Revista Brasileira de Informática na Educação – RBIE
Brazilian Journal of Computers in Education
(ISSN online: 2317-6121; print: 1414-5685)
http://br-ie.org/pub/index.php/rbie

Submission: 16/Oct/2022; 1st round notif.: 24/Jan/2023; New version: 08/Feb/2023; 2nd round notif.: 22/Mar/2023;
Camera ready: 23/May/2023; Edition review: 25/Jun/2023; Available online: 25/Jun/2023; Published: 25/Jun/2023;

# Uma Abordagem para Ensino-Aprendizado de Projetos de Sistemas Computacionais com Utilização do Simulador CompSim com Suporte à Arquitetura RISC-V

***Title: A Teaching-Learning Approach to Computer System Design Using the CompSim Simulator with RISC-V Architecture Support***

*Guilherme Álvaro R. M. Esmeraldo*
*Instituto Federal do Ceará (IFCE)*
*ORCID: 0000-0002-6632-6340*
*guilhermealvaro@ifce.edu.br*

*Edna Natividade da Silva Barros*
*Centro de Informática (CIn/UFPE)*
*ORCID: 0000-0001-6479-3052*
*ensb@cin.ufpe.br*

*Harley Macedo de Mello*
*Instituto Federal do Ceará (IFCE)*
*ORCID: 0000-0003-1177-5533*
*harley.mello@ifce.edu.br*

*Esdras L. Bispo Jr.*
*Universidade Federal de Jataí (UFJ)*
*Centro de Informática (CIn/UFPE)*
*ORCID: 0000-0002-6373-6045*
*bispojr@ufj.edu.br*

*Robson Gonçalves Fechine Feitosa*
*Instituto Federal do Ceará (IFCE)*
*ORCID: 0000-0002-0880-0956*
*robsonfeitosa@ifce.edu.br*

*Eduardo Carlos P. da S. Proto*
*Instituto Federal do Ceará (IFCE)*
*ORCID: 0009-0001-3402-820X*
*ecpsproto@gmail.com*

*Edson Barbosa Lisboa*
*Instituto Federal de Sergipe (IFS)*
*ORCID: 0009-0002-4231-1299*
*edson.lisboa@academico.ifs.edu.br*

*Gustavo Augusto Lima de Campos*
*Universidade Estadual do Ceará (UECE)*
*ORCID: 0000-0001-7175-9071*
*gustavo.campos@uece.br*

## Resumo

*O contexto tecnológico atual – com dispositivos eletrônicos inteligentes e interligados à Internet, serviços em nuvem, robótica, carros autônomos, Indústria 4.0, dentre outras inovações – só foi possível graças ao surgimento do Computador (sistemas computacionais). No processo de ensino e aprendizagem de projeto de sistemas computacionais, são necessários conhecimentos relacionados: ao projeto físico da plataforma computacional, sua programação e operação eficiente. Esses conteúdos são extensos, complexos e demandam metodologias desafiadoras e ferramentas computacionais que contemplem o processo de ensino e aprendizagem tanto do arcabouço teórico, quanto da prática de projetos. Assim, o presente trabalho tem como objetivo apresentar uma abordagem para simplificar o aprendizado por meio da prática de projetos de sistemas computacionais em ambiente de simulação. Para tanto, utilizou-se o simulador computacional CompSim que inclui um modelo de simulação de processador RISC-V. Desta forma, é possível: projetar sistemas computacionais de alta complexidade no ambiente de simulação e simulá-los para validar requisitos de projeto, tais como comportamento funcional e desempenho. Estabelece-se assim uma abordagem de apoio ao aprendizado e projeto de novos sistemas computacionais. Para validar a abordagem, foi realizada uma avaliação por meio de questionários aplicados a usuários do sistema, docentes e discentes, em cenários de estudos de casos. As respostas dos questionários, com base na análise de rubricas, permitiram analisar a qualidade da experiência de uso do simulador e do apoio pedagógico proporcionado pela abordagem.*

***Palavras-chave:*** *Apoio ao Ensino-Aprendizado; Projetos de Sistemas Computacionais; Ferramenta de Simulação; CompSim; RISC-V*

*Cite as: Esmeraldo, G. Á. R. M., Feitosa, R. G. F., Barros, E. N. S., Proto, E. C. P. S., Mello, H. M., Lisboa, E. B., Bispo Jr., E. L. Campos, G. A. L. Uma Abordagem para Ensino-Aprendizado de Projetos de Sistemas Computacionais com Utilização do Simulador CompSim com Suporte à Arquitetura RISC-V. Revista Brasileira de Informática na Educação, 31, 271-288. DOI: 10.5753/rbie.2023.2951.*

## Abstract

*The current technological context - with intelligent electronic devices interconnected to the Internet, cloud services, robotics, autonomous cars, and Industry 4.0, among other innovations - was only possible due to the emergence of the Computer. However, in the teaching and learning processes on computational systems design, knowledge related to the physical design of the computing platform, programming, and efficient operation is required. These contents are extensive and complex and demand challenging methodologies and computational tools that contemplate the teaching and learning process of both the theoretical framework and the project practice. Thus, this paper aims to present an approach to simplify learning through the practice of projects of computational systems in a simulation environment. In order to achieve this, the computational simulator CompSim with a RISC-V processor simulation model has been used. In this manner, it is possible to: design highly complex computational systems in the simulation environment and simulate them to validate design requirements, such as functional behavior and system performance, thus providing an approach to support the learning and design of new computer systems. For validating the proposed approach, an evaluation process was conducted through questionnaires applied to system users, professors, and students, in case study scenarios. The questionnaire responses, based on rubric analysis, made it possible to analyze the quality of the simulator user experience and the pedagogical support provided by the proposed approach.*

***Keywords:*** *Support for Teaching-Learning; Design of Computational Systems; Simulation Tool; CompSim; RISC-V*

# 1 Introdução

Os processos de ensino-aprendizagem em projetos de sistemas computacionais envolvem o estudo dos componentes do computador, suas funções e modelos de comunicação, bem como dos aspectos e atributos visíveis ao programador (William, 2010). Esses conhecimentos são abordados em disciplinas de cursos na área de Computação e de Engenharia Eletrônica (ACM & Society, 2013; Zorzo et al., 2017), sendo necessários ao projeto, programação e operação eficiente de sistemas computacionais (Penna & Freitas, 2013; Nikolic, Radivojevic, Djordjevic, & Milutinovic, 2009). O estado da arte em aplicações de sistemas computacionais abrange os dispositivos eletrônicos inteligentes e interligados à Internet (*Internet of Things* - IoT) (Ray, 2018), telecomunicações, serviços em nuvem, robótica, carros autônomos, Indústria 4.0, entretenimento, segurança, entre outras.

Para que os processos de ensino-aprendizagem sejam efetivos, é importante que eles incluam nas suas abordagens metodológicas, além das aulas teóricas, as práticas em laboratório de forma a permitir que os estudantes apliquem os conhecimentos obtidos em aulas teóricas, pela observação e exploração das características dos sistemas atuais (Nikolic et al., 2009). Muitas são as habilidades práticas que devem ser desenvolvidas em disciplinas que tratam de projetos de sistemas computacionais, tais como: projeto de blocos básicos de um computador; escrita de programas simples ao nível de máquina (*Assembly*) para diferentes cenários de uso do computador; análise de desempenho para acesso aos dados, considerando diferentes cenários de configurações de memória, entre outros (ACM & Society, 2013). O desenvolvimento dessas habilidades pode ser realizado com o suporte de laboratórios de hardware especializados, onde deve-se ter, como pressuposto, a disponibilidade de componentes e circuitos eletrônicos variados, bem como modernos equipamentos de alimentação, testes e medições (e.g. geradores de funções, osciloscópios, multímetros, fontes de bancada, entre outros), o que pode aumentar os custos do ambiente e necessitar de apoio técnico especializado para a montagem e manutenção do espaço físico do laboratório, bem como configuração e supervisão dos experimentos.

Visando dirimir várias dessas dificuldades, surgiram os simuladores computacionais (Penna & Freitas, 2013) como uma alternativa viável para apoio ao desenvolvimento das habilidades práticas estabelecidas em cursos que tratam de projetos de sistemas computacionais. O uso de simuladores, como prática pedagógica complementar, não é uma atividade nova (Balamuralithara & Woods, 2009). Estudos, como (Uribe, Magana, Bahk, & Shakouri, 2016; García, Pacheco, & Garcia, 2014; Balamuralithara & Woods, 2009; Tan, Tan, Fang, May, & Koh, 2009), mostram que, ao se utilizar simuladores para fins educacionais, é possível aumentar o desempenho acadêmico em cursos de tecnologia e engenharia.

Dentre os simuladores computacionais, nos últimos anos, surgiram aqueles que contemplam arquiteturas mais modernas, como é o caso da RISC-V (2022b). A RISC-V (2022a) consiste de uma especificação aberta (*open hardware*) da Arquitetura de um Conjunto de Instruções (*Instruction Set Architecture* - ISA), que tem como objetivos: 1) Atender a diferentes requisitos de aplicações, que podem variar desde pequenos sistemas embarcados até complexos sistemas de computação de alto desempenho; 2) Ter compatibilidade com diferentes softwares e linguagens de programação; 3) Comportar diferentes tecnologias de circuitos digitais para sua implementação, tais como FPGAs (*Field-Programmable Gate Arrays*) e ASICs (*Application-Specific Integrated Circuits*); 4) Ser eficiente; 5) Ser customizável para suportar especializações para atender a

diferentes requisitos de projeto; e 6) Ser estável, evitando assim descontinuidade da ISA, como ocorreu com várias ISAs proprietárias. A ISA RISC-V possui um conjunto básico de instruções, tais como RV32I, RV64I e RV128I, para tratar números inteiros (I) em arquiteturas de 32, 64 e 128-bits, respectivamente, que pode ser ampliado com extensões, tais como M (extensão para multiplicação e divisão), A (extensão para operações atômicas) e F (extensão para suportar números ponto-flutuante de precisão simples).

O projeto RISC-V, por ser aberto, oferece uma grande diversidade de recursos, tais como simuladores, compiladores, depuradores e, até mesmo, implementações abertas em Linguagens de Descrição de Hardware (*Hardware Description Language* – HDL) para síntese em FPGA. Esses recursos não só beneficiam o estudo/aprendizado dos detalhes da especificação e da programação com a ISA do RISC-V, mas também abrem um conjunto de oportunidades de modificá-la e, com isso, compreender seu processo de implementação e os respectivos impactos no desempenho, tamanho físico e consumo de energia do sistema computacional em desenvolvimento.

Assim, este artigo apresenta uma nova abordagem para apoiar os processos de ensino-aprendizagem de projetos de sistemas computacionais, por meio da realização de práticas em ambiente de simulação. Para tanto, aplicou-se o simulador computacional CompSim, que inclui um modelo de simulação de processador RISC-V, para projetar sistemas computacionais de alta complexidade no ambiente de simulação e simulá-los para validar requisitos de projeto, tais como comportamento funcional e desempenho. A abordagem proposta foi avaliada, por meio de questionários aplicados aos atores diretamente ligados ao processo educacional, docentes e discentes, em cenários de estudos de caso.

Para fins de apresentação, o restante deste artigo foi organizado da seguinte forma: na Seção 2, são ilustrados os trabalhos relacionados presentes na literatura. Na Seção 3, são listados os materiais e métodos para o desenvolvimento do presente trabalho. Na Seção 4, apresenta-se os detalhes da abordagem aqui proposta. Por fim, na Seção 5 são discutidas as considerações finais e propostas de trabalhos futuros.

## 2 Trabalhos Relacionados

### 2.1 Simuladores Computacionais

Na literatura, o uso de simuladores como abordagem de apoio ao aprendizado em projeto de sistemas computacionais não é uma abordagem recente. Alguns trabalhos, como os apresentados em (Nikolic et al., 2009; Penna & Freitas, 2013; Sartor, Soares, & Daniel, 2020; Esmeraldo, Mendes, Cartaxo, & Lisboa, 2019), realizam comparações entre diferentes simuladores, de acordo com métricas predeterminadas, para estabelecer um compromisso na adoção de algum deles e, com isso, obter apoio ao aprendizado em projetos de sistemas computacionais. Entre as métricas adotadas, destacam-se: granularidade (nível de detalhes, opções de configuração e variabilidade de componentes de hardware); desempenho e precisão dos resultados da simulação; suporte à simulação de aplicações de usuário; e interfaces de usuário (em linha de comando, gráficas e web).

Ainda na literatura, há os simuladores que buscam abstrair os detalhes de implementação

dos respectivos componentes de hardware reais, visando aumentar o desempenho das simulações, e os que abordam desde componentes específicos de hardware, para tratar de conceitos específicos ou mais avançados –, como são os casos dos simuladores de determinados processadores e de memórias cache (Xavier, Rodrigues, & Júnior, 2011) –, até os simuladores de sistemas completos (Penna & Freitas, 2013; Esmeraldo et al., 2019), que trazem uma visão macro das funcionalidades de cada componente e de comunicação entre eles. Outros trabalhos incluem simuladores com implementação em nível de hardware, como é o caso daqueles descritos em Linguagem de Descrição de Hardware (HDL) (Awedh & Mueen, 2015), com objetivo de apresentar os diferentes níveis de abstração no projeto de um sistema digital real; e, por fim, os que incluem, além do ambiente de simulação, mecanismos para sua integração com componentes físicos de hardware (Black, 2016; Neto, Borges, & Silva, 2017; Esmeraldo et al., 2019), os quais buscam estimular os estudantes a produzirem sistemas eletrônicos reais.

As abordagens que fazem uso de simuladores que atuam em nível de hardware físico são particularmente interessantes, pois, desde que lidam com componentes e/ou plataformas eletrônicas reais, oferecem oportunidades para verticalizar o aprendizado de diversos aspectos do projeto de um sistema computacional, tais como: a complexidade dos cálculos de tempos de acesso aos dados, devido às hierarquias de memórias e de barramentos; aspectos de interfaces de subsistemas de entrada/saída, como os diferentes tipos, características e protocolos de comunicação dos periféricos, entrada/saída programada e o uso de interrupções para sincronização; e suporte à análise e customização dos componentes, visando otimizar desempenho, tamanho físico e consumo de energia dos sistemas em desenvolvimento.

### 2.2 Simuladores RISC-V

Na literatura, há diferentes simuladores da ISA RISC-V, como pode ser visto em (RISC-V, 2020a). No entanto, assim como nos demais simuladores computacionais, cada simulador RISC-V possui suas próprias particularidades. Nesse sentido, visando realizar um levantamento de suas principais características, o presente trabalho, inicialmente, analisou simuladores com licenças livres, por permitirem acesso às tecnologias empregadas e à própria solução de software. Em seguida, considerando o levantamento realizado, definiu-se as seguintes métricas para fins de comparação, com respectivas descrições:

- **Interface**: modelo de interação do usuário com o simulador, em que compreende-se que o uso de interfaces mais intuitivas tratam questões como acessibilidade, usabilidade e experiência de uso do simulador;
- **IDE**: inclusão de ambiente de desenvolvimento com oferta de recursos de integrados (IDE) para dar apoio às ações relacionadas à simulação, tais como codificação da aplicação em baixo nível, configuração da plataforma de simulação, simulação, visualização, entre outras;
- **Completude**: considera-se, no presente trabalho, que um simulador completo possui implementado, pelo menos, o conjunto base da ISA RISC-V (RV32I ou RV64I, para arquiteturas 32 ou 64-bits respectivamente);
- **Interatividade**: suporte à interação com o simulador e com a aplicação durante uma simulação;

- **Visualização**: inclusão de recursos que permitam visualizar os estados da aplicação e da plataforma computacional durante uma simulação;
- **Depuração**: inclusão de recursos que permitam depurar a execução da aplicação;
- **Feedback**: inclusão de recursos que permitam avaliar o sistema computacional após simulação, visando encontrar gargalos ou otimizar o sistema (aplicação e plataforma computacional);
- **Hardware**: suporte de integração do simulador com hardware físico, visando estabelecer cenários mais complexos e reais para o aprendizado e projetos de sistemas computacionais.

A Tabela 1 apresenta um comparativo entre simuladores RISC-V analisados, considerando as métricas previamente estabelecidas.

| Simulador | Interface | IDE | Completude | Interatividade | Visualização | Depuração | Feedback | Hardware |
|---|---|---|---|---|---|---|---|---|
| Vulcan | Web | | | | x | | | |
| WebRISC-V | Web | | | x | x | x | x | |
| Venus | Web | | x | x | x | | | |
| BRISC-V | Web | | | x | x | x | | |
| emulsiV | Web | | | x | x | | | |
| rv8 | Linha de Comando | | x | x | | | x | |
| riscv-rust | Linha de Comando /Web | | x | x | x | x | | |
| Whisper | Linha de Comando | | x | x | x | x | x | |
| Spike | Linha de Comando | | x | x | x | x | | |
| riscOVPsim | Linha de Comando | | x | | | x | x | |
| TinyEMU | Linha de Comando | | x | | | | | |
| terminus | Linha de Comando | | x | | | | | |
| RISC-V Virtual Prototype | Linha de Comando Linha de | | x | | | x | | |
| RVS | Desktop | | | x | x | x | | |
| Ripes | Desktop | x | x | x | x | x | x | |
| RARS | Desktop | x | x | x | x | x | | |
| Jupiter | Linha de Comando /Desktop | x | x | x | | x | | |

Tabela 1: Comparativo entre simuladores RISC-V.

Entre os simuladores presentes na Tabela 1, Vulcan, Venus, WebRISC-V, BRISC-V e emulsiV, apresentam interface web. Vulcan implementa parcialmente as extensões RV32I, RV32M, RV32A e RV32F e apresenta um editor de código simples, componentes de visualização de registradores, de instruções, respectivos bytecodes e de memória. A simulação não é interativa e não há feedback do sistema pós-simulação. WebRISC-V permite visualizar o caminho de dados de um processador RISC-V, conteúdo da memória e registradores, bem como carregar em memória um

programa *Assembly* e executar a simulação em diferentes modos (contínuo, passo a passo progressivo e regressivo). WebRISC-V implementa ainda as extensões RV64I (com exceção da instrução "fence") e RV64M, e traz um recurso chamado de "*Pipeline Table*" que mostra um *trace* de execução da aplicação. Já o simulador Venus implementa as extensões RV32I e RV32M, apresenta um editor de código simples e mecanismos para visualização dos conteúdos dos registradores e da memória (apresenta controles para configuração de uma memória cache, porém não é possível visualizá-la). Não foi possível identificar quais extensões o BRISC-V suporta, porém, em testes simples, algumas instruções da extensão base não foram reconhecidas pelo simulador. BRISC-V permite visualização dos conteúdos dos registradores e permite adicionar *breakpoint* no código Assembly para suportar depuração. O simulador emulsiV mostra graficamente o datapath do processador, onde, durante uma simulação, é exibido o caminho em que os dados percorrem e o que está ocorrendo no processador nas etapas de execução de uma instrução (*fetch*, *decode*, ALU, *Compare*, Mem/Reg e PC).

Os simuladores Terminus, rv8, riscv-rust, Whisper, Spike, TinyEMU, RISC-V Virtual Prototype e riscOVPsim possuem essencialmente interface de linha de comando, sendo que o riscv-rust pode ser exibido na web com ajuda de um complemento de código, e implementa, pelo menos, a extensão RV32I. Os simuladores rv8 e riscv-rust permitem interagir com o simulador durante uma simulação, bem como o rv8 traz, como feedback de simulação, histogramas de registradores, instruções e frequência do contador de programa, enquanto que o riscv-rust apresenta um debugger com suporte de execução passo a passo, inclusão de breakpoints, bem como a visualização de conteúdo de registradores e memória. Os simuladores Whisper e Spike apresentam diferentes modos de execução, como contínuo e interativo, sendo que neste último modo é possível controlar o fluxo de execução de simulação, examinar os conteúdos dos registradores e da memória (Whisper permite modificar esses conteúdos). Antes de iniciar uma simulação, Whisper pode ser parametrizado para: suportar diferentes extensões e arquiteturas 32 e 64-bits do RISC-V, gerar *traces* e profiles de aplicação, depuração do programa, entre outros. Da mesma forma, Spike suporta diferentes extensões/arquiteturas e depuração do programa. riscOVPsim se destaca por implementar uma grande variedade de extensões e as arquiteturas de 32, 64 e 128-bits, bem como suporta geração de *traces* de execução, depuração do programa com suporte do GDB (*GNU Debugger*) e geração de estatísticas básicas de execução do processador (como o número de instruções executadas por tempo de simulação - MIPS). Por fim, TinyEMU, terminus e RISC-V *Virtual Prototype* são os mais simples, dentre os simuladores com interface de linha de comando, e suportam apenas execução contínua, sendo que o último permite a depuração da aplicação com suporte do GDB RSP (*GNU Debugger Remote Serial Protocol*).

Ripes, RARS e Jupiter possuem interface gráfica *desktop*, sendo que: Ripes implementa as extensões RV32I e RV32M, com diferentes recursos de *pipeline* (*5-Stage, 5-Stage w/o Forwarding or Hazard Detection*); RARS implementa várias extensões e as arquiteturas 32 e 64-bits; e Jupiter, as extensões RV32I e RV32F. Os três apresentam recursos para interação com o simulador e depuração do programa; Ripes e RARS permitem a visualização dos conteúdos dos registradores e memória; Ripes apresenta ainda a visualização das instruções em execução em cada ciclo do pipeline e estatísticas de memória *cache*). Jupiter possui adicionalmente um modo de execução em linha de comando, onde é possível parametrizar a simulação para execução ou depuração, as opções de configuração da memória cache (associatividade, tamanho e número de blocos e a política de substituição, que varia entre LRU, FIFO ou *Random*).

Em geral, percebe-se que os simuladores com interface web trazem recursos de interação com o simulador e visualização gráfica da aplicação e da plataforma computacional; os simuladores com interface em linha de comando são completos, por implementar todas as instruções presentes na especificação da extensão base (e de outras extensões); e os simuladores com interface *desktop* trazem recursos integrados (IDE) que auxiliam a codificação das aplicações, interação com o simulador, visualização dos conteúdos dos registradores e memória, bem como depuração da aplicação.

Porém, da observação desses simuladores, onde levantou-se um conjunto de características importantes e necessárias para otimizar o aprendizado e o projeto integrado de sistemas computacionais, verificou-se que nenhum deles suporta conjuntamente todas essas características. Desta maneira, o presente trabalho propõe uma nova abordagem para estudos e projetos integrados de sistemas computacionais, através da combinação de: 1) simulador computacional, que apresenta as principais características dos simuladores do estado da arte; e 2) modelo de simulação de processador com ISA RISC-V, para atender a diferentes requisitos educacionais e ampliar o espectro de opções de projeto de sistemas computacionais.

## 3 Materiais e Métodos

O desenvolvimento do presente trabalho foi dividido em etapas. A primeira etapa consistiu na investigação da literatura sobre os ambientes de simulação computacionais, com suporte a RISC-V, conforme discutido na Seção 2.

A segunda etapa consistiu na formulação da abordagem apresentada no presente trabalho, que faz uso do simulador CompSim, contemplando suporte a RISC-V, para apoio aos processos de ensino-aprendizado de projetos de sistemas computacionais, conforme detalhada a seguir (Seção 4).

Por fim, a terceira etapa envolveu a avaliação e análise dos resultados, após o emprego em estudos de caso, da abordagem aqui apresentada. Para o levantamento e análise dos estudos de caso foram consideradas as seguintes questões de pesquisa:

1. Quais cenários de experimentos foram elaborados?
2. Como os experimentos foram realizados?
3. Como foi conduzida a coleta de dados no questionário de pesquisa?
4. Quais foram os critérios considerados para avaliação do suporte pedagógico do ponto de vista do estudante?
5. Quais foram os critérios considerados para avaliação do suporte pedagógico do ponto de vista do professor?
6. Quais foram os critérios considerados para avaliação do aprendizado do estudante?
7. Quais foram os critérios considerados para avaliação da experiência de uso do simulador?

# 4 Apresentação da Abordagem Proposta

A seção de trabalhos relacionados detalha a existência de lacunas nas soluções de simuladores computacionais presentes na literatura, uma vez que elas não contemplam todas as oito propriedades de um ambiente de simulação computacional, conforme ilustrado na Tabela 1. Assim, visando preencher tais lacunas, foi desenvolvida uma nova versão do simulador CompSim, com suporte à arquitetura RISC-V, conforme detalhado a seguir.

## 4.1 Simulador CompSim com Suporte à Arquitetura RISC-V

CompSim é um simulador computacional, que segue a abordagem de projetos baseados em plataforma (Keutzer, Newton, Rabaey, & Sangiovanni-Vincentelli, 2000), na qual há uma plataforma de hardware simulável customizável, que inclui os principais componentes do computador. São eles: 1) CPU: um processador RISC-V de 32-bits (padrão RV32I), que inclui 37 instruções para realização de operações de transferência de dados, lógicas e aritméticas e desvio de fluxo de programa. Possui ainda os seguintes submódulos: banco de registradores, contador de instrução, unidade de controle, e unidade lógica e aritmética; 2) Memória *Cache*: é utilizada para otimizar o desempenho dos programas (aproveitando as características de localidade espacial e temporal) e suporta diferentes tipos de configuração, como de técnicas de mapeamento e políticas de substituição; 3) Memória RAM: é utilizada para armazenamento de dados, instruções e pilha dos programas que serão executados no simulador; 4) Barramento: o simulador inclui um barramento, que permite a comunicação entre o processador, as memórias *cache* e RAM, e periféricos; e 5) Subsistema de Entrada/Saída: inclui uma interface padronizada com módulos de entrada/saída, que possibilita a comunicação do processador com periféricos dos tipos virtual (desenvolvidos em software e que simulam o comportamento dos respectivos periféricos reais) e físico (compostos de software, para integração com a plataforma virtual, e *hardware* físico, consistindo de um periférico real).

Conforme ilustrado na Figura 1, o simulador CompSim também conta com uma interface com os seguintes 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 montagem (*Assembly*), como número de linhas, teclas de atalho para recursos de edição (copiar/recortar/colar, desfazer/refazer, saltar para determinada linha, etc.), e destaque de palavras-chave (*syntax highlight*). Este componente está integrado a um montador (*Assembler*) que realiza análises léxica, sintática e semântica no código-fonte da aplicação, tradução deste para *bytecodes*, carregamento dos *bytecodes* na memória RAM, além de gerar um relatório do programa, que inclui a tabela de símbolos e *bytecodes* gerados; B) Processador: durante uma simulação, exibe os registradores do processador e respectivos valores assumidos; C) Memória *cache*: exibe as linhas de cache e respectivos conteúdos (instruções e dados); D) Memória RAM: exibe os conteúdos de todos os endereços do componente virtual memória RAM; e E) Componentes de controle de configuração e execução de simulação: inclui controles que permitem configurar o tempo total de simulação e a frequência de relógio de sistema (*clock*), iniciar, executar (em modos contínuo e passo a passo com visualização, ou contínuo sem visualização para maior desempenho), parar e reiniciar uma simulação.

O CompSim também inclui outros componentes de visualização, que são: 1) *Flow*: ferramenta de depuração, que mostra os endereços de memória, os respectivos *bytecodes* e as instru-

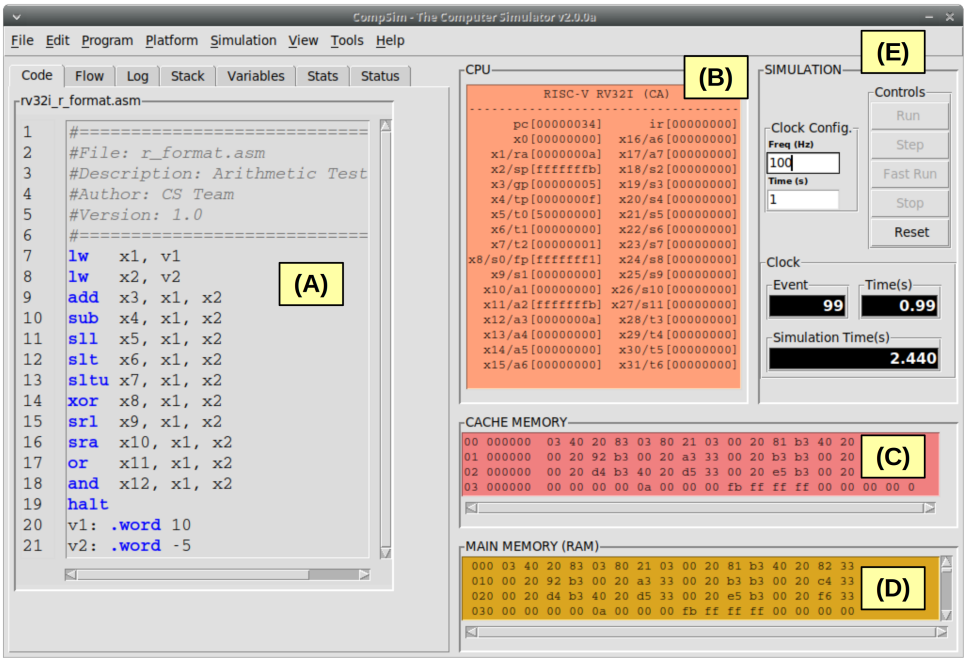



Figura 1: Interface gráfica do simulador CompSim.

ções, bem como permite acompanhar a sequência de execução das instruções do programa; 2) *Logs*: apresenta, durante uma simulação, os eventos e notificações gerados por todos os componentes da plataforma. O *profile* gerado durante uma simulação pode ser gravado em arquivo no disco; 3) Stack: permite acompanhar os estados da pilha do programa (dados adicionados e endereçamento do apontador de topo da pilha), durante uma simulação; 4) Variables: permite visualizar os valores assumidos pelas variáveis e estruturas de dados do programa, durante uma simulação; e 5) Stats: após o encerramento de uma simulação, apresenta gráficos estatísticos, que podem ser gravados em arquivo no disco, que sumarizam os eventos gerados pela CPU, Memórias RAM e Cache, bem como barramentos (e.g. são apresentados os tipos e quantidades de instruções executadas pela CPU, contabilizações de blocos lidos e escritos na memória RAM, totais de *Hits* e *Misses* de Cache, operações de entrada/saída realizadas no barramento de periféricos, entre outros).

O simulador CompSim conta ainda com outras ferramentas gráficas que podem ser utilizadas para apoio ao aprendizado, tais como um conversor de números inteiros entre bases numéricas (decimal, hexadecimal e binária), bem como uma ferramenta que permite consultar os respectivos códigos ASCII de diferentes caracteres; e, para auxiliar na acessibilidade de pessoas com necessidades visuais especificas é possível configurar o ambiente gráfico, por meio de ajustes: nas cores dos componentes gráficos e de visualização; e, no tamanho da fonte do editor de código.

A plataforma virtual de hardware do CompSim está integrada à plataforma aberta de prototipação Arduino, de forma que pode-se compor um ambiente híbrido (software e hardware) para o aprendizado e projetos de novos sistemas computacionais. Com a integração, o Arduino pode ser utilizado como um periférico físico da plataforma virtual de hardware, de forma que pode-se realizar simulação em que a plataforma virtual (software) interage com periféricos reais (hardware). Dentre os periféricos virtuais disponíveis, estão os componentes virtuais de Teclado, Vídeo e

VArduino, sendo que este último simula um Arduino UNO que pode ser conectado a diferentes componentes eletrônicos virtuais, tais como LEDs, Sensores, Display e atuadores. No CompSim, é possível ainda, com as devidas orientações, desenvolver novos periféricos virtuais ou integrar componentes eletrônicos físicos diversos para conexão com a plataforma virtual do CompSim.

Nesta proposta de abordagem, objetiva-se apresentar os resultados de utilização do CompSim em cenários pedagógicos de estudos de caso, visando oferecer uma nova abordagem de aprendizado e projeto de sistemas computacionais mais complexos, baseados na arquitetura RISC-V. Essa abordagem possui um alto potencial para criação de aplicações computacionais mais complexas, tais como processamento de alto desempenho, sistemas de computação científica, captura e decodificação de vídeo em tempo real, aplicações de aprendizagem de máquina (inteligência artificial), entre outras.

## 5 Apresentação e Análise dos Resultados

Para que os processos de ensino-aprendizagem de sistemas computacionais sejam otimizados é fundamental a alocação de carga horária para atividades e desenvolvimento de experimentos práticos, combinados a uma sólida base teórica, com o objetivo de desenvolver habilidades imprescindíveis ao exercício profissional, além de ratificar a compreensão dos diferentes conteúdos abordados teoricamente (Esmeraldo et al., 2019). Tais atividades podem ser desenvolvidas de diferentes formas no contexto dos projetos pedagógicos, com suas vantagens e desvantagens, considerando eficácia, eficiência e custos: 1) processos de simulação (virtualização); 2) ambientes laboratoriais especializados em montagens eletrônicas e robótica (manipulação física efetiva); 3) virtualização e manipulação física concomitantemente, ou seja, integrando processos de simulação com montagens físicas; e, 4) laboratórios remotos de aprendizagem que, atualmente, são objeto de estudo do estado da arte em experimentações em estruturas de cursos técnicos e engenharias (Hayashi & Hayashi, 2020).

Um cenário pedagógico ideal para apoiar o processo de ensino-aprendizagem de sistemas computacionais deve proporcionar ambientes para a utilização de simuladores de software e laboratórios de eletrônica bem equipados, que possam interagir de forma concomitante dentro de uma metodologia educacional, favorecendo a eficiência e eficácia desse processo em áreas de alta complexidade científica, técnica e tecnológica (Esmeraldo et al., 2019).

Após a utilização do simulador CompSim como suporte ao ensino-aprendizado em cursos de graduação e de pós-graduação, nas disciplinas de “Arquitetura e Organização de Computadores” e “Infra-Estrutura de Hardware”, foram realizados questionários para coleta dos dados apresentados nesta seção, considerando as questões de pesquisa apresentadas na Seção 3. São elas descritas a seguir com os respectivos resultados.

### 5.1 Quais cenários de experimentos foram elaborados?

Dentre as respostas dos professores, relatadas em questionário, destacam-se: atividades de alocação de estruturas de dados em memória, operações aritméticas/lógicas, acessos avançados à memória, definição de estruturas de desvio de fluxo de programa, modularização de programas e análise de desempenho.

Outras atividades citadas foram: para entendimento da linguagem de montagem (*Assembly*), para exploração do uso de registradores e para exploração de configurações de memória Cache; bem como, aquelas relacionadas a blocos básicos de um processador/microcontrolador e subsistemas de entrada/saída.

### 5.2 Como os experimentos foram realizados?

Todos os experimentos foram realizados remotamente. Ou seja, os estudantes tinham acesso às instruções dos cenários de projetos de sistemas computacionais, de forma que eles realizavam os experimentos da sua própria casa, com a utilização de computadores próprios, com o simulador CompSim devidamente instalado.

É importante destacar que os estudantes tiveram suporte remoto de monitores de disciplina, que dirimiam as dúvidas em sessões em salas de conferências web (Google Meet). Utilizaram também os aplicativos de mensagens instantâneas Whatsapp e Telegram para uma comunicação mais ágil entre estudantes e os monitores.

Como materiais complementares, além do simulador CompSim, foram utilizados periféricos virtuais que simularam os periféricos de Vídeo, Teclado e Arduino. Também foram disponibilizados (no site do CompSim) aos estudantes alguns materiais de referência, tais como *datasheet* (um documento com a relação de instruções do processador RISC-V, com respectivas sintaxes de uso e exemplos ilustrativos), exemplos de código-fonte, ilustrando o uso e combinação de uso das instruções do processador RISC-V, e um mapa com os endereços (portas) para comunicação com periféricos.

### 5.3 Como foi conduzida a coleta de dados no questionário de pesquisa?

Para a aplicação dos questionários, foram criados dois formulários, um direcionado aos professores e outro direcionado aos estudantes.

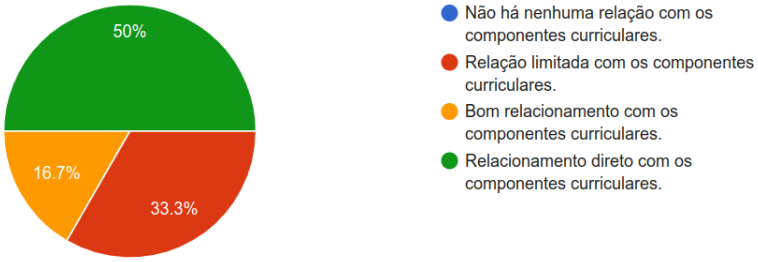
Os formulários foram criados e disponibilizados por meio da plataforma Google Forms, ao fim das práticas didáticas com o simulador. Os experimentos foram realizados por três turmas com média de 30 estudantes por turma, onde no total 53 estudantes e 3 professores participaram voluntariamente da pesquisa, respondendo os respectivos formulários.

### 5.4 Quais foram os critérios considerados para avaliação do suporte pedagógico do ponto de vista do estudante?

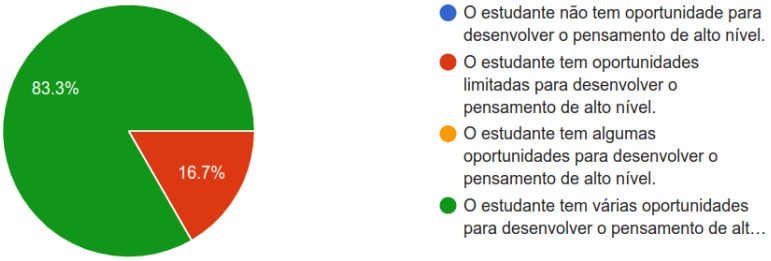
A avaliação do suporte pedagógico por estudantes considerou as seguintes dimensões: Se houve relação entre as práticas realizadas com o simulador e os componentes curriculares da disciplina; Se o uso do simulador permitiu a construção de novos conhecimentos, através da análise, avaliação e síntese (pensamento de alto nível); Se o simulador foi efetivo no ensino do conteúdo desejado, em uma forma interativa e que os estudantes fossem atraídos pelo aprendizado; e se o uso do simulador permitiu que a taxa de aprendizado fosse apropriada às necessidades individuais.

Os gráficos ilustrados na Figura 2 mostram os percentuais das respostas dos estudantes para cada uma das dimensões avaliadas. O gráfico da Figura 2 (a) mostra que 65% dos estudantes concordaram que os conteúdos trabalhados no simulador possuem uma boa relação com os conteúdos

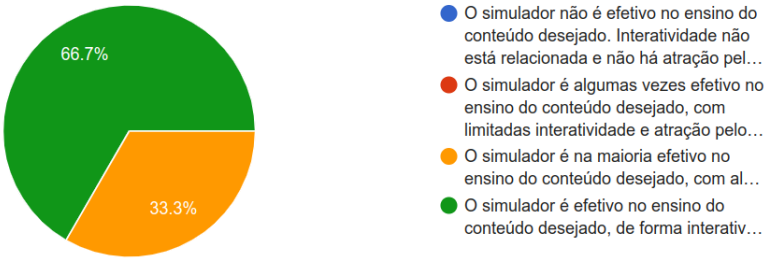
curriculares da disciplina. Além disso, mais de 80% dos estudantes afirmaram que o uso de simulação trouxe várias oportunidades para o desenvolvimento do pensamento de alto nível (Figura 2 (b)). Por fim, todos os estudantes afirmaram que a dinâmica interativa permitiu motivá-los e que isso trouxe efetividade no ensino (Figura 2 (c)); bem como foi possível obter um certo nível de ajuste da taxa de aprendizado as suas necessidades individuais (Figura 2 (d)).



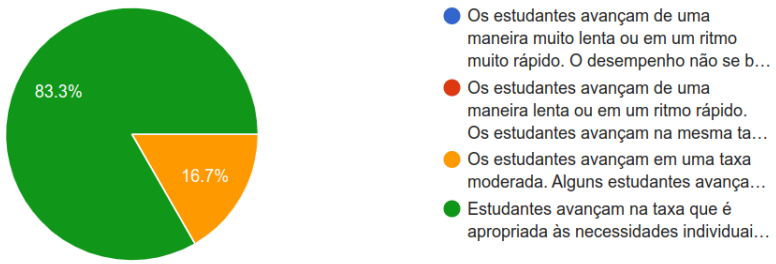

(a) Conteúdo curricular.




(b) Pensamento de alto nível.




(c) Efetividade no ensino.




(d) Desempenho do estudante.

Figura 2: Avaliação de aprendizado por estudantes.

### 5.5 Quais foram os critérios considerados para avaliação do aprendizado do estudante?

A avaliação do aprendizado dos estudantes se deu de diferentes formas, entre as turmas avaliadas. Os professores participantes da pesquisa, e que responderam o questionário, informaram que utilizaram a metodologia de Aprendizado Baseado em Projetos (ABP) (Allen, Donham, & Bernhardt, 2011) em suas disciplinas, sendo que, além de ABP também fizeram uso de outros instrumentos avaliativos, como listas de exercícios, projeto de disciplina e prova, visando direcionar o aprendizado e permitir avaliar o desempenho dos estudantes.

Conforme ilustrado na Figura 3, a maioria dos estudantes (66,7%) informaram que os conteúdos trabalhados nesses recursos estavam de acordo com os objetivos da disciplina e com as habilidades que deveriam ser desenvolvidas pelos estudantes. Do total de estudantes, 50% concordou que foi facilmente possível identificar os conteúdos e habilidades que estavam sendo aferidos nas avaliações.

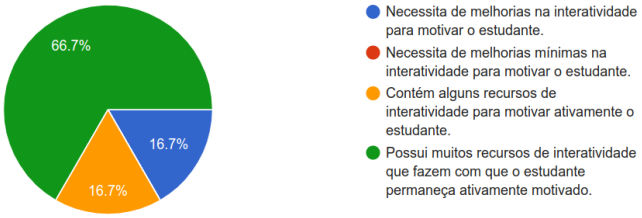

Figura 3: Componentes de avaliação.

### 5.6 Quais foram os critérios considerados para avaliação da experiência de uso do simulador?

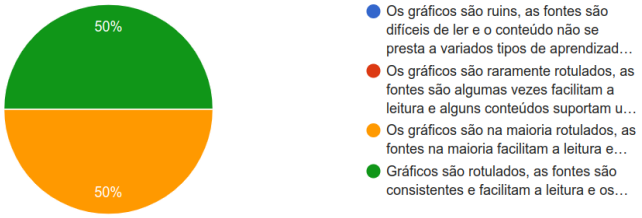
Em (Molich & Nielsen, 1990), são apresentadas orientações para construção de boas interfaces de interação humano-computador, e em (Nielsen & Molich, 1990) é descrito um método de avaliação dessas interfaces. Nesses trabalhos, são apresentados princípios de usabilidade, dentre os quais são abordadas linguagens amigáveis, e facilidade de uso da interface, visando diminuir a carga de informação que usuários devem memorizar. Nesse sentido, no presente trabalho, são adaptados tais princípios conforme detalhado a seguir.

A experiência de uso do simulador foi avaliada pelos estudantes em três dimensões, sendo elas: Interatividade - se havia relação do simulador com a atividade do usuário; Acessibilidade - se as informações gráficas estavam devidamente rotuladas, se fontes estavam consistentes e fáceis de ler, e se foram empregados diferentes estilos de aprendizagem e níveis de habilidade; e, Amigabilidade - se o *design* do simulador era limpo, fácil de navegar e se dispunha de meios para auxiliar o estudante no aprendizado.

A Figura 4 ilustra as respostas da avaliação da experiência de uso do simulador pelos estudantes. Na Figura 4 (a) é possível observar que a maioria dos estudantes (83,4%) afirmou que os recursos de interatividade do simulador permitiram estimulá-los ativamente para a realização dos experimentos propostos. Já no gráfico da Figura 4 (b), observa-se que todos os estudantes afirmaram que os recursos gráficos permitiram uma boa experiência na visualização das informações e absorção dos conteúdos, sob uma variedade de tipos de aprendizagem e níveis de habilidades.




(a) Interatividade.




(b) Acessibilidade.


(c) Amigabilidade.

Figura 4: Avaliação da experiência de uso do simulador por estudantes.

Ademais, alguns estudantes relataram que, por possuírem algum tipo de deficiência visual, foi importante o recurso de configuração das cores do ambiente de simulação e do tamanho da fonte do editor de código do simulador para que eles pudessem estudar, codificar e realizar as simulações com maior conforto visual e por períodos mais longos. Por fim, conforme ilustrado na Figura 4 (c), 67,7% dos estudantes consideraram a interface gráfica do simulador limpa e que a mesma pode ser utilizada adequadamente para o benefício do processo instrucional.

Em resposta ao questionário, por meio de texto livre, os professores afirmaram que os recursos gráficos (considerados, por eles, fáceis de utilizar e amigáveis) foram muito importantes para que os estudantes pudessem acompanhar de uma forma visual e interativa toda a dinâmica de projeto e simulação de sistemas computacionais. Eles ainda afirmaram que os materiais complementares e atividades propostas também foram um diferencial, pois com elas foi possível direcionar e motivar os estudantes a buscarem conhecimentos mais aprofundados para concretizarem as soluções dos problemas propostos. Um dos professores afirmou que o projeto de disciplina também agregou ao processo de aprendizado, desde que o desafio aos estudantes consistiu em propor uma nova aplicação computacional. Os resultados dessa atividade mostraram que, além de estimular a criatividade, os estudantes puderam se concentrar, em equipes, para debater e desenvolver as soluções. O professores destacaram ainda, como pontos diferenciais do uso do simulador CompSim, a possibilidade de realizar simulações de um sistema computacional completo e de extensão de

suas funcionalidades, por meio da integração e interação com hardware real.

Por fim, os docentes apontaram as seguintes sugestões de melhorias: (i) o fato de necessitar de computador com GNU/Linux ou Microsoft Windows para executar o simulador, fez com que algumas das atividades não fossem produtivas e que o aprendizado não fosse contínuo, dada a indisponibilidade de computador em algumas das residências dos estudantes. Desta maneira, foi sugerida a criação de versões do simulador CompSim para web ou dispositivos móveis; (ii) o simulador necessita de aprimoramentos visando melhorar sua estabilidade em um determinado sistema operacional; e, (iii) a adição de novos recursos, tais como permitir uma melhor estruturação dos programas criados em linguagem de montagem (*Assembly*) e a inclusão de suporte à linguagem de programação C, pois poderiam aumentar significantemente o interesse no uso da ferramenta.

## 6 Considerações Finais

Este trabalho apresentou uma proposta de abordagem de ensino-aprendizagem, que faz uso do simulador CompSim com suporte a arquitetura RISC-V, para apoiar o aprendizado dos conceitos e desenvolvimento de habilidades práticas em projetos de sistemas computacionais. Para avaliar a abordagem proposta, foram aplicados questionários onde estudantes e professores puderam avaliar o suporte pedagógico e a experiência de uso do simulador em disciplinas relacionadas a projetos de sistemas computacionais em diferentes cursos.

Com a análise dos dados dos questionários, foi possível constatar que a utilização do CompSim, para apoio às práticas pedagógicas no processo de ensino-aprendizagem de projetos de sistemas computacionais, auxiliou: (i) em atividades de alocação de estruturas de dados em memória, operações aritméticas/lógicas, acessos avançados à memória, definição de estruturas de desvio de fluxo de programa, como modularizar programas e análise de desempenho; (ii) no entendimento da linguagem de baixo nível (*Assembly*), para exploração do uso de registradores e para exploração de configurações de memória *Cache*; bem como, naquelas relacionadas à construção dos blocos básicos de um processador/microcontrolador e à compreensão dos processos de comunicação em subsistemas de entrada/saída. Além disso, a análise dos dados dos questionários revelou que, dentre os aspectos pedagógicos favorecidos pela utilização do CompSim, (iii) mais de 65% dos estudantes concordaram que os conteúdos trabalhados no simulador possuem uma boa relação com os conteúdos curriculares da disciplina; (iv) mais de 80% dos estudantes afirmaram que o uso de simulação trouxe várias oportunidades para o desenvolvimento do pensamento de alto nível. Por fim, todos os estudantes afirmaram que a dinâmica interativa permitiu motivá-los e que isso trouxe efetividade no ensino; bem como, (v) foi possível obter um certo nível de ajuste da taxa de aprendizado as suas necessidades individuais.

Como indicações de trabalhos futuros, será possível incluir novos recursos ao CompSim: (a) de plataformas, tais como variadas arquiteturas de processadores (e.g. x86 e ARM), novos componentes de plataforma (e.g. memória *cache* D/I-L1, L2, L3, barramentos e periféricos), e integração com diferentes plataformas físicas, além da Arduino, visando propiciar aos estudantes novos cenários de projeto e suporte de implementações físicas de sistemas computacionais; (b) de software, tais como adicionar recursos que simplifiquem a codificação de novas aplicações em linguagem de baixo nível (*Assembly*), como os presentes em Ambientes Integrados de Desen-

volvimento (IDEs) (e.g. *Autocomplete* e *Refactoring*) e suporte à linguagem C, com objetivo de permitir a codificação e execução de aplicações mais complexas (e.g. Compiladores e Sistemas Operacionais), fazendo com o que o uso do simulador CompSim permeie entre diferentes disciplinas e, desta maneira, passe a oferecer suporte ao aprendizado multidisciplinar; (c) gráficos, visando ampliar a gama de recursos disponíveis para simplificar o aprendizado em projetos de sistemas computacionais (e.g. Diagramas de onda/tempo para comunicação de barramento, de consumo de memória e de organização dos componentes da plataforma) e otimizar a experiência de uso do simulador; e (d) educacionais: para otimizar os processos de ensino-aprendizagem em projetos de sistemas computacionais, pela oferta de variados materiais complementares (e.g. tutoriais, *datasheets*, exemplos de códigos-fonte e video-aulas), pelo contato com laboratórios de hardware e eletrônica profissionais, e pela de integração do CompSim com laboratórios remotos.

Além disso, planeja-se realizar novos experimentos com o simulador aqui proposto em diferentes turmas, em um cenário pós-pandemia, abrangendo assim um número maior de estudantes, professores e cursos, onde os novos resultados poderão indicar maior consistência e expressividade no apoio ao processo de ensino-aprendizado de projetos computacionais, bem como melhorias na ferramenta proposta.

## Edição Especial: Metodologias de ensino e ferramentas tecnológicas de suporte para o ensino remoto no Pós-Pandemia

Esta publicação compõe a edição especial "Metodologias de ensino e ferramentas tecnológicas de suporte para o ensino remoto no Pós-Pandemia", conduzida pelo Editor convidado Prof. Dr. Marciel Aparecido Consani (Universidade de São Paulo).

## Referências

ACM, & Society, I. C. (2013). *Computer science curricula 2013: Curriculum guidelines for undergraduate degree programs in computer science*. Association for Computing Machinery. [GS Search]

Allen, D. E., Donham, R. S., & Bernhardt, S. A. (2011). Problem-based learning. *New directions for teaching and learning*, *2011*(128), 21–29. [GS Search]

Awedh, M., & Mueen, A. (2015). Teaching computer organization using field programmable gate array: An incremental approach. *Asian Journal Of Advanced Basic Sciences*, *4*, 5–11. [GS Search]

Balamuralithara, B., & Woods, P. C. (2009). Virtual laboratories in engineering education: The simulation lab and remote lab. *Computer Applications in Engineering Education*, *17*(1), 108–118. [GS Search]

Black, M. (2016). Export to arduino: a tool to teach processor design on real hardware. *Journal of Computing Sciences in Colleges*. [GS Search]

Esmeraldo, G. A. R. M., Mendes, C. S. R., Cartaxo, L. F., & Lisboa, E. B. (2019). Apoio ao aprendizado em arquitetura e organização de computadores: Um estudo comparativo entre simuladores computacionais. *Revista Tecnologias na Educação*, *31*(1), 1–17. [GS Search]

García, I. A., Pacheco, C. L., & Garcia, J. (2014). Enhancing education in electronic sciences using virtual laboratories developed with effective practices. *Computer Applications in*

*Engineering Education*, *22*(2), 283–296. [GS Search]

Hayashi, V. T., & Hayashi, F. H. (2020). Labead: Laboratório eletrônico de ensino à distância durante o distanciamento social. In *Anais do v congresso sobre tecnologias na educação* (pp. 21–30). [GS Search]

Keutzer, K., Newton, A. R., Rabaey, J. M., & Sangiovanni-Vincentelli, A. (2000). System-level design: Orthogonalization of concerns and platform-based design. *IEEE transactions on computer-aided design of integrated circuits and systems*, *19*(12), 1523–1543. [GS Search]

Molich, R., & Nielsen, J. (1990). Improving a human-computer dialogue. *Commun. ACM*, *33*(3), 338–348. [GS Search] doi: 10.1145/77481.77486

Neto, A., Borges, J. d. S., & Silva, G. (2017). 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)* (pp. 123–128). [GS Search]

Nielsen, J., & Molich, R. (1990). Heuristic evaluation of user interfaces. In *Proceedings of the sigchi conference on human factors in computing systems* (p. 249–256). New York, NY, USA: Association for Computing Machinery. [GS Search] doi: 10.1145/97243.97281

Nikolic, B., Radivojevic, Z., Djordjevic, J., & Milutinovic, V. (2009). A survey and evaluation of simulators suitable for teaching courses in computer architecture and organization. *IEEE Transactions on Education*, *52*(4), 449–458. [GS Search]

Penna, P., & Freitas, H. C. (2013). Análise e avaliação de simuladores de sistemas completos para o ensino de arquitetura de computadores. *Int. Journal of Computer Architecture Education*, *2*(1), 13–16. [GS Search]

Ray, P. P. (2018). A survey on internet of things architectures. *Journal of King Saud University-Computer and Information Sciences*, *30*(3), 291–319. [GS Search]

RISC-V. (2022a). *RISC-V.* Retrieved from https://riscv.org/

RISC-V. (2022b). *RISC-V Exchange: Available Software.* Retrieved from https://riscv.org/exchange/?_sft_exchange_category=software&_sfm_exchange_software_type=Simulators

Sartor, M., Soares, T. T. M. S., & Daniel, M. (2020). Building a microprocessor architecture at computer engineering undergraduate courses. *International Journal of Advanced Engineering Research and Science*, *7*(7), 036–049. [GS Search]

Tan, H. S., Tan, K. C., Fang, L., May, L. W., & Koh, C. (2009). Using simulations to enhance learning and motivation in machining technology. In *Proceedings of the 17th international conference on computers in education* (pp. 864–871). [GS Search]

Uribe, M. D. R., Magana, A. J., Bahk, J.-H., & Shakouri, A. (2016). Computational simulations as virtual laboratories for online engineering education: A case study in the field of thermoelectricity. *Computer Applications in Engineering Education*, *24*(3), 428–442. [GS Search]

William, S. (2010). *Computer organization and architecture designing for performance eighth edition.* Pearson. [GS Search]

Xavier, M., Rodrigues, J., & Júnior, O. (2011). Simuladores de memória cache, um estudo comparativo direcionado ao ensino. In *Workshop sobre educação em arquitetura de computadores (weac 2011)* (pp. 7–12). [GS Search]

Zorzo, A. F., Nunes, D., Matos, E., Steinmacher, I., de Araujo, R. M., Correia, R., & Martins, S. (2017). *Referenciais de formação para os cursos de graduação em computação.* Sociedade Brasileira de Computação - SBC. [GS Search]