

Editora



Brasilina Passarelli

ORGANIZADORA

Ecosistema de Inovação na Educação

CASE EDUCAÇÃO BÁSICA - Guarujá

Volume 1



Atribuição-Não Comercial
CC BY-NC

Esta obra é de acesso aberto. É permitida a reprodução parcial ou total desta obra, desde que citada a fonte e autoria e respeitando a Licença Creative Commons indicada.

Catálogo na Publicação
Serviço de Biblioteca e Documentação
Escola de Comunicações e Artes da Universidade de São Paulo

E19 Ecossistema de inovação na educação [recurso eletrônico] : case educação básica –
Guarujá : volume 1 / organização Brasilina Passarelli. – São Paulo: ECA-USP : Escola
do Futuro, 2023.
PDF (111 p.) : il. color

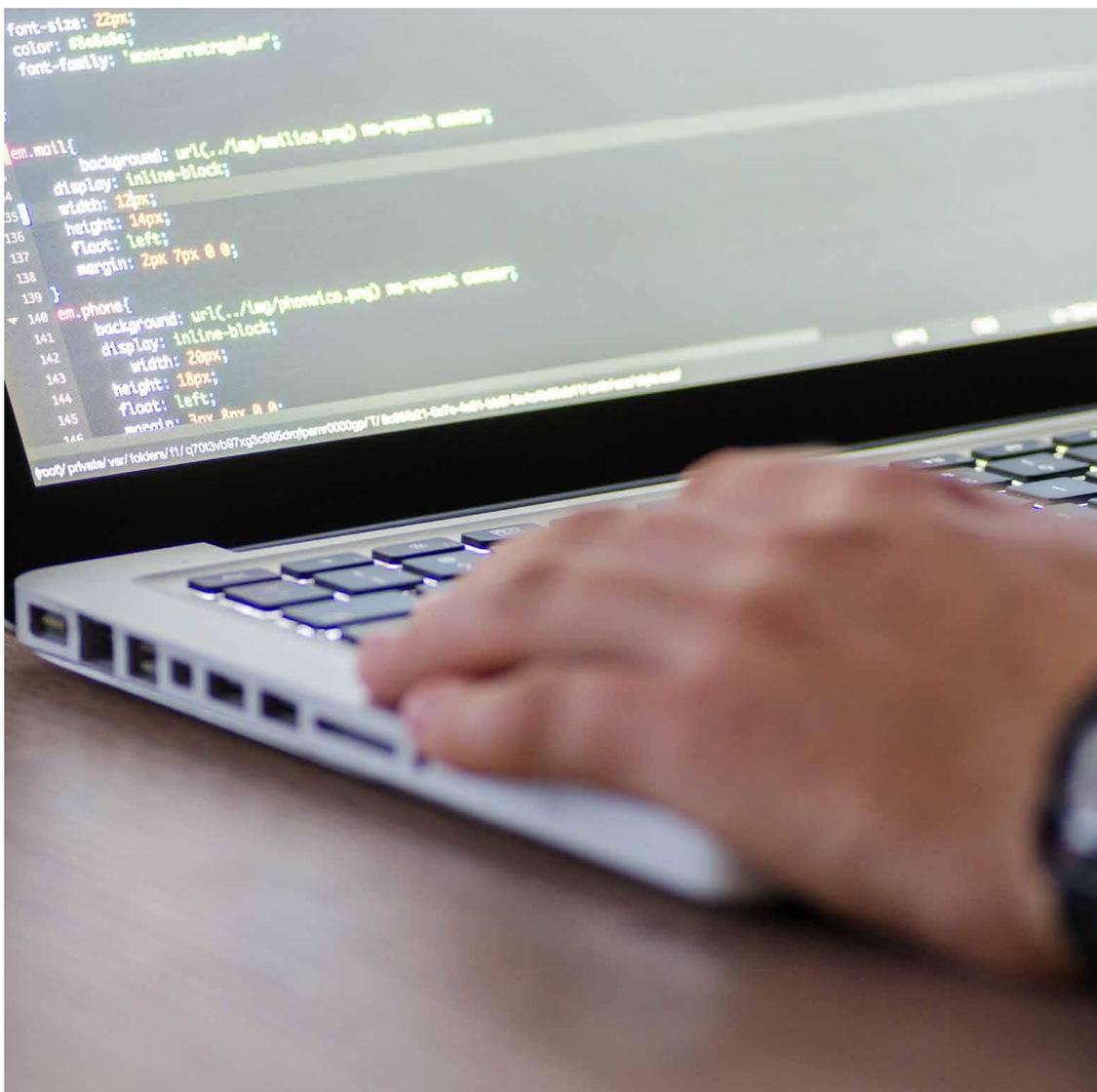
ISBN 978-65-88640-83-8

1. Tecnologia educacional. 2. Alfabetização midiática e informacional. 3. Jogos
eletrônicos. 4. Educação. I. Passarelli, Brasilina.

CDD 23. ed. – 371.3078

Elaborado por: Lilian Viana - CRB-8/8308

6 Pensamento Computacional



ALAN CÉSAR BELO ANGELUCI

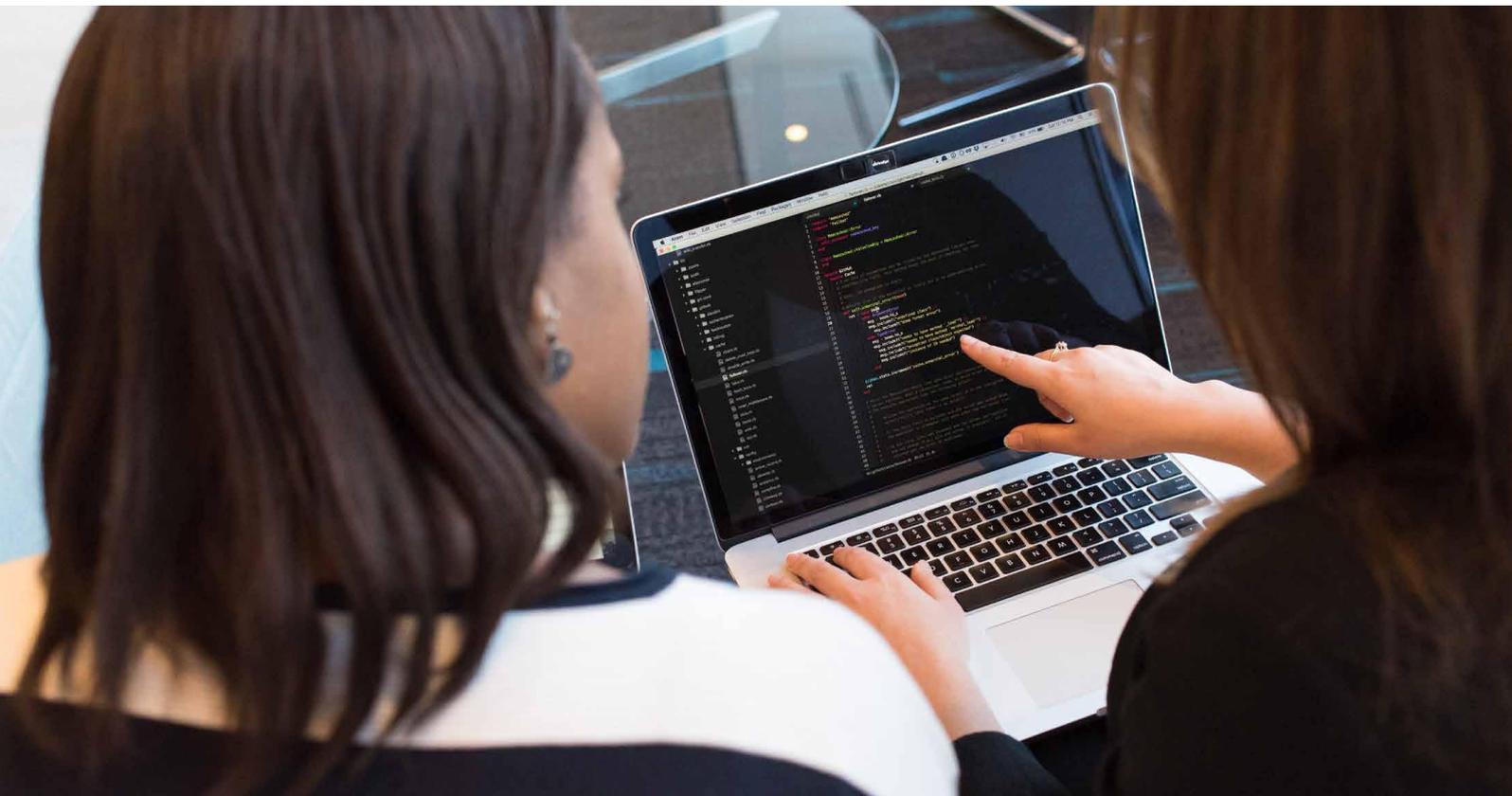
Índice



Clique nos capítulos para ser direcionado ao tema que deseja.

▶ Introdução _____	91
▶ O que é o pensamento computacional? _____	93
▶ A programação (coding) _____	96
▶ O processo de criação: a prototipagem de REA _____	101
▶ Scratch: uma linguagem de programação para educadores _____	106
▶ Créditos das Imagens e Referências _____	110

INTRODUÇÃO

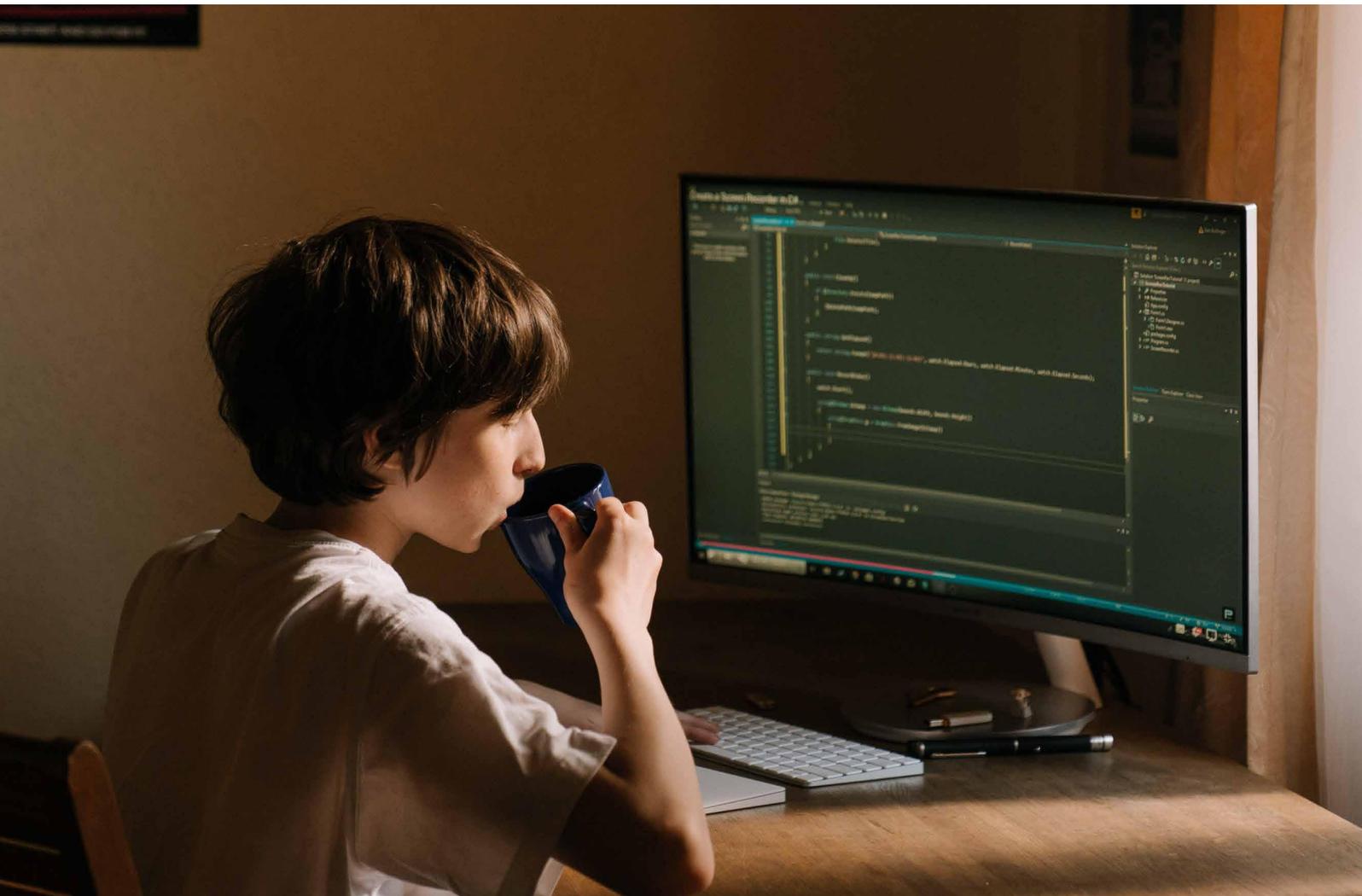


Para compreendermos a ideia de pensamento computacional e sua relação com a programação (coding), devemos primeiramente definir o que a literatura no campo da educação entende como elementos-base desse referencial. Assim, este módulo vai nos ajudar a reconhecer o significado de **DECOMPOSIÇÃO, RECONHECIMENTO DE PADRÕES, ABSTRAÇÃO** e **ALGORITMOS**. Esse é um importante ponto de partida para que nós, professores, possamos aprender como o pensamento computacional requer uma nova atitude diante das criações digitais. Essa mudança de perspectiva é fundamental para que não somente nós possamos criar recursos educacionais abertos (REA) originais e inovadores, mas para que também nossos alunos reconheçam esse novo olhar e passem a enxergar a programação como algo natural e presente em nosso dia-a-dia, tornando-se também produtores de conteúdos significativos.

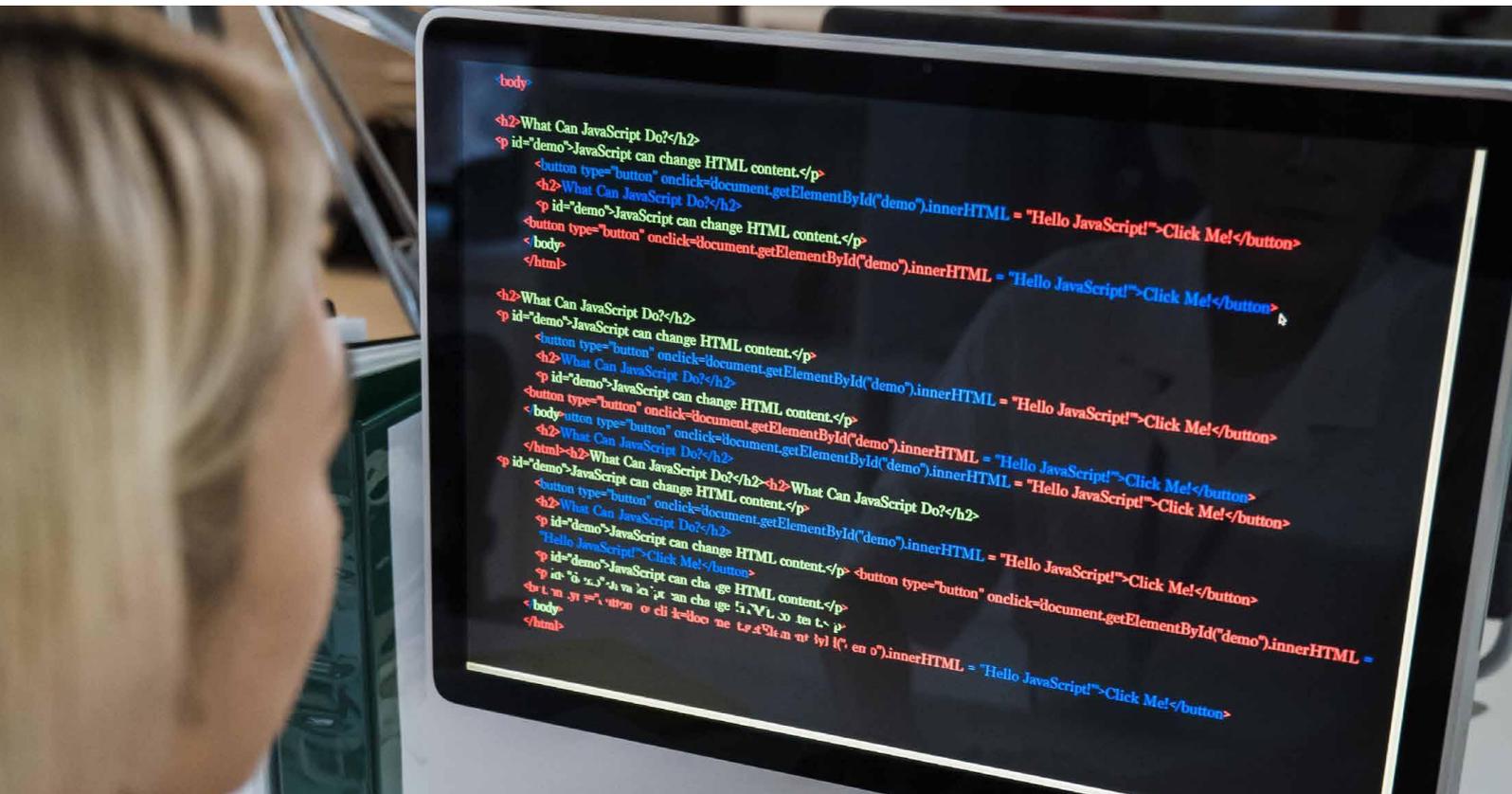
Também, vamos buscar entender como funciona a criação dentro do pensamento computacional, explorando as etapas que envolvem o processo de prototipagem. Na prototipagem, é fundamental ter uma ideia estruturada em objetivos de aprendizagem, um desenvolvimento que estabeleça critérios que nos ajudem a definir as melhores ferramentas de autoria e uma prototipagem planejada no nível adequado à aplicação proposta.

Por fim, iremos explorar o Scratch, uma linguagem de programação aberta com objetivos educativos que possui uma grande comunidade mundial de colaboradores. A partir dela é possível criar animações, jogos e outros programas interativos ideais para transformar aquela sua sequência didática em algo lúdico, envolvente e significativo!

E então, vamos programar?



O QUE É O PENSAMENTO COMPUTACIONAL?



SAIBA MAIS

O pesquisador dinamarquês Stig Hjarvard é um dos precursores da teoria da midiatização. Ela vem da ideia de uma presença cada vez mais intensa da mídia na cultura e na sociedade, não se limitando somente à formação da opinião pública, mas, também, em diversas instituições sociais, como a escola.

Essas cada vez mais necessitam de recursos de mídia, o que inclui competências para representar a informação, construir relações e obter atenção com ações comunicativas. Ele publicou um artigo muito interessante sobre o tema na revista Matrizes, da USP.

Para muitos de nós professores, atuar em sala exige uma dose extra de energia. A prática da sala de aula requer uma atualização constante não só de nossos materiais de apoio e metodologias, mas, sobretudo, para a apropriação de novas tecnologias. Em um mundo cada vez mais **MIDIATIZADO**, em que nossas interações são constantemente balizadas pelo uso de Tecnologias Digitais de Informação e Comunicação (TDIC), o desafio se torna cada vez maior: como tornar as novas tecnologias parceiras de um processo de aprendizagem cada vez mais significativo para o professor e para o aluno?

O primeiro entrave que boa parte dos docentes encontra refere-se à aprendizagem de uma nova linguagem, que é a digital. O pesquisador russo Lev Manovich diz que a linguagem das novas mídias se caracteriza, fundamentalmente, pelo processo de **COMPUTADORIZAÇÃO** que as linguagens tradicionais sofreram com a evolução da Internet e sua popularização a partir dos anos 90 do século XX (MANOVICH, 2001), transformando a maneira pela qual nos relacionamos com elas.

Isso quer dizer que, diferentemente de décadas atrás, grande parte dos conteúdos multissemióticos e multilinguísticos que usamos hoje são, em essência, uma massa de dados digitais – já que são emulados por computadores. Se, por um lado, esses conteúdos digitais permitiram a formação de uma grande rede de colaboração, compartilhamento e reuso de materiais didáticos, por outro têm gerado angústias relacionadas ora à capacidade do professor em manipular e lidar com o digital, ora com as dificuldades de infraestrutura para utilizá-los (restrições de rede de Internet, falta de equipamentos etc.).

SAIBA MAIS

A computadorização da mídia pode ser entendida a partir do processo de digitalização. A mídia passa a ser agora representada por códigos digitais – bits de “zero” e “uns” e pode ser partimentada em unidades fractais de dados discretos. Esses dados podem ser segmentados e terem suas características e funções alteradas, reeditadas, reproduzíveis e remixadas. A mídia se torna, então, uma grande base de dados manipuláveis.

A boa notícia é que esse cenário tem mudado significativamente nos últimos anos. Existe um sem-número de novos recursos educacionais abertos (REA) que podem ser criados por meio de ferramentas de autoria que automatizam boa parte do processo de edição do conteúdo, criando atalhos e facilitando o processo para quem o desenvolve. Trocando em miúdos, hoje em dia é cada vez mais fácil se apropriar das ferramentas digitais de criação, abrindo caminhos para o desenvolvimento de REA interativos e multimídias. Os estudos na área de user experience (UX) têm evoluído muito e os instrumentos de criação de objetos digitais estão se tornando mais simples, gratuitos e acessíveis. Hoje é possível a qualquer pessoa desenvolver REA significativos e inovadores, por menor que seja o seu conhecimento em softwares.

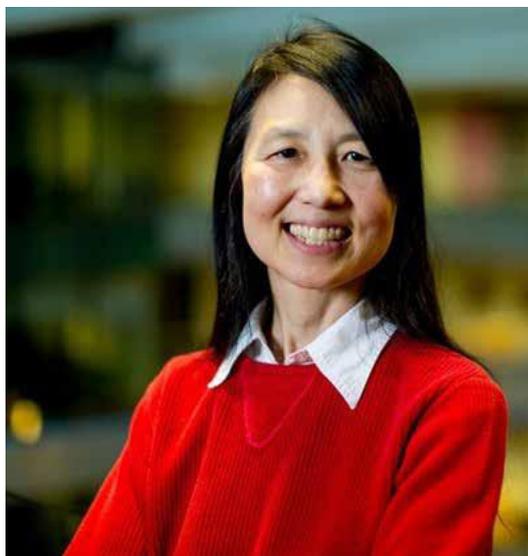
Se a dificuldade não está na ferramenta, onde reside o ponto central da tensão? É no pensamento computacional. Muitos de nós tendemos a criar uma solução para uma questão maquínica aplicando um modelo de pensamento do qual já estamos acostumados ou diferente do que foi utilizado para conceber o computador.

De forma genérica, pode-se dizer que cada linguagem possui uma lógica de funcionamento, um conjunto de códigos, regras e padrões. Por exemplo: é possível aprender mandarim, uma das variações da língua chinesa mais faladas do mundo, por mais difícil que pareça ser. Em algum tempo, se aprende o pinyin, o sistema de representação fonética chinês, e se pode escrever a frase “brasileiros aprendem chinês” — 巴西学中文. No entanto, para além do escrever, é preciso entender a lógica dessa língua chinesa de forma que faça sentido para o interlocutor. Com os computadores acontece algo parecido:

A DIFICULDADE NÃO ESTÁ EM LIDAR COM A FERRAMENTA OU CÓDIGO, MAS SIM COM A LÓGICA DA COMPUTAÇÃO. UMA VEZ COMPREENDIDA A LÓGICA DA MÁQUINA E SUA FORMA DE OPERAÇÃO E FUNCIONAMENTO PARA A EXECUÇÃO DE TAREFAS, TUDO FICA MAIS FÁCIL.

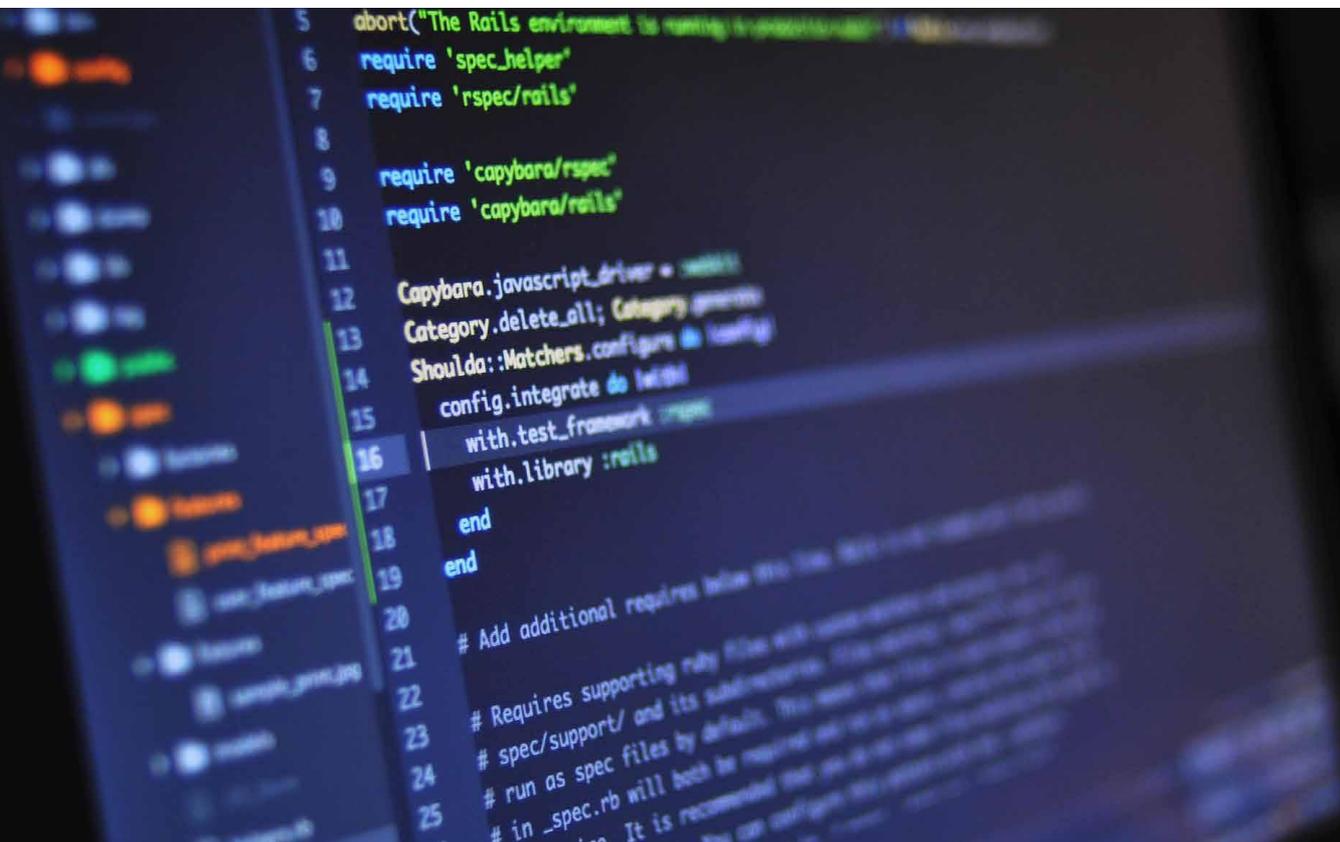
O conceito de pensamento computacional foi formulado pela pesquisadora norte-americana Jeannette Wing, e chama a atenção para que profissionais de diversas áreas pensem em problemas computacionais de forma analítica e desenvolvam soluções digitais na forma de sequenciamento de padrões e regras, ou seja, de forma algorítmica (WING, 2006). Por todo o mundo, muitas políticas públicas no campo da educação têm sido desenvolvidas e implementadas com o objetivo de permitir entender “como criar com as tecnologias digitais, e não simplesmente utilizá-las como máquinas de escritório” (VALENTE, 2016).

Portanto, se o pensamento computacional é um tipo específico de raciocínio lógico aplicado ao computador, é importante primeiramente entender do que ele é feito. Se todo conteúdo digital é um objeto computacional, ele precisa ser “lido” por um computador – e existem algumas linguagens para “escrever” (ou programar) esses objetos digitais. A compreensão do pensamento computacional passa, portanto, pelo entendimento da lógica da linguagem que escreve/programa os objetos digitais no computador: a conhecida linguagem de programação – ou coding. Como foi dito nas linhas anteriores, não basta escrever em mandarim – mas também compreender sua lógica de aplicação, seus limites e possibilidades de uso.



Jeannette Wing, vice-presidente corporativa da Microsoft, precursora do **Pensamento Computacional**

A PROGRAMAÇÃO (CODING)





Marca da linguagem de programação **Python**

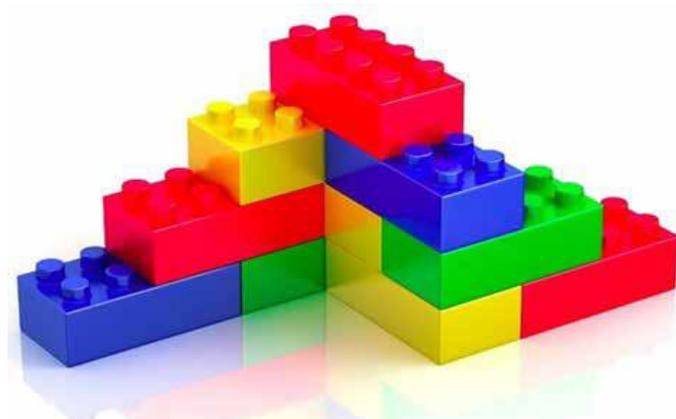
A linguagem de programação (coding) é um conjunto de formatos, regras e normas que devem ser seguidas para que seja possível criar um objeto de computador. Existem muitas delas por aí, desde a precursora COBOL, utilizada por muito tempo em operações bancárias, até as mais atuais como a Python, vocacionada para tratamento de dados digitais web complexos. Na maioria das vezes, interagimos com essas linguagens sem perceber: por exemplo, quando utilizamos ferramentas de edição de texto e de slides ou interagimos com aplicativos móveis de redes sociais. Não notamos, pois, na realidade, interagimos diretamente com uma “camada cultural” – a interface da máquina, cheia de ícones e símbolos dispostos em uma tela com estrutura amigável – que vai iniciando, executando e finalizando processos cíclicos na “camada maquina”, cheia de códigos e algoritmos opacos para muitos de nós.

A linguagem de programação é uma forma de ditar comandos e ações de maneira que o computador entenda o que deve ser feito. Quando se domina a linguagem e a lógica do computador se torna possível criar um REA digital, pois para ele terão sido programadas ações com objetivos pré-definidos que reagirão a determinados comandos.

Quando se narra uma história, por exemplo, é preciso pensar em uma narrativa estruturada em lugar, enredo e personagens, articulando-os em uma sequência de fatos que seguem uma “lógica” de storytelling – “introdução/ situação inicial”, seguida de “desenvolvimento/ complicação”, “clímax” e “desfecho/situação final” – tal como preconiza a lógica aristotélica no clássico livro “Poética”. Observe que, mesmo com algumas alterações desses componentes, há um conjunto de elementos que, concatenados, resultam na história narrada. Qualquer pessoa consegue entender a história pois os indivíduos compartilham de um mesmo repertório da lógica da linguagem narrativa.

Programar é, portanto, uma forma de contar uma história para um computador usando uma linguagem que ele entenda. Ao entender nossa história, ele será capaz de reagir da forma que esperamos, nos entregando exatamente o que demandamos. Seria muito mais fácil se simplesmente abrissemos nossa boca e contássemos ao computador o que gostaríamos de fazer. Por ora, bater um papo com um computador não vai dar muito certo: por mais que as tecnologias de Inteligência Artificial (IA) estejam em amplo desenvolvimento, há ainda um longo caminho para seu amadurecimento. O que nos resta fazer é entender as características e os requisitos da linguagem de programação para que possamos dialogar com os computadores.

Para isso, a literatura nos mostra que o pensamento computacional se estrutura em quatro processos principais: **DECOMPOSIÇÃO, RECONHECIMENTO DE PADRÕES, ABSTRAÇÃO E ALGORITMOS**. Vamos usar um jogo de peças encaixáveis para nos ajudar a entender esses processos.



COMO USAR O PENSAMENTO COMPUTACIONAL PARA CRIAR UMA ESCADA COLORIDA DE PEÇAS ENCAIXÁVEIS IGUAL À DA IMAGEM ABAIXO?

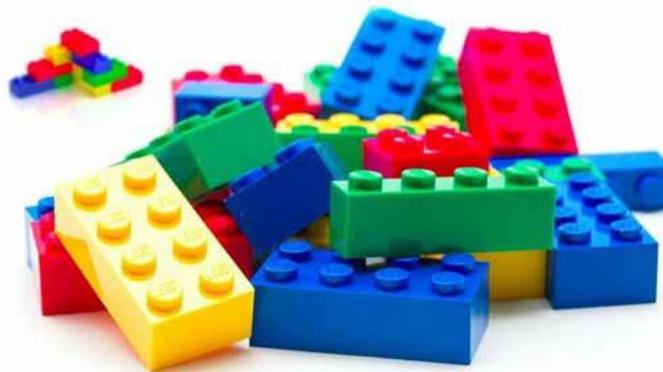
SAIBA MAIS

A IA é um campo de estudo que investiga maneiras de os computadores executarem funções que hoje são melhor executadas por humanos. Isso envolveria a capacidade de aprendizado da máquina (machine learning), em se adaptar de forma flexível e interpretar corretamente dados externos, acelerando e melhorando processos. Entenda um pouco mais sobre a IA nesta produção da TecMundo.

DECOMPOSIÇÃO

A ideia de decomposição é um primeiro processo muito importante para entender o pensamento computacional. Quando falamos em decomposição, queremos dizer que um problema grande e complexo (criar a escada colorida), para ser resolvido, deve ser dividido em várias partes de problemas menores. Um problema decomposto em tarefas menores se torna mais fácil de ser solucionado. Vamos pensar em estruturas fractais: todo objeto digital é um conjunto de módulos. Decompor um objeto digital é identificar quais partes formam o todo e separá-las, individualizá-las. Portanto, como eu “decomponho” a minha escada? Fracionando-a em vários pedaços.

DECOMPOSIÇÃO: A ESCADA É FORMADA POR VÁRIOS BLOCOS.



RECONHECIMENTO DE PADRÕES

O que esses padrões têm em comum? Quando decomparamos um problema em muitas outras partes menores, percebemos que existem algumas diferenças e similaridades em cada uma dessas outras partes.

SEMELHANÇAS:

- blocos de oito pinos;
- blocos de formato retangular;
- blocos encaixáveis entre si.

DIFERENÇAS:

- 3 blocos de cor amarela;
- 2 de cor verde;
- 2 de cor azul;
- 3 de cor vermelha.

RECONHECER ESSAS SEMELHANÇAS EVIDENTES AJUDA A CRIAR SOLUÇÕES NÃO SOMENTE PARA UM ÚNICO MÓDULO, MAS PARA VÁRIOS QUE SE ASSEMELHAM.

ABSTRAÇÃO

No processo de abstração, podemos ignorar (abstrair) as informações que denotam as diferenças e usar os elementos em comum para identificar uma solução válida para mais de um problema. Nesta etapa, você foca somente nos aspectos relevantes que são comuns a todas as partes e deixa para resolver depois os elementos que as diferenciam. No caso da nossa escada colorida, vamos ignorar a variável “cor”.

“Encaixe o bloco A da cor (vermelha/verde/azul/amarela) em cima de quatro pinos do bloco B da cor (vermelha/verde/azul/amarela), na direção (esquerda/direita/acima/abaixo).”

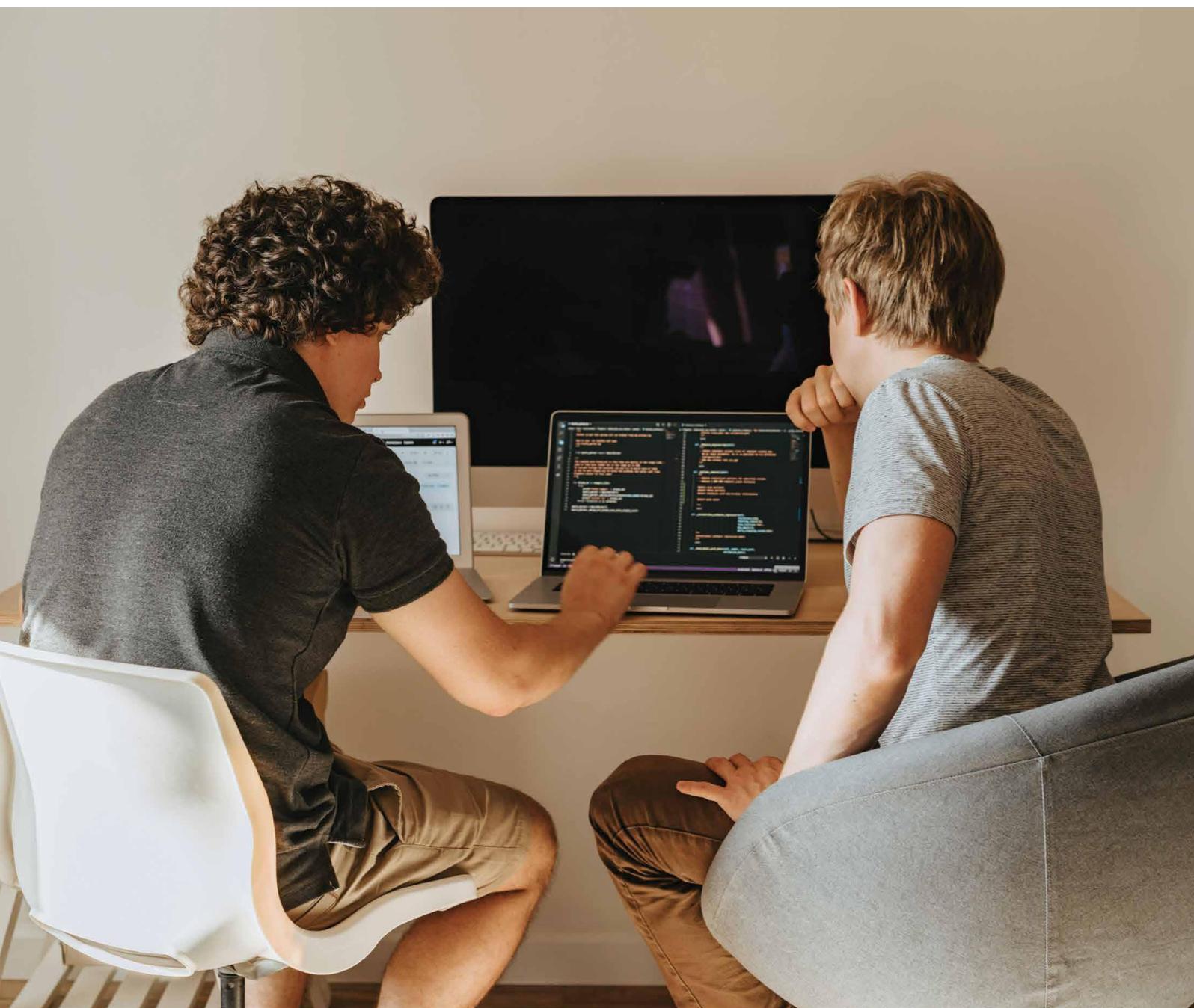
ALGORITMOS

A última etapa envolve a escrita do que chamamos de algoritmos – uma série de instruções sequenciais que devem ser processadas em passo-a-passo. Para escrever um algoritmo é necessário usar uma linguagem de programação. Para finalizar nosso exemplo, vamos simplesmente descrever essas sequências de instruções, como se fosse uma receita de bolo, de forma que qualquer pessoa possa reproduzi-la da mesma forma.

- Coloque o bloco 1 da cor azul na base.
- Encaixe o bloco 1 da cor vermelha em cima de quatro pinos do bloco 1 da cor azul, na direção esquerda.
- Encaixe o bloco 1 da cor amarela em cima de quatro pinos do bloco 1 da cor vermelha, na direção esquerda.
- ...

O pensamento computacional é, portanto, um modelo de raciocínio lógico que pode ser aplicado na solução de problemas que envolvam objetos digitais. Esses objetos, por serem computadorizados, são considerados programas, que podem ser escritos em linguagem de programação e lidos pelas máquinas, executando ações pré-estabelecidas com determinado objetivo. No caso dos REA digitais, tratam-se de programas devem criados partindo de um objetivo de aprendizagem.

Qualquer criação e desenvolvimento de programas digitais passa por diversas versões, das mais simples às mais robustas. Ao seguir a lógica do pensamento computacional, permite que possamos expandir e melhorar o objeto digital na medida em que eu faço testes de uso, checo o funcionamento dos comandos e amplio seus recursos e funções. Portanto, qualquer produto digital, antes de sua versão final, é chamado de protótipo e passa por diversos estágios de prototipagem. Protótipos podem ser muito úteis em muitos cenários de aplicação, inclusive os educacionais. Vamos tentar entender melhor como funciona a criação de REA digitais a partir da prototipagem.



O PROCESSO DE CRIAÇÃO: A PROTOTIPAGEM DE REA

AGORA QUE JÁ COMPREENDEMOS BEM O QUE É O PENSAMENTO COMPUTACIONAL E A LINGUAGEM DE PROGRAMAÇÃO, PODEMOS AVANÇAR NA IDEIA DE PROTOTIPAGEM DE UM REA DIGITAL. COMO CRIAR UM REA DIGITAL? QUAL O PONTO DE PARTIDA?



SAIBA MAIS

A pedagogia dos multiletramentos (pedagogy of multiliteracies) surgiu de um manifesto de pesquisadores vinculados ao Grupo da Nova Londres (New London Group) em 1996 que, atentos às mudanças tecnológicas que já estavam em curso na época, preconizavam que a construção grafocêntrica do conhecimento não era mais suficiente para dar conta da circulação das diferentes formas de produção, veiculação e consumo de conteúdo, atendendo para a necessidade de se pensar um design do conteúdo que abarque a diversidade de linguagens e culturais do contemporâneo global.

No campo da educação, o termo prototipagem tem emergido com grande força nos últimos anos. A pesquisadora brasileira Roxane Rojo nos explica que novos paradigmas de aprendizagem interativa têm se tornado imperativos diante de um mundo multiletrado e diverso culturalmente (ROJO, 2017). A ideia de um webcurrículo nos leva a pensar no uso de materiais digitais interativos, gamificados e que se utilizem de várias linguagens midiáticas (vídeos, infográficos, textos, áudios, etc.) de forma integrada e convergente. Dentro dessa perspectiva multiletrada, a prototipagem surge como parte de uma proposta de design de conteúdo, em que o professor e o aluno se apropriam de diversas tecnologias e linguagens com o objetivo de replanejar e recriar, criticamente, seus REA digitais.

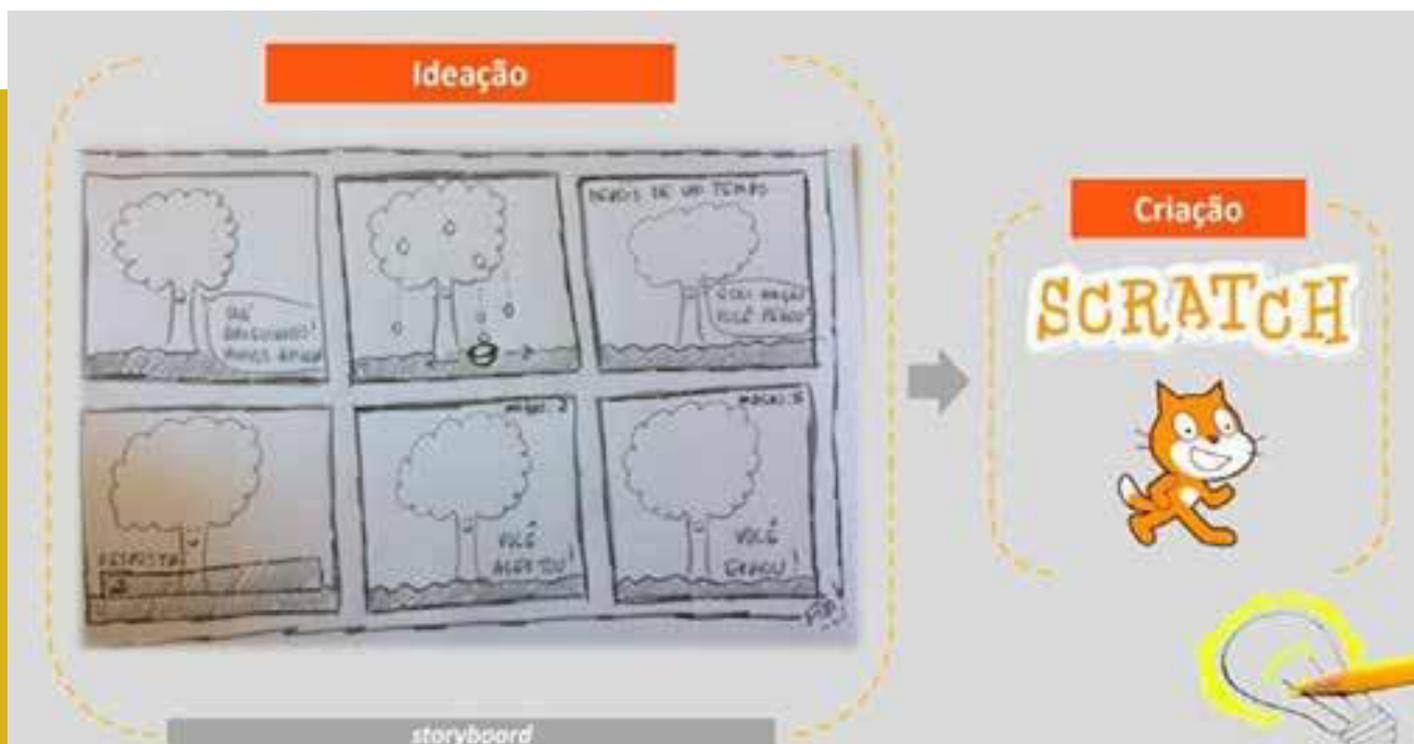
A prototipagem é, portanto, uma forma de projetar conteúdos digitais, baseada nos princípios do design, se utilizando de múltiplas linguagens e referências culturais. Em uma perspectiva autoral e colaborativa, professor e aluno deixam de somente utilizar um conteúdo digital pronto e passam a criar novos a partir de suas referências locais. É possível compartilhar protótipos e remixá-los, redistribuí-los e dar novos sentidos e usos, em uma rede de colaboração que só privilegia o processo de aprendizagem. Para criar um REA digital é preciso seguir procedimentos de prototipagem. Para nos auxiliar na compreensão desses procedimentos, vamos utilizar como exemplo a criação de um REA no formato de jogo digital educativo.

A IDEIAÇÃO: QUAL É O OBJETIVO DE APRENDIZAGEM?

A ideação é a primeira etapa da prototipagem de um REA digital. É importante que façamos os seguintes questionamentos:

- Que tipo de REA digital quero desenvolver?
- É um material didático com múltiplas linguagens integradas ou de uma única linguagem?
- Possui um formato específico ou misto?

O principal questionamento, no entanto, está relacionado aos objetivos de aprendizagem: o que se espera que o aluno aprenda com o REA? Essa aprendizagem está associada a determinada sequência didática? É um conteúdo paradidático ou está atrelado a determinado conceito presente no plano de ensino? Quando essas questões estiverem bem claras, parte-se para o estágio final da ideação – o desenvolvimento de um storyboard ou roteiro.



Partindo de nosso exemplo hipotético – um jogo digital educativo – devemos pensar nas respostas aos questionamentos anteriores. Trata-se, portanto, de um REA digital do tipo game e que deve se utilizar de várias linguagens integradas (áudio, texto e imagens). Quanto aos objetivos de aprendizagem, o jogo digital, direcionado aos alunos do Ensino Infantil, deve estimular a prática de contagem numérica. Para isso, se elabora um roteiro desse jogo a partir da formulação de uma ação interativa: o aluno deve movimentar um cesto com o objetivo de coletar o maior número de maçãs que despencam de uma macieira para, ao final, tentar acertar a quantidade de maçãs que ele coletou. A árvore cumpre o papel do personagem responsável por conduzir o aluno ao longo de sua jornada, dando instruções e informando, ao final, se o objetivo fora ou não cumprido. A partir da ideação concluída, passa-se para a etapa de criação na ferramenta de autoria.

O DESENVOLVIMENTO: QUAL É A FERRAMENTA DE AUTORIA IDEAL?

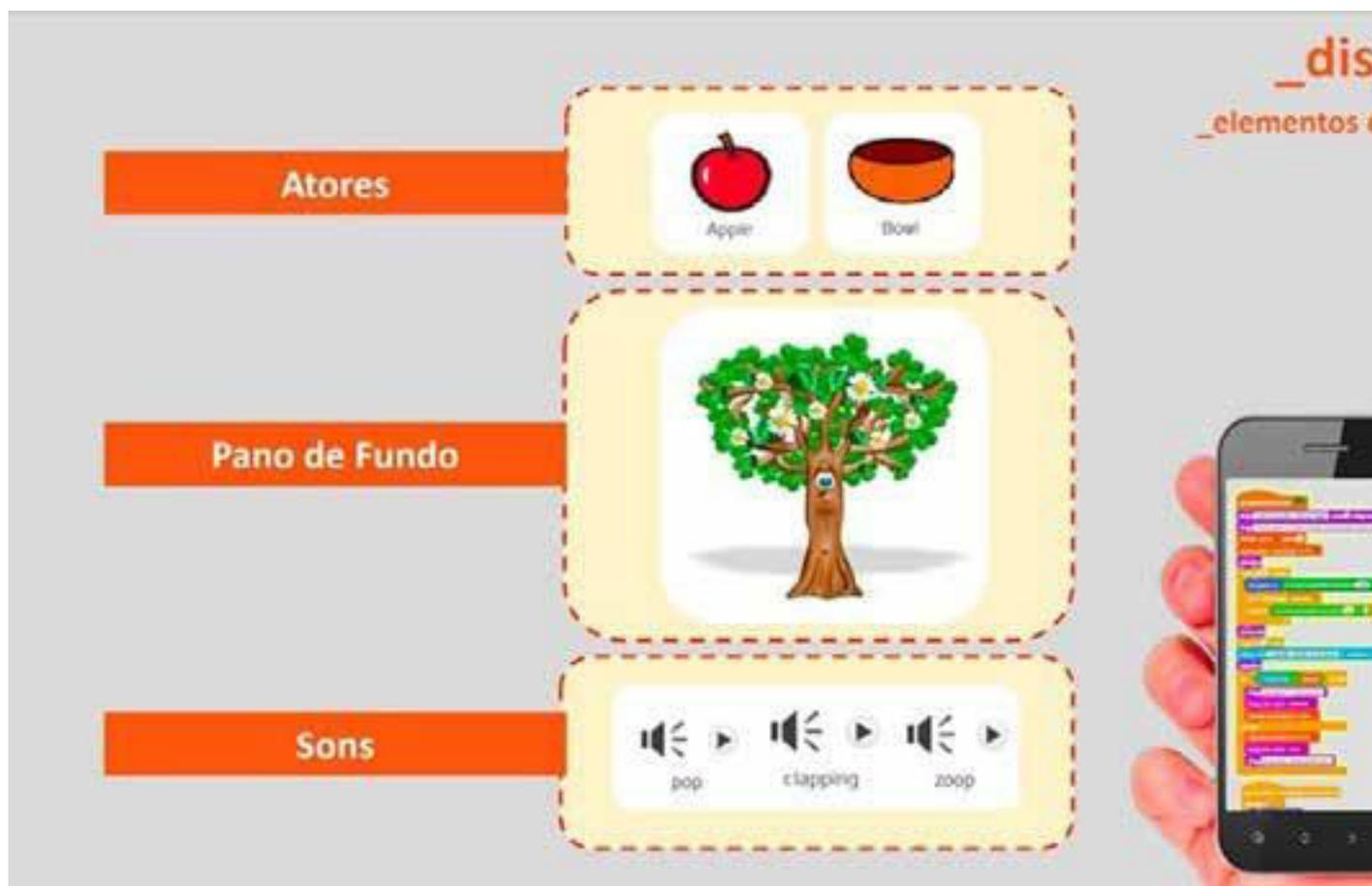
A imagem acima já dá um spoiler de uma das ferramentas de autoria mais utilizadas no campo educacional voltadas à prática do pensamento computacional e da linguagem da programação: o Scratch. Sobre ele, falaremos mais adiante em detalhes.

O que importa, neste momento, é compreender que diversas ferramentas de autoria estão disponíveis na web para serem utilizadas como instrumentos de prototipagem digital. Muitas delas são gratuitas e de interface amigável. No entanto, é importante que o professor reflita novamente sobre o objetivo do seu REA: o objeto que quero desenvolver requer que tipo de recursos linguísticos e interativos?

O quadro abaixo sintetiza um pouco do universo de possibilidades de linguagens e de uso de ferramentas de autoria.

Objetivo criativo	Linguagem	REA	Ferramentas de autoria
Narrativas audiovisuais	Audiovisual	Vídeo e Tags	PC / Smartphone / Windows Movie Maker / YouTube
Narrativas sonoras	Sonora	Podcast	PC / Smartphone / Audacity / Soundcloud
Narrativas remixadas	Remixagem/mashup	Instagram	Smartphone / conta no Instagram
Mapas de geolocalização interativos	Espacial / interativa	My maps	PC / Conta no Google
Aplicações de realidade aumentada	Realidade aumentada	Aurasma Studio/HP Reveal	Smartphone / conta no HP Reveal
Aplicativos móveis	Móvel	Fábrica de aplicativos	PC / Conta em Fábrica de Aplicativos
Linhas do tempo interativas	Temporal / interativa	Timeline JS ou Storymap	PC / Conta no Google
Documentos multimídia on- line em grupos	Colaboração síncrona e assíncrona	Suíte Google Drive	PC / Conta no Google
Animações interativas	Multimídia interativa	Games e histórias interativas	PC / Conta no Google / Scratch

No caso específico do Scratch, lembrando de nosso exemplo, observe que ele permite que criemos recursos multimídia do tipo de texto, imagens e áudio, a partir de uma programação que permita ao usuário manipular as setas da direita e da esquerda do teclado de um computador para movimentar a cesta. Neste caso específico, os "atores" são os objetos que têm alguma ação interativa específica no jogo; o "pano de fundo" refere-se à imagem estática (a macieira) e os "sons", estão relacionados às ações interativas de erros, acertos e coleta de maçãs ao longo da interação. Observe que todo conteúdo de mídia (imagens e áudios) precisa estar previamente elaborado antes de se direcionar à plataforma. Dessa maneira, é importante que o professor levante, na etapa de ideação, todos os requisitos necessários para o desenvolvimento do REA antes de começar a programar.



A PROTOTIPAGEM: EM QUAL NÍVEL DE FIDELIDADE?

O nível de fidelidade em prototipagem está relacionado à capacidade de um produto entregar recursos e interações. Os chamados protótipos de baixa fidelidade normalmente são aqueles em formato de rascunho, que passam a ideia do projeto sem muitas preocupações com o design da interface, mostrando de forma simples como seria a interação do usuário do com objeto digital. Esse nível de prototipagem é mais utilizado na etapa de ideação.

Já a prototipagem em média fidelidade tem uma importante relação com a arquitetura da informação do objeto. Ele já se torna algo concreto, aplicado à ferramenta de autoria, com algum nível de navegação e interação. Esses protótipos não possuem recursos avançados de interação nem interfaces arrojadas, mas permitem um nível satisfatório de interação e podem cumprir o papel em diversos REA, mesmo apresentando ainda limitações ou até mesmo falhas de programação.

Por fim, os protótipos de alta fidelidade estão próximos à ideia de produtos finalizados. Nesses casos, é possível simular o fluxo completo de interação e todas as funcionalidades previstas, com o mínimo de falhas possíveis.

No caso de nosso exemplo de jogo, a ideia é demonstrar como é possível criar um protótipo de alta fidelidade de um game educativo usando o pensamento computacional e a linguagem de programação. Vamos avançar na seção final deste nosso guia para entender um pouco mais sobre o Scratch e ver nosso game educativo finalizado e em funcionamento.

SAIBA MAIS

É importante ressaltar que não há uma padronização ou regramento para se definir se determinado protótipo é de baixa, média ou alta fidelidade. Os critérios para diferenciar esses níveis costumam estar relacionados com cada projeto específico. O Balsamiq é uma excelente ferramenta de prototipagem de baixa, média e alta fidelidade. Explore o Balsamiq e depois nos conte o que achou!



SCRATCH: UMA LINGUAGEM DE PROGRAMAÇÃO PARA EDUCADORES





O Scratch é uma linguagem de programação desenvolvida pelo Media Lab do Massachusetts Institute of Technology, o famoso MIT, nos Estados Unidos. Em 2007, Mitchel Resnick e sua equipe pensaram em uma linguagem de programação que fosse acessível a todos – principalmente àqueles que estão começando a programar. O foco inicial eram pessoas acima dos oito anos de idade, que estavam aprendendo conceitos matemáticos e computacionais. Com o passar do tempo, o projeto cresceu e se tornou uma comunidade mundial de criadores de objetos digitais interativos. A ferramenta tem vocação para a área de educação e se dedica, sobretudo, à formação de uma comunidade colaborativa. Em seu site é possível explorar recursos voltados a professores, alunos e outros materiais instrucionais, além de um extenso repositório de criações que permite o compartilhamento e remixagem das histórias animadas, jogos e outros programas interativos criados lá.

O Media Lab tem uma longa tradição no desenvolvimento de linguagens de programação acessíveis e voltadas à educação. É lá que surgiu a linguagem LOGO, desenvolvida por Seymour Papert no final dos anos 1960. Também foi por meio desse laboratório que se desenvolveu a ferramenta de autoria para aplicativos móveis MIT App Inventor, criada em 2010 na esteira das primeiras versões do Scratch.



CENÁRIOS POSSÍVEIS DE APLICAÇÃO

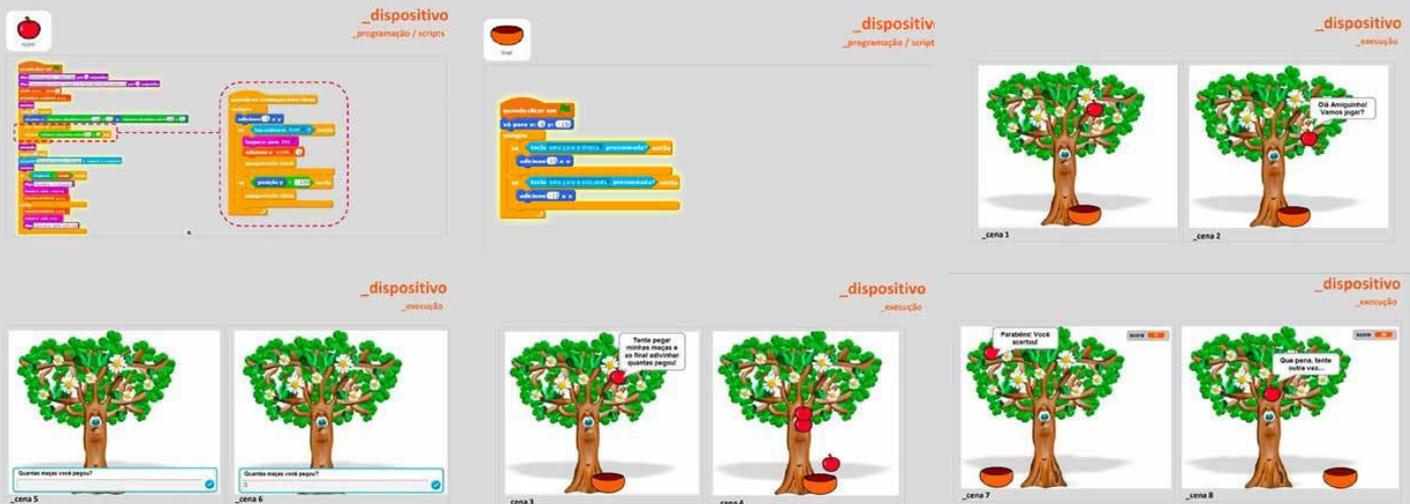
O Scratch disponibiliza um conjunto de recursos interativos que permite a criação de histórias, animações e jogos. Basta explorar a criatividade e utilizar os recursos disponíveis, desde REA simples relacionados à animação de objetos e tipos, até narrativas mais complexas que envolvem uso de recursos multimídia e games. A dica é navegar e explorar os tutoriais amplamente disponíveis não só na área de ideias do site mas também no YouTube, repleto de gameplays e vídeos de outros desenvolvedores.

Activity Guides

What do you want to make with Scratch? For each activity, you can try the Tutorial, download a set of Coding Cards, or view the Educator Guide.

- Animate a Name**
Animate the letters of your name, initials, or favorite word.
- Imagine a World**
Imagine a world where anything is possible.
- Make a Chase Game**
Make a game where you chase a character to score points.
- Make Music**
Choose instruments, add sounds, and press keys to play music.
- Create a Story**
Choose characters, add conversation, and bring your story to life.
- Make It Fly**
Choose any character and make it fly!
- Pong Game**
Make a bouncing ball game with sounds, points, and other affects.
- Animate a Character**
Bring characters to life with animation.
- Video Sensing**
Interact with a project using the Video Sensing extension.

[See All Tutorials](#)



‘HANDS ON!’: PROTOTIPANDO APLICATIVOS EDUCATIVOS

Voltando ao nosso exemplo do game educativo, chegamos ao ponto que passamos a programar de fato, a partir da linguagem do Scratch, utilizando a lógica do pensamento computacional. Observe que, para tornar nossos atores (“maçã” e “cesto”) de fato interativos e tornar nossa proposta realmente um jogo, precisamos decompor a ideia em sequências de ações menores (abertura do jogo e diálogo convocando para o ‘call to action’; coleta de maçãs e interação com o cesto; finalização do jogo com a pontuação final). Também foram necessárias as abstrações, como a destacada na imagem na sequência, que se relaciona às ações contínuas de coleta de maçãs. O mesmo podemos dizer para a programação do cesto.

Mas essa etapa é mais interessante com a mão na massa. Que tal vir preparado para desenvolver o game da maçã em nosso workshop?

Até breve!



CRÉDITOS DAS IMAGENS

PEXELS e Google



REFERÊNCIAS

MANOVICH, Lev. **The language of new media**. MIT press, 2001.

ROXANE, R. **Entre Plataformas, ODAs e Protótipos**: Novos multiletramentos em tempos de WEB2. The Specialist, v. 38, n. 1, 2017.

VALENTE, José Armando. **Integração do pensamento computacional no currículo da educação básica**: diferentes estratégias usadas e questões de formação de professores e avaliação do aluno. Revista E-curriculum, v. 14, n. 3, p. 864-897, 2016.

WING, Jeannette M. **Computational thinking**. Communications of the ACM, v. 49, n. 3, p. 33-35, 2006.