Entramado
ISSN: 1900-3803, e-ISSN: 2539-0279
Volumen 22, número 2, julio-diciembre 2026

Editorial Universidad Libre

Design of a Portable Retrieval-Augmented Generation Architecture for Offline Intelligent Tutoring on Resource-Constrained Hardware

https://doi.org/10.18041/1900-3803/entramado.2.13645

Yeison Javier Muñoz-Gómez
Institución Universitaria Colegio Mayor del Cauca, Popayán, Colombia

Fabian Armando Hoyos-Cerón
Institución Universitaria Colegio Mayor del Cauca, Popayán, Colombia

Juan José Caiza-Narváez
Institución Universitaria Colegio Mayor del Cauca, Popayán, Colombia

Abstract

Dependence on cloud infrastructure limits the adoption of intelligent tutoring systems in contexts with restricted connectivity, affecting approximately 2.6 billion people without internet access. **Objective**: To design a portable intelligent tutor that operates entirely offline from a USB device on mid-range hardware without a graphics processing unit (GPU). **Methodology**: The design was developed using Design Science Research (DSR), structured into three iterative cycles addressing component selection, architectural integration, and quality evaluator tuning. **Results**: A reference architecture is proposed that integrates hybrid retrieval (dense vector search, BM25 lexical retrieval, and reciprocal rank fusion), local inference using quantized language models in GGUF format, a pedagogical prompt builder with four teaching modes, an automated quality evaluator, and AES-256-GCM encryption at rest. Technical and preliminary validation was conducted by running the system from the same USB drive on three computers with heterogeneous hardware configurations. The system operated successfully on a computer with a dedicated GPU and on a mid-range laptop without a GPU, whereas a third computer lacking AVX instructions experienced an execution failure, thereby establishing the minimum compatible hardware requirements. **Conclusions**: The results suggest the technical feasibility of reducing barriers to access to artificial intelligence-assisted tutoring in contexts with limited technological resources and connectivity; its pedagogical effectiveness remains to be evaluated with end users.

Keywords

Artificial Intelligence in Education, Intelligent Tutoring Systems, Retrieval-Augmented Generation, Large Language Models, Educational Technology, Offline Learning Environments, Local Language Models

Registration

Research article
Received: 06/04/2026
Accepted: 29/05/2026
Published: 20/06/2026

Diseño de una arquitectura portátil de generación aumentada por recuperación para tutoría inteligente offline en hardware con recursos limitados

Resumen

La dependencia de infraestructura en la nube limita la adopción de sistemas de tutoría inteligente en contextos con conectividad restringida, que afecta a aproximadamente 2.600 millones de personas sin acceso a internet. **Objetivo**: Diseñar un tutor inteligente portátil que opere completamente offline desde un dispositivo USB, sobre hardware de gama media sin unidad de procesamiento gráfico (GPU). **Metodología**: El diseño se desarrolló mediante Design Science Research (DSR), estructurado en tres ciclos iterativos para la selección de componentes, la integración arquitectónica y el ajuste del evaluador de calidad. **Resultados**: Se propone una arquitectura de referencia que integra recuperación híbrida (búsqueda vectorial densa, recuperación léxica BM25 y fusión por rango recíproco), inferencia local con modelos de lenguaje cuantizados en formato GGUF, un constructor de prompts pedagógicos con cuatro modos de enseñanza, un evaluador automático de calidad y cifrado AES-256-GCM en reposo. La validación técnica y preliminar se realizó ejecutando el sistema desde la misma memoria USB en tres equipos con hardware heterogéneo. El sistema operó funcionalmente en un equipo con GPU dedicada y en un portátil de gama media sin GPU, mientras que un tercer equipo sin instrucciones AVX presentó un fallo de ejecución, delimitando el piso de hardware compatible. **Conclusiones**: Los resultados sugieren la viabilidad técnica de reducir las barreras de acceso a la tutoría asistida por inteligencia artificial en contextos con recursos tecnológicos y conectividad limitados; su efectividad pedagógica requiere evaluación con usuarios finales.

Palabras clave

Inteligencia artificial en educación; sistemas de tutoría inteligente; generación aumentada por recuperación; grandes modelos de lenguaje; tecnología educativa; entornos de aprendizaje sin conectividad; modelos de lenguaje locales

How to cite this article / Cómo citar este artículo

MUÑOZ-GÓMEZ, Yeison Javier; HOYOS-CERÓN, Fabian Armando; CAIZA-NARVÁEZ, Juan José. Design of a Portable Retrieval-Augmented Generation Architecture for Offline Intelligent Tutoring on Resource-Constrained Hardware. En: Entramado. Julio - diciembre, 2026. vol. 22, no. 2. p. 1-17 e-13645 https://doi.org/10.18041/1900-3803/entramado.2.13645

1. Introduction

Intelligent tutoring systems (ITS) have demonstrated promising capabilities for personalizing the learning experience and improving academic outcomes across various educational levels and domains (Kasneci et al., 2023; Woolf, 2009). However, the vast majority of current implementations presuppose permanent internet connectivity, whether to access language models hosted on remote servers or to synchronize student progress with centralized platforms. This dependency introduces a far-reaching structural limitation: according to data from the International Telecommunication Union (ITU), approximately 2.6 billion people (one-third of the world's population) lacked internet access at the end of 2023 (ITU, 2023). In Latin America, the Economic Commission for Latin America and the Caribbean (ECLAC) has documented that 46% of students in the region lived in households without internet access and that more than 40 million households lacked connectivity (ECLAC, 2020). In Colombia, the National Administrative Department of Statistics (DANE) reports that 63.9% of households have internet access, a proportion that drops to 41.4% in population centers and dispersed rural areas (DANE, 2023). These figures show that cloud dependency is not a mere technical inconvenience but a structural access barrier that excludes a substantial fraction of the student population.

An economic barrier compounds this connectivity gap. Access to commercial language model APIs such as GPT-4 or Claude entails an estimated recurring cost of 10 to 60 USD per user per month under moderate educational usage, based on the providers' published per-token rates (OpenAI, 2026), an amount that is prohibitive for institutions with limited budgets and that additionally requires transferring educational data to third-party servers outside the national jurisdiction. The combination of both barriers, connectivity and cost, constitutes the design problem that motivates this work.

Recent advances in large language model quantization, particularly through the GGUF format and the llama.cpp library (Gerganov, 2023), make it possible to run models with several billion parameters directly on conventional mid-range hardware without requiring dedicated graphics processing units. This capability, combined with retrieval-augmented generation techniques (Lewis et al., 2020), suggests that it is feasible to design a tutoring system that operates fully autonomously on instructor-provided documents, stored on and executed from a portable device such as a USB drive whose unit cost does not exceed 15 USD, dispensing with any network infrastructure or recurring subscription.

The recent state of the art nevertheless reveals a gap at the intersection of these development lines. On the one hand, established offline educational platforms such as Kolibri (Learning Equality, 2024) and RACHEL (World Possible, 2024) distribute curated educational content in connectivity-deprived contexts but lack generative capabilities and adaptive conversational tutoring. On the other hand, the literature on LLM-based tutors concentrates on cloud-dependent deployments, and systematic reviews of the field identify access equity and data privacy as open challenges (Yan et al., 2024). In parallel, the emergence of compact open-source models oriented toward efficient inference, such as TinyLlama (Zhang, Zeng, Wang and Lu, 2024) and the Phi family (Abdin et al., 2024), reinforces the feasibility of running generative capabilities on mid-range hardware. This work is positioned at the intersection of these three lines: generative capabilities with documentary grounding, fully offline operation, and low-cost pre-existing hardware.

This article describes the design of such an architecture, termed a portable intelligent tutor, with emphasis on the design decisions that enable its offline operation, its adaptability to any knowledge domain, and its portability on resource-constrained hardware. This article does not report the evaluation of a production system; rather, it presents a reference architecture whose technical viability is argued on the basis of the characteristics of the selected components, the results obtained during the design process, and the portability tests conducted on machines with heterogeneous hardware profiles.

The contributions of this work are fourfold: (i) an eight-layer reference architecture for fully offline intelligent tutoring, executable from a USB drive without installation; (ii) the documented traceability of ten design decisions with their evaluated and discarded alternatives, which supports the reproducibility of the process;

(iii) preliminary portability evidence across three heterogeneous hardware profiles, with latency and quality indicators; and (iv) the identification of three diagnostic findings (the AVX absence failure, the first-query tax, and the first-execution quality degradation) that delimit the operating conditions of the offline tutoring paradigm.

2. Conceptual Framework

2.1. Intelligent Tutoring Systems and Pedagogical Adaptation

Intelligent tutoring systems (ITS) constitute a well-established branch of educational artificial intelligence. Their classical architecture (Woolf, 2009) comprises four modules: domain model, student model, pedagogical model, and interface. Developments such as AutoTutor (Graesser et al., 2004) demonstrated that tutorial dialogue through natural language processing yielded learning gains comparable to human tutoring (effect sizes of up to 0.8σ), although they relied on rule-based models and latent semantic analysis (a statistical technique that identifies word co-occurrence patterns to infer thematic similarity), which limited their generalization to predefined domains. The incorporation of large language models (LLMs) has revitalized this field (Kasneci et al., 2023): platforms such as Khanmigo (Khan Academy, 2024) and Duolingo Max (Duolingo, 2023) employ GPT-4 as a conversational tutor but depend on external APIs that transfer the connectivity constraint and recurring cost to the core of the system. With respect to learner adaptation, systems such as ALEKS (Falmagne, Cosyn, Doignon and Thiéry, 2006) employ probabilistic knowledge models; the proposed design addresses this dimension through a memory manager that infers the student's comprehension level from linguistic cues in their queries, without relying on external databases.

2.2. Retrieval-Augmented Generation

The RAG architecture (Lewis et al., 2020) combines the generative capability of LLMs with the retrieval of relevant fragments from a document base, so that the response is grounded in verified content rather than dependent on the model's parametric knowledge, thereby reducing the hallucination phenomenon. Recent implementations incorporate hybrid retrieval (Karpukhin et al., 2020), which combines dense vector search (semantic similarity in a high-dimensional space) with sparse BM25 search, a classical relevance-scoring algorithm based on the frequency and distribution of query terms within each document (Robertson and Zaragoza, 2009). The two result lists are fused through Reciprocal Rank Fusion (RRF), a method that combines the positions of each document across multiple ranked lists to produce a more robust unified ranking (Cormack, Clarke and Buettcher, 2009); the superiority of this hybrid approach over either method alone has been documented in comparative evaluations (Gao et al., 2024). The multi-granular retrieval strategy (Sarthi et al., 2024) extends the paradigm by decoupling the search granularity (child chunks, precise for indexing) from the context granularity delivered to the model (parent chunks, discursively coherent). Studies on failure points in RAG systems (Barnett, Kurniawan, Thudumu, Brannelly and Abdelrazek, 2024) indicate that this decoupling is an effective strategy for improving retrieval precision, since small chunks favor indexing specificity while large chunks preserve the discursive coherence required for generation.

2.3. Quantized Language Models for Local Inference

Quantization reduces the numerical precision of an LLM's parameters (from 16 to 4 bits) to decrease memory requirements and accelerate inference on constrained hardware (Dettmers, Pagnoni, Holtzman and Zettlemoyer, 2023). The GGUF format, implemented in llama.cpp (Gerganov, 2023), standardizes this representation and enables CPU execution with optional GPU support. Mixed quantization variants, particularly Q4_K_M (4-bit compression with adaptive calibration), exhibit marginal degradation relative to their full floating-point counterparts. Post-training quantization techniques such as GPTQ have demonstrated that the reduction to 3–4 bits per weight produces negligible precision loss on evaluated metrics (Frantar, Ashkboos, Hoefler and Alistarh, 2023). This family of techniques makes it possible to run models with 3–8 billion parameters on machines with 8 GB of RAM and no GPU, with inference times compatible with the pace of a tutoring session, and constitutes the technical foundation of the proposed portable architecture.

2.4. Privacy and Data Sovereignty in Educational Settings

The processing of educational data on external servers raises regulatory considerations of growing importance. The General Data Protection Regulation (GDPR) in Europe, the Family Educational Rights and Privacy Act (FERPA) in the United States, and Law 1581 of 2012 in Colombia (Colombia. Congreso de la República, 2012) establish explicit restrictions on the transfer of students' personal information to third parties (Prinsloo and Slade, 2017). In the Colombian context, Decree 1377 of 2013 (Colombia. Presidencia de la República, 2013), which regulates Law 1581, requires express authorization for the processing of minors' data and prohibits their transfer to servers outside the national jurisdiction without verifiable consent. Learning analytics administered by external providers additionally create an informational power asymmetry that may compromise institutional autonomy (Prinsloo and Slade, 2017). The proposed tutor complies with these provisions by design: because it runs entirely from the USB device, no session data, query, or student profile leaves the host machine during the session, in line with the data governance and privacy principles that the OECD sets out for digital education ecosystems (OECD, 2023).

2.5. Digital Divide and the Colombian Educational Context

The relevance of an offline architecture is better understood in light of the country's educational conditions. DANE (2023) figures indicate that internet access reaches 63.9% of households nationally but drops to 41.4% in population centers and dispersed rural areas, where a considerable share of public school enrollment is concentrated. ECLAC (2020) further documented that 46% of students in Latin America lived in households without a connection, a condition that translated into prolonged interruptions of educational services during the pandemic and exposed the fragility of network-dependent remote education strategies. In the Colombian case, the divide is not limited to connectivity: it also encompasses equipment and infrastructure constraints in rural schools, which reduces the applicability of solutions that presuppose permanent cloud services. These conditions suggest three requirements for educational technology oriented to such contexts: network independence, operation on modest pre-existing hardware, and near-zero marginal cost per student. The architecture proposed here was designed around these three requirements; its contribution should accordingly be understood as one of technical feasibility that enables (but does not by itself ensure) educational improvement processes, whose verification requires studies with end users under real conditions.

2.6. Comparable Systems and Positioning of the Proposal

Table 1 contrasts the most relevant tutoring systems with the proposed design in order to identify the architectural niche the proposal occupies.

Table 1.
Comparison of intelligent tutoring systems with the proposed design

System	Connectivity	AI Engine	RAG	Portable	Local Encryption	Cost
Khanmigo (Khan Academy, 2024)	Cloud	GPT-4 (API)	No	No	No	~$44/year
Duolingo Max (Duolingo, 2023)	Cloud	GPT-4 (API)	No	No	No	~$168/year
AutoTutor (Graesser et al., 2004)	Offline (partial)	Rules + LSA	No	No	No	Institutional
ALEKS (Falmagne et al., 2006)	Cloud	Probabilistic	No	No	No	~$20/month
ChatGPT Edu (OpenAI, 2024)	Cloud	GPT-4o (API)	No	No	Partial	Institutional

Continued on the next page

System	Connectivity	AI Engine	RAG	Portable	Local Encryption	Cost
Cognii (Cognii, 2024)	Cloud	Proprietary NLP	Partial	No	Partial	Institutional
PrivateGPT (Martínez-Toro, 2023)	Offline	Local LLM	Yes (vector)	No	Yes	Free
Proposal	Offline (full)	Local GGUF LLM	Yes (5-stage hybrid)	Yes (USB)	Yes (AES-256-GCM)	~$15 (USB)

Note: Author's own elaboration. ALEKS costs (~$20/month) are equivalent to ~$240/year. All other costs are expressed on an annual basis to facilitate comparison.

The comparative analysis reveals three findings. First, the commercial systems with the highest pedagogical quality (Khanmigo, Duolingo Max, ChatGPT Edu) depend on OpenAI APIs, which ties their availability to the external service and entails recurring costs of 44 to 168 USD per year, incompatible with resource-constrained institutions. Second, systems that support offline operation exhibit complementary shortcomings: AutoTutor employs rules and latent semantic analysis without modern generative capability, and PrivateGPT lacks USB portability, a multi-stage hybrid RAG pipeline, and local encryption. Third, the specific combination of fully offline operation, hybrid RAG with semantic reranking, installation-free portability, and encryption at rest is not found in any of the reviewed systems, which constitutes the architectural niche of the proposal.

3. Methodology

The design of the portable intelligent tutor was approached through a Design Science Research (DSR) methodology (Hevner, March, Park and Ram, 2004), which conceives the research process as the iterative construction and evaluation of a technological artifact aimed at solving an identified practical problem. Unlike descriptive or explanatory research approaches, DSR produces as its primary outcome the artifact itself (in this case, the system architecture and its design components), whose contribution to knowledge resides in demonstrating that the problem can be solved in a particular manner and in articulating the design decisions that make that solution possible. This approach is appropriate given that the objective of this work is not to evaluate the behavior of a production system but rather to propose and justify a reproducible reference architecture.

The process was structured following the six activities of the Design Science Research Methodology (DSRM) proposed by Peffers, Tuunanen, Rothenberger and Chatterjee (2007): problem identification and motivation, definition of solution objectives, design and development of the artifact, demonstration, evaluation, and communication, complemented by a review of the relevant literature that cuts across all phases. Figure 1 illustrates the adopted methodological cycle and the iterations traversed during the design process. Each arrow in the cycle represents an actual iteration in which the results of one phase fed back into earlier decisions: the hardware requirements identified during the demonstration phase, particularly the absence of AVX vector instructions in older-generation machines, fed back into the component selection phase, illustrating the iterative nature of the DSR cycle.

3.1. Design Requirements and Evaluation Criteria

The solution objectives were delimited around four non-negotiable requirements: fully offline operation once the initial configuration process is complete, portability on conventional hardware without software installation on the user's machine, adaptability to any educational domain through the indexing of the instructor's own documents, and data protection through encryption at rest. These requirements guided all subsequent design decisions. To make them verifiable, each requirement was associated with an evaluation criterion and an observable indicator, as summarized in Table 2. Within the scope of this work, the architecture is considered viable if it simultaneously satisfies the four criteria on at least one mid-range machine without a graphics processing unit, a condition that corresponds to the design's target usage scenario.

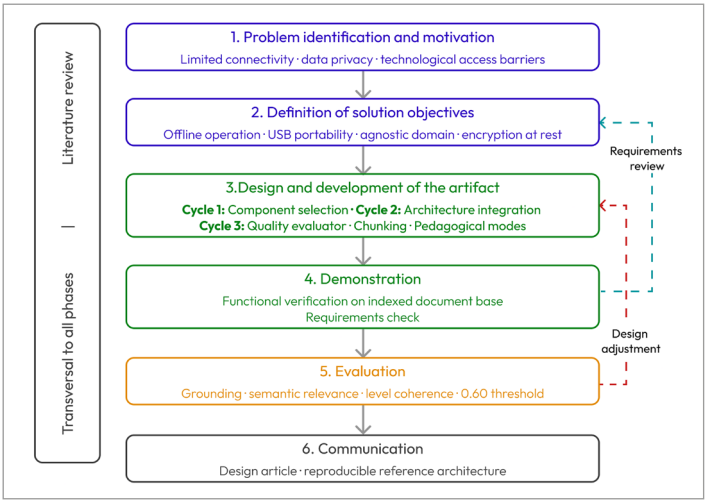

Figure 1. Design Science Research methodological cycle applied to the design of the portable intelligent tutor.
Note: Author's own elaboration

Table 2.
Design requirements, evaluation criteria, and observable indicators.

Requirement	Evaluation Criterion	Observable Indicator	Result
R1. Fully offline operation	Absence of network traffic during the session	External connections blocked by the startup script; functional session without an active network adapter	Met (Machines A and B)
R2. Installation-free portability	Execution from the same USB drive on heterogeneous machines without modifying the host	Successful startup and functional session; 0 GB written to the host disk	Met on A and B; not met on C (no AVX)
R3. Domain adaptability	Indexing of arbitrary instructor-provided documents	Ingestion of PDF/DOCX corpus and fragment retrieval in session	Met (test corpus)
R4. Data protection	Encryption at rest with a non-persisted key	AES-256-GCM with in-memory PBKDF2 derivation; functional encryption/ decryption verification per session	Met

Note: Author's own elaboration.

3.2. Design Cycles and Artifact Development

The artifact design was organized into three iterative cycles whose decisions are documented in the design-decision traceability table (Table 3). Each cycle produced a set of decisions that were evaluated against the defined requirements; discarded alternatives are included to show that the selection was not arbitrary but the result of a systematic evaluation process.

Table 3.
Design decision traceability by DSR cycle

Cycle	Design Decision	Alternatives Evaluated	Adopted Alternative	Rationale for Discarding Alternatives
1	LLM inference engine	(a) transformers (HF), (b) llama-cpp-python, (c) CTranslate2	(b) llama-cpp-python	(a) requires 2× more RAM and automatic model downloads; (c) does not support GGUF format
1	Embedding model	(a) all-MiniLM-L6-v2, (b) paraphrase-multilingual-MiniLM-L12-v2, (c) e5-base	(b) Multilingual MiniLM	(a) English only; (c) 3× larger disk footprint (420 MB vs. 180 MB)

Continued on the next page

Cycle	Design Decision	Alternatives Evaluated	Adopted Alternative	Rationale for Discarding Alternatives
1	Sparse text search	(a) Whoosh, (b) rank-bm25, (c) Elasticsearch	(b) rank-bm25	(a) requires persistent on-disk index; (c) requires JVM service incompatible with USB portability
1	API framework	(a) Flask, (b) FastAPI + Uvicorn, (c) Django	(b) FastAPI + Uvicorn	(a) no native SSE support; (c) excessive footprint for a single-endpoint REST API
2	Encryption protocol	(a) AES-256-CBC, (b) AES-256-GCM, (c) ChaCha20-Poly1305	(b) AES-256-GCM	(a) no integrated authentication (vulnerable to bit-flipping); (c) lower performance on CPUs with AES-NI
2	API-Frontend communication	(a) WebSocket, (b) SSE (EventSource), (c) Long Polling	(b) SSE	(a) unnecessary complexity for unidirectional flow; (c) perceptible latency between tokens
2	Portable Python environment	(a) Anaconda Portable, (b) WinPython 3.11, (c) PyInstaller exe	(b) WinPython 3.11	(a) 4× larger footprint (~6 GB); (c) fragile compilation with native dependencies (llama-cpp)
3	Evaluator metric	(a) Jaccard only, (b) embeddings only, (c) weighted composite metric	(c) Composite metric	(a) insensitive to semantic relationships; (b) excessive computational cost in CPU-only mode
3	Child chunk size	(a) 64 tokens, (b) 128 tokens, (c) 256 tokens	(b) 128 tokens	(a) excessive fragmentation that loses coherence; (c) degraded semantic precision in 384-dim vector spaces (Barnett et al., 2024)
3	Pedagogical modes	(a) single mode, (b) 4 parameterizable modes	(b) 4 modes	(a) insufficient to cover the spectrum of didactic strategies (Anderson and Krathwohl, 2001)

Note: Author's own elaboration.

The cross-cutting analysis of Table 3 reveals a recurring pattern: in each of the ten decisions, the adopted alternative was the one that offered the lowest resource consumption (memory, disk, external dependencies) while maintaining the required functionality. Cycle 1 prioritized compatibility with the portable environment, discarding tools that required background services or automatic downloads. Cycle 2 consolidated security and communication without adding unnecessary complexity. Cycle 3 calibrated the RAG engine and quality evaluator parameters, balancing precision and computational cost in CPU-only scenarios.

In terms of activities and products, cycle 1 (component selection) comprised the review of inference, embedding, and search libraries, together with the verification of their compatibility with the portable environment; its product was the base technology stack (Table 4). Cycle 2 (architectural integration and security) addressed interface-server communication, the encryption protocol, and portable packaging; its product was the eight-layer architecture (Figure 2). Cycle 3 (calibration) tuned the retrieval engine parameters ($k = 60$, $\alpha = 0.6$, 128-token child chunks) and the quality evaluator (0.60 threshold and the weights of Equation 2); its product was the reference configuration used in the demonstration phase. Iterations between phases were documented: the AVX absence failure, detected during the demonstration, fed back into cycle 1 and incorporated processor generation as a compatibility constraint (Section 4.6).

3.3. Validation Procedure and Scope

The demonstration phase consisted of running the system from the same USB drive, without modifications between tests, on three machines selected through convenience sampling, with profiles representative of the anticipated usage scenarios: a development computer with a dedicated GPU (upper performance bound), a mid-range laptop without a GPU (the design's target case), and a low-end machine of an earlier generation (lower compatibility bound). In each run, the detected hardware profile, the assigned context window, per-query latencies, and the automatic evaluator score were recorded from the system logs. This

validation is technical and preliminary in scope: it does not include the participation of end users, it is limited to a single operating system (Windows 10/11), and its sample size (n = 3) does not allow statistical generalization. Its purpose is to verify compliance with the criteria in Table 2 and to document the artifact's operating conditions; pedagogical evaluation is reserved for future work (Section 7).

4. Design Results

4.1. General Architecture and Technology Stack

The portable intelligent tutor brings together on a single USB device all the components required for offline tutoring sessions: portable Python environment, quantized language models, indexed document base, RAG engine, API server, and activity logs. The user accesses the system through a local web browser that consumes a REST API with streaming via Server-Sent Events (SSE), a unidirectional communication protocol that allows the server to progressively send data to the browser without the browser repeatedly requesting it, enabling progressive token-by-token response delivery. Figure 2 illustrates the eight-layer functional architecture with unidirectional top-down dependencies. The architecture is domain-agnostic: all educational content resides in the instructor's documents, and the RAG engine retrieves the relevant context for each query.

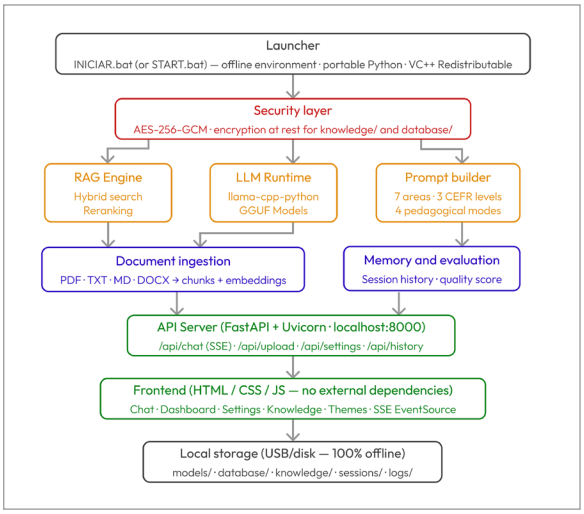

Figure 2. General system architecture: component structure by layers of the portable intelligent tutor.
Note: Author's own elaboration

Component selection was governed by five criteria: offline operation, mid-range CPU performance, disk footprint compatible with 8 GB, educational-use licenses, and stability. Table 4 summarizes the adopted stack.

Table 4.
Technology stack of the portable intelligent tutor

Component	Technology	Selected Variant
LLM Inference	llama-cpp-python	GGUF Q4_K_M (~2–3 GB/model)
Embeddings	sentence-transformers (Reimers and Gurevych, 2019)	paraphrase-multilingual-MiniLM-L12-v2 · 384 dim.
Reranking	sentence-transformers (Nogueira and Cho, 2019)	ms-marco-MiniLM-L-6-v2
Sparse Search	rank-bm25	BM25Okapi

Continued on the next page

Component	Technology	Selected Variant
API Server	FastAPI + Uvicorn	REST + SSE · 127.0.0.1:8000
Encryption	cryptography	AES-256-GCM · PBKDF2-HMAC-SHA256 (600K iter.)
Text Extraction	PyMuPDF / python-docx	PDF and DOCX
Portable Environment	WinPython 3.11	~1.5 GB · no installation
Interface	HTML / CSS / JS (native)	No external frameworks

Note: Author's own elaboration.

The llama-cpp-python engine loads GGUF models directly into RAM, reducing memory consumption by approximately 75% relative to 16-bit formats, which makes it possible to run 3-billion-parameter models on 8 GB of RAM with optional GPU layer offloading. The security layer encrypts and decrypts files at the beginning and end of each session using the AES-256-GCM algorithm (an authenticated encryption standard that ensures both confidentiality and data integrity, detecting any unauthorized alteration) with random initialization vectors. The encryption key is entered manually per session and derived in memory via PBKDF2 (a key derivation function that applies 600,000 hashing iterations to hinder brute-force attacks) without being stored on disk at any point. An initialization script blocks any connection attempt to external repositories and automatically calibrates the model's context window between 2,048 and 8,192 tokens based on the available RAM on the host machine. Table 5 summarizes the hardware requirements.

Table 5.
Minimum hardware requirements

Resource	Minimum	Recommended	Impact if Below
RAM	8 GB	16 GB	Context reduced to 2,048 tokens
CPU	x86-64, 4 cores, AVX	4+ cores, AVX2	Without AVX: error 0xc000001d
GPU	Not required	4+ GB VRAM	Without GPU: substantially higher latency (not quantified in this study)
USB	8 GB (USB 3.0)	16 GB	Model capacity limit
OS	Windows 10 x64	Windows 10/11 x64	Earlier versions not tested
Host Disk	0 GB	N/A	Direct execution from USB

Note: Author's own elaboration.

4.2. Hybrid RAG Engine

The RAG engine integrates five sequential stages whose flow is described in Figure 3. First, up to four query variants are generated (two languages). Second, each variant is submitted in parallel to vector search (cosine similarity, 384 dimensions) and BM25, retrieving the 30 most relevant fragments per method (60 candidates in total before fusion), with a latency below 20 milliseconds on collections of 200 to 2,000 indexed fragments. Third, the RRF algorithm (Cormack et al., 2009) fuses the ranked lists:

$$score_{\mathcal{RRF}}(\mathrm{d}) = \alpha \cdot \frac{1}{k+rank_{\mathcal{VEC}}(d)} + (1-\alpha) \cdot \frac{1}{k+rank_{BM25}(d)} \quad (1)$$

The fusion uses k = 60 and α = 0.6 favoring semantic retrieval. Fourth, each child chunk (128 tokens, 32-token overlap) is replaced by its parent (1,024 tokens), preserving discursive coherence (Barnett et al., 2024). Fifth, a CrossEncoder semantic reranking model, ms-marco-MiniLM-L-6-v2 (Nogueira and Cho, 2019), jointly evaluates each (query, candidate chunk) pair as a unit of meaning, producing a relevance score more precise than isolated vector similarity, with a latency of 50 to 150 milliseconds for ten candidates. The top 5 fragments are incorporated into the prompt.

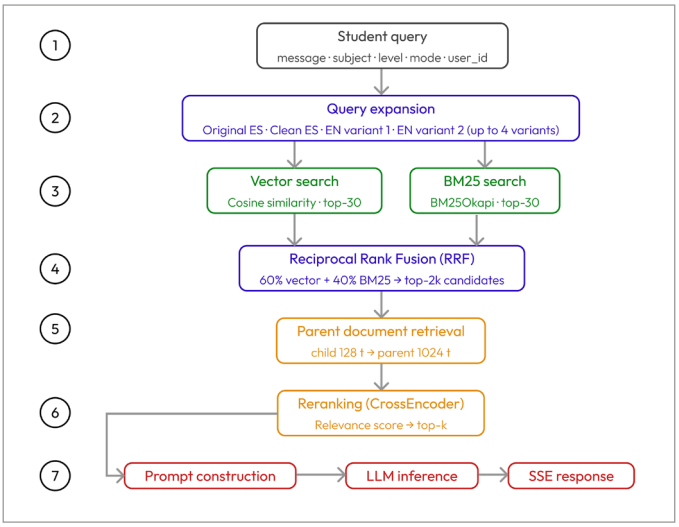

Figure 3. Hybrid RAG pipeline: query processing flow from student input to SSE response.
Note: Author's own elaboration

4.3. Pedagogical Prompt Builder

The builder dynamically assembles the system prompt, student profile, compressed history, and RAG fragments. The prompt is parameterized along three dimensions: subject area, competence level (automatic detection), and pedagogical mode based on Bloom's taxonomy (Anderson and Krathwohl, 2001). Table 6 describes the four available modes.

Table 6.
Pedagogical modes of the prompt builder

Mode	Behavior	Indicated Context
Normal	Direct explanation with examples	Introduction to new concepts
Socratic	Guiding questions without revealing the answer	Critical thinking
Quiz	Assessment question with feedback	Self-evaluation
Visual	Comparative tables and diagrams	Comparisons and taxonomies

Note: Author's own elaboration.

The conversational history employs a sliding window of five recent exchanges; when the session exceeds ten messages, the language model itself automatically generates a compressed three-line summary that replaces the full history, preserving context without exceeding the processing window limits. The student profile is persisted to disk between sessions.

4.4. User Interface

The interface is a static web application served by FastAPI without external dependencies, enabling fully offline operation. It adopts a neumorphic design (a visual style that employs soft shadows and subtle reliefs to simulate physical depth in interface elements) with five views: chat, document management (drag-and-drop), visual customization (eight palettes and eight backgrounds), technical settings, and metrics dashboard. Response delivery via SSE allows users to perceive activity immediately during inference times of 54–111 seconds. Figure 4 and Figure 5 illustrate the main views.

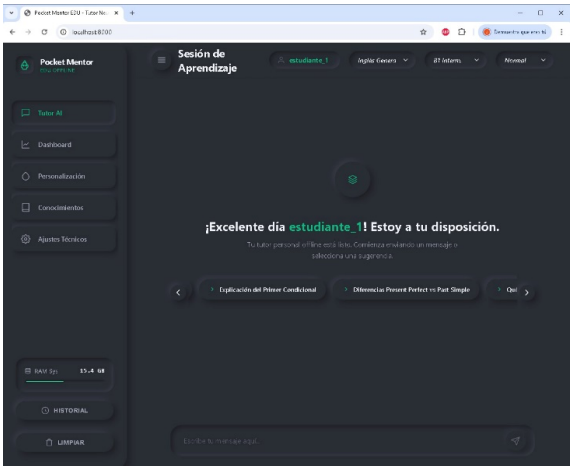

Figure 4. Main chat view of the portable intelligent tutor: neumorphic interface with sidebar, suggestion carousel, and response bubble.

Note: Author's own elaboration

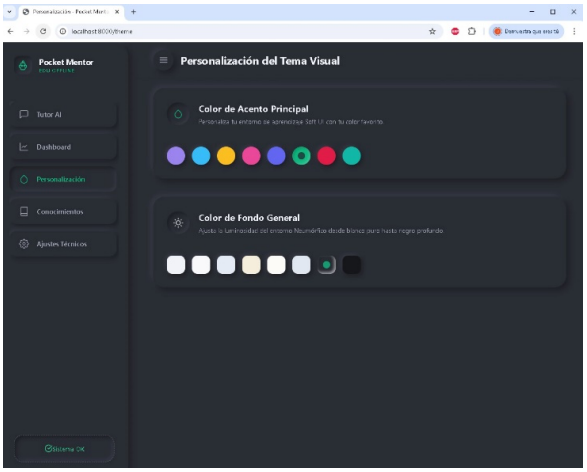

Figure 5. Theme customization screen: color palette selector and configurable backgrounds.

Note: Author's own elaboration

4.5. Automatic Quality Evaluation

The evaluator monitors each response through a composite metric:

$$Q(r,q,\mathcal{C}) = 0.4 \cdot grounding(r,\mathcal{C}) + 0.4 \cdot relevance(q,\mathcal{C}) + 0.2 \cdot level\_coherence(r,profile) \quad (2)$$

which quantifies documentary grounding (i.e., what proportion of the response is supported by the retrieved fragments), the relevance of the retrieved context to the original query, and the response's appropriateness to the student's level. For example, for a response with $grounding$ = 0.82, $relevance$ = 0.79, and $level\_coherence$ = 0.71, the resulting composite score is Q = 0.4(0.82) + 0.4(0.79) + 0.2(0.71) = 0.786. The evaluator reuses the RAG engine's embedding model, avoiding the duplication of ~500 MB in RAM. It operates in three modes according to the desired speed-precision trade-off: disabled (no computational cost), fast via Jaccard lexical similarity (latency below 1 millisecond, default mode), and full via cosine semantic similarity between embedding vectors (200–500 milliseconds). When the score does not exceed the 0.60 threshold, a second response is automatically generated with reduced temperature.

4.6. Portability Validation

The system was executed from the same USB drive, without modifications between tests, on the three machines whose selection and profiles are described in Section 3.3. Table 7 summarizes the results. On Machine A, the disaggregated measurement of three consecutive queries recorded 111.06 seconds for the first one (62.20 s attributable to the retrieval stage with lazy model loading and 48.86 s to generation) and 86.18 and 54.22 seconds for the two subsequent ones, with a generation throughput of 2.01 to 4.38 tokens per second.

Table 7.
Portability validation on heterogeneous hardware

Parameter	Machine A (developer)	Machine B (field, successful)	Machine C (field, failed)
RAM	31.6 GB	7.7 GB	3.8 GB
GPU	RTX 3050 (35 layers)	None	None
AVX	Yes	Yes	No
n_ctx assigned	8,192 tokens	2,048 tokens	2,048 tokens
1st query latency	111 s	Not measured	Critical failure
Subsequent latency	54–86 s	Not measured	Critical failure
1st response quality	Not measured	43%	Critical failure
Result	Functional	Functional	Critical failure

Note: Author's own elaboration.

The dynamic context window adjustment mechanism functioned as intended: Machine A, with 31.6 GB of RAM, operated with an 8,192-token window, while Machines B and C were automatically limited to 2,048 tokens based on available resources. The first query of each session recorded a latency of 111 seconds on Machine A, substantially higher than that of subsequent queries (54–86 s), because the semantic models of the retrieval engine are loaded into memory only when first required; from the second query onward, these models remain in memory and latency is substantially reduced. Machine B (7.7 GB, no GPU) confirmed viability on mid-range hardware, although the first response obtained a quality score of 43%, below the 0.60 approval threshold, a phenomenon analyzed in Section 5. Machine C exhibited a critical incompatibility due to the absence of AVX instructions, also analyzed in Section 5.

Taken together, this evidence supports compliance with the viability criteria of Table 2 in the target use case (mid-range hardware without a GPU) and delimits the compatible hardware floor. It does not, however, allow conclusions about pedagogical effectiveness or the generalization of this behavior to other operating systems or configurations, aspects reserved for future work (Section 7).

5. Discussion

The portability tests documented in Section 4.6 suggest that the combination of quantized models, hybrid retrieval, and parameterizable pedagogical prompts constitutes a technically viable architecture for offline intelligent tutoring. USB portability reduces the access barrier to a one-time cost of ~15 USD, compared to the 44–168 USD per year of cloud-API-based systems, positioning this architecture as a promising alternative for institutions with limited resources or privacy constraints. Nevertheless, the field tests revealed limitations that are analyzed as diagnostic findings of the design process.

In comparative perspective, the results contrast with prior work on three planes. First, they corroborate the evidence of the review by Gao et al. (2024): the combination of hybrid retrieval with CrossEncoder reranking consistently improves precision relative to isolated vector search, a result coherent with the five-

stage architecture implemented; along the same line, the adopted child-parent chunking strategy responds directly to the failure points documented by Barnett et al. (2024), who identify the coupling between search granularity and context granularity as a source of imprecision, and the dense retrieval combined with BM25 follows the line of Karpukhin et al. (2020). Second, they extend the scope of existing offline systems: PrivateGPT (Martínez-Toro, 2023) operates with local models but does not report latency data on hardware without a GPU nor does it implement a multi-stage pipeline with semantic reranking; the evidence presented here provides operating indicators precisely for that scenario. Third, they differ from the response times of cloud systems: the 54–111-second latency recorded on Machine A contrasts with the sub-3-second times of Khanmigo (Khan Academy, 2024) or ChatGPT Edu (OpenAI, 2024), although the latter shift the computational cost to the provider in exchange for a recurring subscription and the transfer of educational data. Compared with AutoTutor (Graesser et al., 2004), a pioneer in natural language processing-based tutoring but limited to rules and latent semantic analysis with closed vocabularies, the proposed architecture extends the tutor's knowledge via RAG to any document corpus provided by the instructor.

In the hardware domain, an "AVX absence failure" was identified: Machine C halted with error 0xc000001d (Illegal Instruction) due to the lack of AVX/AVX2 vector instructions, redefining the compatible hardware floor in terms not only of resource quantity but also of processor generation. Complementarily, a "pedagogical latency effect" was observed: the inference latencies of 54–111 s measured on Machine A (equipped with a GPU) may interrupt the continuity of the learning flow, whereas latency on Machine B, which has no GPU, was not measured. Improvement directions include packaging a library variant without AVX (generic build) with automatic detection at startup, and evaluating sub-2B models (1–1.5 billion parameters) that could reduce latency below 10 seconds, although this hypothesis requires empirical validation.

Regarding first-execution behavior, two related phenomena were documented (Table 7). The "first-query tax" recorded 111 seconds on Machine A versus a range of 54–86 seconds for subsequent queries, due to the lazy loading of the RAG engine's semantic models: the encoder and reranker are initialized only when the user submits the first query. Associated with this phenomenon, the "first-execution quality degradation" produced a composite score of 43% on Machine B (below the 0.60 threshold), attributable to "cold" generation without prior session context and with a reduced context window (2,048 tokens). Both effects are progressively mitigated with use: subsequent queries benefit from accumulated history and the cache of already-loaded models. An asynchronous preloading mechanism (background preloading) constitutes the most direct future improvement.

In terms of educational implications, the scope of these results must be delimited with precision. The architecture enables, in institutions without stable connectivity, document-grounded tutoring with generative capabilities, data sovereignty, and a near-zero marginal cost per student; these are necessary, but not sufficient, conditions for learning improvement. Pedagogical effectiveness (comprehension, retention, and the didactic adequacy of responses) remains a working hypothesis whose verification requires protocols with instructors and students under real conditions of use, including students' tolerance of the 54–111-second latencies recorded on the GPU-equipped machine.

Finally, the design has two scope limitations. First, the linear-complexity O(n) vector search (that is, one whose execution time grows proportionally to the number of indexed fragments), while adequate for typical educational-scale corpora, could become a bottleneck in collections of several thousand fragments; the adoption of approximate search indices, such as FAISS (an efficient vector search library), constitutes a future improvement direction that would enable scaling the system to larger document collections without perceptible degradation in response times. Second, the static quality evaluator threshold (0.60) does not account for variations in the corpus's informational density, which may produce a "partial coverage drift" when the corpus contains scarce information on the queried topic. Automatic evaluation, moreover, captures computable dimensions but does not address higher-order pedagogical aspects (didactic adequacy, conceptual correctness, cultural relevance), whose validation requires protocols with instructors and students under real-use conditions.

6. Conclusions

This work has proposed and justified a reference architecture for a portable intelligent tutor that operates fully offline, through a Design Science Research process structured in three iterative cycles. The main contributions and reflections derived from the process are set out below.

First, the eight-layer architecture with unidirectional dependencies shows that the combination of local inference with quantized models, five-stage hybrid retrieval, and parameterizable pedagogical prompts constitutes a cohesive design for offline intelligent tutoring. The comparative table in Section 2.6 shows that this specific combination (fully offline operation, hybrid RAG with reranking, USB portability, and local encryption) is not present in any of the seven reviewed systems, positioning the proposal in an architectural niche not covered by the available literature.

Second, the decision traceability documented in Table 3 constitutes a methodological contribution in itself. Unlike publications that present their design decisions as settled facts, the explicit documentation of evaluated and discarded alternatives (three alternatives for each of the ten critical decisions) enables future researchers to reproduce the design process in contexts with different constraints, adapting decisions to their own requirements without needing to repeat the preliminary evaluations.

Third, the field tests analyzed in Section 5 revealed three diagnostic findings (the "AVX absence failure," the "first-query tax," and the "first-execution quality degradation") that delimit the system's operating conditions and reveal an inherent trade-off in the offline tutoring paradigm: inference latency (54–111 s, measured with a GPU; not quantified without one) imposes an interactivity ceiling that is offset by the elimination of recurring costs and independence from network infrastructure. In a context where only 41.4% of households in population centers and dispersed rural areas of Colombia have internet access (DANE, 2023), this architecture suggests a concrete pathway for resource-constrained educational institutions to explore the incorporation of AI-assisted tutoring without depending on connectivity or subscriptions to external services.

These conclusions are circumscribed to the scope of the evidence presented: a preliminary technical validation on three machines, on a single operating system, and without the participation of end users. Claims about pedagogical effectiveness or educational impact would require the user studies described in the future work section.

7. Future Work

Based on the findings documented in Sections 5 and 6, five directions for future research are identified.

- First, the quantitative evaluation of system performance through a reproducible benchmark with multiple language models and generation configurations, enabling the establishment of objective quality thresholds for different hardware profiles.
- Second, pedagogical validation with expert instructors and students under real-use conditions through evaluation protocols with Likert scales, which would allow contrasting the evaluator's automatic scores (Equation 2) with human assessments and verifying whether the system produces measurable learning gains.
- Third, the exploration of sub-2B models (1–1.5 billion parameters) that could reduce latency below 10 seconds on CPU, qualitatively altering the interaction dynamics.
- Fourth, the compilation of an inference library variant compatible with processors lacking AVX vector instructions, which would extend system coverage to machines manufactured before 2011.
- Fifth, the implementation of an asynchronous preloading mechanism to reduce the "first-query tax" documented in Section 5, together with the extension to additional operating systems (Linux, macOS) and portability evaluation on lower-capacity storage devices.

About the authors

Yeison Javier Muñoz-Gómez

Engineering Faculty, Institución Universitaria Colegio Mayor del Cauca, Popayán, Colombia, Researcher

yjmunoz@unimayor.edu.co https://orcid.org/0009-0007-5178-0667

Fabian Armando Hoyos-Cerón

Engineering Faculty, Institución Universitaria Colegio Mayor del Cauca, Popayán, Colombia, Researcher

fahoyos@unimayor.edu.co https://orcid.org/0009-0007-3338-787X

Juan José Caiza-Narváez

Engineering Faculty, Institución Universitaria Colegio Mayor del Cauca, Popayán, Colombia, Professor

juanjosecaiza@unimayor.edu.co https://orcid.org/0000-0002-8582-5946

Acknowledgments

The authors express their gratitude to the Institución Universitaria Colegio Mayor del Cauca for the institutional support provided for the development of this research, to the I+D Informática research group of the Faculty of Engineering for the academic and scientific support, and to the BETABIT student research group for its active contribution to the design, testing, and validation of the proposed system.

Data Availability

The data supporting the results of this study, including the portability test logs conducted on the three machines, are available upon request to the corresponding author.

Disclosure statement

The authors declare that there is no potential conflict of interest related to the article.

Disclaimer

The opinions, interpretations, and conclusions expressed in this article are the sole responsibility of the authors and do not represent the official position of the Institución Universitaria Colegio Mayor del Cauca.

Funding

This research did not receive specific funding from any entity in the private, public, commercial, or non-profit sectors.

Co-authorship

Yeison Javier Muñoz-Gómez: Conceptualization; Methodology; Project Administration; Original Draft Writing; Software Development.

Fabian Armando Hoyos-Cerón: Software Development; Validation; Visualization.

Juan José Caiza-Narváez: Literature Review; Manuscript Review; Editing.

All authors read and approved the final version of the manuscript.

References

1. ABDIN, Marah; ANEJA, Jyoti; BEHL, Harkirat; BUBECK, Sébastien; ELDAN, Ronen; GUNASEKAR, Suriya; HARRISON, Michael; HEWETT, Russell J.; JAVAHERIPI, Mojan; KAUFFMANN, Piero; LEE, James R.; LEE, Yin Tat; LI, Yuanzhi; LIU, Weishung; MENDES, Caio C. T.; NGUYEN, Anh; PRICE, Eric; DE ROSA, Gustavo; SAARIKIVI, Olli; SALIM, Adil; SHAH, Shital; WANG, Xin; WARD, Rachel; WU, Yue; YU, Dingli; ZHANG, Cyril; ZHANG, Yi. Phi-4 technical report [preprint]. 2024. https://doi.org/10.48550/arXiv.2412.08905

2. ANDERSON, Lorin W.; KRATHWOHL, David R. A taxonomy for learning, teaching, and assessing: a revision of Bloom's taxonomy of educational objectives. New York: Longman, 2001. 352 p.

3. BARNETT, Scott; KURNIAWAN, Stefanus; THUDUMU, Srikanth; BRANNELLY, Zach; ABDELRAZEK, Mohamed. Seven failure points when engineering a retrieval augmented generation system. In: Proceedings of the IEEE/ACM International Conference on AI Engineering (CAIN 2024). 2024. p. 194-199. https://doi.org/10.1145/3644815.3644945

4. COGNII. Artificial intelligence for education and training. 2024. https://www.cognii.com/
5. COLOMBIA. CONGRESO DE LA REPÚBLICA. Ley 1581. (17, octubre, 2012). Por la cual se dictan disposiciones generales para la protección de datos personales. Diario Oficial. Bogotá, D.C., 2012. No. 48587.
6. COLOMBIA. PRESIDENCIA DE LA REPÚBLICA. Decreto 1377. (27, junio, 2013). Por el cual se reglamenta parcialmente la Ley 1581 de 2012. Diario Oficial. Bogotá, D.C., 2013. No. 48834.
7. CORMACK, Gordon V.; CLARKE, Charles L. A.; BUETTCHER, Stefan. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In: Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2009). 2009. p. 758-759. https://doi.org/10.1145/1571941.1572114
8. DANE. Indicadores básicos de tenencia y uso de tecnologías de la información y las comunicaciones (TIC) en hogares y personas de 5 y más años de edad 2023. Bogotá D.C.: DANE, 2023. https://www.dane.gov.co/index.php/estadisticas-por-tema/tecnologia-e-innovacion/tecnologias-de-la-informacion-y-las-comunicaciones-tic/indicadores-basicos-de-tic-en-hogares
9. DETTMERS, Tim; PAGNONI, Artidoro; HOLTZMAN, Ari; ZETTLEMOYER, Luke. QLoRA: efficient finetuning of quantized LLMs [preprint]. 2023. https://doi.org/10.48550/arXiv.2305.14314
10. DUOLINGO. Duolingo Max uses OpenAI's GPT-4 for new learning features. 2023. https://blog.duolingo.com/duolingo-max/
11. ECLAC. Universalizing access to digital technologies to address the consequences of COVID-19. Santiago de Chile: United Nations, 2020. https://www.cepal.org/en/publications/45939-universalizing-access-digital-technologies-address-consequences-covid-19
12. FALMAGNE, Jean-Claude; COSYN, Eric; DOIGNON, Jean-Paul; THIÉRY, Nicolas. The assessment of knowledge, in theory and in practice. In: MISSAOUI, Rokia; SCHMID, Jürg (Eds.). Formal concept analysis. Berlin: Springer, 2006. p. 61-79. https://doi.org/10.1007/11671404_4
13. FRANTAR, Elias; ASHKBOOS, Saleh; HOEFLER, Torsten; ALISTARH, Dan. GPTQ: accurate post-training quantization for generative pre-trained transformers [preprint]. 2023. https://doi.org/10.48550/arXiv.2210.17323
14. GAO, Yunfan; XIONG, Yun; GAO, Xinyu; JIA, Kangxiang; PAN, Jinliu; BI, Yuxi; DAI, Yi; SUN, Jiawei; WANG, Meng; WANG, Haofen. Retrieval-augmented generation for large language models: a survey. In: Proceedings of the Conference on AI, Science, Engineering, and Technology (AIxSET 2024). 2024. p. 166-169. https://doi.org/10.1109/AIxSET62544.2024.00030
15. GERGANOV, Georgi. llama.cpp: LLM inference in C/C++ [software]. 2023. https://github.com/ggml-org/llama.cpp
16. GRAESSER, Arthur C.; LU, Shulan; JACKSON, George Tanner; MITCHELL, Heather Hite; VENTURA, Matthew; OLNEY, Andrew; LOUWERSE, Max M. AutoTutor: a tutor with dialogue in natural language. In: Behavior Research Methods, Instruments, & Computers. 2004. vol. 36, no. 2. p. 180-192. https://doi.org/10.3758/BF03195563
17. HEVNER, Alan R.; MARCH, Salvatore T.; PARK, Jinsoo; RAM, Sudha. Design science in information systems research. In: MIS Quarterly. 2004. vol. 28, no. 1. p. 75-105. https://doi.org/10.2307/25148625
18. ITU. Facts and Figures 2023: Internet use. Geneva: International Telecommunication Union, 2023. https://www.itu.int/itu-d/reports/statistics/2023/10/10/ff23-internet-use/
19. KARPUKHIN, Vladimir; OĞUZ, Barlas; MIN, Sewon; LEWIS, Patrick; WU, Ledell; EDUNOV, Sergey; CHEN, Danqi; YIH, Wen-tau. Dense passage retrieval for open-domain question answering. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2020). 2020. p. 6769-6781. https://doi.org/10.18653/v1/2020.emnlp-main.550
20. KASNECI, Enkelejda; SESSLER, Kathrin; KÜCHEMANN, Stefan; BANNERT, Maria; DEMENTIEVA, Daryna; FISCHER, Frank; GASSER, Urs; GROH, Georg; GÜNNEMANN, Stephan; HÜLLERMEIER, Eyke; KRUSCHE, Stephan; KUTYNIOK, Gitta; MICHAELI, Tilman; NERDEL, Claudia; PFEFFER, Jürgen; POQUET, Oleksandra; SAILER, Michael; SCHMIDT, Albrecht; SEIDEL, Tina; STADLER, Matthias; WELLER, Jochen; KUHN, Jochen; KASNECI, Gjergji. ChatGPT for good? On opportunities and challenges of large language models for education. In: Learning and Individual Differences. 2023. vol. 103. p. 102274. https://doi.org/10.1016/j.lindif.2023.102274
21. KHAN ACADEMY. Meet Khanmigo: Khan Academy's AI-powered teaching assistant & tutor. 2024. https://www.khanmigo.ai/
22. LEARNING EQUALITY. Kolibri: the offline app for universal education [software]. 2024. https://learningequality.org/kolibri/
23. LEWIS, Patrick; PEREZ, Ethan; PIKTUS, Aleksandra; PETRONI, Fabio; KARPUKHIN, Vladimir; GOYAL, Naman; KÜTTLER, Heinrich; LEWIS, Mike; YIH, Wen-tau; ROCKTÄSCHEL, Tim; RIEDEL, Sebastian; KIELA, Douwe. Retrieval-augmented generation for knowledge-intensive NLP tasks [preprint]. 2020. https://doi.org/10.48550/arXiv.2005.11401
24. MARTÍNEZ-TORO, Iván. PrivateGPT [software]. 2023. https://github.com/zylon-ai/private-gpt

25. NOGUEIRA, Rodrigo; CHO, Kyunghyun. Passage re-ranking with BERT [preprint]. 2019. https://doi.org/10.48550/arXiv.1901.04085

26. OECD. OECD digital education outlook 2023: towards an effective digital education ecosystem. Paris: OECD Publishing, 2023. 411 p. https://doi.org/10.1787/c74f03de-en

27. OPENAI. Introducing ChatGPT Edu. 2024. https://openai.com/index/introducing-chatgpt-edu/

28. OPENAI. API pricing. 2026. https://openai.com/api/pricing

29. PEFFERS, Ken; TUUNANEN, Tuure; ROTHENBERGER, Marcus A.; CHATTERJEE, Samir. A design science research methodology for information systems research. In: Journal of Management Information Systems. 2007. vol. 24, no. 3. p. 45-77. https://doi.org/10.2753/MIS0742-1222240302

30. PRINSLOO, Paul; SLADE, Sharon. An elephant in the learning analytics room: the obligation to act. In: Proceedings of the International Learning Analytics & Knowledge Conference (LAK 2017). 2017. p. 46-55. https://doi.org/10.1145/3027385.3027426

31. REIMERS, Nils; GUREVYCH, Iryna. Sentence-BERT: sentence embeddings using Siamese BERT-networks. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP 2019). 2019. p. 3982-3992. https://doi.org/10.18653/v1/D19-1410

32. ROBERTSON, Stephen; ZARAGOZA, Hugo. The probabilistic relevance framework: BM25 and beyond. In: Foundations and Trends in Information Retrieval. 2009. vol. 3, no. 4. p. 333-389. https://doi.org/10.1561/1500000019

33. SARTHI, Parth; ABDULLAH, Salman; TULI, Aditi; KHANNA, Shubh; GOLDIE, Anna; MANNING, Christopher D. RAPTOR: recursive abstractive processing for tree-organized retrieval [preprint]. 2024. https://doi.org/10.48550/arXiv.2401.18059

34. WOOLF, Beverly Park. Building intelligent interactive tutors: student-centered strategies for revolutionizing e-learning. Burlington: Morgan Kaufmann, 2009. 480 p. https://doi.org/10.1016/B978-0-12-373594-2.X0001-9

35. WORLD POSSIBLE. RACHEL: Remote Area Community Hotspot for Education and Learning. 2024. https://rachel.worldpossible.org/

36. YAN, Lixiang; SHA, Lele; ZHAO, Linxuan; LI, Yuheng; MARTINEZ-MALDONADO, Roberto; CHEN, Guanliang; LI, Xinyu; JIN, Yueqiao; GAŠEVIĆ, Dragan. Practical and ethical challenges of large language models in education: a systematic scoping review. In: British Journal of Educational Technology. 2024. vol. 55, no. 1. p. 90-112. https://doi.org/10.1111/bjet.13370

37. ZHANG, Peiyuan; ZENG, Guangtao; WANG, Tianduo; LU, Wei. TinyLlama: an open-source small language model [preprint]. 2024. https://doi.org/10.48550/arXiv.2401.02385