

ENSINO PRÁTICO E INTEGRADO DE ELETRÔNICA E PROGRAMAÇÃO DE COMPUTADORES EMPREGANDO O MICROCONTROLADOR ARDUINO

PRACTICAL AND INTEGRATED LEARNING OF ELECTRONICS AND COMPUTER PROGRAMMING WITH THE ARDUINO MICROCONTROLLER

Carlos Yujiro Shigue

Universidade de São Paulo, Escola de Engenharia de Lorena
Lorena - SP

Alexandre de Moraes Ricardi

Universidade de São Paulo, Escola de Engenharia de Lorena
Lorena - SP

Eduarda Wiltiner Reis Santana

Universidade de São Paulo, Escola de Engenharia de Lorena
Lorena - SP

Danilo Bellintani

Universidade de São Paulo, Escola de Engenharia de Lorena
Lorena - SP

Vinicius de Souza Meirelles

Universidade de São Paulo, Escola de Engenharia de Lorena
Lorena – SP

Sandra Giacomini Schneider

Universidade de São Paulo, Escola de Engenharia de Lorena
Lorena - SP

RESUMO: O microcontrolador Arduino representa nos dias atuais uma pequena revolução tecnológica e social pela oportunidade que tem oferecido a “designers”, hobistas, “makers”, amadores, curiosos ou qualquer pessoa interessada em desenvolver projetos de automação, robótica e controle, com pequeno ou nenhum conhecimento de eletrônica e computação. A facilidade em programar e operar o microcontrolador desde a retirada da caixa e a colocação em funcionamento requer a autoaprendizagem a partir de livros didáticos, tutoriais e vídeos, que aliado ao seu baixo custo, pode ser usado para estimular alunos de Engenharia a desenvolverem projetos desde o seu ingresso no curso de graduação. Neste trabalho será apresentada a metodologia e a experiência em desenvolvimento desde 2014 no curso de Engenharia Física da Escola de Engenharia de Lorena de ensino prático e integrado de eletrônica e programação de computadores empregando o microcontrolador Arduino.

PALAVRAS-CHAVE: Arduino. Eletrônica. Programação. Computação Física. Educação STEAM.

ABSTRACT: The Arduino microcontroller today represents a small technological and social revolution for the opportunity it has offered designers, hobbyists, makers, curious or anyone interested in developing automation projects with little or no knowledge of electronics and computing. The ease in programming and

operating the microcontroller since the withdrawal of the box and commissioning requires self-learning from textbooks, tutorials and videos, which together with its low cost can be used to encourage engineering students to develop projects since your entry into the undergraduate course. This paper will present the methodology and experience in development since 2014 in the course of Physical Engineering of the Lorena School of Engineering of practical and integrated teaching of electronics and computer programming employing the Arduino microcontroller.

KEYWORDS: Arduino. Electronics. Programming. Physical Computing. STEAM Education.

1. INTRODUÇÃO

Em 1975, apareceu na capa da revista “Popular Electronics” um anúncio de venda de um kit de microcomputador Altair 8800 que marca o nascimento da computação pessoal. Na revista, o leitor encontrava instruções para montar um kit de US\$ 439 do microcomputador Altair 8800, que poderia ser adquirido da empresa MITS por reembolso postal (ALTAIR, 2019). Na época, Paul Allen, um estudante de graduação na Universidade Harvard assíduo leitor daquela revista, mostrou o exemplar de janeiro de 1975 ao seu colega de faculdade, Bill Gates. A partir da reportagem da revista, decidiram juntos abandonar o curso de graduação e trabalhar para a empresa MITS, onde criaram uma versão da linguagem BASIC para o microcomputador Altair. O resto é história conhecida: Paul Allen e Bill Gates posteriormente saíram da MITS e fundaram a Microsoft com a finalidade de criar e comercializar programas aplicativos para os computadores pessoais e que viria revolucionar todo o segmento da indústria de computação comercial.

Entretanto, o grande impulso para a popularização dos microcomputadores pessoais foi dado por dois jovens que também haviam abandonado a universidade, Steve Jobs e Stephen Wozniak, moradores de Cupertino no estado da Califórnia que em 1976, criaram um microcomputador ao qual batizaram “Apple” por causa do serviço que Jobs prestou durante a adolescência na colheita de maçãs no vizinho estado do Oregon.

A história da Apple começa com Wozniak, um engenheiro autodidata, que começou a participar das reuniões do Homebrew Computer Club, um clube de aficionados em eletrônica da região de São Francisco, Califórnia. Novos microcomputadores como o Altair 8800 o inspiraram a construir um

microcomputador completo com seu terminal de vídeo. Na época, as únicas CPUs de microcomputadores geralmente disponíveis eram a Intel 8080 (que equipava o Altair 8800) de US\$ 179 e o Motorola 6800 de US\$ 170. Embora Wozniak preferisse o Motorola 6800, o custo de ambos estava fora de seu alcance e por essa razão passou a projetar o computador no papel, esperando pelo dia em que pudesse comprar uma CPU. Quando a MOS Technology lançou seu chip de US\$ 20 em 1976, Wozniak começou a projetar um computador para rodar nessa CPU. O microprocessador 6502 foi projetado pelas mesmas pessoas que projetaram o 6800 e, como muitos no Vale do Silício, deixaram o seu empregador para criar suas próprias empresas. O primeiro “computador de papel” baseado no chip 6800 precisou apenas de pequenas alterações para rodar no novo chip. Wozniak concluiu a máquina e levou-a para as reuniões do Homebrew Computer Club para exibi-la. No encontro, Wozniak reencontrou seu velho amigo de adolescência, Steve Jobs, que estava interessado no potencial comercial das pequenas máquinas de hobby. A recepção mais que calorosa incentivou Jobs e Wozniak a montarem uma empresa para produzir computadores de uso pessoal. Essa empresa foi a origem da Apple Computers.

A partir das iniciativas de jovens hobistas, radioamadores e amantes de novas tecnologias, a indústria de computação sofreu enorme desenvolvimento e expansão em praticamente todos os lugares onde os jovens tivessem acesso a hardware de baixo custo e recursos de programação livre. Nas décadas seguintes, a indústria de computação alcançou o topo das economias mais desenvolvidas criando o que atualmente é denominada Tecnologias de Informação e Comunicação (TIC).

Paradoxalmente, ao desenvolver o segmento das TIC e expandir para outros setores da economia, houve gradual declínio das iniciativas empreendedoras dos amantes de eletrônica e computação e a subsequente migração ao desenvolvimento de softwares e aplicativos, relegando o desenvolvimento de hardware às corporações. O estouro da bolha da Internet e o advento do Arduino no final da década de 2000, novamente alterou o panorama de desenvolvimento de hardware.

1.1 Arduino

O Arduino é o nome de uma placa de circuito impresso constituída por um microcontrolador e por circuitos de controle e comunicação que possibilitam a criação de projetos de eletrônica “embarcada” de forma simples, barata e aberta. O Arduino representa nos dias atuais uma pequena revolução tecnológica e social pela oportunidade que tem oferecido a “designers”, “hobbistas”, “makers”, curiosos ou qualquer pessoa interessada em desenvolver projetos de automação, com pequeno ou nenhum conhecimento prévio de eletrônica e programação.

A origem do Arduino remonta aos anos 2003-2004 no Instituto de Design Interativo de Ivrea (IDII), Itália, a partir da dissertação de mestrado de Hernando Barragán sob a supervisão de Massimo Banzi e Casey Reas (BARRAGÁN, 2004), cujo objetivo era tornar mais fácil para os artistas e designers trabalharem com projetos eletrônicos e de computação, abstraindo os detalhes complicados da eletrônica para que eles pudessem se concentrar na criação do projeto. Um dos supervisores, Casey Reas, é conhecido por ter criado a plataforma de desenvolvimento de software livre Processing (REAS; FRY, 2010). Essa plataforma foi usada como base para a construção do software Wiring, que seria posteriormente transformado no IDE Arduino (ARDUINO, 2019). A Figura 1 apresenta a imagem da placa Arduino modelo Uno juntamente com a interface gráfica da IDE Arduino de programação do microcontrolador.

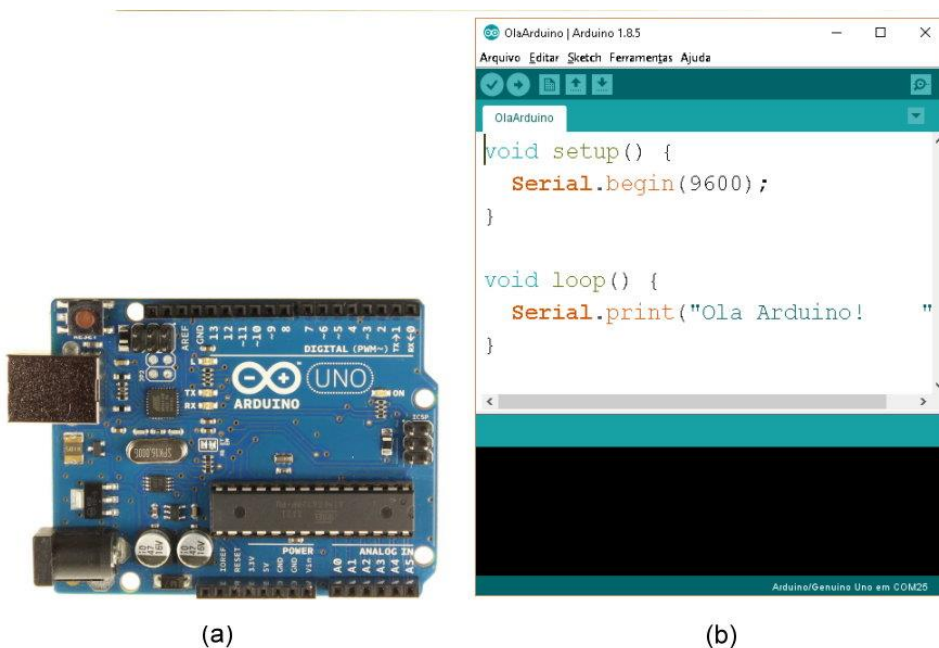


Fig. 1. (a) Placa Arduino modelo Uno; (b) IDE Arduino. Fonte: Próprio autor.

Em 2005, Banzi e colaboradores (ARDUINO, 2019) criaram a marca Arduino, por meio do qual os seus proprietários asseguravam para si os direitos autorais da placa e do software, de maneira que qualquer fabricante que quisesse produzir uma réplica da placa Arduino, deveria pagar uma licença de uso da marca e da propriedade intelectual do projeto. Outros fabricantes que não se dispusessem a pagar a licença de uso da marca, poderiam simplesmente fabricar e comercializar clones da placa com outra denominação que não fosse Arduino. Esse modelo de negócio, baseado em hardware e software livre, assegurou a rápida disseminação de projetos tendo como base a tecnologia aberta proporcionado pelo Arduino. Além disso, outros elementos responsáveis pela popularização do Arduino foram: disponibilização de ambiente de desenvolvimento integrado (IDE) disponível para diferentes sistemas operacionais, tais como Microsoft Windows, Mac OS e Linux; linguagem de programação simples, baseada na linguagem C, contando com recursos de programação para microcontroladores; *bootloader* disponível na IDE para carregamento simples de programas para a memória do microcontrolador; monitor e *plotter* serial para inspeção e envio de dados de e para o microcontrolador; aderência ao modelo de software e hardware livre; disponibilidade de referências *on-line* para comandos, bibliotecas, exemplos, tutoriais, fórum e descrição detalhada de projetos desenvolvidos com Arduino (ARDUINO, 2019).

O sucesso do Arduino deve muito à existência prévia dos ambientes Processing e Wiring. Esses projetos deram ao Arduino um dos seus pontos fortes essenciais: o ambiente de programação amigável ao usuário. Antes do Arduino, a codificação de um microcontrolador possuía uma curva de aprendizagem longa. Com o Arduino, mesmo aqueles sem experiência em eletrônica ganham acesso a um mundo de hardware anteriormente impenetrável. Os iniciantes não precisam aprender muito antes de começar a criar um protótipo que funcione. É um movimento poderoso em um momento em que alguns dos equipamentos eletrônicos mais populares funcionam como "caixas pretas" fechadas e protegidas por patentes. Para os criadores do Arduino, esse talvez seja o seu impacto mais importante: a democratização da engenharia (ARDUINO, 2019).

1.2 Processing

O IDE Processing foi criado por Casey Reas e Ben Fry com a finalidade de facilitar a programação de design gráfico interativo. Na época da criação do Processing, na década de 2000, era muito difícil para pessoas sem formação em computação escreverem esse tipo de software com as linguagens existentes (C, C++ e Java). Os criadores do Processing se inspiraram nas linguagens de programação Logo e BASIC utilizadas para o ensino de programação de crianças e adolescentes da década de 1970 (REAS; FRY, 2010).

O Processing nasceu na primavera de 2001 como uma sessão de *brainstorming* em uma folha de papel. O objetivo era criar uma maneira de esboçar (prototipar) o tipo de software de programação de gráficos, quase sempre em tela cheia e interativa. Casey e Reas (2010) estavam procurando uma maneira melhor de testar as ideias, em vez de apenas falar sobre elas ou gastar muito tempo programando-as em C++. Outro objetivo era fazer uma linguagem para ensinar aos alunos de Design e Arte como programar e ao mesmo tempo proporcionar aos alunos mais técnicos uma maneira mais fácil de trabalhar com gráficos. A combinação é um desvio positivo da forma como a programação é normalmente ensinada. No Processing enfatiza-se os gráficos e interação em vez de estruturas de dados e saída de console de texto, aproximando o usuário das interfaces humano-máquina (IHM) atuais. A Figura 2 mostra as interfaces da IDE Processing e da janela de saída gráfica Canvas.

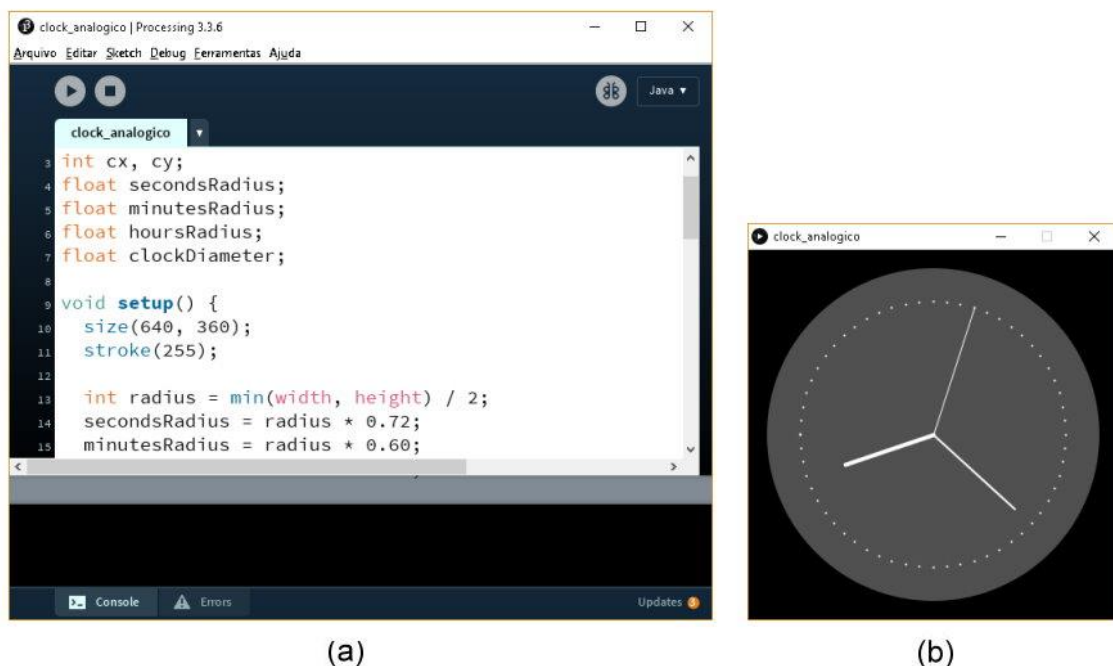


Fig. 2. Interfaces: (a) IDE Processing; (b) Canvas Processing (janela de saída gráfica). Fonte: Autoria própria.

O Processing possui diferentes modos de programação para possibilitar a implementação de programas (*sketchs*) em diferentes plataformas. O modo de programação padrão é o modo Java. Outros modos de programação, como o Android e o Python, são adicionados no menu do IDE. Como o Processing foi utilizado como base do IDE Wiring e, posteriormente, do Arduino, a Figura 3 mostra o relacionamento dele com estas e outras linguagens utilizadas na criação de projetos multiplataforma de uso geral, tornando-o uma versátil e poderosa ferramenta de desenvolvimento de software (REAS; FRY, 2014).

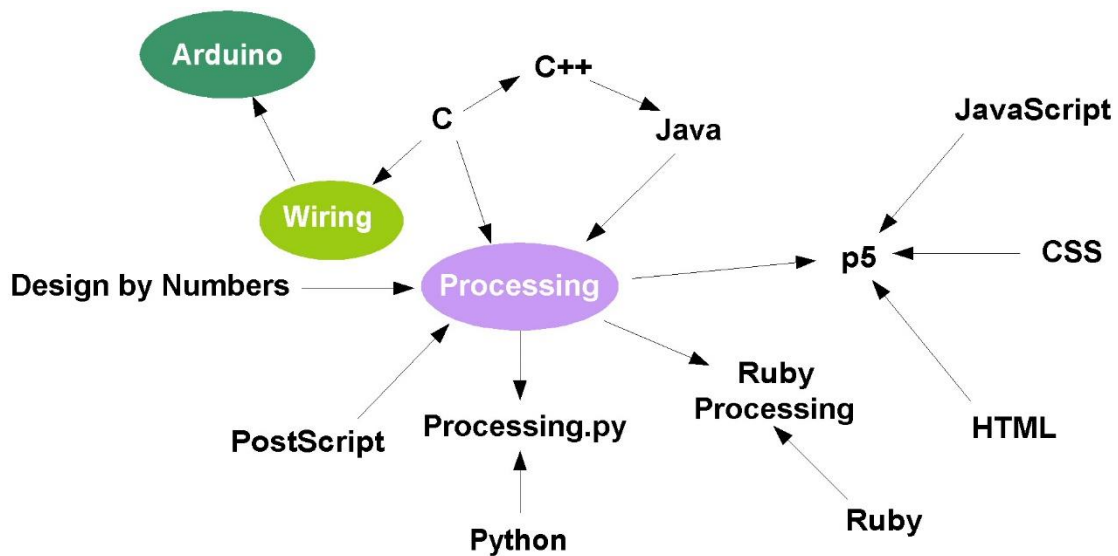


Fig. 3. Diagrama de relacionamento entre as linguagens Wiring, Arduino, Processing e outras linguagens de programação. Fonte: McCarthy; Reas; Fry (2016).

2. EDUCAÇÃO STEM VS. FILOSOFIA STEAM

Ao longo da segunda metade do século XX, as autoridades dos países desenvolvidos concentraram-se em melhorar o ensino de ciências, matemática e de tecnologia, com a intenção não apenas em promover a alfabetização nessas áreas, mas também para expandir a força de trabalho de cientistas e engenheiros. A importância atribuída ao papel dos programas educacionais na preparação de estudantes para participar da força de trabalho e competir na economia global do século XXI fez com que estudantes de ensino básico de dezenas de países participassem de testes de comparação internacional periódicas (TIMSS e PISA) do nível de conhecimento e habilidades cognitivas de ciências e matemática. A esses programas de educação científica pragmática dado o nome “Educação STEM”, sendo que a sigla em inglês STEM é o acrônimo

de Ciência (**S**cience), **T**ecnologia, **E**ngenharia e **M**atemática. Normalmente, inclui atividades educacionais em todos os níveis de ensino - da pré-escola ao pós-doutorado - em ambientes formais (geralmente salas de aula) e informais (MARTINEZ, 2017).

Educadores focados em melhorar a instrução em ciências e matemática empregaram várias abordagens para aplicar a educação STEM para estudantes desde a educação infantil até o ensino médio (K-12). Por exemplo, alguns professores integraram atividades baseadas em projetos que exigiam conhecimento e aplicação de habilidades em áreas específicas, como engenharia. Em alguns casos, atividades extracurriculares, incluindo competições de equipes nas quais os alunos trabalhavam juntos (por exemplo, para construir robôs), foram adicionadas ou expandidas. Os alunos também tiveram a oportunidade de passar tempo com os profissionais nos campos STEM, seja pesquisando como estudantes de iniciação científica ou trabalhando como estagiários (GUNN, 2017).

Em algumas escolas, no entanto, a educação STEM levou a uma expansão e segmentação de cursos de matemática e ciências, ensinados de forma tradicional. O resultado foi a saturação de alunos com aulas e material de estudo de conteúdo STEM tradicionais, levando à redução no desempenho escolar dos estudantes avaliados por testes padronizados, sem contabilizar o seu interesse ou motivação para o estudo. Para fazer frente à problemática da Educação STEM, educadores e designers propuseram uma filosofia educacional que integrasse as Artes na Educação STEM criando o chamado movimento STEAM, com a incorporação da letra A de Artes à sigla STEM (GUNN, 2017).

O movimento de educação STEAM defende o afastamento das áreas de conteúdo segmentado, enfatizando o uso da tecnologia para conectar os assuntos e relacionando o ensino ao mundo exterior. O movimento STEAM favorece a aquisição de habilidades do século XXI para que os alunos tenham proficiência em colaboração, questionamento, resolução de problemas e pensamento crítico (GUNN, 2017).

John Maeda, ex-diretor da Rhode Island School of Design de 2008 a 2013, defendeu o movimento STEM para STEAM, fazendo campanhas para adicionar Artes ao currículo STEM e trazendo a iniciativa para o primeiro plano dos formuladores de políticas educacionais, com o argumento que criatividade e o

pensamento de design (*design thinking*) são ingredientes essenciais para a inovação (GUNN, 2017).

Os objetivos do movimento STEAM incluem: transformar a política de pesquisa para enfatizar a Arte + Design em STEM; defender a integração da Arte na educação da pré-escola à pós-graduação; criação de negócios baseados na chamada Economia Criativa; e influenciar as empresas a contratarem colaboradores com formação em Artes e Design para impulsionar a inovação.

De acordo com Henriksen, Mehta e Mehta (2019) as artes e ciências são muitas vezes tratadas como domínios de conhecimento distintos, exigindo habilidades e competências diferentes e gerando uma falsa dicotomia entre as “ciências duras” versus as disciplinas artísticas. Na realidade, a criatividade é uma habilidade mental central tanto na educação STEAM assim como no design, na medida em que leva ao pensamento mais impactante não apenas nas artes, mas também nas disciplinas STEM. O pensamento de design contesta essa dicotomia entre artes e STEM, pois ela reduz os limites disciplinares que são mais tênues do que se costuma pensar. O design está na interseção entre arte e ciência e se aplica a uma ampla gama de disciplinas centradas no ser humano por meio do trabalho criativo (Fig. 4).

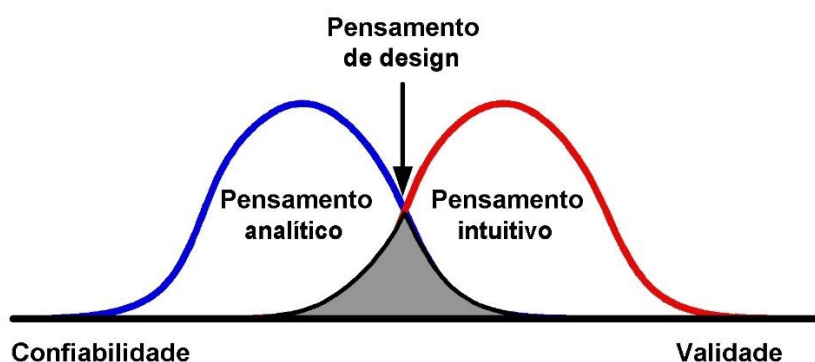


Fig. 4. Regiões delimitadoras do pensamento analítico, pensamento intuitivo e pensamento de design. Fonte: Henriksen, Mehta e Mehta (2019).

O engajamento geral dos alunos para STEM e STEAM pode acontecer por meio de práticas de ensino inovadoras. Não é suficiente preencher os horários dos alunos com aulas STEM ou STEAM ou aumentar a quantidade e o rigor de testes de avaliação. A motivação do interesse em STEM ou STEAM significa torná-los interessantes e com significado para os estudantes (MARTINEZ, 2017).

Em termos do Brasil, os índices de avaliação da Educação Básica atestam a necessidade em superar os limites da educação STEM e ampliá-la com Artes e Humanidades de modo a integrar competências transversais à formação STEM. Todo engenheiro que surge com uma inovação, pratica muito mais do que matemática, engenharia e proezas tecnológicas. Eles também usam o pensamento de design, criatividade, comunicação e habilidades artísticas para trazer essas inovações para o mercado. Isso em contraposição à ideia antiquada de que cientistas são trabalhadores isolados escondidos em laboratórios. Os inovadores de hoje são pessoas criativas, trabalhando em colaboração em espaços de trabalho informais, compartilhando ideias globalmente com outros pensadores e combinando suas capacidades STEM com alguns talentos STEAM (GUNN, 2017). Os atuais funcionários do Google não são apenas engenheiros de software ou codificadores que estão sentados em salas escuras iluminadas apenas por monitores de computador. Eles são pensadores de design que dedicam 20% do seu tempo desenvolvendo projetos de paixão (KATZ, 2015).

O conjunto de habilidades vital para sucesso nas carreiras profissionais de engenharia e tecnologia incluem agilidade de aprendizagem, colaboração e trabalho em equipe, perseverança, curiosidade e capacidade de formular questões sobre todos os aspectos do conhecimento (PALMER; BLAKE, 2019). Na interseção entre Tecnologia e Artes (KATZ, 2015) o papel desempenhado pela criatividade na promoção do progresso material, intelectual e espiritual do ser humano para satisfazer a sua necessidade de realização pessoal, de acordo com a pirâmide de necessidades humanas formulada por Abraham Maslow (1954) e que está representada na Figura 5.



Fig. 5. Pirâmide de necessidades de Maslow. Fonte: Adaptado de Maslow (1954).

3. METODOLOGIA DE ENSINO BASEADO EM PROJETOS

Neste tópico descreveremos a metodologia empregada no ensino prático de eletrônica e programação por meio de projetos, adotando a filosofia STEAM como norteador pedagógico.

As atividades didáticas foram realizadas em dois ambientes educacionais distintos: uma turma de ingressantes do curso de graduação em Engenharia Física na Escola de Engenharia de Lorena da Universidade de São Paulo; uma turma de segundo e terceiro ano do ensino médio de escola pública estadual da cidade de Lorena, estado de São Paulo. Os alunos do curso de graduação de Engenharia Física cursarão a disciplina obrigatória Introdução à Eletrônica e Computação Física, enquanto que os alunos do ensino médio realizarão as atividades didáticas de Arduino como voluntários num projeto de pré-iniciação científica (ou iniciação científica júnior) oferecido no horário de contraturno escolar. Em ambas as turmas, as atividades didáticas serão as mesmas, indistintamente ao grau de formação prévia do educando. Foram aproximadamente 38 alunos de graduação, 24 de ensino médio e oito de ensino fundamental 1.

As aulas são ministradas tanto presencialmente quanto a distância utilizando a plataforma MOODLE. No caso das aulas presenciais, as atividades práticas de montagem de circuitos e a programação do Arduino são realizadas em duplas, de maneira que haja participação efetiva dos estudantes de forma colaborativa. Cada dupla recebe um kit Arduino básico para ser usado em sala de aula e em casa durante um semestre letivo. O kit básico contém uma placa Arduino modelo Uno, um cabo USB, uma “protoboard”, fios “jumpers” para conexão elétrica e diversos componentes eletrônicos, tais como: resistores, capacitores, sensores, atuadores, motor, mostrador LCD, diodos, transistores e circuitos integrados. Um tutorial acompanha cada kit contendo instruções básicas de instalação e montagem de circuitos eletrônicos empregando os componentes disponíveis no kit. Em cada semana, os alunos devem desenvolver um pequeno projeto de aplicação do Arduino, como por exemplo, a montagem de um alarme acionado por presença ou um semáforo interativo. Os projetos constam no tutorial, que fica disponível *on-line*.

O ensino de programação é feito tanto por meio da IDE Arduino quanto pelo Processing. Na IDE Processing é suportada a linguagem C++ orientada a

objeto. Nas aulas são apresentados os conceitos de eletrônica e programação de forma prática, procurando, por meio de programas escritos em linguagem C, controlar os dispositivos eletrônicos, adquirir sinais de sensores e apresentar os dados obtidos pelo Arduino transferidos para o computador pela porta USB para serem apresentados de forma gráfica na IDE Processing. Dessa forma, sendo o hardware de baixo custo e o software gratuito, se permite com pequeno investimento, implementar uma metodologia de aprendizagem ativa “mão-na-massa”, na qual os alunos aprendem fazendo e desenvolvendo projetos contextualizados com sua formação.

Ao final do curso, os alunos devem apresentar um projeto completo com protótipo operacional, detalhando custo de montagem, procedimentos de montagem, teste e depuração, revisão da literatura, resultados e conclusão. O protótipo é apresentado perante uma banca avaliadora que julgará aspectos relacionados à originalidade da ideia, complexidade, dificuldade de montagem e programação, sustentabilidade e inovação.

4. CONCLUSÃO

Desde 2014 a disciplina de Eletrônica e Computação Física vem capacitando mais de 100 alunos no uso do Arduino em projetos científicos e tecnológicos. Além da aprendizagem técnica em eletrônica e programação de computadores, os alunos foram estimulados ao trabalho em equipe de forma colaborativa, a comunicação interpessoal e na formação de mentalidade empreendedora enxuta (*lean*).

Dentre os alunos oriundos do ensino médio, pelo menos 20% deles ingressaram em universidades para estudar, na maioria dos casos, cursos de engenharia, ciências básicas ou tecnologia. Dos alunos dos cursos de engenharia, está havendo uma gradativa transformação na mentalidade, de alunos passivos para assumirem o protagonismo na definição de suas carreiras profissionais.

Por meio do aprendizado de eletrônica e programação, empregando as plataformas Arduino e Processing e a filosofia STEAM de educação científica, tecnológica e humano-social, procuramos estabelecer nos estudantes de ensino básico e superior uma mentalidade criativa orientada ao empreendedorismo e a inovação.

REFERÊNCIAS

ALTAIR 8800. In: Wikipedia, the free encyclopedia. Disponível em: <https://en.wikipedia.org/wiki/Altair_8800>. Acesso em 18 abr. 2019.

ARDUINO. In: Wikipedia, the free encyclopedia. Disponível em: <<https://en.wikipedia.org/wiki/Arduino>>. Acesso em 15 abr. 2019.

BARRAGÁN, H. **Wiring: prototyping physical interaction design**. Dissertação (Mestrado em Design). Interaction Design Institute Ivrea, Itália, 2004. Disponível em: <http://people.interactionivrea.org/h.barragan/thesis/thesis_low_res.pdf>. Acesso em 15 abr. 2019.

GONZALEZ, H. B.; KUENZI, J. J. **Science, Technology, Engineering, and Mathematics (STEM) Education: A Primer**. Congressional Research Service 7-5700 (Relatório). Congresso Nacional dos Estados Unidos, Washington, DC. Estados Unidos, 01 de agosto de 2012. Disponível em: <<https://fas.org/sgp/crs/misc/R42642.pdf>>. Acesso em 15 abr. 2019.

GUNN, J. **The evolution of STEM and STEAM in the U.S.** Concordia University Portland College of Education (Blog), Portland, Oregon, 03 nov. 2017. Disponível em: <<https://education.cu-portland.edu/blog/classroom-resources/evolution-of-stem-and-steam-in-the-united-states/>>. Acesso em 15 abr. 2019.

HENRIKSEN, D.; MEHTA, R.; MEHTA, S. **Design thinking gives STEAM to teaching: A framework that breaks disciplinary boundaries**. In: Khine, M. S.; Areepattamannil, S. (ed). STEAM Education Theory and Practice. Cham, Suíça: Springer Nature, 2019.

KATZ, B. M. **Make it new. The history of Silicon Valley Design**. Cambridge, Massachusetts: The MIT Press, 2015.

KUSHNER, D. **The making of Arduino**. IEEE Spectrum, 26 out. 2011. Disponível em: <<https://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino>>. Acesso em: 15 abr. 2019.

MASLOW, A. H. **Motivation and personality**. Harper & Row Publishers, 1954.

MCCARTHY, L.; REAS, C.; FRY, B. **Getting started with p5.js**. São Francisco, Califórnia: Maker Media, 2016.

MAEDA, J. **Artists and scientists: more alike than different**. Scientific American, Guest Blog, 11 jul. 2013. Disponível em: <<https://blogs.scientificamerican.com/guest-blog/artists-and-scientists-more-alike-than-different/>>. Acesso em 19 abr. 2019.

MARTINEZ, J. E. **The search for method in STEAM education**. Nova York: Palgrave Macmillan, 2017.

NATIONAL Academies of Science, Engineering, and Medicine. **Rising above the gathering storm: energizing and employing America for a brighter economic future**. Consensus Study Report (Relatório), Washington, DC: National Academy Press, 2007.

PALMER, K.; BLAKE, D. **Expertise competitiva**. Rio de Janeiro: Alta Books Editora, 2019.

REAS, C.; FRY, B. **Getting started with Processing**. Sebastopol, Califórnia: O'Reilly Media, 2010.

REAS, C.; FRY, B. **Processing: a programming handbook for visual designers**, 2 Ed., Cambridge, Massachusetts: The MIT Press, 2014.