

A PROGRAMAÇÃO ORIENTADA À GAMBIARRA (POG)

Um paradigma pedagógico para as artes e a educação tecnológica

Henrique Vaz

Resumo: Este artigo investiga e propõe a programação orientada à gambiarra (POG) como um paradigma pedagógico insurgente para a educação tecnológica nas artes. Em oposição aos modelos hegemônicos baseados em eficiência, usabilidade e acabamento, a POG eleva a precariedade, o improviso, o erro, a subversão e o inacabado à condição de operadores epistemológicos. O artigo articula a fundamentação teórica do paradigma — dialogando criticamente com os fundamentos da computação — e apresenta sua proposta operacional, sistematizada em um ecossistema de ferramentas e uma cartografia pedagógica aplicada. A POG se afirma, assim, como um gesto de resistência ao tecnofeudalismo algorítmico, ao colonialismo de dados e à plataformação da educação, reconfigurando o letramento digital como uma prática crítica que devolve à educação tecnológica seu caráter emancipador.

Palavras-chave: programação orientada à gambiarra; educação tecnológica; pedagogia crítica; pensamento computacional; artes e tecnologia.

A PROGRAMAÇÃO ORIENTADA À GAMBIARRA (POG)

Um paradigma pedagógico para as artes e a educação tecnológica

Henrique Vaz

Para mim, a questão que se coloca é: a serviço de quem as máquinas e a tecnologia avançada estão? Quero saber a favor de quem, ou contra quem as máquinas estão postas em uso. [...] Para mim os computadores são um negócio extraordinário. O problema é saber a serviço de quem eles entram na escola (FREIRE, Paulo. *A máquina está a serviço de quem?* Revista Bits, 1984).

1. Educação tecnológica: cenários empíricos, normativos e conceituais

A pesquisa TIC Kids Online Brasil 2019 (CETIC.BR, 2019a) atesta que 89% das crianças e adolescentes de 9 a 17 anos são usuários de internet, sendo o acesso majoritariamente móvel (95%). Esse uso intenso evidencia não apenas a conectividade, mas a interação precoce com tecnologias opacas baseadas em inteligência artificial sob regime corporativo (IA extrativista, AlaaS), levantando tensões em relação a Convenção sobre os Direitos da Criança face à exposição a conteúdos sensíveis e mercadológicos. Na esteira desse fenômeno, a TIC Educação 2019 (CETIC.BR, 2019b) aponta a ubiquidade do WhatsApp (85% dos alunos urbanos) e do Instagram (65%) para atividades escolares e sociais, plataformas pertencentes ao conglomerado Meta (DINO; MACAYA, 2020).

Essa realidade exige políticas públicas que orientem sua incorporação crítica. No Brasil, um arcabouço legal e normativo fornece tais diretrizes. A Lei de Diretrizes e Bases da Educação Nacional (Lei

nº 9.394/1996) já estabelecia a inclusão digital e o uso crítico das tecnologias. O Plano Nacional de Educação (Lei nº 13.005/2014), por sua vez, prevê em suas Metas 7 e 12 a ampliação das TICs no ensino e na democratização do acesso à educação superior. Condição indispensável para isso, o Marco Civil da Internet (Lei nº 12.965/2014) assegura o acesso universal. Já a Lei Geral de Proteção de Dados (Lei nº 13.709/2018) impacta diretamente o setor ao regular o tratamento de dados de estudantes, enquanto a Lei Brasileira de Inclusão (Lei nº 13.146/2015) impõe a acessibilidade digital como requisito transversal, não como adaptação posterior. No âmbito normativo, a Resolução CNE/CES nº 1/2016 e a Resolução CNE/CP nº 2/2017 preveem a incorporação de tecnologias digitais nos currículos e na formação docente, respectivamente, sendo a primeira a norma que estabelece as Diretrizes e Normas Nacionais para a Oferta de Educação Superior na Modalidade a Distância. A Base Nacional Comum Curricular (BNCC), por fim, determina a inserção das tecnologias como elemento essencial ao processo pedagógico. Em conjunto, estes dispositivos não apenas legitimam a inovação pedagógica, mas também situam a urgência de uma “educação tecnológica” no centro das políticas educacionais.

Tal arcabouço, em sua densidade normativa, alinha-se a um movimento global de incorporação da computação na educação básica. No entanto, como observado em contextos estadunidenses, britânicos e nórdicos, a expansão legal precede a consolidação de diretrizes pedagógicas substantivas: todos esses marcos, no Brasil e no exterior, legitimam a inovação, delimitam competências e habilidades, mas permanecem relativamente silenciosos quanto às epistemologias que devem orientar sua abordagem.

No plano internacional, observa-se um movimento convergente de incorporação da Ciência da Computação na educação básica ao longo das últimas décadas. Iniciativas como o *K-12 Computer Science Framework* (K-12 COMPUTER SCIENCE FRAMEWORK, 2016; CODE.ORG, 2020; LYE; KOH, 2014; TOURETZKY et al., 2013) nos Estados Unidos, bem como políticas análogas no Reino Unido (GOV.UK, 2022), Austrália (ACARA, 2015) e Finlândia (EDUFI 2022), evidenciam um consenso crescente em torno da centralidade do pensamento computacional e do letramento digital como competências fundamentais para o século XXI.

Nesse contexto, diferentes propostas metodológicas têm sido formuladas. Souza et al. (2022) apresentam o CTProgER, um *framework* baseado em robótica educacional; Jocius et al. (2020) propõem o modelo 3C (*Code, Connect, Create*), voltado à formação docente; e Zhang et al. (2019) desenvolvem o LPK12, que integra pensamento computacional e ensino de ciências por meio de programação lógica. Apesar de suas contribuições, essas abordagens tendem a estruturar o ensino a partir de ferramentas específicas ou da programação como eixo central, frequentemente subordinando o pensamento computacional a finalidades instrumentais ou disciplinares.

Embora mais detalhado, o arcabouço brasileiro, em seu ordenamento legal, também carece de diretrizes pedagógicas substantivas sobre como ensinar computação frente aos “estudos críticos do código” (SOON, 2016; MARINO, 2020). Iniciativas como a Olimpíada Brasileira de Robótica (OBR) exemplificam essa difusão com resultados mensuráveis de engajamento (DA SILVA NÓBREGA 2022). Fundamentada em abordagens construtivistas e construcionistas (NUNES et al.,

2021), a OBR enfatiza protagonismo estudantil e articulação teoria-prática. Contudo, mesmo em iniciativas bem-sucedidas, observa-se a permanência de uma lógica pedagógica orientada à resolução de problemas dentro de estruturas relativamente estabilizadas — lógica que, como se verá, a POG vem tensionar.

Legitimar o uso da tecnologia não basta; é crucial distinguir entre “tecnologia para a educação” e “educação tecnológica”. A primeira refere-se ao conjunto de ferramentas de apoio ao ensino, como *softwares* e jogos educacionais (KRÜGER, 2006; MILETTO et al., 2004; SCHRAMM, 2009). A segunda, por sua vez, toma a tecnologia como objeto de estudo, vinculada ao desenvolvimento do pensamento computacional (BLIKSTEIN, 2008). Se a “tecnologia para a educação” instrumentaliza, a “educação tecnológica” visa formar sujeitos capazes de compreender criticamente e reinventar a técnica, superando o “adestramento digital”. Enquanto a primeira amplia o acesso e dinamiza práticas, apenas a segunda assegura autonomia cognitiva e criativa face à ubiquidade tecnológica, preenchendo, assim, o vão entre o arcabouço legal e uma formação emancipatória.

O panorama levantado revela uma tendência recorrente: a consolidação de *frameworks* que, embora ampliem o acesso e a difusão da computação, permanecem ancorados em modelos pedagógicos orientados à “eficiência”, à “previsibilidade” e à “estabilização” de práticas. É precisamente aí que emerge a necessidade de paradigmas alternativos capazes de tensionar tais pressupostos e reintroduzir a dimensão crítica e inventiva.

Essa distinção entre “tecnologia para a educação” e “educação tecnológica” não é meramen-

te terminológica ou programática. Ela carrega uma aposta epistemológica mais profunda. Se a primeira trata a técnica como ferramenta — algo a ser usado — a segunda a trata como objeto de compreensão crítica e de transformação. Nesse sentido, a “educação tecnológica” alinha-se a uma tradição que insiste na múltipla realizabilidade do “mental” sobre o “material”: a cognição pode ser realizada em diferentes suportes — silício, código, algoritmos —, o que permite não apenas estudar a mente por modelos artificiais, mas ampliar seu campo para além de seu “habitat naturalizado”. Uma educação tecnológica crítica, portanto, não pode se contentar em ensinar o uso de ferramentas digitais; precisa formar sujeitos capazes de compreender a técnica em sua materialidade, em sua lógica algorítmica e em seu potencial de reconfiguração. Do contrário, a incapacidade de transformar o mundo pode estar diretamente relacionada à incapacidade de compreendê-lo — e uma tecnologia que não se compreende criticamente é uma tecnologia que se limita a reproduzir as relações de poder que a constituem.

No contexto brasileiro, a gambiarra é também herança de práticas de sobrevivência técnica. A POG não eleva essa prática para uma estetização distanciada, mas reconhece nelas um saber que a educação formal historicamente desqualificou. Se a precariedade estrutural é violência, a gambiarra é contra-conduta (CERTEAU, 1990) que a POG sistematiza como método pedagógico. O que se segue articula essa aposta em quatro movimentos: primeiro, o diagnóstico do pensamento computacional e da POO como marcos que a POG vem tensionar; segundo, a crítica à “gamificação” e à “gamblificação” como contexto de urgência; terceiro, a fundamentação teórica do paradigma;

quarto, o diálogo imanente com a computação e a operacionalização pedagógica via Estudos Críticos do Código e laboratório documentado.

2. Do pensamento computacional à POO: fundamentos e abertura para a gambiarra

O pensamento computacional (PC) emerge como estratégia cognitiva inspirada na ciência da computação. Wing (2006) o define como a habilidade de formular problemas e soluções processáveis por um agente humano ou máquina, indo além da programação para abranger a estruturação do raciocínio e a decomposição de complexidades. Essa definição, ao incluir processos exclusivamente humanos no escopo do PC, abre o conceito a interpretações amplas — o que, para alguns críticos, dilui sua especificidade algorítmica (FOOHS et al., 2025), mas que, para os fins deste artigo, indica justamente sua plasticidade normativa: o PC pode servir a pedagogias da eficiência ou a pedagogias da subversão, dependendo de quais aspectos de sua definição se enfatizam.

Blikstein (2008) sublinha a necessidade de introduzir o PC criticamente no currículo, sob risco de a formação se reduzir a um consumo passivo de tecnologias. No Brasil, sua incorporação na educação básica e superior busca atender à demanda por letramento digital, mas persiste o desafio de evitar que se converta em mera tecnicidade, esvaziando sua dimensão crítica. Compreendê-lo como dispositivo formativo que articula lógica, criatividade e autonomia cognitiva exige, portanto, atenção não apenas ao seu núcleo algorítmico, mas às pedagogias que o mobilizam.

Nesse contexto, a programação orientada a objetos (POO) consolidou-se como paradigma pedagógico central. Baseada em classes, objetos, he-

rança e polimorfismo, a POO oferece um modelo cognitivo de resolução de problemas via abstração e modularização (BRENNAN, 2011). Seus conceitos foram adotados não apenas como técnica, mas como forma de pensar a decomposição de sistemas, orientando estudantes a mapear o mundo em entidades e interações formalizadas em código. A POO, nesse sentido, cumpre uma função estruturante: ensina a abstrair, a encapsular, a hierarquizar — talvez operações cognitivas fundantes para qualquer intervenção técnica posterior.

Contudo, a POO carrega uma normatividade técnica implícita que merece ser problematizada, não rejeitada. Sua ênfase em hierarquias, encapsulamento e previsibilidade estabiliza o mundo em taxonomias rígidas: o erro tende a ser corrigido, a eficiência valorizada e a criatividade subordinada a objetivos previamente definidos. A construção de sistemas estáveis ocupa o horizonte, enquanto a experimentação com a precariedade, a improvisação e a abertura dos processos criativos permanecem marginalizadas. Isso não constitui falha da POO, mas limite de seu escopo: ela forma para a engenharia, não para a gambiarra; para a estabilidade, não para o improviso; para o produto finalizado, não para o inacabado.

Se o PC e a POO são marcos — o primeiro por introduzir a dimensão algorítmica na educação, o segundo por oferecer um modelo didático de abstração —, ambos operam dentro de um horizonte comum: o da “previsibilidade como valor normativo”. O que permanece em aberto é se uma pedagogia que valorize a imprevisibilidade como operador epistemológico, e não como defeito a ser corrigido, pode emergir não à margem desses marcos, mas a partir de sua própria lógica interna.

2. Do *homo ludens* à luditadura, da gamificação à gambificação...

A “gamificação” — a aplicação de mecânicas de jogo a contextos não lúdicos — consolidou-se como estratégia hegemônica na educação tecnológica e no design de plataformas. No entanto, sua adoção frequentemente acrítica mascara implicações éticas e políticas profundas. Como modelo persuasivo, ela pode moldar comportamentos, reproduzindo lógicas de consumo, controle e instrumentalização, configurando o que Mattar e Czeszak (2017) intuem como “modelos predatórios”. Esse quadro é análogo à “luditadura” (ESCRIBANO, 2013), um regime de dominação e controle social que se assemelha à McDonaldisação (RITZER, 2021), à Californização (RAESSENS, 2006), à Americanização (KOOIJMAN, 2008) e à Disneyficação (HARVEY, 2000). A crítica cultural revela a contradição central: se o jogo é, por essência, a expressão do *homo ludens* e de sua liberdade criativa, sua apropriação instrumental pela “gamificação” o converte em mecanismo de normalização e performatividade (FUCHS, 2014). O que era para ser espaço de invenção torna-se ferramenta de regulação.

Se há uma década a crítica se concentrava na “gamificação” como captura do lúdico pelo produtivo, o cenário contemporâneo impõe um desdobramento mais crucial: a “gambificação” — a conversão de toda interação digital em estrutura de aposta, onde a recompensa certa é substituída pela promessa incerta do prêmio. O que antes era tornar as tarefas mais atraentes (“gamificação”) agora se expande para tornar a própria vida um ativo negociável (“gambificação”). Exemplos proliferam em escala global: plataformas de e-commerce que condicionam descontos a mecanismos

de sorteio; *fintechs* que migram de modelos de poupança para estruturas de staking e apostas; mercados preditivos que se apresentam como “democratização da informação” mas financeirizam diferenças de opinião sobre eventos geopolíticos, eleitorais e até sanitários, transformando a incerteza coletiva em oportunidade de especulação individual. A lógica é invariante: sob o pretexto da “democratização do acesso”, generaliza-se a financeirização do cotidiano — não como inclusão, mas como ampliação da dependência estrutural de renda variável sobre a qual o trabalhador não exerce controle.

Esses fenômenos não são acidentes ou desvios de percurso. Eles são a expressão explícita de um ciclo estrutural mais amplo, descrito por Cory Doctorow (2025) como *enshittification* (algo como “coisificação” ou “degradação sistêmica”). O ciclo é invariante: primeiro a plataforma é boa para o usuário (atrai massa crítica); depois, boa para o cliente-anunciante (extrai valor da atenção); por fim, boa apenas para o acionista (degradando a experiência ao máximo, pois o usuário já não tem para onde ir). Diante desse cenário, a POG posiciona-se como uma resposta radicalmente distinta. Enquanto a “gamificação” busca otimizar e controlar o engajamento por meio de recompensas externas; a “gambificação” radicaliza essa lógica, financeirizando a incerteza; e a *enshittification* consolida o processo como ciclo estrutural inevitável do capitalismo de plataforma — a POG oferece uma contraposição epistemológica. Se a “luditadura” domestica o gesto lúdico, e a “gambificação” o sequestra para o campo da dependência e da exploração financeira, e a *enshittification* o normaliza como destino inevitável, a gambiarra liberta o gesto criativo

como ato de subversão e invenção. A POG não oferece roletas, nem pontos, nem anúncios camuflados: oferece a “precariedade” (o instável, o incerto) como operador de criação, o “erro” como método, o “inacabado” como horizonte — e, sobretudo, recusa a lógica da extração como destino único da técnica.

4. Paradigma: uma abstração para além da programação

Podemos definir um paradigma como uma função de segunda ordem que relaciona singularidades a inteligibilidades possíveis. Formalmente, seja um conjunto de ocorrências singulares $S = \{s_1, s_2, \dots, s_n\}$. Um paradigma P é uma função: $P: S \rightarrow I$, onde I é o espaço das inteligibilidades. Diferente da indução ou da dedução, o paradigma move-se de singularidade a singularidade, por analogia (AGAMBEN, 2010). Assim, cada gambiarra torna-se uma singularidade que não aponta para um universal, mas para uma rede de inteligibilidades criativas. Formalmente, o paradigma estabelece relações analógicas: $P(s_i) \sim P(s_j), \forall s_i, s_j \in S$. Nesse modelo, a gambiarra é um operador G que age sobre um sistema técnico T , desativando seus pressupostos normativos: $G: T \rightarrow T'$, onde $T' \not\equiv T$, mas $T' \Rightarrow R$, sendo R um conjunto de relações e funcionalidades criativas emergentes.

Em outras palavras, a gambiarra não é “falha”, mas reprogramação subversiva: um gesto que transforma a regra em exceção e a exceção em nova regra. A programação orientada à gambiarra (POG) estrutura-se, assim, em precariedade (π) e improviso (ι) como parâmetros heurísticos: $POG(x) = f(x, \pi, \iota)$, onde f é uma função que não busca a otimização global, mas soluções

localmente adequadas. Nesse modelo, a precariedade é uma variável estrutural, e a improvisação, uma heurística de cálculo. Cada gambiarra é um teorema encarnado. Se o século XX consagrou os paradigmas de programação como abstrações metodológicas (FLOYD, 1979), propomos a POG como um paradigma emergente: uma forma de conhecimento que desativa pressupostos rígidos e expõe novas inteligibilidades criativas, transfigurando as condições do problema ao incorporar a precariedade e o improviso como motores de invenção.

É necessário aqui distinguir a precariedade como operador da POG da precariedade como condição de miséria estrutural. A primeira é escolha metodológica — a restrição deliberada que funciona como motor de invenção (como um poeta que se impõe a métrica para gerar sentido). A segunda é imposição material — a privação de condições básicas de existência que viola dignidade humana. A POG não romantiza a segunda: ela emerge, muitas vezes, em contextos onde a primeira é a única forma de resistência à segunda. A gambiarra de quem improvisa uma antena para captar sinal de rádio educativa; a técnica de patch que comunidades mantêm em software livre para evitar dependência de licenças corporativas — essas não são estetizações, mas táticas de sobrevivência crítica que a POG reconhece como epistemologia legítima. O operador π não descreve a miséria, mas recupera a agência intelectual que a miséria estrutural programática tenta negar.

5. Por um manifesto pedagógico da POG

Se a seção anterior definiu a POG como paradigma epistemológico, esta seção o operacionaliza como função pedagógica — especificando os operado-

res π , ι , σ , λ em sua dimensão formativa. A programação orientada à gambiarra (POG) constitui um paradigma pedagógico no qual o erro, a precariedade e o improviso convertem-se de contingências negativas em operadores do pensamento computacional. Se na programação clássica um algoritmo busca soluções corretas, na POG o processo pode ser formalizado como: $f: X \rightarrow Y$, onde $f(x) = y + \epsilon$, sendo ϵ o erro não como ruído, mas como variável estrutural que abre caminho a novas inteligibilidades — f encarna a função que transforma uma entrada $x \in X$ em uma saída $y \in Y$ contrariando $f(x) = y$. Nesse sentido ϵ é método: cada bug não interrompe a aprendizagem, mas desencadeia hipóteses e invenções. A precariedade, por sua vez, modela-se como um operador que restringe o espaço de soluções. Seja Ω o conjunto de todas as soluções possíveis em condições ideais. O operador de precariedade π atua sobre Ω para produzir o subconjunto S de soluções acessíveis sob restrições materiais: $\pi: \Omega \rightarrow P(\Omega)$, tal que $S = \pi(\Omega) \subset \Omega$. Pedagogicamente, S não é uma ausência, mas um espaço de busca que mobiliza a invenção. A precariedade, assim, é um operador criativo que motiva a reconfiguração a partir do disponível.

O improviso (ι) surge como heurística legítima em contextos de incerteza. Formalmente, pode ser descrito como uma função parcial $\iota: X \dashrightarrow Y$, definida apenas para um subconjunto próprio de X . Esta incompletude não é uma falha, mas a condição de possibilidade para a criação situada: onde a função formal $f: X \rightarrow Y$ é indefinida, o improviso ι opera, preenchendo os vazios com conhecimento local, adaptativo e emergente da urgência do contexto. A subversão (σ) é o princípio pedagógico que [des]opera as normas rígidas, ex-

pondo a contingência de toda regra técnica. Ela não se opõe à funcionalidade, mas à sua “normatização hegemônica”. Esse princípio se materializa na própria ação do operador gambiarra $G: T \rightarrow T'$, onde a subversão garante que $T' \neq T$ e que $T' \Rightarrow R$. Educar pela subversão é demonstrar, através da gambiarra, que um sistema pode ser reconfigurado para servir a novos fins e valores, mantendo-se operante sob critérios reinventados... Educar pela subversão é, por exemplo, demonstrar que um mesmo algoritmo de ordenação pode ser reconfigurado para gerar desordem composicional intencional — não para invalidar o algoritmo, mas para expor que sua neutralidade técnica é construção histórica.

Por fim, o inacabado (λ) é reconhecido como força motriz, opondo-se à ideologia do produto finalizado. Seja P o estado de um projeto em um dado momento. O operador de inacabamento λ atua de forma contínua, gerando uma trajetória de estados sempre abertos: $\lambda: P(t) \rightarrow P(t + \Delta t)$, onde $P(t + \Delta t)$ nunca coincide totalmente com $P(t)$, sendo sempre uma expansão, fragmento ou mutação. O projeto é, assim, um “processo infinito”, cuja abertura estrutural é assumida como condição da criação, onde cada execução é uma variação.

Em síntese, a POG propõe uma pedagogia fundada na precariedade, na improvisação e na subversão. Cada gambiarra pode ser entendida como um teorema encarnado na matéria técnica, cuja demonstração se escreve em cabos improvisados, circuitos reconfigurados e linhas de código remendadas. Ensinar a POG significa ensinar a pensar a técnica como território estético e político, em que cada restrição pode se tornar recurso, cada erro pode se tornar método, e cada improviso pode se tornar conhecimento. Podemos agora reformular a definição da POG como função pedagógica mais

completa: $POG(x) = f(x, \pi, \iota, \sigma, \lambda)$, onde a precariedade (π) e o improviso (ι) fundam a prática, a subversão (σ) garante que cada solução reconfigura normas, e o inacabado (λ) assegura que o processo é sempre provisório, aberto a mutações. Essa formulação não apenas descreve a POG como pedagogia da precariedade e improviso, mas como teoria formal da “abertura” e da “reconfiguração”, que pode ser lida tanto em chave filosófica quanto didática...

6. Diálogo com os fundamentos da computação: a POG como crítica imanente

A programação orientada à gambiarra (POG) não se posiciona à margem da ciência da computação, mas engaja-se em um diálogo profundo e crítico com seus fundamentos epistemológicos. Longe de rejeitá-los, a POG os reconhece como um campo onde significados são construídos. Ela opera uma crítica imanente, utilizando as próprias ferramentas e conceitos da computação para expor seus pressupostos culturais e abrir espaço para alternativas. Este diálogo ocorre nos principais eixos:

6.1. Algoritmos: da eficiência operacional à ineficiência expressiva

Na ciência da computação clássica (KNUTH, 1997), um algoritmo é definido por sua concretude e eficiência, sendo a Análise de Complexidade (Big-O) a métrica soberana. A POG propõe uma resignificação: o algoritmo como texto cultural e “performance estética”. Enquanto a computação tradicional busca otimizar para $O(n \log n)$ ou $O(1)$, a POG explora voluntariamente complexidades “ineficientes” — como $O(n^2)$ ou recursões profundas com custo exponencial $O(2^n)$ — para gerar consequências imprevistas. A ineficiência

deixa de ser um defeito e torna-se um parâmetro composicional. Um *loop* intencionalmente mal-condicionado, em vez de travar, pode gerar uma textura de *buffer overflow* sonoro. O conceito de abstração, central no pensamento computacional (WING, 2006), é reapropriado. Não se abstrai para escamotear complexidade e criar interfaces ótimas, mas para criar “camadas de opacidade”. Uma função *gerarSom()*, na POG, pode não encapsular uma lógica determinística, mas sim uma cadeia de processos onde a relação entre *input* e *output* é não linear e deliberadamente obscura.

6.2. Estruturas de dados: reconfigurando hierarquias e linearidades

Estruturas de dados são os alicerces lógicos sobre os quais os algoritmos operam. A POG intervéem nessas estruturas, questionando suas lógicas inerentes de organização. Estruturas arbóreas (árvores binárias, BSTs), que impõem uma relação hierárquica e ordenada, são desestabilizadas. Na POG, uma “árvore sonora” (grafos de síntese frequentemente estruturados como árvores de dependência) em DSP ou DDSP (processamento digital de sinais diferenciável) pode ser percorrida de forma não-ordenada, com saltos entre nós, ou ter sua hierarquia corrompida para produzir aglomerados caóticos de eventos e artefatos: “corromper a hierarquia” transforma a árvore em um grafo direcionado com ciclos, acessos não determinísticos ou quebra de dependências parentais em pipelines de processamento — gesto que, no domínio do áudio, se materializa como *feedback*, *aliasing* por sobrecarga espectral ou modulação caótica. Estruturas lineares como listas encadeadas e filas (FIFO – *First-In-First-Out*) implementam uma temporalidade sequencial e

progressiva. A POG explora o seu avesso: pilhas (LIFO – *Last-In-First-Out*) para criar palíndromos, ou quebras deliberadas da linearidade para introduzir rupturas. Um Array ou List é tratado não como um contêiner ordenado, mas como um campo de possibilidades onde os índices podem ser acessados de forma “aleatória”, reversa ou através de funções de hash imperfeitas.

6.3. Paradigmas de programação: a gambiarra como metalinguagem

A POG recusa a adesão dogmática a um único paradigma, posicionando-se como uma prática “meta-paradigmática”. Em um OO (Orientação a Objetos) tensionado, o encapsulamento é violado para expor variáveis internas; a herança cria híbridos disfuncionais; o polimorfismo opera por pura contingência, não por substituição de Liskov. A programação funcional subvertida introduz deliberadamente efeitos colaterais e estado mutável em funções “puras”, transformando *map* ou *reduce* em processos imprevisíveis. Estruturas de controle corrompidas *if*, *for*, *while* — a base do fluxo imperativo — são subvertidas, condicionais (*if*) podem ser escritas para que a condição seja quase sempre verdadeira ou falsa, criando vieses sistemáticos. *Loops (for)* podem ter seus contadores corrompidos em tempo de execução, gerando estruturas que se auto-desorganizam.

6.4. A máquina abstrata: colapso entre programa e dado

Em um nível mais profundo, a POG engaja-se com o modelo de von Neumann. Na performance de *live coding*, o código (armazenado na memória) pode ser modificado em tempo de execução por quem estiver programando e imediatamente re-

processado pela UCP, colapsando a distinção entre programa e dado. Esta é uma materialização técnica do princípio filosófico da POG: a indissociabilidade entre a ferramenta e o pensamento. Em termos de teoria da computação, tal gesto pode ser aproximado da máquina de Turing — não como modelo abstrato desencarnado, mas como performance material de reescrita. A máquina de Turing teórica opera sobre uma fita infinita onde lê, escreve e desloca símbolos segundo regras de transição fixas. Na programação em tempo real, quem programa faz algo mais radical: as próprias regras de transição são reescritas enquanto a execução ocorre. A fita, na POG, não é dada — é improvisada; não é infinita — é precária, sujeita a vazamentos de memória e corrupções que se tornam eventos composicionais.

Essa recursividade performativa configura o que se poderia chamar de “metamáquina”: o programa é simultaneamente objeto (o que é executado) e operador (o que modifica a execução). O colapso entre programa e dado, materializado na arquitetura von Neumann onde ambos residem na mesma memória, é aqui elevado à potência criativa: quem programa não apenas habita a máquina, a reprograma enquanto habita. Essa é a essência filosófica da POG: “criar é também reinventar as condições da própria criação” — não como metalinguagem distanciada, mas como ação situada no tempo de execução. Enquanto a engenharia de software tende a abstrair o tempo em favor da eficiência e da previsibilidade, a POG o expõe como matéria “estética”. O erro, o atraso ou o *crash* deixam de ser falhas técnicas e passam a ser eventos performativos — o cálculo não elimina a contingência, mas a acolhe como fonte de sentido e invenção.

6.5. Complexidade e computabilidade: os limites do cálculo

A POG estabelece um diálogo crucial com a teoria da computação, particularmente com os “problemas da parada” e da complexidade de Kolmogorov. A POG opera no território dos algoritmos não garantidamente terminantes. Um processo sonoro que pode, em teoria, executar-se indefinidamente ou entrar em um estado de não-resposta, é abraçado como uma fonte de tensão composicional. A incerteza sobre se um processo terminará torna-se parte da textura da performance onde o “problema da parada” verte-se em recurso criativo. A complexidade de Kolmogorov, entendida como a medida do tamanho da menor descrição possível de um objeto, busca sempre a forma mais concisa de representação. A POG, em contrapartida, subverte esse princípio ao explorar justamente o seu reverso: implementações propositalmente verbosas, redundantes ou circularmente complexas para produzir sons ou comportamentos que, em essência, seriam simples. A “ineficiência” não é desperdício: é camada de opacidade sonora que a transparência funcional eliminaria. O que na teoria da informação seria lido como desperdício de recursos, na POG se converte em potência criativa, pois desafia a ideia de que a melhor representação é necessariamente a mais curta e revela que a redundância também pode ser produtora de sentido.

6.6. Lógica e sistemas formais: a falha como inferência

A gambiarra engaja-se de modo singular com os fundamentos da lógica computacional ao transformar princípios rígidos em matéria plástica. Os sistemas de tipos, tradicionalmente concebidos

como garantias formais de consistência, passam a operar na POG como sugestões maleáveis, sujeitas a transgressões deliberadas. *Type casting* indiscriminado, heranças circulares e estruturas de dados logicamente inconsistentes são cultivados pelo seu potencial de invenção, instaurando aquilo que se pode chamar de uma “lógica do malfuncionamento”. Do mesmo modo, as falhas de inferência, que em um compilador ortodoxo resultariam na interrupção do processo, são aqui reapropriadas como método criativo. A mensagem de erro, o *stack trace* e até o estado de memória corrompido deixam de ser ruído indesejado e se tornam insumos para a composição, instaurando uma espécie de raciocínio abduativo inscrito no próprio código. A POG, assim, revela que a inconsistência e a falha não precisam ser corrigidas, mas podem ser vertidas em “estatuto poético” ou como operadores de criação.

Ao realizar este diálogo ampliado com os fundamentos da computação — da eficiência algorítmica à lógica formal —, a POG demonstra que a gambiarra não é sinônimo de ignorância técnica, mas o oposto: é a aplicação de um “conhecimento técnico” com uma intenção crítica. Quem programa nesta perspectiva identifica os valores e hierarquias embutidos nas estruturas mais fundamentais das tecnologias computacionais e, sobretudo, assume reconfigurá-las para epistemológicas plurais.

7. POG e os limites da Programação Criativa: uma leitura a partir dos Estudos Críticos do Código

A noção de “programação criativa” (*creative coding*) emergiu no início dos anos 2000, vinculada a linguagens como Processing e plataformas

como Arduino, com a nobre intenção de deslocar a ênfase da eficiência e do rigor formal para a expressividade e o acesso democrático ao código (FRY; REAS, 2007). Nesse contexto, programar deixou de ser apenas uma atividade técnica para se tornar também uma prática estética. Contudo, ao se consolidar, esse campo passou a carregar suas próprias normatividades. A proliferação de ambientes amigáveis, bibliotecas pré-configuradas e estéticas “polidas” instauraram um regime de criatividade que, se por um lado democratiza, por outro domestica a potência da experimentação. O risco, como alertam Berry e Dieter (2015), é que a “programação criativa” acabe aprisionada em um design de interfaces que repete a lógica da indústria cultural: intuitiva, fluida, mas também previsível e reprodutível.

É precisamente contra essa domesticação que se insurgem os “estudos críticos do código” (*critical code studies* – CCS). Propostos por Mark C. Marino (2020), os CCS aplicam a hermenêutica crítica à interpretação do código computacional, argumentando que as linhas de código não são neutras — elas carregam valores, retóricas e pressupostos culturais que podem (e devem) ser analisados como se analisa um texto literário. O código não é apenas funcional; é também um sistema sóico com sua própria retórica. Essa virada interpretativa, no entanto, não se limita à leitura estática do código-fonte. Como demonstra Winnie Soon (2016) em sua tese *Executing Liveness*, é preciso ir além: o código precisa ser executado, testado e subvertido em suas operações vivas. Soon propõe uma *reflexive coding practice* que combina *close reading* do código, *iterative trials* (BERRY, 2011) e *cold gazing* (ERNST, 2013; PARIKKA, 2011, 2012) para examinar a dimensão viva e performativa do código em rede.

A programação orientada à gambiarra (POG) pode ser lida, nesse quadro, como uma pedagogia que operacionaliza as meditações do CCS e da *reflexive coding practice* no contexto da criação artística e da educação tecnológica. Enquanto a “programação criativa” tradicionalmente se preocupa em superar barreiras técnicas para a expressão, a POG as erige em matéria-prima da criação. Enquanto o *creative coding* busca a fluidez e o acabamento, a POG cultiva a precariedade (π), o improvisado (ι) e o erro (ε) como operadores epistemológicos. Enquanto os CCS nos ensinam a ler o código como texto cultural, a POG nos ensina a reprogramá-lo como gesto subversivo — transformando a regra em exceção e a exceção em nova regra.

Cada gambiarra, nesse paradigma, é mais do que um arranjo provisório: é um teorema encarnado (nos termos da POG) e, simultaneamente, um objeto de estudo crítico (nos termos dos CCS). O código não é só cálculo, mas matéria plástica, dobrável, suscetível à torção criativa. A mensagem de erro, o *stack trace*, o *buffer vazio*, a consulta inexecutável — tudo o que na engenharia de software é ruído ou falha — torna-se, na POG, insumo para a criação e documento para a análise crítica. Assim, a “programação criativa” deixa de ser um espaço domesticado de expressividade e se reconfigura como um território de precariedade insurgente, onde *bug*, remendo, *glitch* e risco se convertem em condições de possibilidade para formas de arte, pensamento e existência que escapam à lógica da eficiência e da reprodutibilidade industrial.

7.1. Da hermenêutica crítica à poiese gambiológica

Se os “estudos críticos do código” nos ensinam a ler o código como texto cultural — carregado de valores, retóricas e pressupostos ideológicos —, resta a questão de como essa leitura se converte em gesto pedagógico. A resposta da POG não está em aplicar os CCS como conteúdo a ser ensinado, mas em operacionalizar sua lógica hermenêutica como método de criação. O código, nessa transposição, deixa de ser objeto de análise estática para tornar-se território de intervenção performática: cada linha é simultaneamente leitura crítica e reescrita subversiva.

Essa transposição foi sistematizada em um protocolo pedagógico aplicado no Departamento de Música do Instituto de Artes e Design da Universidade Federal de Juiz de Fora (UFJF) no primeiro semestre de 2022, envolvendo três disciplinas distintas — Música e Tecnologia I (MUS156), Composição Musical (MUS141/142/143/146/147) e Arranjo Vocal & Instrumental I (MUS290) —, totalizando 28 estudantes de graduação. O dispositivo metodológico foi o “ensaio algorítmico”: não uma “peça” ou “obra” finalizada, mas um experimento documentado que parte da engenharia reversa dos algoritmos de design de instrumentos musicais



Figura 1: Capa e contracapa do álbum Ensaios Algorítmicos I (2022): documentação fonográfica da disciplina Música e Tecnologia I (MUS156-2022/1), resultado da aplicação da POG como método pedagógico. Os ensaios musicais materializam a engenharia reversa de algoritmos de design instrumental (acústico, eletrônico, digital, cultural) via código criativo e gambioluteria. A publicação como avaliação subverte o portfólio acadêmico privado, instaurando ciclo de criação de ponta a ponta (concepção, gravação, design, distribuição) operado em laboratório com as turmas. Arte de capa: Катерина Богатова. Coordenação: Henrique Vaz. UFJF/Estúdio IAD.

Disponível em: <https://estudio-iad-ufjf.bandcamp.com/album/ensaios-algor-tmicos-i>. Acesso em: 15 abr. 2026.

(acústicos, eletrônicos, digitais e culturais) para reconfigurá-los via código criativo e gambioluteria enquanto método de luteria pós-digital.

A inversão pedagógica operada aqui é fulcral. Em disciplinas tradicionalmente analíticas, a lógica hegemônica prescinde da dissecação de repertório autorizado, subordinando o estudante à posição de receptor crítico. A POG reverte esse fluxo: o estudante de bacharelado em instrumento, por exemplo, cuja formação se concentra em interpretação, torna-se autor, compositor e criador não por acumulação de competências técnicas prévias, mas por imersão direta na materia-

lidade do fazer. A gambioluteria — método que colapsa a distinção entre lutier, compositor e executante — funciona como operador de desidentificação: o estudante não aplica conhecimentos sobre instrumentos, reprograma o próprio conceito de instrumento.

O protocolo desenvolveu-se em três movimentos interdependentes. Primeiro, a desmontagem epistemológica: cada estudante selecionava um instrumento (convencional ou imaginário) e mapeava seus algoritmos de design — acústico (resonância, timbre, afinação), eletrônico (circuito, sensoriamento, amplificação), digital (sín-



Figura 2: Capa e contracapa do álbum Ensaios Algorítmicos II (2022): segundo volume da série, documentando a aplicação da POG em cinco disciplinas de Composição Musical (MUS141/142/143/146/147-2022/1), IAD/UFJF. Onde o Volume I operou em disciplina tecnológica (Música e Tecnologia I), o Volume II demonstra a transversalidade do método — a gambiarra como operador de integração curricular, não como conteúdo isolado. A sobreposição de nomes com o Volume I indica continuidade longitudinal de estudantes em múltiplas disciplinas simultâneas. Arte: Катерина Богатова. Coordenação: Henrique Vaz.

Disponível em: <https://estudio-iad-ufjf.bandcamp.com/album/ensaios-algor-tmicos-ii>. Acesso em: 15 abr. 2026.

tese, processamento, protocolo de comunicação) e cultural (função social, simbologia, história de uso). Esse mapeamento não buscava fidelidade taxonômica, mas “opacidade produtiva”: a incompreensão parcial, a suposição abductiva, o erro de leitura eram cultivados como pontos de fuga para a invenção.

Segundo, a reconfiguração gambiológica: a partir desse mapeamento, o estudante construiu um artefato sonoro utilizando materiais de baixo custo, hardware reaproveitado e código criativo. Aqui, os operadores da POG atuam em conjunto: a precariedade (π) impõe restrições materiais que

funcionam como motor de invenção; o improvisado (ι) preenche lacunas de conhecimento técnico com soluções situadas; a subversão (σ) quebra a distinção software/hardware, compositor/instrumentista, teoria/prática; e o inacabado (λ) suspende a ideia de produto finalizado em favor do processo documentado.

Terceiro, a publicação como avaliação: a avaliação das disciplinas não se deu mediante prova ou rubrica convencional, mas pela publicação fonográfica dos ensaios em três volumes. Essa escolha não é meramente administrativa: ela quebra o paradigma da academia como espaço de acu-



Figura 3: Capa e contracapa do álbum Ensaios Algorítmicos III (2022): terceiro volume da série, documentando a aplicação da POG na disciplina Arranjo Vocal & Instrumental I (MUS290-2022/1), IAD/UFJF. Os ensaios consolidam o ciclo formação — da disciplina tecnológica às disciplinas composicionais, chegando à disciplina de arranjo. A POG aqui opera como maturação metodológica: o discente que antes improvisava circuitos integrados, sistemas modulares e desmontava algoritmos agora reconfigura a lógica de agrupamento instrumental sob o prisma da gambioluteria. A série lançada constitui prova de conceito da transversalidade e replicabilidade do método. Arte: Катерина Богатова. Coordenação: Henrique Vaz.

Disponível em: <https://estudio-iad-uffj.bandcamp.com/album/ensaios-algor-tmicos-iii>. Acesso em: 15 abr. 2026.

mulação simbólica privada e instaura um ciclo de criação integrado — composição, gravação, edição, design gráfico (por Катерина Богатова), publicação e distribuição — operado inteiramente em laboratório com as turmas. O “selo” aqui não imita a indústria cultural, mas subverte a lógica do portfólio acadêmico: o ensaio não é arquivado para avaliação do professor, mas lançado como artefato público.

O termo “entalhe” — presente no subtítulo dos três volumes — opera como alegoria metodológica. Em sua acepção tradicional, o entalhe é o gesto do lutier que esculpe a madeira para dar forma ao instrumento. Na POG, o entalhe torna-se intervenção multimodal: é o corte no código, o *glitch* no circuito, a modificação no protocolo de comunicação, a reinvenção da interface humano-máquina. Remete aos campos de *New Interfaces for Musical Expression* (NIME) e *Human Interface Device* (HID), mas não como aplicação de tecnologias pré-fabricadas — como crítica imanente à própria lógica da interface, que a POG desestabiliza em favor de corporeidades computacionais precárias.

Os Ensaio Algorítmicos materializam, assim, a conversão dos “estudos críticos do código” em pedagogia. Se Marino (2020) propõe o *close reading* do código e Soon (2016) propõe a *reflexive coding practice*, a POG acrescenta uma terceira dimensão: fazer-como-pensamento, onde a gambiarra é simultaneamente leitura crítica, escrita subversiva e performance documentada. O código não é apenas interpretado ou executado, mas reconfigurado como instrumento — e o instrumento, por sua vez, é reconfigurado como código. Essa recursividade performativa é o cerne da transposição metodológica: os CCS operacionalizam a pedagogia não como teoria aplicada, mas como prática que teoriza a si mesma no ato.

8. Sistematização e horizontes: da poiese aos paradigmas

Os três volumes dos Ensaio Algorítmicos (Figuras 1-3) não são ilustrações da POG: são sua materialização inaugural. A partir dessa poiese, as cartografias das Figuras 4 e 5 — antes rascunhos prospectivos — tornam-se mapas de um território já percorrido. O que se segue é a derivação teórica dos princípios operacionais testados em laboratório, com suas confirmações, complexificações e limites.

8.1. Teoremas encarnados: das abstrações aos artefatos

A Figura 4 mapeia a transfiguração de blocos clássicos da computação em operadores da POG. A experiência dos 28 estudantes em três disciplinas distintas revela que essa transfiguração não ocorre apenas no código, mas estruturalmente nas práticas curriculares. Quando a Composição Musical e o Arranjo são convertidos em laboratório de engenharia reversa, ocorre um deslocamento metacurricular: o operador subversão (σ) atua não apenas sobre algoritmos, mas sobre a arquitetura institucional do currículo. A “árvore de decisão” da ementa é percorrida de forma não-ordenada, com saltos entre nós que deveriam ser sequenciais, gerando aglomerados de atividades onde a teoria e a prática se confundem — espelhando, no nível pedagógico, a corrupção hierárquica que a Figura 4 atribui às estruturas de dados.

A Figura 5, por sua vez, lista operadores gambiologicos vinculados a ferramentas técnicas específicas. A poiese dos Ensaio complexifica essa vinculação: as ferramentas não são mobilizadas por eficiência, mas por potencial de desvio. O SuperCollider, listado sob “síntese errática e abstração opaca”, foi utilizado não para geração sonora

BLOCOS CLÁSSICOS DA COMPUTAÇÃO	TRANSFORMAÇÃO NOS EIXOS POG	COMENTÁRIO CRÍTICO-CRIATIVO	
MEMÓRIA (RAM, REGISTRADORES, PILHA, HEAP)	A MEMÓRIA DEIXA DE SER MERO REPOSITÓRIO ESTÁVEL E PASSA A SER ESPAÇO PERFORMATICO, INSTÁVEL E SUJEITO A VAZAMENTOS, SOBREPOSIÇÕES E CORRUPÇÕES.	A PRECARIÉDADE DA MEMÓRIA É CULTIVADA COMO OPERADOR EXPRESSIVO: MEMORY LEAKS OU ESTADOS "CORROMPIDOS" SE TORNAM MATÉRIA SONORA OU VISUAL, INSTAURANDO UM ESPAÇO DE IMPROVISO.	1 2 3 4 5 6 7 8
FLUXO DE CONTROLE (CONDICIONAIS, LAÇOS, LOOPS, CONCURRENCIA)	EM VEZ DE ESTRUTURAS RÍGIDAS, O FLUXO É CONSTANTEMENTE DESVIADO, SUSPENSO OU SATURADO POR INTERFERÊNCIAS EXTERNAS E IMPROVISOS EM TEMPO REAL.	O INACABADO SE MANIFESTA: LOOPS INFINITOS, CONDIÇÕES PARADOXAIS E RACE CONDITIONS SÃO ASSUMIDOS COMO DRAMATURGIAS DO TEMPO.	9 10 11 12 13 14 15 16
DADOS E TIPOS (INTEIROS, FLOATS, BOOLEANOS, STRINGS, ARRAYS)	TIPOS NÃO COMO LEIS, MAS COMO SUGESTÕES PLÁSTICAS: TYPE CASTING ARBITRÁRIO E ESTRUTURAS INCONSISTENTES SÃO EXPLORADAS CRIATIVAMENTE.	A LÓGICA ABDUTIVA DA GAMBIARRA: ONDE A COMPUTAÇÃO EXIGE CONSISTÊNCIA, A POG ENCONTRA NA CONTRADIÇÃO UMA FONTE DE INVENÇÃO.	17 18 19 20 21 22 23 24
EXCEÇÕES E ERROS (TRY+CATCH, STACK TRACE)	ERROS NÃO SÃO CORRIGIDOS, MAS PERFORMADOS: MENSAGENS DE EXCEÇÃO, STACK TRACES E CRASHES SÃO MATERIALIZADOS COMO SOM, IMAGEM OU GESTO.	O ERRO É CONVERTIDO EM ESTÉTICA: A FALHA DEIXA DE SER OBSTÁCULO PARA SE TORNAR RECURSO EPISTEMOLÓGICO.	25 26 27 28 29 30 31 32
COMPILADORES E INTERPRETADORES	A DISTINÇÃO ENTRE TRADUÇÃO E EXECUÇÃO SE DISSOLVE: CÓDIGO PODE SER MODIFICADO EM TEMPO REAL, COM RESULTADOS IMPREVISTOS.	A SUBVERSÃO CONTRA O RIGOR DO COMPILADOR: EM LIVE CODING, O CÓDIGO É FLUXO ABERTO E NUNCA SE ESTABILIZA COMO PRODUTO FINAL.	33 34 35 36 37 38 39 40
ARQUITETURA DE HARDWARE (VON NEUMANN, PARALELISMO)	A SEPARAÇÃO CLÁSSICA ENTRE PROGRAMA E DADO É COLAPSADA: AMBOS SE CONFUNDEM NA PERFORMANCE, ESPELHANDO A FILOSOFIA POG.	A PRECARIÉDADE ONTOLÓGICA: O QUE ERA DISTINÇÃO TÉCNICA TORNA-SE METÁFORA DA INDISSOCIABILIDADE ENTRE FERRAMENTA E PENSAMENTO.	41 42 43 44 45 46 47 48
REDES E COMUNICAÇÃO (SOCKETS, PROTOCOLOS, APIS, OSC+MIDI)	REDES INSTÁVEIS, CONEXÕES QUEBRADAS E LATÊNCIAS SÃO INTEGRADAS AO PROCESSO CRIATIVO, EM VEZ DE SUPRIMIDAS.	A FALHA DE REDE COMO GESTO ESTÉTICO: A PRECARIÉDADE DA COMUNICAÇÃO É CONVERTIDA EM IMPROVISO COLETIVO.	49 50 51 52 53 54 55 56
ALGORITMOS CLÁSSICOS (ORDENAÇÃO, BUSCA, OTIMIZAÇÃO)	A EFICIÊNCIA DEIXA DE SER VALOR CENTRAL: ALGORITMOS SÃO "MAL OTIMIZADOS" DE PROPOSITO PARA GERAR RUÍDO, LATÊNCIA OU REDUNDÂNCIA CRIATIVA.	COMPLEXIDADE DE KOLMOGOROV INVERTIDA: A REDUNDÂNCIA E A VERBOSIDADE SE TORNAM CAMINHOS DE INVENÇÃO ESTÉTICA.	57 58 59 60 61 62 63 64
INTERFACES GRÁFICAS E I+O	A INTERAÇÃO NÃO É SUAVIZADA, MAS TORNADA OPACA, GLITCHADA OU IRREGULAR.	A ESTÉTICA DO GLITCH: FALHAS GRÁFICAS E SONORAS COMO INTERRUPÇÕES PRODUTIVAS QUE DESAFIAM A ERGONOMIA HEGEMÔNICA DO DESIGN.	65 66 67 68 69 70 71 72
INTELIGÊNCIA ARTIFICIAL (APRENDIZADO SUPERVISIONADO E NÃO SUPERVISIONADO)	A "ALUCINAÇÃO" DA IA É VALORIZADA COMO RECURSO CRIATIVO E NÃO CORRIGIDA COMO ERRO, ARTEFATOS PERFAZEM-SE EM ESTATUTO POÉTICO.	CONTRA A TIRANIA DO "RACIOCÍNIO", A ALUCINAÇÃO É CELEBRADA COMO DIREITO COMPUTACIONAL DE SONHAR.	73 74 75 76 77 78 79 80

Figura 4: Cartografia da transfiguração de blocos clássicos da computação em operadores da POG. Originalmente prospectiva, esta matriz foi reconfigurada após aplicação em três disciplinas (MUS156, MUS141/142/143/146/147, MUS290) com 28 estudantes, 2022/1, IAD/UFJF. A precariedade (π), o improviso (ι), a subversão (σ) e o inacabado (λ) emergem não apenas como operadores de código, mas como deslocamentos metacurriculares: a “árvore de decisão” da ementa é percorrida de forma não-ordenada, espelhando a corrupção hierárquica atribuída às estruturas de dados.

Fonte: Elaboração própria em impressora matricial.

OPERADOR GAMBIOLOGICO	FERRAMENTAS, AMBIENTES e PLATAFORMAS DE DESENVOLVIMENTO	FUNÇÃO NA PEDAGOGIA POG
SÍNTESE ERRÁTICA E ABSTRAÇÃO OPACA	SUPERCOLLIDER, PURE DATA, CSOUND, CHUCK, FAUST	NUCLEAÇÃO DA POG: A SÍNTESE ALGORÍTMICA É FUNDADA NA PRECARIIDADE E NO ERRO COMO PARÂMETROS COMPOSICIONAIS. CRIA-SE CAMADAS DE OPACIDADE SONORA COMO ALTERNATIVA CRÍTICA À TRANSPARÊNCIA FUNCIONAL, TRANSFORMANDO RADICALMENTE O MATERIAL SONICO.
IMPROVISO COMO FUNÇÃO PARCIAL	TIDALCYCLES, SARDINE, SONIC PI, FOXDOT, STRUDEL, ORCA, OVERTONE	ESTRUTURA O IMPROVISO COMO UMA LÓGICA DE INCOMPLETUDF, ONDE A CODIFICAÇÃO É PERFORMANCE E O CÓDIGO É TRATADO COMO PARTITURA EM FLUXO CONSTANTE, NUNCA COMO TEXTO FINALIZADO.
SUBVERSÃO META-PARADIGMÁTICA	C++, JAVA, LISP, PYTHON, LUA, HASKELL	ATUA COMO METALINGUAGEM PARA TENSIONAR E CORROMPER PARADIGMAS DE PROGRAMAÇÃO, EXPONDO A CONTINGÊNCIA DE TODA REGRA SINTÁTICA ATRAVÉS DE VAZAMENTOS DE ESTADO, EFEITOS COLATERAIS DELIBERADOS E CONTRADIÇÕES PRODUTIVAS.
INTERFACES, CORPOREIDADE E IA	PROCESSING, OPENFRAMEWORKS, OPENCV, OPENGL, VULKAN, WEBGL, GODOT, OPENSOURCE3D, COMFYUI	criação de interfaces não-ergonômicas que convertem bugs gráficos e rastros de movimento em matéria expressiva. Cria interfaces não-ergonômicas e visualidades glitch, convertendo bugs gráficos e rastros de movimento em matéria expressiva. subverte a aceleração gráfica e a modelagem 3D para produzir corporeidades computacionais precárias. modelagem de alucinações via arquiteturas de aprendizagem profunda.
SOB O ENTALHE DA GAMBIOLOTERIA	ARDUINO, ESP32, RASPBERRY PI, DAISY POD, BELLA SYSTEMS, AXOLOTI CORE, TEENSY (PJRC), TINYFPGA e ULX3S	OPERADOR DE PRECARIIDADE MATERIAL POR EXCELENCIA: A CONSTRUÇÃO DE INSTRUMENTOS A PARTIR DO DISPONÍVEL COLAPSA A DISTINÇÃO SOFTWARE/HARDWARE. A LIMITAÇÃO MATERIAL É O MOTOR DA INVENÇÃO E DA RECONFIGURAÇÃO TÉCNICA.
REDE COMO FRONTEIRA INSTÁVEL	OSC, MIDI, WEBSOCKETS, WEBRTC, FLOK, ESTUARY, ROTEADORES COM FIRMWARE CUSTOMIZADO (OPENWRT), EMULADORES DE REDE (NETEM), RÁDIOS DEFINIDOS POR SOFTWARE (SDR), IMPLEMENTAÇÕES DE DIAL-UP (PPP), REDE MESH COM PROTOCOLOS EXPERIMENTAIS	EXPLORA A LATÊNCIA, A PERDA DE PACOTES E A INTERFERÊNCIA COMO ELEMENTOS COMPOSICIONAIS PRIMÁRIOS. TRANSFORMA A REDE DE UM CANAL TRANSPARENTE EM UM MEIO DE INCERTEZA E IMPROVISO COLETIVO, ONDE A FALHA É MATÉRIA-PRIMA.
SONIFICAÇÃO DE SISTEMAS E EXPLORAÇÃO MULTIMODAL	BIBLIOTECAS DE ANÁLISE DE DADOS. ECOSISTEMA PYTHON (LIBROSA, ESSENTIA, MARSYAS), SCMR (MUSIC INFORMATION RETRIEVAL FOR SCI), MISCELLANEOUS.LIB (DE TILL BOVERMANN); BIO-SENSORES (ECG, EEG, GSR), CÂMERAS WEB COM COMPRESSÃO ARTEFATADA, MICROFONES DE CONTATO EM SUPERFÍCIES INUSITADAS, LEITORES DE REDE (WIRESHARK) CAPTURANDO TRÁFEGO LOCAL.	APLICA A GAMBARRA A DADOS DE SISTEMAS COMPLEXOS E ORGÂNICOS. A SONIFICAÇÃO NÃO BUSCA FIDELIDADE, MAS EXPORTE INTELIGIBILIDADES EMERGENTES A PARTIR DA PRECARIIDADE DOS DADOS E DE ALGORITMOS DE TRADUÇÃO NÃO-LINEARES, OUVINDO O INAUVEL DA CULTURA TÉCNICA.

Figura 5: Cartografia do ecossistema POG: ferramentas ressignificadas não por eficiência, mas por capacidade de serem desviadas, recombinadas, corrompidas. Originalmente prospectiva, esta matriz foi reescrita pelos Ensaios Algorítmicos — cada operador (Síntese Errática, Improvise como Função Parcial, Sob o Entalhe da Gambioloteria) testado em laboratório com 28 estudantes, 2022/1, IAD/UFJF. O hardware reaproveitado, o código em tempo real, a rede instável: tudo é matéria-prima de invenção, não recurso a ser otimizado.

Fonte: Elaboração própria em impressora matricial.

otimizada, mas para criação deliberada de *memory leaks* sonoros — camadas de opacidade que transformam a “transparência funcional” em textura. O TidalCycles e linguagens Uzu para codificação ao vivo de padrões algorítmicos, vinculado ao “improvisado como função parcial”, operou como partitura em fluxo constante onde o *commit* do código funcionava como gesto performativo, não como registro de produto. A distinção entre ferramenta e operador dissolve-se na prática: o que importa não é o que a ferramenta faz, mas o que a gambiarra faz com ela.

8.2. O ecossistema como corpo de operadores

A sistematização dos Ensaios Algorítmicos permite derivar princípios operacionais que extrapolam o caso específico. Se a Figura 4 mapeia a transfiguração dos blocos clássicos da computação, a Figura 5 lista o ecossistema que materializa essa transfiguração. A articulação entre ambas não é meramente ilustrativa: cada operador da Figura 5 (síntese errática, improvisado como função parcial, sob o entalhe da gambioluteria) é uma encarnação técnica dos deslocamentos que a Figura 4 atribui à memória, ao fluxo de controle, à arquitetura de von Neumann. Apresentam-se quatro desdobramentos teóricos confirmados pelo exercício laboratorial:

Primeiro: a precariedade como métrica composicional. A Figura 5 lista hardware reaproveitado (Arduino, ESP32, Raspberry Pi) sob o operador “sob o entalhe da gambioluteria”. Nos Ensaios, a limitação material dessas plataformas — memória insuficiente, latência de processamento, ruído de circuito — não foi superada nem aceita passivamente, mas incorporada como parâmetro

estrutural. Um ensaio utilizou a latência de comunicação OSC como elemento rítmico, transformando a “falha de rede” em padrão temporal. Isso confirma a inversão da complexidade de Kolmogorov proposta: a redundância e a verbosidade tornam-se caminhos de invenção.

Segundo: o código como interface corporificada. O operador “interfaces, corporeidade e IA” (Figura 5) foi mobilizado nos Ensaios através de biossensores (ECG, EEG, GSR), microfones de contato em superfícies inusitadas e DDSF. A inversão pedagógica aqui é: o estudante de instrumento, cuja formação corporifica a dicotomia “executante/compositor”, torna-se interface viva de seu próprio instrumento reconfigurado. O código não medeia a expressão; é a expressão materializada — o gesto físico e o gesto algorítmico indistinguíveis. Há aqui uma distinção crucial entre técnicas de aprendizado de máquina (redes neurais, *backpropagation*, modelos generativos) e o que a literatura crítica passou a denominar IA Extrativista ou IA como Serviço (AlaaS). As primeiras constituem um campo legítimo de exploração matemática e criativa — e são, inclusive, passíveis de gambiarra via *fine-tuning* local, *datasets* curados comunitariamente e execução em hardware reaproveitado. A segunda designa um regime econômico e político: plataformas proprietárias de código fechado, operadas por data centers que drenam recursos, treinadas com dados espoliados sem consentimento e empacotadas como assistentes que, na prática, servem para terceirizar a cognição e extrair mais-valia informacional. A POG não é ludista: ela não recusa o cálculo; recusa a expropriação.

Terceiro: a publicação como performance de conclusão. O ciclo de criação de ponta a ponta —

composição, gravação, design gráfico, publicação — operado em laboratório com as turmas, subverte a lógica do portfólio acadêmico tradicional. A avaliação via publicação fonográfica (três volumes lançados em um semestre) não é mera substituição de nota por produto, mas instauração de um regime de visibilidade pública: o ensaio é lançado, não arquivado. Isso opera o princípio do inacabado (λ) na esfera institucional: o “trabalho de conclusão de disciplina” torna-se artefato em circulação, sujeito à contingência da recepção.

Quarto: a subversão como metalinguagem curricular. O operador “subversão meta-paradigmática” (Figura 5) foi mobilizado não apenas no código, mas na arquitetura institucional das disciplinas. Quando o Arranjo e a Composição Musical foram convertidas em laboratório de engenharia reversa, ocorreu um deslocamento metacurricular: a ementa, estruturada como árvore de decisão hierárquica, foi percorrida de forma não-ordenada, com saltos entre nós que deveriam ser sequenciais. O operador σ , nesse contexto, não viola apenas encapsulamentos de código: corrompe a própria lógica de progressão curricular.

Esses desdobramentos, porém, não se apresentam como triunfo irrestrito. A precariedade como métrica composicional exige disposição ao risco: o ensaio que celebra a latência de rede pode, em outro contexto, simplesmente se manifestar de forma “não-produtiva”. A interface corporificada exige acesso a materiais (sensores, microcontroladores) que nem sempre estão disponíveis em instituições subfinanciadas. A publicação como avaliação pode exigir infraestrutura (selo, plataforma, distribuição) que a maioria dos cursos não possui. A POG, nesse sentido, é “paradigma de condição”: floresce onde há alguma margem de

manobra, e não pode ser exportada como receita universal sem atenção às desigualdades materiais que a tornam possível — ou impossível.

8.3. Limites e contaminações

O exercício dos Ensaio Algorítmicos revela também tensões não resolvidas, que a POG não pode ocultar se pretende rigor epistemológico. A publicação como nota final resolve a lógica do produto, mas deixa em aberto como avaliar o processo sem recair em rubricas tradicionais. Como quantificar — ou melhor, como qualificar — o “entalhe” enquanto gesto técnico-emergente? Os Ensaio operaram com acompanhamento individual contínuo, mas essa escala (28 estudantes) é sustentável. A questão permanece: como institucionalizar a avaliação da gambiarra sem domesticá-la?

Sobre escalabilidade e resistência institucional, os Ensaio funcionaram em disciplinas com estudantes voluntariamente imersos na lógica experimental. A transferência para cursos obrigatórios de grande massa enfrentaria resistência estrutural: currículos fechados, turmas numerosas, formação docente alinhada a paradigmas de eficiência. A POG, nesse cenário, corre o risco de ser neutralizada como “metodologia ativa” domesticada, mais uma técnica no repertório do “engajamento estudantil”.

No que tange à formação docente, quem forma os formadores na lógica da gambioluteria? Os Ensaio foram coordenados por pesquisador com formação em música, computação e luteria — hibridismo que não é replicável por mera capacitação docente. A POG exige docentes que operem como artistas-pesquisadores, não como aplicadores de método. Isso levanta a questão política: a disseminação da POG depende de mudança na formação de professores, não apenas de inserção curricular.

8.4. POG como gesto de resistência

Documentos orientadores recentes, como “Educação digital e midiática: como elaborar e implementar o currículo nas escolas” (BRASIL, 2025), sinalizam esforços institucionais para integrar letramento digital, pensamento computacional e educação midiática. Tais iniciativas, ao articularem políticas públicas, fundamentos pedagógicos e competências da BNCC, legitimam a urgência da educação tecnológica crítica. Contudo, ao operarem predominantemente no registro da construção de competências mensuráveis e da cidadania digital como uso consciente, permanecem relativamente silenciosos quanto às epistemologias que podem orientar o ensino de computação nas artes — lacuna que a programação orientada à gambiarra busca preencher ao reposicionar a precariedade, o erro e o inacabado como motores de invenção pedagógica.

Se a plataformização impõe regimes de captura e controle — onde 72% das instituições de ensino superior brasileiras adotam soluções GAFAM (OBSERVATÓRIO EDUCAÇÃO VIGIADA, 2023) —, os Ensaios Algorítmicos demonstram que a resistência não está em recusar a técnica, mas em reprogramar as condições de sua própria reprodução. O ciclo de criação de ponta a ponta operado em laboratório não é escala reduzida de indústria cultural, mas escala insurgente de autonomia formativa: o selo do Estúdio IAD/UFJF não compete com plataformas de *streaming* e nega a lógica da centralização algorítmica ao operar como infraestrutura comunitária.

A pergunta freireana — “a serviço de quem as máquinas estão?” — encontra na POG uma resposta materializada: a serviço de quem as reconfigura como instrumento de criação coletiva, não

como dado técnico a ser consumido. A gambiarra, elevada a paradigma pedagógico, devolve à educação tecnológica seu caráter emancipador por poiese documentada: 28 estudantes, 3 disciplinas, 3 volumes de ensaios, 6 dias de lançamentos, 1 ciclo de criação que da concepção à fonografia permaneceu — precário, improvisado, subversivo, inacabado — inteiramente em laboratório.

A formalização e lógica empregada ao longo deste artigo não opera, contudo, como tentativa de domesticação ou taxonomização da gambiarra. Funciona, antes, como um gesto de tradução estratégica: um léxico provisório sem neutralizar a potência insurgente do improvisado. Assim como a fita crepe que sustenta estruturas sem selá-las definitivamente, a notação aqui proposta (π , ι , σ , λ , ε) é um andaime conceitual removível, cujo valor reside justamente em sua capacidade de ser desmontado, reinterpretado e refeito no chão da prática. Essa tradução não visa domesticar o erro, mas equipá-lo para circular nos circuitos de validação institucional sem perder seu caráter de contra-conduta. A operabilidade da POG, por sua vez, não se restringe ao código ou à produção sonora. Seus operadores são intrinsecamente transversais e se deslocam com igual potência para outras linguagens artísticas, não é um método setorial, mas uma ecologia do fazer que se espalha por qualquer território onde a técnica encontre a contingência.

Por fim, é necessário reconhecer que a própria escrita deste artigo negocia com a lógica que pretende descrever. A decisão de entrelaçar formalização lógica, crítica institucional e registro poético obedece à exigência de um texto que se deixa costurar por atalhos, que aceita a sobreposição de registros e que convida a leituras não lineares.

res. Há aqui uma recusa deliberada da fluidez asséptica: as seções, as cartografias e as referências não buscam uma progressão inevitável, mas funcionam como pontos de acesso múltiplos, como remendos teóricos que se sustentam mutuamente. Escrever sobre a gambiarra sem praticá-la na forma seria cair na mesma armadilha da domesticação que a POG se propõe a desmontar. Assim, este texto se assume como um artefato em estado de obra, aberto a cortes, reapropriações e novas costuras por quem o ler — gambiarrear o próprio método, transformando a precariedade em potência e o inacabado em horizonte de resistência.

9. Nota de conformidade ética

Este artigo apresenta a documentação e reflexão crítica de uma prática pedagógica desenvolvida no âmbito das atividades docentes regulares do Departamento de Música do Instituto de Artes e Design da UFJF. Por tratar-se de relato de experiência de ensino e produção artística colaborativa, sem intervenção em sujeitos vulneráveis, sem coleta de dados sensíveis e sem fins de avaliação comportamental ou clínica, a atividade não se configura como pesquisa envolvendo seres humanos nos termos das Resoluções CNS nº 466/2012 e 510/2016, dispensando submissão a Comitê de Ética em Pesquisa. Todos os estudantes participantes foram informados sobre o caráter público dos artefatos produzidos (Ensaio Algorítmico I, II e III) e consentiram livremente com a autoria, divulgação e utilização acadêmica de suas composições. A manutenção da identificação e autoria dos discentes é intencional, em consonância com a prática de reconhecimento público da produção acadêmica e artística e com o princípio de visibilidade que estrutura a avaliação por publicação na POG.

REFERÊNCIAS

- AUSTRALIAN CURRICULUM, ASSESSMENT AND REPORTING AUTHORITY (ACARA). *The Australian Curriculum*. Sydney: ACARA, 2015. Disponível em: <https://www.australiancurriculum.edu.au>. Acesso em: 14 abr. 2026.
- AGAMBEN, Giorgio. *Signatura rerum: sobre el método*. Tradução de Flávia Costa, Mercedes Ruvituso. Barcelona: Anagrama, 2010.
- BERRY, David M. *The philosophy of software: code and mediation in the digital age*. Basingstoke: Palgrave Macmillan, 2011.
- BERRY, David M.; DIETER, Michael. *Postdigital aesthetics: art, computation and design*. Basingstoke: Palgrave Macmillan, 2015.
- BLIKSTEIN, Paulo. *O pensamento computacional e a reinvenção do computador na educação*. Education & Courses, v. 1, 2008.
- BRASIL. Ministério da Educação. Secretaria de Educação Básica. *Educação digital e midiática: como elaborar e implementar o currículo nas escolas*. Brasília: MEC, 2025.
- BRENNAN, Karen. *Computação criativa: uma introdução ao pensamento computacional baseada no conceito de design*. Massachusetts: Massachusetts Institute of Technology, 2011.
- CERTEAU, Michel de. *A invenção do cotidiano: 1. Artes de fazer*. Tradução de Ephraim Ferreira Alves. Nova edição, estabelecida e apresentada por Luce Giard. Petrópolis, RJ: Vozes, 1994.
- CETIC.BR. *TIC Kids Online Brasil 2019: Pesquisa sobre o uso da Internet por crianças e adolescentes no Brasil*. Comitê Gestor da Internet no Brasil, 2019. Disponível em: https://cetic.br/media/analises/tic_kids_online_brasil_2019_coletiva_imprensa.pdf. Acesso em: 1 out. 2025.
- CETIC.BR. *TIC Educação 2019: Pesquisa sobre o uso das tecnologias de informação e comunicação nas escolas brasileiras*. Comitê Gestor da Internet no Brasil, 2019. Disponível em: https://cetic.br/media/analises/tic_educacao_2019_coletiva_imprensa.pdf. Acesso em: 1 out. 2025.
- CODE.ORG. *Nine policy ideas to make computer science fundamental to K-12 education*. 2020. Disponível em: https://code.org/files/Making_CS_Fundamental.pdf. Acesso em: 14 abr. 2026.
- DA SILVA NÓBREGA, J. *A aplicação da robótica educacional para auxiliar no desenvolvimento das dez competências gerais da base nacional comum curricular*. 2022. Tese (Doutorado em Educação) — Faculdade de Educação, Universidade de Brasília, Brasília.
- DINO, Luísa Adib; MACAYA, Javiera F. M. *Inteligência Artificial: incluindo a perspectiva de crianças e adolescentes no debate*. In: Panorama Setorial da Internet, Ano XII, n. 3, p. 10-16. São Paulo: CGI.br | NIC.br, 2020.
- DOCTOROW, Cory. *Enshittification: why everything suddenly got worse and what to do about it*. First edition. New York: MCD, Farrar, Straus and Giroux, 2025.

- OPETUSHALLITUS (EDUFI). *National core curriculum for basic education*. Helsinki: Finnish National Agency for Education, 2022. Disponível em: <https://www.oph.fi/en/education-and-qualifications/national-core-curriculum-basic-education>. Acesso em: 14 abr. 2026.
- ERNST, Wolfgang. *Media archaeology: method and machine versus history and narrative of media*. Minneapolis: University of Minnesota Press, 2013.
- ESCRIBANO, Flavio. *Gamificación versus Ludictadura. Obra Digital*, Barcelona, n. 5, p. 58-72, 2013
- FLOYD, Robert W. *The paradigms of programming. Communications of the ACM*, New York, v. 22, n. 8, p. 455-460, 1979.
- FUCHS, Mathias. *Gamification as twenty-first-century ideology*. *Journal of Gaming & Virtual Worlds*, v. 6, n. 2, p. 143-157, 2014.
- FOOHS, Marcelo Magalhães; PISSOLATTO NETA, Antinesca Joana; KRÜGER, Júlia Andressa Paes dos Santos; REBOUÇAS, Randerson Oliveira Melville. *Armadilhas do pensamento computacional: análise crítica*. Aracê, São José dos Pinhais, v. 7, n. 1, p. 2002-2026, 2025.
- GOVERNMENT DIGITAL SERVICE (GOV.UK). *National curriculum*. Londres: GOV.UK, 2022. Disponível em: <https://www.gov.uk/government/collections/national-curriculum>. Acesso em: 14 abr. 2026.
- HARVEY, David. *Spaces of hope*. Berkeley: University of California Press, 2000.
- JOCIUS, R. et al. *Code, connect, create: the 3C professional development model to support computational thinking infusion*. In: Proceedings of the 51st ACM Technical Symposium on Computer Science Education, 51., 2020. Anais... New York: ACM, 2020. p. 971-977.
- K-12 COMPUTER SCIENCE FRAMEWORK. *K-12 Computer Science Framework*. 2016. Disponível em: <http://www.k12cs.org>. Acesso em: 14 abr. 2026.
- KOOIJMAN, Jaap. *Fabricating the absolute fake: America in contemporary pop culture*. Amsterdam: Amsterdam University Press, 2008.
- KNUTH, Donald Edwin. *The Art of Computer Programming: Fundamental algorithms*, Volume 1. 3ª ed. Massachusetts: Addison-Wesley, 1997.
- KRÜGER, Susana Ester. *Educação Musical Apoiada pelas Novas Tecnologias de Informação e Comunicação (TIC): Pesquisas, Práticas e Formação de Docentes*. *Revista da ABEM*, Porto Alegre, v. 14, n. 1, p. 75-89, mar. 2006.
- LYE, S. Y.; KOH, J. H. L. *Review on teaching and learning of computational thinking through programming: What is next for K-12? Computers in Human Behavior*, v. 41, p. 51-61, 2014.
- MARINO, Mark C. *Critical code studies*. Cambridge, Massachusetts: The MIT Press, 2020. (Software studies).
- MATTAR, João; CZESZAK, Wanderlucy. *Gamificação como um novo componente da indústria cultural*. *Revista Intersaberes*, [S.L.], v. 12, n. 25, p. 60-67, 7 jun. 2017.

- MILETTO, Evandro Manara et al. *Educação Musical Auxiliada por Computador: Algumas Considerações e Experiências*. *RENOTE: Revista Novas Tecnologias na Educação*, Porto Alegre, v. 2, n. 1, p. 1-11, jul. 2004.
- NUNES, T. F. B.; VIANA, C. C.; VIANA, L. A. F. de C. *Perspectivas da robótica como recurso pedagógico aplicada à educação 4.0: uma análise bibliométrica sobre robótica educacional*. *Research, Society and Development*, v. 10, n. 1, p. 1-14, 2021.
- OBSERVATÓRIO EDUCAÇÃO VIGIADA. *Plataformização do ensino superior público brasileiro*. 2023. Disponível em: <https://educacaovigiada.org.br/pt/sobre.html>. Acesso em: 10 out. 2024.
- PARIKKA, Jussi. *Operative media archaeology: Wolfgang Ernst's materialist media diagrammatics*. *Theory, Culture & Society*, v. 28, n. 5, p. 52-74, 2011.
- PARIKKA, Jussi. *What is media archaeology?* Cambridge: Polity Press, 2012.
- RAESSENS, Joost. *Playful identities, or the ludification of culture*. *Games and Culture*, v. 1, n. 1, p. 52-57, 2006.
- REAS, Casey; FRY, Benjamin. *Processing: A Programming Handbook for Visual Designers and Artists*. MIT Press, 2007.
- RITZER, George. *The McDonaldization of Society: Into the Digital Age*. 10. ed. Thousand Oaks, US: Sage Publications, 2021.
- SCHRAMM, Rodrigo. *Tecnologias aplicadas à Educação Musical*. *RENOTE: Revista Novas Tecnologias na Educação*, Porto Alegre, v. 7, n. 2, p. 1-8, out. 2009.
- SOON, Winnie. *Executing Liveness – An examination of the live dimension of code inter-actions in software (art) practice*. 2016. Tese (Doutorado) – School of Communication and Culture, Aarhus University, Aarhus, 2016.
- SOUZA, I. M.; ANDRADE, W. L.; SAMPAIO, L. M. *A framework for teaching programming in high school through educational robotics*. In 2022 IEEE Frontiers in Education Conference (FIE), pages 1–9. IEEE, 2022.
- TOURETZKY, D. S.; MARGHITU, D.; LUDI, S.; BERNSTEIN, D.; NI, L. *Accelerating K-12 computational thinking using scaffolding, staging, and abstraction*. In: Proceedings of the 44th ACM Technical Symposium on Computer Science Education, 2013. Anais... New York: ACM, 2013. p. 609–614.
- WING, Jeannette. M.. *Computational thinking*. In Communications of the ACM, vol. 49, no. 3, 2006, pp. 33–35.
- ZHANG, Y.; WANG, J.; BOLDUC, F.; MURRAY, W. G. *LP based integration of computing and science education in middle schools*. In: ACM Conference on Global Computing Education, 2019, Chengdu. *Proceedings...* New York: ACM, 2019. p. 44-50.

Henrique Vaz é pesquisador-criador, professor permanente do Programa de Pós-Graduação em Artes, Cultura e Linguagens da UFJF, onde coordena o grupo “Gambioluteria: da programação orientada à gambiarra ao entalhe da luteria pós-digital”. Atua na intersecção entre modelagem computacional e composicional, síntese sonora e visual, e explora notações e sintaxes emergentes no campo das práticas criativas. Contato: henrique.maia.vaz@ufjf.br