

Ideias para uso do Google Colab no Ensino de Computação

Ricardo Ferreira¹, Michael Canesche², Jose Nacif¹
ricardo@ufv.br, michael.canesche@gmail.com, jnacif@ufv.br

¹ Universidade Federal de Vicosa

² Universidade Federal de Minas Gerais

RESUMO

O *Google Colab* vem sendo amplamente utilizado pela comunidade de programadores. Porém, a maior parte dos usuários fazem uso de laboratórios para ensino e pesquisa nos domínios de aprendizado de máquina e inteligência artificial. Este trabalho traz novas ideias para explorar a grande diversidade de recursos que o *Colab* pode oferecer mas são poucos explorados. Através do uso do próprio *Colab* criamos uma série de atividades para a comunidade docente desenvolver laboratórios de aula. Além de *Python*, mostraremos que outras linguagens podem ser incluídas, assim como é possível instalar ferramentas de linha de comandos de forma amigável. Mostraremos como gerar banco de questões para os estudantes realizarem autoavaliações dentro do ambiente e também recursos para que os professores possam monitorar o uso do ambiente.

PALAVRAS-CHAVE

Google Colab, Laboratório de Ensino na Nuvem, Jupyter Notebook

1 INTRODUÇÃO

O *Google Colaboratory* ou *Google Colab* é uma plataforma gratuita que possibilita o uso de *Jupyter notebooks* [7] utilizando navegadores de internet. O *Colab* permite a inclusão de documentação e código em um mesmo ambiente, apresentando, assim, diversas vantagens. Primeiro, o código executa na nuvem sem a necessidade de roteiros de instalação. Segundo, o *Colab* é integrado ao *Google Drive* e *Github*, permitindo acesso fácil a dados e códigos compartilhados. Terceiro, o ambiente pode ser usado no ensino de conteúdo de diversas áreas como, por exemplo, Biologia, Física e Engenharias. Entretanto, estes ambientes vêm sendo mais utilizados apenas pela comunidade de aprendizado de máquina e inteligência artificial.

A proposta deste trabalho é ilustrar vários recursos disponíveis no *Colab* que são pouco explorados para ensino. Este trabalho dá continuidade a um trabalho anterior [1] que ilustra vários recursos do *Colab* aplicados ao ensino de arquitetura de computadores e GPUs [2]. A documentação e os exemplos de uso dos recursos será feita usando uma série de *Colabs* [3]. A seção 2 mostra como incluir linguagens de programação e ferramentas de linha de comando. A seção 3 mostra os recursos para autoavaliações. A seção 4 apresenta recursos de monitoramento dos laboratórios para que o professor possa acompanhar o perfil de acesso dos alunos. Finalmente, a seção 5 discute as considerações finais, pois o objetivo é proporcionar

a comunidade de ensino todos os recursos integrados no navegador, permitindo ao docente o monitoramento e acesso a facilitadores para desenvolver material.

2 LINGUAGENS E LINHAS DE COMANDO

O *Jupyter* tem células de texto e de código. Nas células de texto temos os recursos da linguagem *Markdown*. A célula de código executa *Python* ou comandos Linux usando como prefixo os caracteres `!` ou `%`. Comandos mágicos podem ser criados para executar outras linguagens. Exemplos para Verilog e CUDA foram ilustrados em [1]. Neste trabalho apresentamos um *Colab* para o qual adicionamos facilidades para criar comandos mágicos para inclusão de outras linguagens. Além disso, criamos também uma classe para generalizar a instalação de ferramentas de linha de comando com interface de botões, que auxilia também no desenvolvimento de ensino com simuladores [4, 5].

3 PERGUNTAS E RESPOSTAS

A biblioteca *Ipython* oferece vários recursos para elaborar questões e formulários. Porém, existem poucos exemplos que mostram a diversidade e como usá-los para fazer avaliações ou autoavaliações dentro dos laboratórios para oferecer um retorno ao estudante. Elaboramos um *Colab* para ilustrar como o professor pode incluir questões e/ou gerar um banco de questões aleatórias para proporcionar vários recursos de autoavaliação ao estudante.

4 MONITORAMENTO DO USO

Cada estudante executa suas atividades de forma assíncrona. O professor pode fazer a correção das atividades dos *colabs*, porém esta tarefa pode ser bem custosa em termos de tempo. Existem algumas ferramentas de autocorreção de células de código *Python* apenas para *Jupyter notebooks* [6]. Para solucionar este problema, sugerimos o uso de mensagens MQTT [8]. MQTT (*Message Queuing Telemetry Transport*) é um protocolo simples e leve de troca de mensagens que é fundamentado no modelo Publicador-Assinante. Criamos algumas funções para encapsular a comunicação e dois *templates*: *Colab Professor* e *Colab Aluno*. O professor define os tópicos que são usadas para gerenciar a comunicação. O *colab Professor* irá receber e armazenar informações da execução das questões dos *Colab* Estudantes.

5 CONSIDERAÇÕES FINAIS

As ideias apresentadas aqui podem ser avaliadas nos *Colabs* [3]. Os *Colabs* são exemplos que os professores podem replicar e adaptar para suas atividades e estarão em constante desenvolvimento. Os exemplos estarão sendo avaliados pelos alunos da Universidade Federal de Viçosa no ano letivo de 2023.

Fica permitido ao(s) autor(es) ou a terceiros a reprodução ou distribuição, em parte ou no todo, do material extraído dessa obra, de forma verbatim, adaptada ou remixada, bem como a criação ou produção a partir do conteúdo dessa obra, para fins não comerciais, desde que sejam atribuídos os devidos créditos à criação original, sob os termos da licença CC BY-NC 4.0.

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

© 2023 Copyright mantido pelo(s) autor(es). Direitos de publicação licenciados à Sociedade Brasileira de Computação (SBC).

REFERÊNCIAS

- [1] Michael Canesche, Lucas Bragança, Omar Paranaíba Vilela Neto, Jose A Nacif, and Ricardo Ferreira. 2021. Google Colab CAD4U: Hands-on Cloud Laboratories for Digital Design. In *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 1–5.
- [2] 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*.
- [3] Ricardo Ferreira, Jose Nacif, and Michael Canesche. 2023. Ideias para o uso do Google Colab no Ensino de Computação. <https://colab.research.google.com/drive/1-j6L3enqukGJs-dujGpDjtfBwF-1Tyop?usp=sharing>. [Online].
- [4] 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.
- [5] 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.
- [6] Cristian D González-Carrillo, Felipe Restrepo-Calle, Jhon J Ramírez-Echeverry, and Fabio A González. 2021. Automatic Grading Tool for Jupyter Notebooks in Artificial Intelligence Courses. *Sustainability* 13, 21, 12050.
- [7] Fernando Pérez and Brian E Granger. 2007. IPython: a system for interactive scientific computing. *Computing in science & engineering* 9, 3, 21–29.
- [8] Dipa Soni and Ashwin Makwana. 2017. A survey on mqtt: a protocol of internet of things (iot). In *International conference on telecommunication, power analysis and computing techniques (ICTPACT-2017)*, Vol. 20. 173–177.