ENGENHARIA DE SOFTWARE · UnB

A IA escreve.

Quem garante?

O novo eixo da Engenharia de Software

Do gargalo da revisão ao seu futuro profissional



PARA Engenharia de Software

UnB · 2026

Produzido pelo Prof. Dr. Sergio Antônio Andrade de Freitas • CEDIS / UnB

Um protótipo em 10 minutos… é um produto pronto?

PROTÓTIPO

10 min

Gerar uma demonstração é rápido.

≠

PRODUTO

? dias

Validar, testar, revisar e operar.

ATENÇÃO Velocidade para gerar ≠ capacidade de entregar.

Uma pesquisa grande. Uma pergunta simples.

655

empresas

720.542

trabalhadores

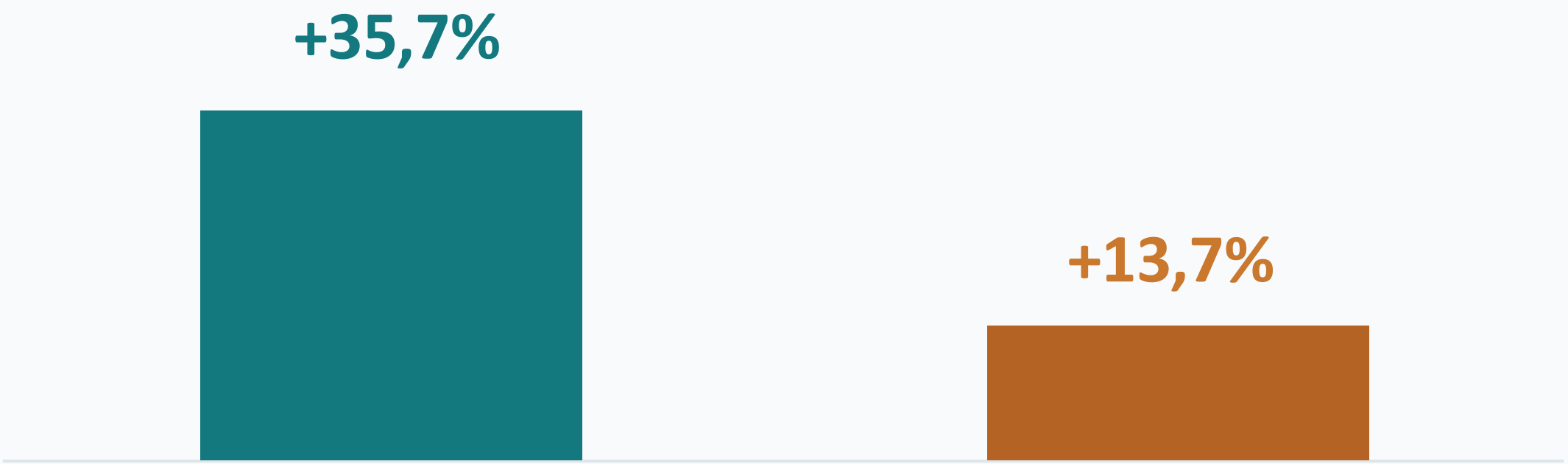
445 mi

eventos de trabalho

GitHub **Jira** **Google Calendar** 2021–2026

ATENÇÃO Estudo observacional por adoção em momentos diferentes; não é experimento aleatório.

Mais código. Muito menos ganho na entrega.



ATENÇÃO O estudo não comprova ganho significativo em entregas finais.

A velocidade aumenta. A fila apenas muda de lugar.

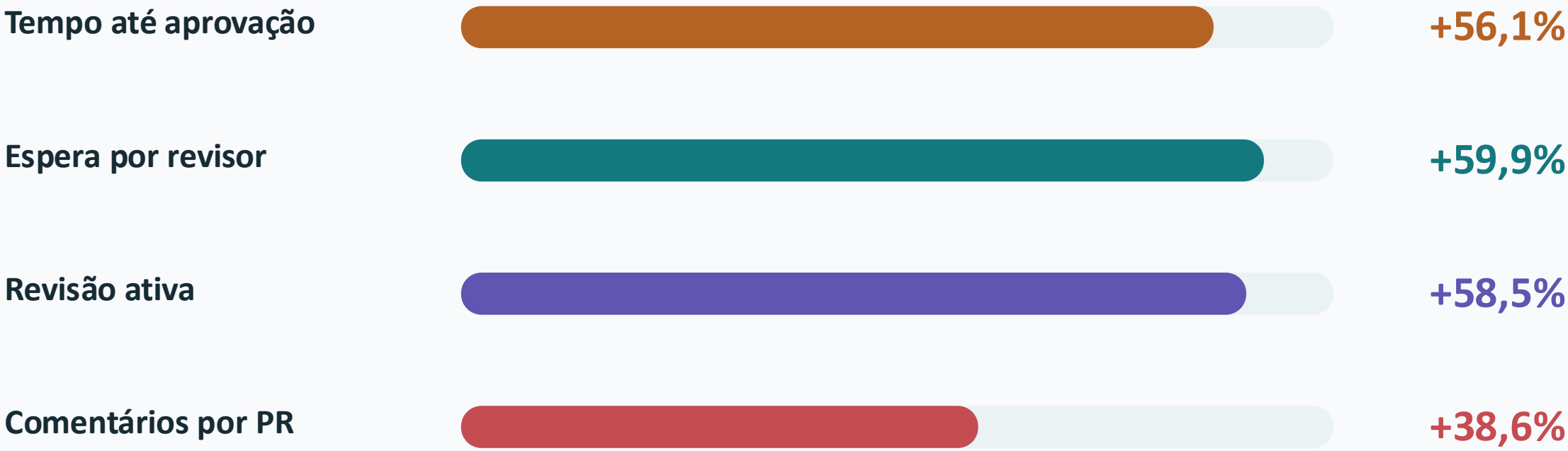


O gargalo não desapareceu: passou da escrita para a revisão.

ATENÇÃO O desempenho de um processo depende da etapa limitante.

Modelo conceitual; Chen & Stratton (2026)

O que aconteceu com a revisão de código?



Aumento observado após adoção de agentes de IA

Mais IA na revisão resolve automaticamente?

79,1%

empresas usaram IA em revisão

23,3%

dos comentários foram de IA

10,8%

dos PRs tiveram comentário de IA

ATENÇÃO As porcentagens têm denominadores diferentes. A decisão de integração continua crítica.

Cuidado: indício não é prova.

O estudo mede

- PRs e commits
- Tempo de espera e revisão
- Comentários de revisão
- Issues resolvidas e emprego

Ele não comprova

- Que cada comentário seja um defeito
- Que IA eliminou vagas no período
- Que PR de 300 linhas é ótimo
- Que os resultados valem para todas as firmas

ATENÇÃO Associe as afirmações ao tipo de evidência: dado, modelo ou conselho.

A proposta do vídeo: menos pressa, mais engenharia.

01

Planejar

Requisitos, arquitetura, tarefas claras

02

Incrementar

Mudanças pequenas e coesas

03

Explicar

No PR: o que mudou e por quê

04

Testar

Evidências antes da revisão humana

ATENÇÃO “300 linhas por PR” é uma heurística, não um limite cientificamente validado.

A IA amplifica o processo que encontra.

Processo sólido

Requisitos claros
Testes e controle de versão
Feedback rápido

Processo frágil

Regras ambíguas
Poucos testes
Retrabalho e filas

IA + **PROCESSO** → **RESULTADO**

DORA / Google Cloud, State of AI-assisted Software Development (2025)

O eixo da profissão está se deslocando.

FOCO TRADICIONAL

Escrever mais código

</>

FOCO EMERGENTE

Decidir → verificar → integrar → entregar

ATENÇÃO A competência de programar continua necessária para avaliar o que foi gerado.

O código funciona. Mas resolve o problema?

A IA implementou uma funcionalidade solicitada.

EXECUTA 01

Sem erros aparentes

É VERIFICÁVEL 02

Testes e contraprovas

É ADEQUADA 03

Regras, usuários e riscos

ATENÇÃO Passar nos testes disponíveis não equivale a atender a todos os requisitos.

Você aprovaria este PR gerado por IA?

Requisitos?

A mudança atende à necessidade real?

Arquitetura?

Preserva contratos e responsabilidades?

Testes?

Evidências independentes dos erros possíveis?

Riscos?

Segurança, regressões e reversibilidade?

ATENÇÃO O profissional responde pela integração — mesmo quando o agente escreve.

Seu currículo é um mapa de responsabilidades.

BASE

Algoritmos · Programação
OO · Métodos

DECISÕES

Requisitos · Arquitetura
Testes · Qualidade

INTEGRAÇÃO

IHC · Paradigmas
Evolução · Produto

PROJETOS INTEGRADORES · ESTÁGIO · TCC

ATENÇÃO As disciplinas se conectam. Use cada uma para construir julgamento de engenharia.

Antes do código: decidir o que precisa existir.

Requisitos de Software

Entrevistar, esclarecer ambiguidades e definir critérios de aceitação.

Arquitetura e Desenho

Escolher estruturas, contratos, limites e justificar decisões técnicas.

PERGUNTA-CHAVE → Estamos construindo a solução certa?

ATENÇÃO Planejamento de qualidade reduz retrabalho antes da geração.

Depois do código: demonstrar que funciona.

Testes de Software	Produzir contraprovas e proteger contra regressões.
Qualidade de Software 1	Medir resultado, risco e retrabalho — não linhas.
Gerência de Configuração e Evolução	Revisar PRs e integrar alterações com segurança.

ATENÇÃO O código não entra em produção porque “parece correto”. Exija evidências.

Quase centrais? São as que ampliam seu alcance.

Fundamentos de Programação

Algoritmos, leitura de código e depuração

Interação Humano-Computador

Usuário, confiança e carga cognitiva

Paradigmas de Programação

Escolher abstrações e estilos adequados

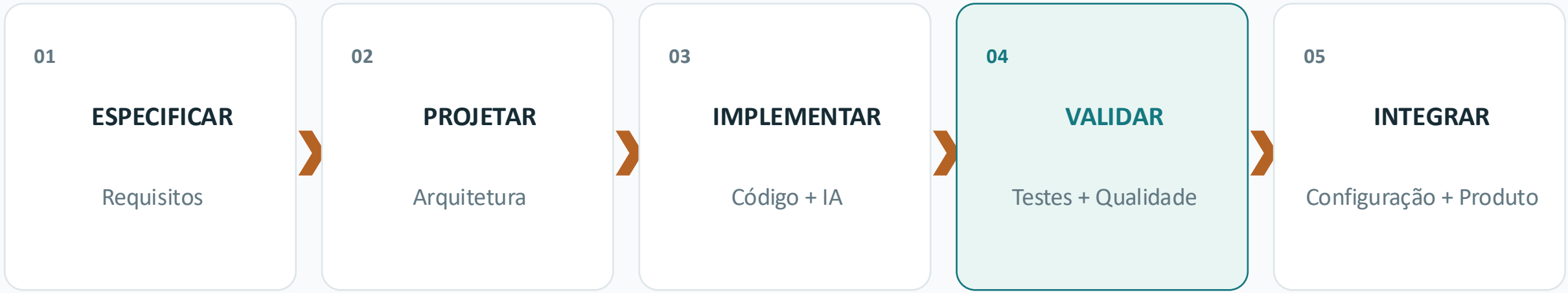
Métodos de Desenvolvimento

Trabalho incremental e colaboração

Engenharia de Produto

Valor, prioridades e decisões de negócio

Não são matérias isoladas. São um único sistema.



Projetos Integradores / Estágio / TCC

A IA já reduziu as contratações? Calma com o dado.

OBSERVADO

+3,1%

assistentes

–2,6%

agentes

Ambos sem significância estatística

SIMULAÇÃO

–13,2%

emprego total no cenário modelado

Não é demissão observada nem previsão certa.

ATENÇÃO Nos dados, não houve queda detectável no emprego; modelos dependem de hipóteses.

Não é só sobre vagas. É sobre o trabalho.

FORMULAR

O problema e os requisitos

DECIDIR

A arquitetura e os compromissos

VERIFICAR

A qualidade, os testes e os riscos

COORDENAR

Pessoas, ferramentas e entregas

ATENÇÃO Seu diferencial não será a quantidade de código — será a qualidade das decisões.

As 6 competências que não cabem em um prompt.

Pensamento crítico

Questionar evidências

Comunicação técnica

Explicar decisões

Colaboração

Revisar sem hostilidade

Autonomia intelectual

Entender sem copiar

Gestão da atenção

Conter a sobrecarga

Ética e responsabilidade

Assumir consequências

Soft skill só existe quando aparece na prática.

Revisar o PR de um colega	Comunicação + colaboração
Contestar a saída da IA	Pensamento crítico + autonomia
Negociar um requisito ambíguo	Escuta + argumentação
Justificar o risco de um merge	Ética + responsabilidade

ATENÇÃO Não basta “trabalhar em grupo”: é preciso deixar evidências da colaboração.

O que você pode construir a partir de hoje?

AGORA / 2027

DOMINE

Fundamentos, testes, PRs explicados

2028–2029

INTEGRE

Projetos reais, arquitetura e revisão cruzada

2030+

LIDERE

Decisões, sistemas, riscos e equipes

ATENÇÃO Não espere o próximo currículo: comece a construir seu repertório agora.

A IA pode ser sua melhor ferramenta de engenharia.



Automatizar o repetitivo

Libera tempo para desafios difíceis



Experimentar mais

Permite comparar hipóteses e soluções



Concentrar o julgamento

Pessoas decidem o que é válido e útil

Ferramenta + conhecimento + responsabilidade

Na próxima entrega, experimente este pacto.

01

Especifique

O que deve funcionar?

02

Decomponha

Qual é a menor mudança coesa?

03

Teste

Que entrada quebraria a solução?

04

Explique

Por que eu deveria aprovar o PR?



A IA pode escrever o código. A responsabilidade pelo software é sua.

Continue a investigação: artigo, vídeo e currículo.

ARTIGO

Chen & Stratton · Harvard (2026)

Estudo de 28 out. 2026

ABRIR FONTE

VÍDEO

Vinícius Pasquantonio · pasquadev

O pesadelo do Code Review

ABRIR FONTE

CURRÍCULO

Engenharia de Software · UnB

Matriz curricular e PPC

ABRIR FONTE

Nota: o vídeo comenta a versão anterior do estudo (718 empresas). Os slides usam a atualização de outubro (655).

Bases para estudar Engenharia de Software com IA.

- **CHEN, F.; STRATTON, J. (2026).** Artificial Intelligence in the Firm: Bottlenecks in Software Production. Working paper (28 out.).
- **UNIVERSIDADE DE BRASÍLIA. (2023/2024).** Projeto Pedagógico do Curso de Engenharia de Software; matriz curricular 6360/2.
- **DORA / GOOGLE CLOUD. (2025).** State of AI-assisted Software Development.
- **WASHIZAKI, H. (org.). (2024; rev. 2025).** SWEBOK Guide, v4.0a. IEEE Computer Society.
- **UNESCO. (2024).** AI Competency Framework for Students.
- **ACM / IEEE-CS / AAAI. (2024).** Computer Science Curricula 2023 (CS2023).
- **PASQUANTONIO, V. (2026).** O pesadelo do Code Review: IA vs. Produtividade. Vídeo, pasquadev.

Autoria e curadoria: Prof. Sergio Antônio Andrade de Freitas • Centro de Estudos, Desenvolvimento e Inovação em Software — CEDIS / UnB