

# Como se tornar um Hacker

## De MinholiWiki

Este texto foi originalmente escrito em Inglês por Eric Steven Raymond e pode ser encontrado em sua forma original [neste endereço](#). A tradução abaixo foi realizada com o intuito de facilitar a sua leitura para iniciantes e permitir a difusão dos conceitos nele dispostos. As referências em primeira pessoa ao autor e o seu conteúdo pertencem ao autor original.

## Tabela de conteúdo

Por que este documento?.....	2
O que é um Hacker? .....	2
A atitude Hacker .....	3
O mundo está cheio de problemas fascinantes esperando para serem resolvidos. ....	3
Nenhum problema jamais deve ter que ser resolvido duas vezes. ....	4
Tédio e trabalho repetitivo são ruins. ....	4
A liberdade é boa. ....	4
Atitude não substitui competência. ....	5
Habilidades Básicas de Hackear .....	5
Aprenda como programar. ....	5
Pegue um dos Unixes de código aberto e aprenda a usar e executá-lo. ....	6
Aprenda a usar a World Wide Web e escrever HTML. ....	7
Se você não tem Inglês funcional, aprenda. ....	8
Status na Cultura Hacker .....	8
Escreva software de código aberto .....	9
Ajude a testar e depurar software de código aberto .....	9
Publique informações úteis. ....	9
Ajude a manter a infra-estrutura funcionando .....	9
Sirva-se à cultura hacker .....	10
A Conexão Hacker/Nerd .....	10
Pontos por Estilo .....	10
Outros recursos .....	12
Perguntas Feitas com Frequência .....	12
Como posso saber se eu já sou um hacker? .....	12
Você vai me ensinar a hackear? .....	13
Como posso começar, então? .....	14
Quando você tem que começar? É tarde demais para eu aprender? .....	14
Quanto tempo leva para eu aprender a hackear? .....	14
Visual Basic é uma boa linguagem para começar? .....	14
Você poderia me ajudar a quebrar um sistema, ou me ensinar a quebrá-lo? .....	14
Como posso obter a senha da conta de outra pessoa? .....	14
Como posso invadir/ler/monitorar o e-mail de outra pessoa? .....	15
Como eu posso roubar privilégios de op de canal no IRC? .....	15
Eu fui invadido. Você me ajudará a evitar novos ataques? .....	15
Estou tendo problemas com meu software Windows. Você vai me ajudar? .....	15
Onde posso encontrar alguns hackers de verdade para conversar? .....	15
Você pode me recomendar livros úteis sobre assuntos relacionados com hacking? .....	15
Eu preciso ser bom em matemática para se tornar um hacker? .....	15
Que linguagem devo aprender primeiro? .....	16
Que tipo de hardware eu preciso? .....	16
Eu quero contribuir. Você pode me ajudar a escolher um problema para trabalhar? .....	16
Preciso odiar e esmurrar a Microsoft? .....	16
Mas o software de código aberto não deixará os programadores incapazes de ganhar a vida? ...	17
Onde posso pegar um Unix livre? .....	17

## Por que este documento?

Como editor do [Jargon File](#) e autor de outros documentos populares de natureza similar, eu frequentemente recebo pedidos de entusiastas novatos da rede perguntando (insistentemente) "como eu posso aprender a ser um hacker consciente?". De volta a 1996 eu percebi que não parecia haver nenhum outro FAQ ou documento web que focasse essa pergunta vital, então eu comecei a escrever este. Muitos hackers agora consideram-o definitivo, e eu suponho que isso significa que ele seja. Espere, eu não desejo ser a autoridade exclusiva sobre este assunto; se você não gostar do que leu aqui, escreva o seu.

Se você está lendo uma cópia deste documento offline, a versão atual encontra-se em <http://catb.org/~esr/faqs/hacker-howto.html>.

Nota: há uma lista de Perguntas Feitas com Frequência no fim deste documento. Por favor as leia (duas vezes) antes de me enviar mensagens com quaisquer perguntas sobre este documento.

Numerosas traduções deste documento estão disponíveis: [Árabe](#), [Búlgaro](#), [Catalão](#), [Chinês \(Simplificado\)](#), [Dinamarquês](#), [Holandês](#), [Estoniano](#), [Persa](#), [Finlandês](#), [Alemão](#), [Hebraico](#), [Grego](#), [Italiano](#), [Japonês](#), [Norueguês](#), [Polonês](#), [Português \(Brasileiro\)](#), [Romeno](#), [Russo](#), [Espanhol](#), [Turco](#) e [Sueco](#). Note que, já que este documento muda ocasionalmente, elas podem estar desatualizadas em graus variados.

O diagrama de cinco-pontos-em-nove-quadrados que decora este documento é chamado de glider. Ele é um padrão simples com algumas propriedades surpreendentes em simulação matemática chamada [Vida](#) que tem fascinado hackers por muitos anos. Eu penso que ele dá uma boa representação visual do que os hackers se parecem — abstrato, primeiramente um pouco misterioso ao olhar, mas uma porta de entrada para todo um mundo com uma lógica intrincada sobre si mesmo. Leia mais sobre o emblema glider [aqui](#).



## O que é um Hacker?

O Jargon File contém um monte de definições para o termo 'hacker', a maioria tem a ver com intimidade técnica e o deleite de resolver problemas e superar limites. Se você quer saber como se tornar um hacker, então, apenas dois destes tem realmente relevância.

Existe uma comunidade, uma cultura compartilhada, de programadores experts e magos de redes que traçam sua história desde décadas atrás nos primeiros mini-computadores de tempo compartilhado e nos mais remotos experimentos da ARPAnet. Os membros dessa comunidade originaram o termo 'hacker'. Hackers construíram a Internet. hackers criaram o sistema operacional Unix da forma como ele é hoje. Hackers rodam a Usenet. Hackers fizeram a World Wide Web funcionar. Se você é parte dessa cultura, se você contribuiu para isso e outras pessoas nisto o chamam de hacker, então você é um hacker.

A forma de pensar hacker não se restringe a essa cultura de hacker de software. Existem pessoas que aplicam a atitude hacker a outras coisas, como eletrônica ou música — atualmente, você pode encontrá-los nos níveis mais elevados de qualquer ciência ou arte. Hackers de software reconhecem esses espíritos semelhantes em outros lugares e podem chamá-los de 'hackers' também — e alguns afirmam que a natureza hacker é realmente independente do meio particular no qual o hacker trabalha. Mas no restante deste documento você irá se focar nas habilidades e atitudes de hackers de software, nas tradições e na cultura compartilhada que originou o termo 'hacker'.

Há um outro grupo de pessoas que ruidosamente se auto-entitulam hackers, mas não são. Essas são pessoas (em grande parte adolescentes do sexo masculino) que possuem uma atração por entrar em

computadores a força e por burlar o sistema telefônico. Verdadeiros hackers chamam essas pessoas de 'crackers' e não têm nada a ver com eles. Hackers verdadeiros em grande parte pensam que crackers são preguiçosos, irresponsáveis e não tão brilhantes, e sustentam que ser capaz de quebrar segurança não faz de você um hacker mais do que ser capaz de fazer ligação direta em um carro faz de você um engenheiro automotivo. Infelizmente, muitos jornalistas e escritores têm usado enganosamente a palavra 'hacker' para descrever crackers; isso irrita profundamente os verdadeiros hackers.

A diferença básica é esta: hackers constroem coisas, crackers as quebram.

Se você quer ser um hacker, continue lendo. Se você quer se tornar um cracker, vá ler o newsgroup [alt.2600](#) e prepare-se para pegar de cinco a dez na condicional antes de achar que você não é tão esperto quanto você pensa que é. E isso é tudo que eu vou dizer sobre crackers.



## A atitude Hacker

Hackers resolvem problemas e constroem coisas, e eles acreditam na liberdade e na ajuda mútua voluntária. Para ser aceito como um hacker, você precisa se comportar, bem como ter esse tipo de atitude, por conta própria. E para se comportar dessa forma e possuir a atitude, você deve realmente acreditar na atitude.

Mas se você pensa em cultivar atitudes hacker apenas como uma forma de ganhar aceitação na cultura, você irá perder o ponto. Tornar-se o tipo de pessoa que acredita nessas coisas é importante para você — para ajudá-lo a aprender e manter-se motivado. Como em todas as artes criativas, a maneira mais efetiva de se tornar um mestre e imitar o jeito de pensar dos mestres — não apenas intelectualmente mas emocionalmente também.

Ou, como o poema Zen moderno diz:

Para seguir o caminho:  
olhe para o mestre,  
siga o mestre,  
ande com o mestre,  
olhe através do mestre,  
torne-se o mestre.

Então, se você quer ser um hacker, repita as seguintes coisas até que você acredite nelas:

### **O mundo está cheio de problemas fascinantes esperando para serem resolvidos.**

Ser um hacker é muito divertido, mas é o tipo de divertimento que exige muito esforço. O esforço exige motivação. Atletas de sucesso obtêm sua motivação de um tipo de deleite físico ao fazer seus corpos atuarem, ao compelir a si próprios para superar seus limites físicos. Similarmente, para ser um hacker você tem que ter uma atração natural por resolver problemas, afiando as suas habilidades e exercitando a sua inteligência.

Se você não é o tipo de pessoa que sente isso de maneira natural, então você terá que se tornar uma para fazer isso como um hacker. De outra forma você irá achar que a sua energia para hackear está sendo minada por distrações como sexo, dinheiro e aprovação social.

(Você também terá que desenvolver um tipo de fé na sua própria capacidade de aprendizagem — uma crença em que, mesmo que você não saiba tudo que é necessário para resolver um problema, se você dispuser apenas de uma peça para isso e aprender a partir dela, você irá aprender o suficiente para resolver a próxima peça — e assim por diante, até que você termine.)

## **Nenhum problema jamais deve ter que ser resolvido duas vezes.**

Cérebros criativos são um recurso valioso e limitado. Eles não devem ser desperdiçados re-inventando a roda enquanto existem tantos novos problemas fascinantes esperando por aí.

Para se comportar como um hacker, você tem que acreditar que o tempo de pensamento de outros hackers é precioso — tanto que é quase um dever moral para você compartilhar informação, resolver problemas e dar as soluções de graça apenas para que outros hackers possam resolver novos problemas ao invés de terem que se dedicar perpetuamente a resolver os velhos.

Note, entretanto, que "Nenhum problema jamais deve ter que ser resolvido duas vezes." não implica que você tem que considerar todas as soluções existentes sagradas, ou que há apenas uma solução correta para um dado problema. Frequentemente nós aprendemos muito sobre o problema que nós não conhecemos antes estudando a primeira abordagem em uma solução. É legal, e frequentemente necessário, decidir o que nós podemos fazer melhor. O que não é legal são artificios técnicos, legais, ou barreiras institucionais (como código fonte fechado) para prevenir que uma boa solução seja re-utilizada e forçar pessoas a re-inventar rodas.

(Você não tem que acreditar que você é obrigado a dar toda a sua produção criativa de graça, embora os hackers que o fazem sejam os mais respeitados por outros hackers. É consistente com os valores hacker vender o suficiente para manter sua comida, aluguel e computadores. É ótimo usar suas habilidades de hackear para ajudar sua família ou mesmo para ficar rico, desde que você não se esqueça da lealdade a sua arte e do seus colegas hackers enquanto estiver fazendo isso.)

## **Tédio e trabalho repetitivo são ruins.**

Hackers (e pessoas criativas em geral) nunca deveriam ficar entediadas ou ter que se mourejar com trabalho repetitivo, porque quando isso acontece significa que eles não estão fazendo o que apenas eles podem fazer — resolver novos problemas. Esse desperdício machuca todo mundo. Portanto, tédio e trabalho repetitivo não são apenas desagradáveis, mas maléficos.

Para se comportar como um hacker, você tem que acreditar nisso o suficiente para querer automatizar as partes chatas tanto quanto for possível, não apenas para você mas para todo mundo (especialmente outros hackers).

(Não há nenhuma exceção aparente para isso. Os hackers irão vez por outra fazer coisas que parecem ser repetitivas ou tediosas para um observador como um exercício de limpeza mental, ou como meio para adquirir uma habilidade ou ter algum tipo de experiência que não seria possível de outra maneira. Mas isso se dá por escolha — ninguém capaz de pensar deveria jamais ser forçado a uma situação que o deixa entediado.)

## **A liberdade é boa.**

Hackers são naturalmente anti-autoritários. Qualquer um que possa lhe dar ordens pode impedi-lo de resolver qualquer problema pelo qual você esteja fascinado — e, dada a maneira com a qual mentes autoritárias trabalham, irão geralmente encontrar alguma razão terrivelmente estúpida para isso. Então a atitude autoritária deve ser combatida toda vez que você encontrá-la, já que ela sufoca você e outros hackers.

(Isso não é o mesmo que lutar contra toda autoridade. Crianças precisam ser orientadas e criminosos detidos. Um hacker pode concordar em aceitar alguns tipos de autoridade de forma a conseguir algo que ele queira mais que o tempo que ele gasta seguindo ordens. Mas isto é uma limitada, barganha consciente; o tipo de sujeição pessoal que os autoritários querem não está em oferta.)

Pessoas autoritárias prosperam na censura e no segredo. E desconfiam de cooperação voluntária e compartilhamento de informação - só gostam de "cooperação" que eles controlam. Portanto, para se comportar como um hacker, você tem que desenvolver uma hostilidade instintiva à censura, ao

segredo, e ao uso da força ou mentira para compelir adultos responsáveis. E você tem que estar disposto a agir com base nessa crença.

### **Atitude não substitui competência.**

Para ser um hacker, você tem que desenvolver algumas dessas atitudes. Mas exibir acanhadamente apenas uma atitude não fará de você um hacker, não mais do que ela o tornará um atleta campeão ou uma estrela do rock. Tornar-se um hacker irá exigir inteligência, prática, dedicação e trabalho duro.

Portanto, você tem que aprender a desconfiar de atitude e respeitar competências de todo tipo. Hackers não deixam posers gastarem seu tempo, mas eles admiram competência - especialmente competência em hackear, mas competência em qualquer coisa é valorizada. A competência em habilidades que poucos conseguem dominar é especialmente boa, e competência em habilidades que envolvem acuidade mental, perícia e concentração é a melhor.

Se você reverenciar a competência, você gostará de desenvolvê-la em si mesmo - o trabalho duro e a dedicação se tornarão uma espécie de jogo intenso, ao invés de labuta. Essa atitude é vital para se tornar um hacker.



## **Habilidades Básicas de Hackear**

O atitude do hacker é vital, mas habilidades são ainda mais vitais. Atitude não substitui competência, e há uma certa coleção de habilidades que você deve ter antes que qualquer hacker sonhe em chamá-lo assim.

Essa coleção muda lentamente com o tempo a medida que a tecnologia cria novas habilidades e torna outras obsoletas. Por exemplo, ela costumava incluir linguagem de programação de maquina, e não abrangia HTML até recentemente. Mas agora ela muito claramente inclui o seguinte:

### **Aprenda como programar.**

Esta, naturalmente, é a habilidade fundamental do hacker. Se você não conhecer nenhuma linguagem de computador, eu recomendo começar com Python. Ela é projetada de forma limpa, bem documentada e relativamente gentil para com iniciantes. Apesar de ser uma boa primeira linguagem, ela não é apenas um brinquedo, é muito poderosa e flexível e bem adequada para grandes projetos. Eu escrevi uma [avaliação mais detalhada do Python](#). Bons [tutoriais](#) estão disponíveis no [web site do Python](#).

Eu costumava recomendar Java como uma boa linguagem para aprender cedo, mas [essa crítica](#) mudou minha opinião (procure por "As Armadilhas do Java como primeira linguagem de programação" dentro dela). Um hacker não pode, como eles duramente colocam "abordar a resolução de problemas como um encanador em uma loja de ferramentas", você tem que saber o que os componentes realmente fazem. Agora eu acho que é provavelmente melhor aprender C e Lisp primeiro, depois Java.

Se você se envolver com programação séria, você vai ter que aprender C, a linguagem fundamental do Unix. C++ é muito estreitamente relacionada com a C, se você conhece uma, aprender a outra não vai ser difícil. No entanto, não é uma boa linguagem para tentar aprender como a primeira. E, realmente, quanto mais você puder evitar programação em C, mais produtivo você será.

C é muito eficiente, e faz uso parco dos recursos da sua máquina. Infelizmente, C obtém essa eficiência requerendo de você a realização de muito gerenciamento de baixo nível de recursos

(como memória) manualmente. Todo este código de baixo nível é complexo e propenso a erros, e irá absorver enormes quantidades do seu tempo na depuração. Com as máquinas de hoje tão poderosas como estão, isso geralmente é uma troca ruim - é mais inteligente usar uma linguagem que usa o tempo da máquina de forma menos eficiente, mas seu tempo de forma muito mais eficiente. Sendo assim, Python.

Outras linguagens de particular importância para os hackers incluem [Perl](#) e [LISP](#). Aprender Perl vale a pena, por razões práticas, é muito utilizado para páginas web e administração de sistemas, de modo que mesmo se você nunca escreva Perl você deve aprender a lê-lo. Muitas pessoas usam Perl da maneira que eu sugiro que você deve usar Python, a fim de evitar programação C em trabalhos que não exigem a eficiência de máquina do C. Você precisa ser capaz de compreender o seu código.

Vale a pena aprender LISP por um motivo diferente - a experiência de profunda iluminação que você terá quando finalmente consegui-lo. Essa experiência vai fazer você um programador melhor pelo resto de seus dias, mesmo se você nunca realmente utilizar muito LISP em si. (Você pode obter alguma experiência inicial com LISP facilmente escrevendo e modificando os modos de edição para o editor de texto Emacs, ou plugins Script-Fu para o GIMP).

É melhor, na verdade, aprender todos os cinco de Python, C/C++, Java, Perl e LISP. Além de serem as linguagens mais importantes para hackear, elas representam abordagens muito diferentes de programação, e cada uma vai educá-lo de maneiras importantes.

Mas fique ciente que você não vai alcançar o nível de habilidade de um hacker ou mesmo um programador simplesmente pela acumulação de linguagens - você precisa aprender a pensar sobre problemas de programação de um modo geral, independente de qualquer linguagem. Para ser um hacker de verdade, você precisa chegar ao ponto onde você pode aprender uma nova linguagem em dias, relacionando o que está no manual com o que você já sabe. Isto significa que você deve aprender várias línguas muito diferentes.

Eu não posso dar instruções completas sobre como aprender a programar aqui - é uma habilidade complexa. Mas eu posso lhe dizer que livros e cursos não irão fazê-lo - muitos, talvez a maioria dos melhores hackers são auto-didatas. Você pode aprender recursos de linguagem - pedaços de conhecimento - a partir de livros, mas a mentalidade que faz desse conhecimento habilidade viva pode ser aprendido apenas pela prática e treinamento. O que vai fazer isso é (a) ler código e (b) escrever código.

Peter Norvig, que é um dos melhores hackers do Google e co-autor do livro mais amplamente usado em I.A., escreveu um excelente ensaio chamado [Ensine Você a Programar em Dez Anos](#). Sua "receita para o sucesso com a programação" merece atenção especial.

Aprender a programar é como aprender a escrever bem linguagem natural. A melhor maneira de fazer isso é ler algumas coisas escritas por mestres da forma, escrever algumas coisas você mesmo, ler muito mais, escrever um pouco mais, ler mais um monte, escrever mais algumas ... e repetir até que sua escrita comece a desenvolver o tipo de força e economia que você vê em suas referências.

Achar bom código para ler costumava ser difícil, porque havia poucos programas grandes disponíveis na fonte para que hackers novatos pudessem ler e mexer neles. Essa situação mudou dramaticamente; software open-source, ferramentas de programação e sistemas operacionais (todos feitos por hackers) estão amplamente disponíveis agora. O que me leva justamente ao nosso próximo tópico ...

## **Pegue um dos Unixes de código aberto e aprenda a usar e executá-lo.**

Eu vou assumir que você tem um computador pessoal ou tem acesso a um. (Tire um tempo para apreciar o quanto isso significa. A cultura hacker surgiu originalmente no passado quando os computadores eram tão caros que os indivíduos não podiam tê-los.) O passo mais importante que um novato deve dar para adquirir habilidades de hacker é pegar uma cópia do Linux ou um dos

BSD Unices ou OpenSolaris, o instalar em uma máquina pessoal, e executá-lo.

Sim, há outros sistemas operacionais no mundo além do Unix. Mas eles são distribuídos em forma binária - você não pode ler o código, e você não pode modificá-lo. Tentar aprender a "hackear" em uma máquina com o Microsoft Windows ou qualquer outro sistema de código fonte fechado é como tentar aprender a dançar com o corpo engessado.

No Mac OS X é possível, mas apenas uma parte do sistema é de código aberto - é provável que você bater em um monte de paredes, e você tem que ter cuidado para não desenvolver o mau hábito de depender de código proprietário da Apple. Se você se concentrar no que há sob o capô do Unix você pode aprender algumas coisas úteis.

Unix é o sistema operacional da Internet. Embora você possa aprender a usar a Internet sem conhecer Unix, você não pode ser um hacker de Internet sem entender Unix. Por esta razão, a cultura hacker, atualmente, é fortemente centralizada no Unix. (Isto não foi sempre assim, e alguns hackers os velhos tempos ainda não estão felizes com isso, mas a simbiose entre o Unix e a Internet se tornou tão forte que até mesmo os músculos da Microsoft não parecem ser capazes de avariá-la seriamente.)

Então, pegue um Unix - eu pessoalmente gosto do Linux, mas existem outras maneiras (e sim, você pode rodar Linux e Windows na mesma máquina). Aprenda-o. Execute-o. Mexa com ele. Converse com a Internet com ele. Leia o código. Modifique o código. Você terá ferramentas melhores de programação (incluindo C, LISP, Python e Perl) do que qualquer sistema operacional da Microsoft pode sonhar de oferecer, você vai se divertir, e você irá absorver mais conhecimento do que você acredita estar aprendendo até você olha para trás como um mestre hacker.

Para mais sobre aprendizado Unix, veja [O Loginataka](#). Você também deverá querer dar uma olhada no [A Arte da Programação Unix](#).

Para colocar suas mãos em um Linux, veja o site [Linux Online!](#); você pode baixar de lá ou (ideia melhor) encontrar um grupo de usuários Linux local para ajudá-lo com a instalação.

Durante os primeiros dez anos de vida deste HOWTO, eu relatei que do ponto de vista de um novo usuário, todas as distribuições Linux são quase equivalentes. Mas, em 2006-2007, uma melhor escolha real surgiu: o [Ubuntu](#). Enquanto outras distribuições têm suas próprias áreas de força, o Ubuntu é de longe o mais acessível para novatos.

Você pode encontrar ajuda e recursos do Unix BSD em [www.bsd.org](http://www.bsd.org).

Uma boa maneira de mergulhar os dedos na água é bootar o que os fãs do Linux chamam de [live CD](#), uma distribuição que roda inteiramente do CD sem ter que alterar o seu disco rígido. Ela será lenta, porque os CDs são lentos, mas é uma maneira de dar uma olhada nas possibilidades sem ter que fazer algo drástico.

Eu escrevi uma cartilha sobre os [princípios básicos do Unix e da Internet](#).

Eu costumava desaconselhar a instalação de Linux ou BSD como um projeto solo se você fosse um novato. Atualmente, os instaladores se tornaram bons o suficiente para que fazê-lo inteiramente por conta própria seja possível, mesmo para um novato. No entanto, eu ainda recomendo fazer contato com o seu grupo de usuários do Linux local e pedir ajuda. Isso não irá machucar, e pode facilitar o processo.

## **Aprenda a usar a World Wide Web e escrever HTML.**

A maioria das coisas que a cultura hacker construiu faz o seu trabalho fora das vistas, ajudando no funcionamento de fábricas, escritórios e universidades sem nenhum impacto óbvio na vida dos não-hackers. A Web é a única grande exceção, o enorme e brilhante brinquedo dos hackers que até mesmo políticos admitem ter mudado o mundo. Por esta única razão (e um monte de outras tão boas quanto) você precisa aprender como trabalhar a Web.

Isto não significa apenas aprender como pilotar um browser (qualquer um pode fazer isso), mas aprender como escrever HTML, a linguagem de marcação da Web. Se você não sabe programar, escrever HTML lhe ensinará alguns hábitos mentais que o ajudarão a aprender. Então faça uma home page. Tente avançar para o XHTML, que é uma linguagem mais limpa que o HTML clássico. (Existem bons tutoriais para iniciantes na Web; [aqui está um](#).)

Mas apenas ter uma home page não está nem perto de ser bom suficiente para fazer de você um hacker. A Web está repleta de home pages. A maioria delas é inútil, porcaria sem conteúdo algum - porcaria muito bem apresentada, note bem, mas porcaria mesmo assim (para saber mais sobre isso veja o [The HTML Hell Page](#)).

Para valer a pena, sua página deve ter conteúdo - deve ser interessante e/ou útil para outros hackers. E isso nos leva ao próximo assunto...

## **Se você não tem Inglês funcional, aprenda.**

Como Americano e falante nativo de Inglês, eu fiquei relutante em sugerir isso, sob pena de ser considerado como uma espécie de imperialismo cultural. Mas vários falantes nativos de outras línguas, pediram-me para salientar que o Inglês é a língua de trabalho da cultura hacker e da Internet, e que você precisará conhecê-lo para atuar na comunidade hacker.

De volta a 1991 eu aprendi que muitos hackers que têm o Inglês como segunda língua o usam em discussões técnicas, mesmo quando eles compartilham uma língua nativa, isso foi relatado para mim na época em que o Inglês tinha um vocabulário técnico mais rico do que qualquer outra língua sendo, portanto, simplesmente uma ferramenta melhor para o trabalho. Por razões semelhantes, traduções de livros técnicos escritos em Inglês são muitas vezes insatisfatórias (quando elas são feitas, em resumo).

Linus Torvalds, um finlandês, comenta seu código em Inglês (aparentemente nunca ocorreu a ele fazer o contrário). Sua fluência em Inglês tem sido um fator importante em sua capacidade de recrutar uma comunidade mundial de desenvolvedores para o Linux. É um forte exemplo a seguir.

Ser um falante nativo de Inglês não garante que você tenha conhecimentos linguísticos suficientes para atuar como um hacker. Se a sua escrita é semi-analfabeta, com erros gramaticais, e cheia de erros ortográficos, muitos hackers (inclusive eu) tenderão a ignorá-lo. Escrita malfeita não invariavelmente significa pensar desleixado, nós geralmente achamos que tal correlação seja forte - e nós não temos uso para os pensadores desleixados. Se você ainda não pode escrever com competência, aprenda.



## **Status na Cultura Hacker**

Como a maioria das culturas sem economia monetária, a dos hackers se baseia em reputação. Você está tentando resolver problemas interessantes, mas quão interessantes eles são, e o quanto suas soluções são realmente boas, é algo que somente seus iguais ou superiores tecnicamente são normalmente capazes de julgar.

Assim, quando você joga o jogo hacker, você aprende a marcar pontos principalmente pelo que outros hackers pensam da sua habilidade (por isso você não é realmente um hacker até que outros hackers lhe chamem assim). Esse fato é obscurecido pela imagem de hacking como trabalho solitário; também por um tabu cultural hacker (deteriorando gradualmente desde o final dos anos 1990, mas ainda potente) contra admitir que o ego ou a aprovação externa estão envolvidas na motivação de todos.

Especificamente, a cultura hacker é o que os antropólogos chamam de cultura de doação. Você ganha status e reputação não por dominar outras pessoas, nem por ser bonito, nem por ter coisas que outras pessoas querem, mas sim por doar coisas. Especificamente, por doar seu tempo, sua criatividade, e os resultados de sua habilidade.

Existem basicamente cinco tipos de coisas que você pode fazer para ser respeitado por hackers:

## **Escreva software de código aberto**

O primeiro (o mais central e mais tradicional) é escrever programas que outros hackers achem divertidos ou úteis, e doar o código-fonte para toda a cultura hacker usar.

(Nós costumávamos chamar isto de "software livre", mas isso confundia muitas pessoas que não tinham certeza do que exatamente "livre" supostamente significa. Muitos de nós preferem o termo software "de [código aberto](#)").

Os semideuses mais reverenciados da cultura hacker são pessoas que escreveram programas grandes e capazes, que correspondem a uma necessidade generalizada, e os doaram, de modo que agora todo mundo os usa.

Mas há um pouco de bons pontos históricos aqui. Embora os hackers sempre tenham olhado para os desenvolvedores de código aberto entre eles como núcleo mais duro da nossa comunidade, antes de meados dos anos 1990 a maioria dos hackers, na maior parte do tempo, trabalhava em código fechado. Isto ainda era verdade quando eu escrevi a primeira versão deste HOWTO em 1996, passando pelo auge do software de código aberto depois de 1997, para mudar as coisas. Hoje, "a comunidade hacker" e "desenvolvedores de código aberto" são duas descrições para o que é essencialmente a mesma cultura e população - mas vale lembrar que isso não foi sempre assim.

## **Ajude a testar e depurar software de código aberto**

Eles também servem quem apoia e depura software de código aberto. Neste mundo imperfeito, nós inevitavelmente passamos a maior parte do nosso tempo de desenvolvimento de software na fase de depuração. É por isso que qualquer autor de código aberto que pense lhe dirá que bons beta-testers (que saibam descrever sintomas claramente, localizar bem problemas, que possam tolerar bugs em um lançamento apressado, e que estejam dispostos a aplicar algumas simples rotinas de diagnóstico) tem seu peso avaliado em ouro. Até mesmo um desses pode fazer a diferença entre uma fase de depuração que seja demorada, um cansativo pesadelo e uma que é apenas um aborrecimento saudável.

Se você é um novato, tente achar um programa em desenvolvimento no qual você esteja interessado e seja um bom beta-tester. Há uma progressão natural de ajudar a testar programas para ajudar a depurar e depois ajudar a modificá-los. Você aprenderá muito assim, e criará um bom karma com pessoas que irão ajudá-lo mais tarde.

## **Publique informações úteis.**

Outra coisa boa é colecionar e filtrar informações úteis e interessantes em páginas ou documentos como listas de Perguntas Realizadas com Frequência (FAQ), e torná-las disponíveis para todos.

Mantenedores dos maiores FAQs técnicos recebem praticamente o mesmo respeito que os autores de código aberto.

## **Ajude a manter a infra-estrutura funcionando**

A cultura hacker (e o desenvolvimento da engenharia da Internet, para o que importa) é executada por voluntários. Existe uma grande quantidade de trabalho necessário, mas sem glamour que precisa

ser feito para mantê-la viva - administrar listas de email, moderar grupos de discussão, manter grandes sites que armazenam software, desenvolver RFCs e outros padrões técnicos.

Pessoas que fazem esse tipo de coisa conseguem muito respeito, porque todo mundo sabe que esses serviços são grandes sugadores de tempo e não são tão divertidos como brincar com código. Fazê-los mostra dedicação.

## Sirva-se à cultura hacker

Finalmente, você pode servir e propagar a cultura por conta própria (ao, por exemplo, escrever um apurado manual sobre como se tornar um hacker :-)). Isso não é algo que você será incumbido de fazer até que você esteja por aí há algum tempo e torne-se conhecido por uma das primeiras quatro coisas.

A cultura hacker não têm líderes, exatamente, mas tem seus heróis culturais, chefes tribais e historiadores e porta-vozes. Quando você estiver tempo suficiente nas trincheiras, você pode se transformar em um desses. Cuidado: hackers desconfiam de egos espalhafatosos em seus chefes tribais, então ser reconhecido por este tipo de fama é perigoso. Ao invés de lutar por ele, você tem meio que se posicionar, assim isso cai no seu colo, e então ser modesto e cortês sobre seu status.



## A Conexão Hacker/Nerd

Contrariamente ao mito popular, você não tem que ser um nerd para ser um hacker. Isso ajuda, realmente, e muitos hackers são de fato nerds. Ser algo como um proscrito social o ajuda a se manter concentrado nas coisas realmente importantes, como pensar e hackear.

Por esta razão, muitos hackers adotaram o rótulo de "geek" como um símbolo de orgulho - é um modo de declarar sua independência de expectativas sociais normais (assim como um gosto por outras coisas como ficção científica e jogos de estratégia que muitas vezes vêm junto com ser um hacker). O termo "nerd" costumava ser usado dessa forma no passado na década de 1990, na época "nerd" era ligeira pejorativa e 'geek' mais descolado, algum tempo depois de 2000 eles trocaram de lugares, pelo menos na cultura popular dos E.U., e agora há até mesmo uma cultura significativa de orgulho geek entre as pessoas que não são técnicas.

Se você consegue se organizar para concentrar-se o suficiente em hackear para ser bom nisso e ainda ter uma vida, tudo bem. Isto é muito mais fácil hoje do que era quando eu era um novato na década de 1970, a cultura popular é muito mais amigável para com techno-nerds agora. Há até mesmo um número crescente de pessoas que percebem que hackers são frequentemente materiais de alta qualidade para cônjuges e amantes.

Se hackear o atrai porque você não tem uma vida, tudo bem também - pelo menos você não terá problemas para se concentrar. Talvez você consiga uma vida mais tarde.



## Pontos por Estilo

Mais uma vez, para ser um hacker, você tem que entrar na mentalidade hacker. Há algumas coisas que você pode fazer quando você não estiver em um computador que parecem ajudar. Elas não são substitutas para o hacking (nada é), mas muitos hackers as fazem, e sentem que se conectam de alguma maneira básica com a essência do hacking.

- Aprenda a escrever bem em seu idioma nativo. Embora seja um estereótipo comum que os programadores não conseguem escrever, um número surpreendente de hackers (incluindo todos os mais talentosos que eu conheço) são escritores muito capazes.
- Leia ficção científica. Vá para convenções de ficção científica (uma boa maneira de encontrar hackers e proto-hackers).
- Treine uma forma de artes marciais. O tipo de disciplina mental necessária para artes marciais parece ser semelhante em aspectos importantes com a que os hackers usam. As formas mais populares entre os hackers são definitivamente as artes asiáticas de mãos vazias, como Tae Kwon Do, várias formas de Karate, Kung Fu, Aikido, ou Ju Jitsu. Esgrima ocidental e artes espada asiáticas também têm seguidores visíveis. Em locais onde é legal, o tiro de pistola vem crescendo em popularidade desde a década de 1990. As artes marciais mais hackers são aquelas que enfatizam a disciplina mental, consciência relaxada e controle, em vez de força bruta, atletismo, ou a resistência física.
- Estude uma disciplina de meditação efetiva. O eterno favorito entre os hackers é Zen (e o importante, é possível se beneficiar do Zen sem adquirir uma religião ou descartar uma que você já possui). Outros estilos podem funcionar bem, mas tome cuidado para escolher um que exija que você acredite em coisas malucas.
- Desenvolva um ouvido analítico para música. Aprenda a apreciar tipos peculiares de música. Aprenda a tocar bem algum instrumento musical, ou como cantar.
- Desenvolva sua apreciação por trocadilhos e jogos de palavras.

Quanto mais dessas coisas você já fizer, mais provável que você seja um material hacker natural. Porque essas coisas em particular não estão completamente claro, mas elas são ligadas com uma mistura habilidades dos lados direito e esquerdo do cérebro que parecem ser importantes, os hackers precisam ser capazes tanto de raciocinar logicamente quanto de caminhar para fora da lógica aparente de um problema em dados momentos.

Trabalhar tão intensamente quanto você joga e jogar tão intensamente quanto você trabalha. Para os hackers de verdade, todas as fronteiras entre "jogo", "trabalho", "ciência" e "arte" tendem a desaparecer ou a fundirem-se em uma jog de alto nível criativo. Além disso, não se contentar com uma pequena gama de habilidades. Embora a maioria dos hackers se auto-descrevam como programadores, eles são muito susceptíveis a serem mais que competentes em várias habilidades relacionadas - administração de sistemas, web design, e solução de problemas de hardware de PC são mais comuns. Um hacker que é um administrador de sistema, por outro lado, é provável que seja bastante hábil no script de programação e web design. Hackers não fazem as coisas pela metade, se investem em uma habilidade dentre outras, eles tendem a ficar muito bons nela.

Finalmente, algumas coisas para não fazer.

- Não use uma identificação de usuário ou pseudônimo idiota e enorme.
- Não entre em discussões sem sentido na Usenet (ou em qualquer outro lugar).
- Não se defina como um "cyberpunk", e não perca seu tempo com ninguém que o faça.
- Não publicar ou enviar e-mail cuja escrita esteja cheia de erros de ortografia e gramática ruim.

A única reputação que você conseguirá fazendo alguma dessas coisas será a de bobo. Hackers tem memória boa - pode levar anos para se deixar suas tolices para trás o suficiente para ser aceito.

O problema com os nomes de tela ou apelidos merece alguma amplificação. Esconder sua identidade por trás de um apelido é um comportamento juvenil e idiota característico de crackers, warez d00dz e outras formas de vida mais baixas. Hackers não fazem isso, eles estão orgulhosos do que fazem e querem associados com seus nomes reais. Então se você tem um apelido, abandone-o. Na cultura hacker, ele só irá marcá-lo como um perdedor.



## Outros recursos

Paul Graham escreveu um ensaio chamado [Great Hackers](#), e outro na [graduação](#), no qual ele fala com muita sabedoria.

Existe um documento chamado [Como ser um programador](#) que é um excelente complemento deste. Ele tem conselhos valiosos não apenas sobre codificação e conjuntos de habilidades, mas sobre a forma como atuar em uma equipe de programação.

Eu também escrevi [A Brief History Of Hackerdom](#).

Eu escrevi um documento, [A Catedral e o Bazar](#), que explica muito sobre como as culturas do Linux e do código-fonte aberto funcionam. Eu abordei este tema ainda mais diretamente na sua continuação [Homesteading Noosfera](#).

Rick Moen, escreveu um excelente documento sobre [como montar um grupo de usuários Linux](#).

Rick Moen e eu tenho colaboramos em outro documento sobre [Como Fazer Perguntas Inteligentes](#). Isso irá ajudá-lo a procurar ajuda de uma forma que seja mais provável que você realmente consiga.

Se você precisa de instruções sobre os princípios de como computadores pessoais, Unix, e os trabalhos de Internet funcionam, veja [The Unix and Internet Fundamentals HOWTO](#).

Quando você disponibilizar software ou escrever correções para software, tente seguir as normas descritas no [Software Release Practice HOWTO](#).

Se você gostou do poema Zen, você também poderá gostar do [Rootless Root: The Unix Koans of Master Foo](#).



## Perguntas Feitas com Frequência

### Como posso saber se eu já sou um hacker?

Faça a si mesmo as três seguintes perguntas:

- Você fala código fluentemente?
- Você se identifica com os objetivos e valores da comunidade hacker?
- Algum membro bem estabelecido da comunidade hacker já o chamou de hacker?

Se você puder responder sim a *\*todas as três\** dessas perguntas, você já é um hacker. Duas apenas não são suficientes.

O primeiro teste é uma questão de competências. Você provavelmente passará se você tiver o mínimo de habilidades técnicas descritas anteriormente neste documento. Você irá passar direto através dela se você tiver uma quantidade considerável de código aceito por um projeto de desenvolvimento open-source.

O segundo teste é uma questão de atitude. Se os [cinco princípios da mentalidade hacker](#) parecem óbvios para você, mais como uma descrição da maneira que você já vive do que uma novela sobre qualquer coisa, você já está a meio caminho para passá-lo. Essa é a metade do aperfeiçoamento, o outra metade que resta é o grau com o qual você se identifica com os projetos de longo prazo da comunidade hacker.

Aqui está uma lista incompleta, mas indicativa de alguns desses projetos: Qual a importância para você de que o Linux melhore e se espalhe? Você está apaixonado por liberdade de software? É hostil aos monopólios? Você age com a crença de que os computadores podem ser instrumentos de capacitação que fazem do mundo um lugar mais rico e mais humano?

Mas uma nota de aviso está posta aqui. A comunidade hacker tem alguns específicos e principalmente defensivos, interesses políticos - dois deles são a defesa dos direitos da liberdade de expressão e a extirpação do poder que a "propriedade intelectual" tem de tornar ilegal o software de código aberto. Alguns desses projetos a longo prazo são organizações de liberdades civis como a Electronic Frontier Foundation, e tal atitude devidamente inclui o apoio deles. Mas, além disso, a maioria dos hackers vê tentativas de sistematizar a atitude hacker em um explícito programa político com suspeita, nós aprendemos, a duras penas, que essas tentativas são divisivas e distraentes. Se alguém tentar recrutá-lo para marchar sobre a sua capital em nome da atitude hacker, eles perderam o ponto. A resposta certa é, provavelmente, "Cale-se e mostre-lhes o código".

O terceiro teste tem um elemento complicado de recursividade sobre ele. Eu observei na seção chamada "[O que é um Hacker?](#)" que ser um hacker é em parte uma questão de pertencer a uma subcultura particular ou rede social com uma história comum, uma dentro e uma fora. No passado distante, os hackers eram um grupo muito menos coeso e auto-consciente do que são hoje. Mas a importância do aspecto de rede social tem aumentado ao longo dos últimos trinta anos a medida que a Internet tem criado conexões com o núcleo da subcultura hacker mais fáceis de desenvolver e manter. Um registro comportamental da mudança é este, neste século, nós temos a nossas próprias camisetas.

Sociólogos, que estudam as redes como as da cultura hacker sob a rubrica geral de "faculdades invisíveis", notaram que uma característica dessas redes é que elas possuem porteiros - membros centrais com a autoridade social para aprovar novos membros na rede. Porque a "faculdade invisível", que é a cultura hacker é flexível e informal, o papel do porteiro é informal também. Mas uma coisa que todos os hackers entendem profundamente é que nem todo hacker é um porteiro. Porteiros tem que ter um certo grau de antiguidade e de realização, antes de poderem ostentar o título. Quanto é difícil de quantificar, mas todo hacker o reconhece assim que o vê.

## **Você vai me ensinar a hackear?**

Desde a primeira publicação desta página, que eu tenho recebido vários pedidos por semana (frequentemente vários por dia) de pessoas para "ensinar-me tudo sobre hacking". Infelizmente, eu não tenho tempo nem energia para fazer isso, meus próprios projetos hackers, e o trabalho como um defensor do código aberto, ocupam 110% do meu tempo.

Mesmo se eu o fizesse, hacking é uma atitude e uma habilidade que você basicamente tem que ensinar a si mesmo. Você verá que, embora hackers de verdade queiram ajudá-lo, eles não irão respeitá-lo se você implorar para ser tratado na boca com tudo o que sabem.

Aprenda algumas coisas primeiro. Mostre que você está tentando, que você é capaz de aprender por conta própria. Então vá até os hackers que você conhece com perguntas específicas.

Se você escrever um e-mail para um hacker pedindo conselhos, aqui estão duas coisas a saber primeiro. Primeiro, nós achamos que as pessoas que são preguiçosas ou descuidadas com sua escrita são geralmente muito preguiçosas e negligentes em seus pensamentos para dar bons hackers - por isso tome cuidado para soletrar corretamente, e usar uma boa gramática e pontuação, caso contrário, você provavelmente vai ser ignorado. Em segundo lugar, não se atreva a pedir uma resposta para uma conta de ISP que é diferente da conta a partir da qual você está escrevendo. Nós achamos que pessoas que fazem isso geralmente são ladrões utilizando contas roubadas, e não temos interesse em premiar ou ajudar ladrões.

## **Como posso começar, então?**

A melhor maneira para você começar seria provavelmente ir a uma reunião de um LUG (grupo de usuários Linux). Você pode encontrar esses grupos na página [Informações Gerais sobre o Linux do LDP](#). Provavelmente há um perto de você, possivelmente associada a uma faculdade ou universidade. Membros do LUG provavelmente vão te dar um Linux, se você pedir, e certamente vão ajudá-lo a instalar um e começar.

## **Quando você tem que começar? É tarde demais para eu aprender?**

Qualquer idade na qual você estiver motivado para começar é uma boa idade. A maioria das pessoas parece se interessar entre as idades de 15 e 20 anos, mas eu sei de exceções em ambos os sentidos.

## **Quanto tempo leva para eu aprender a hackear?**

Isso depende de quão você é talentoso e quão duro você trabalha nisso. A maioria das pessoas que tentam podem adquirir um conjunto de habilidades respeitável em dezoito meses a dois anos, se elas se concentrarem. Não pense que termina ali, entretanto. No hacking (como em muitos outros campos) se leva cerca de dez anos para atingir a maestria. E se você é um hacker de verdade, você vai passar o resto de sua vida aprendendo e aperfeiçoando o seu ofício.

## **Visual Basic é uma boa linguagem para começar?**

Se você está fazendo esta pergunta, é quase certo que significa que você está pensando em tentar hackear no Microsoft Windows. Esta é uma má idéia por si só. Quando eu comparei tentar aprender a hackear no Windows com tentar aprender a dançar com o corpo engessado, eu não estava brincando. Não vá lá. É feio, e nunca deixa de ser feio.

Existe um problema específico com o Visual Basic. Principalmente porque ele não é portátil. Embora existam um protótipos de implementações em código aberto do Visual Basic, os padrões ECMA aplicáveis não cobrem mais do que um pequeno conjunto de suas interfaces de programação. No Windows a maioria das suas bibliotecas de apoio é de propriedade de um único fornecedor (Microsoft). Se você não for extremamente cauteloso com os recursos que você usar - mais cuidadoso do que qualquer novato é realmente capaz de ser - você vai acabar preso a apenas aquelas plataformas que a Microsoft escolhe apoiar. Se você está começando com Unix, linguagens muito melhores com melhores bibliotecas estão disponíveis. Python, por exemplo.

Também, como em outros Basics, o Visual Basic é uma linguagem mal concebida que irá lhe ensinar maus hábitos de programação. Não, não me peça para descrevê-los em detalhes, já que a explicação poderia encher um livro. Aprenda uma linguagem bem concebida em seu lugar.

Um desses maus hábitos é se tornar dependente de bibliotecas, widgets e ferramentas de desenvolvimento de um único fornecedor. Em geral, qualquer linguagem que não é totalmente compatível com Linux ou pelo menos um dos BSDs, e/ou pelo menos três fornecedores de sistemas operacionais, é pobre para aprender a hackear com ela.

## **Você poderia me ajudar a quebrar um sistema, ou me ensinar a quebrá-lo?**

Não. Qualquer pessoa que possa ainda perguntar tal coisa depois de ler este FAQ é estúpida demais para ser educável mesmo se eu tivesse tempo para tutoria. Quaisquer pedidos por e-mail que eu receba serão ignorados ou respondidos com rudeza extrema.

## **Como posso obter a senha da conta de outra pessoa?**

Isso é cracking. Vá embora, seu idiota.

## **Como posso invadir/ler/monitorar o e-mail de outra pessoa?**

Isso é cracking. Suma, seu imbecil.

## **Como eu posso roubar privilégios de op de canal no IRC?**

Isso é cracking. Vá embora, cretino.

## **Eu fui invadido. Você me ajudará a evitar novos ataques?**

Não. Toda vez que eu tenho recebido esta pergunta até agora, tem sido de algum pobre coitado rodando Microsoft Windows. Não é possível efetivamente proteger os sistemas Windows contra ataques de invasores. O código e a arquitetura simplesmente tem muitas falhas, o que torna a proteção do Windows parecida com tentar resgatar um barco com uma peneira. A única prevenção confiável começa com a mudança para o Linux ou outro sistema operacional que foi projetado para, pelo menos, ser capaz de oferecer segurança.

## **Estou tendo problemas com meu software Windows. Você vai me ajudar?**

Sim. Vá para um prompt do DOS e digite "format c:". Qualquer problema que você está enfrentando cessará em poucos minutos.

## **Onde posso encontrar alguns hackers de verdade para conversar?**

A melhor maneira é encontrar um grupo local de usuários de Unix ou Linux para você e ir às suas reuniões (você pode encontrar links para várias listas de grupos de usuários no site do [LDP](#) no [ibiblio](#)).

(Eu costumava dizer aqui que você não encontraria algum hacker de verdade no IRC, mas eu passei a entender que isso está mudando. Aparentemente algumas comunidades hacker de verdade, ligadas a coisas como o GIMP e Perl, têm canais de IRC agora.)

## **Você pode me recomendar livros úteis sobre assuntos relacionados com hacking?**

Eu mantenho uma [HOWTO de Lista de Leitura de Linux](#) que você pode achar útil. O [Loginataka](#) também pode ser interessante.

Para uma introdução ao Python, veja os [materiais introdutórios](#) no site do Python.

## **Eu preciso ser bom em matemática para se tornar um hacker?**

Não. Hacking usa muito pouco da matemática formal ou aritmética. Em particular, você geralmente não precisa de trigonometria, cálculo ou análise (há exceções a isso em um punhado de áreas de aplicação específicas, como gráficos computacionais 3-D). Saber alguma lógica formal e álgebra booleana é bom. Algum embasamento em matemática finita (incluindo a teoria dos conjuntos finitos, análise combinatória e teoria dos grafos) pode ser útil.

Muito mais importante: você precisa ser capaz de pensar logicamente e seguir as cadeias de raciocínio exato, a forma como os matemáticos fazem. Embora o conteúdo da maior parte da matemática não vá ajudá-lo, você irá precisar da disciplina e inteligência para lidar com a matemática. Se lhe falta a inteligência, há pouca esperança para você como um hacker, se você não tiver a disciplina, é melhor cultivá-la.

Acho que uma boa maneira de descobrir se você tem o que é preciso é pegar uma cópia do livro de Raymond Smullyan *What Is The Name Of This Book?*. Os lúdicos enigmas lógicos de Smullyan representam muito do espírito dos hackers. Ser capaz de resolvê-los é um bom sinal; desfrutar ao

resolvê-los é um ainda melhor.

## **Que linguagem devo aprender primeiro?**

XHTML (dialeto mais recente do HTML) se você ainda não souber. Há um monte de livros brilhantes e cheios de hype sobre HTML lá fora, e lamentavelmente poucos bons. O que eu mais gosto é o [HTML: The Definitive Guide](#).

Mas HTML não é uma linguagem de programação completa. Quando estiver pronto para começar a programar, eu recomendaria começar com [Python](#). Você vai ouvir um monte de gente recomendando Perl, e Perl ainda é mais popular do Python, mas é mais difícil de aprender e (na minha opinião) menos bem concebido.

C é realmente importante, mas também é muito mais difícil do que qualquer Python ou Perl. Não tente aprendê-lo primeiro.

Usuários de Windows, não se contentem com o Visual Basic. Ela vai lhes ensinar maus hábitos, e ele não é portátil fora do Windows. Evitem.

## **Que tipo de hardware eu preciso?**

Costumava ser aqueles computadores pessoais que eram bastante limitados e pobres em memória, suficientes mas que impunham limites artificiais no processo de aprendizado de um hacker. Isso deixou de ser verdade em meados da década de 1990. Qualquer máquina de um Intel 486DX50 acima é mais que suficientemente poderoso para o trabalho de desenvolvimento, X, e as comunicações da Internet, e os menores discos que você pode comprar hoje em dia são exageradamente grandes o suficiente.

A coisa mais importante na escolha de uma máquina na qual aprender é saber se o seu hardware é compatível com Linux (ou compatível com BSD, você deveria escolher aquele caminho). Novamente, isso será verdade para quase todas as máquinas modernas. As únicas áreas realmente pegajosas são modems e placas wireless, algumas máquinas possuem hardware específico para Windows que não funcionam com o Linux.

Existe um FAQ sobre a compatibilidade de hardware, a versão mais recente está [aqui](#).

## **Eu quero contribuir. Você pode me ajudar a escolher um problema para trabalhar?**

Não, porque eu não conheço seus talentos ou interesses. Você tem que ser auto-motivado ou você não vai crescer, é por isso que ter outras pessoas escolhendo seu caminho quase nunca funciona.

Tente isto. Acompanhe a rolagem dos anúncios dos projetos pelo [Freshmeat](#) por alguns dias. Quando você ver um que o faz pensar "Legal! Eu gostaria de trabalhar nisso!", junte-se a ele.

## **Preciso odiar e esmurrar a Microsoft?**

Não, você não precisa. Não que a Microsoft não seja repugnante, mas havia uma longa cultura hacker muito antes da Microsoft e ainda haverá uma muito tempo depois que a Microsoft virar história. Qualquer energia que você gastar odiando a Microsoft seria melhor gasta ao amar o seu ofício. Escrever bom código - isso irá esmurrar a Microsoft suficientemente bem sem poluir seu carma.

## **Mas o software de código aberto não deixará os programadores incapazes de ganhar a vida?**

Isso parece improvável - até agora, a indústria de software de código aberto parece estar criando empregos ao invés de tirá-los. Se ter um programa escrito é um ganho econômico líquido em relação não tê-lo escrito, um programador será pago tendo sido ou não o programa de código aberto depois que ele for feito. E, não importa o quanto software "livre" é escrito, sempre parece haver mais demanda por aplicações novas e personalizadas. Eu escrevi mais sobre isso nas páginas da [Open Source](#).

## **Onde posso pegar um Unix livre?**

Se você não tem um Unix instalado em sua máquina ainda, em algum outro lugar desta página eu incluí indicações de onde pegar os Unix livres mais usados. Para ser um hacker você precisa de motivação e iniciativa e da habilidade de educar a si próprio. Comece agora...