

APLICATIVO HTML PARA CONSTRUÇÃO E COMPARTILHAMENTO DE OBJETOS INTERATIVOS FACILITADOS POR IA PARA ENSINO E PESQUISA

HTML APPLICATION FOR BUILDING AND SHARING AI-ENABLED INTERACTIVE OBJECTS FOR TEACHING AND RESEARCH

APLICACIÓN HTML PARA LA CREACIÓN Y EL INTERCAMBIO DE OBJETOS INTERATIVOS FACILITADOS POR IA PARA LA ENSEÑANZA Y LA INVESTIGACIÓN

José Maurício Schneedorf Ferreira da Silva

Departamento de Bioquímica, Universidade Federal de Alfenas

RESUMO. Independentemente da teoria de ensino e aprendizagem que se ofereça à apropriação de conteúdos curriculares, do ensino fundamental ao pós-doutoramento, um ponto pacífico entre autores elenca a tríade de estímulo, proximidade cognitiva, e compromisso do aluno sobre o tema abordado. Não obstante a interatividade digital e social presente nos AVAs do EaD e ensino híbrido sobrepõe-se nesses tempos a clássica abordagem por aulas expositivas e livros didáticos frequentes no ambiente presencial. Não é de se estranhar, portanto, que numa década em que se apresentam a realidade aumentada/virtual e a IA generativa na ponta dos dedos de dispositivos móveis dos jovens, haja um profundo descompasso entre as ferramentas tecnológicas disponíveis, e a formação desses ao olimpo da Educação 5.0. Este trabalho apresenta uma solução tecnológica formada por um simulador e um bot personalizado para geração de códigos, visando a criação e compartilhamento de objetos didáticos interativos variados, tais como gráficos, mapas, fluxogramas, animações, simulações, jogos, sonorização, e interfaceamento com microcontroladores. O simulador apresenta-se com um simples arquivo de texto (< 30 kb) interpretado em qualquer browser, flexibilizando seu uso de desktops a smartphones, para inserção/edição de códigos JavaScript em janela única e com geração lado a lado do objeto interativo. Encontra-se sob licença CC BY-NC-SA e depósito no eduCAPES, estando disponível e ilustrado ao ensino superior e básico junto a site educacional. A plataforma encontra-se em avaliação junto a alunos(as) de graduação e professores(as) de Escolas Estaduais, tendo-se mostrado promissora como ferramenta auxiliar ao ensino e aprendizagem em diferentes níveis, modalidades, e áreas do conhecimento. Por envolver o pensamento computacional de códigos facilitados por assistente de IA personalizado e voltados a conteúdos curriculares, a plataforma pretende contribuir como uma alternativa aberta, simples, rápida, lúdica, reproduzível e compartilhável à uma Educação 5.0 que ora se faz premente.

Palavras-chave: Pensamento computacional. Ensino reproduzível. JavaScript. Objetos interativos. Inteligência Artificial.

1 INTRODUÇÃO

Do quadro de giz à realidade virtual, o ensino e aprendizagem têm passado por incontáveis recursos tecnológicos e metodologias afins. Entre essas, um



amadurecimento das abordagens ativas ao longo do século XX tem possibilitado aplicar o leque de ferramentas tecnológicas ao ensino presencial, remoto, semipresencial, híbrido, ou EaD. Ilustram-se nesta seara o aprendizado baseado em problemas de Barrows (*Problem-based learning*) ao final dos anos 60 no Canadá (Barrows, 1986; Walker e Leary, 2009), e fartamente empregado ao ensino médico, bem como o aprendizado baseado em projetos (*Project-based learning*) inspirado em Dewey e na Escola Nova, para uma abordagem construtivista de “aprender fazendo” a um produto tangível (Zhang e Ma, 2023).

Outras abordagens metodológicas do caldo dessa contemporaneidade surgiram em decorrência da popularização de softwares de animação nos anos 1980-90, como plataformas multimídia para animações integradas por gráficos vetoriais e rasterizados, áudio, vídeo e interatividade como PowerPoint® e Flash® (Macromedia, Adobe). Até a descontinuidade do segundo em 2020, sua disseminação consolidou um conjunto extenso de objetos didáticos digitais voltados a um aprendizado com cenários em animação (*Animation-based learning*; Wulansari et al., 2020), sua inclusão em páginas da web, e mesmo sua oferta como material suplementar comercializável junto a obras didáticas.

Também vale elencar a iniciativa para conteúdos didáticos por meio de jogos e simulações. Diferente da *gameificação*, em que atividades tradicionais do ensino são estimuladas com elementos de competição (pontuação, níveis, desafios), o aprendizado baseado em jogos (*Game-based learning*) busca inserir a temática abordada na estrutura de um jogo (Plass, 2015), permitindo sua compreensão progressiva por acertos, erros e *feedback* para soluções de contorno. Outra abordagem ativa de relevância dá-se pelo aprendizado baseado em simulação (*Simulation-based learning*), fortalecido na última década frente às novas tecnologias digitais de informação e comunicação (NTDICS), e que permitem replicar no ambiente virtual os modelos físicos da experimentação (Chernikova et al., 2020).

O aprendizado baseado em simulação (SBL) remonta à década de 1965 (Hallinger e Wang, 2020), e direciona-se ao “uso de programas que podem representar visualmente fenômenos reais ou hipotéticos, e permitir que os utilizadores interajam com os programas” (Krisnadi et al., 2022). Para agregar valor a uma aprendizagem significativa, o

SBL apoia-se 1) na interatividade por ação do usuário e retorno do sistema, 2) exploração ou manipulação paramétrica para observação de alterações no fenômeno abordado, e 3) na discussão do fenômeno frente à modelagem por simulação.

Consonante aos tempos recentes, a possibilidade de trabalhar-se com grande volume de dados (*big data*), realidade aumentada e virtual, computação quântica, aprendizado de máquina, conexão de banda larga, laboratórios virtuais, internet das coisas (IoT), inteligência artificial generativa (IA) e IA geral almejada (IAG), têm como um todo possibilitado o uso da modelagem por simulação da academia à indústria (ex: simuladores para engenharia aeroespacial, cirurgia robótica), e vice-versa (ex: simuladores para automobilismo, indústria nuclear). Ao ensino-aprendizagem é expressivo o emprego de algumas aplicações web que permeiam a simulação computacional, quer desprovida da inserção de códigos pelo usuário, como [Geogebra](#) (Matemática) e [PeTh Colorado](#) (Ciências e Matemática), quer pela inserção desses, como [MIT Scratch](#) (programação visual para crianças) e [Autodesk TinkerCAD](#) (modelagem 3D e circuitos eletrônicos; C/C++). Em território nacional destaca-se a iniciativa para a plataforma [SablÁ](#) (Delta, FALE-UFMG), voltada à curadoria e exploração de ferramentas de IA aplicadas ao ensino e à aprendizagem de línguas. Méritos à parte, as duas últimas capacitam o aprendiz à inserção do pensamento computacional (PC) “plugado”, ao envolver lógica e linguagens de programação. O PC confere uma abordagem símile ao algoritmo experimentado em programação para solução de um problema, e abrange as etapas de abstração, decomposição, reconhecimento de padrões, e criação de algoritmo (Wing, 2006).

Dessa forma, a simulação direcionada a um alvo pedagógico e empregada junto a uma linguagem de programação agrega valor educacional ímpar junto às novas tecnologias. E como previsto por McKenney e Dill há quase seis décadas (1966), pelo:

...aperfeiçoamento das técnicas e linguagens de simulação e com as expansões tanto na capacidade dos computadores quanto na flexibilidade dos arranjos pelos quais os alunos podem interagir com a máquina.

Assim, o emprego de simulações com linguagem de programação como abordagem metodológica ativa tende a lubrificar o caminho do ensino e aprendizagem mediante alguns benefícios, tais como (Farmonov, 2023):

1. Infraestrutura dispensável para as atividades (basta um desktop ou dispositivo móvel);
2. Ausência de treinamento técnico requerido ao uso de instrumentação física;
3. Dispensa de mão-de-obra especializada em paralelo ao treinamento do usuário;
4. Tempo exíguo e mesmo instantâneo para consolidação da simulação;
5. Possibilidade de se trabalhar com situações não reproduzíveis em experimentação (ex: fissão nuclear, observações astronômicas);
6. Dispensa de recursos financeiros para consolidar a simulação (insumos, utensílios, equipamentos);
7. Desnecessidade de proteção contra incidentes e descarte de resíduos;
8. Modificação do objeto em simulação e *feedback* imediato do sistema por manipulação do código (exploração paramétrica);
9. Possibilidade de inserção do aprendiz em pensamento computacional;
10. Possibilidade de inserção do aprendiz em lógica e linguagens de programação diretamente para conteúdos curriculares;
11. Contribuição para um *ensino reproduzível* de códigos para conteúdos curriculares.

Tangente à última característica itemizada acima, o ensino reproduzível (Dogucu, 2024) consiste numa metodologia ativa incipiente para reprodução/modificação/criação de trechos de códigos de programação direcionados a um conteúdo do ensino-aprendizagem. Tal como a pesquisa reproduzível de onde se origina, o ensino reproduzível baseia-se na oferta aberta de códigos acessíveis e executáveis, documentação e metadados, uso de ferramentas e linguagens de programação reproduzíveis, testabilidade, repositórios públicos, transparência, e compartilhamento de dados.

Face às abordagens acima, às necessidades cada vez mais urgentes para um ensino em aderência a uma Educação 5.0 direcionada à sustentabilidade e inclusividade, e às iniciativas públicas dos anos recentes para incentivo e aprimoramento de competências digitais, bem como inserção do PC na Escola e na academia (vide Quadro Europeu de Competências Digitais Para Educadores, [DigCompEdu](#), Política Nacional de Educação Digital - [PNED](#), Futuro da Educação e Competências 2030/2040 - [OCDE](#), entre vários), este trabalho apresenta uma ferramenta digital aberta para construção e edição de objetos didáticos interativos por linguagem de programação auxiliada por IA. A plataforma permite utilizar qualquer uma das abordagens ativas descritas acima, e pode ser empregada tanto para ensino e aprendizagem em modalidades e níveis distintos, como para pesquisa científica.

2 A PLATAFORMA

A plataforma é composta por um simulador para inserção de códigos e um gerador desses por bot personalizado. O simulador permite a inserção de comandos em editor

próprio por JavaScript e HTML, duas linguagens presentes na construção moderna de páginas de internet para interatividade e conteúdo, respectivamente. Como representado na Figura 1, a interpretação do código resulta em um produto visualizado lado a lado ao editor, o que permite rápidos ajustes na busca do objeto final desejado. O editor incorpora a principal biblioteca para construção de gráficos e mapas interativos da linguagem, [Plotly.js](https://plot.ly), embora possa trabalhar com qualquer outra.

Embora os comandos apresentados à Figura 1 possam suscitar desconforto (e mesmo desespero) aos não iniciados nas Ciências da Computação, a plataforma conta com um gerador de códigos por GPT customizado, tal como fora empregado para construção e ajustes dos objetos da própria Figura 1. Dessa forma, o gerador por assistente de IA permite:

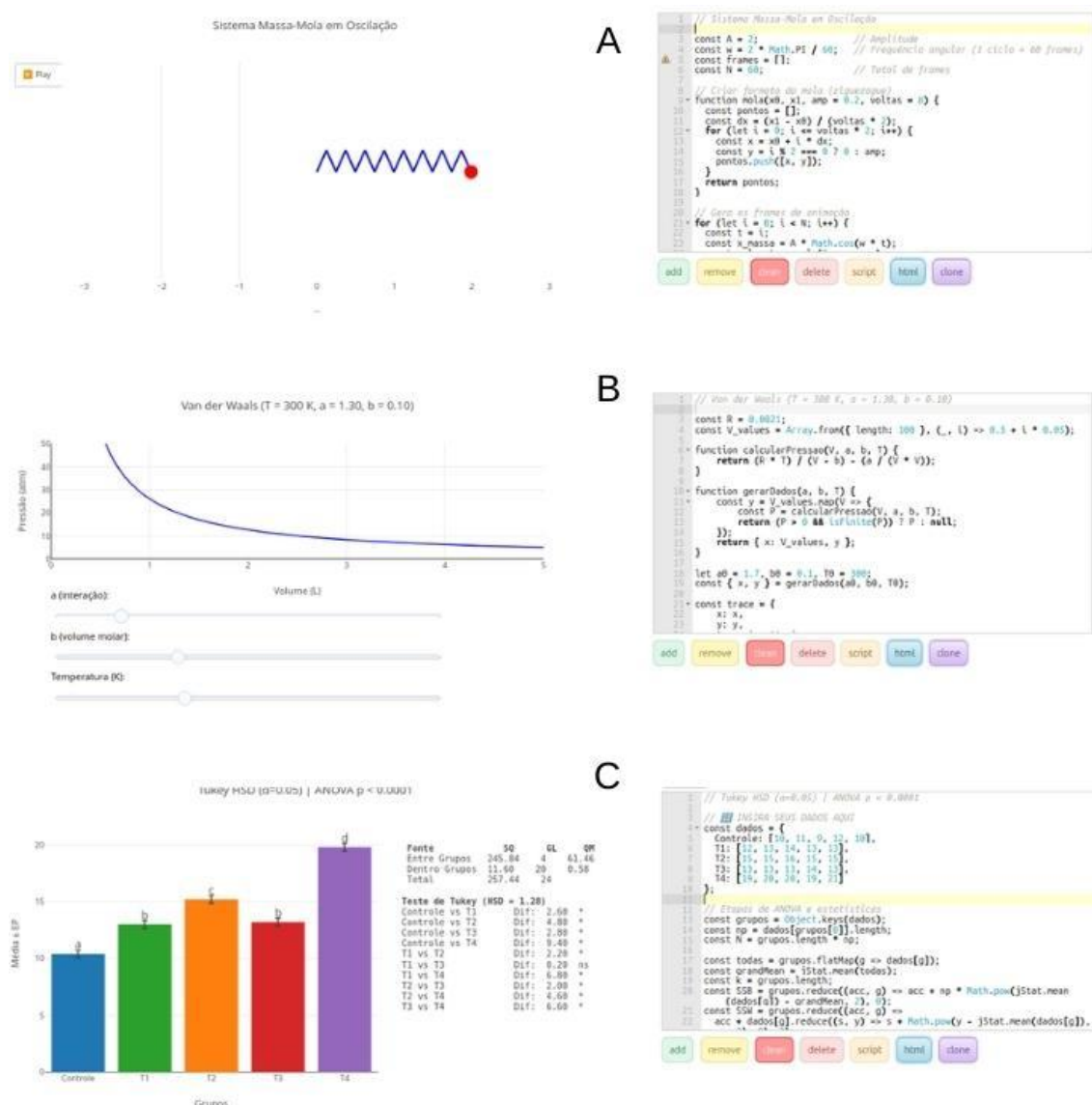
1. Gerar um código instantaneamente para o simulador;
2. Gerar um código para virtualmente qualquer tema, nível de profundidade e área do conhecimento;
3. Aliviar qualquer necessidade do usuário no conhecimento da sintaxe e operação das linguagens e bibliotecas envolvidas;
4. Inserir ou aprimorar o usuário em curva progressiva e suave para o PC, bem como para a lógica e linguagem de programação (declaração de variáveis, laços iterativos, estruturas condicionais, etc);
5. Aprimorar boas práticas de engenharia de prompts, pela troca de códigos e informações a um produto final almejado;
6. Construir diversos objetos didáticos interativos para ensino, como gráficos, mapas, fluxogramas, animações, simulações, jogos, sonorização, entre outros;
7. Construir diversos objetos interativos relacionados à pesquisa científica, como gráficos estatísticos, cálculos variados, e análise de dados, dentre outros.

2.1 A estrutura do simulador

Para possibilitar uma disseminação tão ampla quanto possível, o simulador consta de um arquivo único com menos de 30 kilobytes construído em HTML, JavaScript e CSS, com design responsivo, licença de uso CC BY-NC-SA, e depósito no portal eduCAPES. Dessa forma, buscou-se contemplar seu uso aberto para qualquer navegador moderno (HTML5), bem como sua personalização facultativa num simples editor de texto. Em adição, o arquivo é interpretado indistintamente de sistemas operacionais (Windows, Linux, Android, iOS, macOS, etc) e de hardware (*desktop, notebook, tablets, smartphones*), permitindo o uso para virtualmente qualquer dispositivo utilizado por alunos, professores e gestores, e independentemente de configuração mínima exigida (memória RAM, armazenamento físico, por ex).

Por ser elaborado como arquivo de texto e tamanho inferior a 30 kb (e aqui o autor permite-se a um saudosismo pranteado sobre aquele disquete de 360 kb e 5 ¼ de polegada dos anos 80), o app pode ser compartilhado instantaneamente e sem restrições mesmo por redes sociais.

Figura 1 – Simulador HTML em janela única apresentando o editor de códigos JavaScript à direita e o ecrã para o objeto interativo à esquerda. A - animação para uma oscilação harmônica simples aplicada a um sistema massa-mola. B - equação de estado de Van der Waals para gases reais com deslizadores para variação de temperatura, e correção da lei de gases ideais para atrações intermoleculares e volume excluído. C - Análise de variância de teste de Tukey-HSD para dados de usuário.



Fonte: Elaborado pelo autor.

As principais características do simulador interativo, seu potencial educacional, vídeo explicativo, funcionalidades, procedimentos de uso, diferencial com materiais pré-existentes, e aplicações variadas ao ensino básico e superior, encontram-se ilustrados em website autoral e prêmio nacional em Educação. Na versão atual, o app conta com alguns aprimoramentos não reportados, tais como:

1. Suporte para múltiplos traços (útil para gráficos com representações simultâneas);
2. Suporte para animação;
3. Suporte para sons em eventos mapeados;
4. Barra de funcionalidades expandida para o ecrã (ex: desenhos à mão livre e figuras geométricas, anotações de texto e marcação de pontos);
5. Programação reativa parcial (ação do usuário na área gráfica altera a sequência de interpretação do código);
6. Exportação de arquivo HTML autossuficiente para animação e programação reativa - botão "html" (útil para compartilhamento de uma animação ou objeto interativo "congelado" a um tema específico, para compartilhamento sem necessidade do simulador);
7. Clonagem do simulador com o código desejado - botão "clone" (útil para compartilhamento simultâneo do objeto juntamente a seu código, permitindo a personalização do objeto, em paralelo à exploração do tema abordado e da linguagem de programação).

2.2. Pensamento computacional reverso

O simulador apresentado neste texto permite a produção de um número virtualmente ilimitado de objetos para temas de igual magnitude, e expressos de modo interativo na forma de gráficos, mapas, diagramas, animações, simulações, tabelas, cálculos, mapas conceituais, jogos, sonorização, e treinamentos específicos, entre outros. Essa riqueza de configurações oportuniza ao usuário desde a apropriação progressiva da temática envolvida, pela exploração paramétrica de parâmetros do código, até a lógica e linguagem de programação trazidos por esse, permitindo-o à inserção e aprimoramento do PC (Oliveira, 2025).

Como de conhecimento natural, qualquer linguagem de programação oferece uma barreira monolítica ao usuário para a concretização do alvo principal: um algoritmo. Entretanto, a plataforma também apresenta um assistente de IA customizado para a produção de tais códigos. Esses são passíveis de obtenção instantânea para objetos mais simples (gráficos de pontos ou barras, por ex), ou ao longo da troca de informações com o bot por engenharia de prompts (um código com animação e programação reativa modificada pelo usuário, por ex).

Em síntese, ainda que o usuário não precise conhecer a estrutura (lógica) ou sintaxe (linguagem) de programação envolvida, o algoritmo gerado pelo assistente de IA oferece uma oportunidade ímpar para um aprendizado suave e progressivo às competências digitais de programação requeridas.

Isso tangencia um aspecto reverso do PC (Oliveira, 2025). Nesse, o problema é inicialmente dividido em partes para melhor compreensão (decomposição), seguindo-se a identificação de similaridades e repetições apresentadas (reconhecimento de padrões), a separação do que é essencial à solução (abstração), e a criação de um código que automatize (algoritmo). Por sua vez, a plataforma apresentada oferece inicialmente o código para solução do problema pautado (algoritmo), seguindo-se facultativamente a compreensão progressiva de sua estrutura geral (reconhecimento de padrão e abstração), como a lógica de programação ou pseudolinguagem que o contém, e culminando nos detalhes de sintaxe de seus comandos (decomposição).

Ainda que aparente um atalho fácil para a obtenção de um algoritmo à solução de um problema, essa é a etapa mais longa e determinante da abordagem convencional, exigindo tempo e dedicação expressivas a um único produto pretendido, quando não o trabalho colaborativo. Por outro lado, essa reversão do PC passa a focar não no algoritmo em si, mas na compreensão ampla do conteúdo que este é capaz de abranger, posto que da oferta do código por IA pode seguir-se a exploração de seus parâmetros, mudanças de configurações, observações da linguagem e de seus padrões sintáticos, resultando em distintos objetos interativos correlatos. Pela celeridade com que o objeto é prontamente oferecido, isso possibilita uma exploração mais profunda e dedicada ao tema abordado e, como resultado, a possibilidade de uma apropriação mais concreta e ágil, frente também à interatividade de objetos múltiplos compartilháveis em diversos formatos, tal como para

gráficos, mapas, animações e simulações. Ou seja: foca-se no conteúdo e não na código (forma), embora a exploração desse seja natural e paulatinamente incorporada aos mecanismos cognitivos do aprendiz. Em adição, a persistência desse para a compreensão de um assunto por meio da reconfiguração de objetos editados por códigos em tempo ágil por assistente de IA tende a privilegiar um aprendizado natural, tanto da lógica como da linguagem de programação envolvidas, posto que a etapa determinante e longa do processo é facilitada pelo bot. Finalmente, a própria engenharia de prompts faculta a criação de trechos de códigos de destaque, e passíveis de incorporação em novos códigos para temáticas distintas.

2.3. Aplicação da plataforma ao ensino básico e superior

Ao longo deste ano, a plataforma tem sido utilizada no ensino presencial para alunos de graduação (Química e C. Biológicas) da IES de lotação do autor, bem como para capacitação docente junto a todas as 7 Escolas Estaduais do entorno municipal, em parceria com a Superintendência Regional de Ensino, e para as 4 grandes áreas da educação básica (matemática, ciências da natureza, linguagens, ciências humanas). Ao momento da redação deste texto, contudo, dados quantitativos ainda não estão disponíveis, face à etapa atual de aplicação da plataforma junto ao público-alvo.

3 CONSIDERAÇÕES FINAIS

Este texto teve por objetivo apresentar uma plataforma formada por um simulador e um gerador de códigos por GPT personalizado para a construção de objetos didáticos para ensino e pesquisa. Como um todo, o gerador permite a elaboração quase instantânea de código JavaScript e bibliotecas correlatas para a interpretação no simulador, facilitando a aplicação desse para a criação e compartilhamento abertos de objetos interativos variados (gráficos, mapas, animações, simulações), em qualquer temática, nível ou modalidade de ensino, e facultando a inserção do aprendiz no pensamento computacional e lógica de programação.

Assim, a plataforma pretende contribuir em consonância à iniciativas globais e nacionais para uma Educação 5.0 pautada no uso consciente, ético e reflexivo das

NTDICs, e aplicada diretamente a conteúdos escolares e acadêmicos que tangenciam a formação inicial e continuada dos atores envolvidos no processo de ensino e aprendizagem.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

REFERÊNCIAS

BARROWS, Howard S. A taxonomy of problem-based learning methods. **Medical Education**, v. 20, n. 6, p. 481-486, 1986. Disponível em: <https://doi.org/10.1111/j.1365-2923.1986.tb01386.x>. Acesso em: 17 ago. 2025.

CHERNIKOVA, Olga; HEITZMANN, Nicole; STADLER, Matthias; HOLZBERGER, Doris; SEIDEL, Tina; FISCHER, Frank. Simulation-based learning in higher education: a meta-analysis. **Review of Educational Research**, v. 90, n. 4, p. 499-541, 2020. Disponível em: <https://doi.org/10.3102/0034654320933544>. Acesso em: 17 ago. 2025.

DE OLIVEIRA JUNIOR, Emerson Rogério; DE BORTOLI, Lis Ângela; CASTAMAN, Ana Sara. Confraria da Lógica: uma organização educacional para potencializar o pensamento computacional entre estudantes da Educação Profissional e Tecnológica. **Caderno Pedagógico**, v. 22, n. 1, p. e13548, 2025. Disponível em: <https://doi.org/10.54033/cadpedv22n1-210> . Acesso em: 17 ago. 2025.

DOGUCU, Mine. Reproducibility in the classroom. **Annual Review of Statistics and Its Application**, v. 12, 2024. Disponível em: <https://doi.org/10.1146/annurev-statistics-112723-034436>. Acesso em: 17 ago. 2025.

FARMONOV, Utkir. Advantages of using virtual laboratories in teaching of physics. **Mental Enlightenment Scientific-Methodological Journal**, v. 4, n. 6, p. 70-74, 2023. Disponível em: <https://mentaljournal-jspu.uz/index.php/mesmj/article/view/315> . Acesso em: 17 ago. 2025.

HALLINGER, Philip; WANG, Ray. The evolution of simulation-based learning across the disciplines, 1965-2018: a science map of the literature. **Simulation & Gaming**, v. 51, n. 1, p. 9-32, 2020. Disponível em: <https://doi.org/10.1177/1046878119888246>. Acesso em: 17 ago. 2025.

MCKENNEY, James L.; DILL, William R. Influences on learning in simulation games.

American Behavioral Scientist, v. 10, n. 2, p. 28-32, 1966. Disponível em:

<https://doi.org/10.1177/000276426601000205>. Acesso em: 17 ago. 2025.

WALKER, Andrew; LEARY, Heather. A problem based learning meta analysis: differences across problem types, implementation types, disciplines, and assessment levels.

Interdisciplinary Journal of Problem-Based Learning, v. 3, n. 1, p. 6, 2009. Disponível em:

<https://doi.org/10.7771/1541-5015.1061>. Acesso em: 17 ago. 2025.

WING, Jeannette M. Computational thinking. **Communications of the ACM**, v. 49, n. 3, p.

33-35, 2006. Disponível em: <https://doi.org/10.1145/1118178.1118215>. Acesso em: 17

ago. 2025.

WULANSARI, Ni Kadek Ayu Mita; SUARNI, Ni Ketut. Animation in science learning with brain based learning models to improve student learning outcomes. **International Journal of Elementary Education**, v. 4, n. 2, p. 160-168, 2020. Disponível em:

<https://doi.org/10.23887/ijee.v4i2.25244>. Acesso em: 17 ago. 2025.

ZHANG, Lu; MA, Yan. A study of the impact of project-based learning on student learning effects: a meta-analysis study. **Frontiers in Psychology**, v. 14, p. 1202728, 2023.

Disponível em: <https://doi.org/10.3389/fpsyg.2023.1202728>. Acesso em: 17 ago. 2025.

Sobre os autores

José Maurício Schneedorf Ferreira da Silva

Biólogo com doutorado direto em Bioquímica pela UFMG (Belo Horizonte, 1998; interação experimental ligante-proteína), e estágio pos-doc pela UFV (Viçosa-MG, 1999; termodinâmica de células tumorais). Professor titular, líder do grupo de pesquisa, e chefe do Departamento de Bioquímica da Universidade Federal de Alfenas (UNIFAL-MG), onde atua como docente em disciplina homônima em Cursos de Graduação e Pós-Graduação da Instituição. Temas de interesse: Ensino Reprodutível, e Bioeletroquímica.

Link do Lattes: <http://lattes.cnpq.br/0436922594542722>

Link do ORCID: <https://orcid.org/0000-0002-2031-6315>

E-mail: jose.dasilva@unifal-mg.edu.br

Licença de acesso livre



A **ESUD | CIESUD | SIGATEC** utiliza a [Licença Creative Commons - Atribuição 4.0 Internacional](#), pois acredita na importância do movimento do acesso aberto ao conhecimento.