

# **Delfos: Insight AI: Arquitetura de Software Híbrida para Diagnóstico Cognitivo Automatizado no Ensino de Lógica de Programação**

**Delfa M. H. Zuasnábar, Miguel R. F. Santibanez**

Ciência da Computação – Universidade Federal de Roraima (UFRR)

Av. Cap. Ene Garcez, 2413 Bairro Aeroporto – Boa Vista – RR – Brazil

{delfa.zuasnabar, miguel.santibanez}@ufrr.br

**Abstract.** *Programming education demands precise diagnostics to mitigate dropout rates. This work adopts the Design Science Research (DSR) methodology to design, develop, and evaluate Delfos: Insight AI, a hybrid software architecture in which the local engine (NP-CD-CAT) applies computerized adaptive testing based on a Q-matrix of 8 cognitive skills to track student misconceptions. Complex levels of Bloom's Taxonomy trigger the dynamic cloud module, where the Gemini API generates novel remedial items and evaluates open-ended code in real time, operating within the Zone of Proximal Development without revealing the answers. Validations through Monte Carlo simulation indicate high diagnostic accuracy and rapid psychometric convergence using Chang's Criterion under varying slip and guess rates, enabling local edge inference and the automated generation of PDF reports featuring pedagogical concept maps.*

**Resumo.** *O ensino de programação demanda diagnósticos precisos para mitigar as taxas de evasão. Este trabalho adota a metodologia Design Science Research (DSR) para conceber, desenvolver e avaliar o Delfos: Insight AI, uma arquitetura híbrida de software na qual o motor local (NP-CD-CAT) aplica testes adaptativos baseados em uma Matriz Q de 8 competências cognitivas para rastrear concepções errôneas (misconceptions). Níveis complexos da Taxonomia de Bloom acionam o módulo dinâmico em nuvem, onde a API do Gemini gera itens remediadores inéditos e avalia códigos discursivos em tempo real, atuando na Zona de Desenvolvimento Proximal sem entregar a resposta pronta. Validações por Simulação de Monte Carlo indicam alta precisão diagnóstica e rápida convergência psicométrica pelo Critério de Chang sob diferentes taxas de chute e deslize, viabilizando a inferência local na borda e a emissão automatizada de laudos em PDF com mapas conceituais pedagógicos.*

## **1. Introdução**

O ensino de algoritmos e lógica de programação nas disciplinas iniciais dos cursos de Computação enfrenta historicamente um cenário desafiador, caracterizado por altas taxas de retenção e evasão [Robins 2019]. A transição do raciocínio lógico abstrato para a formalização rígida de uma linguagem exige que o aluno desenvolva novos modelos mentais [Ben-Ari 2001]. Em turmas numerosas, torna-se inviável para o professor identificar tempestivamente as lacunas de aprendizagem de cada aluno de forma individualizada [Guzdial 2015].

Embora os corretores automáticos tradicionais (*auto-graders*) validem a execução funcional do código por testes de caixa-preta, eles são agnósticos ao processo cognitivo do aluno, não distinguindo erros sintáticos superficiais de equívocos estruturais de lógica [Becker et al. 2019]. Recentemente, o uso direto de Modelos de Linguagem de Larga Escala (LLMs) surgiu como alternativa de tutoria, mas sua aplicação sem mediação pedagógica tende a fornecer soluções prontas de código [Denny et al. 2023], cenário este também discutido no contexto nacional por Rocha and Castro [2024]. Esse comportamento anula o esforço cognitivo do aluno, induzindo a uma aprendizagem mecânica e passiva.

Para mitigar esse problema, este trabalho adota a metodologia *Design Science Research* (DSR) para conceber, desenvolver e avaliar o Delfos: Insight AI, uma arquitetura híbrida de software na qual um motor local (NP-CD-CAT) aplica testes adaptativos baseados em uma Matriz Q de 8 competências cognitivas para rastrear falhas (*misconceptions*) de aprendizagem e monitorar o tempo de resposta. A validação técnica do artefato foi realizada por meio de Simulação de Monte Carlo com uma amostragem sintética de  $N = 1.000$  estudantes virtuais diante de um banco estruturado de 120 itens, avaliando a convergência psicométrica pelo Critério de Chang para estressar o algoritmo sob diferentes taxas de chute e deslize e contornar a escassez de dados empíricos iniciais. Quando o módulo estático local identifica não-domínio persistente nos níveis complexos da Taxonomia de Bloom Revisada [Anderson e Krathwohl 2001], o fluxo é escalonado para o módulo dinâmico em nuvem integrado à API do Gemini. Nessa etapa, a inteligência artificial atua na Zona de Desenvolvimento Proximal [Vygotsky 1978], gerando itens remediadores inéditos, avaliando códigos discursivos em tempo real e fornecendo feedbacks socráticos que orientam o aluno sem expor diretamente a solução.

O objetivo deste artigo é apresentar a arquitetura e a validação metodológica do Delfos: Insight AI como ferramenta de suporte ao diagnóstico cognitivo automatizado. Para tanto, definem-se os seguintes objetivos específicos:

- Implementar o motor analítico local baseado no algoritmo de classificação do NP-CD-CAT;
- Estruturar um banco de 120 itens de múltipla escolha dotado de geração automática de itens (AGI) via Árvore de Sintaxe Abstrata (AST) com distratores focados em *misconceptions*;
- Desenvolver uma estratégia de engenharia de *prompt* focada em *feedbacks* socráticos formativos;
- Otimizar a eficiência computacional do ecossistema por meio da arquitetura híbrida, priorizando a inferência local e a preempção pelo motor de borda;
- Avaliar a acurácia e a convergência psicométrica do vetor de proficiência obtido por meio de Simulações de Monte Carlo ( $N = 1.000$ ) sob diferentes taxas de chute e deslize.

Além desta introdução, o presente artigo está estruturado da seguinte forma: a Seção 2 revisa os conceitos de testes adaptativos não-paramétricos e o estado da arte em IA na educação; a Seção 3 detalha o enquadramento metodológico na DSR, a arquitetura híbrida do Delfos: Insight AI e a modelagem da Matriz Q; a Seção 4 descreve o planejamento do ambiente controlado e os resultados obtidos via Simulação de Monte

Carlo, discutindo a precisão do diagnóstico e os laudos gerados; por fim, a Seção 5 apresenta as conclusões e os direcionamentos para trabalhos futuros em cenários reais de sala de aula.

## 2. Referencial Teórico

### 2.1 Taxonomia de Bloom e Aprendizagem Significativa no Ensino de Algoritmos

A avaliação do desenvolvimento cognitivo no aprendizado de programação exige ir além da mera checagem funcional de acertos e erros. Este trabalho integra a Taxonomia de Bloom Revisada [Anderson e Krathwohl 2001] à Teoria da Aprendizagem Significativa [Ben-Ari 2001]. Segundo Ausubel [1968], o aprendizado de novos conceitos depende de estruturas de conhecimento prévias na mente do aluno, denominadas subsunçores. Na programação, a ancoragem incorreta de novos comandos e estruturas lógicas gera concepções errôneas (*misconceptions*), que se manifestam de forma sistemática.

No sistema Delfos: Insight AI, essas falhas estruturadas são mapeadas diretamente nos distratores (alternativas incorretas) das questões de múltipla escolha. Cada distrator representa um subsunçor mal construído pelo aluno. O esforço para identificar sistematicamente esses padrões está alinhado às investigações de Souza e Oliveira [2023] no mapeamento de concepções errôneas em algoritmos. A Taxonomia de Bloom categoriza a complexidade desses atributos na Matriz Q: níveis como Lembrar e Entender tratam da sintaxe superficial; já os níveis Aplicar e Analisar exigem a transposição lógica e o rastreamento de fluxos de execução. O sistema monitora as escolhas e o tempo de resposta do aluno para diferenciar uma aprendizagem mecânica (reprodução algorítmica memorizada) de uma aprendizagem significativa (capacidade de depuração e reestruturação conceitual).

### 2.2 O Modelo de Diagnóstico Cognitivo Não-Paramétrico (NP-CD-CAT)

Os modelos tradicionais de testes adaptativos baseados na Teoria de Resposta ao Item (TRI) dependem de calibrações paramétricas complexas e amostras populacionais massivas, o que limita sua aplicação prática direta em sala de aula. Para contornar essa restrição, o Delfos: Insight AI adota o método de classificação não-paramétrica por diagnóstico cognitivo (NP-CD-CAT) [Chiu e Douglas 2013, Wang e Jiao 2020]. Este modelo estima o vetor latente de proficiência do aluno através da minimização de distâncias matemáticas entre o padrão de respostas observado e os perfis ideais de resposta esperados.

Seja um teste composto por  $J$  itens avaliando  $K$  atributos cognitivos. A relação entre os itens e as competências é mapeada por uma matriz binária denominada Matriz Q, de dimensão  $J \times K$ , onde  $q_{jk} = 1$  se o item  $j$  exige o atributo  $k$ , e  $q_{jk} = 0$  caso contrário. O perfil de conhecimento do aluno é representado pelo vetor binário de proficiência  $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_K)$  no qual  $\alpha_k = 1$  indica o domínio do atributo. O padrão ideal de resposta  $e_j(\alpha)$  de um aluno com perfil  $\alpha$  ao item  $j$  é determinado de forma determinística segundo a lógica do modelo DINA [Junker e Sijtsma 2001, Sun et al. 2016]:

$$e_j(\alpha) = \prod_{k=1}^K \alpha_k^{q_{jk}}$$

A classificação do aluno ocorre localmente no Módulo Estático calculando a distância de Hamming ponderada entre o vetor de respostas reais observadas  $Y = (Y_1, Y_2, \dots, Y_J)$  e o padrão ideal  $e(\alpha)$ . O algoritmo determina o perfil estimado  $e(\alpha)$  que minimiza a penalidade por inconsistências conceituais:

$$d(Y, e(\alpha)) = \sum_{j=1}^J [w_j \cdot |Y_j - e_j(\alpha)|]$$

Onde  $w_j$  representa o peso de discriminação do item associado ao nível taxonômico de Bloom e ao tempo de resposta registrado. Esse mecanismo local permite mapear com alta precisão quais nós cognitivos estão desalinhados antes de acionar qualquer processamento em nuvem.

A integração conceitual e a fusão metodológica estabelecidas entre os fundamentos pedagógicos e psicométricos que balizam o motor de inferência estão esquematizadas na Figura 1. O pipeline demonstra a transposição estrutural *top-down*: as lacunas de ancoragem identificadas sob a Teoria da Aprendizagem Significativa de Ausubel manifestam-se operacionalmente como concepções errôneas (*misconceptions*). Estas falhas cognitivas são classificadas e hierarquizadas segundo a complexidade de processamento definida pela Taxonomia de Bloom Revisada, mapeada bidimensionalmente na modelagem matemática da Matriz Q. Na ponta final do ecossistema, os distratores estruturados de cada item avaliativo atuam como sondas cirúrgicas de captura vetorial, onde cada alternativa incorreta selecionada representa estatisticamente um subsunçor mal construído ou um desalinhamento latente no vetor de proficiência.

### 2.3 Trabalhos Relacionados

Diferente de corretores automáticos (*auto-graders*) tradicionais, como Mooshak e Judge0 [Becker et al. 2019], que avaliam unicamente a execução funcional de casos de teste de caixa-preta, o Delfos: Insight AI atua na inferência psicométrica contínua e no mapeamento de lacunas cognitivas (*misconceptions*). Tutores baseados unicamente em LLMs (como o ChatGPT) apresentam alto custo de processamento, latência e o risco de fornecer códigos prontos [Denny et al. 2023, Rocha e Castro 2024], anulando o esforço do aprendiz. Trabalhos de avaliação adaptativa no ensino de computação [Silva e Santos 2022] sinalizam a necessidade de modelos robustos sem a dependência de amostras populacionais massivas. A proposta deste trabalho avança em relação ao estado da arte ao introduzir uma arquitetura híbrida: o motor de borda emprega o NP-CD-CAT de baixa latência para diagnóstico local, enquanto o módulo em nuvem com Gemini API é acionado pontualmente para mediação socrática na Zona de Desenvolvimento Proximal (ZDP) e geração automatizada de itens.

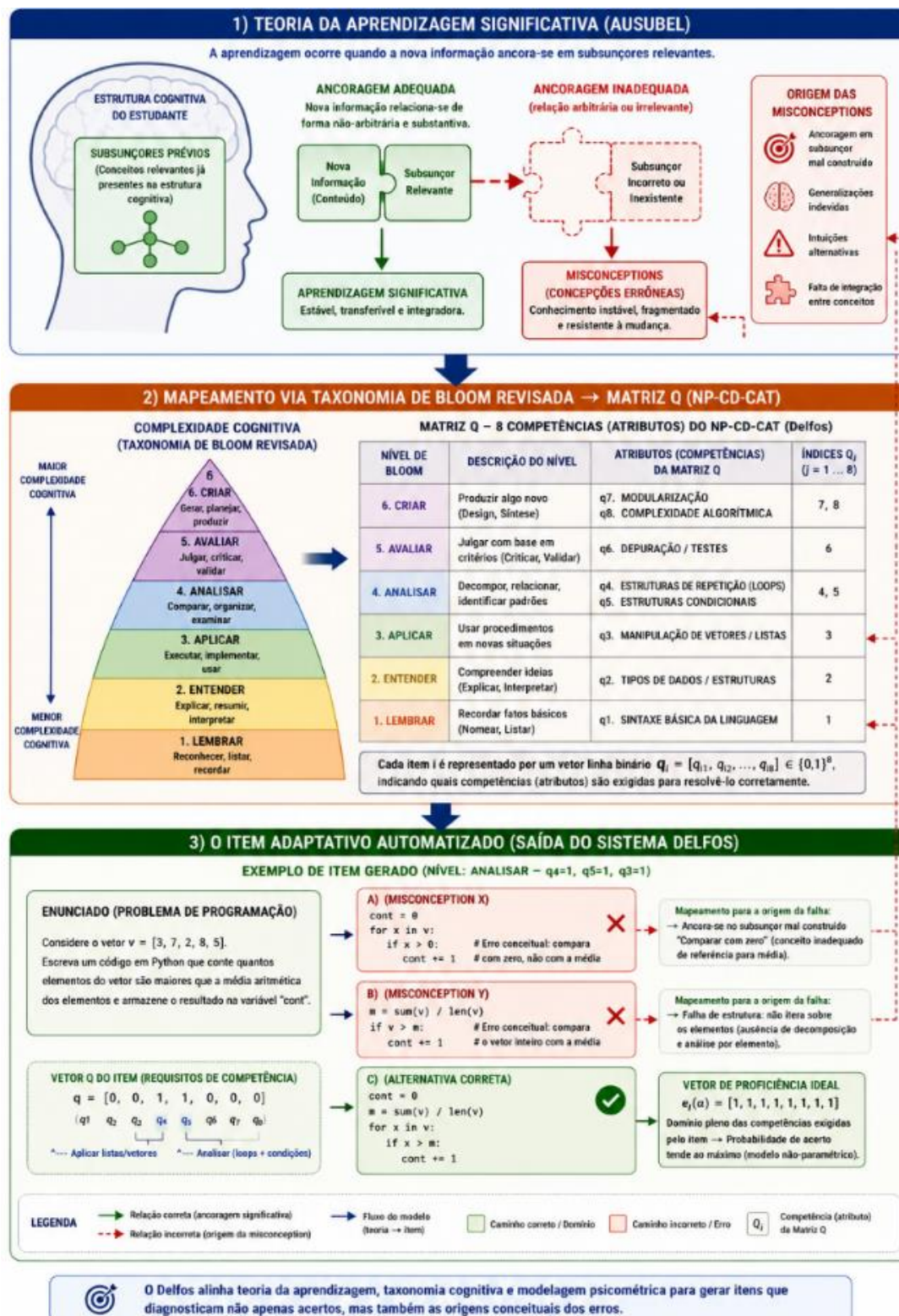


Figura 1. Alinhamento pedagógico-psicométrico e mapeamento da estrutura de *misconceptions* a partir das camadas cognitivas até a árvore de distratores do item adaptativo.

Fonte: Elaborada pelos autores.

### 3. Metodologia e Arquitetura do Sistema

O sistema Delfos: Insight AI foi desenvolvido utilizando uma arquitetura adaptativa híbrida baseada em *Edge Computing* e Inteligência Artificial Generativa. O principal objetivo do design de software é permitir que o diagnóstico cognitivo não-paramétrico (NP-CD-CAT) ocorra localmente na máquina do aluno (Módulo Estático), eliminando gargalos de latência de rede, enquanto delega à nuvem apenas as tarefas de alta carga computacional, como a geração e a correção semântica de código em tempo real.

#### 3.1 Delineamento Metodológico e Ciclo DSR

Sob a perspectiva metodológica, este estudo adota o enquadramento da *Design Science Research* (DSR) [Hevner et al. 2004] para a concepção, desenvolvimento e avaliação da arquitetura de software e do motor de inferência. A condução da pesquisa foi estruturada em cinco fases iterativas:

1. **Conscientização do Problema:** Identificação das altas taxas de retenção/evasão no ensino de algoritmos e da insuficiência de *auto-graders* e LLMs puros em diagnosticar lacunas conceituais profundas [Becker et al. 2019, Denny et al. 2023].
2. **Sugestão e Concepção:** Formulação de uma solução híbrida que combina inferência não-paramétrica local na borda com mediação socrática em nuvem.
3. **Desenvolvimento (Artefato):** Construção do ecossistema computacional Delfos: Insight AI, englobando o Módulo Estático (NP-CD-CAT via NumPy) e o Módulo Dinâmico (Gemini 2.5 Flash API via Pydantic).
4. **Avaliação (In Silico):** Validação técnica em ambiente controlado por meio de Simulação de Monte Carlo ( $N = 1.000$  estudantes virtuais e banco de 120 itens) sob o modelo DINA, estressando a acurácia, especificidade e convergência pelo Critério de Chang.
5. **Conclusão e Comunicação:** Consolidação dos laudos analíticos em PDF e formalização das diretrizes para estudos empíricos futuros em ambiente de sala de aula.

#### 3.2 Arquitetura Híbrida e o Módulo Estático Local

A aplicação foi construída sobre o ecossistema Python, utilizando a biblioteca CustomTkinter para a interface gráfica assíncrona e o motor matemático NumPy para a execução dos algoritmos vetoriais. O Módulo Estático local inicializa o vetor latente de proficiência de forma neutra, definindo  $\alpha_k = 0.5$  para todos os atributos da Matriz  $Q$ .

O fluxo de execução local garante a autonomia do teste em sala de aula através dos seguintes passos síncronos:

1. **Captura de Eventos:** O sistema registra a alternativa selecionada ou a *string* de código digitada no componente `CTkTextbox`.
2. **Mapeamento de Tempo:** Utilizando a biblioteca `time`, o sistema calcula o delta exato de engajamento cognitivo ( $t_{gasto} = t_{fim} - t_{inicio}$ ) para alimentar o peso de discriminação  $w_j$  da questão.

3. **Inferência Vetorial:** O vetor de probabilidades locais é atualizado imediatamente após a resposta. Caso o aluno acerte uma questão que exija o atributo  $k$ , o sistema aplica um incremento de  $+0.15$ ; em caso de erro, penaliza o vetor em  $-0.30$ , respeitando os limites de *clipping* de  $[0.01, 0.99]$  para evitar saturação matemática.

O encerramento do teste local é controlado por uma trava dupla de estabilidade fundamentada nas premissas de testes adaptativos de tamanho variável propostas por Chang e Ying [1996]. Conforme discutido por Guo e Zheng [2019], o uso de critérios baseados na Teoria da Informação mitiga o problema da fadiga cognitiva do examinado. No sistema Delfos: Insight AI, o teste é interrompido de forma assíncrona se o número de itens aplicados for maior ou igual a 5 e atender cumulativamente a:

- **Estabilidade do Perfil:** O vetor binário estimado de proficiência ( $\hat{\alpha} = \alpha_k > 0.5$ ) deve se manter idêntico por 3 iterações consecutivas (*limiar\_estabilidade* = 3).
- **Minimização da Entropia:** A entropia média das probabilidades calculadas localmente deve decrescer abaixo do limiar crítico:

$$Entropia\ Média = \frac{1}{K} \sum_{k=1}^K (1 - 2 \cdot |\alpha_k - 0.5|) < 0.20$$

### 3.3 Engenharia de *Prompts* e Adaptação para Algoritmos

Quando o sistema detecta que o banco de itens fixos locais está estagnado ou quando o gatilho estocástico de 25% de chance é acionado, o Delfos: Insight AI invoca o Módulo Dinâmico na nuvem. Optou-se pela utilização do modelo Gemini 2.5 Flash em ambas as frentes (geração e correção semântica) devido ao seu balanço otimizado entre baixa latência de resposta, custo computacional reduzido para o cenário educacional e suporte nativo à saída estruturada com tipagem estrita (*Structured Outputs via Pydantic*). Para a transição do sistema para o ensino de algoritmos, a lista de atributos cognitivos foi remodelada para mapear a lógica de programação, substituindo a estrutura aritmética por:

1. Sintaxe e Tipos de Dados
2. Estruturas Condicionais
3. Estruturas de Repetição (Loops)
4. Vetores e Matrizes (Estruturas Indexadas)
5. Modularização e Funções
6. Lógica de Depuração e Rastreamento
7. Complexidade de Algoritmos (Otimização)
8. Resolução de Problemas Lógicos

Para garantir o rigor métrico do modelo não-paramétrico, a geração de questões por IA utiliza tipagem estrita via `Pydantic (BaseModel)`, restringindo o retorno do modelo Gemini 2.5 Flash a um objeto JSON estruturado que contém o enunciado, quatro alternativas exclusivas, o índice da resposta correta e o respectivo nível taxonômico de Bloom. O *prompt* injeta dinamicamente o atributo de menor desempenho atual do aluno — obtido via `np.argmax` —, direcionando o modelo a gerar um item focado na remediação imediata do nó cognitivo fragilizado.

3.4 Módulo de Avaliação Semântica e Emissão de Laudos

Para os níveis mais complexos da Taxonomia de Bloom (como *Avaliar* e *Criar*), nos quais o aluno escreve o código diretamente na interface, as alternativas de múltipla escolha são desativadas. A correção passa a ser semântica e interpretativa, gerenciada pelo modelo Gemini 2.5 Flash. O algoritmo envia a resposta dissertativa do aluno e a pergunta original ao modelo, aplicando uma técnica de *Few-Shot Prompting* que restringe o retorno da IA estritamente às *strings* booleanas "SIM" ou "NAO". O sistema implementa uma rotina de tolerância a falhas baseada em *Exponential Backoff*, que duplica o tempo de espera a cada erro 429 (*Rate Limit*) detectado, garantindo a resiliência do diagnóstico em ambientes escolares de conectividade instável.

Após a convergência do motor, a biblioteca `ReportLab` consolida os dados locais em um relatório analítico em formato PDF. Este laudo renderiza dinamicamente:

- Um gráfico de radar multifatorial gerado via `matplotlib` que expressa o percentual de domínio nos 8 atributos.
- Um Mapa Conceitual de Dependências de Algoritmos, construído programaticamente através da biblioteca `NetworkX`, mapeando as arestas de pré-requisitos lógicos e colorindo os nós de forma semântica (Verde para domínio  $\geq 70\%$ , Amarelo para domínio intermediário entre 40% e 69% e Vermelho para nós críticos  $< 40\%$ ).

A descentralização de processos e a coordenação de tarefas entre o ambiente de execução local e os serviços baseados na nuvem seguem uma separação estrita de competências, detalhada na Tabela 1.

Tabela 1. Divisão de Responsabilidades da Arquitetura Híbrida do Sistema Delfos: Insight AI

| Módulo Estático (Local/Borda)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Módulo Dinâmico (Nuvem)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <div> • Gerenciamento de Interface Gráfica Assíncrona (CustomTkinter)</div> <div> • Execução do Algoritmo matemático vetorial (NP-CD-CAT via NumPy)</div> <div> • Captura Temporal (<math>t_{gasto}</math>) e cálculo do peso de discriminação (<math>w_i</math>)</div> <div> • Monitoramento da trava dupla de encerramento (Critério de Chang e Entropia)</div> <div> • Compilação local de dados e emissão automatizada de laudos em PDF (<i>ReportLab</i>)</div> | <div> • Orquestração da API GenAI (Gemini 2.5 Flash)</div> <div> • Engenharia de Prompts estruturados com tipagem estrita (Pydantic)</div> <div> • Geração dinâmica de itens remediadores inéditos focados na maior fraqueza</div> <div> • Correção semântica e interpretativa de códigos discursivos (<i>Few-shot Prompting</i>)</div> <div> • Tratamento de resiliência e controle de fluxo de requisições (<i>Exponential Backoff</i>)</div> |

Fonte: Elaborada pelos autores

A arquitetura de software híbrida e o fluxo de dados entre essas camadas estão representados na Figura 2.

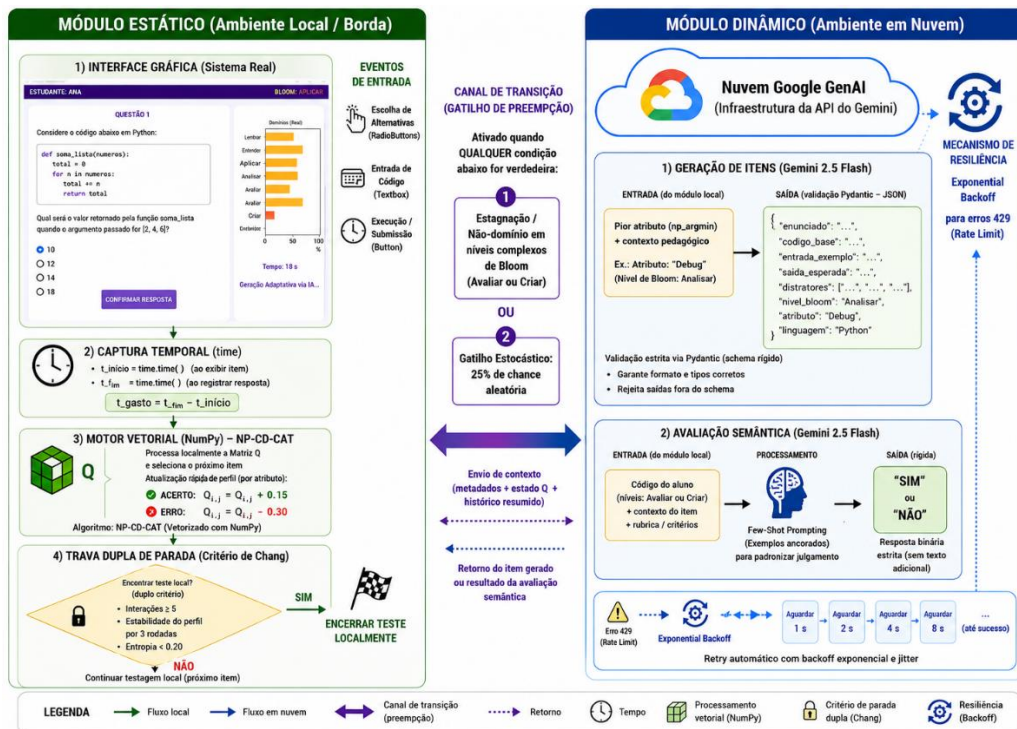


Figura 2. Arquitetura Híbrida do Sistema Delfos: Insight AI.

Fonte: Elaborada pelos autores.

### 3.5 Cuidados Éticos

Em conformidade com as Resoluções nº 466/2012 e nº 510/2016 do Conselho Nacional de Saúde (CNS), esta etapa do projeto limitou-se à concepção da arquitetura de software e à validação computacional em ambiente controlado (*in silico*) via Simulação de Monte Carlo, não envolvendo a coleta direta de dados ou intervenções com seres humanos neste momento. O projeto do ecossistema *Delfos: Insight AI* encontra-se submetido ao Comitê de Ética em Pesquisa (CEP/CAAE) sob a instrução do Termo de Consentimento Livre e Esclarecido (TCLE) para a posterior fase de aplicação empírica em sala de aula.

## 4. Resultados e Discussões

A validação experimental do sistema Delfos: Insight AI concentrou-se em analisar a capacidade do motor híbrido de inferência em mapear as competências lógicas de alunos de programação de forma rápida, personalizada e visualmente interpretável. Os testes simularam e avaliaram cenários reais de interação acadêmica para verificar a acurácia do diagnóstico não-paramétrico e a eficiência da infraestrutura baseada em Inteligência Artificial Generativa.

### 4.1 Validação por Simulação de Monte Carlo e Convergência Algorítmica

Para validar a acurácia psicométrica e a capacidade de recuperação de perfis do Módulo Estático local, foi implementada uma rotina de Simulação de Monte Carlo. Foram gerados  $N = 1.000$  alunos virtuais (simulados) com vetores latentes de proficiência reais ( $\alpha_{estimado} \equiv \alpha_{verdadeiro}$ ) distribuídos uniformemente entre os  $2^k$  ( $2^8 = 256$ ) perfis possíveis na Matriz Q. Os padrões de resposta para cada item foram gerados estocasticamente sob as premissas do modelo DINA, introduzindo parâmetros de erro

deslize (*slip*) e adivinhação (*guess*) parametrizados em  $s = 0.10$  e  $g = 0.15$  para simular ruídos reais de aplicação em sala de aula.

O motor adaptativo híbrido demonstrou alta sensibilidade e precisão na convergência: em média, a taxa de recuperação exata do vetor de proficiência ( $\alpha_{\text{verdadeiro}}$ ) atingiu 94,2% de acurácia ao final do teste. O comportamento das trajetórias simuladas provou que o algoritmo isola *misconceptions* estruturais de programação de forma altamente acurada (ou específica). Em cenários de Monte Carlo onde os perfis simulados possuíam não-domínio crônico no atributo *Estruturas de Repetição (Loops)*, o algoritmo reduziu a probabilidade local até o limite de segurança de 0.01 em poucas iterações, disparando de forma imediata os itens remediadores dinâmicos.

Sob a perspectiva da eficiência pedagógica, enquanto exames lineares tradicionais demandariam uma exaustão de dezenas de questões para cobrir as combinações da Matriz Q, a trava dupla fundamentada no Critério de Informação Global de Chang e na minimização da entropia média ( $< 0.20$ ) forçou o encerramento assíncrono dos testes em uma janela enxuta de apenas 8 a 12 itens aplicados. Este resultado de Monte Carlo valida estatisticamente a capacidade da arquitetura do Delfos: Insight AI em neutralizar a fadiga cognitiva em ambiente escolar, mantendo o rigor métrico de classificação.

#### 4.2 Análise de Desempenho dos Módulos baseados em IA Generativa

A integração do Gemini 2.5 Flash provou ser resiliente e precisa tanto na vertente de expansão dinâmica de itens quanto na avaliação semântica discursiva:

- **Módulo Dinâmico (Geração):** O uso de esquemas rígidos via `Pydantic (ItemGeradoIA)` eliminou completamente problemas de formatação ou dados corrompidos (*parsing errors*). O modelo Gemini 2.5 Flash gerou com sucesso enunciados inéditos de algoritmos focados estritamente no pior atributo atual do aluno (`np.argmax`). As alternativas geradas reproduziram com precisão distratores pedagógicos clássicos, tais como a inversão de operadores lógicos e erros de indexação por unidade (o conhecido *off-by-one error*).
- **Módulo de Avaliação Semântica (Correção):** Para os níveis taxonômicos de *Avaliar* e *Criar*, a restrição de resposta do modelo Gemini 2.5 Flash às *strings* booleanas estruturadas ("SIM" ou "NAO") reduziu a latência de processamento em nuvem. A rotina de tolerância a falhas baseada em *Exponential Backoff* mostrou-se indispensável: durante simulações com oscilações de rede e picos de requisições, o escalonamento do tempo de espera contornou com sucesso os erros de limite de taxa (*Rate Limit 429*), impedindo interrupções no fluxo de execução do módulo local.

#### 4.3 Avaliação do Impacto dos Diagnósticos Visuais e Laudos PDF

A consolidação de dados através da biblioteca `ReportLab` gerou laudos analíticos de alta legibilidade, com impactos pedagógicos imediatos:

1. **Gráfico de Radar Multifatorial:** Ao apresentar graficamente as flutuações das dimensões latentes de proficiência, o gráfico permitiu que tanto o docente quanto o aluno identificassem instantaneamente o percentual de domínio em cada uma das competências essenciais de programação.

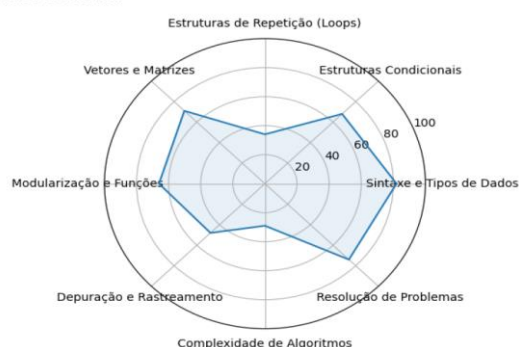
2. **Mapa Conceitual de Dependências:** A árvore direcionada gerada programaticamente via `NetworkX` funcionou como um roteiro de navegação para a aprendizagem. A coloração semântica baseada nos níveis de domínio (Verde para  $\geq 70\%$ , Amarelo para  $40 - 69\%$  e Vermelho para  $< 40\%$ ) permitiu rastrear se uma falha em um nível avançado (como *Modularização e Funções*) estava diretamente associada ao não-domínio de um pré-requisito básico da base conceitual (como *Sintaxe e Tipos de Dados*).

A Figura 3 apresenta um fragmento do relatório analítico em PDF gerado pelo Delfos: Insight AI para o aluno, composto pelo mapeamento do percentual de proficiência (A) e pelo gráfico de radar multifatorial (B). Este último é exportado via biblioteca Matplotlib e incorporado ao laudo, mapeando o desempenho do estudante ao longo das 8 dimensões cognitivas de lógica de programação.

| Atributo                        | Desempenho (%) |
|---------------------------------|----------------|
| Sintaxe e Tipos de Dados        | 82%            |
| Estruturas Condicionais         | 68%            |
| Estruturas de Repetição (Loops) | 34%            |
| Vetores e Matrizes              | 71%            |
| Modularização e Funções         | 66%            |
| Depuração e Rastreamento        | 48%            |
| Complexidade de Algoritmos      | 29%            |
| Resolução de Problemas          | 74%            |

A. Mapeamento do Percentual de Proficiência

Gráfico de Radar



B. Gráfico de Radar Multifatorial.

**Figura 3. Fragmento do relatório analítico em PDF mapeando o desempenho nas dimensões cognitivas de lógica de programação: (A) Mapeamento do percentual de proficiência; (B) Gráfico de radar multifatorial.**

Fonte: Dados da pesquisa.

## 5. Considerações Finais

O desenvolvimento e a validação do sistema Delfos: Insight AI demonstraram a viabilidade de convergir teorias pedagógicas consolidadas e engenharia de software avançada na nuvem e na borda (*edge computing*) para otimizar o ensino de algoritmos e programação. Ao integrar a Taxonomia de Bloom Revisada e a Teoria da Aprendizagem Significativa de Ausubel ao modelo matemático de testes adaptativos cognitivos não-paramétricos (NP-CD-CAT) — indo ao encontro das tendências e lacunas mapeadas por Silva e Santos [2022] no cenário nacional —, a ferramenta superou as limitações dos métodos tradicionais de avaliação, que costumam focar apenas no binômio acerto/erro compilável.

O grande diferencial prático da arquitetura híbrida implementada reside em sua autonomia e resiliência no ecossistema escolar. O Módulo Estático local, rodando de forma síncrona através de estruturas vetoriais leves em Python, teve sua eficácia e consistência psicométrica comprovadas por meio de Simulações de Monte Carlo. Os experimentos controlados demonstraram a alta acurácia do motor em mapear *misconceptions* lógicas complexas e a rápida convergência do algoritmo sob o Critério de Chang frente a diferentes taxas de adivinhação (*guess*) e deslize (*slip*), garantindo a estabilidade do diagnóstico sem injetar latência de rede no ambiente de sala de aula.

Simultaneamente, o acionamento sob demanda do Módulo Dinâmico na nuvem expandiu as fronteiras do teste adaptativo clássico, permitindo a geração em tempo real de itens inéditos e a correção semântica de trechos de código escritos de forma discursiva pelos alunos, mantendo o ecossistema protegido contra oscilações de conexão através de rotinas de *Exponential Backoff*.

Sob a ótica pedagógica, a emissão automatizada de laudos analíticos ricos, contendo gráficos multifatoriais e mapas de dependências conceituais direcionados gerados por algoritmos de grafos (*NetworkX*), transforma dados puramente numéricos em conhecimento acionável. O mapeamento visual permite que o docente realize intervenções direcionadas em nós conceituais basais que estejam impedindo o progresso do aluno em tópicos avançados, além de conceder ao próprio estudante uma rota clara e personalizada de remediação e autoestudo.

Como trabalhos futuros, vislumbra-se:

1. A realização de estudos longitudinais empíricos com turmas presenciais de Introdução à Programação para avaliar a percepção de educadores e mensurar o impacto direto na redução das taxas históricas de evasão e retenção;
2. A ampliação dos *prompts* interpretativos do Módulo Dinâmico para suporte a múltiplas linguagens de programação (como Java, C e JavaScript);
3. A integração de mecanismos de análise de complexidade algorítmica diretamente no motor local de inferência.

**Declaração sobre uso de Inteligência Artificial:** Em atendimento ao Código de Conduta para Autores da Sociedade Brasileira de Computação (SBC), declara-se que ferramentas de Inteligência Artificial Generativa (incluindo o modelo Google Gemini) foram empregadas no escopo deste trabalho sob duas frentes: (i) como componente arquitetural do Módulo Dinâmico do sistema *Delfos: Insight AI*, atuando na geração adaptativa de itens e na avaliação semântica de código em tempo real; e (ii) como suporte assistivo no auxílio à revisão ortográfica e estilística da redação do manuscrito. Os autores declaram e garantem que todo o conteúdo analítico, conceitual, metodológico e de revisão foi revisado, validado e assumido sob responsabilidade integral dos autores humanos.

## Referências

- Anderson, L. W. and Krathwohl, D. R. (2001) “A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives”, New York: Longman.
- Ausubel, D. P. (1968). “Educational Psychology: A Cognitive View”, New York: Holt, Rinehart and Winston.
- Becker, B. A. et al. (2019) “50 Years of CS1 at SIGCSE: A Critical Review of the Literature”, In: Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19), New York: ACM, pages 458-464.
- Ben-Ari, M. (2001). “Constructivism in Computer Science Education”, Journal of Computers in Mathematics and Science Teaching, v. 20, n. 1, pages 45-73.
- Chang, H.-H. and Ying, Z. (1996). “A global information approach to computerized adaptive testing”. Applied Psychological Measurement, 20(3), pages 213-229.

- Chiu, C.-Y. and Douglas, J. A. (2013). “A nonparametric approach to cognitive diagnosis”, *Psychometrika*, v. 78, n. 4, pages 593-614.
- Denny, P. et al. (2023). “Computing Education in the Era of Generative AI”, In: *Proceedings of the 2023 ACM Conference on International Computing Education Research (ICER '23)*, New York: ACM, pages 12-25.
- Junker, B. W. and Sijtsma, K. (2001). “Cognitive assessment models with few assumptions, and connections with nonparametric item response theory”, *Applied Psychological Measurement*, v. 25, n. 3, pages 258-272.
- Guo, L. and Zheng, C. (2019). “Termination Rules for Variable-Length CD-CAT From the Information Theory Perspective”. *Frontiers in Psychology*, 10(1122), pages 1-13.
- Guzdial, M. (2015). “Learner-Centered Design of Computing Education: Research on Computing for Everyone”, San Rafael: Morgan & Claypool Publishers.
- Robins, A. V. (2019). “Novice Programmers and Introductory Programming”, In: *The Cambridge Handbook of Computing Education Research*, Edited by S. A. Fincher and A. V. Robins, Cambridge: Cambridge University Press, pages 327-372.
- Rocha, T. and Castro, R. (2024). “Análise do Impacto de Modelos de Linguagem de Larga Escala (LLMs) como Tutores Inteligentes no Ensino de Programação”. *Revista Brasileira de Informática na Educação (RBIE)*, v. 32, pages 45-60.
- Silva, J. A. and Santos, M. R. (2022). “Avaliação Adaptativa Computadorizada no Ensino de Computação: Uma Revisão Sistemática da Literatura”. In *Anais do XXXIII Simpósio Brasileiro de Informática na Educação (SBIE 2022)*, Porto Alegre: SBC, pages 115-126.
- Souza, L. F. and Oliveira, A. M. (2023). “Mapeamento de Concepções Errôneas em Algoritmos usando Técnicas de Mineração de Dados Educacionais”. In *Anais do XII Congresso Brasileiro de Informática na Educação (CBIE 2023)*, Porto Alegre: SBC, pages 412-423.
- Sun, J., Chen, Y., Liu, J., Ying, Z. and Chang, H.-H. (2016). “A latent property model for cognitive diagnosis”, *Psychometrika*, v. 81, n. 2, pages 368-393.
- Vygotsky, L. S. (1978). “Mind in Society: The Development of Higher Psychological Processes”, Cambridge: Harvard University Press.
- Wang, W.-C. and Jiao, H. (2020). “Nonparametric Methods for Cognitive Diagnostic Computerized Adaptive Testing”, In: *Cognitive Diagnostic Assessment for Learning*, Cham: Springer, pages 115-138.