

Self-Contained Console as a Dual-Channel Tool with AI Feedback

Guillermo Sentoni¹ 

¹ Universidad Nacional de La Matanza, F. Varela 1903, Bs. As. Argentina
gsentoni@gmail.com

Abstract. This work presents the CPIAE console, a dual-channel pedagogical tool for teaching programming in higher education. The first channel is an interactive Python editor embedded in a self-contained HTML artifact that executes code non-blockingly via WebAssembly, keeping the interface responsive even during computationally expensive operations such as neural network training. The tool runs entirely in the browser with no installation or server required and is fully functional on mobile devices. The second channel is a non-anthropomorphic three-dimensional avatar that reads aloud the feedback generated by a large language model, with phoneme-synchronized mouth animation. The assistant module also operates as a virtual tutor: students ask questions in natural language and receive responses mediated by the avatar, which explains concepts, corrects code, and generates ready-to-apply snippets. The design is grounded in Cognitive Load Theory and the Cognitive Theory of Multimedia Learning. The tool was deployed in a neural computing course with fifteen students over twelve weeks, achieving approximately eighty percent autonomous error resolution without instructor intervention.

Keywords: cognitive load, dual-channel, pedagogical avatar, AI feedback, browser-based Python, virtual tutor, mobile device.

Consola Autocontenida como Herramienta de Doble Canal con Retroalimentación IA

Resumen. Este trabajo presenta la consola CPIAE, una herramienta pedagógica de doble canal para la enseñanza de programación en educación superior. El primer canal es un editor Python interactivo embebido en un artefacto HTML autocontenido que ejecuta código de forma no bloqueante mediante WebAssembly, manteniendo la interfaz reactiva incluso durante operaciones costosas como el entrenamiento de redes neuronales. La herramienta opera íntegramente en el navegador, sin instalación ni servidor, con plena compatibilidad en dispositivos móviles. El segundo canal es un avatar tridimensional no antropomórfico que narra en voz alta la retroalimentación generada por un modelo de lenguaje de gran escala, con animación de boca sincronizada por fonemas. El módulo de asistencia opera además como tutor virtual: el estudiante formula preguntas en lenguaje natural y recibe respuestas mediadas por el avatar. El diseño se fundamenta en la Teoría de Carga Cognitiva

y la Teoría Cognitiva del Aprendizaje Multimedia. La herramienta fue desplegada en un curso de computación neuronal con quince estudiantes durante doce semanas, logrando aproximadamente el ochenta por ciento de resolución autónoma de errores sin intervención docente.

Palabras clave: carga cognitiva, doble canal, avatar pedagógico, retroalimentación IA, Python en navegador, tutor virtual.

1 Introducción

La enseñanza de programación en educación superior enfrenta una tensión central: los estudiantes necesitan retroalimentación inmediata y situada cuando sus programas fallan, pero la disponibilidad docente es inevitablemente limitada. Las herramientas basadas en modelos de lenguaje de gran escala (LLM), como ChatGPT o Gemini, exigen que el estudiante reconstruya manualmente el contexto del error: copiar el código, pegar la traza y describir el problema. Ese proceso introduce fricción cognitiva precisamente en el momento de mayor sobrecarga. La Consola Python con Inteligencia Artificial Educativa (CPIAE) (Ver Figura 1) fue diseñada para eliminar esa fricción.

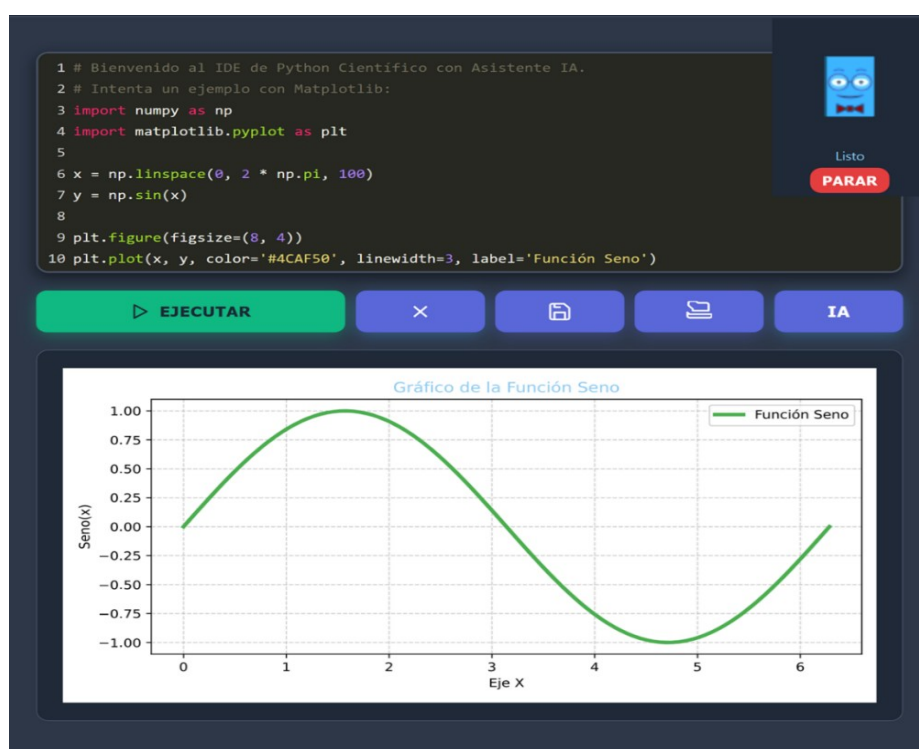


Fig. 1. La consola CPIAE en uso: editor Python con resaltado de sintaxis, salida y gráficos Matplotlib SVG, y avatar TalkingBook3D con panel de asistencia IA y tutor virtual conversacional.

Este artículo se focaliza en el modelo pedagógico de doble canal y en la evidencia empírica obtenida durante un curso completo de computación neuronal; la arquitectura técnica y el análisis de analíticas secuenciales se presentan en papers separados. Un rasgo distintivo es que opera íntegramente en el navegador —sin instalación ni servidor—, lo que la hace accesible desde cualquier dispositivo, incluyendo teléfonos móviles.

El concepto de doble canal se apoya en dos marcos: la Teoría Cognitiva del Aprendizaje Multimedia (CTML, Mayer, 2001) y la Teoría de Carga Cognitiva (CLT, Sweller et al., 2011). Ambos postulan que la información visual y auditiva se procesa en subsistemas cognitivos distintos. Cuando el diseño instruccional distribuye la retroalimentación entre ambos canales de manera coherente, la carga extrínseca disminuye y se liberan recursos cognitivos para el aprendizaje de los contenidos disciplinares.

2 CPIAE como Herramienta de Doble Canal

2.1 Canal visual–interactivo

El canal visual es un editor Python (CodeMirror 5) integrado en un artefacto HTML autocontenido que ejecuta código mediante Pyodide/WebAssembly directamente en el navegador. La contribución central es la ejecución no bloqueante: el intérprete corre en un Web Worker dedicado, manteniendo la interfaz reactiva incluso durante el entrenamiento de redes neuronales (1–8 s). En entornos síncronos previos estos cómputos activaban el diálogo "página sin respuesta", interrumpiendo el flujo cognitivo. La salida gráfica (Matplotlib SVG) y el historial del asistente coexisten en la misma ventana, eliminando cambios de contexto. Al correr como artefacto HTML autocontenido, la herramienta es plenamente funcional en dispositivos móviles.

2.2 Canal auditivo–narrativo: el "libro parlante"

El segundo canal es un avatar 3D estilizado (TalkingBook3D) —un libro con boca animada por fonemas— que narra las respuestas del LLM mediante la Web Speech API. El diseño no humanizado es deliberado: los avatares hiperrealistas introducen carga extrínseca por interpretación de expresiones faciales complejas (Drobnjak & Boticki, 2025). La narración entrega la explicación conceptual mientras el canal visual entrega el código sugerido —separación que la CTML predice como reductora de carga extrínseca. El avatar cumple además una función de acompañamiento durante el proceso de depuración: transforma la experiencia de fallo en una instancia dialogada de retroalimentación contextual, interactuando sobre la base del código, la traza de ejecución y el historial de la sesión.

2.3 Circuito de retroalimentación IA situado

Ante un error de ejecución, el módulo Asistente construye automáticamente un prompt que incluye el código activo del estudiante, la traza completa de ejecución y el historial de conversación. El prompt se envía a Gemini 2.5 Flash; la respuesta se narra por el avatar y se muestra como Markdown con código aplicable con un clic. El costo

cognitivo de pedir ayuda se reduce a un botón. Más allá de la resolución reactiva, el módulo opera como tutor virtual: el estudiante formula preguntas en lenguaje natural —"¿por qué mi red no converge?", "¿cómo agrego dropout?"— y recibe respuestas acumulativas a lo largo de la sesión, en un diálogo similar al de una tutoría presencial.

3 Fundamentos Pedagógicos: CLT y CTML

La CLT distingue carga intrínseca (complejidad del contenido), extrínseca (diseño instruccional) y germinal (construcción de esquemas). Un diseño óptimo minimiza la carga extrínseca para liberar recursos cognitivos (Sweller et al., 2011). CPIAE interviene por tres vías: la eliminación de fricción de contexto (el sistema captura y estructura el error automáticamente, sin acción del estudiante); la distribución dual (el código al canal visual, la explicación al canal auditivo); y la continuidad de flujo (la ejecución no bloqueante preserva el estado del entorno, evitando la ruptura cognitiva del reinicio de página).

Fink et al. (2024) documentan que los avatares de IA modifican el aprendizaje con beneficios dependientes de las condiciones de carga cognitiva. La CTML predice que el canal auditivo complementa —no reemplaza— el visual cuando cada canal entrega información diferente y coherente, que es exactamente el diseño de CPIAE.

4 Despliegue y Resultados Observacionales

CPIAE fue desplegada como entorno primario en un curso de computación neuronal de grado (Moodle, online, un encuentro semanal de 4 horas, 15 estudiantes, 3 semanas) que cubrió perceptrones y arquitectura MLP, backpropagation, regularización y clasificación con Scikit-learn. Todos los ejercicios se ejecutaron directamente en el navegador, distribuidos como artefactos HTML en Moodle, sin ninguna instalación local ni configuración de entorno. La herramienta resultó igualmente accesible desde computadoras de escritorio y dispositivos móviles, eliminando barreras de acceso habituales en cursos remotos.

Table 1. CPIAE características distintivas.

Característica	CPIAE	JupyterLite	Google Colab
Retroalimentación contextual automática	✓	✗	Manual
Canal auditivo: Avatar + TTS	✓	✗	✗
Ejecución no bloqueante	✓	✗	N/A
Sin servidor	✓	✓	✗
Integración Moodle	✓	Parcial	✓
Disponble en móvil	✓	Limitado	Limitado

Un análisis de los registros de interacción de la plataforma reveló que cerca del 80% de los errores de ejecución (registrados en el log) fueron solucionados dentro de la misma sesión de trabajo, sin necesidad de intervención docente posterior. Aunque se trata de

una observación descriptiva, el resultado es consistente con la hipótesis de que reducir la fricción asociada a la solicitud de ayuda constituye un mecanismo relevante del sistema (Yang et al., 2024). Se observó además que varios estudiantes emplearon el módulo de asistencia para formular preguntas conceptuales fuera de los momentos de error, sugiriendo su utilización como recurso de tutoría en lenguaje natural.

5 Discusión y Conclusiones

Un estudio paralelo de analíticas secuenciales, presentado en un paper separado, aplica FSM y HMM sobre los logs del mismo curso, identificando las interacciones con CPIAE como la dimensión con mayor poder discriminativo para predecir el rendimiento final. La interpretación conjunta sugiere que el doble canal habilita una práctica activa sostenida: al eliminar interrupciones por error irresuelto, el estudiante permanece más tiempo en práctica regulada, y el tutor virtual provee asistencia inmediata sin abandonar el entorno de programación.

Frente a los entornos más utilizados en educación en programación con asistencia IA, CPIAE presenta una combinación de características que ningún entorno alternativo ofrece en su totalidad (Tabla 1): ejecuta Python en el navegador de forma no bloqueante, incorpora canal auditivo con avatar TTS y genera retroalimentación contextual automática, todo ello sin instalación y con plena compatibilidad móvil.

Referencias

- Droettboom, M. (2018). *Scientific Python in the browser* [Blog post]. Droettboom. <https://droettboom.com/blog/2018/04/04/python-in-the-browser/>
- Drobnjak, A., & Boticki, I. (2025). Modelling virtual educational agents: Challenges of realism, performance and quality of audiovisual content. En R. A. Sottilare & J. Schwarz (Eds.), *Adaptive instructional systems. HCII 2025. Lecture Notes in Computer Science*, vol. 15813 (pp. 180–189). Springer. https://doi.org/10.1007/978-3-031-92970-0_13
- Fink, M. C., Robinson, S. A., & Ertl, B. (2024). AI-based avatars are changing the way we learn and teach. *Frontiers in Education*, 9, Article 1416307. <https://doi.org/10.3389/feduc.2024.1416307>
- Mayer, R. E. (2001). *Multimedia learning*. Cambridge University Press. <https://doi.org/10.1017/CBO9781139164603>
- Sweller, J., Ayres, P., & Kalyuga, S. (2011). *Cognitive load theory*. Springer. <https://doi.org/10.1007/978-1-4419-8126-4>
- Yang, B., Tian, H., Pian, W., Yu, H., Wang, H., Klein, J., Bissyandé, T. F., & Jin, S. (2024). CREF: An LLM-based conversational software repair framework for programming tutors. In *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA 2024)* (pp. 882–894). ACM. <https://doi.org/10.1145/3650212.3680328>