings; o assistente então sugeria implementações completas ou parciais; estas sugestões eram avaliadas, frequentemente aceitas integralmente para estruturas básicas e adaptadas para lógicas específicas do domínio. Desta forma, estabeleceu-se um fluxo de trabalho onde o desenvolvedor humano atuava predominantemente como arquiteto e avaliador, enquanto o assistente operava como implementador primário.

Para a solução de análise de desmatamento, o processo envolveu primeiramente a implementação de rotinas de preparação e transformação dos dados históricos, seguida pela modelagem com HMM para identificação de regimes temporais e, posteriormente, desenvolvimento da Rede Bayesiana para modelagem causal. O assistente mostrou-se particularmente eficaz na geração de código para manipulação de dados e implementação dos algoritmos de aprendizado de máquina, sugerindo estruturas otimizadas e boas práticas de implementação estatística.

3 Resultados e Análise da Produtividade

A experiência de desenvolvimento com assistência do GitHub Copilot revelou dimensões importantes da contribuição dos LLMs para o processo. O aspecto mais imediatamente perceptível foi o impacto no tempo para obter a solução. A redução significativa no tempo de pesquisa em documentação técnica representou uma das maiores vantagens - tarefas que tradicionalmente exigiriam consultas extensas a APIs e exemplos foram substituídas por simples prompts descritivos. Estima-se que o tempo necessário para implementar funcionalidades foi aproximadamente 40-50% menor, não apenas pela velocidade de digitação, mas principalmente pela capacidade de obter funções complexas prontas através de prompts bem elaborados. Com o conhecimento conceitual de o que e como fazer, bastava articular a intenção para receber implementações funcionais que poderiam ser imediatamente incorporadas no fluxo do pipeline, particularmente em componentes como transformações estatísticas, visualizações de dados e algoritmos de aprendizado de máquina.

Para além da aceleração quantitativa do desenvolvimento, observou-se uma transformação qualitativa no processo. O assistente mostrou-se capaz não apenas de completar padrões sintáticos previsíveis, mas frequentemente de antecipar necessidades estruturais, sugerir implementações alternativas e introduzir boas práticas que não haviam sido explicitamente solicitadas. No desenvolvimento do sistema de análise de desmatamento, o assistente espontaneamente sugeriu visualizações estatísticas complementares e métricas adicionais que enriqueceram a análise. Esta capacidade generativa transcende a mera automação, aproximando-se de uma genuína colaboração criativa.

A qualidade do código produzido com assistência do LLM mostrou-se consistentemente elevada em diversos aspectos. Observou-se particular atenção a padrões de design estabelecidos, tratamento apropriado de exceções, documentação automática de funções e classes, e estruturação modular.

Um benefício significativo foi a facilitação da implementação de testes unitários. O assistente demonstrou notável capacidade de gerar casos de teste abrangentes, considerando tanto cenários típicos quanto condições limítrofes, o que resultou em uma cobertura de testes substancialmente mais ampla do que seria provável em desenvolvimento manual com as mesmas restrições temporais. Para o sistema de análise de desmatamento, o assistente gerou automaticamente verificações para cenários específicos como ausência de dados e condições extremas nos modelos probabilísticos.

Após obter os dados, a assistência do LLM foi fundamental para analisar, desenvolver códigos, levantar questionamentos metodológicos, sugerir funções apropriadas, recomendar o uso de bibliotecas específicas, e até mesmo auxiliar no desenvolvimento do texto e da documentação. O assistente funcionou efetivamente como um recurso para consultas sobre técnicas estatísticas específicas, implementações algorítmicas e boas práticas em análise de dados, tornando-se efetivamente "um núcleo de conteúdo à disposição da curiosidade" do desenvolvedor.

4 Transformações no Processo de Desenvolvimento

Para além dos ganhos quantificáveis em velocidade e qualidade, a assistência do LLM induziu transformações fundamentais no próprio processo de desenvolvimento. Observou-se um deslocamento significativo no foco atencional do desenvolvedor: das minúcias sintáticas e implementações algorítmicas para considerações conceituais e decisões arquiteturais de mais alto nível. Esta reconfiguração pode ser interpretada como uma forma de "liberação

cognitiva", onde o assistente absorve parte da carga mental associada à codificação, permitindo ao desenvolvedor operar predominantemente no nível do design e da resolução de problemas.

A dinâmica do processo também se alterou substancialmente, tornando-se mais exploratória e iterativa. A capacidade do assistente de gerar rapidamente implementações funcionais facilitou a experimentação com abordagens alternativas - uma característica particularmente valiosa no desenvolvimento do sistema de análise probabilística, onde diferentes configurações dos modelos podiam ser rapidamente implementadas e avaliadas. Este aspecto contrasta com a abordagem tradicional, geralmente mais linear e comprometida com decisões iniciais devido ao custo associado a mudanças tardias.

Notou-se também uma transformação na prática de documentação e na expressão de intenções. A necessidade de comunicar claramente ao assistente a funcionalidade pretendida - frequentemente através de comentários detalhados ou docstrings elaboradas - resultou em uma documentação mais rica e em uma reflexão mais sistemática sobre os requisitos e comportamentos esperados antes da implementação. Esta "programação por intenção" representa uma inversão interessante do fluxo tradicional onde a documentação frequentemente é adicionada após a codificação.

As limitações da assistência do LLM ficaram evidentes em aspectos específicos do desenvolvimento. Observou-se que o assistente ocasionalmente gerava código sintaticamente correto mas semanticamente inapropriado para requisitos específicos do domínio, particularmente em componentes centrais da lógica de negócio. No sistema de análise de desmatamento, por exemplo, as sugestões para interpretação dos resultados dos modelos probabilísticos frequentemente requeriam refinamento substancial para alinhamento com o conhecimento especializado. Esta limitação sugere que o valor da assistência varia significativamente conforme o componente, sendo máximo para implementações padronizadas e técnicas, e mais restrito para lógicas específicas de domínio.

Foi observado também que a eficácia da assistência dependia fortemente da qualidade da "comunicação" estabelecida com o assistente. Descrições vagas ou ambíguas resultavam em sugestões igualmente imprecisas, enquanto especificações detalhadas e contextualizadas geravam implementações notavelmente alinhadas com a intenção original. Esta correlação sugere a emergência de uma nova competência profissional: a capacidade de articular requisitos e intenções de forma que maximize a utilidade das sugestões geradas por IA.

5 Implicações para a Prática Profissional

A experiência documentada neste trabalho suscita reflexões importantes sobre as implicações da assistência de LLMs para a prática profissional. Longe de representar uma simples ferramenta de produtividade, estas tecnologias parecem catalisar uma reconfiguração mais profunda da relação entre desenvolvedor e código, e potencialmente do próprio perfil de competências valorizadas no campo.

Os ganhos de produtividade observados sugerem que ferramentas como GitHub Copilot podem atuar como poderosos amplificadores das capacidades do desenvolvedor, permitindo que um único profissional produza soluções de complexidade e escopo tradicionalmente associados a equipes maiores. Esta amplificação não se restringe a aspectos quantitativos, mas estende-se à qualidade e robustez das implementações, potencialmente elevando o patamar médio de qualidade mesmo para desenvolvedores com menos experiência.

Simultaneamente, observa-se um deslocamento no centro de gravidade da expertise valorizada: de conhecimento sintático e memorização de APIs para capacidade conceitual,

pensamento arquitetural e articulação clara de intenções. Este fenômeno sugere uma possível evolução do papel do desenvolvedor na direção do que poderíamos denominar "arquiteto-comunicador- um profissional cuja principal função é conceber soluções em alto nível e comunicá-las efetivamente ao assistente artificial para implementação.

As limitações identificadas revelam que, no atual estágio evolutivo, estas tecnologias funcionam mais como complementos do que como substitutas do conhecimento especializado. A necessidade persistente de refinamento humano, particularmente em lógicas específicas de domínio, sugere que o valor máximo emerge da combinação entre as capacidades generativas do LLM e o discernimento contextual do desenvolvedor humano - uma forma de simbiose cognitiva onde cada parte contribui com suas fortalezas complementares.

Por fim, o aparente paradoxo da "liberação cognitiva" merece consideração: enquanto o assistente alivia a carga mental associada à codificação detalhada, observa-se concomitantemente uma elevação no nível de engajamento conceitual e criativo. Isto sugere que estas tecnologias podem não apenas aumentar a produtividade, mas potencialmente enriquecer a experiência profissional, permitindo que desenvolvedores dediquem maior proporção de seu tempo e energia a aspectos intrinsecamente mais desafiadores e gratificantes do processo criativo.

6 Conclusão

A experiência de desenvolvimento com assistência de GitHub Copilot, documentada neste trabalho, evidencia o potencial transformador dos LLMs aplicados à engenharia de software. A análise da implementação realizada - um sistema de análise probabilística de desmatamento - revela que a contribuição destas tecnologias transcende a mera aceleração do processo de codificação, induzindo transformações qualitativas na natureza do desenvolvimento e potencialmente reconfigurando o perfil de competências valorizadas na profissão.

Os benefícios observados foram multifacetados: redução significativa no tempo de implementação, melhoria na qualidade e robustez do código, facilitação de testes abrangentes, e liberação da capacidade atencional do desenvolvedor para questões conceituais e arquiteturais. Simultaneamente, identificaram-se limitações importantes, particularmente na compreensão de domínios específicos e na consistência de soluções complexas, evidenciando que estas tecnologias complementam, mas não substituem, o discernimento humano.

A experiência sugere que estamos testemunhando não apenas a introdução de uma nova ferramenta no arsenal do engenheiro de software, mas potencialmente o início de uma transformação paradigmática na relação entre desenvolvedor e código - uma evolução de "escrever código" para "dirigir sua geração". Esta mudança fundamental demanda tanto adaptações nas práticas profissionais e formativas quanto reflexões sobre a natureza da criatividade e da expertise em um contexto onde a implementação técnica é crescentemente mediada por inteligência artificial.

Longe de representar uma ameaça à relevância do desenvolvedor humano, a experiência documentada sugere uma revalorização de capacidades distintivamente humanas - pensamento conceitual, design holístico, julgamento contextual - em um cenário onde aspectos mais mecânicos e previsíveis da programação são progressivamente automatizados. Nesta perspectiva, ferramentas como GitHub Copilot podem ser vistas não como substitutas, mas como amplificadoras do potencial criativo humano, abrindo novas fronteiras de possibilidades para o que um desenvolvedor pode conceber e implementar, funcionando efetivamente como "um núcleo de conteúdo à disposição da sua curiosidade."