

Incluindo dicas de top desenvolvedores do  
Google, Microsoft, Spotify, Amazon e mais

# 14 HÁBITOS DE DESENVOLVEDORES ALTAMENTE PRODUTIVOS

Zeno Rocha

# Parte um: Princípios

## Olá mundo

*"Não há elevador para o sucesso, você precisa subir as escadas." - Zig Ziglar*

Comecei a desenvolver software há mais de dez anos. Desde então, construí muitos aplicativos, criei dezenas de projetos de código aberto e escrevi milhares de commits. Além disso, palestrei em mais de uma centena de conferências e tive a oportunidade de conversar com vários desenvolvedores ao longo do caminho.

Tive a sorte de estar em contato com alguns dos melhores engenheiros de software da indústria, mas também conheci muitos programadores que ainda fazem a mesma coisa há anos.

O que separa um grupo do outro? O que há de único nas pessoas que trabalham nas maiores empresas da indústria? O que há de especial nas pessoas que criam os aplicativos mais usados no mundo? Como alguns desenvolvedores podem ser tão prolíficos no trabalho e também fora de seus empregos?

Essas perguntas ficaram na minha cabeça por um longo tempo. Percebi que podia comprar os melhores teclados mecânicos, participar das conferências de tecnologia mais famosas do mundo e aprender todos os novos frameworks. Ainda assim, se eu cultivasse maus hábitos, seria impossível me tornar um desenvolvedor de primeira linha. Por isso, decidi procurar os melhores desenvolvedores que conheço e pedir dicas de como ser mais produtivo.

Este livro não oferece um caminho reto ou fórmula predefinida para o sucesso. Este livro é o resultado de uma busca. Uma busca para descobrir quais hábitos podem ser cultivados para se tornar um melhor engenheiro de software.

## Metodologia

Este não é um livro convencional. Você não encontrará o mesmo formato ou estrutura que um livro comum possui. Na verdade, este livro foi projetado para ser o mais simples e objetivo possível. Você pode seguir a ordem dos capítulos ou lê-los individualmente. Tudo é independente e não requer conhecimentos anteriores.

No final de cada hábito, você encontrará uma seção marcada como "**Perguntas e Respostas**", onde entrevisto desenvolvedores seniores e líderes de tecnologia de várias empresas para entender como eles chegaram lá. Fui atrás de gigantes da tecnologia como Google, Amazon, Microsoft e Adobe. Startups poderosas como GitHub, Spotify, Elastic, Segment, GoDaddy e Shopify. E até organizações estabelecidas, como Citibank, BlackBerry e The New York Times.

Essas pessoas vêm de todo o mundo e têm um histórico bastante diversificado. De São Francisco a Nova York. De São Paulo a Montreal. De Londres a Estocolmo. A ideia é apresentar a você não o ponto de vista de apenas uma pessoa, mas uma coleção de ideias sobre como navegar em sua carreira.

Você também encontrará seções marcadas como "**TODO**", onde recomendo que você reflita sobre determinados tópicos e tome medidas com instruções específicas. Eu recomendo levar alguns minutos para mergulhar neles, pois isso gerará ainda mais conhecimento.

E, finalmente, você pode encontrar algumas seções marcadas como "**Bônus**". Esses são conteúdos extras que eu preparei para você. Eles podem ser encontrados fora do livro e complementam o que você está lendo.

## Por quê hábitos?

Se você já tentou perder peso, sabe como esse processo é frustrante. Você pode se exercitar o máximo que puder por três horas, mas se fizer isso apenas uma vez por semana, o efeito será nulo. O que realmente gera resultados é quando você se exercita várias vezes por semana. Então, alguns meses depois, você começará a perceber mudanças no seu corpo.

Consistência é importante e esse mesmo conceito também se aplica à sua carreira profissional. As coisas levam tempo, e a intensidade nem sempre é a resposta. Os hábitos que você decide cultivar (ou não cultivar) determinarão suas futuras oportunidades de vida. Assim como foi dito no livro *Atomic Habits*:

*"Os hábitos são os juros compostos do auto-aperfeiçoamento. Da mesma forma que o dinheiro se multiplica através dos juros compostos, os efeitos de seus hábitos se multiplicam à medida que você os repete. Eles parecem fazer pouca diferença em um determinado dia e, no entanto, o impacto que eles proporcionam ao longo dos meses e dos anos podem ser enormes. É apenas quando se olha para trás dois, cinco ou talvez dez anos é que o valor dos bons hábitos e o custo dos maus se tornam surpreendentemente aparentes."*

— James Clear

Eu recebo muitos e-mails de outros engenheiros de software perguntando qual linguagem de programação eles deveriam aprender. E embora essa seja uma pergunta muito importante, sinto que a pergunta mais valiosa seria: "Que hábitos eu preciso cultivar para ser eficaz em qualquer linguagem de programação?"

É por isso que decidi focar este livro em hábitos, e não em táticas.

Agora vamos nessa! Você está pronto?

## Hábito 1: Procure os sinais

*"As palavras mais antigas e mais curtas - sim e não - são aquelas que requerem mais reflexão." - Pitágoras*

Todos os dias somos bombardeados com tweets, e-mails e vídeos nos dizendo o que devemos fazer, o que devemos aprender e o que devemos focar. Estamos constantemente diante do FOMO (*the fear of missing out*) ou "o medo de estar perdendo algo". E se estivermos perdendo nosso tempo com uma linguagem de programação errada? E se o framework mais produtivo não for o que você está usando atualmente?



Para mim, tudo começou com a escolha do sistema operacional. Na época, eu tinha uma máquina Windows, mas todo mundo me dizia que os computadores da Apple eram melhores e que eu deveria trocar. No entanto, os preços das máquinas da Apple estavam muito longe da minha realidade de compra, então conseguir um nem era uma possibilidade. Depois de alguns anos, um dos meus chefes me deu a opção de escolher entre Windows e MacOS. Escolhi o MacOS para ver o que tinha de tão incrível nele. Depois de algum tempo usando

MacOS, o Ubuntu começou a se tornar muito popular e todo mundo me dizia que eu deveria mudar para o Ubuntu. Então pensei em experimentar e comecei a usá-lo. Meu ponto aqui não é dizer qual você deve escolher, o ponto aqui é que **não existe a melhor ferramenta**.

Em vez disso, devemos praticar o JOMO (*the joy of missing out*) ou "a alegria de estar perdendo algo", que é principalmente sobre ser feliz e se contentar com o que você já sabe. Dê a si mesmo algum crédito e reflita sobre tudo o que aprendeu até agora. Obviamente, você não deve ser complacente e parar de estudar novas tecnologias. Seria melhor se você encontrasse algum equilíbrio entre praticar suas habilidades existentes e aprender novas.

As pessoas tentarão convencê-lo de qual é o melhor sistema operacional, a melhor linguagem de programação, o melhor framework. Eles vão falar sobre todas as coisas incríveis que a ferramenta deles faz e que a sua ferramenta não faz. A realidade, porém, é que cada ferramenta é diferente e também somos diferentes como usuários. O que é melhor para você, pode não ser o melhor para mim ou para os outros.

Pense nisso como uma estação de rádio que você está tentando sintonizar. Eu sei que você provavelmente não usa mais rádios, mas fique comigo. Imagine que você gira o dial e está apenas captando ruído estático, mas depois de alguns segundos frustrantes, ele finalmente consegue captar um sinal e sintonizar uma estação. O sinal é a informação significativa na qual você realmente está interessado. O ruído estático é apenas a variação aleatória e indesejada que interfere no sinal. É por isso que a autoconsciência é tão importante, você precisa ser capaz de identificar qual é o sinal e o que é apenas ruído.

É crucial entender que o barulho sempre estará lá. Você não precisa necessariamente abandonar as mídias sociais, cancelar as listas de e-mail e parar de assistir vídeos do YouTube. Uma desintoxicação digital pode definitivamente ajudar por um tempo, no entanto, não é uma solução a longo prazo. O que você precisa fazer é escolher o que é relevante para você neste momento da sua carreira.

Aceite o fato de que você simplesmente não consegue aprender tudo. **Lembre-se, desejos são infinitos, necessidades são limitadas.** Aceite o fato de que o novo nem sempre é o melhor. Existem pessoas que trabalham com linguagens de programação arcaicas e ainda ganham muito dinheiro. Pratique diariamente a arte sutil de dizer 'não'. Não à nova biblioteca. Não àquela plataforma mais sofisticada. Diga mais 'não' para poder dizer 'sim' ao que realmente importa para você.



**1 SIM !== 1 NÃO**  
**1 SIM === MUITOS NÃO'S**

Ouçá o barulho, mas preste atenção apenas aos sinais.

// TODO

Crie uma lista de todas as tecnologias e ferramentas que você gostaria de aprender. Agora marque cada uma delas com uma prioridade diferente: "Esta semana", "Próximo mês", "Próximo ano". Sempre que sentir que está perdendo alguma nova tendência brilhante, revise a lista e reorganize a prioridade.

## # Perguntas e Respostas: Como você decide qual tecnologia aprender e investir tempo?

Daniel Buchner (Microsoft):

*"No início da minha carreira, lembro de prestar muita atenção a todos os novos frameworks que apareciam na primeira página do Hacker News. Eu lia sobre qualquer novidade que eles adotassem e passava horas extras mexendo com o framework ou a biblioteca para ter uma noção do que era. É perfeitamente tranquilo se você tiver tempo livre suficiente para explorar e aprender, mas muitas vezes pode levar a fadiga, porque a lista de coisas novas não tem fim.*

*Atualmente, tento me concentrar em algumas coisas importantes que conduzem minhas avaliações:*

- 1. Quais são os requisitos técnicos indispensáveis para qualquer coisa em que estou trabalhando? Isso pode incluir itens como desempenho, UX desejado ou integração com outros sistemas.*
- 2. Quão fácil será para os outros trabalharem no que estou construindo e como será para eles integrarem em quaisquer projetos em que estejam trabalhando?*
- 3. O que estou fazendo está alinhado com a trajetória da web aberta, dos padrões e das especificações que acredito que permanecerão a longo prazo?*

*Os três pontos acima tendem a eliminar muitas das bibliotecas, frameworks e outros utilitários que encontro. Isso me poupou tempo e permitiu que eu me concentrasse mais em entregar o que precisava, em vez de ser pego no turismo de desenvolvimento."*

Addy Osmani (Google):

*"Aceite que você não pode aprender tudo, mas pode aprender o suficiente para ser produtivo. Quando noto que uma ideia, framework ou tecnologia está ganhando força, invisto uma tarde estudando para ter uma ideia de duas coisas: 'Isso melhora minha produtividade?' e 'Isso melhora a experiência do usuário dos tipos de projetos que costumo criar?' Se a resposta para qualquer uma dessas perguntas for afirmativa, eu considero gastar mais tempo aprendendo sobre o framework ou a tecnologia com mais profundidade.*

*Como nosso tempo é limitado, há muitas tecnologias que irei achar convincentes o suficiente para ficar de olho, mas devo pensar na troca sobre aprender e investir em favor de outra coisa. Eu acho que isso é saudável. Originalmente, eu tentei (e usei) o React no primeiro ano em que foi lançado, mas agora o uso regularmente. Gosto de muitas ideias do Svelte, mas como eu já havia aprendido o React, o Preact e o Vue, decidi que era um investimento melhor de tempo subir de nível em áreas completamente diferentes com esse tempo, como a API de animações da Web.*

*Como disse, com o tempo limitado, é bom encontrar um equilíbrio no que você escolhe aprender para manter-se produtivo. Quando você escolhe apenas algumas opções de alto impacto, deve reservar um tempo para se aprofundar na tecnologia que você escolheu. Isso pode ser imensamente útil quando você está tentando criar algo não trivial no mundo real."*

Obrigado por usar o seu tempo lendo esse capítulo gratuito.

Espero que você queira ler o livro inteiro agora :)

Quer ajudar? Compartilhe com os seus amigos!

- Zeno