

# Cultura Maker na Educação utilizando Programação em Blocos

Giordano Arantes <sup>1</sup>

Jessica Miranda <sup>2</sup>

Miriam Lúcia Barbosa <sup>3</sup>

Rômulo Franco <sup>4</sup>

**RESUMO:** Conhecida como a Cultura do Fazer ou cultura do faça você mesmo, a cultura *maker* ganha adeptos no mundo todo. Essa nova cultura é uma das formas de remodelação do ensino em sala, ela é caracterizada pela forte identificação que o indivíduo cria durante o envolvimento e participação na criação de conteúdos e produtos. Em se tratando de identificação com o que se cria, o indivíduo estabelece um importante elo afetivo o que torna o aprendizado mais efetivo. Como parte do aprendizado, algum dos princípios defendidos pela cultura *maker* são a colaboração, a criatividade e a sustentabilidade quando recursos materiais e de fácil uso, acesso e apropriação podem ser utilizados na elaboração de produtos ou conteúdos. Com isso, esse artigo disserta sobre a Cultura *Maker* e suas diferentes formas de implantação na Educação, contribuindo especificamente com atividades, conceitos e metodologias e principalmente a Programação Visual em Blocos. Esta por sua vez, considerada como aliada na formação de crianças para um perfil apropriado em um futuro muito mais dependente de tecnologia. Ainda, para reforçar as oportunidades como ferramenta com alto potencial a ser aplicada no ensino-aprendizagem, apresentamos um mini-curso, inclusive com material disponível para download sobre o Google MIT App Inventor, bem como, um tutorial para desenvolvimento de aplicativos utilizando o conceito de Programação Visual em Blocos para Deficientes Visuais.

**PALAVRAS-CHAVE:** Cultura *Maker*; App Inventor; Programação; Deficientes; Educação.

## Breve introdução a temática

A cultura do fazer associada à tecnologia surgiu em meados dos anos 70 junto com o computador pessoal, porém o termo *maker* começou a ser usado com a conotação atual pela revista *Maker* sobre projetos tecnológicos Make. Criada em 2005 por Dale Dougherty, nos EUA. Em 2006, a publicação organizou a primeira Maker

---

<sup>1</sup> Mestrando em Educação, Faculdade de Educação da UNICAMP, Campinas, SP. e-mail: <giordanoarantes@gmail.com >

<sup>2</sup> Mestranda em Engenharia Elétrica e da Computação, Faculdade de Engenharia Elétrica e da Computação da UNICAMP, Campinas, SP. e-mail:< jessicasmiranda@gmail.com >

<sup>3</sup> Mestranda em Educação, Faculdade de Educação da UNICAMP, Campinas, SP. e-mail: <miriamluci@hotmail.com >

<sup>4</sup> Doutorando em Educação, Faculdade de Educação da UNICAMP, Campinas, SP. e-mail: <romulojosefranco@gmail.com >

Faire, uma feira que passou a ser ponto de encontro anual de adeptos do movimento em algumas cidades do mundo (Futura, 2018).

Ao estudar a cultura do fazer como princípio da cibercultura, Cardins (2014) afirma que “a cultura do faça você mesmo, na internet, é caracterizada pela criação de conteúdo, através do qual os usuários têm oportunidade de criar produtos, imprimindo suas opiniões, desejos, personalidades e criatividade” [...].

Em se tratando da programação, existem diversas linguagens que proporcionam a criação de aplicativos, softwares, websites entre outros. A programação é usada para solução de problemas e para isso é feita a lógica da programação, definindo a utilização dos códigos de cada linguagem para elaboração do programa.

Não obstante, o uso das tecnologias podem favorecer o ensino e possibilitar ferramentas que auxiliem os professores na aplicação dos conceitos trabalhados em suas aulas e na realização das tarefas de aprendizagem. Além de utilizar softwares que já existem, o professor pode aprofundar-se na elaboração de aplicativos que irão ajudá-los da maneira que ele necessitar.

Diante disso, o presente estudo objetiva dissertar sobre a Cultura Maker e suas diferentes formas de implantação na Educação, contribuindo especificamente com atividades, conceitos e metodologias e principalmente a Programação Visual em Blocos. Utilizou-se como metodologia a revisão bibliográfica e o relato de experiência. Assim sendo, este estudo aborda os seguintes tópicos: As Tecnologias e Cultura Maker no Desenvolvimento de Atividades em Sala de Aula, Programação visual baseada em blocos, Ferramentas para aplicação de Programação visual em blocos, Pensamento Computacional, Design Thinking, Ambiente do App Inventor, Aplicação de um Minicurso Utilizando o App Inventor, Tecnologias Para Deficientes Visuais, Tecnologias Assistivas e TIC's em Sala de Aula.

## **As Tecnologias e Cultura Maker no Desenvolvimento de Atividades em Sala de Aula**

Além do uso pessoal e profissional, a utilização das tecnologias no ensino aprendizagem, sistema de avaliações e demais áreas da educação vêm sendo grande foco de transformações e debates (Comitê Gestor da Internet no Brasil, 2016).

Em se tratando da cultura maker na escola, o professor Moisés Zylbersztajn afirma que na busca por uma pedagogia que privilegie o protagonismo do aluno, que produza colaboração e criatividade, atitude crítica e autonomia, a proposta é criar oficinas de invenções – espaços maker – na escola, onde, além de artefatos, seja produzida uma atitude de empoderamento e transformação da realidade dos alunos envolvidos e capaz de mobilizar a atenção dos adultos educadores para que experimentem um pouco dessa atitude maker em suas práticas (Padilha; Martinelli, 2015).

Cabe aqui destacar que, pode-se considerar que as expressões que constam no Projeto Político Pedagógico - PPP da escola bem como na Base Nacional Comum Curricular – BNCC priorizam as ações voltadas à cultura maker (criar, fazer, colaborar, produzir, investigar, cooperar, exercitar, compartilhar, participar...).

Assim sendo, consta na BNCC na 1ª competência geral – conhecimento – ao qual propõe valorizar e utilizar conhecimentos sobre o mundo físico, social, cultural e digital, com a proposta de entender e explicar a realidade, com o propósito de continuar aprendendo e colaborar com a sociedade, na 4ª competência geral – comunicação – recomenda utilizar as diferentes linguagens para expressar-se e partilhar informações, experiências, ideias, sentimentos e produzir sentidos que levem ao entendimento mútuo, bem como na 5ª competência – cultura digital – preconiza compreender, utilizar e criar tecnologias digitais de forma crítica, significativa e ética para comunicar-se, acessar e produzir informações e conhecimentos, resolver problemas e exercer protagonismo e autonomia, ou seja, subentende-se que o BNCC orienta a utilização da cultura maker na educação.

Não obstante, Sieves (2018) cita que o primeiro passo para implantar o movimento maker na educação é esse: criar uma comunidade. O autor esclarece que realidade financeira de muitas escolas não permite a aquisição de impressoras 3D, máquinas de corte, perfuração, entalhe ou arduino, contudo, segundo ele essa limitação não pode impedir a propagação da cultura maker. Dessa forma Sieves cita 3 dicas para implantar o movimento maker na educação, por meio do projeto pedagógico.

Inicialmente, Sieves (2018) afirma que para fortalecer o movimento maker na educação é necessário a **criação do espaço maker**, que pode ser um laboratório já existente ou uma sala de aula adaptada para as criações. Neste ambiente, é necessário disponibilizar vários materiais, como papéis coloridos, retalhos de tecidos,

materiais recicláveis, balões, colas, madeira, tintas, botões, massinha de modelar e muitas ferramentas. Com esses materiais, os estudantes poderão criar protótipos e tornar reais os seus projetos, de qualquer natureza. Segundo Sieves, é importante sugerir a criação de brinquedos, jogos, trabalhos artísticos e até produtos funcionais em outras áreas (grifo nosso).

Em seguida, o autor afirma é fundamental **estimular à comunidade maker**, ou seja, professores de todas as disciplinas podem se envolver com o movimento maker na educação. Algumas disciplinas têm um pouco mais de afinidade, como artes e ciências, mas o ideal é que todos os professores se envolvam com a cultura de “colocar a mão na massa”. Como o ambiente geralmente é compartilhado por várias turmas e professores, a organização é fundamental. É importante também definir critérios quanto ao bom uso do espaço e deixar clara a responsabilidade de cada aluno manter tudo organizado e limpo (grifo nosso).

Finalizando, Sieves (2018) cita que é indispensável realizar **intercâmbio entre makers** através da promoção de desafios, workshops e até intercâmbios entre comunidades makers. Essa interação pode ser tanto presencial, quanto por videoconferência. O importante é que os estudantes conheçam outras criações e mostrem os seus protótipos, descrevendo o que aprenderam no desenvolvimento do projeto e se ele pode resolver algum problemas sociais (grifo nosso).

Além disso, conforme afirmações de Padilha e Martinelli (2015) é de suma importância permitir que esse espaço esteja sempre aberto, sempre disponível para receber os alunos quando estes tiverem tempo e quiserem concluir alguma coisa que começaram.

Para as autoras existem quatro possíveis começos para implantar a cultura maker na escola: através do ensino de programação – possibilita perceber como funciona, de modo geral, a lógica que produz a inteligência nas máquinas; ensino de eletrônica e princípios de automação – induz um conhecimento essencial sobre o funcionamento dos circuitos elétricos básicos, base para a eletrônica digital; aplicativos para celular – criar aplicações úteis para resolver problemas de outros ou próprios, ensinando-os a otimizar o uso de seus aparelhos antes de descartá-los pela próxima novidade, e/ou até mesmo reciclar as partes úteis de um celular antigo. Há inúmeros programas de sustentabilidade mostrando caminhos e há bons softwares para desenvolvimento de aplicativos, como o *APP Inventor*, da Google/MIT;

impressora 3D – maquininhas geniais que nos transformam imediatamente em produtores, seu processo de fabricação é envolvente e muito intuitivo.

É importante destacar que para que a implantação da cultura maker seja eficazmente realizada na escola é fundamental que haja o envolvimento dos educadores na criação do espaço e da proposta. Pode-se começar adaptando espaços já existentes na escola, sendo relevante organizar o espaço junto com todos, especialmente com os alunos; promover a visitaç o de outros espa os e manter-se colaborando com eles, bem como, comprar poucos equipamentos, adquirindo outros a partir das necessidades que surgir o do uso.

Contudo, a proposta da cultura maker na educa o busca criar autonomia e criatividade, capacidade de modificar o mundo a seu redor e enfrentamento do consumismo, al m disso, tem grande potencial e conte do para trabalhar com diferentes  reas e projetos,   muito poderosa para concretizar ideias de forma  gil e pr xima dos usu rios al m disso, tamb m tem grande potencial no apoio   mudan a de cultura, mais autonomia e uma vis o mais colaborativa.

N o obstante, vale salientar que a cultura maker deve ser utilizada de maneira significativa, de forma que o aluno compreenda a import ncia dessa utiliza o e a funcionalidade dela dentro do contexto escolar, dessa maneira as disciplinas do conte do trabalho devem ser utilizadas com o objetivo de uma constru o conjunta do conhecimento de forma aut noma e participativa, que possibilite ao aluno perceber e compreender a import ncia da cultura maker para constru o desse conhecimento. Para isso,   essencial que a escola aproprie da cultura maker em seu PPP e que o professor concilie em seu plano de aula as funcionalidades, particularidades e especificidades da cultura maker dentro do conte do abordado, ou seja, utiliz -la como uma importante ferramenta de assimila o de conte dos e n o apenas como projetos de extens o de carga hor ria e/ou como oficinas.

### **Programa o visual baseada em blocos**

*“Saber programar vai se transformar no novo ler e escrever. Quem n o souber c digo ter  mais dificuldades para entender o mundo”, Hadi Partovi, Iraniano fundador ONG code.org*

Em raz o da velocidade do avan o tecnol gico na sociedade, tem se criado uma forte depend ncia tecnol gica exigindo novos pap is de profissionais para lidar

melhor com a informação. Atualmente estamos nos deparando com mudanças na forma como se comunica, vende, consome, se alimenta, locomove, trabalha e em todos os segmentos da sociedade há a tecnologia, seja em smartphone, dispositivos portáteis, televisores, computadores. Ainda, todos estes elementos estão interligados uns aos outros e a informação passa de um lado para o outro como que se tivesse vida própria. Nós como agentes participativos e maiores interessados na informação dependemos de entender e dominar a tecnologia para que possamos viver nos moldes da sociedade atual. Esta mudança cria também novas necessidades de mão de obra e de pensamento criativo com lógicas operacionais e funcionais apoiadas em estruturas educacionais muito diferentes das quais vivenciamos no século passado. Seria difícil encontrar algum segmento hoje na sociedade que não esteja dependente de algum tipo de tecnologia que envolva um simples aplicativo no Android. Assim, na essência do que Hadi Partovi menciona em sua frase sobre a necessidade de ensino de programação e tecendo um paralelo do analfabetismo funcional que presenciamos hoje no Brasil, essa realidade estará também presente em um futuro próximo com elementos um pouco diferentes, onde jovens e adultos que são avessos a tecnologias, serão como os analfabetos funcionais de hoje, no qual não entenderão e nem mesmo compreenderão a tecnologia e viverão com dificuldades nesta nova composição da sociedade.

Para programação existem diversas linguagens que proporcionam a criação de aplicativos, softwares, websites entre outros. A programação é usada para solução de problemas e para isso é feita a lógica da programação, definindo a utilização dos códigos de cada linguagem para elaboração do programa.

Cristovão (2008) sugere que, para que o aluno consiga resolver um problema, ele precisa ter um domínio completo da linguagem e isso implica em dedicação e tempo de estudos.

Segundo Gomes e Melo (2013) durante os cursos das linguagens de programação, é ensinado ao aluno o funcionamento de cada linha de código da linguagem, situação essa que pode não estimular o interesse dos alunos, desestimulando a continuidade dos estudos.

Com desafios para incentivar os alunos programarem, Wolber (2010) propôs um curso introdutório para alunos da Universidade de São Francisco sem conhecimento em programação, com o objetivo de apresentar o *App Inventor* e criar aplicativos para resolver problemas do cotidiano.

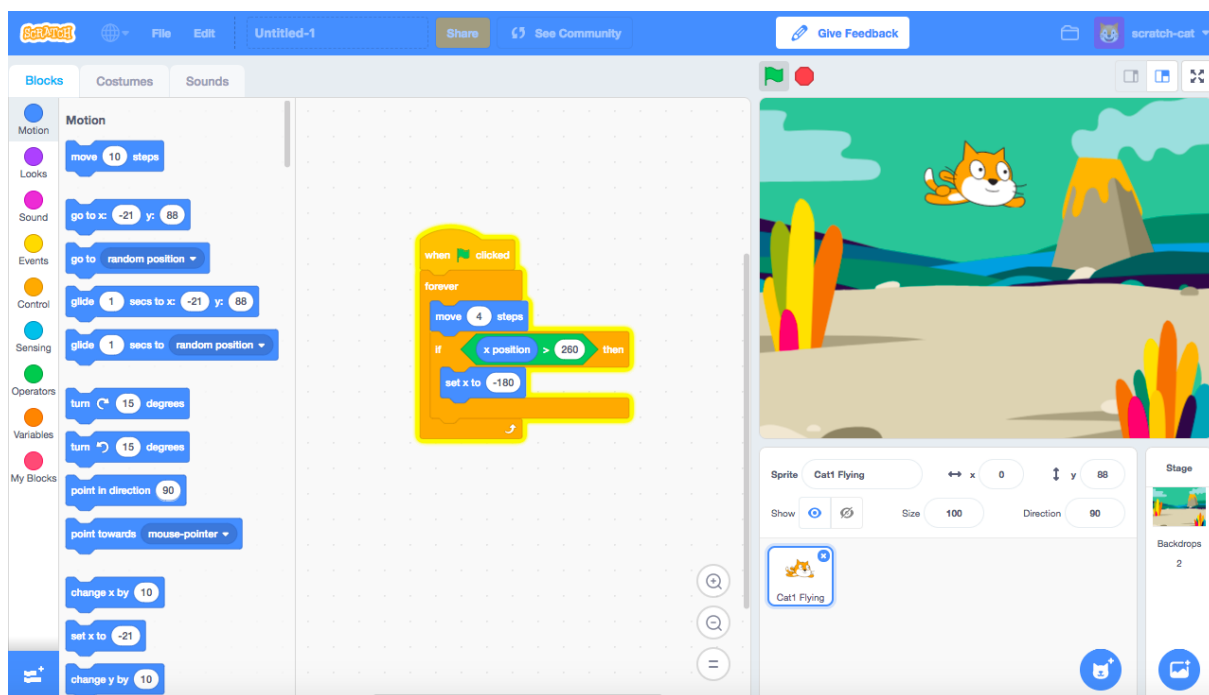
Durante o curso os alunos conseguiram criar seus aplicativos como, por exemplo, um aplicativo para encontrar onde a pessoa estacionou o carro. Wolber (2010) deixa claro que o interesse dos alunos com a plataforma foi de total desfastio, com um aprendizado satisfatório, além dos alunos demonstrarem uma grande motivação por criarem aplicativos eficientes e que podiam ser instalados em seus celulares *androids*.

A programação visual em blocos traz uma forma lúdica de aprender a falar, pensar e aprender a língua da tecnologia. A maneira pela qual se monta blocos lógicos com sequências, ao mesmo tempo se brinca. Como demonstrado na Figura 1, os ambientes estudados e apresentados por este estudo demonstram ambientes coloridos, com recursos inovadores, abstrai muito da complexidade e transforma a programação em código e de difícil aplicação no ensino do público infantil para algo possível e estimulante (Poredu, 2017). A programação em código muitas vezes exige o inglês, a estruturação do código exige muito conhecimento de lógica, ou seja, em si é de fato complexa, e às vezes impossível de aprender na visão de muitas pessoas. Já a programação visual baseada em blocos, o código fica em segundo plano, o que o programador vê são blocos semelhantes a um quebra cabeça, cada bloco carrega um objetivo para o contexto operacional que se deseja atingir. Além disso, os objetos pelos quais o bloco lógico tem que interagir são criados bastando que o programador selecione-os e arraste para a tela de desenvolvimento do aplicativo.

O MIT App Inventor e o Scratch são ambientes famosos de desenvolvimento para programação visual baseada em blocos. Estes ambientes foram desenvolvidos baseados na linguagem LOGO criada por Seymour Papert e sua equipe no MIT (Instituto de Tecnologia de *Massachussets*). Seymour Papert foi parceiro de Jean Piaget e a forma pela qual ele e sua equipe desenvolveram essa ideia, eles denominaram de teoria do construcionismo que descreve que o indivíduo por meio da máquina podia construir seu próprio conhecimento, resultando na produção de algo importante, seja em jogo, programa, algo artístico como uma música, ou um poema. Essa produção estaria muito mais ligada a algo ou objeto pertencente ao universo de necessidades do indivíduo, motivando-o a continuar. Além disso, a construção de algo pertinente a sua rotina, ou algo que seja importante caracteriza a identidade que se obtém fruto do laço criado durante o desenvolvimento do objeto. Essa motivação é a parte motriz é fundamental porque caracteriza um envolvimento afetivo, o que torna o

aprendizado efetivo deste complexo mundo que é a programação (Resnick et. Al, 1988).

Figura 1: Ambiente colorido do Scratch que permite criar histórias e animações interativas e jogos usando Programação visual baseada em Blocos



Fonte: <https://scratch.mit.edu>

## Ferramentas para aplicação de Programação visual em blocos

Atualmente para se obter uso, acesso e apropriação destes recursos, seja em sala de aula ou em casa é muito simples. Com um computador com processamento médio conectado a Internet pode se ter acesso gratuito a uma gama de aplicativos que contenham a abordagem de Programação visual em blocos. Destacamos aqui algumas ferramentas que podem ser usadas no Ensino-Aprendizagem de programação usando Programação Visual em Blocos e são todas gratuitas e geram aplicativos com licença e uso gratuito:

- 1) Scratch<sup>5</sup> - É possível criar histórias com animações interativas, jogos e compartilhá-las com demais usuários do Scratch em uma comunidade;
- 2) MIT App Inventor – é um ambiente intuitivo de programação visual que permite qualquer um – inclusive crianças – a construir aplicativos

<sup>5</sup> Scratch - <https://scratch.mit.edu/about/>



funcionais para dispositivos como tablets e smartphones. O MIT App Inventor gera aplicativos que podem ser usados em Smartphones com Android;

- 3) Google Blockly e o SantaTracker<sup>6</sup> – é uma biblioteca contendo um ambiente de desenvolvimento visual para aplicações web e dispositivos móveis;
- 4) Alice<sup>7</sup> – é um ambiente de programação visual e que contém um ambiente em 3D muito rico em gráficos. Diferente dos demais ambientes que rodam diretamente via browser, para desenvolver em Alice é necessário instalar o ambiente no computador;
- 5) Code Combat<sup>8</sup> – é um ambiente de programação em código mas que contém um ambiente visual que simula jogos medievais. A escrita do código é passo-a-passo e mantém toda a simplicidade desses ambientes para o aprendizado de programação;
- 6) Code.org – é na verdade um portal de uma ONG fundada por Hadi Partovi e é muito rico em conteúdo para ensino e aprendizagem de programação. Ele conta com vários vídeos explicativos, tutoriais, jogos e ambientes de desenvolvimento visual. Além disso, vários atores, artísticas e líderes da área da computação como Bill Gates, Mark Zuckerberg já participaram de vídeos do site.
- 7) A Hora do Código<sup>9</sup> – é um movimento iniciado pela Fundação Code.org para promover a programação pelo mundo todo. O evento vem ocorrendo desde 2013 e no ano de 2018 ocorrerá em novembro.

Com esta gama de ferramentas e sendo todas gratuitas e com documentação online a única necessidade para o aprendizado de programação é a necessidade de um computador com acesso a Internet.

## **Pensamento Computacional**

---

<sup>6</sup> Google Santa Tracker - <https://santatracker.google.com/intl/pt-BR/codeboogie.html>

<sup>7</sup> Alice - <https://www.alice.org/>

<sup>8</sup> Code Combat - <https://br.codecombat.com/>

<sup>9</sup> <https://hourofcode.com/br>

A programação visual em blocos tem vários benefícios, ela contribui com a criatividade, colaboração, comunicação e o pensamento crítico e criativo (Poredu, 2017). Assim, o aprendizado de programação leva o indivíduo a adquirir o pensamento computacional (YouthMobile, 2016). Embora esse tipo de pensamento seja mais propício em profissionais e cientistas já envolvidos com a computação, ela é universalmente aplicável em vários segmentos. Envolve um conjunto de habilidades como o pensamento e lógica recursiva, a abstração e generalização, a decomposição de partes de um problema maior em menores, e a solução de problemas relacionados a vida cotidiana de modo geral (Wing, 2006). Durante o entendimento de um problema, o programador é levado a ter um pensamento criativo embasado nas necessidades estruturais de algo que se está desenvolvendo mas que exige conhecimento computacional para aplicá-lo. Ou seja, o pensamento computacional exige que o programador pense no problema usando conhecimento em computação que ele já possui ou tem noção de como deveria ser. Neste caso, mesmo que ele ainda não possua um conhecimento completo dos detalhes da solução, ele sabe dos elementos possíveis dentro do universo computacional que resultaria em uma solução performática.

O pensamento computacional ajuda o indivíduo a resolver um problema desde a sua formulação, decompondo o problema em várias partes, permitindo a simulação de várias formas, criando abstração e generalizando partes do problema e que estas partes podem ser aplicadas em diferentes contextos ou reutilizá-las em outros problemas. Ao expressar e executar diferentes formas de solução do problema ocorre a simulação mental, o reconhecimento de padrões o que leva a um aperfeiçoamento da solução. Durante estas fases em que o indivíduo tem uma ideia diante de um problema, ele se questiona o que seria importante e eficiente como solução para resolver este problema. As fases são não sequenciais em que o indivíduo é levado a compreender em como pode decompor um problema, reconhecer padrões inerentes ao problema e a solução, a abstração e generalização e por fim, a sequência de passos lógicos para rastrear o caminho que vai de um ponto ao outro para a execução completa da solução, gerando ao final um algoritmo que como uma espécie de produto ou uma ideia.

Uma vez que este tipo de pensamento faz parte do cotidiano do indivíduo, a facilidade em criar soluções inovadoras para problemas será uma realidade intrínseca a ele. Tendo um viés mais útil e dinâmico na sociedade. Diante de um problema o

indivíduo se questiona o que necessariamente levaria a uma solução via processos computacionais, envolvendo uma sequência lógica de etapas. Contudo, o pensamento computacional está mais ligado em: **qual é o problema e o que necessariamente seria importante para uma solução** sem questionar de onde emerge o problema e o porquê de sua existência.

## **Design Thinking**

Ao projetar ideias usando o pensamento computacional, ideias inovadoras surgem incentivando o indivíduo a continuar sempre buscando soluções para problemas reais na vida em sociedade. Já o Design Thinking, é uma outra metodologia próxima do pensamento computacional, que embute todo o espectro de atividades de inovação com um espírito de projeto centrado no ser humano (Sebrae, 2018). Com isso, a inovação é alimentada pela compreensão direta, do que as pessoas querem e precisam em suas vidas e do que gostam ou não gostam sobre a maneira como determinados produtos são produzidos, embalados, comercializados, vendidos e suportados (Brown, 2008).

O Design Thinking contém um conjunto de etapas sequenciais e que são trabalhadas para entender primeiramente o problema que é colocado como desafio, e a principal diferença entre Pensamento Computacional e Design Thinking está na abordagem em fases iniciais ao retratar um problema e o caminho que se leva a solução. As principais fases do Design Thinking são (DTEducadores, 2012):

- 1) **Imersão** – o entendimento do problema em profundidade, usando de pesquisas e o envolvimento dos indivíduos alvo e profissionais. A ideia é transformar um problema em uma oportunidade com uma solução inovadora e centrada nas preocupações, motivações dos indivíduos alvo. Por isso, a empatia é trabalhada pelos profissionais a capturarem através da pesquisa com os interessados a buscar no âmago do problema o que se quer de fato resolver em um dado contexto. Nesta fase identifica-se também a origem e as causas do problema;
- 2) **Análise e Síntese** – os dados coletados da primeira fase são analisados através de insights e são organizados para criar padrões identificáveis, através de diagramas, associações entre entidades e atores envolvidos, como dependências gerando um organograma, mapas conceituais e critérios para nortear o entendimento do problema;

3) **Ideação** – como o próprio nome diz, a ideação é a fase onde o estímulo a criatividade ajudará a gerar soluções de acordo com o contexto do assunto trabalhado. Além disso, a colaboração é fator essencial neste processo. A colaboração permitirá que cada um exponha seus pontos de vista, as preocupações, a identificação das motivações e as possíveis soluções que cada um identifica, ou seja, a interpretação de cada profissional envolvido de um mesmo problema;

4) **Experimentação ou prototipagem** – é a etapa da validação das ideias geradas. Modelos de testes são criados e executados e a coleta de feedbacks juntos aos interessados. Os testes são executados mais de uma vez para aperfeiçoar as soluções encontradas através da legitimação de hipóteses que foram dimensionadas no início e mapear as dificuldades encontradas;

5) **Implementação** – após o aperfeiçoamento realizado com os testes, um produto, uma metodologia, um processo ou procedimento é criado e está apto a ser colocado em prática pelos interessados alvo.

Há muitas semelhanças entre as duas abordagens para solucionar um dado problema. Embora o pensamento computacional não tenha uma sequência como o Design Thinking, o Pensamento Computacional necessita que o programador tenha intrínseco as etapas de decomposição, abstração, reconhecimento de padrões e a criação do algoritmo e isso é algo pertinente ao próprio pensamento dele, ou seja, é um pensamento que ele passará a ter naturalmente frente a maioria dos problemas.

Contudo, o Pensamento Computacional em contraste com o Design Thinking tem algumas limitações em relação a abordagem inicial do problema. Esta metodologia nem sempre o indivíduo é levado a procurar as causas e origens do problema. A busca por uma solução imediata pode caracterizar uma limitação. No Design Thinking já nas primeiras fases, tanto durante a imersão e a análise e interpretação levará Designers e Programadores a entenderem o problema antes mesmo de sua existência e a questionar indivíduos interessados na solução do problema quais são as possíveis soluções (Kumar, 2017).

Por isso, a união das duas metodologias gera uma abordagem interessante para solucionar problemas em contextos e segmentos de uma sociedade moderna, principalmente no que tange problemas relacionados à Educação e problemas enfrentados por professores e aluno em sala de aula. Vale ressaltar que metodologias e conceitos envolvendo diferentes abordagens e aplicações diversas são um conjunto

de procedimentos mas que a programação visual em blocos não se restringe somente a uma dada metodologia ou conceito, mas sim à criatividade humana, a motivação em solucionar problemas usando os conceitos tecnológicos existentes de cada era.

## **Ambiente do App Inventor**

O *App Inventor* desenvolvido pela Empresa Google e o Instituto de Tecnologia de *Massachussets* (MIT) possui um ambiente de desenvolvimento da programação feito por blocos. Segundo Gomes e Melo (2013) o ambiente deste aprendizado está estruturado no construcionismo que defende a ideia da criação e aplicações pelos próprios alunos, que a descoberta estimula a criatividade do aluno ao mesmo tempo que permite um aprendizado mais lúdico.

Dentro desta plataforma existem dois ambientes: *Designer* e *Blocks* que são destinados para a criação do aplicativo.

Em *Designer* (Figura 2) o usuário pode efetuar a criação da interface do usuário de forma visual, composta por cinco grupos de editores que são: *Palette*, *Viewer*, *Components*, *Media* e *Properties*.

*Palette* é composto por diversas ferramentas de interação com o usuário como: botões, caixa de texto, sensores entre outros, que podem ser adicionados à criação do aplicativo bastando arrastar estes elementos para a tela *Viwer*.

*Viwer* simula basicamente a tela de um celular onde são organizados todos os itens adicionados da aba *Palette*.

Nos *Components* estão localizados todos os elementos adicionados para facilitar a localização e edição do nome de cada componente.

Nos *Properties* são exibidas as propriedades de cada ferramenta adicionada, sendo possível alterar cores, tamanhos, nomes entre outros.

*Media* possibilita a adição de arquivos de mídia como vídeos e imagens para serem incluídos na aplicação.

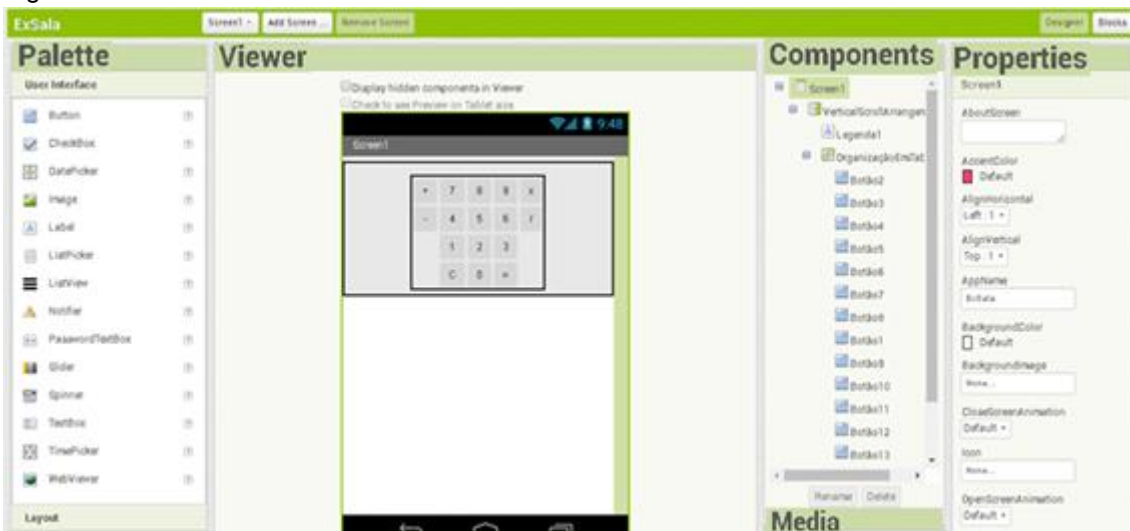
Figura 2:Tela *Viwer*



Fonte: Tela *Viewer* da plataforma do *App Inventor*

Em *Blocks* (Figura 3) são exibidos todos os comandos que possibilitam o controle de todas as ferramentas que foram incluídas na janela de *Designer*. Ao clicar em cada elemento dentro da aba *Bloks* são exibidas as possibilidades que podem ser eventos ou métodos, que ao selecionar basta arrastar para a tela *Viewer* e juntar os comandos de blocos.

Figura 3:Tela *Blocks*



Fonte: Tela *Blocks* da plataforma do *App Inventor*

No site do *App Inventor* é possível ter acesso a tutoriais, exemplos, métodos, entre outras informações que ajudam no aprendizado da utilização da plataforma. Para nosso seminário criamos um material de apoio, para inicializar a utilização do *App Inventor*. Para acessar basta utilizar o *QR Code* na figura 4.

Figura 4:QR Code Download do material



### **Aplicação de um Minicurso Utilizando o App Inventor**

Durante o seminário realizado no primeiro semestre de 2018 foi proposto um minicurso referente ao uso do *App Inventor*. A maior parte do público era constituída de alunos da Pós-graduação da Faculdade de Educação da UNICAMP e professores da rede pública e privada de ensino. Dentre os participantes, 50% dos alunos não sabiam da existência da plataforma e outros afirmaram que nunca haviam feito a programação.

O minicurso foi baseado no material que criamos (imagem 4), com o objetivo de introduzir o passo a passo de como utilizar o *App Inventor* e suas principais funções. Cada participante, utilizando um computador ou mesmo um celular, pode acessar o site da plataforma para acompanhar o seminário.

Primeiramente foi apresentado a Plataforma, como cadastrar, ter acesso à tela inicial, como iniciar um projeto, apresentação detalhada da tela *Viewer* e da tela *Bloks*. Durante a apresentação da tela *Bloks* foi exposto um exemplo na lousa de um fluxograma, explicando o funcionamento da lógica de programação.

Em seguida cada aluno pode criar uma calculadora para efetuar somas, subtrações, multiplicações e divisões seguindo as orientações de como efetuar o design do aplicativo na tela *Viewer* e a programação na tela *bloks*. Por fim os alunos foram guiados a compilar o aplicativo para baixar em seu celular.

Durante a execução do mini curso alguns alunos tiveram algumas dúvidas em como organizar as interfaces visuais na tela *Viewer* e também dúvidas na utilização dos blocos de programação. Mesmo com algumas dificuldades todos foram capazes

de ter um entendimento satisfatório sobre a plataforma, tanto na tela de design como na tela dos blocos de programação e ter um primeiro encontro com a lógica de programação.

Recebemos alguns *feedbacks* positivos mostrando um grande interesse e entusiasmo dos participantes ao se envolver na criação de um aplicativo para seu celular, tendo curiosidade em continuar a explorar a plataforma e criar outros aplicativos.

## **Tecnologias Para Deficientes Visuais**

De acordo com os dados do IBGE (2003) os resultados do Censo 2000 mostram que, aproximadamente, 24,6 milhões de pessoas, ou 14,5% da população total, apresentam algum tipo de incapacidade ou deficiência. São pessoas com ao menos alguma dificuldade de enxergar, ouvir, locomover-se ou uma deficiência física ou mental.

Conforme Villela (2015) em meio os tipos de deficiência estudados, a visual é a mais representativa e abrange 3,6% dos brasileiros, sendo mais comum entre as pessoas com mais de 60 anos (11,5%). Nesse caso, a cegueira ou baixa visão inviabiliza que 16% dos deficientes visuais realizem suas atividades habituais como ir à escola, trabalhar, usar tecnologias e brincar.

Segundo a Organização Mundial da Saúde, as principais causas de cegueira no Brasil são catarata, glaucoma, retinopatia diabética, cegueira infantil e degeneração macular. Além disso, a OMS aponta que, se houvesse um número maior de ações efetivas de prevenção e/ou tratamento, 80% dos casos de cegueira poderiam ser evitados. Ainda segundo a OMS cerca de 40 milhões a 45 milhões de pessoas no mundo são cegas; os outros 135 milhões sofrem limitações severas de visão.

Glaucoma, retinopatia diabética, atrofia do nervo óptico, retinose pigmentar e degeneração macular relacionada à idade (DMRI) são as principais causas da cegueira na população adulta. Entre as crianças as principais causas são glaucoma congênito, retinopatia da prematuridade e toxoplasmose ocular congênita.



Em virtude do grande número de deficientes visuais existentes no mundo, diversas leis e decretos foram criados para garantir-lhes o direito à educação, socialização e inclusão.

A Lei de Diretrizes e Bases da Educação, nº 4.024/61 descreve em seu artigo 88 que “a educação de excepcionais deve, no que for possível, enquadrar-se no sistema geral de educação, a fim de integrá-los na comunidade”. O texto da lei mostra uma relativa obrigatoriedade da educação inclusiva e por isso, pouco se fez para que houvesse efetivamente o atendimento das pessoas com necessidades educativas especiais no sistema regular de ensino.

A Constituição Federal de 1988 representa um avanço para a educação inclusiva, pois o documento define no inciso III, do Artigo 208, que é dever do Estado "atendimento educacional especializado aos portadores de deficiência, preferencialmente na rede regular de ensino". Neste momento, o Estado não têm a inclusão de pessoas deficientes no ensino regular como uma opção e sim com um dever que precisa ser cumprido e obedecido, garantindo dessa forma o ensino obrigatório dos deficientes preferencialmente na rede pública de ensino.

Dessa forma, a Lei de Diretrizes e Bases da Educação Nacional 9394/96, reservou o capítulo V e os artigos 58, 59 e 60 para a Educação Especial, a fim de direcionar normativamente o papel da escola inclusiva. Após a LDB 9394/96 o Ministério da Educação começou a elaborar e distribuir materiais sobre educação inclusiva, a fim de dar suporte às instituições e professores nesse processo (BRASIL, 2000, 2005, 2007, 2010).

## **Tecnologias Assistivas e TIC's em Sala de Aula**

Tecnologia Assistiva - TA é um termo recente, empregado para identificar todo o conjunto de recursos e serviços que colaboram para proporcionar ou estender habilidades funcionais de pessoas com deficiência e conseqüentemente promover vida independente e inclusão. (BERSCH & TONOLLI, 2006).

A formulação do conceito de TA é resultado de muitos estudos e pesquisas, alguns dos colaboradores deste estudo, como Cook e Hussey (1995) afirmam que “a Tecnologia Assistiva é uma ampla gama de equipamentos, serviços, estratégias e práticas concebidas e aplicadas para minorar os problemas funcionais encontrados pelos indivíduos com deficiência”.

Conforme Bersch (2008) a Tecnologia Assistiva deve ser compreendida como um meio de auxílio que promoverá a expansão de uma habilidade funcional deficitária ou tornará possível a realização de atividades ou funções almeçadas que se encontram impedidas por situações de deficiência ou envelhecimento.

Tecnologias da informação e comunicação (TIC's) é uma expressão que se refere ao papel da comunicação na moderna tecnologia da informação. Abrangemos que as TIC's incidem de todos os meios técnicos usados para abordar a informação e auxiliar na comunicação, o que inclui o hardware de computadores, rede, telemóveis, bem como todo software necessário.

Segundo Pinheiro (2008) as TIC's aumentam substancialmente o mundo dos alunos com deficiência, ampliando seus relacionamentos interpessoais e até servindo de estímulo de vida, pois, como sabemos, alguns destes alunos são desmotivados frequentemente pela coletividade, que confunde deficiência com ineficiência.

O autor ainda afirma que com o nascimento de novas técnicas computacionais, muitos conteúdos na Internet e até mesmo espaços completos adotaram de maneira excludente a forma em detrimento do conteúdo, haja vista o surgimento de tecnologias proprietárias que segregam os usuários – tenham eles deficiência ou não – de diversas formas, seja pelo fato de não terem um sistema operacional específico, seja por não terem um determinado navegador ou até pela falta de determinados aplicativos associados (*plugins*).

A garantia de facilidade no acesso a todos os níveis de ensino é obrigação do Estado, como citado anteriormente, mas a maneira de dispor este acesso é que ainda devemos estudar e discutir de como deve ser feita. A prática pedagógica em educação inclusiva ainda na atualidade compõe um desafio para os educadores em todo o mundo, uma vez que os fatores que influenciam o ensino como: estrutura institucional, formação docente, apoio da família e sociedade, dentre outros, ainda precisa de um melhor desenvolvimento.

Segundo Luz e Souza (2012) em se abordando sobre trabalho pedagógico com alunos cegos, os problemas são ainda maiores, já que por conta dos fatores interventores, os mesmos diversas vezes têm suas oportunidades de aprendizagem reduzidas ou até mesmo descuidadas. Somam-se a isso, o preconceito suportado no cotidiano social e o tratamento pejorativo que agridem o processo de autoafirmação pessoal.

Levando em consideração os fatos acima, percebemos que o professor é uma das figuras mais importantes na construção do conhecimento e da vida educacional de uma pessoa com deficiência, uma vez que será o docente a ponte que ligará este aluno com o mundo educacional e fará as devidas observações e ligações com a família deste educando.

Tratando-se particularmente de informática e matemática, podemos citar vários programas que foram desenvolvidos para auxiliar o professor nesta tarefa trabalhosa, porém realizadora, que é educar um deficiente. O primeiro programa que podemos citar é o: ALFAMATECA, um software que está sendo finalizado e foi desenvolvido na Universidade Estadual de Campinas que consiste na reprodução das atividades do livro didático do Programa Nacional do Livro Didático com o objetivo de auxiliar o professor no processo de alfabetização matemática com deficientes visuais.

Outro exemplo que podemos citar que constitui um software que auxilia o desenvolvimento cultural de alunos com deficiência auditiva é o Projeto Ludwig. Tal ferramenta também foi desenvolvida na Universidade Estadual de Campinas. Consiste em um aplicativo desenvolvido para dispositivos móveis, que possuem o sistema operacional iOS e funciona em conjunto com uma pulseira confortavelmente vestível. Ele é capaz de tornar a música acessível para portadores de deficiência auditiva e ainda permite que eles aprendam a tocar músicas inteiras.

Além de utilizar softwares que já existem, o professor pode aprofundar-se na elaboração de aplicativos que irão ajudá-los da maneira que ele necessitar. Nesta tarefa, ele poderá utilizar o *MIT App Inventor*, desta maneira o docente poderia criar programas que satisfaçam as suas necessidades em sala de aula de maneira específica. Como por exemplo, poderá construir uma calculadora simples, porém com síntese de voz, para isso, bastará utilizar apenas o comando “Texto para falar” presente no desenvolvedor de softwares.

Em um mundo cheio de tecnologias e ferramentas que possibilitam a inclusão, o docente tem todas as oportunidades para tornar suas aulas acessíveis, interessantes e lúdicas, tanto para os alunos sem deficiência quanto com deficiência, basta apenas a força de vontade e um pequeno esforço que resultará em uma grande diferença.

## **Considerações Finais**

Finalmente, a Programação visual em blocos promove esse ambiente de um pensar voltado nas preocupações atuais em torno do ensino mais adequado nesta era de evolução tecnológica. Mas respeitando valores como empatia, criatividade, colaboração, e principalmente centrado nos indivíduos em torno de um problema que o ser humano enfrenta em diferentes contextos. A formação de indivíduos considerando os conceitos discutidos aqui, formatada em uma estrutura de material pedagógico e didático com propósitos bem definidos, certamente será útil como ferramenta aplicada à Cultura Maker na Educação.

O App Inventor mostrou ter grandes benefícios para aprender programação com sua linguagem em blocos. Porém, existem algumas limitações na plataforma como algumas limitações e dificuldades na tela de designer não permitindo alocar figuras, botões ou outros de maneira simples, sendo obrigado utilizar a ferramenta de organização, que limita algumas ações. Também na a criação de aplicativos apenas para sistemas *android*, deixando sistemas como IOS de fora, com isso pessoas com celulares *Iphone* perdem motivação para criação de aplicativos utilizando a plataforma.

Mesmo com suas limitações, a plataforma oferece excelentes ferramentas, que possibilitam atividades makers dentro das escolas. Fazendo do aluno o criador de seus aplicativos.

## Referências

BERSCH, Rita.; TONOLLI, J. C. **Tecnologia Assistiva**. 2006. Disponível em: < <http://www.assistiva.com.br/> > Acesso em: 15 de maio 2018.

BERSCH, Rita. **Introdução à Tecnologia Assistiva**. Centro Especializado em Desenvolvimento Infantil. Porto Alegre. 2008.

BRASIL, Ministério da Educação. **Base Nacional Comum Curricular**. 2010. Disponível em: < <http://basenacionalcomum.mec.gov.br/download-da-bncc> > Acesso em: 25 maio 2018.

BROWN, T., 2008 – **How to deliver a great plan** – Design Thinking – Published on Harvard Business Review – June 2008 – page 60. Disponível em: < <https://hbr.org/2008/06/design-thinking> >. Acesso em: 15 de Maio de 2018.

CARDINS, Jitana Sara da Cunha. **A cultura do faça você mesmo como princípio da cibercultura e a crítica da inversão da lógica da indústria cultural**. Dissertação

(Mestrado) - UFPB - Programa de Pós-Graduação em Comunicação. João Pessoa, 2014. 96 f.

COOK, Albert M.; HUSSEY, S M. **Assistive Technologies: Principles and Practices**. St. Louis, Missouri. Mosby - Year Book, Inc. 1995.

CRISTÓVÃO, Henrique Monteiro. **Aprendizagem de algoritmos num contexto significativo e motivador: um Relato de Experiência**. In: Anais do XXVII Congresso da Sociedade Brasileira de Computação. 28, 2008, Belém do Pará, PA. P. 30-40.

DTEDUCADORES, 2018. **Design Thinking para Educadores**, Portal de Curso de Design Thinking gratuito, Disponível em: < <http://www.dtparaeducadores.org.br/site/sobre-o-material/> > Acesso em: 20 de maio de 2018.

FUTURA. **O que é Cultura Maker e o que ela tem a ver com a educação?** Disponível em: < <http://futura.org.br/trilhas-do-conhecimento/o-que-e-a-cultura-maker-e-o-que-ela-tem-a-ver-com-a-educacao/#> > Acesso em: 21 maio 2018.

GOMES, Tancicleide. C. S.; MELO, Jeane C. B. **App Inventor for Android: uma Nova Possibilidade para o Ensino de Lógica de Programação**. In: Anais do II Congresso Brasileiro de Informática na Educação. 2, 2013, Limeira, SP. P. 620-629.

HENNIG, V. F. A. **A inclusão de pessoas com deficiência visual e informática**. Trabalho de conclusão de curso. IFMG. Mato Grosso. 2009. IBGE. **Censo Demográfico 2000**. 2003. Disponível em: < <http://www.ibge.gov.br/home/presidencia/noticias/27062003censo,shtm> >. Acesso em 31/05/2018.

KUMAR, Vivek. 2017 – **Design Thinking vs. Computational Thinking in Education**. Disponível em: < <https://blog.quicksense.org/design-thinking-vs-computational-thinking-in-education-2dcf5b23aa12> > Acesso em: 25 maio de 2018.

LUZ, Cláudia; SOUZA, Ana. **Educação Inclusiva E Tecnologias Assistivas: Uma Análise Acerca Da Aprendizagem De Deficientes Visuais**. VI Colóquio Internacional de Educação e Contemporaneidade. São Cristóvão. 2012.

PADILHA, Márcia.; MARTINELLI, Adriana. **Cultura maker na escola: por que faz sentido**. 16 nov. 2015. Disponível em: < <http://www.arede.inf.br/cultura-maker-na-escola-por-que-faz-sentido/> > Acesso em: 26 maio 2018.

PINHEIRO, Flávia. **Tecnologias para a Inclusão de Alunos com Deficiência Visual no Ensino a Distância**. Fundação Centro de Ciências e Estudos a Distância. Rio de Janeiro. 2008.

POREDU, 2017 - Silva, L. A. et Al., 2017 - **Poredu: um ambiente de programação em blocos** - VI Congresso Brasileiro de Informática na Educação (CBIE 2017).

RESNICK, M., Ocko, S., and Papert, S. (1988). **Lego, logo, and design**. *Children's Environments Quarterly*, pages 14–18.

SEBRAE, 2018 - **Entenda o design thinking** - Portal Sebrae, Disponível em: < <http://www.sebrae.com.br/sites/PortalSebrae/artigos/entenda-o-design-thinking> > Acesso em: 20 maio 2018.

SIEVES, Cristiano. **3 exemplos de como incentivar o movimento maker na educação.** Disponível em: < <http://playtable.com.br/blog/3-exemplos-de-como-incentivar-o-movimento-maker-na-educacao/> > Acesso em: 18 maio 2018.

VILLELA, Flávio. **IBGE: 6,2% da população têm algum tipo de deficiência.** EBC Agência Brasil. 2015. Disponível em: < <http://agenciabrasil.ebc.com.br/geral/noticia/2015-08/ibge-62-da-populacao-tem-algum-tipo-de-deficiencia> > Acesso em 31 de maio 2018.

WOBER, David. **App Inventor and Real-World Motivation.** In Anais do 42º Simpósio Técnico da ACM sobre Educação em Informática. 42, 2011, Dallas, TX. P. 601-606.

WING, J. M., 2006. **Computational thinking.** *Communications of the ACM*, 49(3):33–35.

YouthMobile, MIT App Inventor, Kamriani F. & Roy K., 2016 - Portal Unesco.org [http://www.unesco.org/new/fileadmin/MULTIMEDIA/HQ/CI/CI/pdf/In\\_Focus/mit\\_app-inventor.pdf](http://www.unesco.org/new/fileadmin/MULTIMEDIA/HQ/CI/CI/pdf/In_Focus/mit_app-inventor.pdf) - Acesso em: 18 maio 2018.