

Infraestrutura para Atividades Práticas das Disciplinas de Computação a Distância

Francisco Fambrini¹, José Hiroki Saito²

¹Fesb – Fundação de Ensino Superior de Bragança Paulista.

²Unifaccamp - Centro Universitário Campo Limpo Paulista, SP
ffambrini@gmail.com, jose.saito@faccamp.br

Abstract. *Practical activities can promote a more adequate understanding of the program content, providing more interesting lessons and improving the student experience. In the Virtual Learning Environment (VLE), such activities represent a great challenge for teachers, especially for courses in the area of computing, as they require a definition of operational environment infrastructure. This infrastructure is not always adequate to the student's possibilities. In this context, this work aimed to carry out a study to identify the development environment requirements according to the profiles of distance education students and propose a reference guide for content trainers of subjects in the area of computing.*

Resumo. *Atividades práticas podem favorecer uma compreensão mais adequada do conteúdo programático, proporcionando aulas mais interessantes e melhorando a experiência do estudante. No Ambiente Virtual de Aprendizagem (AVA), tais atividades representam um grande desafio aos professores, especialmente para os cursos da área da computação, pois requerem uma definição de infraestrutura de ambiente operacional. Esta infraestrutura nem sempre está adequada às possibilidades dos estudantes de educação a distância. Neste contexto, este trabalho teve por objetivo propor um guia de referência para os formadores de conteúdo das disciplinas da área de computação, no ambiente de educação a distância.*

1. Introdução

Atividades práticas são fundamentais no processo educacional, permitindo aos alunos aplicar conceitos teóricos em situações reais e desenvolver habilidades práticas. De acordo com a Taxonomia de Bloom, essas atividades estão inseridas no domínio cognitivo e são essenciais para a aplicação de regras, métodos e teorias em contextos concretos (Ferraz & Belhot, 2010). Em cursos de computação, onde a prática é crucial para o domínio de linguagens e técnicas, o ambiente virtual de aprendizagem (AVA) representa um desafio significativo.

Um aluno dificilmente irá aprender uma determinada linguagem de programação sem executá-la, apenas estudando sua teoria. Sem esse aprendizado prático pelo estudante, ocorre o desinteresse pelo curso e como consequência a evasão. De acordo com Zorzo *et al.* (2017), novas mídias e tecnologias devem ser inseridas no meio acadêmico conduzindo a uma reflexão sobre as práticas pedagógicas.

Este trabalho tem como objetivo a identificação do perfil dos alunos em cursos típicos de educação a distância no Brasil, quanto aos recursos de infraestrutura disponíveis, e a proposição de um guia de referência para os docentes formadores de conteúdo das disciplinas da área de computação, em ambiente virtual de aprendizagem (Oliveira et al., 2020).

As descrições seguintes correspondem à Seção 2, pesquisa do perfil de alunos em educação a distância; Seção 3, proposição de atividades práticas; e Seção 4, resultados e considerações finais.

2. Perfil do aluno em educação a distância

Ao longo da elaboração deste trabalho, foi realizada uma pesquisa em relação aos recursos utilizados pelos alunos de um curso típico de educação a distância, para execução das atividades propostas no AVA, em especial às atividades práticas das disciplinas da computação. Para isso foi disponibilizado um questionário aos estudantes das disciplinas de computação do quarto bimestre, de 2021. Quanto ao ambiente computacional, a maioria dos estudantes (98%) utilizava computador pessoal (*desktop* ou *notebook*); 31% utilizava celular e apenas 8% utilizava *tablet*. Em relação ao Sistema Operacional (SO) nos computadores pessoais, a maioria utilizava o *Windows* e 10% utilizava o *Linux*. Quanto a capacidade de memória RAM: 2% utilizava até 2 Gigabytes; 37% utilizava de 3 a 4 Gigabytes; 50% de 5 a 8 Gigabytes e 12%, acima de 8 Gigabytes. Sobre a velocidade da internet a maioria (52,1%) informou que possuía mais de 100 Megabits por segundo (Mbps): 23% até 100 Mbps; 8% até 50Mbps; 4% até 30Mbps; 4% até 20Mbps; 6% até 10Mbps e 6% até 5 Mbps. Dos estudantes que utilizam celular para realizarem as atividades, todos indicaram que utilizavam o *Android* como SO. Dos que utilizavam *tablets*, 50% utilizavam *Android*, 25% *iOS* e 25% *Fire*. Na Tabela 1 ilustra-se um resumo da pesquisa, em termos de equipamentos utilizados, porcentagem de usuários dos mesmos, os sistemas operacionais utilizados, memória RAM e velocidade de internet.

Quanto à utilização dos polos, a maioria (55%) afirmou não utilizar os polos da instituição de ensino para fazer as atividades, entretanto, cerca de 43% disseram que só não estavam utilizando em virtude do fechamento provocado pela pandemia de COVID-19 (Garbin et. al., 2021). Referente aos desafios encontrados para a realização das atividades práticas (Figura 1), 49% informaram não ter nenhuma dificuldade e 34,7% relataram alguma dificuldade. Dos restantes, 6,1% não conseguiam por falta de compreensão; 4% por falta de equipamento adequado; 2% indicaram não conseguir nada relacionado a máquinas virtuais; 2% não conseguiam por falta de equipamentos pessoais; 2% tinham dificuldades de solucionar os erros encontrados na execução das atividades (Garbin et. al., 2021).

Tabela 1. Resumo dos dados sobre os Equipamentos, usuários, sistemas operacionais, memória RAM e velocidade de internet.

Equipamentos	Usuários	Sist. Oper.	Mem. RAM	Veloc. Internet
PC/Notebook	98%	Windows (90%)	2GB (2%)	100Mbps (23%)
		Linux (10%)	3 a 4GB (37%)	50Mbps (8%)
Smartphone	31%	Android	5 a 8GB (50%)	30Mbps (4%)
Tablet	8%	Android (50%)	> 8GB (12%)	20Mbps (4%)
		iOS (25%)		10Mbps (6%)
		Fire (25%)		5Mbps (6%)

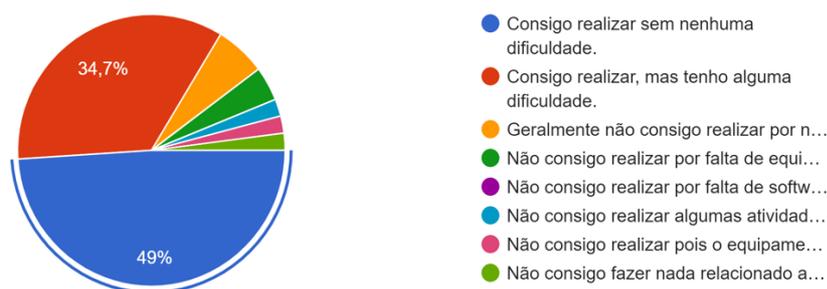


Figura 1- Dificuldades para a realização das atividades práticas.

Fonte: (Garbin et. al. 2021).

3. Proposição para Atividades Práticas em Disciplinas de Computação

Uma universidade típica no Brasil oferece os seguintes cursos específicos na área de computação: Engenharia de Computação, Bacharelado em Tecnologia da Informação e Bacharelado em Ciência dos Dados; além de diversos cursos de graduação, que também contêm disciplinas de computação básica. Para atender às necessidades das atividades práticas nessas disciplinas de forma geral são propostas as infraestruturas descritas a seguir, levando-se em conta as pesquisas descritas na Seção 2, quanto aos recursos e dificuldades enfrentadas pelos estudantes. As alternativas englobam infraestruturas: 1) totalmente web por meio de serviços de armazenamento em nuvem; 2) parcialmente web, com o uso de computador local; 3) por meio de máquina virtual; 4) por meio de computador local; e 5) ambiente de avaliação online.

3.1. Ambiente de desenvolvimento totalmente web

Uma solução que não exige nenhuma configuração externa para o seu uso e pode atender os perfis dos estudantes de educação a distância, que não contam com grande poder computacional local. O *Google Collaboratory* ou simplesmente *Google Collab*, é um serviço de armazenamento em nuvem voltado à criação e execução de códigos diretamente em um navegador, sem a necessidade de nenhum tipo de instalação de software em uma máquina (COLAB, 2024). É uma solução que fornece aos seus usuários a opção de trabalhar em um ambiente de execução totalmente em nuvem. A ferramenta funciona em ambiente colaborativo, permitindo o compartilhamento do

código fonte. Sua linguagem de programação nativa é *Python*, mas é possível executar códigos nas linguagens *R*, *Julia*, *Swift* e outras. É possível misturar código fonte com texto e imagens, por meio da estrutura de *markdown*, como se fosse um “caderno”, daí o termo *notebook*. Para Silva (2020), o uso desta ferramenta propicia a prática de programação, reduz a complexidade da aprendizagem de programação e possibilita o acompanhamento de forma colaborativa. Os recursos necessários são: um navegador web com acesso à internet e uma conta no Google, portanto, pode ser utilizado um computador pessoal, *tablet* ou *smartphone*, sem nenhuma configuração robusta. Entretanto, este tipo de solução não é capaz de abranger todas as disciplinas de Computação e por isso existe a necessidade de se implantar outras soluções, que englobam a instalação parcial de *software* e também o uso de *softwares* localmente instalados.

3.2. Ambiente de desenvolvimento parcialmente web

A primeira proposta é a utilização de uma ferramenta *online* (como o interpretador de linguagem *Python*) mas com possibilidade de se instalar a ferramenta no computador local. Um website que oferece a ferramenta desta maneira é o Jupyter (2024). A instalação no próprio computador é sugerida porque a ferramenta *online* (modo “try”) nem sempre está disponível (o número de notebooks que pode ser aberto *online* é limitado). Uma segunda proposta, que envolve um juiz online (Kurnia et al., 2001) é utilizar uma ferramenta online para efetuar testes iniciais no Código-Fonte gerado pelos alunos, aproveitando-se também que tais ferramentas trazem um grande número de exercícios propostos. Após testar o código inicialmente no website (que caracteriza o desenvolvimento *online*) o aluno poderia copiar e colar seu código num compilador localmente instalado, como o DevC++ (BLOODSHED, 2024). Um exemplo de ambiente dessa segunda proposta é URI Online Judge (2024), que é um projeto desenvolvido pela *University of Rhode Island*, URI, com o objetivo de fornecer prática de programação e compartilhamento de conhecimento, especialmente nas linguagens C++17, C++ e C#. Estes dois websites (Jupyter e URI Online Judge) trazem propostas interessantes para os interessados em linguagens *Python*, C e suas variantes: C++17, C++ e C#. Em particular, URI é interessante por apresentar um grande número de problemas. O URI checa erros de sintaxe no código escrito e enviado, porém não indica em quais linhas do código estão os erros. Ainda quanto ao URI, há falta de linguagens, como *Python*, JAVA ou R, mas os problemas propostos podem ser adaptados para estas e outras linguagens. O uso do ambiente *online* em conjunto com os compiladores localmente instalados fornece valiosa ajuda para os testes dos códigos produzidos pelos alunos no desenvolver das disciplinas que incluem programação.

3.3. Ambiente de desenvolvimento por máquina virtual

É comum se deparar com softwares de desenvolvimento muito distintos (exemplos: diferentes versões de *PHP*, versões diferentes do *Apache*, diferentes versões do *mySQL*,

e outros). É muito comum também um usuário desenvolver suas aplicações em ambiente *Windows* e depois, quando envia estas mesmas aplicações para um Servidor (que normalmente executa alguma versão de Linux) encontrar problemas, devido a diferentes configurações de *Apache*, *JAVA*, *PHP*, *mySQL* e outros. Uma solução para esses problemas é fazer com que todas as aplicações sejam executadas em configurações idênticas, fazendo uso ferramentas de virtualização. Virtualização é o processo de simular um *hardware* ou sistema operacional sobre uma máquina local ou na nuvem, para uma determinada aplicação. Nesse sentido, resume-se a seguir as seguintes ferramentas para auxiliar no processo de virtualização (Agarwal et al., 2020): **1) Docker**- uma plataforma baseada em serviços que implementa uma virtualização ao nível de SO e entrega softwares como pacotes em *containers*; **2) Kubernetes**- uma ferramenta de orquestração de *containers* que é usada para a aplicação, escalção e gerenciamento de software de forma automática; **3) Vagrant** - uma plataforma para construção e gerenciamento de ambientes virtuais; e **4) VirtualBox** - um software de virtualização para criação de ambientes virtuais.

3.4. Ambientes de desenvolvimento em computador local

Como ambientes de desenvolvimento local, os autores sugerem o uso de três diferentes IDEs (*Integrated Development Environment*), sendo a primeira voltada para a linguagem C/C++, a segunda para a linguagem *Python* e a terceira é um ambiente genérico, capaz de ser integrado a diversos compiladores e interpretadores de diversas linguagens distintas, incluindo C/C++ e Python. As IDE sugeridas são: **1) DevC++** (BLOODSHED, 2024); **2) Spyder** (SPYDER, 2024) e **3) Visual Studio Code** (VISUAL, 2024).

3.5. Ambientes de avaliação online

Um ambiente de avaliação online utilizado por cursos de educação a distância é o AVA Blackboard (AVA, 2024). Neste ambiente, podem ser disponibilizadas questões e exercícios online, para execução pelos alunos. Entretanto, neste ambiente, não é possível checar automaticamente a validade e a correção de um determinado código elaborado pelo aluno. Tal correção necessita ser feita por um corretor humano (que no caso dos cursos atuais é feita através dos facilitadores ou tutores). Por outro lado, existem alguns ambientes *online* que automatizam a checagem de um código postado pelo aluno, como por exemplo o já anteriormente citado *URI Online Judge*. Existem, entretanto, outros websites semelhantes ao URI, que também incorporam a funcionalidade de propor problemas e julgar as soluções online (Wasik et al., 2018) (Liu et al., 2023). Citam-se, dentre outros: **1) UVA**: que reúne um conjunto de problemas dos mais variados assuntos: *ad hoc*, programação dinâmica, *backtracking* e outros; também inclui vários problemas usados em maratonas de programação mundiais e permite submeter soluções para os problemas em diversas linguagens, tais como C/C++, Java e outros (UVA, 2024; Revilla et al., 2008); **2) SPOJ** (*Sphere Online Judge*): semelhante ao UVA, possui tradução para o português (SPOJ, 2024); **3) TopCoder**: semelhante aos

anteriores, algumas vezes apresenta campeonatos de programação patrocinados por grandes empresas, como NASA, Yahoo e Google (TOPCODER, 2024); 4) Project Euler: voltado para problemas matemáticos, onde o usuário deve entrar com a resposta, e também deve criar algoritmos para resolver os problemas propostos (EULER, 2024); 5) CodingBat: voltado para iniciantes que queiram praticar Java e/ou Python (CODINGBAT, 2024); 6) Coderbyte: que oferece um ambiente para codificar e testar os algoritmos antes de submeter as soluções para os problemas propostos (CODERBYTE, 2024); 7) CodeChef: com formato e funcionamento semelhantes ao SPOJ e UVA e aceita soluções em várias linguagens como C/C++, Java, Python, Go, Lisp, PHP, entre outras (CODECHEF, 2024); 8) Hacker Rank: outro juiz online no qual os alunos podem submeter soluções em diversas linguagens (HACKER RANK, 2024); e 9) CodinGame: para a resolução de problemas através de *puzzles* e jogos simples, suportando várias linguagens distintas (CODINGAME, 2024). Modernamente (2024), também é possível usar o *ChatGPT* para checar a sintaxe dos códigos gerados e até mesmo gerar códigos inteiros, o que não deve ser estimulado visando o bom aproveitamento dos alunos.

4. Resultados e Considerações Finais

Como resultados cita-se a instalação de uma Máquina Virtual especialmente desenvolvida por Eduardo Santarém para a disciplina em Banco de Dados (Santarém, 2022) e também uma outra máquina virtual customizada e disponibilizada no repositório Github, destinada ao ensino de PHP. Foi criada uma máquina virtual baseada em Ubuntu com os seguintes pacotes pré-configurados e prontos para uso: PHP 7.1, MySQL 5.5, Git, Php MyAdmin, Composer, cURL (Client URL), Vim e Redis, destinada aos cursos do eixo de computação. Também pode-se incluir como resultado uma máquina virtual criada por Carlos Ferreira, para o Curso “Vagrant”, oferecido na Plataforma UDEMY(2023). Atualmente, existem esses e outros projetos pilotos de atividades práticas para algumas disciplinas da área de computação: 1) Banco de Dados, exercícios a serem resolvidos e entregues em VM; 2) Microeletrônica, exercícios e projetos em simuladores e *protoboards*; 3) Computação Gráfica, exercícios em VM e desenvolvimento de software de reconhecimento de imagens em *Deep Learning*; 4) Eletrônica Embarcada, exercícios e projetos em simuladores e plataformas Arduino, ESP32 e *Shields*; 5) Projeto de Sistemas Computacionais, exercícios e projetos em simuladores; 6) Infraestrutura para Sistemas de Software, atividades para instalação de uma VM e de um sistema operacional numa VM, instalação e configuração dos servidores Apache e MySQL, instalação e configuração do PHP, aplicação com Apache, PHP e MySQL, customização de ambientes virtualizados com Virtual Box, aplicação PHP utilizando uma interface REST API, instalação e configuração do Apache, PHP e MySQL em ambiente Docker, aplicação PHP baseado em web e via REST API no Docker (FAMBRINI, 2021).

Em relação à justificativa do levantamento de perfil dos alunos, Seção 2, quanto ao uso de SO's, banda de internet disponíveis, e da proposta de infraestrutura, Seção 3, segue o seguinte: 1) pela pesquisa, a maioria dos alunos não utilizava os computadores

dos Polos da instituição de educação a distância, e sim computadores próprios; 2) diferentes computadores apresentam diferentes configurações de SO's, implicando nos recursos como VM (Máquina Virtual) e Vagrant, evitando assim diferentes resultados na execução de programas; 3) o uso de diferentes versões de Python pode acarretar no não funcionamento de um código exemplo em aulas; 4) esse perfil dos alunos é importante para justificar a necessidade de se criar Máquinas Virtuais e Dockers e deixar preparada uma instalação dos programas em diversos computadores, com plena compatibilidade.

Embora a máquina virtual tenha sido útil, o *feedback* dos alunos indica a necessidade de melhor suporte técnico para resolver problemas encontrados. Este suporte seria em princípio oferecido pelos facilitadores.

Uma consideração final é que a aplicação da infraestrutura proposta neste trabalho, não seria restrita à instituição de educação a distância pesquisada, mas poderia ser utilizada em outras instituições, em ensino de disciplinas de computação.

Referências

- [1] Agarwal A., Rao S.K.S., Mahendra B.M. (2020) “Comprehensive Review of Virtualization Tools”, *International Journal of Engineering and Technology (IRJET)*, vol.7, n.06, p. 4394-4397.
- [2] BLOODSHED(2024) “Open Source C/C++ IDE for Windows”, [https:// www.bloodshed.net](https://www.bloodshed.net), March.
- [3] CODECHEF (2024) “Start your coding journey today”, <https://codechef.com/>, July.
- [4] CODERBYTE(2024), “Evaluate Candidates Quickly, Affordably, and Accurately”, <https://coderbyte.com/>, July.
- [5] CODINGAME (2024) “Play coding games. Have fun. Build skills”, July.
- [6] CODINGBAT (2024) “CodingBat code practice”, <https://codingbat.com/java>, July.
- [7] COLAB (2024) “Olá, este é o colab”, <https://colab.research.google.com>, January.
- [8] EULER (2024) “Project Euler”, <https://projecteuler.net/>, July.
- [9] Fambrini F., Freitas A., Progetti C.B., Machado D. (2021), “Infraestrutura para Atividades Práticas das Disciplinas dos Cursos de Computação na UNIVESP, Trabalho de Conclusão de Curso de Especialização: Desenvolvimento de Soluções Inovadoras para a Educação a Distância”, UNIVESP.
- [10] Ferraz A.P.C.M., and Belhot R.V. (2010) “Taxonomia de Bloom: revisão teórica e apresentação das adequações do instrumento para definição de objetivos instrucionais”, *Gestão e Produção*, São Carlos, v.17, n.2, p. 421-431.
- [11] Garbin M.C., Oliveira E.T., Haas C.M., Verão G. B. Telles S. (2021) “Tecnologias na Educação: Ensino, Aprendizagem e Políticas”. Consultado online em 03/10/2024 no website: https://univesp.br/sites/58f6506869226e9479d38201/assets/614b28557c1bd10585e07b69/tecnologias-na-educacao_04_1_.pdf

- [12] HACKER RANK (2024) “Skills speak louder than words”, <https://hackerrank.com>, April.
- [13] Jupyter (2024) “Try Jupyter - Use our tools without installing anything”, <https://jupyter.org/try>, July.
- [14] Liu K., Han Y., Zhang J.M., Chen Z., Sano F., Harman M., Huang, G., MaY. (2023) “Who Judges the Judge: An Empirical Study on Online Judge Tests”, *ISSTA, International Symposium on Software Testing and Analysis*, Seattle-USA, July 17-21, p.1-13.
- [15] Kurnia A., Lim A., Cheang B. (2001) “Online Judge”, *Computers&Education*, vol. 36, p. 299–315.
- [16] Oliveira N.J.S., Procaci T.B., Siqueira S.W.M. (2020) “Captura e aceitação do Blackboard e do tipo de motivação de alunos de cursos presenciais de ciências exatas em universidade privada”, *Revista Brasileira de Informática na Educação*, vol.20, p. 229-259.
- [17] Online Judge (2024) “Online Judge”, <https://onlinejudge.org>, May.
- [18] Revilla M.A., Manzoor S., Liu R. (2008) “Competitive Learning in Informatics: The UVA Online Judge Experience”, *Institute of Mathematics and Informatics, Vilnius, Olympiads in Informatics*, vol. 2, p.131-148.
- [19] Santarém E. (2022) “Notas de aulas da disciplina COM300-Banco de Dados”, UNIVESP.
- [20] Silva M. D. (2020) “Aplicação da Ferramenta Google Colaboratory para o Ensino da Linguagem Python”, *Anais da Escola Regional de Eng.de Software (ERES)*, p.67–76.
- [21] SPOJ (2024) “Sphere Online Judge”, <https://www.spoj.com/>, July.
- [22] SPYDER(2024) “The Python IDE that scientists and data analysts deserve”, <https://www.spyder-ide.org/>, March.
- [23] TOPCODER (2024) “Find bugs, before your users do”, <https://www.topcoder.com>, July.
- [24] UDEMY (2023) “Udemy”, <https://www.udemy.com>, January.
- [25] UVA (2024) “Virtual Judge”, <https://vjudge.net/>, July.
- [26] VISUAL(2024) “Visual Studio”, <https://code.visualstudio.com/>, March.
- [27] Wasik S., Antczak M., Badura J., Lskowski A., Sternal T. (2018) “A Survey on Online Judge Systems and their Applications”, *ACM Computing Surveys*, vol. 51, n. 1, article 3.
- [28] Zorzo, A. F., Nunes D., Matos E., Steinmacher I., Leite J., Araujo R. M., Correia, R., Martins S. (2017) “Referenciais de Formação para os Cursos de Graduação em Computação”, *Sociedade Bras.de Computação (SBC)*, 153p.