

## **Utilizando VArduino para Criação de Periféricos Virtuais baseados em Arduino UNO para o Simulador CompSim**

**Lucas Fontes Cartaxo<sup>1</sup>**

**Cícero Samuel Rodrigues Mendes<sup>2</sup>**

**Edson Barbosa Lisboa<sup>3</sup>**

**Guilherme Álvaro Rodrigues Maia Esmeraldo<sup>4</sup>**

### **RESUMO**

Organização e Arquitetura de Computadores é uma disciplina que trata do projeto de computadores, tendo o tópico Subsistemas de Entrada/Saída como um dos mais complexos e extensos. Na literatura científica tem-se optado pelo uso de simuladores computacionais para apoio às práticas pedagógicas, porém, devido ao uso de ferramentas de simulação muito abstratas, o tópico de entrada/saída é tratado de forma puramente conceitual, deixando lacunas no seu aprendizado. Neste artigo é apresentada uma nova ferramenta de aprendizagem, a qual implementa virtualmente diferentes componentes eletrônicos para permitir a criação de periféricos virtuais e com isso apoiar o processo de ensino-aprendizagem em Subsistemas de Entrada/saída. Para avaliar a ferramenta proposta, foram definidos cenários básicos de projetos de periféricos, onde comparou-se os periféricos reais com os respectivos periféricos virtuais. Os resultados mostram que, ao utilizar a ferramenta proposta, é possível otimizar e dinamizar estudos práticos na criação de Subsistemas de E/S, uma vez que evita-se a necessidade de conhecimentos prévios de eletrônica básica pelos estudantes e a aquisição, manipulação e manutenção de componentes eletrônicos físicos.

**Palavras-chave:** Organização e Arquitetura de Computadores. Subsistema de E/S. Simulador Computacional. Arduino.

### **1. Introdução**

Em cursos técnicos e superiores nas áreas de computação e eletrônica, a disciplina de Arquitetura e Organização de Computadores (AOC) trata do projeto de sistemas computacionais, cujos conteúdos variam desde o estudo das estruturas e comportamento dos componentes do computador até sua programação em baixo nível (ACM, 2013) (ZORZO et al., 2017). Entre esses conteúdos, um dos mais importantes consiste em Subsistemas de Entrada/Saída, o qual refere-se à interação do sistema computacional com seus periféricos.

---

<sup>1</sup>IFCE – Instituto Federal do Ceará – [lfonteesc@gmail.com](mailto:lfonteesc@gmail.com)

<sup>2</sup>IFCE – Instituto Federal do Ceará – [mr.samuelmendes@gmail.com](mailto:mr.samuelmendes@gmail.com)

<sup>3</sup>IFS – Instituto Federal de Sergipe – [edson.lisboa@academico.ifs.edu.br](mailto:edson.lisboa@academico.ifs.edu.br)

<sup>4</sup>IFCE – Instituto Federal do Ceará – [guilhermealvaro@ifce.edu.br](mailto:guilhermealvaro@ifce.edu.br)

Essa interação envolve diversos conceitos, como os de interfaces, barramentos e protocolos de comunicação, módulos de Entrada/Saída (E/S) e os próprios periféricos. Esses conceitos são considerados amplos e complexos e têm sido frequentemente abordados de forma puramente conceitual (LARRAZA-MENDILUZE; GARAY-VITORIA, 2014).

Na literatura tem-se optado pela utilização de simuladores computacionais para apoio ao aprendizado dos diferentes aspectos de AOC (FERNANDES; SILVA, 2017). Os simuladores são ferramentas que buscam representar cenários reais no projeto de sistemas computacionais e têm como principais benefícios a abstração dos diferentes recursos do computador, não necessitarem de laboratórios de hardware e de apoio de técnicos especializados e permitem rápido feedback nas simulações. Entretanto, segundo Esmeraldo et al. (2019), por conta das abstrações das ferramentas de simulação, não é possível a exploração dos detalhes técnicos que os periféricos reais apresentam. Nesse sentido, na literatura, alguns simuladores, como o SimuS (NETO; BORGES; SILVA, 2017) e CompSim (LISBOA et al., 2018), vêm com a proposta de oferecer simulações com interações com hardware físico. O Simulador CompSim, particularmente, explora, de forma prática, a interação entre um computador virtual simulável e seus periféricos, que podem ser virtuais ou reais (físicos). Para tanto, o CompSim dispõe de uma interface de E/S padronizada que permite conectar o sistema computacional simulado a diferentes periféricos. Entre seus periféricos, há uma interface para controle de uma plataforma de prototipação Arduino, com a qual é possível realizar experimentos que envolvam desde o projeto físico de periféricos eletrônicos até sistemas computacionais reais completos.

O uso do Arduino como uma ferramenta para o ensino de microcontroladores e subsistemas de E/S tem se tornado bastante popular, devido a sua simplicidade e baixo custo. No entanto, alguns estudos, como os apontados em (GONÇALVES; DURAES, 2019) e (KATAYAMA et al., 2019), demonstram que a utilização de Arduino pode apresentar desvantagens, tais como: alta demanda de tempo para configuração e montagem dos experimentos, considerando o curto período de tempo das aulas práticas em laboratório; dependendo do cenário de projeto, pode necessitar de maiores habilidades técnicas em eletrônica; com o tempo, há o desgaste de seus componentes eletrônicos; dependência de vários componentes eletrônicos complementares que podem elevar rapidamente o custo; e, estudantes que não possuem o nível adequado em eletrônica podem projetar circuitos sem os devidos cuidados técnicos que podem danificá-lo e/ou a seus módulos complementares.

Neste artigo é apresentado um novo simulador da plataforma Arduino, chamado de “VArduino” (Virtual Arduino) (ESMERALDO et al., 2020), o qual foi concebido como um periférico virtual do CompSim, para apoiar o aprendizado em subsistemas de E/S na disciplina de AOC. O simulador proposto consiste de uma ferramenta de software com interface gráfica que remete a uma plataforma Arduino e a diferentes componentes eletrônicos básicos, tais como LEDs, botões, *display* e sensores. O simulador de Arduino aqui proposto, quando integrado ao CompSim, permite a criação de cenários virtuais de projetos de sistemas computacionais semelhantes aos reais, onde pode-se explorar os detalhes de subsistemas de E/S, e com isso abstrair a necessidade da utilização do hardware físico e, consequentemente, tornar o aprendizado mais dinâmico.

O restante do artigo está dividido da seguinte forma: A Seção 2 apresenta os trabalhos relacionados; A Seção 3 apresenta os materiais e métodos utilizados no desenvolvimento deste trabalho; Na Seção 4, introduz-se o simulador CompSim e a sua interface com periféricos; A Seção 5 apresenta o simulador aqui proposto; Na Seção 6, realiza-se uma discussão sobre o potencial de aprendizado em subsistemas de E/S pelo uso do simulador aqui proposto; e, por fim, a Seção 7 demarca as considerações finais e trabalhos futuros.

## 2. Embasamento Teórico

O avanço das tecnologias de circuitos integrados e a alta demanda por novas aplicações embarcadas, fez com que a indústria apresentasse diferentes soluções para projetos de sistemas computacionais. Entre as soluções, estão as plataformas de prototipação que, em termos gerais: são placas de circuitos que incluem componentes eletrônicos comerciais, frequentemente, baseados em microcontroladores ou microprocessadores; são utilizadas para atender a requisitos de processamento e/ou de E/S do sistema em desenvolvimento; bem como simplificam o desenvolvimento e testes de novos projetos eletrônicos (GRETLEIN, 2013). As plataformas de prototipação são regularmente utilizadas em estágios iniciais do projeto de um sistema computacional por trazerem feedback rápido na verificação/validação dos requisitos, bem como possibilitam a realização de refinamentos antes da produção do sistema final. A Tabela 1 apresenta opções de plataformas de prototipação com suas respectivas características.

**Tabela 1. Comparativo entre plataformas abertas de prototipação.**

Característica	Plataforma de Prototipação				
	Raspberry Pi 3 Model A+	BeagleBone Black RevC	ESP32 Bluetooth	NodeMCU ESP8266	Arduino UNO

<b>Preço médio (R\$)</b>	299,00	500,00	85,00	50,00	55,00
<b>Área física (mm x mm)</b>	60 x 56	86,4 x 54,6	52 x 28	49 x 25,5	76 x 64
<b>Capacidade de memória</b>	512 MB SDRAM	512 MB SDRAM	520 KB SRAM	128 kB SRAM	2 KB SRAM
<b>Frequência da CPU (MHz)</b>	1.400	1.000	60 a 240	80	16
<b>Número de pinos de E/S</b>	40	65	11	11	19
<b>Sistema Operacional</b>	Sim	Sim	Não	Não	Não

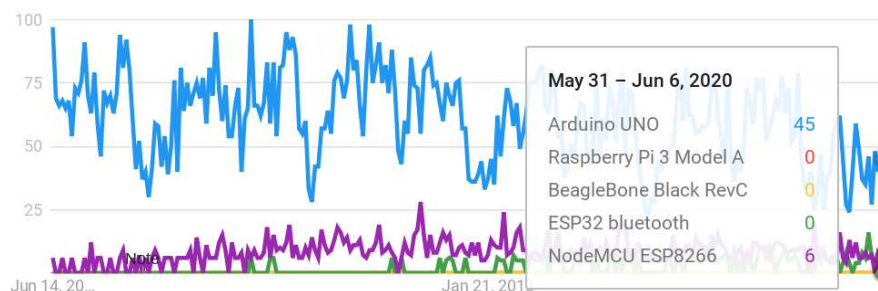
Fonte: Dos próprios autores (2020).

Analizando a Tabela 1, percebe-se que os modelos Raspberry Pi 3 Model A+ e BeagleBone Black RevC apresentaram as melhores configurações de capacidade de armazenamento em memória, frequência de operação e número de pinos de E/S, porém, seus custos médios de aquisição são bem superiores aos das demais, além de necessitarem de suporte de um sistema operacional embarcado para operação (este último fator agrega complexidade ao uso da plataforma). Os modelos Esp32 Bluetooth e NodeMCU ESP8266 apresentaram menores áreas físicas, maiores capacidades de memória e maiores frequências de operação em relação ao Arduino UNO. Ainda comparando ao Arduino UNO, ressalta-se que o NodeMCU ESP8266 apresenta menor custo médio de aquisição. Por fim, a plataforma Arduino UNO, em relação a essas duas últimas, se destacou com maior número de pinos de E/S, bem como custo médio de aquisição inferior ao do ESP32 Bluetooth.

Considera-se que a plataforma de prototipação escolhida será utilizada como periférico do simulador CompSim, ou seja, será utilizada para construção de periféricos básicos. Ao estabelecer um *trade-off* entre as métricas apresentadas na Tabela 1, pode-se concluir que o Arduino UNO apresenta o melhor custo-benefício, pois além de não necessitar de um sistema operacional embarcado para operação, possui um baixo custo médio de aquisição e um número intermediário de pinos de E/S.

O gráfico da Figura 1 mostra as tendências de pesquisas (popularidade), nos últimos 5 anos, das plataformas de prototipação apresentadas na Tabela 1. Percebe-se que, ao longo de todo esse período, Arduino UNO permaneceu a mais popular, pois, além de estar disponível para aquisição em vários web sites de vendas de produtos eletrônicos, há uma farta disponibilidade de materiais didáticos digitais, tais como livros, tutoriais, apostilas, exemplos de experimentos e de projetos completos, fóruns de discussões, entre outros. Alguns estudos, como o apresentado em (COSTA; SILVA, 2018), citam que o Arduino UNO é uma das plataformas mais utilizadas no mundo.

**Figura 1. Gráfico comparativo de popularidade entre plataformas de prototipação.**



Fonte: Autores (2020).

Arduino UNO é uma placa de circuitos eletrônicos baseada no microcontrolador Atmega328 da Atmel. Ele possui 19 pinos de E/S de propósito geral (*General Purpose Input/Output* - GPIO), que são utilizados para interagir com outros componentes eletrônicos por meio de operações de escrita (envio) e leitura (recepção) de dados digitais e analógicos (D'AUSILIO, 2012). O Arduino UNO pode ser programado e utilizado mediante uma conexão via USB com um computador. Assim, ao conectar um Arduino UNO à porta USB do computador, com seus pinos de GPIO e com disponibilidade de componentes eletrônicos complementares, é possível compor diversos tipos de projetos eletrônicos, tais como acender LEDs, ler dados de sensores e acionar motores.

Na literatura, há diferentes simuladores do Arduino UNO. O estudo apresentado por (GONÇALVES; DURAES, 2019), por exemplo, apresenta um simulador que possui uma interface web e possibilita a criação de diferentes projetos com suporte de alguns componentes eletrônicos virtuais, tais como LEDs, botões, displays e motores. Esse simulador possibilita programar a aplicação na linguagem C/C++ nativa do Arduino e, durante uma simulação, é possível visualizar e interagir com os componentes. Além desse, pode-se encontrar diferentes simuladores de Arduino UNO, tais como: Arduino Simulator (XEVRO, 2020), que apresenta uma interface para desktop, pode ser instalado em diferentes sistemas operacionais e inclui uma gama de complementos para criação de variados tipos de projetos; e Autodesk Tinkercad (PROGRAMMING ELECTRONICS ACADEMY, 2019), que consiste de um portal web educacional para aprendizado em projetos *Maker*, e que busca simplificar a construção de experimentos eletrônicos e compartilhá-los entre usuários do portal. Uma vez que os simuladores de Arduino UNO, presentes na literatura, necessitam ser programados a partir da linguagem C/C++ nativa do Arduino, percebe-se que, no contexto do CompSim, em que o Arduino UNO não é utilizado como plataforma computacional do sistema – ele é utilizado apenas para construção de periféricos e não há necessidade de programá-lo –, Revista Tecnologias na Educação -Ano 21-número/vol. 33 - Dezembro -2020 - Edição Temática XIV – - tecnologiaseducacao.pro.br - tecedu.pro.br

conclui-se que a criação de um novo simulador torna-se uma tarefa mais proveitosa, em função de buscar integrar com os já existentes.

### 3. Materiais e Métodos

Para o desenvolvimento do simulador aqui proposto, o primeiro passo consistiu no aprofundamento teórico em projetos de sistemas computacionais embarcados. Fez-se um levantamento bibliográfico e verificou-se que, dentre as opções, o Arduino UNO é uma das mais populares e possui recursos suficientes para o desenvolvimento de variados projetos eletrônicos. Em seguida, realizou-se um estudo detalhado dos recursos do Arduino UNO, visando compreender todos os seus fundamentos e características para criação de projetos eletrônicos, bem como levantou-se os requisitos considerados no desenvolvimento do novo simulador. Neste passo, foram utilizados os seguintes materiais: uma placa de circuitos do Arduino UNO e componentes eletrônicos analógicos diversos, entre eles: LEDs, botões, resistores, potenciômetros, *display* de 7 segmentos, *buzzer* (campainha), *jumpers* (fios condutores elétricos utilizados para conectar dois pontos em um circuito), sensores de luminosidade (LDR), de temperatura (NTC) e de umidade (higrômetro), entre outros.

O simulador CompSim possui um modelo de interface padronizada de E/S que permite a inclusão de novos periféricos a sua plataforma computacional virtual (ver Subseção 4.1). Desta forma, o simulador do Arduino UNO aqui proposto buscou seguir esse modelo de interface, de forma a simplificar sua integração. Para a codificação do simulador proposto, utilizou-se as mesmas tecnologias empregadas no desenvolvimento do simulador CompSim, como a linguagem Python v3.x e o *framework* gráfico Tkinter.

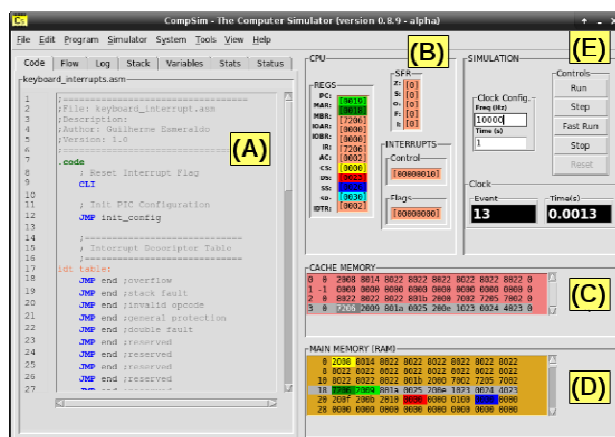
Após a codificação do novo simulador Arduino UNO, foram realizados diferentes estudos para verificar se ele seguiu os requisitos levantados para o projeto (e.g. integração com interface de E/S, suporte de interface gráfica, protocolos de comunicação e desempenho) e o comportamento de um Arduino UNO real.

### 4. O Simulador CompSim

CompSim é um simulador de sistema completo (ESMERALDO et al., 2019), o qual inclui uma Plataforma Virtual de Hardware (PVH) simulável, com os seguintes componentes do computador: 1) CPU: um processador de 16-bits, composto por 16 instruções para realização de operações de transferência de dados, lógicas e aritméticas, desvio de fluxo de programa e de entrada/saída com periféricos; 2) Memória Cache: é utilizada para otimizar o

desempenho dos programas e suporta diferentes tipos de configuração, como de técnicas de mapeamento e políticas de substituição; 3) Memória RAM: é utilizada para armazenamento de dados e instruções dos programas que serão executados no simulador; 4) Barramentos: a PVH inclui dois barramentos, sendo um de sistema, que permite a comunicação entre o processador e as memórias Cache e RAM, e um de periféricos, que permite que o processador se comunique com o Subsistema de Entrada/Saída; 5) Subsistema de E/S: consiste de uma interface padronizada que simplifica a inclusão de novos periféricos e sua comunicação com o processador. A Figura 2 ilustra a interface gráfica do CompSim.

**Figura 2. Interface gráfica do Simulador CompSim.**



Fonte: Autores (2020).

A interface gráfica do CompSim permite configurar os componentes da PVH, suportar a codificação de novos programas, realizar simulações e visualizar o comportamento da PVH durante uma simulação. A Figura 2 mostra a interface gráfica do CompSim, onde pode-se visualizar os componentes gráficos: A) Editor de código: inclui recursos de ambientes de desenvolvimento integrados (IDEs) profissionais para simplificar a codificação de uma aplicação em linguagem de máquina, como número de linhas, teclas de atalho para recursos de edição (copiar/recortar/colar, desfazer/refazer, saltar para determinada linha, etc.), um assistente de codificação (permite auxiliar a construção de instruções de código com a sintaxe correta), destaque de palavras-chave (*syntax highlight*), entre outros. Este componente está integrado a um montador (Assembler), o qual realiza diferentes análises no código-fonte da aplicação e geração de código de máquina; B) Painel de visualização do Processador: durante uma simulação, exibe os status dos componentes internos do processador; C) Painel de visualização da Memória Cache: exibe a estrutura da memória Cache e os dados armazenados, destaca também quais dados estão sendo acessados pelo processador durante

uma simulação; D) Painel de visualização da Memória RAM: exibe os dados e instruções armazenados na memória RAM (os componentes gráficos Memória RAM e processador estão integrados, de forma que, durante uma simulação, as posições de memória são destacadas de acordo com os diferentes tipos de acesso pelo processador); e E) Componentes de configuração e execução de simulações: inclui controles que permitem configurar, iniciar, executar (em modos contínuo e passo a passo com visualização, ou contínuo sem visualização para maior desempenho), parar e reiniciar uma simulação.

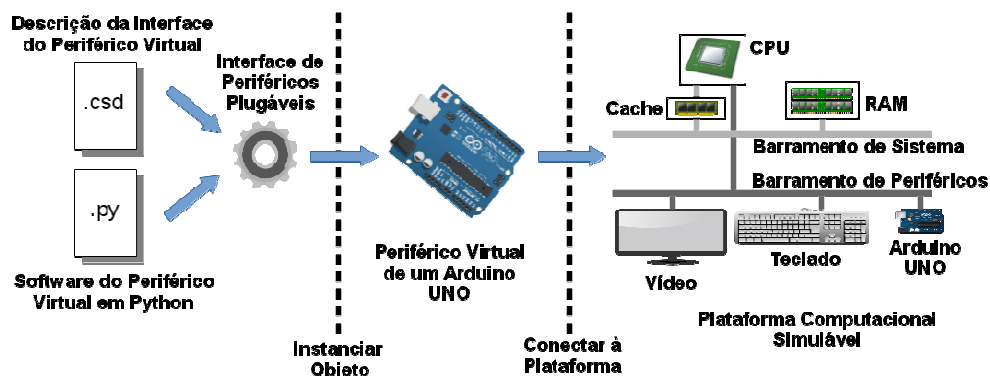
#### **4.1. Interface Padronizada para Conexão com Periféricos**

O simulador CompSim conta com uma interface de E/S padronizada para permitir a conexão simplificada de novos periféricos à PVH. Um periférico torna-se “plugável” à PVH ao implementar essa interface, sendo possível, por exemplo, plugar determinados periféricos para compor um sistema computacional específico.

Cada periférico plugável deve incluir um arquivo de especificação de sua interface (arquivo com extensão “.csd”), o qual deve incluir as seguintes definições: número(s) da(s) porta(s) de E/S; número da interrupção (IRQ), caso o periférico suporte; dados para instanciar o componente de software (nome do pacote e da classe de software); e uma descrição textual para identificação do periférico. Além disso, o software do periférico deve ser codificado com a linguagem de programação Python v3.x, assim, além do arquivo “.csd”, também haverá um outro arquivo com extensão “.py”. Desta maneira, ao se criar uma PVH no simulador, sua interface de E/S padronizada buscará os arquivos de descrição de interface de periférico (“.csd”) e, de forma automatizada, instanciará os respectivos componentes de software (via arquivos “.py”), que serão plugados ao barramento virtual de periféricos, como está ilustrado na Figura 3, abaixo. Após esse processo, os periféricos virtuais já estarão aptos para interação com o restante dos componentes de hardware do sistema computacional simulado.



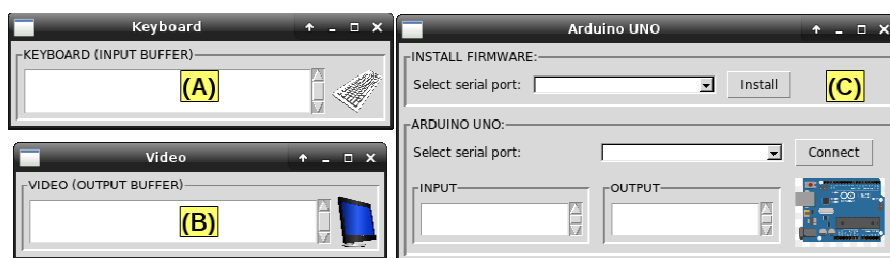
**Figura 3. Ilustração da criação de um periférico virtual plugável e conexão com a plataforma computacional.**



Fonte: Dos próprios autores (2020).

Atualmente, o CompSim conta oficialmente com os periféricos Teclado, Vídeo e Arduino UNO, como podem ser vistos nas Figuras 4-A, 4-B e 4-C, respectivamente. Os componentes Teclado e Vídeo são do tipo virtual, pois consistem de software para emular os comportamentos dos respectivos periféricos reais.

**Figura 4. Periféricos virtuais plugáveis do CompSim.**



Fonte: Autores (2020).

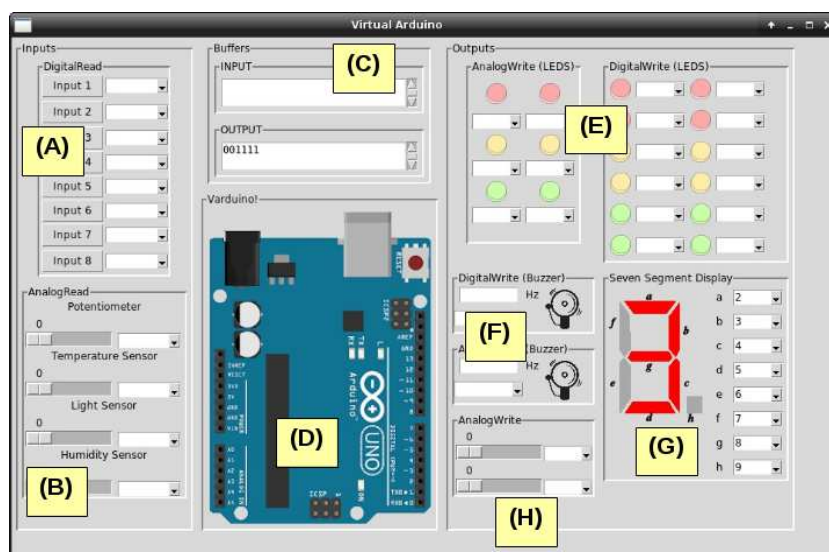
Já o periférico Arduino UNO é dividido em dois submódulos, sendo que o primeiro consiste de um software que implementa a integração do dispositivo ao barramento de periféricos, e o segundo consiste de uma placa de circuitos física da plataforma aberta de prototipação Arduino UNO (BADAMASI, 2014). Com este periférico do Arduino UNO é possível criar cenários de projetos de sistemas computacionais em que se pode realizar a simulação da PVH interagindo com hardware real.

## 5. O Componente VArduino

O simulador VArduino foi desenvolvido como um componente virtual gráfico (ver Figura 5), plugável à PVH do CompSim, para apoiar o estudo das interações de E/S entre o sistema computacional simulado e periféricos montados com apoio do Arduino UNO. A

ideia básica é permitir a criação de cenários virtuais de projetos de sistemas computacionais semelhantes aos reais, abstraindo a necessidade do uso de um Arduino UNO físico.

**Figura 5. Interface gráfica do Arduino Virtual.**



Fonte: Autores (2020).

A interface gráfica do VArduino é composta por um Arduino UNO virtual e por diferentes componentes gráficos que emulam dispositivos eletrônicos reais, denominados aqui de Dispositivos Eletrônicos Virtuais (DEV). A interface gráfica está dividida em três macroblocos de componentes gráficos, onde: do lado esquerdo, estão os DEVs de entrada de dados, que são 8 botões (*push buttons*) (Figura 5-A), 1 potenciômetro e 3 sensores analógicos (temperatura, luminosidade e umidade) (Figura 5-B); ao centro, os componentes gráficos apresentam os *buffers* de entrada (*input*) e saída (*output*) (Figura 5-C), usados na comunicação entre o simulador CompSim e o periférico VArduino, e, logo abaixo, há o diagrama do Arduino UNO virtual (Figura 5-D); por fim, à direita, estão os DEVs de saída de dados, que são 18 LEDs (Figura 5-E) e *buzzers* (Figura 5-F), os quais podem ser utilizados em operações de escrita digital/analógica, um *display* de 7 segmentos (Figura 5-G), e dois dispositivos genéricos de escrita analógica (Figura 5-H), que podem simular, por exemplo, o acionamento de *coolers*/exaustores de computadores, acionar e controlar o movimento de motores, gerar sinais modulados para controle de um LED infravermelho (e.g. para uso como controle remoto), gerar efeitos sonoros, entre outros.

Para o controle dos DEVs, é necessário inicialmente conectá-los ao VArduino. Para tanto, cada DEV inclui uma ou mais caixas de combinação (*combo boxes*), nas quais é possível escolher em quais pinos de GPIO do VArduino os dispositivos estarão conectados

(ver Figura 5). O VArduino simula os quatro modos de operação dos pinos de GPIO de um Arduino UNO real, que são escrita e leitura digital, bem como escrita e leitura analógica. A partir do suporte de conexões aos pinos de GPIO e dos modos de operação do VArduino, uma PVH poderá interagir com os DEVs.

## 6. Discussão e Resultados

Para demonstrar os benefícios do uso do simulador VArduino, foram definidos quatro cenários de sistemas computacionais virtuais no CompSim. O objetivo do estudo consistiu em criar quatro sistemas computacionais diferentes que interagem com um periférico Arduino, em que, para cada cenário, realizou-se uma comparação entre o uso de uma placa de circuitos do Arduino UNO, conectada a diferentes componentes eletrônicos, e o VArduino. Para as comparações em cada cenário, definiu-se dois estudos que são a análise da complexidade para configuração dos periféricos propostos e análise do desempenho de simulação.

Os cenários de sistemas computacionais propostos consistiram de aplicações iterativas, cujas tarefas executadas continuamente estão descritas a seguir:

1. Piscar um LED: a aplicação consiste em ligar um LED, que permanece por um certo intervalo de tempo ligado, e, em seguida, é desligado pelo mesmo intervalo de tempo;
2. Controlar a luminosidade de um LED utilizando um potenciômetro: a aplicação consiste em aferir a resistência configurada em um determinado potenciômetro e, dependendo do valor aferido, configura-se a intensidade de luminosidade de um LED;
3. Acender LED vermelho ou verde, a depender da luminosidade do ambiente: a aplicação consiste em aferir a luminosidade do ambiente, através de um sensor de luminosidade, e, em ambientes considerados claros, acende-se o LED verde, já em ambientes considerados escuros, acende-se o LED vermelho.; e
4. Imprimir a sequência numérica de 0 a 9 em um *display* de 7-segmentos: a aplicação consiste em imprimir uma sequência numérica de 0 a 9 no *display* de 7-segmentos.

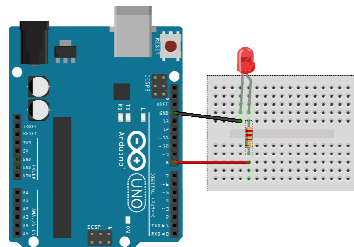
Para a criação dos sistemas computacionais virtuais, inicialmente, no CompSim, foi criada e configurada uma PVH para cada um dos cenários de projeto propostos e, em seguida, codificou-se o software da aplicação de controle dos respectivos periféricos com Arduino.

O passo seguinte consistiu em compor os periféricos dos cenários propostos. Dessa forma, optou-se, a princípio, por configurar os periféricos de hardware físico. Assim, foi necessário – além do periférico plugável do Arduino UNO (ver Figura 4-C), uma placa de

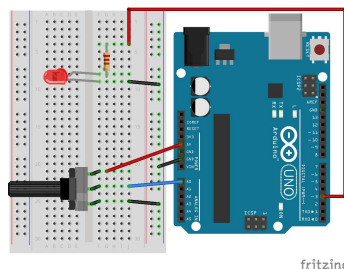
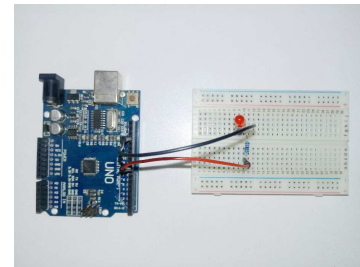
circuitos do Arduino UNO rev3, uma *proto board* e alguns *jumpers*, utilizados para conectar os componentes eletrônicos com apoio da *proto board* – o suporte dos seguintes componentes, de acordo com cada cenário:

1. Cenário 1 - Piscar LED: um resistor de 220 Ohm e um LED. O diagrama de montagem e o respectivo circuito montado podem ser vistos na Figura 6 (a);
2. Cenário 2 - Controlar a luminosidade de um LED utilizando um potenciômetro: um potenciômetro de 10k Ohm, um resistor de 220 Ohm e um LED. O diagrama de montagem e o respectivo circuito montado podem ser vistos na Figura 6 (b);
3. Cenário 3 - Acender LED vermelho ou verde, a depender da luminosidade do ambiente: um resistor de 10K Ohm, um sensor de luminosidade (LDR), dois resistores de 220 Ohm e dois LEDs, sendo um da cor vermelha e outro da cor verde. O diagrama de montagem e o respectivo circuito montado podem ser vistos na Figura 6 (c);
4. Cenário 4 - Imprimir sequência numérica de 0 a 9 em um *display* de 7-segmentos: um *display* de 7-Segmentos (do tipo ânodo comum) e 8 resistores de 220 Ohm. O diagrama de montagem e o respectivo circuito montado podem ser vistos na Figura 6 (d).

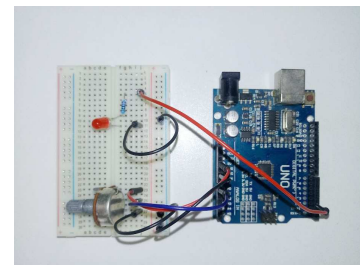
**Figura 6. Diagramas de montagem e circuitos montados para os cenários de projeto propostos.**

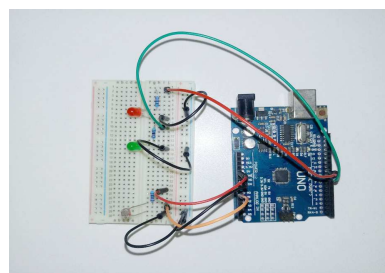
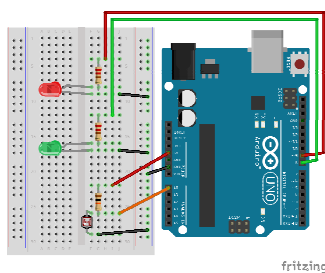


(a) Cenário 1: Piscar LED.

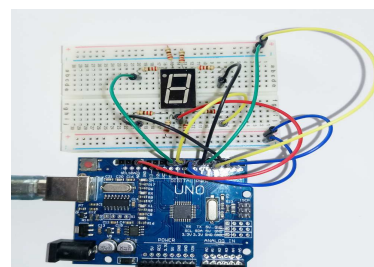
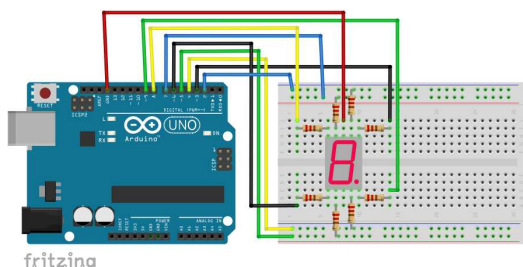


(b) Cenário 2: Controle de luminosidade de LED com potenciômetro.





(c) **Cenário 3: Acender LED verde ou vermelho, dependendo da luminosidade do ambiente.**



(d) **Cenário 4: imprimir a sequência numérica de 0 a 9 em um *display* de 7-segmentos.**

Fonte: Autores (2020).

Analisando os cenários de projeto aqui propostos, verifica-se que consistem de aplicações bastante simples, comparados a sistemas mais elaborados, tais como os presentes em aplicações de robótica, automação, internet das coisas e computação de alto desempenho. Ao analisar os diagramas de montagem e respectivos circuitos montados apresentados na Figura 6, percebe-se que a confecção dos periféricos físicos, por um estudante não introduzido em eletrônica, pode se tornar uma tarefa complexa, pois envolve o conhecimento das especificidades e a aquisição de cada componente eletrônico, além dos procedimentos necessários para montagem dos circuitos com auxílio da *protoboard*.

Em seguida, considerando que foram utilizados os mesmos sistemas computacionais virtuais criados anteriormente no CompSim, utilizou-se o VArduino para implementação dos periféricos de cada um dos cenários propostos. Sabendo que o VArduino inclui um Arduino UNO, LEDs analógicos e digitais em diferentes cores, potenciômetro, sensor de luminosidade e *display* de 7-segmentos virtuais, o passo seguinte consistiu em configurar as conexões entre eles para compor cada um dos cenários propostos. Esse procedimento foi realizado pelo uso das caixas de combinação, selecionando quais pinos de GPIO do Arduino UNO estão conectados a cada um dos componentes eletrônicos virtuais.

Por exemplo, para a configuração do periférico do Cenário 4, na Figura 5-G, à direita do *display* de 7-segmentos, pode-se visualizar as caixas de combinação com a configuração de conexão dos pinos de GPIO com os segmentos (LEDs). Ressalta-se que cada um desses segmentos recebe uma identificação, uma letra de “a” à “h”, e que os números nas caixas de combinação ao lado consistem dos respectivos pinos de GPIO utilizados nas conexões. Revista Tecnologias na Educação -Ano 21-número/vol. 33 - Dezembro -2020 - Edição Temática XIV – -tecnologiasnaeducacao.pro.br - tecedu.pro.br

Analizando as configurações de conexão dos pinos de GPIO do Arduino UNO com os LEDs do display de 7-segmentos nos periféricos físico e VArduino, apresentadas nas Figuras 6 (d) e 5-G, respectivamente, percebe-se que são idênticas.

Desta maneira, observa-se que ao abstrair o uso de dispositivos eletrônicos, a solução aqui proposta para simulação de uma plataforma Arduino UNO, VArduino, trata diretamente os problemas apontados em (GONÇALVES; DURAES, 2019) e (KATAYAMA et al., 2019), eliminando assim os custos na aquisição e manutenção dos componentes eletrônicos, bem como reduzindo os ônus no aprendizado dos conceitos básicos de eletrônica e no manuseio desses dispositivos.

O segundo estudo comparativo aqui proposto está relacionado ao desempenho de simulação dos sistemas computacionais criados para cada um dos cenários de projeto propostos, considerando periféricos físicos – implementados com o componente virtual Arduino (Figura 4-C), uma placa de circuitos Arduino UNO rev3 e componentes eletrônicos reais – e virtuais, implementados com o VArduino. Para todos os cenários, definiu-se, como configurações de simulação, a frequência do relógio (*clock*) do sistema em 1 kHz e o tempo simulável de 10 segundos. A Tabela 2 mostra os resultados de desempenho de simulação para os cenários de projetos propostos, considerando os respectivos periféricos físicos e virtuais.

**Tabela 2. Desempenhos de simulação para os cenários de projeto propostos.**

Cenário	Tempo Médio de Simulação Arduino UNO (s)	Tempo Médio de Simulação VArduino (s)
1. Pisca LED	14,4	14,1
2. Controlar a luminosidade de um LED utilizando um potenciômetro	15,2	14,1
3. Acender LED vermelho ou verde, a depender da luminosidade do ambiente	14,5	13,5
4. Imprimir sequência numérica de 0 a 9 em um display de 7-segmentos	18,9	21,9

Fonte: Autores (2020).

Analizando a Tabela 2, observa-se que houve pequenas diferenças entre os tempos médios de simulação para os cenários de projeto propostos, considerando os periféricos físicos e virtuais. Os cenários 1, 2 e 3 obtiveram melhores tempos médios com o uso do periférico virtual VArduino. Já no cenário 4, o periférico físico apresentou tempo médio melhor para a aplicação de impressão de sequência numérica no *display* de 7-segmentos. Este último resultado justifica-se pelo fato de que, no VArduino, o procedimento de “ligar um

Revista Tecnologias na Educação -Ano 21-número/vol. 33 - Dezembro -2020 - Edição Temática XIV – -  
tecnologiasnaeducacao.pro.br - tecedu.pro.br

segmento” do *display* de 7-segmentos consiste em plotar graficamente um polígono. Dessa forma, imprimir um número no *display* de 7-segmentos do VArduino consiste em plotar dois ou mais polígonos, o que, conseqüentemente, acrescenta *overhead* no tempo de simulação.

Observa-se ainda que, nos cenários de projeto de sistemas computacionais que fazem uso de periféricos físicos, o esforço maior e, por consequência, a maior parcela de tempo consumido, concentra-se na composição do próprio periférico. Esse esforço pode ser minimizado ao utilizar a solução aqui proposta, permitindo, desta maneira, a composição ágil de periféricos para uso em diferentes cenários de projetos de sistemas computacionais e obter desempenhos médios de simulação comparáveis aos que fazem uso de periféricos físicos.

## 7. Conclusões

Neste artigo apresentou-se o VArduino, um simulador criado para apoiar o processo de ensino-aprendizagem em conceitos de subsistemas de E/S nas disciplinas de AOC. VArduino inclui componentes gráficos que emulam um Arduino UNO e diferentes componentes eletrônicos, e pode ser utilizado na criação de diferentes periféricos virtuais para o simulador CompSim. Os estudos de caso apresentados mostraram que, em relação a uma placa de circuitos física do Arduino UNO, ao utilizar o VArduino, pode-se eliminar custos de aquisição e manutenção de componentes eletrônicos, abstrair a necessidade de conhecimentos prévios de eletrônica básica, tornar a criação de novos periféricos mais dinâmica e obter desempenhos comparáveis de simulação, em relação ao uso de periféricos físicos.

O simulador CompSim, que já vem sendo utilizado por turmas de diferentes cursos técnicos e de graduação, nas áreas de computação e eletrônica, tem se mostrado efetivo no apoio ao processo de ensino-aprendizagem dos conceitos de AOC. Da mesma maneira, espera-se, no futuro, também validar o VArduino junto à comunidade acadêmica, através da criação de diferentes cenários de aplicações computacionais com requisitos de E/S, para aplicação em práticas laboratoriais. Desta maneira, espera-se que os *feedbacks* possam contribuir na correção de falhas, adição de novos recursos, no aumento da qualidade da solução e no aperfeiçoamento do apoio ao aprendizado em subsistemas de E/S.

## 8. Referências Bibliográficas

ACM. Association for Computing Machinery; IEEE Computer Society. **Curriculum Guidelines for Undergraduate Degree Programs in Computer Science**, 2013. Disponível

em: <[https://www.acm.org/binaries/content/assets/education/cs2013\\_web\\_final.pdf](https://www.acm.org/binaries/content/assets/education/cs2013_web_final.pdf)>. Acesso em: 8 jun. 2020.

XEVRO. **Arduino Simulator 1.5**, 2020. Disponível em: <<https://xevro.be/products/arduino%2520simulator%25201.5.html>>. Acesso em: 8 jun. 2020.

PROGRAMMING ELECTRONICS ACADEMY. **The Arduino Simulator you've been looking for!**, 2020. Disponível em: <<https://www.programmingelectronics.com/arduino-simulator-tinkercad/>>. Acesso em: 8 jun. 2020.

BADAMASI, Y. A. **The working principle of an Arduino**. In: 11th International Conference on Electronics, Computer and Computation (ICECCO), 2014. pp. 1-4.

COSTA, E.; SILVA, G. P. Um Simulador Didático para o Ensino de Arquitetura de Computadores e Internet das Coisas. In: **International Journal of Computer Architecture Education (IJCAE)**, v. 7, n. 1, 2018. pp. 29-38.

D'AUSILIO, A. **Arduino: A low-cost multipurpose lab equipment**. In: Behavior research methods, 44(2), 2012. pp. 305-313.

ESMERALDO, G. A. R. M.; CARTAXO, L. F.; MENDES, C. S. R.; LISBOA, E. B. **VArduino: Um Componente Virtual do Arduino UNO como Dispositivo Periférico do Simulador CompSim**. In: Anais do V Congresso sobre Tecnologias na Educação (Ctrl+E), 2020.

ESMERALDO, G. A. R. M.; MENDES, C. S. R.; CARTAXO, L. F.; LISBOA, E. B. Apoio ao Aprendizado em Arquitetura e Organização de Computadores: Um Estudo Comparativo entre Simuladores Computacionais/ In: **Revista Tecnologias na Educação**, 2019.

FERNANDES, S. R.; SILVA, I. S. Relato de Experiência Interdisciplinar Usando MIPS. In: **International Journal of Computer Architecture Education (IJCAE)**, V. 6, n. 1, 2017. pp. 52-61.

GONÇALVES, P. F.; DURAES, J. **An Arduino Simulator for Practical Embedded Programming Teaching**. In: International Symposium on Ambient Intelligence and Embedded Systems, 2019



GRETLEIN, S. **Software Modeling for Embedded Systems**. In: Software Engineering for Embedded Systems. Newnes, 2013. pp. 59-91.

KATAYAMA, T., NISHIDA, T., KITA, Y., YAMABA, H., ABURADA, K.; OKAZAKI, N. Implementation of Arduino Simulator ADVIS Visualizing the Value of Voltage on the Circuit. In: **Journal of Robotics, Networking and Artificial Life**, 5(4), 2019. pp. 249-252.

LARRAZA-MENDILUZA, E.; GARAY-VITORIA, N. Approaches and tools used to teach the computer input/output subsystem: A survey. In: **IEEE Transactions on Education**, v. 58, n. 1, 2014. pp. 1-6.

LISBOA, E. B.; CARTAXO, L. F.; MENDES, C. S. R.; ESMERALDO, G. A. R. M. **Ambiente Integrado de Hardware e Software Aplicado ao Ensino de Projeto de Sistemas Computacionais**. In: III Congresso sobre Tecnologias na Educação (Ctrl+E 2018), 2018.

NETO, A. M. A.; BORGES, J. A. dos S.; SILVA, G. P. **Extensão do Simulador SimuS com uso do Protocolo Firmata**. In: XVIII Workshop de Iniciação Científica do XVII Simpósio em Sistemas Computacionais de Alto Desempenho (WIC-WSCAD), 2017. p.123-128.

ZORZO, A. F.; NUNES, D.; MATOS, E.; STEINMACHER, I.; LEITE, J.; ARAUJO, R. M.; CORREIA, R.; MARTINS, S. **Referenciais de formação para os cursos de graduação em computação**. In: Sociedade Brasileira de Computação (SBC), 2017. 153p.

Recebido em Dezembro 2020

Aprovado em Dezembro 2020