

DURANIUM

CARREIRAS DE TECNOLOGIA EM TEMPOS DE IA

Uma Nova Taxonomia para o Desenvolvimento Profissional

Abril 2026

DURANIUM

Segurança da Informação, Antifraude e IA

www.duranium.io

Sumário

1. A Nova Revolução Industrial — Desta Vez, Digital
2. O Que os Dados Revelam: O Impacto Real no Mercado
3. O Que a IA Faz Bem — e o Que Não Faz
4. As Habilidades que a IA Não Substitui
5. O Papel da Empresa: Pipeline de Talentos e Cultura
6. Recomendações Estratégicas
7. Conclusão: O Futuro Pertence a Quem Orquestra
- A. Apêndice A: Referências e Leituras Recomendadas
- B. Apêndice B: Rastreabilidade das Afirmações

CAPÍTULO 1

A Nova Revolução Industrial — Desta Vez, Digital

Cada onda tecnológica de grande magnitude redefine as competências profissionais valorizadas pelo mercado. A mecanização do século XVIII substituiu o tecelão artesanal pelo operador de máquinas; a eletrificação criou engenheiros elétricos e tornou obsoletos os acendedores de lampião; a computação transformou datilógrafos em programadores e deu origem a uma indústria inteiramente nova.

Agora, estamos diante de mais uma destas inflexões. A popularização dos Large Language Models (LLMs) e das ferramentas de codificação assistida por IA (Claude Code, Cursor, GitHub Copilot, Amazon Kiro, entre outras) está redesenhando o que significa ser um profissional de tecnologia — do estagiário ao arquiteto de soluções, do analista de segurança ao CTO.

Se há cinco anos um desenvolvedor júnior levava uma semana para construir um trecho de código específico sob acompanhamento estrito, hoje, com ferramentas baseadas em LLMs, bastam alguns minutos.

O paralelo com a revolução industrial é preciso: não se trata do desaparecimento de profissões, mas da transformação radical do que cada nível profissional precisa saber e entregar. As habilidades manuais e repetitivas perdem valor relativo; as habilidades de julgamento, contexto e orquestração ganham protagonismo.

1.1 A Eletrificação como Metáfora

Kai-Fu Lee, em *AI Superpowers*, utiliza a metáfora da eletrificação: assim como a energia elétrica não substituiu os trabalhadores, mas transformou todas as indústrias e criou categorias profissionais inteiramente novas, a IA está fazendo o mesmo. A questão não é se você será afetado, mas como.

A questão central não é se a IA vai afetar as carreiras. A questão é: como devemos nos preparar, como empresa e como profissionais, para que essa transformação gere crescimento e não obsolescência?

CAPÍTULO 2

O Que os Dados Revelam: O Impacto Real no Mercado

Os números são contundentes e merecem atenção cuidadosa. Não se trata de alarmismo, mas de compreender a dimensão da mudança para agir de forma estratégica.

2.1 A Contração das Vagas de Entrada

Um estudo da Universidade de Stanford (Stanford Digital Economy Study) constatou que o emprego de desenvolvedores entre 22 e 25 anos caiu quase 20% desde o pico no final de 2022. [Fonte 1]

Contratações de entrada nas 15 maiores empresas de tecnologia caíram 25% de 2023 a 2024, segundo relatório do SignalFire (maio/2024), citado pela IEEE Spectrum. [Fonte 2]

No Reino Unido, vagas de tecnologia em nível de entrada caíram 46% em 2024, com projeções de 53% de queda até o final de 2026 (Denoise Digital, fev/2026). Nos EUA, alguns datasets apontam queda de até 60–70% nas posições de entrada no setor de TI em comparação ao pico. [Fonte 3]

Pesquisa NACE Job Outlook 2026: a avaliação dos empregadores sobre o mercado para recém-formados está no ponto mais pessimista desde 2020; ainda assim 49% consideram o mercado "bom" ou "muito bom". [Fonte 4]

Estudo da Harvard Business School (2025) analisou 285.000 empresas e 62 milhões de trabalhadores nos EUA (2015–2025): quando empresas adotam IA generativa, o emprego júnior cai de 9% a 10% em seis trimestres, enquanto contratações seniores sobem 5%. [Fonte 5]

A Forrester previu para 2026 uma queda de 20% nas matrículas em cursos de Ciência da Computação e duplicação do tempo para preencher vagas de desenvolvedores (Forrester Predictions 2026, citado pelo Infobip Developers Hub). [Fonte 6]

2.2 O Que os Líderes Estão Dizendo

As declarações abaixo foram extraídas de fontes primárias (entrevistas, earnings calls, conferências e podcasts) e também aparecem detalhadas no Apêndice B com links e datas de acesso.

Líder / Empresa	Posição	Fonte
Satya Nadella (Microsoft)	20-30% do código nos repositórios da Microsoft já é gerado por IA. Resultados variam conforme equipe.	LlamaCon (Meta), abr/2025 [Fonte 7]
Sundar Pichai (Google)	IA gera mais de 30% do código novo no Google. IA é acelerador de produtividade, não substituição.	Google Earnings Call, out/2024 [Fonte 8]

Líder / Empresa	Posição	Fonte
Marc Benioff (Salesforce)	Salesforce não contrataria novos engenheiros em 2025 após 30% de ganho de produtividade com IA.	Podcast 20VC, dez/2024 + Earnings Call, fev/2025 [Fonte 9]
Dario Amodei (Anthropic)	Empregos de nível de entrada estão diretamente na mira da automação; em 12 meses IA escreverá a maior parte do código.	Entrevistas, mar/2025 [Fonte 10]
Thomas Dohmke (GitHub)	Manteve programas de estágio; reforçou que juniores continuam vitais para o futuro do desenvolvimento.	CodeConductor, jan/2026 [Fonte 11]
Kevin Scott (CTO Microsoft)	Precisaremos de MAIS empregos de programação, não menos. O papel muda para orquestrar IA.	Entrevista, 2025 [Fonte 12]

Nota: declarações de CEOs de empresas de IA sobre automação de código devem ser lidas no contexto do interesse comercial dessas organizações. Dados empíricos como os da Harvard Business School sugerem impactos reais, mas a magnitude exata ainda está sendo avaliada.

Reflexão

Não é a IA que substitui profissionais — é a IA que substitui o trabalho que nunca deveria ter exigido um engenheiro qualificado em primeiro lugar. A verdadeira questão é: como redefinimos o que cada nível de carreira tem de requisito e precisa entregar?

2.3 O Paradoxo da Produtividade

O artigo "5 Uncomfortable Predictions for Engineering Leaders in 2026", publicado pelo LeadDev em 1º de janeiro de 2026, apresenta dados que sintetizam o paradoxo: assistentes de codificação com IA podem tornar o desenvolvimento 4x mais rápido, mas 10x mais arriscado. A projeção: 1 em cada 5 brechas de segurança será atribuída a código gerado por IA até o final de 2027 (Fonte: Accenture). [Fonte 13]

O artigo complementar do LeadDev ("How AI Will Shape Software Engineering in 2026", jan/2026) reforça essa visão com dados reais: no Spotify, 90% dos desenvolvedores usam IA diariamente, gerando 30% mais code changes por desenvolvedor, mas com preocupações crescentes sobre qualidade e revisão. [Fonte 14]

Adicionalmente, o artigo "GitHub's Copilot may steer you into dangerous waters about 40% of the time – study" reforça essa preocupação, afirmando que 40% do código gerado pelo CoPilot no Github apresenta sérios problemas de segurança. [Fonte 15]

A mesma preocupação aparece no artigo "Asleep at the Keyboard? Assessing the Security of GitHub Copilot's Code Contributions" [Fonte 16]

Velocidade sem revisão qualificada multiplica riscos, não valor. Para empresas que atuam em cibersegurança e IA, este paradoxo é particularmente crítico.

CAPÍTULO 3

O Que a IA Faz Bem — e o Que Não Faz

Para construir um plano de carreira robusto, é fundamental entender com clareza onde a IA é excelente e onde ela falha sistematicamente.

3.1 Capacidades Sólidas da IA

- Gerar código boilerplate, scaffolding, rotas, modelos e APIs
- Automatizar testes unitários e sugerir refatorações em módulos bem definidos
- Reconhecimento de padrões em código, logs e documentação
- Produzir primeiras versões (drafts) para revisão humana
- Documentação automática e resumos de pull requests
- Prototipação rápida de MVPs

3.2 Limitações Persistentes

- Compreender requisitos ambíguos e prioridades conflitantes de produto
- Carregar contexto organizacional e o aprendizado acumulado de incidentes anteriores
- Tomar decisões de julgamento sobre o que é seguro alterar antes de um lançamento
- Projetar sistemas cujas restrições mudam durante a construção
- Assumir responsabilidade quando algo quebra em produção
- Navegar trade-offs entre consistência, disponibilidade, latência e throughput
- Depurar incidentes em produção onde logs mentem e dashboards enganam
- Fazer push-back em decisões de produto antes que gerem dívida técnica

Essa clara definição de capacidades e limitações é o fator chave para a reestruturação da carreira. Dado que a IA pode executar com eficácia tarefas rotineiras como a geração de código base (templates, boilerplates, scripts), o valor do profissional migra para as tarefas que exigem contexto, julgamento e responsabilidade.

“A pergunta deixa de ser 'quem escreve o código?' e passa a ser 'quem é responsável pelo que o código faz?’”

Reflexão

A IA substitui tarefas rotineiras, não profissões inteiras. As tarefas que permanecem exclusivamente humanas são justamente aquelas que diferenciam o profissional pleno ou sênior do júnior tradicional: julgamento, contexto e responsabilidade.

CAPÍTULO 4

As Habilidades que a IA Não Substitui

A pesquisa e as fontes coletadas convergem em um conjunto de competências que se tornam mais valiosas à medida que a IA assume tarefas operacionais. Estas são as habilidades que todo time de engenharia deveria cultivar deliberadamente.

4.1 Problem Framing (Enquadramento de Problemas)

O conceito

Trata-se de algo crucial: identificar o verdadeiro problema antes de mergulhar na construção da solução. Embora pareça uma etapa básica, é onde a maioria das equipes falha e perde tempo valioso. A razão é simples: a demanda raramente chega formulada como um problema — ela chega como uma solução disfarçada.

Na prática, a maioria dos tickets e demandas chega formulada como solução ("adicionar filtros ao dashboard"), não como problema ("os usuários não conseguem identificar anomalias rapidamente"). Um profissional com habilidade de enquadramento transforma a segunda em uma solução mais eficaz — que pode ou não ser adicionar filtros.

Essa habilidade se torna ainda mais valiosa na era da IA porque as ferramentas de geração de código são excepcionais em construir qualquer coisa que se peça, mas são incapazes de questionar se aquilo deveria ser construído.

Como praticar

Antes de iniciar qualquer demanda, o profissional deve ser capaz de responder quatro perguntas:

- Quem é o usuário afetado?
- O que ele está tentando realizar (não o que ele pediu)?
- Qual a frustração ou dor que motiva o pedido?
- Com que frequência isso acontece e qual o impacto quando acontece?

Se não conseguir responder, o problema ainda não está claro — e construir uma solução nesse ponto é o cenário mais caro possível.

Exemplo prático

Um cliente pede: "Precisamos de um relatório semanal com todas as vulnerabilidades encontradas."

Abordagem sem enquadramento: construir o relatório semanal com todas as vulnerabilidades listadas.

Abordagem com enquadramento: por que o cliente quer esse relatório? Porque precisa priorizar o que corrigir primeiro. A solução real pode ser um dashboard com vulnerabilidades ranqueadas por severidade e impacto no negócio, com alertas automáticos para itens críticos — não um PDF de 200 páginas que ninguém lê.

4.2 Pensamento em Métricas, Não em Funcionalidades

O conceito

A maioria dos profissionais de tecnologia pensa em termos de funcionalidades entregues: "implementei o filtro", "refatorei o módulo", "adicionei notificações push". Isso é natural, pois funcionalidades são tangíveis. Mas funcionalidades são *meios*, não *fins*.

Pensamento em métricas é a disciplina de perguntar, antes de construir qualquer coisa: "que indicador deve melhorar com esta entrega?" Quando o profissional faz essa pergunta, o espaço de soluções muda radicalmente.

Na era da IA, essa habilidade ganha uma camada adicional de importância. Ferramentas de geração de código tornam trivial construir qualquer funcionalidade pedida. O diferencial não está mais em quem constrói mais rápido, mas em quem constrói o que realmente move os indicadores.

Como praticar

Toda vez que receber uma demanda, o profissional deve adicionar uma linha antes de começar a trabalhar:

"Esta entrega deve melhorar a métrica _____ de _____ para _____."

Se não conseguir preencher, a demanda precisa ser discutida antes de ser executada. Exemplos de mapeamento:

Funcionalidade pedida	Métrica que deveria melhorar
Melhorar fluxo de onboarding	Taxa de ativação
Adicionar notificações push	Usuários ativos diários (DAU)
Criar dashboard de analytics	Retenção
Melhorar busca interna	Taxa de conclusão de tarefas
Implementar SSO	Tempo de onboarding de novos clientes

Exemplo prático — contexto de Cibersegurança

O time comercial pede: "Precisamos de um relatório mensal bonito para mandar ao cliente com os resultados da varredura de segurança."

Abordagem orientada a funcionalidade: construir template de relatório em PDF com gráficos, exportar automaticamente todo mês, enviar por e-mail.

Abordagem orientada a métrica: qual métrica do cliente esse relatório deveria mover? Provavelmente o tempo de remediação de vulnerabilidades (MTTR). Se o objetivo é que o cliente corrija vulnerabilidades mais rápido, talvez a solução não seja um relatório mensal que chega atrasado, mas alertas em tempo real com priorização automatizada.

Resultado: menos esforço de engenharia no relatório, mais impacto na métrica que importa para o cliente, e um argumento comercial mais forte para renovação do contrato.

4.3 Projeto (Design) de Sistemas e Trade-offs Arquiteturais

O conceito

Design de sistemas é a capacidade de projetar soluções que funcionem não apenas no cenário feliz, mas sob condições reais: escala imprevisível, falhas parciais, requisitos que mudam durante a construção e restrições de negócio que limitam a solução técnica ideal.

Aqui reside a diferença mais nítida entre o que a IA consegue fazer e o que ela não consegue. Uma LLM pode gerar código correto para um endpoint, mas não consegue decidir se aquele endpoint deveria ser síncrono ou assíncrono, se os dados deveriam vir de cache ou de consulta direta, ou se a latência aceitável é 200ms ou 2s — essas decisões dependem de contexto organizacional.

Profissionais que dominam design de sistemas são os mais difíceis de substituir porque o conhecimento que aplicam não está em nenhuma base de treinamento. Ele está na experiência acumulada de ter visto sistemas falharem de formas que a teoria não prevê.

Como praticar

Para cada decisão arquitetural, o profissional deve ser capaz de articular três elementos:

- Qual trade-off está sendo feito (o que se ganha e o que se perde com esta escolha)?
- Em que cenário essa decisão quebraria (qual é o limite)?
- Qual seria o plano B se as premissas se mostrassem erradas?

Um exercício simples para times: ao final de cada design review, perguntar "o que acontece quando isso receber 10x mais tráfego do que previmos?" e "o que acontece quando o serviço X ficar indisponível por 30 minutos?". Se ninguém souber responder, o design ainda não está pronto.

Exemplo prático — contexto Cibersegurança

O time precisa projetar a arquitetura de coleta de dados para uma varredura de segurança em ambiente AWS de um cliente com mais de 200 contas.

Abordagem sem pensamento de trade-off: usar uma Lambda que varre todas as contas sequencialmente, armazena tudo em um bucket S3 e gera o relatório. Funciona no ambiente de testes com 5 contas.

Abordagem com design de sistemas: qual é o trade-off entre varredura paralela (rápida, mas pode estourar rate limits da API da AWS e gerar alertas no cliente) e varredura sequencial (segura, mas pode levar 12 horas)? E se uma conta tiver permissões diferentes e a varredura falhar no meio? O sistema precisa de retry com backoff, dead letter queue para contas que falharam, e uma forma de entregar resultado parcial se o prazo estourar.

A segunda abordagem produz um sistema que funciona na realidade, não apenas na demo.

4.4 Comunicação e Influência

O conceito

Comunicação técnica não é sobre escrever documentação bonita. É sobre a capacidade de traduzir decisões técnicas em linguagem que diferentes audiências conseguem usar para tomar decisões.

Um engenheiro que sabe explicar ao CISO do cliente por que uma vulnerabilidade classificada como "média" pela ferramenta de varredura na verdade representa risco alto no contexto específico daquela empresa está entregando mais valor do que o engenheiro que apenas exporta o relatório gerado. Um profissional que consegue convencer o product manager a adiar uma feature para resolver dívida técnica está protegendo a capacidade futura de entrega do time.

Na era da IA, essa habilidade se torna um multiplicador de carreira por uma razão simples: a IA é capaz de gerar texto, documentação e até apresentações. Mas não é capaz de ler a sala, calibrar o nível de detalhe para a audiência, fazer push-back construtivo quando uma decisão é ruim, ou construir a confiança necessária para que as pessoas ouçam.

Como praticar

Três exercícios concretos que qualquer profissional pode começar a aplicar imediatamente:

- Antes de apresentar uma decisão técnica, reescreva a explicação em uma frase que um gestor não-técnico entenderia. Se não conseguir, você ainda não entendeu o suficiente para explicar.
- Em toda proposta técnica, inclua uma seção "impacto no negócio" com pelo menos uma métrica (custo, tempo, risco, receita). Decisões técnicas apresentadas apenas em termos técnicos são invisíveis para quem decide orçamento.
- Pratique o "push-back construtivo": em vez de dizer "isso não vai funcionar", diga "isso funciona se aceitarmos os seguintes riscos: (...). Sugiro uma alternativa que mitigue esses riscos: (...)".

Exemplo prático — contexto Cibersegurança

O time de engenharia identifica que o ambiente Azure de um cliente possui service principals com permissões excessivas (overprivileged), classificadas como severidade média pela ferramenta de varredura.

Abordagem sem comunicação estratégica: incluir no relatório como "finding de severidade média" com recomendação técnica de redução de escopo. O CISO do cliente olha, vê "média" e prioriza outras coisas.

Abordagem com comunicação e influência: traduzir o achado para linguagem de risco de negócio. "Esses service principals têm acesso de escrita em toda a subscription de produção. Se um deles for comprometido — algo que ocorreu em 23% dos incidentes de cloud em 2025 segundo relatórios do setor

— o atacante teria acesso completo aos dados de clientes e poderia modificar a aplicação em produção."

4.5 Pensamento Crítico sobre Resultados Gerados por IA

O conceito

A IA generativa tem uma propriedade que a torna simultaneamente útil e perigosa: ela é convincente. O código que uma LLM gera parece correto. A documentação que ela produz parece completa. A análise que ela entrega parece fundamentada. E em muitos casos, de fato está. Mas o "muitos casos" não é "todos os casos".

A disciplina do pensamento crítico em relação aos resultados de IA exige encarar qualquer saída como um rascunho qualificado a ser revisado, e não como uma entrega finalizada. Para isto, é crucial possuir conhecimento técnico (mantendo a fluência em pelo menos uma linguagem como requisito essencial) como base para essa revisão.

O profissional que confia cegamente no gerado pela IA é um passivo. O profissional que valida, corrige e melhora o resultado é um ativo insubstituível. A diferença entre os dois é exatamente o espírito crítico que transforma a IA de risco em alavanca.

Como praticar

Antes de aceitar qualquer resposta gerada por IA, aplicar a regra do **VACS** (Verificar, Analisar, Contextualizar, Simplificar):

- **Verificar:** o resultado atende o que foi pedido? Testei com casos limite e entradas inesperadas?
- **Analisar:** entendo por que essa solução funciona? Consigo explicar cada decisão que o código toma? Se não consigo, não devo entregar.
- **Contextualizar:** essa solução se encaixa no nosso contexto específico (arquitetura existente, requisitos de segurança do cliente, padrões do time)?
- **Simplificar:** há complexidade desnecessária? LLMs tendem a gerar soluções mais complexas do que o necessário — o profissional que simplifica sem perder funcionalidade está agregando valor real.

Exemplo prático — contexto Cibersegurança

Um engenheiro usa o Claude Code para gerar uma porção de código Python que automatiza a coleta de achados de segurança via API do Prowler e consolida em um relatório para o cliente.

O que a IA entrega: um script funcional que coleta os dados, formata em JSON e gera um PDF. O código roda, os testes passam.

O que o pensamento crítico revela ao revisar: o script armazena temporariamente as credenciais da API em uma variável de ambiente sem criptografia. O relatório gerado inclui nomes internos de recursos (ARNs) que o cliente não deveria ver porque pertencem à infraestrutura de outro tenant. O tratamento de erros silencia exceções de rate limiting, o que significa que uma varredura parcial pode ser entregue como resultado completo.

Nenhum desses problemas impede o script de funcionar nos testes. Todos eles poderiam causar incidentes sérios em produção — vazamento de credenciais, exposição de dados cross-tenant, relatório falso-negativo entregue ao CISO de um cliente enterprise.

Esse é o "quase" que vale: "a IA quase sempre acerta de primeira". O valor do profissional está inteiramente no que ele faz com esse "quase".

Reflexão

Em cibersegurança e proteção reputacional, código com vulnerabilidades sutis gerado por IA pode ser mais perigoso que código escrito manualmente com mais cuidado. O padrão de revisão precisa ser mais rigoroso, não menos.

CAPÍTULO 5

O Papel da Empresa: Pipeline de Talentos e Cultura

A tentação de curto prazo é clara: se a IA faz o trabalho de juniores, por que contratar juniores? A perspectiva de médio e longo prazo, no entanto, aponta na direção oposta.

5.1 O Risco do Pipeline Vazio

Empresas que eliminem posições juniores em 2026 poderão enfrentar uma escassez severa de talentos de nível pleno e sênior em 3 a 5 anos. O pipeline de talentos é um investimento, não um custo. Organizações que mantêm seus programas de formação terão vantagem competitiva quando essa escassez se materializar.

5.2 A Carga Cognitiva dos Sêniores

Sem juniores para quem delegar tarefas de menor risco, seniores ficam presos em um ciclo de auditoria cognitiva exaustiva: revisar milhares de linhas de código gerado por IA buscando alucinações sutis, sem a válvula de pressão que a delegação proporciona.

5.3 Mentoria Estruturada

Empresas devem estruturar seu plano de carreira em torno de três pilares:

- **Formação Contínua:** cada profissional terá um plano de desenvolvimento com metas técnicas e comportamentais.
- **Mentoria Formal:** seniores supervisionam juniores com encontros regulares, não apenas no dia a dia.
- **Exposição Deliberada:** profissionais são deliberadamente expostos a frentes de trabalho diferentes (cibersegurança, IA, privacidade, dados) para construir amplitude.

Os Três Tipos de Mentor

Técnico: ensina o ofício, revisa código, transfere conhecimento de domínio.

Carreira: ajuda a navegar decisões de carreira, visibilidade e crescimento.

Rede: conecta o profissional a oportunidades, pessoas e contextos além da bolha atual.

5.4 IA como Ferramenta de Nivelamento

Um aspecto frequentemente esquecido: a IA também democratiza o acesso ao conhecimento. Profissionais em início de carreira que sabem usar IA para aprender — não apenas para produzir — podem acelerar significativamente sua curva de aprendizado. A empresa que ensina seus profissionais a aprenderem com IA está investindo no multiplicador mais potente disponível.

CAPÍTULO 6

Recomendações Estratégicas

Com base em toda a pesquisa consolidada neste documento, apresentamos recomendações concretas para elaboração de planos de carreira em TI.

6.1 Adotar uma Taxonomia Expandida de Carreira

Revisar os planos de carreira, adicionando um aspecto claro sobre IA para cada nível. Cada um deve ter critérios claros de competência técnica, competência comportamental e proficiência com ferramentas de IA.

6.2 Integrar Proficiência em IA nos Critérios de Promoção

Proficiência no uso responsável de IA deve ser um critério explícito em cada nível. Não como habilidade isolada, mas como competência transversal que permeia todas as funções: de estagiários que aprendem a formular prompts a seniores que definem políticas de uso de IA nos projetos de clientes.

6.3 Manter e Fortalecer Programas de Estágio

Programas de estágio são um investimento estratégico no pipeline de talentos. Recomendamos: mentores designados por setor, acompanhamento de capacidade, exposição a múltiplas frentes de trabalho, e avaliação tanto técnica quanto cultural.

6.4 Criar um Framework de 1:1s Orientados a Carreira

Implementar um formato estruturado de 1:1s entre liderados e gestores focado em desenvolvimento de carreira, com perguntas como:

- Quais habilidades você desenvolveu este mês?
- Onde você quer estar em 12 meses?
- Qual a maior barreira para seu crescimento?
- Como a IA tem mudado seu dia a dia?

6.5 Métricas de Acompanhamento (Uma Proposta)

Métrica	Descrição	Meta Sugerida
Tempo para produtividade	Dias até um novo contratado/estagiário entregar valor real	< 45 dias
Retenção 12 meses	% de profissionais que permanecem após 1 ano	> 85%
Progressão de nível	% de profissionais que avançam de nível por ano	20–30%
Satisfação com mentoria	NPS interno do programa de mentoria	> 70
Proficiência em IA	% de profissionais com certificação ou demonstração de uso proficiente de IA	> 80%

CAPÍTULO 7

Conclusão: O Futuro Pertence a Quem Orquestra

A pergunta que motivou este documento — "não precisamos mais de juniores?" — tem uma resposta nuançada. Precisamos, sim, de juniores. Mas de juniores diferentes. Profissionais que entrem na carreira já sabendo da necessidade de usar IA como ferramenta, não como substituto do pensamento; que tenham espírito crítico para revisar saídas de LLMs; que consigam enquadrar problemas antes de construir soluções.

Para profissionais plenos e seniores, a IA é um amplificador, não uma ameaça. O profissional que aprende a orquestrar IA para entregar mais, com mais qualidade, será mais valioso do que nunca.

A oportunidade é clara: construir um plano de carreiras que prepare nossos profissionais para esse novo mundo, que valorize tanto a profundidade técnica quanto a amplitude de visão, e que faça da IA uma aliada no desenvolvimento de pessoas — não uma substituta.

“O futuro não pertence a quem compete com a IA. Pertence a quem colabora com ela.”

APÊNDICE A

Referências e Leituras Recomendadas

Artigos e Pesquisas Acadêmicas

- Harvard Business School — Estudo com 285.000 empresas sobre impacto da IA generativa no emprego (2025)
- Stanford Digital Economy Study — Emprego de desenvolvedores 22-25 anos (2025)
- Forrester Predictions 2026 — Previsão de queda de 20% em matrículas de Ciência da Computação
- NACE Job Outlook 2026 — Perspectivas de emprego para recém-formados
- Stack Overflow Developer Survey 2025 — 84% dos desenvolvedores já usam ferramentas de IA
- JetBrains State of Developer Ecosystem 2025 — 85% usam ao menos uma ferramenta de IA
- LeadDev — 5 Uncomfortable Predictions for Engineering Leaders in 2026

Leitura Complementar: Artigos

- The Hustling Engineer — Engineers AI Can't Replace
- The Hustling Engineer — What's the Developer's Edge in 2026?
- The Hustling Engineer — Career Playbook: First 8 Years in Tech
- The Hustling Engineer — 2025 Career Growth Roadmap with AI
- Stack Overflow Blog — AI vs Gen Z: Junior Developer Careers
- CIO.com — Demand for Junior Developers Softens as AI Takes Over
- IEEE Spectrum — AI Shifts Expectations for Entry Level Jobs
- Denoise Digital — The Disappearance of the Junior Developer
- Infobip Developers — AI, Hiring and the Future of Coding in 2026
- Addy Osmani — The Next Two Years of Software Engineering
- LinkedIn — Reuven Cohen: Which Jobs Survive AI?
- DZone — The Truth About AI and Job Loss

Leitura Complementar: Livros

- AI Superpowers — Kai-Fu Lee (a metáfora da eletrificação e o impacto da IA nas carreiras)
- The Pragmatic Programmer — David Thomas & Andrew Hunt (fundamentos atemporais de engenharia de software)
- Team Topologies — Matthew Skelton & Manuel Pais (como organizar times de engenharia modernos)
- Staff Engineer — Will Larson (o caminho para papéis de liderança técnica)
- An Elegant Puzzle — Will Larson (gestão de engenharia em escala)
- The Manager's Path — Camille Fournier (do IC ao CTO: cada transição de carreira em tecnologia)

APÊNDICE B

Rastreabilidade das Afirmações

Este apêndice documenta as fontes primárias de cada afirmação utilizada nos tópicos 2.1, 2.2 e 2.3. Para cada fonte, são indicados: a referência numérica no texto, a origem, a data de publicação, a URL de acesso e a data em que o conteúdo foi consultado para elaboração deste documento.

Fontes do Tópico 2.1 — Contração das Vagas de Entrada

#	Afirmação	Fonte Primária	Data Pub.	Acesso
1	Emprego de devs 22-25 anos caiu ~20% desde pico de 2022	Stanford Digital Economy Study, citado por Stack Overflow Blog (AI vs Gen Z)	Dez/2025	22/03/2026
2	Contratações de entrada nas 15 maiores techs caiu 25% (2023→2024)	SignalFire (maio/2024), citado pela IEEE Spectrum (Top Tech 2026)	Dez/2025	22/03/2026
3	Vagas entry-level UK: -46% em 2024; projeção -53% até 2026; EUA -60/70%	Denoise Digital — The Disappearance of the Junior Developer	Fev/2026	22/03/2026
4	Avaliação de empregadores mais pessimista desde 2020	NACE Job Outlook 2026, citado por IEEE Spectrum	Dez/2025	22/03/2026
5	Adoção de GenAI: emprego júnior cai 9-10% em 6 trimestres (285k empresas)	Harvard Business School (2015–2025), via FinalRound AI	2025	22/03/2026
6	Previsão: -20% matrículas em CS; 2x tempo para preencher vagas de dev	Forrester Predictions 2026, citado pelo Infobip Developers Hub	Fev/2026	22/03/2026

URLs de acesso:

[1] stackoverflow.blog/2025/12/26/ai-vs-gen-z/

[2] spectrum.ieee.org/ai-effect-entry-level-jobs

[3] denoise.digital/the-disappearance-of-the-junior-developer

[4] spectrum.ieee.org/ai-effect-entry-level-jobs (NACE survey)

[5] finalroundai.com/blog/ai-is-making-it-harder-for-junior-developers

[6] infobip.com/developers/blog/ai-hiring-and-the-future-of-coding

Fontes do Tópico 2.2 — O Que os Líderes Estão Dizendo

#	Líder	Contexto da Declaração	Fonte Primária	Acesso
7	Satya Nadella	20-30% do código Microsoft é gerado por IA	Fireside chat com Zuckerberg no LlamaCon (Meta), 29/abr/2025. Reportado por CNBC	22/03/2026
8	Sundar Pichai	IA gera >30% do código novo no Google	Google Earnings Call Q3/2024, out/2024. Referenciado pelo TechCrunch (abr/2025)	22/03/2026
9	Marc Benioff	Salesforce não contratará engenheiros em 2025; 30% de ganho com IA	Podcast 20VC com Harry Stebbings, dez/2024. Confirmado na Earnings Call Q4/2025 (fev/2025)	22/03/2026
10	Dario Amodei	Entry-level jobs estão na mira da automação; em 12 meses IA escreverá maior parte do código	Entrevistas, mar/2025. Citado pelo CodeConductor (jan/2026)	22/03/2026
11	Thomas Dohmke	GitHub manteve programas de estágio; juniores continuam vitais	Citado pelo CodeConductor (jan/2026).	22/03/2026
12	Kevin Scott (CTO Microsoft)	Precisaremos de MAIS empregos de programação; papel muda para orquestrar IA	Entrevista citada pelo The Hustling Engineer (newsletter KB/2025)	22/03/2026

URLs de acesso:

[7] cnbc.com — Nadella says 30% of Microsoft code written by AI

[7b] techcrunch.com — Microsoft CEO says up to 30% of code written by AI

[8] codeconductor.ai — Junior Developers in the Age of AI (Pichai citado)

[9] salesforceben.com — Salesforce Will Hire No More Engineers in 2025

[9b] sfstandard.com — Benioff says Salesforce will hire no engineers

[10] codeconductor.ai — Junior Developers in the Age of AI (Amodei citado)

[11] codeconductor.ai — Junior Developers in the Age of AI (Dohmke citado)

[12] [The Hustling Engineer](https://thehustlingengineer.com) — Engineers AI Can't Replace (Kevin Scott citado)

Fontes do Tópico 2.3 — O Paradoxo da Produtividade

#	Afirmção	Fonte Primária	Data Pub.	Acesso
13	IA coding: 4x mais rápido, 10x mais arriscado;	LeadDev — 5 Uncomfortable Predictions for Engineering Leaders in 2026 (Chris Stokel-Walker)	01/jan/2026	22/03/2026
3	1 em 5 breaches é de código IA (Accenture 2027)			
14	Spotify: 90% devs usam IA; +30% code change/dev; aumento de preocupações de qualidade	LeadDev — How AI Will Shape Software Engineering in 2026 (Niklas Gustavsson, VP Spotify)	13/jan/2026	22/03/2026
15	40% do código gerado pelo CoPilot no Github apresenta sérios problemas de segurança	GitHub's Copilot may steer you into dangerous waters about 40% of the time – study	25/Ago/2021	24/03/2026

#	Afirmação	Fonte Primária	Data Pub.	Acesso
1	Levantamento com base em código gerado	Asleep at the Keyboard? Assessing the	16/dez/2	24/03/20
6	pelo Copilot detectou vulnerabilidades significativas	Security of GitHub Copilot's Code Contributions	021	26

URLs de acesso:

[13] leaddev.com — 5 Uncomfortable Predictions for Engineering Leaders in 2026

[14] leaddev.com — How AI Will Shape Software Engineering in 2026

[15] GitHub's Copilot may steer you into dangerous waters about 40% of the time – study

[16] Asleep at the Keyboard? Assessing the Security of GitHub Copilot's Code Contributions

Data de elaboração deste documento: 22 de março de 2026.

Todas as URLs foram acessadas e verificadas na data indicada. Conteúdos de newsletters (paywalled) foram acessados via e-mail recebido na caixa do autor.

O futuro pertence a quem orchestra.

— Duranium —

Cibersegurança e Proteção Reputacional

contato@duranium.io

www.duranium.io