

Lifelong Kindergarten: Cultivating Creativity through Projects, Passion, Peers, and Play

Mitchel Resnick, MIT Media Lab
Publicado pela MIT Press (2017)

Trechos do Capítulo 2: Projetos

© 2017. Não copie, compartilhe ou distribua sem permissão do autor.

Pessoas que fazem e criam

Em janeiro de 2009, em um grande anfiteatro do campus do MIT, assisti à inauguração de Barack Obama como o 44º presidente dos Estados Unidos. O lugar estava lotado, com mais de 500 pessoas, e um vídeo do discurso inaugural de Obama foi projetado em duas grandes telas na sala. Como o público era formado por muitos cientistas e engenheiros do MIT, não é surpreendente que a reação mais intensa tenha sido quando Obama declarou: "Vamos restaurar a ciência ao seu devido lugar". O lugar foi preenchido por aplausos.

Mas não foi essa frase do discurso inaugural que chamou minha atenção. Para mim, o momento mais memorável foi quando Obama disse: "Foram as pessoas que se arriscam, que fazem, que criam coisas — algumas reconhecidas, mas normalmente homens e mulheres cujo trabalho normalmente é deixado nas sombras — que nos carregaram no longo e tortuoso caminho em direção à prosperidade e à liberdade."

As pessoas que se arriscam. Que fazem. Que criam coisas. Elas são os alunos X, os pensadores criativos. Elas foram a força motriz da mudança econômica, tecnológica e cultural na história. Hoje, todos precisam arriscar, fazer e criar coisas; não necessariamente para mudar o rumo da história, mas para mudar o rumo de suas próprias vidas.

Ao usar a frase *pessoas que criam coisas*, Obama estava fazendo uma referência implícita a um movimento que estava apenas começando a se espalhar pela nossa cultura: o Movimento Maker. Ele começou como um movimento de base, em garagens e centros comunitários, entre pessoas apaixonadas por fazer coisas, e por compartilhar suas ideias e criações umas com as outras. Em 2005, o movimento ganhou força quando Dale Dougherty lançou a revista *Make.*, que celebra as alegrias de construir, criar e inventar coisas. A revista tinha o intuito de democratizar o fazer, mostrando como todos podem se envolver em atividades de "faça você mesmo". A primeira edição descreveu "coisas incríveis que pessoas comuns estão fazendo em suas garagens e quintais", oferecendo instruções sobre como criar uma pipa para tirar fotos aéreas, um barril termoeletrico para manter a cerveja gelada e, varas luminosas para criar padrões luminosos dinâmicos à noite.

No ano seguinte, em 2006, Dale organizou a primeira Maker Faire, descrita como "um festival de invenções, criatividade e engenhosidade para toda a família". Foram exposições e oficinas

sobre como fazer jóias, móveis, robôs; quase tudo que você possa imaginar. Na última década, centenas de Maker Faires surgiram ao redor do mundo, atraindo milhões de engenheiros, artistas, designers, empreendedores, pedagogos, pais e crianças.

Para várias pessoas, o que mais chama a atenção no Movimento Maker é a tecnologia. Houve uma proliferação de novas tecnologias, como impressoras 3D e cortadoras a *laser*, que permitiram que as pessoas desenvolvessem, produzissem e personalizassem objetos físicos. Muita gente está animada com o potencial dessas tecnologias para os negócios e prevê que o Movimento Maker dará início a uma nova revolução industrial, na qual pequenos negócios (ou até mesmo indivíduos) poderão fabricar, em escala, produtos que anteriormente dependiam de grandes fábricas.

Eu me sinto atraído pelo Movimento Maker por razões diferentes. Acredito que ele tenha o potencial de ser não apenas um movimento tecnológico e econômico, mas também um movimento de aprendizagem, que oferece novas formas de as pessoas se envolverem com experiências de aprendizagem criativa. Conforme as pessoas fazem e criam, elas têm a oportunidade de se desenvolver como pensadoras criativas. Afinal de contas, *criar* está na raiz da *criatividade*.

O que talvez seja ainda mais importante é que o Movimento Maker incentiva as pessoas a trabalharem em projetos — o primeiro dos quatro Ps da aprendizagem criativa. As matérias da revista *Make*: e as exposições da Maker Faire não ensinam apenas técnicas para fazer; elas apoiam uma abordagem de aprendizagem baseada em projetos, na qual pessoas aprendem novas ideias, habilidades e estratégias enquanto trabalham em projetos pessoalmente significativos para elas. Dale Dougherty considera projetos "as unidades básicas do fazer".

Eu senti pessoalmente o poder dos projetos quando estava crescendo. Quando era criança, eu gostava de vários esportes: beisebol, basquete, tênis, entre outros. Mas não gostava de apenas praticar esportes; eu gostava ainda mais de "fazer" esportes. Eu estava constantemente inventando novos esportes para praticar com meu irmão e meu primo. Tive a sorte de ter um quintal para construir e brincar, e sorte também de ter pais que permitissem que eu transformasse o quintal em um local de trabalho para os meus projetos.

Em uma das minhas férias, cavei o quintal para criar meu próprio campo de golfe em miniatura. Foi uma experiência contínua de aprendizagem. Comecei cavando buracos simples no chão para criar os buracos de golfe, mas percebi que eles perdiam o formato com o tempo, então comecei a colocar latas de alumínio nos buracos. Isso funcionou até que choveu e as latas ficaram cheias de água, difíceis de esvaziar. Minha solução: cortei os dois lados das latas antes de colocá-las no solo para que a água pudesse ser drenada pelo fundo.

Conforme eu adicionava paredes e obstáculos no campo de minigolfe, precisava descobrir como a bola ricocheteava neles. Isso me deu um contexto motivacional para aprender a física de colisões. Passei horas calculando e medindo ângulos para que eu pudesse fazer a bola de

golfe bater nos obstáculos e cair no buraco. Essa experiência foi mais memorável do que qualquer aula de Ciências que eu tive em sala.

No caminho, comecei a compreender não só o processo de criação de um campo de minigolfe, mas também o processo geral de se fazer coisas: como começar com uma ideia inicial, desenvolver planos preliminares, criar a primeira versão, testar, pedir que outras pessoas experimentassem, revisar os planos com base no que acontece e continuar fazendo isso, várias vezes. Ao trabalhar no meu projeto, estava ganhando experiência com a Espiral da Aprendizagem Criativa.

Por meio desse tipo de projetos, comecei a me ver como alguém capaz de fazer e criar coisas. Comecei a ver as coisas do mundo de uma nova forma, pensando em como elas foram feitas. Como é feita uma bola ou um taco de golfe? Comecei a pensar em outras coisas que eu poderia fazer.

Se hoje você acessar o site da revista *Make*: (makezine.com), verá várias matérias que descrevem projetos de golfe em miniatura, com títulos como "Faça você mesmo um minigolfe de mesa" e "Putt urbano: golfe em miniatura 2.0". As tecnologias evoluíram desde que construí meu campo de golfe em miniatura, quase 50 anos atrás. Agora é possível produzir obstáculos customizados usando uma impressora 3D ou cortadoras a *laser*, é possível inserir sensores nos obstáculos para acionar motores ou ligar LEDs quando a bola de golfe passar por um obstáculo.

Ainda tenho orgulho do "antiquado" campo de minigolfe que eu construí quando era criança. Mas também estou animado com as novas tecnologias, que podem expandir os tipos de projetos que as crianças podem criar e inspirar novas crianças a se tornarem pessoas que fazem e criam coisas.

Aprender fazendo

Ao longo dos anos, vários pedagogos e pesquisadores defenderam o *aprender praticando*, argumentando que a melhor forma de aprender é estar ativamente envolvido em *praticar* algo, aprender por meio de atividades mão na massa.

Mas na cultura do Movimento Maker, não é suficiente *praticar* algo: você precisa *fazer* algo. De acordo com essa ética, as experiências de aprendizagem mais valiosas ocorrem quando você está ativamente envolvido no planejamento, construção ou criação de algo — quando você *aprende fazendo*.

Se você quiser entender melhor as conexões entre fazer e aprender, e como apoiar o "aprender fazendo", a melhor maneira é ler as obras de Seymour Papert. Tive a sorte de trabalhar com Seymour por vários anos no MIT. Mais do que qualquer outra pessoa, Seymour

desenvolveu as bases teóricas do aprender fazendo, além de tecnologias e estratégias de apoio. Seymour deveria ser considerado o santo padroeiro do Movimento Maker.

Seymour adorava todas as dimensões da aprendizagem: compreender, apoiar, fazer. Depois de fazer um Doutorado em Matemática na Universidade de Cambridge, em 1959, Seymour se mudou para Genebra para trabalhar com o grande psicólogo suíço Jean Piaget. Por meio de cuidadosa observação e entrevistas com milhares de crianças, Piaget descobriu que as crianças construíam conhecimento a partir de suas interações cotidianas com pessoas e objetos do mundo. O conhecimento não é despejado nas crianças, como água em um vaso. Em vez disso, as crianças estão constantemente criando, revisando e testando suas próprias teorias sobre o mundo quando brincam com seus brinquedos e amigos. De acordo com a teoria *construtivista* de aprendizagem de Piaget, crianças constroem o conhecimento ativamente, não o recebem passivamente. Crianças não *recebem* ideias, elas *fazem* ideias.

No início dos anos 1960, Seymour se mudou de Genebra, na Suíça, para Cambridge, Massachusetts, para assumir uma posição no MIT. Com isso, Seymour estava se mudando do epicentro de uma revolução no desenvolvimento infantil para o epicentro de uma revolução na tecnologia computacional. E ele passou as décadas seguintes fazendo conexões entre essas duas revoluções. Quando Seymour chegou ao MIT, computadores ainda custavam centenas de milhares de dólares ou mais e eram usados apenas em grandes empresas, agências do governo e universidades. Mas Seymour previu que computadores eventualmente se tornariam acessíveis a todos, até mesmo crianças, e viu na computação uma forma de transformar a maneira como crianças aprendiam e brincavam.

Seymour logo emergiu como líder em uma batalha intelectual sobre como introduzir computadores na educação. A maioria dos pesquisadores e pedagogos adotou uma abordagem de *ensino assistido por computadores*, em que computadores assumem o papel de professores: apresentando informações e ensinando alunos, conduzindo testes para medir o que os alunos aprenderam, depois adaptando o ensino com base nas respostas dos alunos às perguntas do teste.

Seymour tinha uma visão radicalmente diferente. Para ele, computadores não eram substitutos de professores, mas sim um novo meio de expressão, uma nova ferramenta para fazer coisas. Em 1971, cinco anos antes do primeiro computador de uso pessoal ser lançado, Seymour foi coautor (com Cynthia Solomon) de um artigo chamado "Twenty Things to Do with a Computer" ("Vinte coisas para se fazer com um computador"). O artigo descrevia como crianças poderiam usar computadores para desenhar imagens, criar jogos, controlar robôs, compor músicas, entre outras atividades criativas.

A abordagem de Seymour usou como base o que ele havia aprendido com Piaget: ver crianças como construtoras ativas do conhecimento, não como recipientes passivas. Seymour foi um passo além, argumentando que crianças constroem o conhecimento de forma mais eficaz quando se envolvem ativamente na construção de coisas no mundo, ou seja, quando estão fazendo coisas. Seymour chamou sua abordagem de *construcionismo*, porque junta dois tipos

de construção: conforme crianças constroem coisas no mundo, elas constroem novas ideias em sua mente, o que incentiva que elas construam novas coisas no mundo e assim por diante, em uma espiral infinita de aprendizagem.

Para dar vida a essas ideias, Seymour e seus colegas desenvolveram uma linguagem de programação para crianças, chamada Logo. Até então, a programação era vista como uma atividade especializada, acessível apenas a pessoas com conhecimento matemático avançado. Mas Seymour via a programação como uma linguagem universal usada para fazer coisas em um computador e argumentou que todos deveriam aprender a programar.

Em seu livro *Mindstorms*, Seymour comparou a abordagem de ensino assistido por computadores, na qual "o computador é usado para programar a criança", com sua própria abordagem, na qual "a criança programa o computador". Sobre o processo de aprender programação, ele escreve que uma criança "adquire uma sensação de domínio sobre uma peça da mais moderna e poderosa tecnologia e estabelece um contato íntimo com algumas das ideias mais profundas das ciências, da matemática e da arte de construção de modelos intelectuais".

Quando a linguagem Logo foi desenvolvida, crianças a usavam principalmente para controlar os movimentos de uma "tartaruga" robô (chamada assim devido ao seu casco hemisférico usado para proteger seus componentes eletrônicos). Conforme os computadores de uso pessoal se tornaram mais comuns, no final dos anos 1970, as crianças usavam a Logo para desenhar imagens na tela, digitar comandos como "forward 100" ("para frente 100") e "right 60" ("para direita 60") para dizer à "tartaruga da tela" como se mover, virar e desenhar. Conforme as crianças escreviam programas em Logo, elas aprendiam conceitos matemáticos de forma significativa e incentivadora, no contexto de projetos que elas consideravam importantes.

Durante os anos 1980, milhares de escolas ensinaram milhões de alunos a programar em Logo, mas o entusiasmo inicial não durou. Vários professores e alunos tinham dificuldade de aprender a programar com Logo, porque a linguagem era cheia de sintaxe e pontuação não intuitivas. Para piorar, a linguagem Logo era muitas vezes apresentada em atividades que não eram de interesse de professores nem de alunos. Várias salas de aula ensinavam Logo como um objetivo final, não como uma forma de os alunos se expressarem e explorarem o que Seymour chamou de "ideias poderosas". Não demorou para a maioria das escolas passar a usar computadores de outras formas. Elas começaram a ver os computadores como ferramentas para entregar e acessar informações, não para fazer e criar, como Seymour havia imaginado.

As ideias de Seymour sobre aprender fazendo estão voltando a ganhar peso, conforme mostrado pelo surgimento do Movimento Maker. Embora o trabalho de Seymour com a Logo tenha começado há mais de 50 anos e seu livro fundamental *Mindstorms* tenha sido publicado em 1980, suas ideias principais nunca foram tão importantes e pertinentes quanto hoje.

Fluência

Nos últimos anos, houve um aumento no interesse em aprender programação, ou codificação, como é chamada hoje em dia. Agora existem milhares de aplicativos, sites e oficinas que ajudam crianças a aprender a codificar. Nosso programa Scratch é parte dessa tendência, mas com uma diferença importante.

A maioria dos cursos introdutórios de programação é baseada em desafios ou *quebra-cabeças*. As crianças precisam criar um programa para mover um personagem virtual por alguns obstáculos até alcançar a linha de chegada. Por exemplo, mover o robô BB-8 do Guerra nas Estrelas para pegar uns pedaços de sucata sem passar pelo bandido, ou programar R2-D2 para levar uma mensagem aos pilotos rebeldes. Ao criar programas para resolver esses quebra-cabeças, crianças aprendem habilidades básicas de programação e conceitos de ciência da computação.

Com Scratch, o foco é em *projetos* em vez de quebra-cabeças. Quando apresentamos o Scratch para crianças, nós as encorajamos a criar suas próprias histórias interativas, jogos e animações. Elas começam com ideias e as transformam em projetos que podem compartilhar com outras pessoas.

Por que focamos em projetos? Nós vemos a programação como uma forma de fluência e expressão, assim como a escrita. Quando você aprende a escrever, não é suficiente aprender ortografia, gramática e pontuação. É importante aprender a contar histórias e comunicar suas ideias. O mesmo se aplica à programação. Quebra-cabeças podem ser bons para aprender a gramática básica e a pontuação da programação, mas não para você aprender a se expressar. Imagine como seria aprender a escrever fazendo apenas palavras-cruzadas. Isso poderia melhorar sua ortografia e vocabulário e poderia ser divertido, mas será que você se tornaria um bom escritor, capaz de contar histórias e expressar suas ideias de forma fluida? Eu acho que não. Uma abordagem baseada em projetos é o melhor caminho para ter fluência, seja na escrita ou na programação.

Mesmo que a maioria das pessoas não se torne um jornalista ou escritor profissional, é importante que todos aprendam a escrever. Isso também se aplica à programação, por razões similares. A maioria das pessoas não será um programador profissional ou cientista da computação, mas aprender a programar com fluência é uma habilidade valiosa para todos. Tornar-se fluente, seja na escrita ou na programação, ajuda a *desenvolver seu pensamento*, *desenvolver sua voz* e *desenvolver sua identidade*.

Desenvolvendo o seu pensamento

No processo de escrita, você aprende a organizar, refinar e refletir sobre suas ideias. Conforme você se torna um escritor melhor, você se torna um pensador melhor.

Conforme você aprende a programar, você também se torna um pensador melhor. Por exemplo, você aprende a dividir problemas complexos em partes mais simples. Você aprende a identificar problemas e depurá-los. Você aprende a aprimorar iterativamente e melhorar o *design* ao longo do tempo. A cientista da computação Jeannette Wing popularizou o termo *pensamento computacional* para se referir a esse tipo de estratégia.

Essas estratégias de pensamento computacional podem ser úteis em todos os tipos de atividades de *design* e resolução de problemas, não apenas em programação e ciência da computação. Ao aprender a depurar programas de computador, você estará se preparando para descobrir o que deu errado em uma receita na cozinha ou para seguir as instruções de alguém quando você se perder.

Resolver quebra-cabeças pode ser útil para desenvolver algumas habilidades de pensamento computacional, mas criar seus próprios projetos leva você além, ajuda a desenvolver sua voz e sua identidade.

Desenvolvendo a sua voz

Escrever e programar são formas de expressão, maneiras de comunicar suas ideias a outras pessoas. Quando aprende a escrever, por exemplo, você consegue mandar uma mensagem de feliz aniversário para um amigo, enviar um depoimento para o jornal local ou registrar seus sentimentos em um diário.

Vejo a programação como uma extensão da escrita, que permite que você "escreva" novos tipos de coisas, como histórias interativas, jogos, animações e simulações. Vou dar um exemplo da comunidade *online* do Scratch. Alguns anos atrás, na véspera do Dia das Mães, decidi usar o Scratch para fazer um cartão interativo de Dia das Mães para a minha mãe. Antes de começar, verifiquei se alguém já havia feito cartões de Dia das Mães no Scratch. Digitei "Dia das Mães" na caixa de busca e fiquei deslumbrado com as dezenas de projetos, vários deles criados nas últimas 24 horas, por procrastinadores como eu!

Por exemplo, um dos projetos começava com as palavras "FELIZ DIA DA MÃE" desenhadas no topo de um grande coração vermelho. Cada uma das 13 letras era interativa e virava uma palavra quando tocada pelo cursor do mouse. Conforme eu movia o cursor pela tela, tocando cada letra, foi revelada uma mensagem especial de Dia das Mães com 13 palavras: "Te amo e penso em você com carinho. Feliz dia das mães, mãe".

Quem criou este projeto estava claramente desenvolvendo sua voz com o Scratch, aprendendo a se expressar de novas maneiras e integrando o programa no fluxo de sua vida cotidiana. Acredito que, no futuro, será tão natural para os jovens se expressarem por meio de programas quanto por meio da escrita.

(Aliás, eu acabei não fazendo o cartão de Dia das Mães para a minha mãe. Em vez disso, mandei a ela links de dezenas de projetos de Dia das Mães que encontrei no site do Scratch.

Minha mãe, que foi pedagoga a vida inteira, respondeu com a seguinte mensagem: "Mitch, gostei muito de ver os cartões do Scratch de todas essas crianças... e eu amo ser mãe de um filho que ajudou a dar a essas crianças as ferramentas para comemorar assim!!!!").

Desenvolvendo a sua identidade

Quando as pessoas aprendem a escrever, começam a ver a si mesmas e seu papel na sociedade de um jeito diferente. O pedagogo e filósofo brasileiro Paulo Freire liderou campanhas de alfabetização em comunidades de baixa renda não apenas para ajudar pessoas a conseguir empregos, mas também para ajudá-las a aprender que "elas podem se fazer e refazer a si mesmas" (como ele escreveu em *Pedagogia da indignação*).

Eu vejo o mesmo potencial na programação. Na sociedade de hoje, as tecnologias digitais são um símbolo de possibilidade e progresso. Quando crianças aprendem a usar tecnologias digitais para se expressar e compartilhar ideias por meio da programação, elas começam a se ver de novas formas. Elas começam a ver a possibilidade de contribuir ativamente para a sociedade. Elas começam a se ver como parte do futuro.

Quando apresento o Scratch para jovens, eu me animo com o que eles criam e o que aprendem no processo. Mas o que mais me anima é a forma como vários usuários do Scratch começam a se ver como criadores, desenvolvendo confiança e orgulho da sua habilidade de criar coisas e se expressar fluentemente com novas tecnologias.