

## Capítulo

# 2

## Computação criativa com Scratch, Mixly e Arduino: Prototipando com HackEduca Conecta

Elaine Silva Rocha Sobreira, Veronica Gomes dos Santos e Edson Sidnei Sobreira

### **Resumo**

*O Arduino é uma placa de prototipagem eletrônica programada em linguagem C++. Por ser de baixo custo, despertou o interesse de instituições educacionais. No entanto, sua linguagem de programação dificulta a utilização com crianças. Diante disso, a equipe do HackEduca desenvolveu o software HackEduca Conecta permitindo a programação do Arduino por meio da linguagem de programação por blocos - Scratch, Mixly, Aprendizagem de Máquina (“Machine Learning”) e conexão com dispositivos Android. Essa proposta, visa uma imersão em projetos desenvolvidos em instituições escolares, compartilhando experiências reais de projetos curriculares integrados a robótica educacional. Também apresenta um percurso prático formativo que compartilha estratégias exitosas para um trabalho significativo e contextualizado tanto com alunos formais, quanto na condução de formação docente e oficinas abertas.*

### **Abstract**

*Arduino is an electronic prototyping board programmed in C ++ language. Due to its low cost, it raised the interest of educational institutions. However, its programming language makes it difficult to use with children. Taking that in consideration, the HackEduca team developed the software HackEduca Conecta, allowing Arduino to be programed by blocks with Scratch, Mixly, Machine Learning and connection with Android devices. This proposal aims at an immersion in projects developed in school institutions, sharing real experiences of curricular projects integrated with educational robotics. It also presents a practical formative path that shares successful strategies for meaningful and contextualized work not only with students, but also with teachers and workshops.*

## 1. Introdução e justificativa

O trabalho com a robótica alternativa, com placas de prototipagem, sensores, atuadores e materiais reaproveitáveis vem crescendo dentro das salas de aulas, qualificando muitos conteúdos e diversificando as estratégias de aprendizagens. Porém, tal trabalho acaba sendo desenvolvido por poucos, pois encontra uma barreira natural na dificuldade de muitos professores em utilizar tais recursos, pela falta de conhecimento e experiência com as linguagens de programação.

Uma alternativa viável é programar as placas de prototipagem utilizando a linguagem de programação por blocos, por ser de fácil compreensão tanto para os alunos dos anos iniciais do ensino fundamental, quanto para os professores que não dominam a linguagem de códigos utilizada em cada um destes recursos, além disso, a linguagem de blocos também exclui possíveis erros de sintaxe, facilitando assim a sua execução.

Diante da necessidade de criar alternativas para viabilizar a integração do Scratch 2.0 com a placa Arduino (inicialmente), em 2011 criamos o software HackEduca Conecta, com o intuito de trabalhar com materiais de baixo custo nas aulas de robótica e com uma linguagem de programação já conhecida e utilizada pelos alunos. Outro diferencial é a sua facilidade de instalação com arquivo único, pois engloba todos os programas necessários que permitem a sua funcionalidade e execução.

Por meio de projetos desenvolvidos em sala de aula, foi possível verificar a facilidade de se trabalhar com placas de prototipagem envolvendo conteúdos curriculares da educação básica, tornando a aprendizagem dos conteúdos mais prática e significativa para os alunos [Sobreira, Viveiro, d'Abreu, 2017; Sobreira, 2017; Sobreira, Viveiro, d'Abreu, 2018].

Diante de tal cenário, elegemos práticas de computação criativa, com linguagem de computação por blocos, em projetos que tenham interação física e virtual, favorecendo a aprendizagem de conteúdos curriculares. Sendo assim, a seleção de algumas experiências de uso do HackEduca Conecta em escolas públicas envolvendo projetos curriculares em um contexto de computação criativa, podem inspirar outras propostas educacionais, para isso, estruturamos um percurso prático formativo aos interessados em desenvolver novas propostas.

A proposta visa apresentar a interface “HackEduca Conecta”, um instalador de extensões e plugins que possibilita a integração de algumas placas de prototipagem como Arduino, ESP8266, micro:bit, Raspberry pi e dispositivos Android com o Scratch 2.0 e Scratch 3.0, deixando acessível à linguagem de programação e facilitando a utilização de recursos variados integrados às propostas curriculares. Seu potencial também encontra-se na possibilidade de ser utilizado diretamente no computador offline, sem necessidade de conexão com a internet, viabilizando o uso em locais em que não possui acesso à internet, tornando-o mais acessível.

Propomos a estruturação de um percurso prático formativo, com o intuito de propiciar a outros educadores a oportunidade de, conhecer o software HackEduca Conecta para Windows, aprendendo a instalar e conhecendo suas funcionalidades, e, inclusive, projetar a utilização como recurso formativo ou de planejamento docente, baseado na criação de desafios de programação e reflexão sobre os mesmos.

Apesar de reconhecer a abrangência de interesse na temática, este percurso prático formativo foi elaborado considerando principalmente o crescente interesse de perfis que

buscam ampliar seus conhecimentos na integração de tais recursos na educação, seja de caráter formativo ou como recurso de ensino e aprendizagem. Porém, o conteúdo disponibilizado atende a uma variedade de perfis, como professores de educação básica, estudantes, pesquisadores, curiosos e interessados em geral em programar a placa Arduino utilizando linguagem de programação por blocos.

Delineamos como objetivo principal a oportunidade de disseminar a programação de placas Arduino integrando-o em projetos de contextos escolares, por meio da programação por blocos, por ser mais acessível a crianças, adolescentes e demais pessoas que não são familiarizadas com a linguagem escrita de programação C++. E como objetivos específicos pretendemos:

- Contribuir para o desenvolvimento de projetos que envolvam propostas de computação criativa, utilizando softwares que facilitem a programação das placas de prototipagem, sem exigir conhecimentos avançados em programação.
- Vislumbrar possibilidades educacionais, contribuindo para aprendizagens mais significativas, autorais e mão na massa.

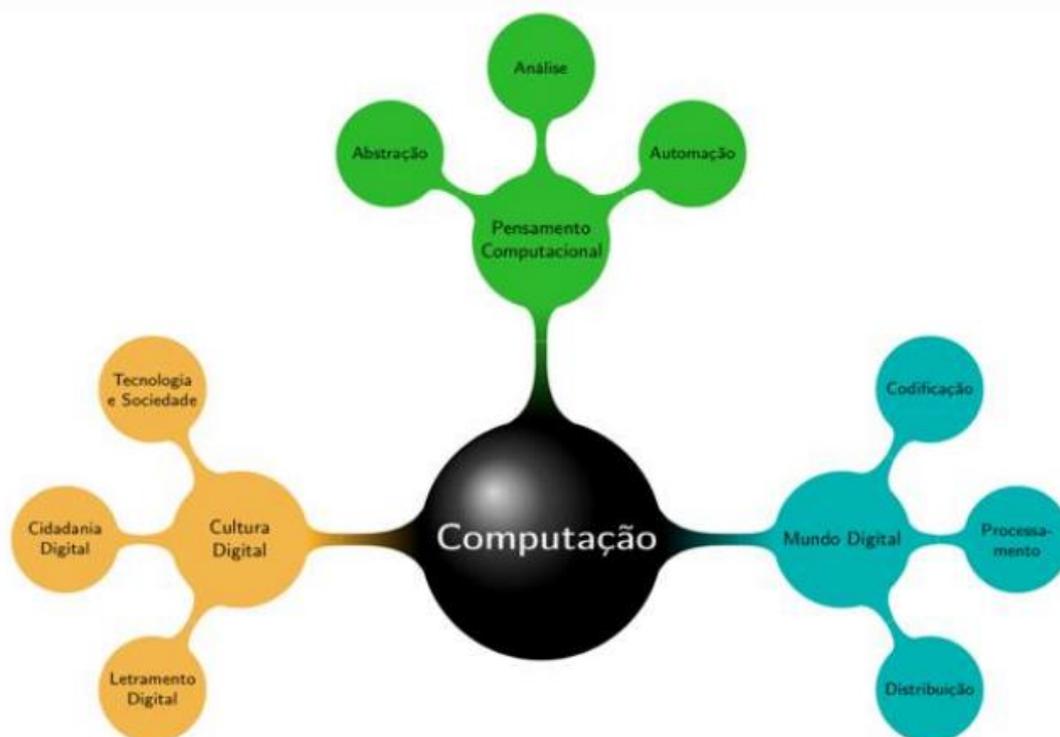
A partir deste material espera-se que os interessados estejam aptos a programarem uma diversidade de recursos utilizando a linguagem de programação por blocos e a conduzirem estratégias formativas ou planejamento para integração curricular. Para isso, estruturamos uma proposta de percurso prático formativa, contextualizada nos referenciais teóricos que vamos descrever neste trabalho, e organizamos esse texto que será utilizado como base na condução do mesmo, da seguinte forma: a seção 2 explicita a importância da integração das tecnologias na educação desenvolvendo habilidades relacionadas a cultura digital. A seção 3 aborda o conceito de computação criativa com base em referenciais teóricos que abrangem o construcionismo e a aprendizagem criativa. A seção 4 apresenta as placas de prototipagem como Arduino, micro:bit e Raspberry pi que favorecem o desenvolvimento de projetos com alunos da educação básica. A seção 5 apresenta o aplicativo HackEduca Conecta e a sua facilidade em programar estes dispositivos, apresentando exemplos viáveis, além das possibilidades da interface para a promoção da computação criativa. A seção 6 trará algumas experiências desenvolvidas em escolas de educação básica utilizando o HackEduca Conecta, apresentando projetos que demonstram a aplicação dos conceitos apresentados anteriormente. A seção 7 apresenta as escolhas metodológicas para a construção de um percurso prático formativo como possibilidade para disseminar o uso da programação do Arduino com a linguagem de blocos a partir da condução de aulas, planejamento de projetos e/ou criação de formações e, por fim, a seção 8 trará as considerações finais.

## **2. A importância da integração das tecnologias na educação para o desenvolvimento de habilidades relacionadas a cultura digital**

As tecnologias digitais estão cada vez mais sendo integradas na educação como recursos essenciais para o desenvolvimento de aprendizagens e competências próprias para o convívio social, além de favorecer a aprendizagem de conteúdos curriculares. Importantes documentos destacam a integração das tecnologias na educação básica, como o Itinerário

Formativo de Computação (SBC, s.d.) e as Diretrizes para ensino de Computação na Educação Básica (SBC, 2017).

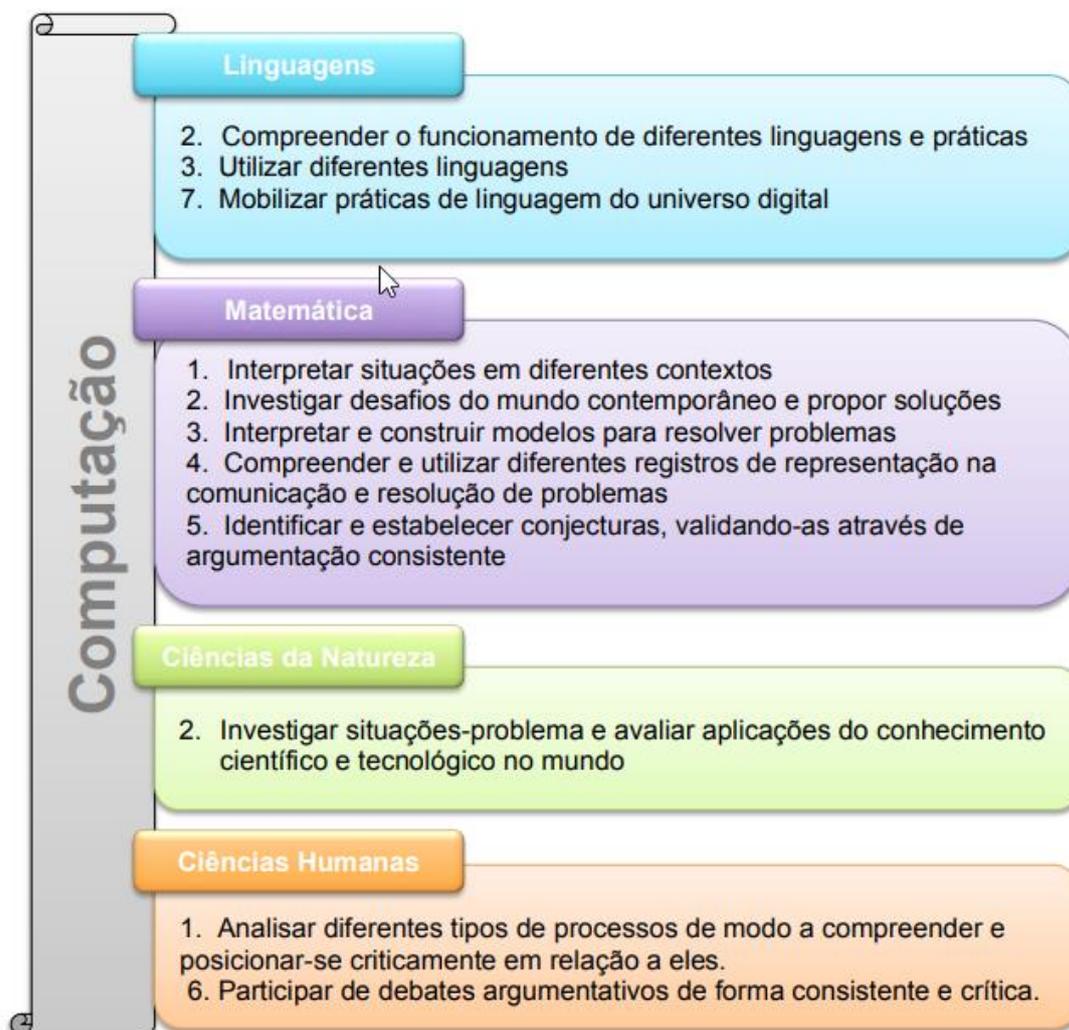
O Itinerário Formativo de Computação da SBC (s.d.) propõe três eixos para o ensino da computação, com ênfase nos eixos de Pensamento Computacional, Cultura Digital e Mundo Digital, conforme ilustra a (Fig.1) a seguir. Ele visa desenvolver as competências específicas e mobilizar competências de diferentes áreas.



**Figura 1: Três eixos para o ensino da computação**

**Fonte: Itinerário Formativo (SBC, s.d., p.1)**

O Itinerário formativo compreende a computação como uma área fundamental a qual “mobiliza todas as outras áreas do conhecimento, permitindo um gama de possibilidades de atividades integradoras na implementação das unidades deste itinerário” (SBC, s.d. p. 5). Busca desenvolver as seguintes competências das áreas da BNCC, conforme ilustra a (Fig.2):



**Figura 2: Competências da área da BNCC que podem ser desenvolvidas com o ensino da computação**

**Fonte: Itinerário Formativo (SBC, s.d., p.5)**

As Diretrizes para ensino de Computação na Educação Básica da SBC (2017), considera que o ensino de Computação desenvolve competências complementando a formação das outras áreas do conhecimento. Organiza essas competências em 5 específicas:

1. Interpretação e transformação do mundo (C1, C2, C6, C7, C10): Aplicar conhecimentos de Computação para compreender o mundo e ser um agente ativo e consciente de transformação do mundo digital, capaz de entender e analisar criticamente os impactos sociais, culturais, econômicos, legais e éticos destas transformações.
2. Aplicação de Computação em diversas áreas (C2, C3, C6, C7, C8, C10): Compreender a influência dos fundamentos da Computação nas diferentes áreas do conhecimento, incluindo o mundo artístico-cultural, sendo capaz de criar e utilizar ferramentas computacionais em diversos contextos, reconhecendo que a Computação contribui no desenvolvimento do raciocínio lógico, do pensamento computacional, do espírito de investigação, da criatividade, e da capacidade de produzir argumentação coerente.

3. Formulação, execução e análise do processo de resolução de problemas (C2, C4, C5, C6, C9, C10): Utilizar conceitos, técnicas e ferramentas computacionais para identificar e analisar problemas cotidianos, sociais e de todas áreas de conhecimento, modelá-los e resolvê-los, individual e/ou cooperativamente, usando representações e linguagens adequadas para descrever processos (algoritmos) e informação (dados), validando estratégias e resultados.
4. Desenvolvimento de projetos (C2, C5, C6, C7, C9, C10): Desenvolver e/ou discutir projetos de diversas naturezas envolvendo Computação, com base em princípios éticos, democráticos, sustentáveis e solidários, valorizando a diversidade de opiniões de indivíduos e de grupos sociais, sem preconceitos de qualquer natureza.
5. Computação é uma ciência (C1, C2, C4, C5): Compreender os fundamentos da Computação e reconhecê-la como uma ciência que contribui para explicar e transformar o mundo, solucionar problemas de diversas áreas do conhecimento e para alicerçar descobertas, com impactos no mundo cotidiano e do trabalho. (SBC, 2017, p. 6)

Na terceira versão da Base Nacional Comum Curricular - BNCC [BRASIL, 2017], as tecnologias aparecem como competências gerais, denominada como “Cultura Digital” e está integrada em todas as disciplinas da educação básica, reforçando a necessidade de educar os estudantes “para usos mais democráticos das tecnologias e para uma participação mais consciente na cultura digital” (BRASIL, 2017, p. 57).

A BNCC destaca dez habilidades essenciais, dentre elas, a competência 5 que refere-se a Cultura Digital e tem por objetivo que os alunos possam:

Compreender, utilizar e criar tecnologias digitais de informação e comunicação de forma crítica, significativa, reflexiva e ética nas diversas práticas sociais (incluindo as escolares) para se comunicar, acessar e disseminar informações, produzir conhecimentos, resolver problemas e exercer protagonismo e autoria na vida pessoal e coletiva. (Brasil, 2017, p. 7)

A inserção das tecnologias na educação favorece o desenvolvimento de habilidades para um uso crítico, autônomo, no qual Buckingham (2010), ressalta a necessidade de se desenvolver um letramento midiático que estimule “uma compreensão mais sistemática de como funciona a mídia e daí promover formas mais reflexivas de usá-la”. Porém, mais que desenvolver a criticidade para o uso das mídias digitais, destacamos aqui também a necessidade de desenvolver competências para que os estudantes possam desenvolver conhecimentos relacionados à produção de tecnologias. É o que Passarelli denomina de “prosumers”, onde “os usuários da modernidade (...) são denominados prosumers (produtor + consumidor) com a consequente redefinição dos papéis desses atores em rede” (2012, p. 14).

Sobre este tema, Brennan e Resnick (2012, p. 10), relatam que “a maioria de nossas experiências com mídia interativa é como consumidor” (tradução nossa) e que o tempo que dedicamos explorando as tecnologias são “atividades importantes para aprender a usar a tecnologia, mas que não são suficientes para o desenvolvimento como pensador computacional”. Para eles, “um pensador computacional vê a computação como um meio e pensa: “Eu posso criar”. e “Eu posso expressar minhas idéias através deste novo meio”.” (BRENNAN e RESNICK. 2012, p. 10 - tradução nossa).

A este respeito, a BNCC destaca:

Em decorrência do avanço e da multiplicação das tecnologias de informação e comunicação e do crescente acesso a elas pela maior disponibilidade de computadores, telefones celulares, tablets e afins, os estudantes estão dinamicamente inseridos nessa

cultura, não somente como consumidores. Os jovens têm se engajado cada vez mais como protagonistas da cultura digital, envolvendo-se diretamente em novas formas de interação multimidiática e multimodal e de atuação social em rede, que se realizam de modo cada vez mais ágil. (Brasil, 2017, p. 57)

Para isso, faz-se necessário trabalhar com todas as dimensões e subdimensões propostas na competência 5 da Cultura Digital no decorrer da escolaridade, envolvendo os conceitos demonstrados no quadro 1.

**Quadro 1 - Dimensões e Subdimensões da Competência "Cultural Digital" proposta pela BNCC**

<b>Dimensão</b>	<b>Subdimensões</b>
<b>Computação e Programação</b>	Utilização de ferramentas digitais; Produção multimídia; Linguagens de programação
<b>Pensamento Computacional</b>	Domínio de algoritmos; Visualização e Análise de dados
<b>Cultura e Mundo Digital</b>	Mundo digital; Uso ético

**Fonte: Dimensões e Desenvolvimento das Competências Gerais da BNCC<sup>1</sup>**

Todas estas subdimensões trazem grandes desafios para os professores, por nem sempre estarem familiarizados com estes conceitos, principalmente os que envolvem linguagens de programação e pensamento computacional. Para isso, faz-se necessário um investimento em formação continuada de professores, de modo a desenvolver conhecimentos também relacionados ao uso de tecnologias.

De acordo com Mishra e Koehler (2006), é necessário investir em formação de professores onde estes possam desenvolver conhecimentos relacionados tanto aos conteúdos específicos do currículo, quanto aos relativos às tecnologias e às abordagens pedagógicas. Defendem que o conhecimento sobre conteúdo (C), pedagogia (P) e tecnologia (T) é central para o desenvolvimento de um bom ensino, os quais devem ser trabalhados de forma integrada, derivando assim o TPCK (*Technological Pedagogical Content Knowledge* - conhecimento tecnológico e pedagógico do conteúdo).

Para Mishra e Koehler (2006) trata-se de um conhecimento diferente do conhecimento de um especialista disciplinar ou tecnológico. O TPCK é a base de um bom ensino com tecnologia onde, através de seu uso é possível representar conceitos utilizando tecnologias, auxiliar os alunos a resolver desafios, construir novos conhecimentos ou fortalecer conhecimentos existentes.

Diante do exposto, ressaltamos a necessidade de se desenvolver propostas de formação continuada dos professores para que compreendam a importância de incluir um trabalho consolidado com as tecnologias na educação básica, com abordagens que favorecem o desenvolvimento da autoria, do protagonismo e da criatividade, envolvendo conceitos de computação criativa, como exposto a seguir.

---

<sup>1</sup> [http://movimentopelabase.org.br/wp-content/uploads/2018/03/BNCC\\_Competencias\\_Progressao.pdf](http://movimentopelabase.org.br/wp-content/uploads/2018/03/BNCC_Competencias_Progressao.pdf)

### 3. Computação criativa

Além de propor a resolução de problemas de maneira eficazes e desenvolver a aprendizagem de conceitos relacionados a programação, a computação criativa favorece o desenvolvimento de outras competências, conforme relatado na página do Exploratorium<sup>2</sup>, do Tinkering Studio, ao descrever que essas práticas podem favorecer o “desenvolvimento pessoal, socioemocional e o pensamento criativo” (JEAN, 2016).

Trata-se de desenvolver os conceitos da programação aliados a projetos que tenham um significado pessoal, promova engajamento e se relaciona com questões sociais. Tem como base os conceitos construcionistas onde Papert (1980) e Kafai (2006), afirmam que as pessoas aprendem melhor quando estão construindo coisas que têm significado pessoal e social. Desta forma, propomos trabalhar com a integração dos conceitos de computação na educação com propostas que envolvem o desenvolvimento de projetos criativos, com o intuito de tornar a computação mais envolvente e significativa para todos os nossos estudantes, dos diversos níveis de escolaridade.

De acordo com o Guia de Computação Criativa da Escola de Pós-Graduação em Educação de Harvard [ScratchEd, 2019], a computação criativa propõe uma relação entre a computação e os interesses pessoais de cada indivíduo, baseado na criatividade, imaginação e interesses pessoais. Trabalha a computação como oportunidade de criação, não apenas para consumo de tecnologias prontas, favorecendo assim, o desenvolvimento de práticas para a criação de meios computacionais interativos que podem ser utilizados no cotidiano. Define a computação criativa de acordo com três aspectos: *creativity*, *agency* e *computing*, conforme descrito a seguir:

- *Creative computing is about creativity* (a computação criativa está relacionada com a criatividade): “A computação criativa apoia o desenvolvimento de conexões pessoais com o computador, estimulando a criatividade, a imaginação e os interesses pessoais.” (ScratchEd, 2019, p. 5)
- *Creative computing is about agency* (a computação criativa está relacionada com engajamento): “A computação criativa enfatiza o conhecimento, as práticas e a formação que os jovens precisam ter para criar as mídias computacionais dinâmicas e interativas que fazem parte do seu dia a dia.” (ScratchEd, 2019, p. 5)
- *Creative computing is about computing* (a computação criativa está relacionada com a computação): “(...) favorece o pensamento computacional, formando indivíduos capazes de usar conceitos, práticas e perspectivas computacionais em todos os aspectos de suas vidas, em diferentes disciplinas e contextos.” (ScratchEd, 2019, p. 5)

Para Presicce (2017), trata-se de criar propostas que envolvam uma abordagem que valoriza o desenvolvimento de novas ideias, a criação de projetos pessoalmente significativos, envolvidos em um processo de colaboração e reflexão. Segundo este autor, o que está realmente relacionado é a maneira como a atividade é planejada, conduzida e finalizada. É a proposta e a abordagem envolvida que poderão fornecer condições para o estudante desenvolver o aprendizado, conforme descrito a seguir:

---

<sup>2</sup> Site Exploratorium:

[https://www.exploratorium.edu/?gclid=CjwKCAjwiOv7BRBREiwAXHbv3OO7Q2SUq6f2Ywafxh-RqmMCgXgkkGsesiIAeM9M\\_bSooQ6DI19yRoCdkQQA vD\\_BwE](https://www.exploratorium.edu/?gclid=CjwKCAjwiOv7BRBREiwAXHbv3OO7Q2SUq6f2Ywafxh-RqmMCgXgkkGsesiIAeM9M_bSooQ6DI19yRoCdkQQA vD_BwE)

Em todos os casos, disponibilizar kit de ferramentas por si só, não definem a maneira como as pessoas se envolvem com o material. É a maneira como a atividade é projetada - em termos de instruções, estrutura e facilitação de atividades que podem fornecer mais ou menos espaço para as pessoas explorarem a tecnologia ou para alavancar a computação enquanto exploram. Presicce (2017, p. 26, tradução nossa<sup>3</sup>).

Verificamos que a computação criativa vai além da formação de pessoas para atuar na área, ela busca utilizar a computação para ampliar as possibilidades criativas e contribuir com a formação e desenvolvimento de outras competências e habilidades inerentes ao processo propiciado pela mesma.

Pretendemos assim, compartilhar projetos nos quais os estudantes possam envolver-se ativamente e construir tecnologias utilizando recursos físicos e computacionais, de forma que a computação possa tornar-se mais significativa e envolvente, favorecendo a formação de “pensadores computacionais” (BRENNAN E RESNICK, 2012).

Para viabilizar o acesso aos recursos, propomos além do uso de materiais reaproveitáveis e de baixo custo, a utilização de algumas placas de prototipagem, conforme relatado no item 4.

#### **4. O uso de placas de prototipagem na educação básica**

O trabalho com robótica educacional muitas vezes envolve um investimento alto de recursos e dispositivos robóticos, inviabilizando a sua efetivação em muitas instituições. Por questões de acessibilidade econômica, dentre outras vantagens, as placas de prototipagem passaram a ser uma alternativa acessível para o desenvolvimento das aulas de robótica. Elas possibilitam o acesso a diversos sensores e atuadores por um preço mais acessível e permite o desenvolvimento de projetos variados. A seguir, destacamos alguns recursos que viabilizam o desenvolvimento de projetos em instituições públicas.

##### **4.1. Arduino**

O Arduino (Fig. 3) foi criado em 2005 por um grupo de cinco pesquisadores: Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis. O principal objetivo era criar um dispositivo que fosse barato, funcional e fácil de programar, sendo dessa forma acessível a estudantes e projetistas amadores e artistas. Desde o princípio foi adotado o conceito de hardware livre, o que significa que qualquer um pode montar, modificar, melhorar e personalizar o Arduino, partindo do mesmo hardware básico.

O uso do Arduino é muito simples e com a ampla produção de empresas chinesas ele se tornou uma opção barata e popular, dessa forma, ampliou-se o uso por diversas escolas, alunos e curiosos em prototipagem, robótica e programação.

---

<sup>3</sup> Texto original em inglês: “*In all cases, the affordances of the toolkit alone do not define the way in which people engage with it. It is how the activity is designed - in terms of prompts, activity structure, and facilitation that can provide more or less space for people to tinker with computation or to leverage computation while tinkering*”. (PRESICCE, 2017, p. 26).



**Figura 3: Arduino Uno**

Fonte da imagem: Wikimedia<sup>4</sup>

## 4.2. Micro:bit

O Micro:bit (Fig.4) é uma placa de desenvolvimento criada pela BBC com o objetivo de ensinar programação para crianças. Permite a integração do mundo físico com o virtual, facilitando o desenvolvimento de projetos criativos, uma vez que já possui vários sensores embutidos, como o acelerômetro, bússola, botões programáveis, matriz de Leds e conexões de entradas e saídas analógicas e digitais, além da possibilidade de comunicação sem fio por meio do bluetooth ou radiofrequência.

O micro:bit pode ser programado por meio do Java ou Python e ambas podem ser feitas também através de blocos o que torna mais acessível para crianças. O Scratch 3.0 possui extensão nativa para comunicação via bluetooth, com possibilidade de uso de alguns poucos sensores. Utilizando a extensão personalizada disponível no Hackeduca Conecta é possível utilizar mais sensores e também todos os pinos. Devido ao preço, o acesso ao micro:bit pelas escolas brasileiras é mais limitado que o Arduino.



**Figura 4: Micro:bit**

Fonte da imagem: Wikimedia<sup>5</sup>

## 4.3. Raspberry pi

Diferente do Arduino e do micro:bit, o Raspberry Pi (Fig. 5) é um minicomputador, semelhante a um computador doméstico (PC). Ele é compacto e é necessário somente um cartão miniSD para instalação do sistema operacional (normalmente uma versão específica do Linux - raspbian), um teclado, monitor e um mouse.

O dispositivo foi criado no Reino Unido pela Fundação Raspberry Pi, uma organização sem fins lucrativos com foco na promoção e no ensino de ciência da

---

<sup>4</sup> <https://commons.wikimedia.org/wiki/File:Arduino-uno-perspective-transparent.png>

<sup>5</sup> [https://commons.wikimedia.org/wiki/File:BBC\\_Microbit.jpg](https://commons.wikimedia.org/wiki/File:BBC_Microbit.jpg)

computação básica para jovens em escolas e universidades da Europa, com produtos de preço acessível para os britânicos. Como computador é possível instalar softwares desenvolvidos para Linux, como a própria IDE do Arduino e a IDE do micro:bit. É possível programar o Raspberry Pi e usá-lo para programar um Arduino e um micro:bit.



**Figura 5: Raspberry Pi 4 - Modelo B**

**Fonte da imagem: Wikimedia<sup>6</sup>**

Essas três placas, associadas a outros recursos como sensores e atuadores são alternativas mais acessíveis para as instituições escolares se comparadas com os kits prontos vendidos exclusivamente para aulas de robóticas. No entanto, a linguagem de programação utilizada para programar esses recursos, também necessitam ser acessíveis para alunos da educação básica, principalmente para os alunos dos anos iniciais. Com este intuito, desenvolvemos o aplicativo Hackeduca Conecta.

## **5. HackEduca Conecta, facilitando a programação das placas de prototipagem**

A ideia inicial começou em 2011 com o desejo pessoal de integrar o Scratch 2.0 com o Arduino, como já existia a opção S4A para o Scratch 1.4, desejávamos utilizar a mesma versão do Scratch 2.0 para programar o Arduino nas nossas aulas de robótica.

Nesse momento conhecemos o trabalho do Alan Yorinks<sup>7</sup>, que havia desenvolvido uma extensão que já integrava o Scratch 2.0 com o Arduino utilizando a linguagem de programação Python 2.7 para esse propósito.

No mesmo ano conhecemos o trabalho do Kreg Hanning<sup>8</sup> que havia desenvolvido uma extensão para integração do Scratch 2.0 com dispositivos Android, neste caso, colaboramos para a tradução e correção de problemas relacionados ao uso no idioma português devido ao uso de acentos ortográficos, inexistente no idioma inglês.

Devido à complexidade de uso de múltiplas linguagens de programação, necessárias para comunicação, todo o processo se tornava muito trabalhoso para usuários que não possuem formação técnica e tampouco dispõe de tempo para a execução, pois alguns passos com ações múltiplas eram necessários para a execução no momento da comunicação entre o Scratch e o Arduino (o que em sala de aula pode demandar muito tempo).

Segue exemplo de algumas ações:

---

<sup>6</sup> [https://commons.wikimedia.org/wiki/File:Raspberry\\_Pi\\_4\\_Model\\_B\\_-\\_Side.jpg](https://commons.wikimedia.org/wiki/File:Raspberry_Pi_4_Model_B_-_Side.jpg)

<sup>7</sup> <https://github.com/MrYsLab>

<sup>8</sup> <https://github.com/khanning>

- Download do Scratch e Adobe Air
- Download e instalação da linguagem de programação Python
- Extensa sequência de passos para instalação
- Download do firmware FirmataPlus
- Download e instalação da IDE do Arduino
- Transferência dos firmware para o Arduino
- Abrir o terminal de comandos para acessar o Python
- Digitar o comando de inicialização (3 vezes - 1 vez para cada programa)

Pela necessidade de todos os passos descritos anteriormente, foi criado uma versão simples e rudimentar que consistia em um script programado com linguagem Batch (utilizada para automatização de tarefas - já integrada ao Windows), para executar automaticamente quase tudo que o usuário deveria fazer para instalação e uso dos programas.

Embora inicialmente ele fora concebido para uso particular, começamos a perceber que havia interesse por parte de outros professores, de outras escolas também, em utilizar o HackEduca Conecta. Para atender esta demanda, decidimos disponibilizar alguns tutoriais e manuais de instruções para auxiliar professores, mas após algum tempo, percebemos que havia dificuldade por uma grande parte dos professores, pois o processo semiautomático de instalação, dependia de um pouco de conhecimento da linguagem de programação Python.

Em meados de 2011 surgiu, então, algo parecido com a versão atual do HackEduca Conecta, substituindo todos esses passos descritos acima pelo download e instalação de um único arquivo que pode ser baixado gratuitamente no website do Hackeduca:

Na primeira versão, havíamos encapsulado os seguintes programas:

- Versão simplificada do Python
- Avrdude (permite a transferência de um arquivo compilado para o Arduino)
- Firmware para diversas opções de Arduino
- Scratch 2.0
- Adobe Air

Dessa forma, o professor deveria fazer um download, instalar e abrir o aplicativo por meio de um ícone em sua área de trabalho - ação idêntica a qualquer outra aplicação do computador.

Nos últimos anos passamos a traduzir, desenvolver nossas próprias extensões e colaborar com as novas criações de outros desenvolvedores, como a extensão do Alan Yorinks que atualmente permite a comunicação do Arduino com o Scratch 3.0.

A fundação Scratch incentiva a criação de extensões que possibilitam conexão entre dispositivos externos ao Scratch, como o Arduino, micro:bit, diversos robôs, e drones, por exemplo. Por fazer parte do grupo de apoiadores/financiadores, desde a versão 2.0 já existia extensões oficiais para alguns produtos da Lego. Hoje as extensões oficiais são para micro:bit, Lego e Makey Makey.

Extensões não-oficiais, como a do Arduino, por exemplo são feitas por desenvolvedores autônomos, normalmente utilizando linguagem de programação diferente da original do Scratch, como o Python, NodeJS, por exemplo. Além deste, também há a necessidade de um outro programa conhecido como “helper” para atuar como uma central de trocas de informações, pois o Scratch, desde a versão 2.0 não permite comunicação direta com sua interface, causando uma dificuldade adicional aos desenvolvedores, devido a criação de múltiplos servidores locais para troca de informações entre o Scratch e demais tipos de placas, conforme pode ser analisado pela Fig.6, considerando a comunicação entre Scratch 3.0 e Arduino.



**Figura 6: Passos de comunicação entre Scratch e Arduino**

Inicialmente o HackEduca Conecta, incorporava somente a possibilidade de comunicação entre Scratch 2.0 e Arduino via USB e também com dispositivos Android via rede Wifi local, porém, com o passar dos anos, o software foi sendo aperfeiçoado incorporando extensões desenvolvidas pela própria equipe do HackEduca bem como, em colaboração com outros desenvolvedores<sup>9</sup>. Há também extensões - “open source” - de outros desenvolvedores<sup>10</sup>, como por exemplo, a extensão DataView por João Adriano e Cássia Fernandez<sup>11</sup> e a extensão Scratch Control por Kreg Hanning<sup>12</sup>.

A versão de 4.0.0.4, lançada em Agosto de 2020, permite que o usuário possa fazer diversas atividades, como:

- Funcionalidades disponíveis nas versões para Windows, Linux e Raspberry Pi:
  - Transferência direta do Firmware para o Arduino
  - Transferência direta do Firmware para o micro:bit
  - Integrar Arduino ao Scratch 2.0
  - Integrar Scratch 2.0 ao micro:bit via porta USB (essa possibilidade permite o uso de comunicação via rádio)
  - Integrar Scratch 2.0 a dispositivos Android.
  - Programação do Arduino por meio de blocos com o software Mixly. (aplicativo Chinês, o qual incluímos novas extensões e traduzimos para o português)
- Funcionalidades disponíveis apenas na versão para Windows:
  - Integrar Scratch 3.0 ao Arduino
  - Versão Online e Offline do Tensorflow (algoritmo K-nn para comparação de imagens)
  - Disponibilidade de drivers para comunicação entre o Arduino (FTDI e CM340)

---

<sup>9</sup> s2aio, s2m, s3onegpio por Alan Yorinks - <https://github.com/MrYsLab>

<sup>10</sup>Scratch Microbit More por Koji Yokokawa - (<https://lab.yengawa.com/project/scratch-microbit-more>)

<sup>11</sup> <https://jaafreitas.github.io/scratch-dataviewer>

<sup>12</sup> <https://github.com/khanning>

O HackEduca Conecta pode ser baixado no site do HackEduca<sup>13</sup>, na área de download ou diretamente nas versões correspondentes, como a **versão 64bits**<sup>14</sup> e a **versão 32bits**<sup>15</sup>.

O HackEduca Conecta está sendo utilizado em 18 países e em 4 idiomas, conforme é possível conferir na imagem abaixo (Fig.7):



Figura 7: Informações numéricas do HackEduca Conecta

### 5.1. O que tem por dentro do HackEduca Conecta?

A *UI - User Interface* - Interface do Usuário (a parte que se vê e clica) do HackEduca Conecta foi criada em Visual Basic, para Windows e em Python, nas versões baseadas em sistemas Linux. O programa (parte do software que executa ações invisíveis ao usuário), por sua vez, foram utilizadas diversas linguagens como Python, Batch Script, Node Js, Java Script, Java, PHP e HTML.

Na maioria das vezes que se escolhe uma ação do HackEduca Conecta é efetuada uma ação em Python ou Batch Script.

Veja na Fig. 8 um exemplo do processo de passar o firmware para o Arduino (programa que possibilita a comunicação do Scratch 3.0 com o Arduino):

<sup>13</sup> <https://www.hackeduca.com.br>

<sup>14</sup> [https://www.hackeduca.com.br/download/hackeduca\\_conecta/](https://www.hackeduca.com.br/download/hackeduca_conecta/)

<sup>15</sup> [https://www.hackeduca.com.br/download/hackeduca\\_conecta32b/](https://www.hackeduca.com.br/download/hackeduca_conecta32b/)

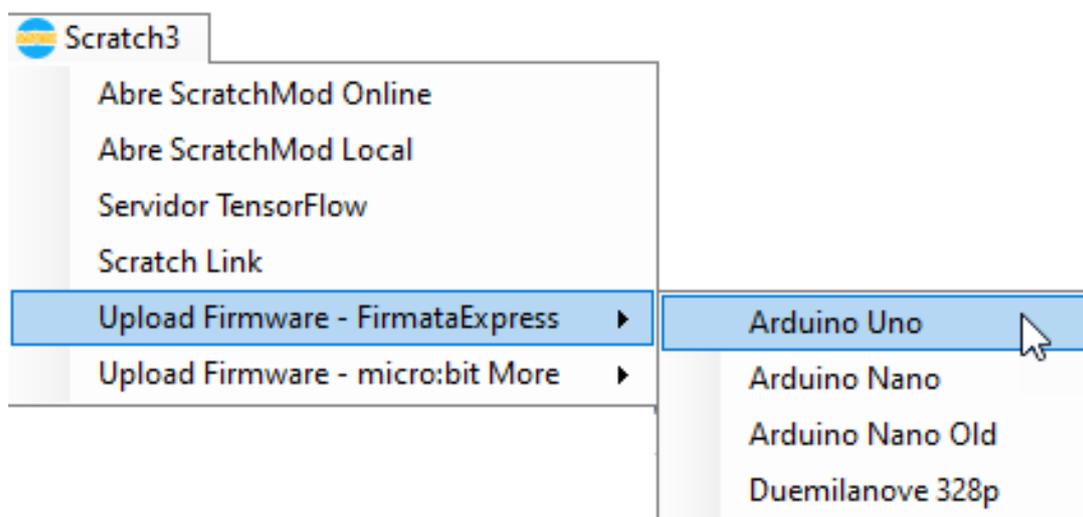


Figura 8: Carregando o Firmware para a placa Arduino

O Comando por trás do clique é:

```
Shell("cmd.exe /c" + "c: && cd\HackEducaConecta\BurnFirmataPlus\ && avrdude -  
Cavrdude.conf -v -patmega328p -carduino -P" + ComboBoxCOM.Text + " -b115200 -D -  
Uflash:w:FirmataExpress_Uno.hex:i && pause")
```

Sem esse comando o usuário precisaria abrir a interface IDE do Arduino, abrir o programa e transferir para a placa. Esse processo em uma sala de aula pode levar vários minutos para ser executado, porém, com o HackEduca Conecta pode ser realizado em segundos.

A comunicação entre Arduino e Scratch é feita através da linguagem Python, pois o Scratch 3.0 é programado em Java Script e o Arduino, por sua vez, é programado em C++ - não há uma maneira direta de comunicação entre ambas as linguagens.

Todo o processo da comunicação entre Scratch 3.0 e o Arduino ocorre por meio da criação de três servidores (ws\_gateway, backplane e Arduino\_gateway) locais programados em Python.

O Scratch envia uma mensagem para o servidor 1 (ws\_gateway), que por sua vez envia para um servidor comum (backplane) que traduz a solicitação e envia para o servidor 3 (Arduino-gateway) o qual finalmente envia ao Arduino, conforme pode ser verificado na Fig 6.

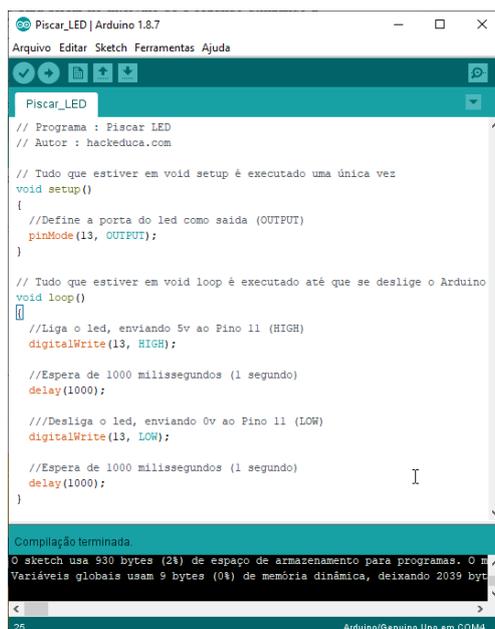
O mesmo processo ocorre de maneira similar mudando somente os dois últimos servidores, no caso de uma comunicação com ESP8266, Raspberry pi e Picoboard. Veja na Fig. 9 o exemplo da comunicação com o Raspberry Pi:



Figura 9: Passos de comunicação entre Scratch e Raspberry Pi

## 5.2. Programando sensores e atuadores do Arduino com o HackEduca Conecta

A facilidade proporcionada pela programação por blocos permite a criação de projetos por crianças desde o ensino fundamental I. Verifique a diferença de códigos de programação ao programar um Led para piscar no intervalo de um segundo utilizando a IDE do Arduino com linguagem C++ (Fig. 10) e utilizando o Scratch 3.0 na extensão OneGpio Arduino que está disponível no HackEduca Conecta (Fig. 11):



```
Piscar_LED | Arduino 1.8.7
Arquivo Editar Sketch Ferramentas Ajuda

Piscar_LED
// Programa : Piscar LED
// Autor : hackeduca.com

// Tudo que estiver em void setup é executado uma única vez
void setup()
{
  //Define a porta do led como saída (OUTPUT)
  pinMode(13, OUTPUT);
}

// Tudo que estiver em void loop é executado até que se desligue o Arduino
void loop()
{
  //Liga o led, enviando 5v ao Pino 11 (HIGH)
  digitalWrite(13, HIGH);

  //Espera de 1000 milissegundos (1 segundo)
  delay(1000);

  //Desliga o led, enviando 0v ao Pino 11 (LOW)
  digitalWrite(13, LOW);

  //Espera de 1000 milissegundos (1 segundo)
  delay(1000);
}

Compilação terminada
O sketch usa 930 bytes (2%) de espaço de armazenamento para programas. 0 bytes de memória dinâmica são reservados. Variáveis globais usam 9 bytes (0%) de memória dinâmica, deixando 2039 bytes disponíveis.
25 Arduino/Genuino Uno em COM4
```

Figura 10: Programação do Led na IDE do Arduino, linguagem C++

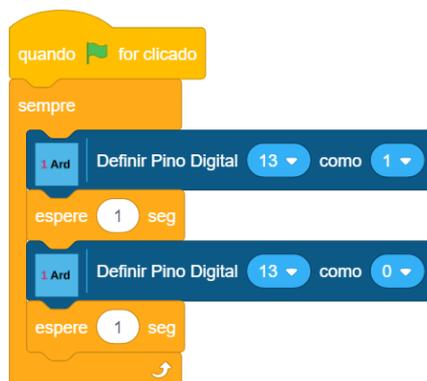


Figura 11: Programação do Led no Scratch 3.0 utilizando a extensão OneGpioArduino com HackEduca Conecta

Ao utilizar o Scratch para programar o Arduino ganhamos mais uma funcionalidade que propicia ricos projetos de programação criativa, pois permite a integração de projetos entre o mundo físico e o virtual, com iterações na tela e em dispositivos robóticos simultaneamente. Descrevemos aqui alguns exemplos de como essa integração é possível, como a proposta conduzida na oficina “Monte seu monstrinho” (Fig. 12 e 13). Os participantes foram desafiados a criarem monstrinhos, que realizasse alguma interação com o público a partir dos sensores disponíveis.

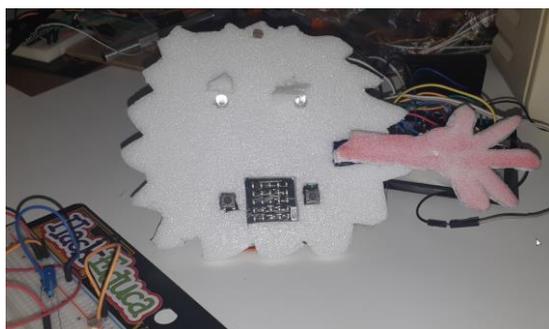


Figura 12: Exemplo de uso em Oficina “Monte o seu monstrinho”

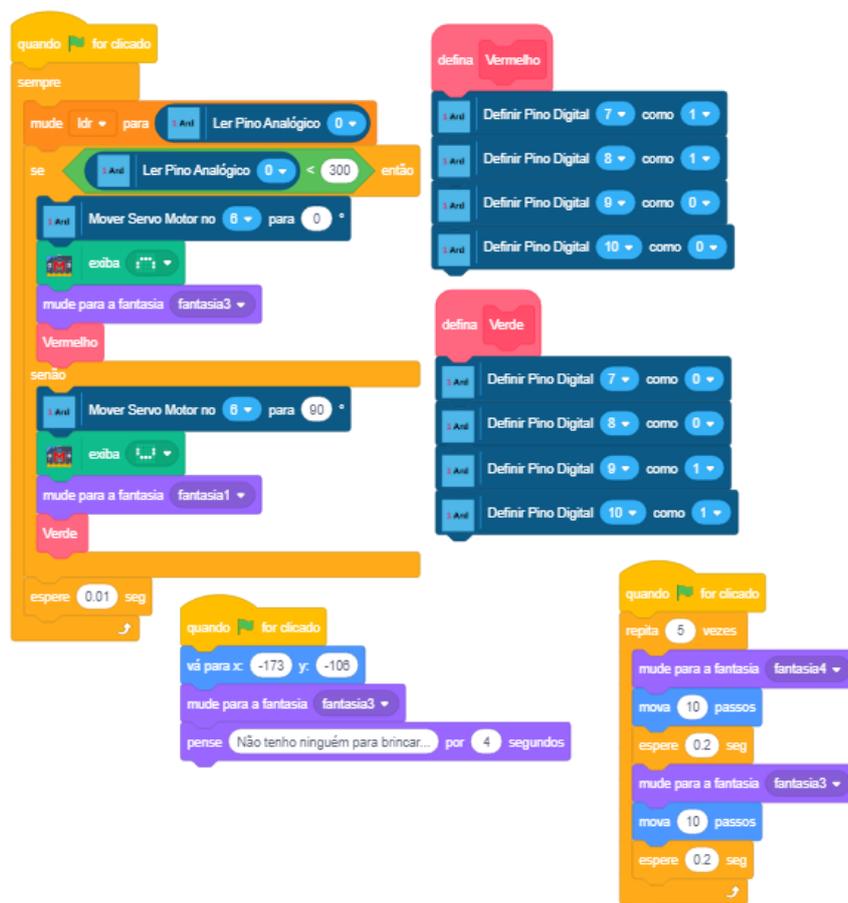


Figura 13: Exemplo do código utilizado na Oficina “Monte o seu monstrinho”

### 5.3. Mixly

A integração do Scratch com o Arduino é muito versátil, porém, há uma limitação que se torna crucial conforme se avança no entendimento de linguagens e lógica de programação, programar algo que não precise estar conectado com o computador o tempo todo.

Nessa fase, em que o aluno ainda não está apto para entender uma linguagem de programação escrita, como C++, Python, Java, por exemplo, o Mixly é um excelente caminho para essa migração.

Ele possibilita programar o Arduino, por meio de blocos, assim como o Scratch. O diferencial está no fato que, após a programação, é possível transferir para o Arduino o código criado, permitindo desconectar o Arduino do computador.

Veja um exemplo, no Mixly (Fig.14), equivalente ao que já vimos em (Fig. 10) e (Fig. 11):

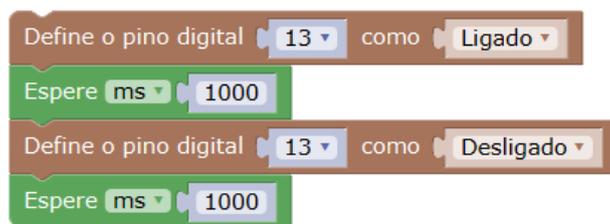


Figura 14: Exemplo do código similar a Figura 10 e Figura 11

No Mixly é também possível ver a programação escrita em C++ ao mesmo tempo que o que foi programado em blocos (Fig. 15):



Figura 15: Exemplo do código similar a Figura 10 e Figura 11, com a versão escrita em C++

Veja um exemplo para um termômetro, a programação na Fig. 16, a montagem na Fig. 17 e a execução final na Fig. 18:

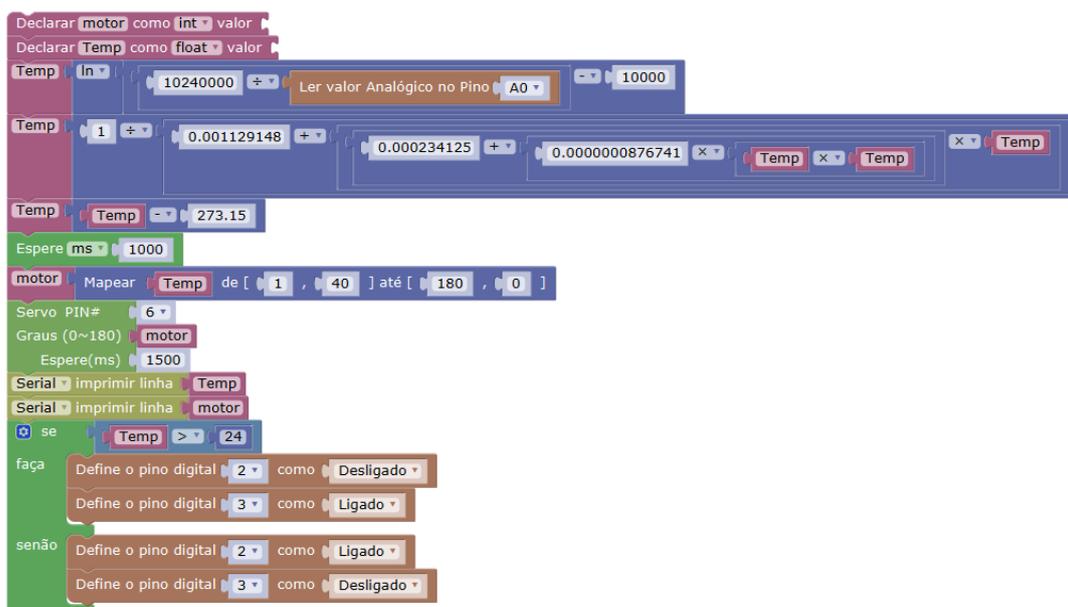


Figura 16: Exemplo do código usado na oficina “Monte um termômetro Maker”

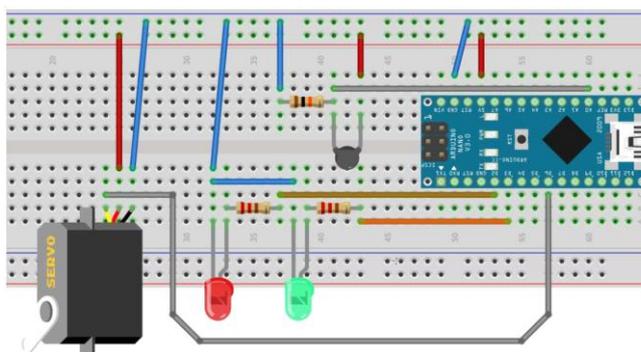


Figura 17: Exemplo do esquema de ligação usado na oficina “Monte um termômetro Maker”

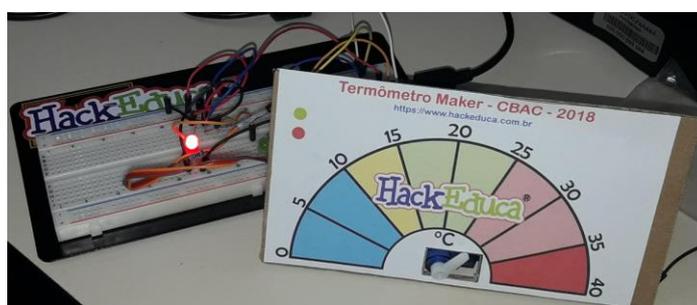


Figura 18: Protótipo de uma das possibilidades utilizada na na oficina “Monte um termômetro Maker”

#### 5.4. Micro:bit

No HackEduca Conecta há duas extensões para comunicação com Scratch, uma para a versão 2.0 e outra para a versão 3.0. Descrevemos abaixo a diferença entre elas.

##### 5.4.1. Extensão micro:bit para Scratch 2.0

Essa extensão surgiu da modificação da extensão criada por Alan Yorinks. Incluímos o sensor de temperatura e a possibilidade de envio e recebimento de mensagens via rádio com oitenta e três canais. Com o uso do rádio é possível enviar mensagens de uma placa para outra - opção muito versátil para os computadores que não possuem bluetooth 4.0.

Veja na Fig. 19 um exemplo de uso dessa extensão com um jogo de forca, onde um micro:bit envia mensagens para outra placa na sala de aula.

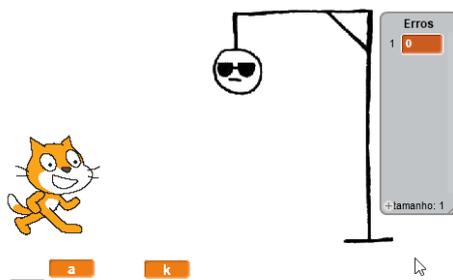


Figura 19: Exemplo de uso da Extensão para micro:bit e Scratch 2.0

### 5.4.2. Extensão micro:bit para Scratch 3.0

A extensão para o Scratch 3.0 foi desenvolvida por Koji Yokokawa<sup>16</sup> e o HackEduca foi responsável pela tradução para o português.

Essa extensão, diferente do padrão, desenvolvida pela equipe do Scratch, possibilita acesso a todos os pinos do micro:bit e não somente a 3 pinos, além de permitir controlar servo motores, pinos PWM, sensor de luz, entre outros.



Figura 20: Exemplo de uso da Extensão Microbit More incluída no HackEduca Conecta

### 5.5. Raspberry Pi

Por ser uma mescla de computador e um microcontrolador, é possível instalar o HackEduca Conecta e controlar um Arduino e um Micro:bit, da mesma forma que é possível fazer com um computador Windows, porém, aqui é possível também controlar o Raspberry Pi por meio da extensão criada por Alan Yorinks<sup>17</sup>. Nesse caso é possível instalar o S3OneGpio no Raspberry Pi e controlá-lo via remotamente, via Wifi, com o seguinte bloco de comando:



Figura 21: Exemplo de uso da Extensão OneGpioRpi para uso do Raspberry Pi, controlando LED e ServoMotor

<sup>16</sup> <http://www.yengawa.com/>

<sup>17</sup> <https://github.com/MrYsLab>

## 5.6. Aprendizagem de Máquina (“Machine Learning”)

A extensão TensorFlow K-nn foi criada pelo HackEduca e utiliza a biblioteca de código aberto para aprendizagem de máquina, TensorFlow, criada pelo Google. O algoritmo que utilizamos para essa extensão é o K-nn (vizinhos mais próximos) que compara imagens previamente armazenadas (ensinadas/treinadas) por meio da webcam no Scratch 3.0. A Extensão permite que uma vez que o algoritmo esteja treinado ele pode ser guardado para reutilização.



Figura 22: Exemplo de botões adicionais na extensão TensorFlow K-nn

Ela pode ser utilizada em programações para reconhecimento de imagens, em conjunto com Arduino, por exemplo, para criar uma lixeira que pré seleciona o tipo de material para ser reciclado ou até mesmo em um jogo com uso da webcam, como demonstrado a seguir, no jogo Pedra, Papel e Tesoura, (Fig. 23):

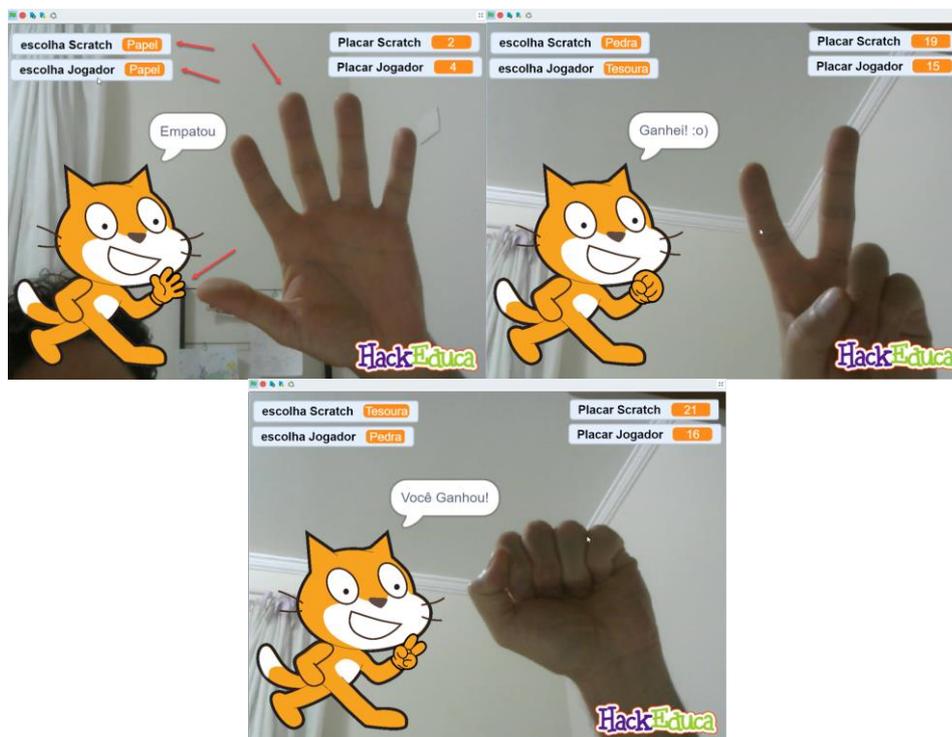


Figura 23: Exemplo de uso da extensão TensorFlow K-nn com o jogo Pedra, Papel e Tesoura

## 5.7. Possibilidades de computação criativa com o HackEduca Conecta

Um das preocupações que permeia a atualização da interface do HackEduca Conecta, é a abertura para o trabalho a partir dos princípios fundamentais da Aprendizagem Criativa, estruturante da computação criativa. A integração com ambientes de programação como Scratch, Micro:Bit, e outros, possibilita que as estratégias de produção, utilizando a interface, permita a exploração lúdica e a livre criação, dentro de propostas engajantes e inspiradoras.

Mas para tanto, é essencial que a intencionalidade que conduz as propostas planejadas considere a liberdade criativa, a autonomia e o protagonismo dos participantes. Resnick (2020) aponta que estratégias que consideram e privilegiam os 4 P da Aprendizagem Criativa, tem um grande potencial de impacto na aprendizagem e de envolvimento dos alunos. Para o autor, propiciar condições para que os alunos possam trabalhar em projetos de interesse pessoal e relevância social, de forma imersiva, compartilhando interesses, dúvidas e resultados, em um ambiente aberto à criação e à liberdade, é oportunizar que os P de projeto, paixão, pares e pensar brincando fluam naturalmente.

Essas são condições que oportuniza a Espiral da Aprendizagem Criativa (Fig. 24) acontecer durante o processo criativo. Identificamos boas propostas na computação criativa quando percebemos o trânsito dos alunos nas etapas da espiral, de modo aleatório, sem uma ordem definida, mas significativa àquele processo. Nesse cenário onde estes dois princípios permeiam as propostas, o erro é considerado fundamental para a aprendizagem, assim como as produções ganham formas de acordo com o envolvimento dos alunos, muitas vezes distante do planejado, validando a ideia de que essa espiral é um ciclo sem fim.



**Figura 24: Espiral da Aprendizagem Criativa**

Mas como fica a estrutura de aulas e oficinas a partir dos princípios da Aprendizagem Criativa? Devemos desconsiderar os materiais de suporte a bem da liberdade criativa?

Não são estas estratégias que o trabalho propõe. Verificamos que Resnick e seus colaboradores (2009) ampliam as ideias de Papert sobre como as condições de trabalho possibilitam o piso baixo, teto alto e paredes largas para a criação de propostas que favorecem o processo criativo. Em relação a isso, Santos e Galembeck (2017) explicam:

Quando o que norteia a utilização de materiais integrados com conceitos ou conteúdos curriculares, de modo que a criatividade e o desenvolvimento garantam um caminho possível, não previsível e estimulante, ou seja, o que Resnick (2009) a partir das ideias iniciais de Papert chamou de low-floor/high-ceiling/wide-walls – piso baixo, acessível para iniciar/ teto alto, para não limitar/paredes largas, possibilitando diversos caminhos - o envolvimento dos alunos é natural e extrapola o simples brincar. (SANTOS; GALEMBECK, 2017, p.04)

Assim, tanto o percurso formativo proposto quanto as sugestões de estratégias compartilhadas, enxergam os materiais de suporte criados, como oportunidade em piso baixo, que permitem que todos os participantes tenham as mesmas condições de iniciar seus projetos e programação. Um exemplo, são as fichas disponibilizadas no site HackEduca sobre a programação de Scratch com Arduino<sup>18</sup>.

São materiais que preconiza o suporte e as oportunidades, mas não encaixam todas as criações no mesmo formato, tolhendo a criatividade dos participantes. É o respeito ao tempo de cada um, os caminhos escolhidos, a liberdade de extrapolar as propostas iniciais, como propõe a ideia de teto alto (sem limites) e paredes largas (com diversos caminhos para criação). O exemplo de produções compartilhadas por meio de um livrinho<sup>19</sup> criado pelos alunos, demonstram tais oportunidades.

Tomando a Aprendizagem Criativa como abordagem central na condução dessas propostas, entendemos que as estratégias compartilhadas na estrutura da oficina e na reflexão sobre a mesma, respeita e considera os princípios almejados em propostas de computação criativa.

## **6. Experiências em escolas públicas utilizando o HackEduca Conecta**

No ensino fundamental I, exemplos de projetos com pesquisa e publicações podem ser verificadas nos relatos de Sobreira, Viveiro e D'Abreu (2017) os quais desenvolveram uma sequência didática envolvendo o conteúdo de energia (currículo de ciências) por meio de “montagem e programação de circuitos elétricos utilizando materiais de baixo custo ou reaproveitados, além de placas de prototipagem” (p. 456). Como resultados, analisaram que os estudantes puderam experimentar de forma prática a montagem de circuitos elétricos e ampliar seus conhecimentos em relação ao tema. Também conseguiram relacionar as novas aprendizagens às práticas cotidianas, desenvolvendo conceitos relacionados à ciência, tecnologia, sociedade e ambiente - CTSA.

Sales et. all. (2017) utilizaram a integração do Scratch e Arduino com o Hackeduca Conecta para o desenvolvimento de conteúdos do ensino de matemática e observaram que os alunos desenvolveram novas aprendizagens relacionadas ao conteúdo de matemática, além de interagir em um ambiente motivador. Conceitos geométricos específicos como, sólidos e vértices, além de conceitos gerais como dimensão e polaridade foram trabalhados quando os alunos foram desafiados a criar um cubo de LED e programar o seu funcionamento.

Sobreira, Viveiro e D'Abreu (2018) relatam que o projeto desenvolvido de criação de jogos digitais no Scratch com interação física utilizando Arduino, por meio da interface HackEduca Conecta revelou indícios de que:

A proposta favoreceu a aprendizagem dos conteúdos de Ciências, bem como suas interações com as tecnologias. Observamos essas aprendizagens a partir da reelaboração de conhecimentos prévios em direção aos conhecimentos científicos, aquisição de vocabulário e maior autonomia na tomada de decisões. Os estudantes envolveram-se em ações que favoreceram tanto o letramento científico quanto o digital, reconhecendo-se como produtores de jogos, ricos em contexto e propostas atrativas e significativas (p.1).

---

<sup>18</sup> <https://www.hackeduca.com.br/download/cartas-scratch-arduino/> - em atualização para versão 3.0 do Scratch

<sup>19</sup> <https://drive.google.com/file/d/115KDogwTWljbiCkgZBCtLyniPnczOO2T/view?usp=sharing>



Segundo Bogdan e Biklen (1994) as características da investigação qualitativa favorece o cenário educacional devido a sua abertura a subjetividade e caráter descritivo. Nesse sentido, tais contribuições amplia as possibilidades de olhar para cada participação e destaca avanços e integrações individuais, mesmo diante de uma proposta coletiva.

Apesar de não ser esta uma pesquisa que trabalhará com o levantamento de dados, os benefícios advindos das características de propostas investigativas quantitativas, contribuem também para a formação do público interessado na formação e integração curricular, preconizando que a diversidade de saberes e experiências é fator chave na cocriação de significados compartilhados e no trânsito de conhecimentos prévios. Assim, a subjetividade inerente aos estudos qualitativos assume um viés de valorização de cada percurso construído durante a oficina e seus possíveis impactos em contextos educacionais a partir da integração com projetos reais.

Enquanto metodologia de trabalho, a estrutura geral do percurso prático formativo prevê um desenvolvimento dos participantes a partir da criação e reflexão constante, em um formato organizado do seguinte modo:

- Contextualização e apresentação: etapas de orientação, apresentação da proposta e compartilhamento de informações necessárias;
- Criação: etapas de elaboração de projeto, protótipos a partir do fomento à criatividade e autoria;
- Exploração: etapas para explorar as possibilidades de programação a partir dos tutoriais compartilhados, fichas de suporte e livre exploração.
- Compartilhamento: etapas de socialização, discussão e reflexão sobre o percurso experienciado, os conceitos abordados e as criações resultantes do processo;
- Avaliação: etapa voltada ao reconhecimento das aprendizagens construídas, a partir de propostas colaborativas e reflexivas.

Com base na estrutura de trabalho, o percurso prático formativo foi planejado, respeitando cada etapa e visando alcançar os objetivos iniciais.

### **7.1. Estruturação de um percurso prático formativo para subsidiar a condução de estratégias formativas e adoção na prática do uso da programação do Arduino com a linguagem de blocos.**

A proposta do percurso prático formativo visa possibilitar a condução de estratégias formativas e/ou condução de propostas práticas integradas à educação formal e não formal. Sua estrutura parte da criação de projetos de prototipagem com interação física e virtual utilizando linguagens de programação por blocos (Scratch e Mixly) utilizando a interface do “Hackeduca Conecta”. Essa interface cria blocos adicionais no Scratch 3.0 permitindo a programação das placas Arduino e Microbit, tornando acessível a criação de protótipos com uso de sensores e atuadores e o Scratch possibilita a interação com o ambiente virtual.

A dificuldade em criar condições para que cada vez mais professores e alunos façam uso da integração física programável em situações de relevância e interesse, é afirmada pelo desconhecimento técnico e dificuldade de integração educacional. A

linguagem por blocos torna acessível a programação das placas de prototipagem por alunos desde o ensino fundamental, incluindo também professores das diversas áreas. Dessa forma, a criação e propagação de meios que facilitem tal objetivo, apresenta-se de grande relevância para este cenário, configurando a interface do “HackEduca Conecta” como uma das poucas criações que atendem ao almejado e, o percurso proposto como meio para tal.

Entendendo a necessidade de possibilitar estratégias passíveis de personalização e baseada em vivências reais aos participantes, dando condições para que os mesmos possam compreender o potencial de integração de tais propostas ao currículo, a experiência prática se torna de grande importância. Para tanto, a etapa para a criação “mão na massa” desde a elaboração do projeto até a construção de protótipos apresenta-se como uma das etapas fundamentais do percurso prático formativo. Saber manipular uma diversidade de materiais e enxergar potencialidades em recursos inusitados é parte de propostas voltadas para a computação criativa e de grande valor para o processo.

Com base no interesse em promover uma experiência de criação “mão na massa” com a possibilidade de programação por blocos propiciada pelo HackEduca Conecta, o percurso está estruturado da seguinte forma:

- **Apresentação da interface:** a apresentação da interface, evidenciando seus limites e potencialidades é uma das primeiras etapas indicadas. Nela, os interessados veem a facilidade de manuseio e a utilização acessível a todos. É interessante apresentar alguns projetos construídos a partir do uso da interface, seja de forma física ou por vídeos encontrados no próprio site do HackEduca. Indicamos a instalação de modo coletivo, possibilitando que participantes com experiências variadas possam participar de forma colaborativa. Materiais de suporte como o Manual do usuário - HackEduca Conecta<sup>20</sup> podem auxiliar neste processo.
- **Elaboração de projeto para criação:** Encontrar uma motivação ou interesse pessoal para desenvolver seus projetos é o passo inicial para que as produções ganhem maior engajamento dos envolvidos. Indicamos a elaboração de um desafio envolvendo conteúdos curriculares ou temáticas de relevância social, como proposta que norteará a elaboração de um projeto de criação para ser prototipado pelos participantes.
- **Construção do protótipo:** As propostas de construção, quando voltadas para criação de protótipos programáveis possibilita aos participantes enxergarem as aplicações práticas das placas de prototipagens, sensores e atuadores, além propiciar a integração de uma variedade de materiais. Visando a construção e funcionamento do protótipo, isso possibilita que os participantes selecionem as placas e demais recursos necessários para o funcionamento do projeto elaborado. Caso a proposta não seja presencial ou havendo limitação de recursos, poderá ser utilizado um simulador que contribuirá para a visualização da programação e seu funcionamento;
- **Programação:** A etapa de programação, após planejamento e prototipagem de projetos individuais, ganha um status de grande importância para o processo, pois é quando o funcionamento será testado e as possíveis adaptações serão realizadas.

---

<sup>20</sup> [https://www.hackeduca.com.br/download/manual\\_hackeduca-conecta/](https://www.hackeduca.com.br/download/manual_hackeduca-conecta/)

Esta etapa ganha maior liberdade quando são ofertados aos participantes a possibilidade de utilização de cards de programação Scratch/Arduino, tutoriais e orientação dos formadores. Dessa forma, a programação do protótipo permite que todos tenham condições de participação.

- **Socialização:** É de grande valor a organização de momentos de socialização do percurso por todos, seja de forma presencial ou virtual. Os envolvidos podem apresentar os seus protótipos e explicar a sua relação com a temática central. É uma etapa de cocriação, quando a participação dos demais poderá qualificar as produções e atribuir relações não consideradas pelos autores.
- **Discussão conceitual:** Após vivenciarem todo o processo mão na massa de planejamento, prototipagem, programação e socialização, a promoção de discussão dos conceitos que envolvem a computação criativa, a sua relação com os conteúdos curriculares e demais exemplos e possibilidades de integração é muito importante. Dessa forma, os participantes poderão vislumbrar novas alternativas para o desenvolvimento de projetos em seu contexto de trabalho.
- **Avaliação:** É indicado que os participantes tenham a oportunidade de refletir sobre o processo de criação, prototipação e programação do projeto, de modo que se enxerguem atuantes em todo o processo. Existem algumas formas construtivas de avaliação, como por pares, por meio de uma avaliação por rubricas e autoavaliação. Este movimento instiga a reflexão sobre as possibilidades e dificuldades da utilização de propostas semelhantes, utilizando a estrutura e a interface no seu fazer cotidiano e/ou educacional.
- **Reflexão final:** Sugerimos que a finalização do percurso prático formativo promova o levantamento de propostas para a integração da computação criativa no currículo escolar, considerando as possibilidades e dificuldades elencadas pelo grupo.

Espera-se que a estrutura do percurso prático formativo proposto, possibilite que os participantes, nos seus mais variados contextos, compreendam o funcionamento da interface “HackEduca Conecta” e sua potencialidade na integração de criações físicas programáveis por linguagens de programação por blocos, a partir de vivências práticas e reais. Espera-se ainda que sejam capazes de vislumbrar possibilidades educacionais com a computação criativa, contribuindo para aprendizagens mais significativas e autorais.

## 7.2. Estratégia de condução do percurso prático formativo

A partir de experiências similares, que contribuíram para a estruturação do percurso sugerido, entendemos que sua condução possa ser adaptada em ambientes presenciais e virtuais, com poucas modificações.

Em oportunidades presenciais, indicamos que os organizadores possam prover uma seleção de materiais que poderão ser somados aos que forem solicitados aos participantes, possibilitando a participação de um público misto de interessados. Considerando a condução virtual remota, sugerimos a escolha de simuladores e recursos semelhantes, além da abertura das propostas de criação, oportunizando a participação de todos. Momentos de interação síncronas, visando o compartilhamento, a socialização e a troca entre todos, demonstraram-se eficazes para a realização das atividades, como comprovado em outras experiências.

## 8. Considerações finais

O desenvolvimento de práticas de computação criativa nos contextos educacionais pode favorecer aprendizagens significativas, tanto relacionadas aos conceitos computacionais, quanto ao desenvolvimento de competências voltadas ao pensamento criativo, sócio emocional, relacionamentos interpessoais, entre outros.

Desenvolver práticas mão na massa contextualizadas com os conteúdos curriculares tornam a aprendizagem mais prazerosa, significativa, além de permitir o avanço nos conhecimentos curriculares, pelo fato de envolver os estudantes em contextos que possuem um significado real e social.

Para que a computação criativa possa realmente ser inserida na educação, além de oferecer materiais acessíveis, requer que os aplicativos e linguagens computacionais também sejam acessíveis, considerando que muitos educadores não dominam as linguagens próprias das placas de prototipação, o que muitas vezes inviabilizam o desenvolvimento de projeto.

A proposta do percurso prático formativo poderá favorecer a formação de educadores para o uso desses recursos. Entendemos que a estrutura indicada pode se tornar o ponto inicial de um programa formativo, que poderá ser adaptável e personalizável de acordo com os objetivos de dos formadores e do público fim.

## Referências

- Brasil, Ministério da Educação. (2017) Base Nacional Comum Curricular. 3ª versão. Disponível em: <[http://portal.mec.gov.br/index.php?option=com\\_docman&view=download&alias=78231-anexo-texto-bncc-reexportado-pdf-1&category\\_slug=%20dezembro-2017-pdf&Itemid=30192](http://portal.mec.gov.br/index.php?option=com_docman&view=download&alias=78231-anexo-texto-bncc-reexportado-pdf-1&category_slug=%20dezembro-2017-pdf&Itemid=30192)>. Acesso em jul. 2020.
- Brennan, K.; Resnick, M. (2012). Using artifact-based interviews to study the development of computational thinking in interactive media design. Paper presented at annual American Educational Research Association meeting, Vancouver, BC, Canada.
- Buckingham, D. (2010). Cultura Digital, Educação Midiática e o Lugar da Escolarização. Revista Educação & Realidade: revista da Faculdade de Educação da UFRGS, Porto Alegre, v. 35, n. 3, p. 37-58, set./dez., 2010. Disponível em: <[http://www.ufrgs.br/edu\\_realidade](http://www.ufrgs.br/edu_realidade)>. Acesso em: 12 jun. 2015.
- Jean (2016) The Tinkering Studio. Tinkering with Computational Thinking at ASTC <https://www.exploratorium.edu/tinkering/blog/2016/10/14/tinkering-with-computational-thinking-at-astc> Acesso em jul. 2020.
- Kafai, Y. B. (2006). Constructionism. In R. K. Sawyer (Ed.), The Cambridge handbook of the learning sciences. New York: Cambridge University Press.
- Mishra, P.; Koehler, M. J. (2006). Technological Pedagogical Content Knowledge: A Framework for Teacher Knowledge. Teachers College Record, 108(6), 1017–1054. doi: 10.1111/j.1467-9620.2006.00684.x.
- Papert, S. (1980). Mindstorms: Children, Computers, And Powerful Ideas. Basic Books.

- Passarelli, B. (2012). Mediação da informação no hibridismo contemporâneo: um breve estado da arte. *Ciência Da Informação*, 43(2). Disponível em <<http://revista.ibict.br/ciinf/article/view/1406/1584>>. Acesso em jun. 2020.
- Presicce, C. (2017). Explorations in computational tinkering. Tese Massachusetts Institute of Technology. Disponível em <[https://www.researchgate.net/publication/323722000\\_Explorations\\_in\\_computational\\_tinkering](https://www.researchgate.net/publication/323722000_Explorations_in_computational_tinkering)> Acesso em jul. 2020.
- Sales, S. B.; Silva, R. B.; Sobreira, E. S. R.; Nascimento, M. D. R. (2017). Utilizando Scratch e Arduino como recursos para o ensino da Matemática. Anais do XXIII Workshop de Informática na Escola (WIE 2017). VI Congresso Brasileiro de Informática na Educação (CBIE 2017). Disponível em <<https://br-ie.org/pub/index.php/wie/article/view/7272/5070>>. Acesso em jul.2020. doi: 10.5753/cbie.wie.2017.538.
- Santos, V.; Galembeck, E. (2017). Por uma ciência para o dia a dia possibilidades para aprendizagem criativa e significativa na educação básica. *Enseñanza de las ciencias: revista de investigación y experiencias didácticas*, ISSN 0212-4521, ISSN-e 2174-6486, N° Extra 0, p. 4035-4040; 2017
- SBC, Sociedade Brasileira de Computação. Itinerário Formativo de Computação. Disponível em <<http://www.sbc.org.br/documentos-da-sbc/send/203-educacao-basica/1216-itinerario-formativo-da-computacao>>. Acesso em jun. 2020.
- SBC, Sociedade Brasileira de Computação. (2017). Diretrizes para ensino de Computação na Educação Básica. Disponível em <<https://www.sbc.org.br/documentos-da-sbc/send/203-educacao-basica/1220-bncc-em-itinerario-informativo-computacao-2>>. Acesso em jun. 2020.
- ScratchEd. (2019). Computação Criativa. Escola de Pós-Graduação em Educação de Harvard. Traduzido por Rede Brasileira de Aprendizagem Criativa e Universidade Estadual de Feira de Santana (UEFS), Brasil. Disponível em <<https://learn.media.mit.edu/lcl/resources/readings/creative-computing-guide.pt.pdf>>. Acesso em out. 2020.
- Sobreira, E. S. R., Viveiro, A. A. y Viegas d'Abreu, J.V (2017). Do Paper Circuit à programação de Arduino com Scratch: uma sequência didática para aprendizagem do conteúdo de energia nos anos iniciais do Ensino Fundamental. Anais do XXIII Workshop de Informática na Escola (WIE 2017). VI Congresso Brasileiro de Informática na Educação. Disponível em <<https://www.br-ie.org/pub/index.php/wie/article/view/6852>>. Acesso em jul. 2020. doi: 10.5753/cbie.wie.2016.456.
- Sobreira, E. S. R. (2017) Tecnologias digitais no ensino de ciências para crianças: autoria e interações em uma proposta educativa explorando o tema energia. Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Física Gleb Wataghin, Campinas, SP. Disponível em: <<http://www.repositorio.unicamp.br/handle/REPOSIP/330423>>. Acesso em set. 2018.

Sobreira, E. S. R., Viveiro, A. A. y Viegas d'Abreu, J.V (2018). Aprendizagem criativa na construção de jogos digitais: uma proposta educativa no ensino de ciências para crianças. *Tecné, Episteme y Didaxis: TED*, 44, 71-88