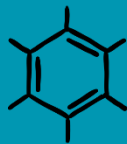


SÉRIE  
PAPO DE  
PROFESSORES



# PAPO DE **PROFESSORES**

Desafios, Inovações e Práticas  
no Ensino de Ciências



Eloísa Gerolin  
Umberto Crisafulli  
Org.



### Conselho Editorial

Profa. Dra. Marilena Rosalen	Prof. Dr. José Guilherme Franchi
Profa. Dra. Angela Martins Baeder	Prof. Dr. Luiz Afonso V. Figueiredo
Profa. Dra. Eunice Nunes	Prof. Dr. Flávio José M. Gonçalves
Profa. Dra. Luciana A. Farias	Prof. Dr. Giovano Candiani
Profa. Dra. Maria Célia S. Gonçalves	Prof. Me. Arnaldo Silva Junior
Profa. Dra. Rita C. Borges M. Amaral	Prof. Me. Pedro L. Castrillo Yagüe
Profa. Dra. Silvana Pasetto	Prof. Me. Everton Viesba-Garcia
Profa. Ma. Beatriz Milz	Profa. Ma. Letícia Moreira Viesba
Profa. Ma. Marta Angela Marcondes	Profa. Ma. Erika Brunelli

### Expediente

Coordenação Editorial: Everton Viesba-Garcia  
Coordenação de Área: Marilena Rosalen

### Organização

Organização: Eloisa Gerolin e Umberto Crisafulli

### Parecer e revisão por pares

Os textos que compõem esta obra foram submetidos para avaliação da Coordenação e/ou Conselho Editorial da V&V Editora, sendo aprovados na revisão por pares para publicação.

Dados Internacionais de Catalogação na Publicação (CIP)

x	Papo de Professores: Desafios, Inovações e Práticas no Ensino de Ciências. Eloisa Gerolin e Umberto Crisafulli (organizadores) – Santo André: V&V Editora, 2026. 316 p. : 14 x 21 cm  Inclui bibliografia ISBN 978-65-6063-129-8 DOI 10.47247/EG/6063.129.8  1. Um. 2. Dois. I. Três. II. Título.
	CDD x

Elaborado por Maurício Amormino Júnior – CRB6/2422

### V&V Editora

Santo André, São Paulo – Brasil  
Tel./Whatsapp: (11) 94019-0635 E-mail: contato@vveditora.com  
vveditora.com

**Alfabetização matemática por meios de jogos e materiais do cotidiano ..... 108**

Edilena Ferreira da Silva

**Concepções de Tecnologia nas Políticas Educacionais para o Ensino Médio: desafios para uma educação tecnológica ..... 118**

Elio Carlos Ricardo

**Curiosidade e encantamento: desvendando os insetos nas aulas de Ciências - como a entomologia pode ser uma atividade pedagógica valiosa de aproximação e conscientização sobre o meio ambiente..... 131**

Eloísa Torrezan, Gabriela Cristina Basso Engler

Marques e João Ângelo Cerigoni

**Proposta de Ensino de Ciências através da Abordagem Temática com Ênfase na Bacia do Rio Doce para Implementação da Educação Ambiental ..... 139**

Emilly Lorenzutti Borghi e Maria das Graças Ferreira

Lobino

**O Estágio Curricular Supervisionado como espaço de Formação Docente em Ciências Biológicas ..... 154**

Ismênia Gurgel Martins e Raquel Crosara Maia Leite

**Ecosocialismo ou Barbárie! Insegurança Alimentar e Falsas Soluções Verdes ..... 166**

Leandro Porto Marques, Bernardo Beck, Rafael Cabral

Cruz e Jefferson Marçal da Rocha

**Isótopos em Ação: Um jogo do gênero RPG dedicado ao ensino de Radioatividade..... 188**

Jhonathan Silva Menezes, Vitor Nader da Silva e Mari

Inez Tavarez

**Simulador e gerador de códigos por IA para gráficos interativos em browser e sua aplicação ao Ensino de Ciências..... 198**

José Maurício Schneedorf Ferreira da Silva

**Semeando educação, cultivando a sucessão: um olhar para as escolas do campo de Três Passos-RS..... 214**

Lediane Maria Feyth Schepp

---

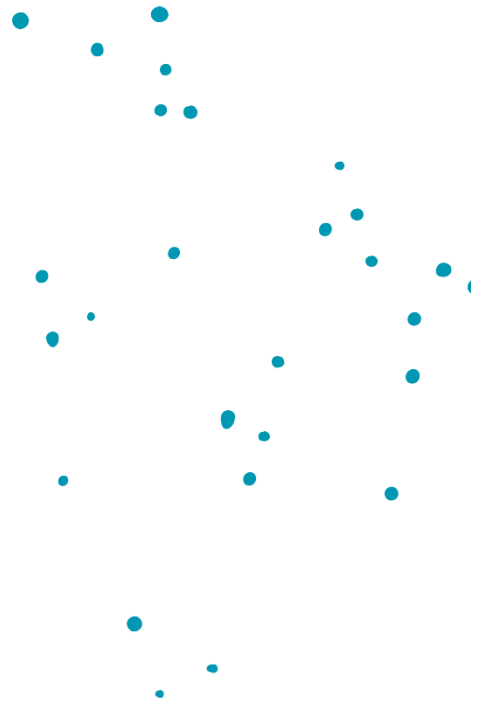
# Simulador e gerador de códigos por IA para gráficos interativos em browser e sua aplicação ao Ensino de Ciências

José Maurício Schneedorf Ferreira da Silva



10.47247/EG/6063.129.8.17

---



## Pensamento computacional e Educação 5.0

A partir das obras de Seymour Papert, “Twenty Things to Do With a Computer.” (Papert e Solomon, 1971), e “Mindstorms: Children, Computers And Powerful Ideas” (Papert, 1980), que sugeriram o computador como ferramenta para solução de problemas, bem como do trabalho seminal de 3 páginas de Jeanette Wing, que alçou o termo em 2006 (Wing, 2006), o pensamento computacional (“computational thinking”, PC) tem sido largamente abordado na literatura pertinente ao ensino e aprendizagem. O PC habilita o aprendiz à solução de situações-problema de modo lógico e estruturado, e resumidamente baseado em 4 pilares de sustentação (D’Alessandro e Neto, 2025):

# Pilares do pensamento computacional:

1. Abstração: leitura e identificação do que é ou não relevante ao problema;
2. Decomposição: divisão do problema em partes;
3. Reconhecimento de padrões: reflexão de procedimentos já utilizados em situações similares;
4. Algoritmos: delineamento sequencial da solução, automatizada ou não.

Obs: Decomposição e Reconhecimento podem agrupar-se como Análise.

Embora o trabalho de Wing dirija o PC como uma habilidade fundamental a executar-se em situações-problema comuns e mesmo *desplugadas* (como refazer um caminho para a busca de um objeto perdido...metaforicamente um *debug*), sua natureza ganha natural robustez quando se almeja lubrificar o caminho para a compreensão de conteúdos curriculares pelo uso de uma linguagem de programação. Nesse quesito, um cotidiano que se apresenta cada vez mais habitual à presença de novas tecnologias digitais imersivas, de aprendizado de máquina, internet das coisas e inteligência artificial, torna-se premente uma coevolução junto ao pensamento e recursos computacionais tangíveis à Educação Básica (Vieira *et al.*, 2023). Essa rota tecnológica, praticamente irreversível para uma *Educação 5.0*, vem sendo trabalhada mesmo fora da área junto às recomendações da União Europeia há mais de 20 anos [Digital Competence Framework for Educators - DigCompEdu](#).

Em princípio, pode-se aplicar o *PC* para qualquer área de atuação, reforçando-o como habilidade disputada já no presente mercado de trabalho (Sommer *et al.*, 2025). Tangente a esse, o uso de uma linguagem de programação “*de ponta*”, como *JavaScript*, empregada na criação de *websites* e de aplicativos interativos modernos, pode corroborar à uma formação técnico-científica alinhada às exigências do mercado de trabalho. A linguagem encontra-se na estrutura de bibliotecas como *React*, *Vue*, *Angular* e *Node.js*, sendo utilizada por empresas de porte, tanto como *frontend* (interface do usuário) quanto no *backend* (servidores, APIs). Nessa seara ilustram-se o ecossistema de ferramentas da Google (desenvolvedora do *Angular*), Facebook (Meta, *React*), Netflix (*Node.js*), e Microsoft (Visual Studio Code com *JavaScript/TypeScript* - Electron).

Contrariamente à sua versão desplugada, o *PC* pode convergir à estrutura de um algoritmo computacional, quando revestido de *lógica de programação*, base para o desenvolvimento de sistemas computacionais (Oliveira Junior *et al.*, 2025). Nesse quesito, um dos grandes desafios ao ensino de algoritmos envolve a conversão de uma linguagem natural falada/escrita para uma linguagem de programação, com sintaxe definida e rígida (quem alguma vez deixou de colocar um ponto onde devia num código-fonte de somente meia página, sabe a dor-de-cabeça que essa inflexibilidade sintática representa).

Pode-se visualizar essa migração da língua falada para uma estrutura de algoritmo com lógica de programação pelo auxílio de algumas ferramentas de código aberto. Entre as que as que facilitam o raciocínio para códigos, sem o lastro de uma linguagem de programação específica, vale mencionar VisualAlg e PortugolStudio.

Em terras brasilienses, o *PC* foi explicitamente inserido no complemento normativo de 2018 da Base Nacional Comum Curricular (BNCC) para Ciências da Computação, seguindo-se o que já fora concretizado desde a década passada para os nove primeiros anos da educação básica (K-9) em diversos países da Europa e os EUA (Silva e Maltempi, 2020). Além disso, as Diretrizes Curriculares Nacionais para a Formação Inicial de Professores para a Educação Básica preconizam, no parágrafo único de seu 12o. artigo, a “*compreensão básica dos fenômenos digitais e do pensamento computacional, bem como de suas implicações nos processos de ensino-aprendizagem na contemporaneidade*”. (Brasil, 2019). Essas ações foram posteriormente normatizadas em outubro de 2022 pela Câmara da Educação Básica junto às Normas sobre Computação na Educação Básica - Complemento à BNCC, e pelo parecer CNE/CEB Nº: 2/2022. Ao ano seguinte, o artigo 3o. da Lei no. 14.533 sobre

Política Nacional de Educação Digital - PNED) consolidou o PC junto às estratégias prioritárias de seu *eixo de inclusão digital*, desde a educação infantil até o ensino médio.

## Gráficos e modelagem matemática

Dentre os vários desafios enfrentados pelo aprendiz em Ciências, no entendimento de relações quantitativas concretizáveis com o uso de recursos digitais, encontram-se os fenômenos naturais representados por funções matemáticas. São diversas as situações em que funções, gráficos, movimentos, e reações, por exemplo, são representados apenas em imagens estáticas em livros didáticos. Ainda que possam agregar valor como linguagem matemática complementar à visualização, o conjunto dessas representações pode gerar certo desconforto ao aprendiz, frente a um conjunto de dificuldades oriundas do nível simbólico de *metavisualização* necessária à sua interpretação: adquirir, monitorar, integrar e ampliar a aprendizagem pelo gráfico respresentado (Gilbert, 2008). Na prática, afloram-se problemas como:

# Dificuldades de metavisualização para interpretação gráfica em Ciências:

1. quantitativo de funções matemáticas para fenômenos naturais em livros didáticos;
2. literacia gráfica exigida para relações quantitativas entre variáveis;
3. trânsito nem sempre suave entre a linguagem matemática e sua visualização;
4. contextualização por vezes complexa para fenômenos naturais em linguagem gráfica.

Embora haja um número expressivo de publicações voltadas à visualização gráfica para funções matemáticas no ensino básico, com foco em conteúdos como Álgebra, Funções e Geometria, o fiel da balança não tem se equilibrado em igual monta às Ciências naturais nesse nível de ensino. Não sendo escopo deste texto, há razões em aberto para discussão do problema, como a separação curricular desses dois domínios de conhecimento, a falta de formação docente específica para integrar ferramentas tecnológicas no ensino de Ciências, bem como a escassez de recursos didáticos que promovam a modelagem matemática para fenômenos científicos.

## Modelagem matemática em Ciências

A modelagem matemática permite uma versão idealizada da situação do mundo real que pode então ser traduzida em termos matemáticos. Nesse momento, o modelo matemático é obtido pela elaboração mental de um modelo explicativo, quando se substitui a linguagem natural das hipóteses por uma linguagem matemática coerente (Souza et al., 2017). É bem estabelecido que a modelagem matemática com uso de recursos digitais possa contribuir fortemente à *alfabetização científica* no ensino de Ciências, embora as dificuldades acima possam contrapor-se à sua concretização.

Há uma diferença crítica entre aplicar a matemática para cálculos ou tabulações de dados científicos, e aplicar a *matemática de processo modelado* (Schuchardt e Schunn, 2016). Enquanto a primeira utiliza conceitos, símbolos, e métodos matemáticos de modo inespecífico a um conjunto de dados, o processo modelado possibilita internalizar com maior facilidade os conceitos científicos da situação-problema. E isso agrega considerável valor, elevando a habilidade do aprendiz na solução de problemas de modelagem pela prévia experiência, e com as ferramentas adequadas (Schuchardt e Schunn, 2016).

Vários são os recursos computacionais para modelagem matemática e construção de gráficos. Independentemente da ferramenta utilizada, a modelagem auxiliada por esses pode permitir ao estudante a representação gráfica de relações entre causa e efeito equacionáveis em uma situação-problema em Ciências. Exemplos são inúmeros, mas basta visualizar uma relação matemática qualquer na área, como a presente na equação geral de gases ideais, na taxa de decaimento exponencial de uma reação química no tempo, na relação de força e deslocamento da 2a. Lei de Newton, na função de descarga de um capacitor, no comportamento cinético de uma enzima sobre seu substrato, e assim por diante.

Nesses e em infinitos outros “casos”, uma modelagem matemática que permita a visualização gráfica do comportamento das variáveis envolvidas numa equação, ou seja, uma simulação gráfica, pode gemular diversas vantagens, tais como facultar:

# Vantagens do uso de simulações gráficas ao ensino de fenômenos naturais:

1. A observação direta das mudanças entre variável(eis) preditora(s) e a variável resposta do modelo;
2. O acompanhamento de covariação da resposta ao longo de uma variável preditora (ex: tendências)

sigmoidais, exponenciais, hiperbólicas);

3. A observação de limites da natureza própria do fenômeno, ou de seu sentido físico, quando contrastado ao modelo matemático (ex: eixo com valores negativos);
4. A visualização do resultado de uma mudança num parâmetro do modelo (“exploração ou manipulação paramétrica”);
5. A visualização do resultado de mudanças sucessivas em um parâmetro, por sobreposição das tendências gráficas da variável resposta (curvas sobrepostas);
6. Um rearranjo para alteração nos membros da equação, visando obter-se uma variável preditora distinta da inicialmente proposta e, nesse sentido, aprimorando-se também as noções matemáticas;
7. Uma apropriação gradual da informação contida em gráficos de funções para situações-problema em conteúdos didáticos de Matemática e Ciências;
8. Uma compreensão progressiva do fenômeno modelado;
9. A possibilidade de conduzir "experimentos virtuais", com simulações sequenciais para uma dada proposta, e com conseqüente economia múltipla de infraestrutura, equipe, instrumentação, e tempo;
10. Atividades de maior complexidade, possibilitando uma aprendizagem baseada em problemas e em projetos;
11. Uma aprendizagem mais dinâmica e mesmo lúdica, complementando as imagens fixas de livros didáticos;
12. A possibilidade de serem aplicadas diretamente em conteúdos temáticos da grade curricular;
13. Seu potencial para uso em qualquer modalidade de ensino (presencial, remoto, EaD);
14. Uma apropriação mais efetiva e significativa do tema abordado.

Assim, simulações gráficas podem facilitar, e muito, a compreensão de um tema normalmente abstrato durante a alfabetização científica que permeia os últimos anos do ensino fundamental e o ensino médio ou tecnológico (vide [Geogebra](#) e [PhET Interactive Simulations](#), por ex). Além disso, simulações gráficas mediadas por recursos digitais trazem consigo a aquisição e aprimoramento para diversas competências digitais, correlatas ao uso de hardware, software, *inclusão computacional*, e *aprendizagem colaborativa* (Oliveira Junior; Bortoli; Castaman, 2025). Além disso, simulações gráficas simples podem estimular o aprendiz à busca de visualizações mais complexas, tais como as simulações 3D preconizadas pelo *Boston Consulting Group* para balizar a indústria 4.0 (Rüßmann *et al.*, 2015), juntamente com *big data*,

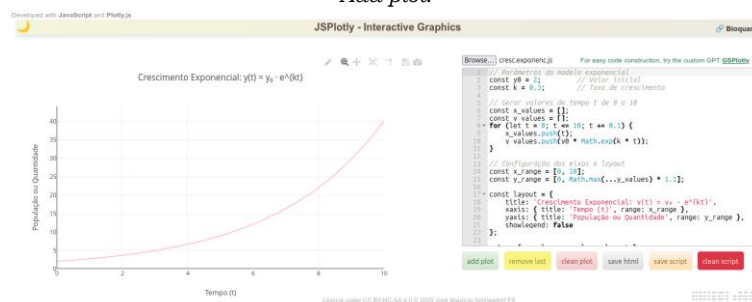
cibersegurança, computação em nuvem, internet das coisas, impressão 3D, automação robótica, e realidade aumentada.

Mediante essa possibilidade de acolhimento para visualização e estudo de fenômenos naturais em Ciências modelados matematicamente para simulação gráfica, este trabalho apresenta um aplicativo prático e seu gerador de códigos por inteligência artificial (IA).

## JSPlotly & GSPlotly: simulador e gerador de códigos para simulações gráficas em Ciências

JSPlotly é um aplicativo desenvolvido para permitir a construção de gráficos interativos em 2D e 3D, com base em equações matemáticas ou dados fornecidos pelo usuário. Elaborado em HTML (conteúdo), CSS (estilo) e *JavaScript* (interatividade), e utilizando a biblioteca *Plotly.js*, o aplicativo funciona diretamente em navegadores modernos, sem necessidade de instalação, conexão com a internet ou requisitos computacionais específicos. Seu código-fonte possui menos de 25 kB (alguém se lembra do disquete de 5 e 1/4 polegadas com 360 kB de capacidade dos anos 80 ?!), e pode ser modificado em qualquer editor de texto, o que favorece sua personalização e compartilhamento fácil, mesmo por redes sociais. A interface foi organizada para uma única janela, dividida entre *editor de código* e área gráfica de visualização interativa, e representados na [Figura 1](#).

Figura 1: Tela única do *JSPlotly*, apresentando o editor de códigos (direita), e o ecrã gráfico (esquerda). O código inserido no editor à direita é interpretado pelo programa, renderizando o gráfico à esquerda pelo botão *Add plot*.



Fonte: Autor.

Entre as principais funcionalidades do *JSPlotly*, destacam-se a inserção de dados ou equação pelo usuário, edição de funções matemáticas, manipulação de parâmetros e comandos,

sobreposição de gráficos, alternância de cores, exclusão de curvas, exportação do gráfico em diferentes formatos (HTML interativo, PNG, SVG), e edição na nuvem por software do desenvolvedor da biblioteca gráfica, [Plotly Chart Studio](#). O usuário também pode interagir com o gráfico por meio de recursos como *hover* (passagem de mouse para *popup* de dados), escalonamento, e movimentação de eixos, com controles acessíveis por botões numa barra de ferramentas.

A interatividade é fornecida pela própria biblioteca *Plotly.js*, que suporta geração de gráficos dinâmicos e mapas, integração com *tooltips* (janela *popup* para dados adicionais em cada ponto), e seleção de pontos. Recursos adicionais como *sliders* (deslizadores), menus interativos, *facetas* (gráficos múltiplos no mesmo ecrã gráfico) e *dashboards* (painéis interativos) podem ser incorporados por meio de arquivos adaptados, estendendo as possibilidades do aplicativo. O *JSPlotly* oportuniza, assim, explorar a relação entre parâmetros e comportamento gráfico de forma direta, podendo ser utilizado como ferramenta complementar no desenvolvimento de competências digitais, na introdução à programação, como para temas diversos em qualquer área.

Como um *recurso educacional aberto digital*, a aplicação está disponível no site [Bioquanti](#), uma iniciativa voltada ao *ensino reprodutível* de códigos para conteúdos didáticos. Detalhes sobre o *ensino reprodutível* e sua aplicação diretamente em conteúdos do ensino básico podem ser observados junto à obra [Vivificando Conteúdos Para o Ensino Básico: Moléculas, Gráficos, e Mapas Interativos](#) (Schneedorf, 2025c). Originalmente concebido ao ensino-aprendizagem para o ensino superior em Bioquímica, o site é dedicado hoje também ao ensino básico. Em síntese o *Bioquanti* busca integrar o aprendizado e aplicação autoinstrucionais para algumas ferramentas digitais abertas para fins didáticos e científicos, tais como aplicativos, ebooks, e objetos educacionais interativos.

Tangente ao *JSPlotly*, o site apresenta uma descrição, vídeo de apresentação, funcionalidades, e modo de uso. Além disso, contempla diversos gráficos interativos gerados no *JSPlotly* junto a seus códigos, tanto para o ensino superior como para o ensino básico. Para acessar um gráfico no site, basta selecionar um tema e clicar na imagem que acompanha sua descrição e equação. Maiores detalhes da construção e uso da aplicação, bem como comparações com outras soluções tecnológicas existentes, são complementadas [neste capítulo](#) (Schneedorf, 2025b). Adicionalmente, uma aplicação direta do *JSPlotly* ao nível superior em Biofísico-química é também apresentada pelo autor em outro texto (Schneedorf, 2025a).

O *JSPlotly* elabora os gráficos interativos utilizando-se a linguagem de programação JavaScript. Como qualquer outra, possui uma sintaxe própria dentro de sua lógica de programação, para declaração de variáveis, entradas e saídas, laços iterativos, entre outros. Por óbvio, nenhuma linguagem de programação é realmente atrativa, estimulante ou catártica, à exceção de um discutível afeto a ela dedicado por poucos (ou loucos!), ou de higidez discutível. Para contornar essa situação, foi elaborado um gerador de códigos por inteligência artificial (IA), permitindo *cópia e cola* desses diretamente no editor do *JSPlotly* para execução.

O gerador de códigos, GSPlotly, foi desenvolvido como um GPT personalizado para a criação automática e instantânea de códigos para gráficos interativos no *JSPlotly*. Para seu uso, basta acessar o [link](#), e conduzir uma proposta para um gráfico interativo para qualquer área em linguagem natural, como normalmente se opera junto às IAs generativas, por engenharia de *prompts*. O *GSPlotly* retorna um código-fonte para simples cópia e cola no editor de códigos do *JSPlotly*. Complementarmente, o GPT personalizado foi desenvolvido para permitir uma breve introdução ao tema desejado, bem como para fornecer a função (ou funções) matemática que o envolve e sua descrição. Tenta lá ! É divertido !

Como um todo, acredita-se que o simulador (*JSPlotly*) e seu gerador de códigos (*GSPlotly*) configurem uma plataforma educacional interativa, com potencial para agregar valor junto a uma aprendizagem significativa por construção ativa de conhecimento e *feedback* imediato auxiliados por uma estrutura cognitiva prévia do aprendiz (como sucedente à abordagem teórica em sala de aula, por ex) (Moreira e Masini, 2009). Dessa forma, a plataforma deve permitir que qualquer usuário possa utilizá-la para a criação de gráficos e simulações interativas ao ensino, e particularmente instigando-o:

# Vantagens aparentes para uso do simulador e gerador de códigos a gráficos interativos:

1. À observação e uso imediato de uma linguagem de programação, embora sem conhecimento prévio algum em lógica homônima ou sintaxe;
2. Ao aprendizado de lógica de programação de forma direcionada a conteúdos didáticos de textos afins;
3. Ao aprendizado da linguagem "JavaScript" que envolve a aplicação, por degustação junto ao editor de códigos;

4. Ao aprendizado dos códigos propriamente ditos, e referentes a um conteúdo didático (ou não) desejado;
5. A uma apropriação ativa, interativa, e gradual, para determinado tema da grade curricular (ou não);
6. À experimentação de novos gráficos, simulações e experimentos virtuais, em função de um ciclo virtuoso inerente ao "ensino reprodutível" (reprodução, modificação, e recriação de códigos para conteúdos).

## Um exemplo de uso do JSPlotly e GSPlotly em Ciências

Para exemplificar o potencial do simulador e gerador de códigos em Ciências, buscou-se ilustrar graficamente o conceito para *movimento retilíneo uniformemente variado* (outros exemplos ao ensino básico com uso da plataforma podem ser observados junto ao site [Bioquanti](#)).

Assim, uma única pergunta foi direcionada ao GPT personalizado:

"Me forneça um código para movimento uniformemente variado ao ensino fundamental".

Em função do prévio treinamento junto ao assistente de IA, sua informação de *saída* envolveu 1) uma rápida descrição do tema, 2) a equação geral envolvida, e 3) o código *JavaScript* para o gráfico interativo, além de 4) sugestões de incrementos ou modificações em torno do tema.

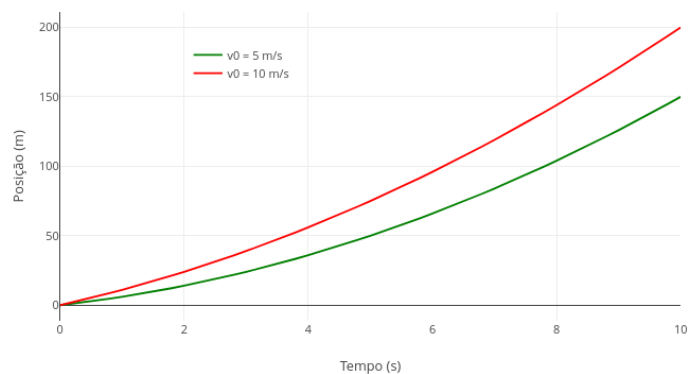
Assim, a partir da equação abaixo e do código fornecido pelo *GSPlotly*, foi possível elaborar o gráfico da [Figura 2](#).

$$S(t) = S_0 + v_0 \cdot t + \frac{1}{2}a \cdot t^2$$

Onde:  $S(t)$  = posição no tempo ( $t$ ), em metros;  $S_0$  = posição inicial (m);  $v_0$  = velocidade inicial (m/s);  $t$  = tempo (s);  $-a$  = aceleração (m/s<sup>2</sup>);

Figura 2: Movimento uniformemente variado, como um exemplo em Ciências para uso do *JSPlotly*, e com código gerado pelo *GSPlotly*. As curvas referem-se à *exploração paramétrica* resultante de uma alteração pontual no valor da velocidade inicial fornecida pela assistente de IA ( $v_0 = 5 \text{ m/s}$ ).

Posição em função do tempo (MUV) — BNCC: EF09CI03



Fonte: Autor.

O código gerado pelo *GSPlotly* é fornecido abaixo.

```
// Parâmetros do movimento
const So = 0; // posição inicial (m)
const vo = 5; // velocidade inicial (m/s)
const a = 2; // aceleração constante (m/s²)

// Tempo de 0 a 10 segundos
const x_values = [ ];
const y_values = [ ];
for (let t = 0; t <= 10; t += 0.2) {
  x_values.push(t);
  const S = So + vo * t + 0.5 * a * Math.pow(t, 2);
  y_values.push(S);
}

// Configuração do gráfico
const layout = {
  title: "Posição em função do tempo (MUV)",
  xaxis: { title: "Tempo (s)" },
  yaxis: { title: "Posição (m)" },
  showlegend: false
};

return { x_values, y_values, layout };
```

Perceba que, independentemente de conhecimento em programação ou de sua lógica, uma simples alteração no valor da velocidade inicial ( $v_0$ ) permite ao aprendiz agregar conhecimento adicional sobre o comportamento da função. Por conseguinte, também conduz a uma melhor compreensão do fenômeno envolvido na situação-problema, por ajustes em algum parâmetro da equação, possibilitando uma *exploração ou manipulação paramétrica* dirigida ao aprendizado do tema (Daugherty, 2008).

Complementarmente, o aprendiz pode exercitar alterações também na equação, recombinação seus termos para outra variável preditora na busca de informação complementar. Isto possibilita aprimorar seu grau de abstração em rearranjos de expressões matemáticas. O aprendiz também pode alterar a própria sintaxe de programação envolvida, nesse caso intencionando melhorias por inclusão de novos comandos da linguagem. E por conseguinte, também aprimorando seus conhecimentos em uma linguagem de programação diretamente voltada a uma situação específica da matriz curricular de Ciências ou de Matemática.

Finalmente, embora longe de esgotar as possibilidades, o *JSPlotly* pode ser também empregado para *interpretação gráfica de soluções matemáticas*, associadas ou não à Ciências. Exemplos referentes à última incluem a identificação de pontos de intersecção de retas ou curvas (identificação de pontos de equilíbrio térmico entre substâncias), a visualização de regiões que satisfazem inequações ou sistemas que as envolvem (análise de intervalos em que uma variável ambiental atinge condições ideais), e a interpretação de valores trigonométricos com apoio visual (estimativa de ângulos ou relações espaciais em fenômenos físicos, como a incidência solar).

## Lógica de programação

O trecho de código fornecido acima não é trivial à interpretação de um usuário comum. Contudo, pode-se facilitar sua decodificação por meio de *lógica de programação*, ilustrada abaixo para um *pseudocódigo* em [Portugol](#). Dessa forma, o código pode ser compreendido pelo *raciocínio algorítmico* que envolve o pensamento computacional *plugado*, como segue:

```

// Definir os parâmetros do movimento
So ← 0 // posição inicial (em metros)
vo ← 5 // velocidade inicial (m/s)
a ← 2 // aceleração constante (m/s²)

// Criar listas para armazenar os valores de tempo e posição
x_values ← lista vazia // tempo
y_values ← lista vazia // posição

// Calcular a posição para cada instante de tempo de 0 a 10
segundos
para t de 0 até 10 com passo 0.2 faça:
  adicionar t em x_values
  S ← So + vo * t + (1/2) * a * t²
  adicionar S em y_values
fim para

// Preparar a saída dos dados (para visualização posterior)
retornar x_values, y_values

```

## Conclusão

Das inúmeras dificuldades que se contrapõem ao ensino de Ciências, uma tange à interpretação de fenômenos físicos, químicos ou biológicos, e trazidos à tona por intermédio de gráficos de funções matemáticas. Para contribuir à redução do desconforto estudantil apreciado em temas dessa natureza, este trabalho apresenta *JSPlotly* como um simulador para gráficos interativos de funções e dados, bem como *GSPlotly*, seu gerador de códigos por IA para linguagem *JavaScript* em GPT personalizado. Entre as características dessa plataforma, destacam-se, por um lado, a dispensa de conhecimento em programação requerida para a confecção de um conjunto virtualmente infinito de códigos focados em possibilidades didáticas pelo assistente de IA. Por outro lado, as funcionalidades do simulador, e que incluem o potencial para gráficos interativos em 2D e 3D instantaneamente gerados para uso em navegador offline, compartilháveis em qualquer meio, físico ou virtual, bem como para uso em ambientes virtuais de aprendizagem. No conjunto, a solução tecnológica intenciona agregar valor ao ensino e aprendizagem em Ciências como em Matemática, ilustrando diversos fenômenos naturais presentes em textos curriculares afins por gráficos e simulações interativas, e ativamente promovidas pelo estudante ou professor.

## A elaboração deste capítulo

A formatação de texto (*RMarkdown*), inserção e referência cruzada de figuras, hiperlinks, o gerenciamento bibliográfico, e a conversão do *documento dinâmico* final para compilação a um modelo de arquivo *DOCX* previamente configurado, foram exclusivamente conduzidos com a linguagem de programação *R* (versão 4.3.3, fev/2024) em ambiente de desenvolvimento integrado *RStudio* (versão 2024.09.1 Build 394), e pacote *quarto* (versão 1.4.4) como sistema de publicação científica.

## Agradecimentos

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

## Referências

BRASIL. Resolução CNE/CP N° 2, de 20 de dezembro de 2019: Diretrizes Curriculares Nacionais para a Formação Inicial de Professores para a Educação Básica (BNC-Formação). <https://bit.ly/3oCFt7M>, 2019.

D’ALESSANDRO, J. C. A.; NETO, R. D. Estudo sobre o uso do Pensamento Computacional na resolução de questões de Olimpíadas de Matemática. *Revista Eletrônica de Educação Matemática*, [s. l.], vol. 20, p. 1–19, 2025.

DAUGHERTY, S. C. A Study of Visualization for Mathematics Education. *Em.*, 2008. MODSIM World 2007 Conference and Expo: Select Papers and Presentations from the Education and Training Track. [S. l.: s. n.], 2008.

GILBERT, J. K. Visualization: An emergent field of practice and enquiry in science education. [S. l.]: Springer, 2008. p. 3–24.

MOREIRA, M. A.; MASINI, E. A. F. S. Aprendizagem significativa: a teoria de David Ausubel. [s. l.], 2009.

OLIVEIRA JUNIOR, E. R. de; BORTOLI, L. Â. de; CASTAMAN, A. S. Confraria da Lógica: uma organização educacional para potencializar o pensamento computacional entre estudantes da

Educação Profissional e Tecnológica. Caderno Pedagógico, [s. l.], vol. 22, n.º 1, p. e13548–e13548, 2025.

PAPERT, S. Children, computers, and powerful ideas. [S. l.]: Harvester Eugene, OR, USA, 1980. vol. 10

PAPERT, S.; SOLOMON, C. Twenty Things to Do With a Computer. Artificial Intelligence Memo Number 248. [s. l.], 1971.

RÜSSMANN, M. *et al.* Industry 4.0: The future of productivity and growth in manufacturing industries. Boston consulting group, [s. l.], vol. 9, n.º 1, p. 54–89, 2015.

SCHNEEDORF, J. M. JSPlotly: um aplicativo para gráficos interativos em Bioquímica para visualizadores HTML. Revista de Ensino de Bioquímica, [s. l.], 2025a.

SCHNEEDORF, J. M. JSPlotly: um aplicativo prático e portátil para simulações gráficas interativas e dinâmicas ao ensino reprodutível. *Em: Recursos Educacionais Abertos: compartilhando experiências na Universidade Federal de Alfenas (UNIFAL-MG)*. 1. ed. São Paulo: Pimenta Cultural, 2025b. vol. 1, p. 72–90.

SCHNEEDORF, J. M. Vivificando Conteúdos Para o Ensino Básico: Moléculas, Gráficos, e Mapas Interativos. São Paulo: Pimenta Cultural, 2025c. p. 132 Disponível em: <https://www.pimentacultural.com/livro/vivificando-conteudos/>.

SCHUCHARDT, A. M.; SCHUNN, C. D. Modeling scientific processes with mathematics equations enhances student qualitative conceptual understanding and quantitative problem solving. *Science Education*, [s. l.], vol. 100, n.º 2, p. 290–320, 2016.

SILVA BARBOSA, L. L. da; MALTEMPI, M. V. Matemática, Pensamento Computacional e BNCC: desafios e potencialidades dos projetos de ensino e das tecnologias na formação inicial de professores. *Revista Brasileira de Ensino de Ciências e Matemática*, [s. l.], vol. 3, n.º 3, 2020.

SOMMER, E. *et al.* Explorando tecnologias para fomentar habilidades do pensamento computacional na educação básica: uma revisão sistemática. *Revista Ibero-Americana de Humanidades, Ciências e Educação*, [s. l.], vol. 11, n.º 3, p. 1253–1269, 2025.

SOUZA, E. S. R. de; ESPÍRITO SANTO, A. O. do *et al.* Alfabetização científica em ambiente de modelagem matemática. REAMEC-Rede Amazônica de Educação em Ciências e Matemática, [s. l.], 2017.

VIEIRA, R. *et al.* Society 5.0 and Education 5.0: A critical reflection. *Em: SOCIETY 5.0 AND EDUCATION 5.0, 2023. 2023 18th Iberian Conference on Information Systems and Technologies (CISTI).* [S. l.]: IEEE, 2023. p. 1-6.

WING, J. M. Computational thinking. *Communications of the ACM*, [s. l.], vol. 49, n.º 3, p. 33-35, 2006.