

Sistema AP²A: ferramenta para ensino-aprendizagem de algoritmos

Irving Lucas¹, Thiago Sales¹

¹NCEX – Universidade Federal de Alagoas (UFAL)
Av. Manoel Severino Barbosa - Bom Sucesso, Arapiraca - AL, 57309-005

irving.lucas@gmail.com, thiagobrunoms@gmail.com

Abstract. *The inclusion of programming education in the scholar curriculum has become more common in the last years. From this perspective, this article presents the AP2A platform: an intuitive and scalable tool for teaching-learning algorithms for children from 7 years.*

Resumo. *A inclusão do ensino de programação nos currículos escolares está cada vez mais comum nos últimos anos. Diante dessa perspectiva, este artigo apresenta a plataforma AP2A: uma ferramenta intuitiva e escalável para o ensino-aprendizagem de algoritmos para crianças a partir de 7 anos.*

1. Introdução

É indiscutível a importância de que qualquer criança aprenda a ler e a escrever, ou ainda que ela aprenda os conceitos básicos e fundamentais relacionados à matemática [Leonardo et al. 2014]. A maioria das pessoas, se questionadas sobre tal importância, responderiam que sim. Apesar disso, aprende-se matemática não necessariamente para que uma pessoa se torne matemático; da mesma forma, a maioria das pessoas também não aprendem a ler e escrever para se tornarem escritores ou jornalistas. Então surge um novo questionamento: por que e para que as pessoas precisam adquirir tais conhecimentos? Elas aprendem tais habilidades pois são importantes para vários aspectos da vida. Da mesma forma, diversos pesquisadores [Barshay 2013] acreditam que a habilidade de programar e desenvolver algoritmos para resolução de problemas já é e será uma habilidade importante para diversos aspectos da sociedade num futuro próximo, mesmo que as pessoas não se tornem programadoras profissionais.

Com efeito, a habilidade de programar dá ao indivíduo a capacidade de organizar ideias de forma estruturada e interligada. Diz-se que não se aprende a escrever, mas se escreve para aprender. Da mesma forma, acredita-se que não se deve apenas aprender a programar, ou a desenvolver algoritmos, mas programar para aprender. Nesta perspectiva, diversas instituições [Dredge 2014] de ensino básico, fundamental e médio de diversos países, propuseram, dentro do processo de ensino-aprendizagem, a inclusão do ensino da programação como disciplina básica a ser ofertada a crianças a partir de 7 anos.

Baseando-se nessa perspectiva, este trabalho apresenta um projeto de hardware e um protocolo de rede que permitem criar soluções sistematizadas e flexíveis para dar suporte ao aprendizado de novos conceitos através da programação. Ademais, a solução proposta permite que outros pesquisadores possam também estender e desenvolver

soluções diferenciadas e flexíveis voltadas ao ensino da programação. Com efeito, crianças poderão lidar com diferentes tipos de cenários enquanto incluem a programação de computadores como ferramenta meio para o aprendizado de novos conceitos.

2. Trabalhos relacionados

Atualmente, existem duas principais soluções em hardware para dar suporte ao processo de ensino-aprendizagem de algoritmos, a saber: *Cubetto* e *Osmo*.



Figura 1. Projeto *Cubetto*

O projeto *Cubetto* [Yacob 2019] pode ser visualizado na Figura 1. Uma *board* possui orifícios nos quais podem ser encaixados blocos com diferentes formatos. Cada bloco é um comando específico que será executado por um robô amigável chamado *Cubetto*, um carrinho que se locomove com base nos comandos encaixados na *board*. O robô deve se locomover em cima de um “tapete”, onde são feitos desafios para o jogador.



Figura 2. Projeto *OSMO*

Por outro lado, como ilustrado na Figura 2, o projeto *Osmo* [Sharma and Scholler 2019] é formado por um conjunto de peças que podem ser

interconectadas para criar os comandos que controlam um personagem virtual. O kit *Osmo* é composto por blocos de instrução, uma base para iPad e um adaptador para a câmera frontal do iPad. Os blocos de instrução desse kit possuem travas magnéticas que permitem a conexão dos blocos uns aos outros. Os blocos não usam qualquer tipo de fonte de energia, pois o processamento dos comandos é feito via visão computacional a partir das imagens coletadas pela câmera do iPad.

Os projetos *Cubetto* e *Osmo* são interessantes, porém, possuem limitações significativas para o processo de aprendizagem, tornando-os também menos atrativos à medida que o jogador passa a utilizá-los com frequência. No contexto do projeto *Cubetto*, o tapete possui um desafio que o robô deve interagir, limitando o número de formas que o equipamento pode ser utilizado por uma criança. Ademais, o hardware *Cubetto* não permite a utilização de mais blocos de instrução dos que aqueles já suportados na plataforma, tornando o conjunto de instruções limitado às peças já vendidas. A abordagem virtual do *Osmo* tem a vantagem de aumentar o número de possibilidades de cenários, alterando personagens e níveis de dificuldades conforme a jogabilidade desejada. Entretanto, fica limitada à empresa detentora do *Osmo* a disponibilização de novos cenários. Vale salientar que o kit *Osmo* funciona exclusivamente na plataforma iPad e não possui previsão para funcionamento em outras plataformas. Por fim, o *Osmo* limita a posição do jogador para estar exatamente de frente ao iPad.

3. Projeto AP²A

Com base na perspectiva apresentada nas seções anteriores, este projeto está inserido no contexto que envolve o ensino de programação para crianças. Mais especificamente, na temática que envolve hardware, software e protocolos de redes de computadores para a construção de ferramentas para dar suporte à criação de algoritmos. Para tal, foi projetada uma plataforma em hardware denominada AP²A (Aprender para Programar, Programar para aprender), cujo papel é fornecer uma interface de comandos genéricos para que usuários possam construir algoritmos. Além disso, foi especificado um protocolo de rede, denominado AP²A-P (AP²A Protocol), para permitir interoperabilidade com diferentes soluções no nível de software ou hardware. Tais soluções podem ser construídas por terceiros para executarem os comandos definidos pelos usuários, sendo necessário, apenas, que essas soluções de terceiros deem suporte ao protocolo AP²A-P. A arquitetura geral do projeto AP²A está ilustrada na Figura 3.

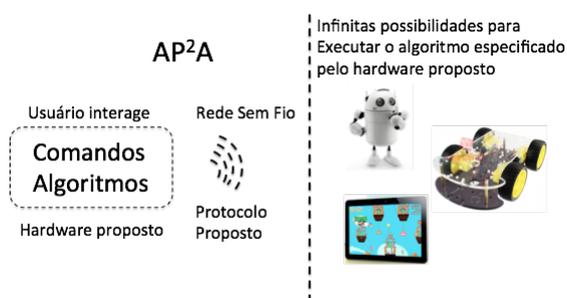


Figura 3. Arquitetura geral da solução. O projeto AP²A define os comandos e o protocolo de rede AP²A Protocol.

A estrutura do projeto consiste em uma central e um conjunto de peças que representam comandos genéricos. Essas, por sua vez, quando interconectadas de maneira sequencial à central, geram um algoritmo a ser executado por uma solução compatível que esteja conectada a essa central. Os comandos são genéricos pois a proposta é que eles possam ser usados para comandar qualquer solução que esteja conectada, podendo ser representados por símbolos que sejam intuitivos e de fácil identificação, cabendo aos desenvolvedores da solução compatível disponibilizar a interpretação dos comandos do AP²A em algum documento.

Por ser um projeto de hardware, existe uma demanda por uma plataforma de hardware estável que supra as necessidades do desenvolvimento. Para tal, foi selecionada a plataforma *Arduino*, pois é uma plataforma flexível e globalmente utilizada para prototipação de projetos em hardware.

Arduino [Arduino 2019] é uma plataforma de prototipagem eletrônica de hardware/software livres, que reúne seus recursos em uma placa projetada com um microcontrolador Atmel AVR com suporte de entrada/saída embutido, linguagem de programação padrão (essencialmente C/C++), uma IDE para escrita/compilação de código e uma vasta comunidade bastante ativa.

No contexto deste projeto, um *Arduino MEGA 2560* foi escolhido para ser a central, denominada Central AP²A, pelo fato do mesmo prover recursos de memória e processamento suficientes para conectar os componentes necessários. Para as peças que representam comandos, utilizou-se o *ATtiny85*, uma versão reduzida do *Arduino*, pois o intuito é confeccionar peças pequenas e de baixo consumo energético.

A interconexão dos comandos com a central necessita de uma interface padronizada de comunicação física e um protocolo de comunicação serial para transmissão de informações. Para tal, considerou-se a interface USB, pois esta se mostra uma ferramenta intuitiva para o usuário e se encaixa na proposta do projeto. Já para comunicação serial, considerou-se o protocolo I²C através da biblioteca *Wire.h* do *Arduino*, pois esse protocolo tem algumas particularidades como a comunicação síncrona e compartilhamento do barramento entre os dispositivos conectados.

Na Figura 4 está exposto o diagrama de uma conexão típica do protocolo I²C. No caso do AP²A, a central funcionará em modo MASTER, e os comandos funcionarão em modo SLAVE.

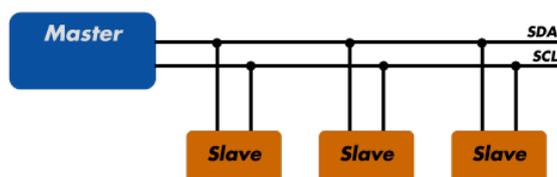


Figura 4. Conexão típica de uma comunicação I²C

Para comunicação da central AP²A com as soluções propostas, fez-se necessário especificar o protocolo AP²A-P. Este, por sua vez, tem a finalidade de determinar um for-

mato de mensagens simples para envio do algoritmo gerado, fazendo com que as partes comunicantes codifiquem e decodifiquem estas mensagens de forma padronizada. O formato dessa mensagem consiste em um objeto JSON contendo um arranjo com a sequência de números correspondentes à sequência de blocos que foram conectados na central. Para realizar a serialização/desserialização desses objetos JSON, utilizou-se a biblioteca *ArduinoJson*, a qual foi projetada especialmente para o ambiente de recursos limitados do *Arduino*. No trecho de Código 1, está descrito um exemplo de mensagem gerada após a montagem do algoritmo.

Código 1. Exemplo de mensagem do protocolo.

```
1{"instrucoes": [1, 4, 2, 3, 1, 4, 2, 3, 2, 3]};
```

Para realizar o envio desses dados, optou-se por utilizar uma conexão *wireless* entre a central e a solução compatível. Visto isso, optamos por utilizar uma comunicação via *Bluetooth* através de dois módulos compatíveis com *Arduino*, a saber: *HC-05* (para a central) e *HC-06* (para a solução compatível).

4. Resultados

Para validar o funcionamento do projeto AP²A, especificou-se uma solução compatível com o hardware e o protocolo descritos anteriormente. Foi idealizado o projeto de um carrinho programável, onde o usuário poderá interagir utilizando as peças definindo para qual direção o mesmo prosseguirá. Para tal, se faz necessário utilizar mais uma placa *Arduino*, dessa vez em sua versão *UNO*, para interpretar o código proveniente da conexão dos comandos na central e controlar um chassi com 4 motores DC.

No presente estado de desenvolvimento do projeto, alcançaram-se os objetivos relacionados à implementação. Modelou-se um protótipo da central AP²A para que fosse possível simular na prática a intercomunicação das peças, bem como variações na demanda de energia à medida que se adiciona mais peças ao conjunto de comandos.

É apresentado na Figura 5 o diagrama geral do protótipo da central AP²A e do comando genérico. A central irá identificar o tipo de cada comando conectado solicitando essa informação sequencialmente, formando assim o algoritmo criado pelo usuário.

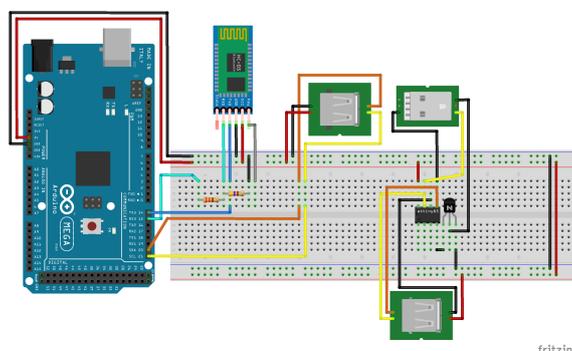


Figura 5. Protótipo da central AP²A.

Por último, como descrito na Seção 4, desenvolveu-se uma proposta de solução

compatível com o AP²A. A solução foi sugerida para que fosse possível validar a proposta principal do projeto AP²A. Como os comandos são genéricos, a solução compatível apenas precisa dar significado a esses comandos, de tal modo que a solução interprete tais comandos e reaja de acordo como especificado.

5. Considerações e trabalhos futuros

Foi exposto no contexto deste trabalho o projeto AP²A, cujo objetivo é prover uma ferramenta intuitiva e escalável para o processo de ensino-aprendizagem de algoritmos para crianças a partir de 7 anos. É sabido que a programação está cada vez mais difundida em escolas e universidades, e para contribuir com esses esforços, o AP²A foi idealizado visando viabilizar a criação de novos cenários e possibilidades que antes, com uso de soluções mais limitadas, não seriam possíveis.

Como trabalhos futuros, pretende-se desenvolver uma versão comercializável do projeto AP²A, minimizando os circuitos com a utilização de componentes *SMD*, bem como a modelagem de carcaças que dão aparência de brinquedo ao equipamento. Após isso, espera-se utilizar o equipamento para testes em escolas, permitindo que crianças utilizem o AP²A para se divertirem e que os professores consigam executar o processo de ensino com uma abordagem inovadora.

Referências

- Arduino (2019). What is arduino? <https://www.arduino.cc/en/Guide/Introduction>. Acessado em: 01 de julho de 2019.
- Barshay, J. (2013). Mit technology trailblazer is a critic of computerized learning. <https://hechingerreport.org/mit-technology-trailblazer-is-a-critic-of-computerized-learning/>. Acessado em: 01 de julho de 2019.
- Dredge, S. (2014). Coding at school: a parent's guide to england's new computing curriculum. <https://www.theguardian.com/technology/2014/sep/04/coding-school-computing-children-programming>. Acessado em: 01 de julho de 2019.
- Leonardo, P. P., Miarka, R., and Menestrina, T. C. (2014). A importância do ensino da matemática na educação infantil. *SIMPEMAD-Simpósio Educação Matemática em Debate*, 1:55–68.
- Sharma, P. and Scholler, J. (2019). Osmo: Use tangible pieces to create a unique, hands-on learning experience. <https://www.playosmo.com/>. Acessado em 01 de julho de 2019.
- Yacob, F. (2019). Cubetto: Screenless coding toy for girls and boys aged 3-6. <https://www.primotoys.com/>. Acessado em 01 de julho de 2019.