CAPÍTULO 15

O UNIVERSO EM SUAS MÃOS: PROTÓTIPO DE APLICATIVO EDUCACIONAL COM INTELIGÊNCIA ARTIFICIAL GENERATIVA INTEGRADA VOLTADO PARA O ENSINO DE FÍSICA

https://doi.org/10.22533/at.ed.1661125170315

Data de aceite: 15/08/2025

Rodrigo Cesar Fonseca da Silva
Universidade Estadual da Paraíba
João Pessoa – PB

Elder Eldervitch Carneiro de Oliveira
Universidade Estadual da Paraíba
João Pessoa – PB

Leandro de Almeida Melo
Universidade de Pernambuco
Nazaré da Mata - PE

Pedro Carlos de Assis Júnior
Universidade Estadual da Paraíba
Patos – PB

Any Caroliny D. Batista de Almeida
Universidade de Pernambuco
Nazaré da Mata - PE

RESUMO: Este artigo científico apresenta o desenvolvimento de um aplicativo gratuito e inovador, baseado em Inteligência Artificial (IA) generativa, voltado para professores de Física na criação eficiente de material didático para uso em sala de aula. A IA generativa viabiliza a produção automatizada de recursos pedagógicos personalizados e atrativos, como exercícios, explicações conceituais, exemplos práticos e visualizações, adaptados às necessidades dos estudantes e aos conteúdos curriculares. A gratuidade do aplicativo promove a democratização do acesso a materiais educacionais de alta qualidade, beneficiando docentes com recursos financeiros limitados. Sua interface intuitiva e a geração de conteúdo sob demanda otimizam o tempo de planejamento, favorecendo a personalização do ensino e a intensificação da interação com os discentes. Espera-se que essa inovação aumente o engajamento estudantil, facilite a compreensão de conceitos complexos de Física e contribua para a melhoria do desempenho acadêmico. Em síntese, o aplicativo representa um avanço significativo no ensino de Física, ao fornecer aos docentes uma ferramenta tecnológica acessível e eficaz.

PALAVRAS-CHAVE: Aplicativo, Inteligência Artificial Generativa, Ensino de Física.

THE UNIVERSE IN YOUR HANDS: PROTOTYPE OF AN EDUCATIONAL APPLICATION WITH INTEGRATED GENERATIVE ARTIFICIAL INTELLIGENCE AIMED AT TEACHING PHYSICS

ABSTRACT: This scientific article presents the development of a free and innovative application, based on generative Artificial Intelligence (AI), aimed at physics teachers in the efficient creation of teaching materials for use in the classroom. Generative AI enables the automated production of personalized and engaging pedagogical resources, such as exercises, conceptual explanations, practical examples, and visualizations, tailored to students' needs and curricular content. The free application promotes democratization of access to high-quality educational materials, benefiting teachers with limited financial resources. Its intuitive interface and on-demand content generation optimize planning time, favoring personalized teaching and intensifying interaction with students. This innovation is expected to increase student engagement, facilitate the understanding of complex physics concepts, and contribute to improved academic performance. In summary, the application represents a significant advance in physics education by providing teachers with an accessible and effective technological tool.
KEYWORDS: Application, Generative Artificial Intelligence, Physics Teaching.

INTRODUÇÃO: A GÊNESE DA INTELIGÊNCIA ARTIFICIAL GENERATIVA

O surgimento da Inteligência Artificial (IA) generativa representa uma das mais marcantes transformações no campo da ciência da computação. Sua gênese remonta às primeiras tentativas de simular o raciocínio humano por meio de algoritmos formais e sistemas baseados em regras. Tais experimentos, embora rudimentares frente às capacidades contemporâneas, estabeleceram fundamentos importantes para o desenvolvimento de tecnologias capazes de produzir conteúdo original de forma autônoma (CCREVIER, 1993). Durante as décadas de 1950 e 1960, os esforços em Processamento de Linguagem Natural (PLN) estavam concentrados na compreensão e tradução de textos. No entanto, esses estudos fomentaram o desenvolvimento de mecanismos de geração textual. Um exemplo seminal é o ELIZA, sistema criado por Joseph Weizenbaum no *Massachusetts Institute of Technology* (MIT), que simulava um psicoterapeuta Rogeriano ao aplicar regras sintáticas em interações textuais com usuários (WEIZENBAUM, 1966).

Nos anos 1980, a emergência dos sistemas especialistas e das Redes Neurais Artificiais (RNAs), dentro da vertente conexionista da IA, trouxe novas abordagens para o aprendizado de representações complexas de dados. Ainda que não focadas diretamente na geração, essas técnicas foram cruciais para o surgimento de aplicações criativas, como os primeiros experimentos em geração musical e artística por algoritmos baseados em regras e estruturas procedurais.

O verdadeiro ponto de inflexão ocorreu no século XXI, com o avanço das técnicas de aprendizado profundo (*deep learning*). A introdução das Redes Neurais Recorrentes (RNNs), capazes de lidar com dados sequenciais, possibilitou progressos substanciais na

geração de texto, áudio e música. Posteriormente, as Redes Transformer, introduzidas por Vaswani et al. (2017), superaram as limitações das RNNs ao permitir o processamento paralelo de sequências longas, transformando-se na arquitetura dominante em modelos de linguagem natural.

Neste contexto, destaca-se o modelo GPT (*Generative Pre-trained Transformer*), desenvolvido pela OpenAI, que demonstrou uma capacidade sem precedentes de gerar texto coerente, contextualizado e relevante a partir de comandos mínimos. O GPT-3, em particular, consolidou essa abordagem, ao ser treinado com centenas de bilhões de parâmetros e demonstrar notável versatilidade em tarefas Brown et al. (2020).Outro avanço notório foi o surgimento das Redes Geradoras Adversariais (GANs), concebidas por Ian Goodfellow e colaboradores. As GANs operam com dois modelos neurais, o gerador e o discriminador, que se retroalimentam em um processo competitivo, produzindo dados sintéticos de alta fidelidade. Essa tecnologia tem sido amplamente aplicada na geração de imagens realistas, vídeos, música e até reconstruções 3D. Mais recentemente, a IA generativa tem se expandido para modelos multimodais, capazes de integrar e gerar simultaneamente texto, imagens, áudio e vídeo. Exemplos notáveis incluem sistemas como DALL·E e Sora, que ampliam os limites da criatividade computacional. A abundância de dados e a crescente capacidade computacional contribuem para o avanço desses sistemas, promovendo aplicações em áreas tão diversas quanto educação, saúde, design, pesquisa científica e entretenimento.

Assim, a trajetória histórica da IA generativa reflete a confluência de avanços teóricos e tecnológicos, culminando em sistemas cada vez mais integrados, autônomos e criativos, com impactos crescentes sobre a forma como humanos produzem, consomem e interagem com a informação. Dentro da perspectiva inovadora, de associar a IA generativa às ferramentas tradicionais de Ensino de Física, apresentamos neste trabalho um aplicativo educacional chamado O Universo em Suas Mãos, capaz de produzir material didático (imagens exclusivas com texto) para ser utilizado por professores de Física, com grande potencial a ser explorado. Além da Introdução, o artigo está dividido da seguinte forma: A seção 2 mostra um breve histórico sobre o Flutter e Linguagem Dart; na seção 3, temos a convergência entre Flutter e Flutterflow; a seção 4 é voltada para descrição das chamadas de API; A seção 5 mostra a programação por trás do Aplicativo Educacional: O Universo em Suas Mãos; na seção 6 é realizada a análise de resultado e discussões.

A ASCENSÃO DO FLUTTER E DA LINGUAGEM DART: UMA PERSPECTIVA HISTÓRICA

O surgimento do Flutter e da linguagem de programação Dart[1] está diretamente relacionado à estratégia da Google de fornecer ferramentas capazes de criar interfaces de usuário (UIs) modernas, responsivas e com desempenho nativo em múltiplas plataformas.

1 https://dart.dev/

Anunciado publicamente em 2015, o Flutter foi concebido como uma solução inovadora para superar as limitações dos frameworks híbridos tradicionais, que, ao dependerem de tecnologias web como HTML, CSS e JavaScript, frequentemente apresentavam desempenho inferior e experiências inconsistentes entre plataformas (IBRAHIM, 2019). Até então, o desenvolvimento mobile demandava a criação de códigos distintos para Android e iOS, resultando em custos elevados e baixa eficiência. As soluções híbridas da época priorizavam a portabilidade em detrimento da performance, o que motivou a Google a propor uma abordagem alternativa. O Flutter introduziu um motor próprio de renderização, o Skia, que permite o desenho direto dos elementos gráficos na tela, conferindo ao desenvolvedor controle preciso sobre cada pixel da interface e garantindo um desempenho próximo ao nativo (JOSHI, 2020).

A linguagem Dart, também criada pela Google e lançada oficialmente em 2011, foi inicialmente pensada como uma alternativa ao JavaScript para aplicações web. Apesar da adoção limitada em sua fase inicial, Dart revelou-se ideal para o Flutter devido à sua arquitetura orientada a objetos, tipagem estática opcional e suporte a recursos como *hot reload*, funcionalidade que possibilita a atualização instantânea da interface em tempo real durante o desenvolvimento sem a necessidade de reinicializar a aplicação (SOSEMAN, B; MORGAN, K., 2019). Um exemplo, que pode ser visto na Figura 1, que ilustra a simplicidade e clareza sintática da linguagem Dart no contexto do Flutter é a criação de um *widget* de texto.

```dart
import 'package:flutter/material.dart';

class MeuTexto extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Text(
      'Olá, Flutter!',
      style: TextStyle(fontSize: 24.0),
    );
  }
}
```

Figura 1 – Exemplo de código na linguagem DART no contexto do Flutter.

Este trecho do código evidencia como Dart permite, com poucas linhas, a construção de componentes de interface visualmente ricos e funcionais. A arquitetura baseada em *widgets* do Flutter, fundamentada na composição e imutabilidade, facilita o desenvolvimento declarativo e altamente personalizável de interfaces. Desde seu lançamento, o Flutter evoluiu significativamente. Inicialmente focado no desenvolvimento para Android e iOS, passou a oferecer suporte a web, desktop (Windows, macOS e Linux) e até sistemas embarcados. Essa versatilidade, expressa no conceito escreva uma vez, rode onde quiser (termo em inglês para *write once, run anywhere)*, tem sido fundamental para sua rápida adoção entre desenvolvedores que buscam alcance multiplataforma sem abrir mão de desempenho e qualidade da experiência do.

A comunidade Flutter desempenha papel essencial em sua expansão, contribuindo ativamente com bibliotecas, pacotes e soluções reutilizáveis. O ecossistema em torno da ferramenta cresce de forma constante, estimulado pelo compromisso da Google com atualizações regulares, melhorias de desempenho e expansão de recursos. Atualmente, o Flutter, aliado ao Dart, se consolida como uma das principais tecnologias para a criação de interfaces modernas, responsivas e eficientes, tanto no setor empresarial quanto educacional.

A CONVERGÊNCIA ENTRE FLUTTER E FLUTTERFLOW: DEMOCRATIZANDO O DESENVOLVIMENTO DE APP COM IA GENERATIVA

O desenvolvimento de aplicativos móveis tem avançado rapidamente, impulsionado pela necessidade de ferramentas que tornem o processo mais acessível, eficiente e adaptável a diferentes perfis de usuários. O Flutter, framework de código aberto criado pela Google, destacou-se como um divisor de águas nesse cenário ao possibilitar o desenvolvimento de interfaces nativas a partir de uma única base de código, com alta performance e ciclos de desenvolvimento reduzidos (IBRAHIM, A. 2019). Com o tempo, a evolução natural desse ecossistema levou à criação de plataformas no-code/low-code como o FlutterFlow, que amplia o acesso à criação de aplicativos, mesmo por indivíduos com pouca ou nenhuma experiência em programação. O FlutterFlow, construído sobre a arquitetura do Flutter, fornece uma interface gráfica baseada em componentes de arrastar e soltar, com suporte para lógica condicional, integração com bancos de dados e serviços externos. Essa abordagem visual democratiza o desenvolvimento de aplicativos ao permitir que *designers,* empreendedores e educadores construam soluções interativas e escaláveis sem a necessidade de codificação tradicional. Tal modelo promove não apenas agilidade, mas também inovação, ao colocar o poder de criação tecnológica nas mãos de uma comunidade mais ampla.

Recentemente, a introdução de recursos baseados em Inteligência Artificial (IA) generativa ao FlutterFlow tem sinalizado uma nova era na automação do desenvolvimento. Um exemplo expressivo é a funcionalidade de geração de imagens a partir de descrições textuais. Com o uso de modelos de IA treinados em grandes bases de dados visuais, o desenvolvedor pode, por meio de um simples *prompt* descritivo, por exemplo: "uma xícara de café fumegante ao lado de um notebook", obter imagens geradas sob demanda e personalizadas para seu aplicativo, eliminando a necessidade de buscas manuais em bancos de imagem e promovendo singularidade visual. Outro avanço relevante é a capacidade de gerar documentos automaticamente, como arquivos PDF, a partir de dados inseridos no aplicativo ou por meio de *prompts* em linguagem natural. Em contextos empresariais, por exemplo, um gerente de projetos pode solicitar a geração de um relatório contendo o status de tarefas e a alocação da equipe. A IA processa esses dados e entrega um documento formatado, estruturado e pronto para distribuição, otimizando processos internos e reduzindo o esforço manual.

A incorporação da IA generativa em plataformas como o FlutterFlow representa uma convergência promissora entre eficiência técnica e criatividade. Ao automatizar tarefas repetitivas e complexas, como a criação de recursos visuais e documentos, os desenvolvedores podem concentrar-se na lógica de negócios, na experiência do usuário e na inovação do produto. A sinergia entre *frameworks* robustos como o Flutter e plataformas visuais como o FlutterFlow, potencializadas por modelos de IA generativa, aponta para um futuro de desenvolvimento mais ágil, inclusivo e inteligente, onde a barreira entre a concepção da ideia e sua realização técnica tende a desaparecer.

GERAÇÃO DE IMAGEM E TEXTO VIA API NO FLUTTERFLOW: INTEGRAÇÃO DE IA EM INTERFACES VISUAIS

A geração automática de imagens por meio de inteligência artificial (IA) emergiu como uma das aplicações mais impactantes no desenvolvimento de interfaces modernas. A utilização de descrições textuais (*prompts*) para gerar ilustrações ou elementos gráficos inaugurou novas possibilidades na criação de aplicativos personalizados e responsivos. Nesse contexto, o FlutterFlow, plataforma no-code/low-code construída sobre o *framework* Flutter, possibilita a integração de chamadas de APIs REST para o consumo de modelos de IA generativa, a exemplo dos disponibilizados pelas plataformas DALL·E (OpenAI) e Gemini (Google AI) (OpenAI, 2025). Através da funcionalidade nativa de chamadas HTTP, o FlutterFlow capacita desenvolvedores, mesmo com conhecimento limitado em programação, a configurar requisições a serviços externos. Ao integrar uma API generativa de imagens, o aplicativo submete um *prompt* de entrada em linguagem natural (como "um robô caminhando sob a chuva em uma cidade futurista") e recebe como resposta um *Uniform Resource Locator* (URL) da imagem gerada, a qual pode ser renderizada automaticamente na interface por meio de um *widget Image*.

Entre as opções disponíveis com planos gratuitos, sobressai a API Gemini da Google, acessível através do Google AI Studio e com suporte à integração via *Google Cloud*[2]. A referida API oferece uma cota gratuita mensal para a geração de conteúdo, abrangendo imagens e outros ativos multimodais, e configura-se como uma alternativa robusta à API DALL·E da OpenAI, também extensivamente utilizada em ambientes Flutter e FlutterFlow. Um exemplo funcional de um layout simplificado no FlutterFlow que implementa essa funcionalidade pode ser estruturado com os seguintes *widgets*:

T*extField*: entrada para o *prompt* textual.

B*utton*: aciona a chamada REST.

Image: exibe a imagem recebida.

Text: fornece instruções ao usuário.

Container: organiza os elementos visualmente.

A chamada à API, cujo exemplo é mostrado na Figura 2, está configurada como uma requisição POST, contendo no corpo o *prompt* textual, as dimensões da imagem e o cabeçalho de autorização com a chave da API.

```
Column
├── Text("Descreva a imagem:")
├── TextField(controller: promptInput)
├── Button("Gerar Imagem", onPressed: callImageAPI)
├── if imageURL != null:
│   └── Image.network(imageURL)
```

Figura 2 – Exemplo de resposta de chamada de API do Flutterflow.

A resposta em formato JSON retorna a URL da imagem gerada, armazenada em uma variável (por exemplo, imageURL) e exibida dinamicamente. Essa abordagem torna a construção de interfaces com conteúdo gerado sob demanda mais acessível, propiciando experiências visuais singulares e dinâmicas. A integração com APIs como Gemini e DALL·E também otimiza o tempo de produção, especialmente em prototipagem rápida, educação, *e-commerce* e *marketing* visual. Em síntese, a possibilidade de incorporar a geração de imagens por IA no FlutterFlow, aliada ao suporte de plataformas com planos gratuitos, consolida uma tendência crescente: a convergência entre acessibilidade, automação e criatividade no desenvolvimento de aplicativos visuais inteligentes.

2 https://cloud.google.com/

APLICATIVO EDUCACIONAL: O UNIVERSO EM SUAS MÃOS

O avanço das tecnologias de Inteligência Artificial (IA) aplicada à educação tem aberto novas possibilidades para o desenvolvimento de recursos interativos e personalizados. As visualizações dinâmicas de fenômenos complexos podem ser geradas sob demanda por meio de IA generativa, o que pode facilitar a compreensão de muitos conceitos abstratos. Esta seção descreve um processo estruturado de desenvolvimento de um aplicativo educacional em FlutterFlow, com integração a APIs de IA, como Gemini, para a geração automática de imagens a partir de comandos textuais fornecidos pelo usuário. O fluxo de dados é contínuo e guiado por ações do usuário, promovendo uma experiência interativa e responsiva. A interface da primeira página é composta por três seções principais: o cabeçalho institucional, o campo de entrada de comando e o botão de envio (FlutterFlow, 2025).

Na página 1 do app um Container é utilizado para agrupar o nome da instituição e seu logotipo, representado pelos *widgets Text* e *Image.asset*. Este componente reforça a identidade visual do aplicativo, além de padronizar a aparência das páginas. A imagem do logotipo deve ser adicionada como ativo local, otimizando seu carregamento. Abaixo do cabeçalho, emprega-se um Container com a finalidade de apresentar o título do aplicativo educacional, representado pelo *widget Text*, com a inscrição “APP - O Universo em Suas Mãos”. Abaixo do título, insere-se uma imagem temática obtida em repositórios de conteúdo livre de direitos autorais, como o *Pixabay* ou *Unsplash*, a qual ilustra visualmente um tema da Física, como por exemplo, a Astronomia.

A seção central utiliza um TextField, com TextEditingController para captar o comando textual que será processado pela API. O *prompt* pode ser uma solicitação como: “Sol Realista”. A interface deve ser amigável e orientar o usuário por meio de *placeholders* e rótulos explicativos, como pode ser visto na Figura 3(a). Na parte inferior da interface, um *ElevatedButton* (Figura 3b) realiza a chamada à API e navega para a página seguinte. O texto inserido no *TextField* é armazenado em uma variável global (*App State*), acessível pela próxima página. Esse controle de estado pode ser feito por ferramentas nativas do FlutterFlow, como FFAppState, ou bibliotecas externas como Provider [24][25].

```
Padding(
  padding: const EdgeInsets.all(16.0),
  child: TextField(
    controller: _aiPromptController,
    decoration: InputDecoration(
      labelText: "Digite o comando detalhado",
      hintText: "",
      border: OutlineInputBorder(),
    ),
  ),
)
```

(a)

```
ElevatedButton(
  onPressed: () {
    context.read<FFAppState>().aiPrompt = _aiPromptController.text;
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) => PaginaDois(useAppState: true),
      ),
    );
  },
  child: Text("Gerar Imagem"),
)
```

(b)

Figura 3 – Código geral para exibição: (a) campo de texto; (b) Botão de Ação e Transição de Página.

A segunda página do aplicativo é composta por um cabeçalho idêntico ao da primeira, um painel central com a imagem e um botão de retorno. Mantém-se a mesma estrutura do cabeçalho para garantir coesão visual. Sua função é reforçar a identidade institucional e manter o padrão de navegação. O *widget* central utiliza FutureBuilder para lidar com a chamada assíncrona da API. Dependendo da resposta, a imagem será exibida por *Image.network* (URL) ou *Image.memory* (binário). A API Gemini, por exemplo, retorna imagens por URL com base em *prompts* textuais, sendo ideal para essa aplicação. Na parte inferior, um botão possibilita o retorno à página anterior para digitação de novo comando. Assim, mantém-se o fluxo interativo e o processo é reiniciado.

A arquitetura descrita proporciona um modelo funcional e replicável para o uso de IA na educação científica. O uso de FlutterFlow mostrado nas Figura 4 e 5), combinado com ferramentas como Gemini, viabiliza a construção de interfaces intuitivas, eficientes e escaláveis para o ensino moderno de Física.

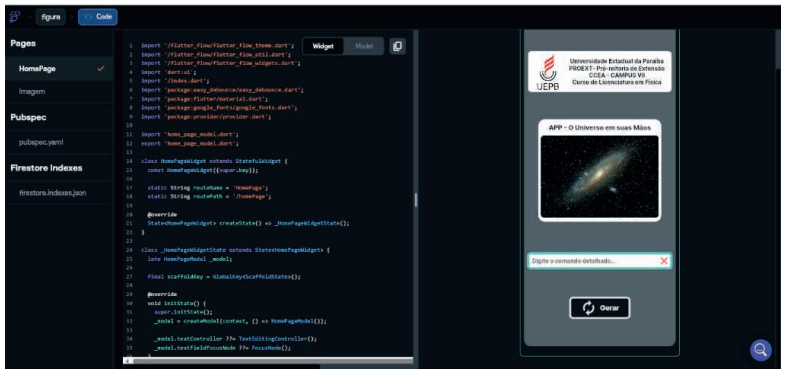

Figura 4 – Prévia de exibição da Página 1 do aplicativo: O Universo em Suas Mãos. À esquerda o código gerador em Dart.

Figura 5 – Resultado Preliminar: Layout do APP O Universo em Suas Mãos, programado para gerar imagens de IA.

CONCLUSÃO

A capacidade dos modelos de Inteligência Artificial (IA) generativa em produzir conteúdo textual e visual com coerência e relevância tem transformado significativamente diversas áreas, incluindo a educação, o design e a pesquisa científica. Entretanto, a qualidade e a precisão dos resultados produzidos por esses sistemas dependem diretamente da forma como os comandos de entrada, denominados *prompts*, são estruturados. Nesse contexto, a engenharia de *prompt* desponta como uma prática fundamental para a maximização do desempenho dos modelos de IA generativa, viabilizando a criação de imagens e textos alinhados a requisitos específicos e contextuais. A engenharia de *prompt* consiste na elaboração estratégica de instruções textuais com o intuito de induzir a IA a gerar respostas que correspondam, com exatidão, às intenções do usuário. No caso da geração de imagens, *prompts* eficazes devem conter descrições detalhadas que abarquem tanto o conteúdo semântico quanto os elementos estéticos desejados. A inclusão de variáveis como o tema central da figura, o estilo visual (realista, técnico, lúdico etc.), a iluminação, a composição espacial e o tipo de fundo contribuem diretamente para a obtenção de imagens mais pertinentes e aplicáveis (Ranesh et al., 2022).

Por exemplo, ao solicitar a geração de uma imagem do "Sol", a inclusão de modificadores no *prompt*, como "Sol realista" ou "Sol foto realista", contribui significativamente para a obtenção de resultados mais condizentes com a proposta pedagógica, conforme mostrado na Figura 6).

Figura 6 – Imagens geradas com os *prompts*: a) Sol; b) Sol Realista; c) Sol Foto realista.

De modo análogo, na criação de uma imagem de "Einstein", a incorporação de descritores como "Einstein em estilo *cartoon* em preto e branco", "Einstein em estilo *cartoon* em preto e branco segurando uma placa" ou ainda "Einstein em estilo *cartoon* em preto e branco segurando uma placa com os dizeres 'AMO PARAÍBA'", pode refinar substancialmente a adequação visual da figura ao contexto didático, tornando-a mais atrativa e funcional para diferentes níveis de ensino em que o professor de Física atue conforme mostra a Figura 7.

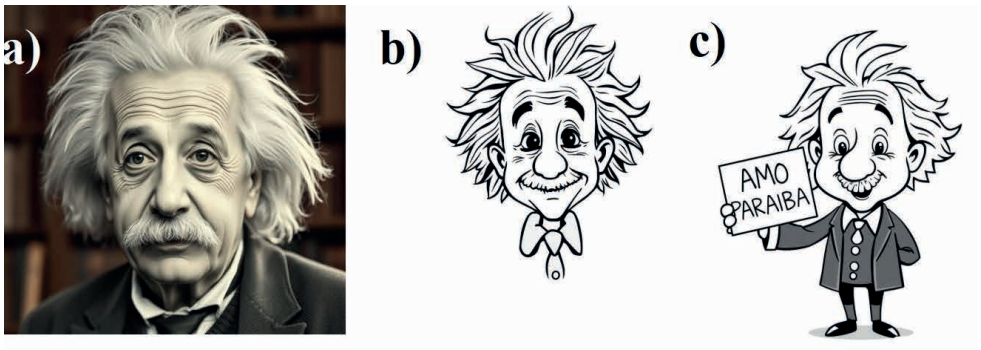

Figura 7– Imagens para colorir, para aplicação nas séries iniciais, geradas com os prompts: a) Einstein; b) Einstein em estilo cartoon em preto e branco; c) Einstein em estilo cartoon em preto e branco segurando uma placa com os dizeres “AMO PARAÍBA”.

O processo iterativo de experimentação com diferentes descritores é uma prática recomendada para refinar os resultados gerados e avaliar sua coerência visual e conceitual (Radford et al. 2017). Na geração textual, a engenharia de *prompt* assume papel igualmente decisivo. *Prompts* bem formulados são capazes de orientar os modelos para produzir textos técnicos, narrativos ou argumentativos com a extensão, a linguagem e o grau de profundidade adequados ao contexto de aplicação. A utilização de abordagens como *few-shot learning*, que incorporam exemplos dentro do próprio comando, tem se mostrado eficaz na indução de respostas com maior consistência e qualidade linguística. A eficiência na construção dos *prompts* também depende do conhecimento sobre as capacidades e limitações do modelo utilizado. Estratégias avançadas de engenharia de *prompt* incluem o uso de palavras-chave específicas, a segmentação lógica das instruções, a separação de blocos informativos com delimitadores e a incorporação de exemplos contextuais. Recentemente, plataformas *no-code* e *low-code*, como o FlutterFlow, têm viabilizado a integração de modelos generativos em aplicações educacionais por meio de interfaces intuitivas. Tais recursos permitem que desenvolvedores e educadores sem formação técnica aprofundada em IA explorem com autonomia o potencial da engenharia de prompt, otimizando a criação de conteúdos visuais e textuais personalizados, que podem ser utilizados para o ensino de Física e outras disciplinas. Em suma, a engenharia de *prompt* representa uma etapa crítica no uso eficiente da IA generativa. Através da construção criteriosa de comandos, é possível orientar os modelos para a produção de conteúdo mais relevantes, contextualizados e de alto valor pedagógico. A próxima etapa evolutiva do aplicativo “**O Universo em Suas Mãos**” concentra-se no aperfeiçoamento da interação entre o usuário e a inteligência artificial generativa. Para esse fim, propõe-se a implementação de um sistema de cadastro de palavras-chave, organizadas tematicamente em áreas como física, desenho técnico e expressão artística. Esse recurso estratégico tem como objetivo refinar a elaboração de *prompts*, fornecendo aos usuários sugestões contextuais e exemplos predefinidos,

previamente gerados por IA. A concretização dessa proposta exigirá o desenvolvimento de um banco de dados robusto e escalável, considerando-se soluções amplamente utilizadas no mercado que disponibilizam planos gratuitos, como o Firebase e o Supabase, de modo a assegurar a gestão eficiente e economicamente viável desse novo conjunto de dados semânticos. A integração dessa funcionalidade ampliará substancialmente a experiência do usuário, potencializando a geração de conteúdos visuais e textuais relevantes para o processo de ensino e aprendizagem em Física.

REFERÊNCIAS

CREVIER, D. **AI: The Tumultuous History of the Search for Artificial Intelligence**. Basic Books, 1993.

Weizenbaum, J. **ELIZA—A Computer Program for the Study of Natural Language Communication Between Man and Machine**. Communications of the ACM, 9(1), 36–45, 1966.

Vaswani, A., et al. (2017) **Attention Is All You Need**. Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, 4-9 December 2017, 6000-6010.

Brown, T. B. et al. **Language Models are Few-Shot Learners**. Advances in Neural Information Processing Systems, 33, 1877–1901, 2020.

Ibrahim, A. **Flutter for Beginners: An Introductory Guide to Building Cross-Platform Mobile Apps**. Packt Publishing, 2019. https://www.packtpub.com/product/flutter-for-beginners/9781788996082. Acesso em: 16 de Jul. 2025.

Joshi, P. (2020). **Flutter in Action. Manning Publications**. https://www.manning.com/books/flutter-in-action. Acesso em: 10 de Jul. 2025.

Soseman, B., & Morgan, K. **Dart Apprentice: Fundamentals**. Big Nerd Ranch Guides, 2019.

FlutterFlow. (n.d.). **FlutterFlow Documentation**. Disponível em: https://flutterflow.io/docs. Acesso em: 10 de Jul. 2025.

OpenAI developer platform [Online]. Available: https://platform.openai.com/docs/overview. Acesso em: 11 de Jul. 2025.

Ramesh, A., Dhariwal, P., Zhang, Y., Gokaslan, C., Jun, S., Narayanan, A., ... & Sutskever, I. (2022). **Hierarchical Text-Conditional Image Generation with CLIP Latents**. arXiv preprint arXiv:2204.06125.

Radford, A., Kim, J. W., Xu, C., Narasimhan, G., Sutskever, I., & Salimans, T. **Learning transferable visual models from natural language supervision**. arXiv preprint arXiv:1703.03230, 2017.