

Acelerando o Aprendizado em Engenharia de Software: Uma Abordagem Pedagógica com Protótipos Gerados por IA na Educação Profissional e Tecnológica

José Anderson Carvalho Brasil

¹Instituto Federal de Educação, Ciência e Tecnologia do Amapá (IFAP)

jose.brasil@ifap.edu.br

Abstract. *The teaching of Software Engineering faces challenges in integrating theory and practice. This article presents a pedagogical approach that uses generative artificial intelligence to support the construction of prototypes based on functional and non-functional requirements, agile methodologies, and iterative prototyping. Applied in a Computer Science teaching degree program, the proposal involved developing a ticketing system with AI support to accelerate coding. The results indicate advances in conceptual understanding, student engagement, and the development of competencies aligned with Professional and Technological Education.*

Resumo. *O ensino de Engenharia de Software enfrenta desafios na integração entre teoria e prática. Este artigo apresenta uma abordagem pedagógica que utiliza inteligência artificial generativa para apoiar a construção de protótipos baseados em requisitos funcionais e não funcionais, metodologias ágeis e prototipação iterativa. Aplicada na Licenciatura em Informática, a proposta envolveu o desenvolvimento de um sistema de chamados com apoio da IA para acelerar a codificação. Os resultados apontam avanços na compreensão conceitual, no engajamento dos estudantes e no desenvolvimento de competências alinhadas à Educação Profissional e Tecnológica.*

1. Introdução

O componente curricular em Engenharia de Software demanda que os estudantes desenvolvam não apenas conhecimentos teóricos sobre requisitos, arquitetura e processos de desenvolvimento, mas também competências práticas voltadas à resolução de problemas técnicos e à implementação de soluções. No entanto, observa-se que, tradicionalmente, o ensino da área tende a dissociar-se da realidade do desenvolvimento de sistemas, adotando abordagens teóricas.

Esse desafio é sensível na Educação Profissional e Tecnológica (EPT), que tem como propósito central a integração entre o saber científico e o saber técnico, conforme estabelecido nas diretrizes do Catálogo Nacional de Cursos Superiores de Tecnologia [Brasil 2008]. Para [Ciavatta et al. 2005], essa articulação é fundamental para a formação de profissionais críticos, capazes de compreender e intervir nos processos produtivos com base em uma perspectiva integrada de educação e trabalho. A EPT exige práticas pedagógicas que articulem teoria e prática de maneira situada, promovendo a aprendizagem

a partir de experiências concretas e do desenvolvimento de competências técnicas e reflexivas.

Nesse contexto, este artigo apresenta uma proposta metodológica que explora o potencial da inteligência artificial generativa como recurso para o ensino de Engenharia de Software. A abordagem baseia-se na construção de protótipos funcionais com suporte de IA, permitindo que os estudantes compreendam e apliquem conceitos fundamentais como requisitos funcionais e não funcionais, metodologias ágeis e prototipação iterativa. A estratégia foi aplicada em um curso de Licenciatura em Informática, com resultados que apontam para benefícios pedagógicos relevantes no que se refere ao desenvolvimento de competências técnicas e reflexivas.

2. Fundamentação Teórica: Aprendizagem Tecnológica e Mediação Digital na EPT

A EPT, enquanto modalidade educacional voltada à formação técnico-científica articulada à realidade produtiva, demanda práticas pedagógicas ativas, baseadas na resolução de problemas e na construção colaborativa do conhecimento [Ciavatta et al. 2005]. Essa perspectiva dialoga com os princípios da construção ativa do conhecimento defendidos [Papert 1980] e com as práticas de fabricação digital e autoria tecnológica discutidas por [Blikstein 2013].

Sob a perspectiva histórico-cultural de [Vygotsky 2007], tais tecnologias funcionam como ferramentas mediadoras do desenvolvimento cognitivo, possibilitando a construção de significados por meio da interação social e da mediação simbólica. Ao externalizar parte do esforço técnico na ferramenta, o aluno pode se concentrar em processos de ordem superior, como a análise e o design. Dessa forma, a interação com a tecnologia não apenas gera um produto, mas reestrutura o próprio pensamento do estudante, que internaliza novas estratégias para a resolução de problemas.

Autores como [Valente 2019] e [Kenski 2012] defendem que as tecnologias digitais não devem ser compreendidas apenas como ferramentas de apoio ao ensino, mas como estruturantes de novos paradigmas pedagógicos, capazes de transformar as relações entre ensinar e aprender. Nesse contexto, a inteligência artificial generativa contribui para romper com o modelo transmissivo tradicional, ao favorecer práticas em que os estudantes se tornam coautores do processo de construção do conhecimento e desenvolvem pensamento computacional por meio da interação com sistemas inteligentes.

Nesse sentido, [Holmes et al. 2022] destacam que o uso da IA na educação deve ser pensado não apenas em termos de eficiência tecnológica, mas a partir de princípios pedagógicos e éticos, de modo que os estudantes possam desenvolver autonomia, senso crítico e compreensão dos próprios processos de aprendizagem mediados por algoritmos. A responsabilidade pedagógica, portanto, desloca-se para o desenho de atividades que exijam a validação, a curadoria e a integração criativa das informações.

Considerando essas potencialidades da IA no processo formativo, torna-se relevante integrá-la a práticas pedagógicas que dialoguem com o mundo do trabalho e as metodologias contemporâneas de desenvolvimento de software. Assim, a prototipação como prática pedagógica aproxima o ensino das metodologias utilizadas na indústria de software, como o desenvolvimento iterativo, o design centrado no usuário e as práticas

ágeis [Beck et al. 2001]. Tais abordagens estão alinhadas à lógica da EPT, que privilegia a aprendizagem situada e a atuação profissional crítica.

3. Metodologia

3.1. Caracterização da Proposta Pedagógica

A proposta foi desenvolvida na disciplina de Engenharia de Software, ofertada no curso de Licenciatura em Informática de uma instituição pública. Participaram 22 estudantes, todos cursando o quarto semestre do curso. A intervenção ocorreu em um único encontro presencial, com duração total de quatro horas/aula, organizado em quatro momentos distintos de aprendizagem.

O objetivo pedagógico foi promover a aplicação prática dos conteúdos abordados em aula, utilizando inteligência artificial generativa (ChatGPT) como ferramenta de mediação para prototipação de um sistema de chamados. A abordagem enfatizou a interpretação de requisitos, o uso de ciclos ágeis e a construção de soluções web com tecnologias básicas (HTML, CSS, JavaScript).

3.2. Estrutura da Atividade

A estrutura da atividade foi organizada em quatro etapas sequenciais para conectar teoria e prática. Na primeira etapa foi desenvolvido a análise e categorização de requisitos funcionais e não funcionais, levando os estudantes a refletir sobre as implicações técnicas do projeto. Na segunda etapa, os grupos utilizaram IA para a geração inicial de código, um processo que exigiu a redação de prompts claros e a análise crítica dos resultados. A terceira etapa foi dedicada ao ciclo de iteração e refinamento, na qual os códigos foram progressivamente testados e ajustados, permitindo explorar conceitos como desenvolvimento incremental e modularização. Por fim, a atividade culminou com a apresentação dos protótipos e uma avaliação crítica, onde os estudantes justificaram suas decisões técnicas e discutiram as potencialidades e os limites do uso da IA, conforme a rubrica da Tabela 1.

Tabela 1. Rubrica de Avaliação da Atividade

Critério	Nível 3: Competente	Nível 2: Suficiente	Nível 1: Insuficiente
1. Requisitos e Prompt	Traduz de forma eficaz os requisitos em prompts claros e detalhados, gerando um código-base relevante.	Traduz os requisitos de forma genérica, gerando um código que necessita de grandes ajustes.	Não consegue traduzir os requisitos em prompts funcionais.
2. Iteração e Refinamento	Refina ativamente o código da IA, corrigindo bugs e adicionando melhorias significativas (ex: usabilidade, responsividade)	Realiza apenas correções mínimas para que o código funcione, com pouco ou nenhum refinamento adicional.	Entrega o código gerado pela IA praticamente sem alterações, mesmo com falhas evidentes.
3. Protótipo Final	Entrega um produto final funcional, coeso e que atende aos requisitos essenciais da atividade.	Entrega um produto parcialmente funcional, com falhas notáveis ou requisitos importantes ausentes.	O produto final não é funcional ou não corresponde ao que foi solicitado.
4. Análise Crítica	Justifica com clareza as decisões técnicas e reflete de forma crítica sobre o papel (vantagens e limites) da IA no processo.	Descreve o que foi feito, mas com pouca profundidade na justificativa das decisões ou na análise do uso da IA.	Não consegue explicar o processo de desenvolvimento nem as decisões tomadas.

4. Aplicação Prática: Protótipos Gerados

Durante a atividade, os estudantes desenvolveram um sistema web com funcionalidades básicas, como autenticação de usuários, registro e visualização de chamados. A IA foi utilizada para gerar estruturas iniciais de código, mas os discentes assumiram o papel de refinadores e integradores da solução.



The screenshot shows the 'Consultar Chamados' (View Tickets) page of a ticket system. On the left is a sidebar with links: Início, Registrar Chamado, Consultar Chamados, Histórico de Chamados, Configurações, Cadastrar Usuário, and Sair. The main area has a green header bar with the title 'Consultar Chamados' and a 'Filtrar e visualize os chamados registrados.' sub-instruction. Below this is a form with dropdowns for 'Status do Chamado' (Todos), 'Categoria do Chamado' (Todos), and date ranges 'Data Inicial' and 'Data Final', both set to 'aaaa/mais/aaaa'. A green 'Filtrar Chamados' button is at the bottom of the form. Below the form is a table with columns: Título, Categoria, Status, Data de Abertura, Prioridade, and Detalhes. Two rows are visible: 'Problema no sistema de login' (Suporte Técnico, Em Andamento, 29/01/2025, Alta, Ver Detalhes) and 'Erro de cobrança no sistema financeiro' (Financeiro, Resolvido, 18/01/2025, Baixa, Ver Detalhes).

Figura 1. Interface do formulário de chamados

Fonte: Elaborado pelos discentes

A interface foi funcional, mas exigiu ajustes para melhorar usabilidade e responsividade.



The screenshot shows the refined version of the ticket system. The left sidebar is identical to Figure 1. The main area has a header 'Bem-vindo ao Sistema de Chamados' with the sub-instruction 'Gerencie seus chamados de maneira rápida e eficiente.' Below this are four cards: 'Total de Chamados' (5), 'Chamados em Andamento' (3), 'Chamados Resolvidos' (1), and 'Chamados Aguardando Resposta' (1). At the bottom are two buttons: 'Registrar Novo Chamado' (green) and 'Consultar Chamados' (blue). The footer contains the text '© 2025 Sistema de Chamados'.

Figura 2. Versão refinada com responsividade e organização modular

Fonte: Elaborado pelos discentes

Os discentes aprimoraram o layout com CSS responsivo. A solução reflete a apropriação dos princípios de modularidade e acessibilidade discutida em aula.

5. Resultados e Discussão

A análise dos resultados, baseada nas produções dos estudantes e em observações diretas do docente, indica que a abordagem pedagógica foi altamente eficaz em promover tanto a aprendizagem significativa de conceitos quanto o engajamento prático. Os principais ganhos podem ser discutidos em três eixos centrais.

Compreensão Ampliada de Requisitos: Um dos avanços mais notáveis foi a profundidade com que os estudantes compreenderam os requisitos de software. A necessidade de traduzir especificações funcionais e não funcionais em prompts precisos para a IA os compeliu a superar uma postura de recepção passiva, engajando-os em um exercício cognitivo de interpretação, decomposição e formalização lógica. Esse processo de "tradução semiótica", onde uma necessidade humana é convertida em uma instrução computacional, está alinhado à perspectiva construtivista de [Papert 1980], na qual o conhecimento é forjado ativamente quando o aprendiz "ensina" a máquina. A aprendizagem tornou-se significativa porque os conceitos de requisitos deixaram de ser abstratos e passaram a ser a matéria-prima para uma ação concreta.

Vivência Autêntica do Ciclo Ágil: Adicionalmente, a metodologia proporcionou uma vivência autêntica do ciclo de desenvolvimento ágil. O fluxo rápido de geração de código pela IA, seguido por testes imediatos e refinamentos em sala de aula, simulou um ambiente de desenvolvimento incremental e responsivo, contrastando fortemente com abordagens acadêmicas tradicionais, muitas vezes lineares e com ciclos de feedback mais longos. Essa prática imersiva permitiu aos estudantes experimentar na prática princípios do Manifesto Ágil [Beck et al. 2001], como a valorização de software funcional em detrimento da documentação exaustiva e a capacidade de adaptação a mudanças, competências essenciais para o mundo do trabalho.

Desenvolvimento de Autonomia e Criticidade: Contrariando a preocupação de que a IA poderia reduzir o esforço intelectual, a atividade demonstrou fomentar a autonomia e o pensamento crítico. Embora a IA tenha gerado o código-base, a responsabilidade pela validação, depuração, integração e, crucialmente, pela qualidade e coerência do produto final permaneceu integralmente com os estudantes. Isso efetivamente deslocou o foco da mera digitação de código para competências de ordem superior, como a avaliação de soluções, o design de interfaces e a tomada de decisões arquitetônicas. Tal resultado dialoga diretamente com os objetivos da EPT de formar profissionais reflexivos [Ciavatta et al. 2005] e com a visão de [Holmes et al. 2022], que defendem um uso da IA para capacitar os alunos a se tornarem gestores críticos do processo tecnológico, e não apenas consumidores de tecnologia.

Em suma, os relatos dos participantes e as observações em sala confirmam que a IA generativa, quando estrategicamente integrada a uma proposta pedagógica, não substitui o conhecimento técnico fundamental. Pelo contrário, atua como uma poderosa ferramenta de mediação, conforme teorizado por [Vygotsky 2007], que permite superar barreiras cognitivas. Ao reduzir o tempo gasto com a sintaxe inicial da codificação, a abordagem otimizou o tempo de aula para discussões conceituais e para o desenvolvimento de competências analíticas e críticas, facilitando a transposição didática de conceitos abstratos da Engenharia de Software e elevando o engajamento discente.

6. Considerações Finais

Este estudo demonstrou que a integração da inteligência artificial generativa na prototipação de software constitui uma abordagem pedagógica de alto impacto para o ensino de Engenharia de Software no contexto da Educação Profissional e Tecnológica

(EPT). A IA permitiu que o tempo em sala de aula fosse otimizado para o desenvolvimento de competências, como a análise crítica de requisitos e o refinamento iterativo de soluções.

A experiência validou que é possível transpor a barreira entre teoria e prática de forma ágil, engajadora e alinhada às metodologias contemporâneas do mundo do trabalho. Ficou evidenciado que a abordagem não visa substituir o conhecimento fundamental, mas sim potencializá-lo, reforçando a premissa de que a sinergia entre a inteligência humana e a artificial enriquece o processo formativo.

Para a consolidação desta proposta, trabalhos futuros devem superar as limitações deste estudo exploratório, avançando para investigações longitudinais sobre a autonomia discente, testes de escalabilidade em projetos complexos (com APIs e bancos de dados) e a validação de novos instrumentos de avaliação para competências desenvolvidas em ambientes assistidos por IA.

Referências

- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., and Thomas, D. (2001). *Manifesto for agile software development*. <https://agilemanifesto.org>. Acesso em: 19 jul. 2025.
- Blikstein, P. (2013). *FabLabs: Of machines, makers and inventors*. Transcript Publishers, Munich.
- Brasil (2008). *Catálogo Nacional de Cursos Superiores de Tecnologia*. Ministério da Educação, Secretaria de Educação Profissional e Tecnológica, Brasília.
- Ciavatta, M., Frigotto, G., and Ramos, M. (2005). *A formação integrada: a escola e o trabalho como lugares de memória e de identidade*. Cortez, São Paulo.
- Holmes, W., Tuomi, I., Persson, O.-P., Taddeo, M., and Floridi, L. (2022). *Artificial Intelligence in Education: Promise and Implications for Teaching and Learning*. UNESCO, Paris.
- Kenski, V. M. (2012). *Tecnologias e ensino presencial e a distância*. Papirus, Campinas.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, New York.
- Valente, J. A. (2019). *A construção do conhecimento na era digital*. Unicamp/NIED, Campinas.
- Vygotsky, L. S. (2007). *A formação social da mente*. Martins Fontes, São Paulo.