Google Colab para Ensino de Computação

Ricardo Ferreira[1], Michael Canesche[2], Jerônimo Penha[3]
ricardo@ufv.br,michael.canesche@gmail.com,jeronimo.penha@ufv.br
[1]Universidade Federal de Viçosa, Viçosa
[2]Universidade Federal de Minas Gerais, Belo Horizonte
[3] CEFET-MG, Campus Leopoldina

RESUMO

Este minicurso teórico-prático tem como objetivo a introdução ao uso do Google Colab como uma ferramenta de ensino à distância e/ou atividades práticas síncronas ou assíncronas em aula presencial ou remotas na Educação em Computação. Tradicionalmente, o Google Colab vem sendo usado no ensino de inteligência artificial e aprendizado de máquina. O minicurso irá apresentar aos participantes vários exemplos, ideias e metodologias para uso em qualquer disciplina. Mostrando como usar outras linguagens além de Python, como inserir outras ferramentas de ensino, criação de comandos mágicos para simplificar o uso, interfaces interativas, interação com *smartphones* com o uso da biblioteca Gradio. Os exemplos utilizados no minicurso ilustrarão o ensino de arquitetura de computadores, programação de alto desempenho com GPU, linguagens formais e autômatos, grafos, linguagens de programação e programação de computadores.

CCS CONCEPTS

• **Social and professional topics** → Computing education.

PALAVRAS-CHAVE

Google Colab, Jupyter Notebooks, Ensino Remoto

1 JUSTIFICATIVA

Este minicurso será ministrado no formato teórico-prático. A inteligência artificial e o aprendizado de máquina são dois tópicos em evidência nos últimos anos. Podemos destacar que além dos avanços na área, o uso de *Jupyter Notebook* localmente ou executando em navegadores democratizou o acesso às técnicas e ferramentas, além de gerar documentação e experimentos que pode ser validados e replicados pelos leitores [4, 13]. Apesar de ser uma plataforma genérica, tanto o *Jupyter notebook* e a biblioteca *Ipython* [12], existem poucos exemplos de uso em outros domínios de ciência da computação.

Este minicurso irá ilustrar como os recursos já disponíveis podem ser utilizados, personalizados e adaptados para outro contexto [3]. Por exemplo, no ensino de sistemas lógicos é necessário a manipulação de expressões, descrever e simular circuitos e máquinas de estados. Na simulação pode-se se usar a linguagem Verilog. Mostraremos como adaptar estas ferramentas para o contexto do *Jupyter Notebook* com criação de comandos mágicos que facilitam o reuso. Outros exemplos em outros domínios serão trabalhados no minicurso, mostrando como é possível generalizar e fazer uso para ensino remoto ou presencial. Esta abordagem simplifica a instalação de ferramentas, pois já está tudo integrado e permite até o uso em *tablet* ou celulares. Além disso, os laboratórios ficam documentados e interativos. Permite também acesso a um sistema de alto desempenho com GPU de forma gratuita.

EduComp'23, Abril 24-29, 2023, Recife, Pernambuco, Brasil (On-line)

2 SUMÁRIO ESTENDIDO

Nesta seção são descritos os quatro módulos principais do minicurso: introdução, ferramentas, acesso com celulares e automação de experimentos.

2.1 Introdução

Estrutura básica do *Jupyter notebook*, como adicionar exemplos para compilar em C, C++, JAVA e outras linguagens, opções de formas interativas do *Ipython* [12] e criação de comandos mágicos para personalizar domínios de ensino.

2.2 Instalando e usando Ferramentas

Mostraremos como instalar um compilador e ferramentas de simulação com diversos formatos de interatividade: comandos mágicos, botões, texto ou linha de comando. O objetivo é simplificar o acesso ao uso, instalação e configuração de ferramentas de ensino. Como casos de uso apresentamos a instalação de Verilog, criação de comandos para mágicos, desenvolvimento de novas ferramentas para simplificar a visualização dos dados. Ilustraremos como instalar uma ferramenta de linha de comando e simplificar sua interface para exercícios, além de mostrar como usar uma ferramenta que precise de mais área em disco e de instalação mais complexa, usando recursos de pré-instalação e compartilhando na nuvem. Além dos exemplos de sistemas lógicos, abordaremos ferramentas para ensino de algoritmos, autômatos finitos, linguagens formais e grafos.

2.3 Acesso com Telefones Celulares

Apresentaremos a ferramenta *Gradio* [1] que foi desenvolvida para simplificar a entrada e saída de dados. O objetivo do *Gradio* é criar interfaces simples e intuitivas para uso de demonstrações. Os exemplos da ferramenta estão focados em inteligência artificial. Mostraremos que a ferramenta é versátil e pode ser usada no ensino interativo, com exercícios e demonstrações em sala ou remoto, onde o estudante pode fazer execução de exemplos, pequenas edições, ajuste de parâmetros no aprendizado de várias disciplinas. Ilustraremos como usar uma GPU, autômatos finitos, equações booleanas

[1]https://gradio.app/

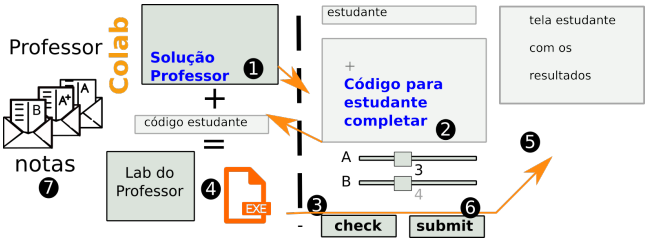

Figura 1: Estrutura dos laboratórios.

e circuitos simples, grafos e linguagens de programação. Mostraremos também como é possível fazer uma atividade avaliativa e coleta de dados.

A figura 1 ilustra a ideia de uma das propostas onde o estudante tem acesso com uma interface simples para fazer atividade, podendo usar um celular ou *tablet*. Primeiro, o professor conduz o laboratório e dispara o processo (1). O estudante completa o código (2) e faz a submissão (3), o *colab* do professor irá fazer a correção (4), enviando a resposta para o estudante (5), assim que o estudante fizer a submissão (6), o *colab* do professor registra as notas (7).

2.4 Experimentos e Visualização

Mostraremos como escrever códigos para automatizar a realização de experimentos, coleta e visualização dos dados. Por exemplo, suponha o ensino de ordenação. O estudante pode escolher alguns parâmetros como: tamanho mínimo e máximo do vetor, escolha do algoritmo e escolha da plataforma (GPU ou CPU), escolha da linguagem. O código será executado para uma combinação de parâmetros e os resultados serão visualizados com gráficos. A visualização pode ser no *Google Colab* ou através da interface *Gradio* no telefone celular, *tablet* ou *notebook*.

3 PÚBLICO-ALVO

O público alvo são professores, alunos de graduação e pós-graduação da computação ou afins. Podem atuar em qualquer área da computação para qual gostariam de desenvolver ferramentas interativas de ensino ou demonstrações de tópicos. Desejável com conhecimentos básicos da linguagem *Python* que será mais utilizada nos exemplos. Não é necessário conhecimentos de redes de computadores nem programação para celulares para fazer uso do *Gradio*, apenas o básico de *Python*. Os exemplos podem ser aplicados em cursos básicos para ensino de programação em qualquer faixa etária até exemplos bem específicos para destacar uma área especializada do conhecimento ou apenas demonstrar uma aplicação.

4 BIOGRAFIA DOS AUTORES

(1) **Ricardo Ferreira:** Professor Ciência da Computação da UFV desde 1992, Doutor pela UCL, Bélgica, autor de diversos artigos de ensino com ferramentas e simuladores [1–3, 6–11] e ministrou minicursos nos eventos SBESC2018, WSCAD2019 [5], WSCAD2020,

(2) **Jerônimo Penha:** Professor do CEFET-MG, Mestre e doutorando em Computação pela UFV, autor de diversos artigos de ensino com ferramentas e simuladores [1, 2, 11].

(3) **Michael Canesche:** Mestre pela UFV e doutorando em Computação pela UFMG, autor de diversos artigos de ensino com ferramentas e simuladores [3, 8–10].

5 ORGANIZAÇÃO DO MINICURSO

O minicurso terá 3 horas de duração, estarão divididas: 10 minutos para apresentação inicial e algumas demonstrações de uso; 20 minutos para introdução ao uso do *Colab*, 30 minutos para as demonstrações de instalação de uso de ferramentas, 30 minutos para o uso do *Gradio* e acesso com telefones celulares, 30 minutos para experimentos com automatização das execuções, intercalados com 45 minutos para a parte prática com interação dos alunos/professores, 15 minutos de intervalo (5 minutos a cada 55 minutos de aula). Idioma será português, como exemplos de *Colab* em inglês e português.

6 INFRAESTRUTURA E MATERIAIS NECESSÁRIOS

O objetivo do minicurso é simplificar a infraestrutura para ensino, sendo necessário apenas acesso à internet e um navegador com uma conta Google. Pode usar MAC, Linux ou Windows.

Todo o material do curso estará disponível através de um Colab Ìndice (clique aqui) de acesso público visando a divulgação e compartilhamento dos exemplos para desenvolvimento colaborativo.

REFERÊNCIAS

[1] Hector Perez Baranda, Jeronimo Costa Penha, and Ricardo Ferreira. 2018. Implementaç ao de um Preditor de Desvio no MIPS 5 Estágios. *International Journal of Computer Architecture Education.*

[2] Jeronimo C. Penha, Lucas B. Silva, Jansen M. Silva, Kristtopher K Coelho, Hector P. Baranda, José Augusto M. Nacif, and Ricardo S. Ferreira. 2019. ADD: Accelerator Design and Deploy-A tool for FPGA high-performance dataflow computing. *Concurrency and Computation: Practice and Experience* 31, 18, e5096.

[3] Michael Canesche, Lucas Bragança, Omar Paranaiba Vilela Neto, Jose A Nacif, and Ricardo Ferreira. 2021. Google colab cad4u: Hands-on cloud laboratories for digital design. In *International Symposium on Circuits and Systems (ISCAS).*

[4] Alan Davies, Frances Hooley, Peter Causey-Freeman, Iliada Eleftheriou, and Georgina Moulton. 2020. Using interactive digital notebooks for bioscience and informatics education. *PLOS Computational Biology* 16, 11, e1008326.

[5] Ricardo Ferreira, Salles Viana Gomes de Magalhães, and José AM Nacif. 2019. Métricas e Números: Desmistificando a Programação de Alto Desempenho em GPU. *Sociedade Brasileira de Computação.*

[6] Ricardo Ferreira, Jose Nacif, Salles Magalhaes, Thales de Almeida, and Racyus Pacifico. 2015. Be a simulator developer and go beyond in computing engineering. In *2015 IEEE Frontiers in Education Conference (FIE).* IEEE, 1–8.

[7] Ricardo S Ferreira, Antonio Carlos S Beck, Luigi Carro, Andre Toledo, and Aroldo Silva. 2005. A java framework to teach computer architecture. In *IFIP International Working Conference on Computer-Aided Learning.* Springer, 25–35.

[8] Fernando Passe, Lucas Bragança, Michael Canesche, Felippe Cathoud, José Nacif, and Ricardo Ferreira. 2020. Plain: Ferramenta para Desenvolvimento de Aceleradores para Overlays em FPGA na Nuvem em Tempo de Execução. In *XXI Simpósio em Sistemas Computacionais de Alto Desempenho.*

[9] Fernando Passe, Michael Canesche, Omar Paranaiba Vilela Neto, Jose A Nacif, and Ricardo Ferreira. 2020. Mind the gap: Bridging verilog and computer architecture. In *International Symposium on Circuits and Systems (ISCAS).*

[10] Fernando Ferreira Passe, Vanessa Cristiny Rodrigues Vasconcelos, Michael Canesche, and Ricardo Ferreira. 2017. Perspectivas para o uso do Node-Red no Ensino de IoT. *International Journal of Computer Architecture Education.*

[11] Jerônimo Penha, Geraldo Fontes, and Ricardo Ferreira. 2016. MIPSFPGA-Um simulador mips incremental com validaçao em fpga. *International Journal of Computer Architecture Education.*

[12] Cyrille Rossant. 2013. *Learning IPython for interactive computing and data visualization.* Packt Publishing Ltd.

[13] Adam Rule, Aurélien Tabard, and James D Hollan. 2018. Exploration and explanation in computational notebooks. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems.* 1–12.