

Vibecoding: concepto, aplicabilidad educativa y herramientas prácticas

2026-02-17

Vibecoding: concepto, aplicabilidad educativa y herramientas prácticas

El vibecoding representa una transformación paradigmática en la creación de recursos digitales educativos, permitiendo que docentes sin experiencia en programación desarrollen aplicaciones funcionales mediante instrucciones en lenguaje natural.

El vibecoding representa una transformación paradigmática en la creación de recursos digitales educativos, permitiendo que docentes sin experiencia en programación desarrollen aplicaciones funcionales mediante instrucciones en lenguaje natural.

VIBECODING: CONCEPTO, APLICABILIDAD, EDUCATIVA Y HERRAMIENTAS PRÁCTICAS

Introducción

El [vibecoding](#) representa una transformación paradigmática en la creación de recursos digitales educativos, permitiendo que docentes sin experiencia en programación desarrollen aplicaciones funcionales mediante instrucciones en lenguaje natural. Este

enfoque emerge del avance significativo en modelos de lenguaje grandes (LLMs) que pueden interpretar descripciones conceptuales y convertirlas en código ejecutable. Lejos de ser un fenómeno temporal, el vibecoding se consolida como metodología educativa legitimada por instituciones académicas y comunidades docentes activas que superan los 1.500 miembros en espacios colaborativos hispanoablantes (Agora Abierta, 2025; Educación Bilateria, 2025a).

El término fue acuñado formalmente en febrero de 2025 por Andrej Karpathy, cofundador de OpenAI, quien lo describió como una forma de "entregarse por completo a las vibraciones" del software, confiando en sistemas de IA para los detalles técnicos mientras el desarrollador mantiene el enfoque en intención y funcionalidad (Cloudflare Learning, 2024).



¿Qué es Vibecoding desde la perspectiva educativa?

Vibecoding no es simplemente "programar sin programar". Es una metodología de bajo umbral técnico que prioriza la intención pedagógica sobre el dominio exhaustivo del código (Educación Bilateria, 2025a; Edukaitech, 2025). Para el docente, significa

reducir significativamente la barrera de entrada hacia la creación de herramientas digitales personalizadas, adaptadas específicamente al contexto, nivel y necesidades de sus estudiantes.

La esencia radica en un proceso iterativo conversacional (Educación Bilateral, 2025a): el educador define qué desea conseguir (una simulación interactiva, un evaluador, un juego), dialoga con un asistente de IA, observa resultados, ajusta mediante nuevas instrucciones, y repite hasta obtener el recurso deseado. El código deviene un medio transparente, no un objetivo en sí mismo (Agora Abierta, 2025).

Diferenciación conceptual

Vibecoding difiere fundamentalmente de las plataformas tradicionales de desarrollo sin código (no-code) porque no requiere navegar interfaces visuales complejas. En su lugar, aprovecha la capacidad de los LLMs para interpretar matices comunicativos incluyendo emociones y contexto (Universidad Europea, 2025). Un docente puede solicitar "una actividad relajada y exploratoria" versus "una evaluación formal y rigurosa", y el sistema ajusta tonalidad, colores, retroalimentación y estructura (Universidad Europea, 2025).

Aplicabilidad educativa: Evidencia de implementación

Tipología de recursos educativos creables

La evidencia práctica de aula documenta la creación exitosa de múltiples tipologías de recursos (INTEF, 2024; Tecnológico de Monterrey, 2025):

- **Ciencias naturales:** Simuladores de leyes de Mendel, visualizadores interactivos de movimiento, modelos de propagación de enfermedades con control de variables en tiempo real.
- **Matemáticas:** Representadores de funciones con controles deslizantes que visualizan cambios instantáneos en gráficas, calculadoras especializadas, convertidores de unidades.

- **Lengua y literatura:** Analizadores estructurales de párrafos argumentativos con retroalimentación codificada por colores, juegos de clasificación léxica, ejercicios de sinónimos y antónimos autocorregibles.
- **Ciencias sociales:** Líneas de tiempo editables, comparadores de ideologías con tablas desplegadas, simuladores de economía (oferta, demanda, puntos de equilibrio).
- **Tecnología:** Editores HTML/CSS con vista previa en directo, simuladores de circuitos eléctricos.
- La característica unificadora es la capacidad de ofrecer interactividad, respuesta inmediata y adaptabilidad sin configuración compleja (Educación Bilateral, 2025a; Agora Abierta, 2025).

Ventajas pedagógicas sustantivas

La literatura educativa contemporánea identifica beneficios que trascienden la mera simplificación técnica (Docentes Digitales MX, 2025; Aguirre Hermida, 2025):

- **Personalización hiperlocal:** Los recursos responden a contextos específicos del aula. Un docente puede crear un simulador de fotosíntesis en español, con ejemplos de flora colombiana, y glosario local, algo que ninguna aplicación genérica ofrece (Agora Abierta, 2025).
- **Transformación del rol docente:** De consumidor pasivo de herramientas a creador activo de experiencias de aprendizaje. Esta transición empodera al docente como diseñador instruccional digital (Docentes Digitales MX, 2025).
- **Retroalimentación automática y personalizada:** Los recursos pueden integrar modelos de IA generativa para proporcionar feedback específico y adaptado al nivel del estudiante. Por ejemplo, una aplicación que evalúa párrafos

puede ofrecer sugerencias de mejora formuladas en lenguaje comprensible para el nivel académico (Tecnológico de Monterrey, 2025).

- **Exploración versus pasividad:** Las simulaciones interactivas transforman estudiantes de espectadores a exploradores activos, formulando hipótesis y observando consecuencias en tiempo real (Tecnológico de Monterrey, 2025).

Plataformas y herramientas: Matriz de decisión

La elección de herramienta depende del objetivo pedagógico, disponibilidad de recursos técnicos y velocidad requerida. La matriz de decisión operativa se presenta a continuación:

Herramientas vía Web para prototipado rápido

Estas ejecutan directamente en navegador sin instalación, ideales para educadores iniciales o contextos con restricciones de software (Educación Bilateria, 2025a).

Plataforma	Características	Ideal Para	Limitaciones
ChatGPT (Canvas)	Generación rápida de código, interfaz limpia con vista previa	Prototipos sencillos, actividades puntuales, docentes principiantes	Diseño básico, requiere refinamiento visual
Gemini (Canvas)	Equilibrio entre diseño y funcionalidad, integración de IA nativa, acceso a documentación reciente	Recursos educativos con visualización mejorada, integraciones con Google Workspace	Menos control fino que editores especializados
Claude (Artefactos)	Código bien estructurado, limpio, legible con comentarios	Educadores que requieren código mantenible y comprensible	Contexto limitado en versiones gratuitas
Qwen Web Dev	Alternativa económica, buena relación calidad-costeo	Instituciones con presupuesto limitado, docentes en entornos de bajos recursos	Menos documentación en español

Editores gráficos con IA integrada (IDE)

Proporcionan entorno de desarrollo completo con asistencia de IA para proyectos más ambiciosos (Educación Bilateria, 2025a).

- **Cursor** emerge como herramienta paradigmática para vibecoding educativo. Es un fork de Visual Studio Code enriquecido con IA (GPT-4 o Claude 3.5 Sonnet integrados) (Learn Agents DIY, s.f.; Tokio School, 2025). Su flujo operativo simplificado mediante atajos y capacidad de generar proyectos completos desde descripciones en lenguaje natural la posiciona como ideal para educadores que requieren control técnico sin complejidad abrumadora (Tokio School, 2025).
- **Visual Studio Code + Gemini Code Assist** ofrece alternativa integrada en el ecosistema Google, útil para instituciones ya inmersas en Google Workspace.
- **Google Antigravity** se destaca y se posiciona indiscutiblemente como la herramienta más potente, versátil y, significativamente, completamente gratuita para optimizar el proceso completo de programación y desarrollo de software. Su fortaleza reside en un ecosistema avanzado que permite la integración y el uso orquestado de **agentes expertos de inteligencia artificial**. Estos agentes no solo ofrecen asistencia, sino que se convierten en colaboradores activos e inteligentes a lo largo de todas las fases críticas del ciclo de vida del desarrollo de software (SDLC).

Plataformas No-Code Full-Stack

Para educadores que requieren aplicaciones con backend, autenticación y bases de datos (Logto Blog, 2025; Hostinger, 2025):

Plataforma	Fortalezas	Debilidades	Caso Educativo
Lovable	Full-stack (React), propiedad del código, exportable a GitHub, prototipado ultra-rápido	Limitado a React, requiere framework moderno	Plataformas educativas con registro, seguimiento de progreso, certificados automáticos
Bolt.new	Múltiples frameworks (React, Vue, Next, Astro), detección de errores activa, terminal integrada	Menos flexible en customización profunda	Aplicaciones educativas medianas con requisitos técnicos variados
Softtr	100+ plantillas, acceso rápido a plantillas comunes, enfoque visual	Requiere fuente de datos propia, limitado a web	Portales educativos, directorios de recursos, dashboards de gestión
Replit	Enfoque educativo explícito, explica código mientras lo genera, ejecución inmediata	Precios basados en uso pueden escalar, UI abrumadora para principiantes	Ideal para proyectos donde aprender el proceso es parte del objetivo

Framer AI y Canva Code ofrecen especialización en prototipado visual y diseño, particularmente valiosos para recursos con énfasis gráfico (Docentes Digitales MX, 2025).

Modelos de lenguaje: Análisis comparativo para educación

La selección del modelo de IA correcta impacta significativamente la calidad de los recursos generados. A enero de 2026, el panorama es el siguiente (Aprender21, 2025; FindSkill AI, 2026):

Claude 4.5 Sonnet (Anthropic)

Fortalezas para educación: Código más limpio, mejor estructura lógica, contexto de 200.000 tokens permite procesar documentos educativos completos (Aprender21, 2025). Particularmente superior para código educativo que requiera mantenibilidad a largo plazo y que pueda necesitar iteraciones futuras del docente (FindSkill AI, 2026).

Debilidades: Límites de uso en plan gratuito más restrictivos que ChatGPT.

Recomendación: Seleccionar para proyectos educativos que el docente modificará posteriormente o que requieran código particularmente limpio.

ChatGPT / GPT-4 / GPT-5.2 (OpenAI)

Fortalezas: Velocidad, versatilidad, capacidad de razonamiento científico y matemático superior, integración con múltiples herramientas educativas (Aprender21, 2025). Acceso a información reciente sin configuración especial (FindSkill AI, 2026).

Debilidades: Diseño visual básico en generación de código, requiere refinamiento de prompts para resultados óptimos.

Recomendación: Primera opción para docentes iniciando, especialmente en contextos de matemáticas y ciencias donde razonamiento es crítico.

Gemini 3 Pro (Google)

Fortalezas: Capacidades multimodales (video, imágenes, audio), integración nativa con Google Workspace (Gmail, Docs, Sheets, Calendar), contexto de 2 millones de tokens, acceso a documentación técnica reciente (FindSkill AI, 2026; Aprender21, 2025). Particularmente robusto para educación visual y análisis de datos educativos (Impulso06, 2025).

Debilidades: A veces introduce complejidad innecesaria, menos especialización en código educativo que Claude.

Recomendación: Ideal para educadores ya inmersos en ecosistema Google y que necesitan integración con herramientas colaborativas.

Qwen (Alibaba)

Fortalezas: Alternativa económica, buena generación de código, creciente soporte para educación.

Debilidades: Menor documentación en español, comunidad educativa menos establecida.

Recomendación: Opción viable para contextos de presupuesto limitado.

Metodología de implementación: Proceso práctico

La literatura de vibecoding educativo identifica un proceso estructurado que optimiza resultados (OpenWebinars, 2025; Edukaitech, 2025; Tecnológico de Monterrey, 2025):

Fase 1: Definición de la intención pedagógica

Antes de abrir cualquier herramienta, el docente debe clarificar por escrito:

- **Función central:** ¿Qué hará exactamente la aplicación? (Ejemplo: "Evaluar la comprensión de la ley de conservación de energía mediante simulación interactiva")
- **Público y contexto:** ¿Quiénes son los usuarios? ¿Qué nivel? ¿Cuál es el tema específico? (Ejemplo: "Estudiantes de física de grado 10, 15-17 años, tema: energía cinética y potencial")
- **Resultado deseado:** ¿Qué debería aprender o experimentar el estudiante? (Ejemplo: "Visualizar en tiempo real cómo cambia la energía potencial al mover un objeto, y recibir retroalimentación sobre cálculos")

Fase 2: Formulación del prompt educativo

Un buen prompt educativo sigue una estructura clara y pedagógicamente orientada (Edukaitech, 2025; Docentes Digitales MX, 2025):

Rol: "Eres un experto en educación en física para secundaria"

Tarea principal: "Crea una simulación interactiva para enseñar energía potencial gravitatoria"

Contexto: "Los estudiantes tienen 15 años, conocen cálculo básico, necesitan entender cómo $PE = mgh$ "

Requisitos funcionales:

- Un slider para controlar altura (0-100 metros)
- Un dropdown para seleccionar masa (1-100 kg)
- Cálculo automático y visualización de PE en tiempo real
- Gráfico que muestre cómo cambia PE con la altura
- Un botón para "soltar" el objeto y ver conversión a energía cinética

Audiencia: Estudiantes de 15 años en clase de física

Resultado: Que comprendan la relación matemática entre altura, masa y energía potencial mediante exploración interactiva

Tono: Pedagógico, claro, motivador

Colores: Azul y verde (tema científico)

Fase 3: Generación y prototipado

El docente introduce el prompt en la herramienta seleccionada (ChatGPT Canvas, Gemini, Claude Artifacts, o Cursor) (Tecnológico de Monterrey, 2025). La IA genera un prototipo funcional en cuestión de minutos (OpenWebinars, 2025).

Fase 4: Iteración cuidadosa

Aquí sucede la maestría del vibecoding educativo. El docente prueba el prototipo, identifica gaps pedagógicos, y solicita refinamientos específicos. Ejemplos de iteraciones típicas (Tecnológico de Monterrey, 2025):

- "Añade un contador que muestre la fórmula $PE = mgh$ con los valores actuales"
- "Cambia el gráfico para que muestre simultáneamente PE y energía cinética"

- "Añade un check: el estudiante debe predecir el valor antes de que la IA lo calcule"

Nota crítica sobre iteraciones: La evidencia indica que cada ronda de correcciones sobre código generado puede introducir vulnerabilidades (Kaspersky, 2025). Después de cinco iteraciones, el código contiene aproximadamente 37% más vulnerabilidades críticas que la versión inicial (Kaspersky, 2025). Esto sugiere que mantener el diseño inicial simple y validado es preferible a iteraciones excesivas.

Fase 5: Validación pedagógica

Antes de desplegar con estudiantes, el docente prueba con un subgrupo (3-5 personas) para evaluar si cumple objetivos de aprendizaje. Sesiones de 10-15 minutos proporcionan feedback cualitativo que guía ajustes finales (OpenWebinars, 2025).

Fase 6: Integración en plataformas educativas

Los recursos generados se integran en ecosistemas educativos existentes (Tecnológico de Monterrey, 2025):

- **Moodle:** Mediante incrustación de código HTML/JavaScript
- **Google Classroom:** Enlace compartible o incrustación en tareas
- **Descarga local:** Simuladores descargables para uso sin conexión (particularmente valioso en contextos con conectividad limitada)
- **Google Sites:** Inserción directa en portafolios educativos

Consideraciones críticas: Limitaciones y riesgos

Limitaciones técnicas y pedagógicas

La investigación contemporánea identifica limitaciones sustanciales que todo educador debe considerar (DataCamp, 2025; Geeknetic, 2025; Kaspersky, 2025):

- **Calidad y fragilidad del código:** El código generado por IA típicamente "funciona" para flujos principales, pero puede contener ineficiencias ocultas, lógica frágil, o código que falla bajo condiciones inesperadas (DataCamp, 2025). Un simulador que funciona correctamente con 100 partículas puede colapsar con 10.000.
- **Escalabilidad limitada:** Aplicaciones pequeñas prototipadas rápidamente frecuentemente no escalan hacia proyectos más grandes. La arquitectura inicial puede no soportar funcionalidades adicionales sin reescrituras sustanciales (Geeknetic, 2025).
- **Vulnerabilidades de seguridad:** El código generado por IA frecuentemente adolece de deficiencias críticas de seguridad incluyendo validación inadecuada de datos, inyección de SQL, scripting entre sitios (XSS), y claves API codificadas en el front-end (Kaspersky, 2025). Aunque educativamente menos crítico que en aplicaciones comerciales, estos riesgos merecen consideración si datos estudiantiles están involucrados.
- **Degradación con iteraciones:** Cada corrección solicitada puede introducir nuevas vulnerabilidades. El protocolo recomendado es mantener el diseño inicial simple y aprobado, evitando iteraciones extensas (Kaspersky, 2025).

Limitaciones pedagógicas emergentes

- **Exceso de confianza sin comprensión subyacente:** Los educadores pueden generar herramientas que funcionan sin entender completamente el mecanismo subyacente, creando riesgo de propagar concepciones erróneas (DataCamp, 2025).

- **No reemplaza programación educativa:** Vibecoding automatiza codificación, pero no elimina la necesidad de que estudiantes avanzados comprendan fundamentos de programación, estructura de datos, o lógica computacional (Campus MVP, 2025).
- **Depende fuertemente de calidad de prompts:** Un prompt mal formulado genera recursos educativamente débiles. La habilidad de "prompting pedagógico" deviene crítica y requiere entrenamiento (Docentes Digitales MX, 2025).

Conclusión

El vibecoding representa genuinamente una transformación en accesibilidad para creación de recursos educativos digitales. Eliminando barrera técnica de programación, empodera a docentes para actuar como diseñadores instruccionales digitales, creando experiencias hiper-personalizadas imposibles de lograr con herramientas genéricas comerciales.

Sin embargo, no es solución mágica. Requiere claridad pedagógica, validación experimental, conciencia de limitaciones técnicas, y aprendizaje continuo. Educadores que adopten vibecoding con intencionalidad pedagógica clara, combinado con validación empírica, descubrirán capacidad revolucionaria para ampliar y mejorar sus prácticas docentes.

Ahora es el momento ideal para experimentar. Las herramientas son fácilmente accesibles, la comunidad educativa está en constante crecimiento y es muy activa, y la tecnología que la sustenta sigue evolucionando positivamente. Para un educador comprometido con la innovación, *vibecoding* no es solo una herramienta disponible, sino una exploración profesional esencial y sistemática.

Referencias

- Agora Abierta. (2025, 2 de noviembre). *Del vibe coding a las guías interactivas: diseñar para pensar con la ayuda de la IA*. Recuperado de <https://www.agorabierta.com/2025/11/del-vibe-coding-a-las-guias-interactivas-disenar-para-pensar-con-la-ayuda-de-la-ia/>
- Aprender21. (2025, 31 de diciembre). *ChatGPT vs Claude vs Gemini: Comparativa Completa 2026*. Recuperado de <https://www.aprender21.com/blog/chatgpt-vs-claude-vs-gemini-comparativa>
- Campus MVP. (2025, 18 de junio). *El espejismo del Vibe Coding: NO hay atajos para aprender a programar*. Recuperado de <https://www.campusmvp.es/recursos/post/el-espejismo-del-vibe-coding-no-hay-atajos-para-aprender-a-programar.aspx>
- Cloudflare Learning. (2024, 31 de diciembre). *¿Qué es el vibe coding? | Codificación de IA*. Recuperado de <https://www.cloudflare.com/es-es/learning/ai/ai-vibe-coding/>
- DataCamp. (2025, 27 de abril). *¿Qué es la codificación Vibe? Definición, herramientas y más*. Recuperado de <https://www.datacamp.com/es/blog/vibe-coding>
- Docentes Digitales MX. (2025, 28 de julio). *Vibe Coding: Crear experiencias educativas sin saber programar*. Recuperado de <https://docentesdigitales.mx/vibe-coding-crear-experiencias-educativas-sin-saber-programar/>
- Educación Bilateria. (2025a, 13 de diciembre). *La elección de herramientas para el Vibe Coding Educativo*. Recuperado de <https://educacion.bilateria.org/la-eleccion-de-herramientas-para-el-vibe-coding-educativo>

- Educación Bilateria. (2025b, 11 de enero). *¿Qué IA me conviene? Comparativa de chatbots gratuitos*. Recuperado de <https://educacion.bilateria.org/que-ia-me-conviene-comparativa-de-chatbots>
- Edukaitech. (2025, 8 de noviembre). *Vibe coding educativo*. Recuperado de <https://www.edukaitech.com/blog/inteligencia-artificial-2/vibe-coding-educativo-30>
- FindSkill AI. (2026, 11 de enero). *ChatGPT vs Claude vs Gemini: ¿Cuál es la mejor IA para tus necesidades?*. Recuperado de <https://findskill.ai/es/blog/chatgpt-vs-claude-vs-gemini/>
- Geeknetic. (2025, 26 de mayo). *Vibe-Coding: Ventajas y Desventajas de Programar Aplicaciones con la ayuda de la IA*. Recuperado de <https://www.geeknetic.es/Guia/3195/Vibe-Coding-Ventajas-y-Desventajas-de-Programar-Aplicaciones-con-la-ayuda-de-la-IA.html>
- Hostinger. (2025, 14 de diciembre). *Las 10 mejores herramientas de vibe coding*. Recuperado de <https://www.hostinger.com/co/tutoriales/las-mejores-herramientas-de-vibe-coding>
- Impulso06. (2025, 2 de febrero). *ChatGPT, Gemini, Claude, Perplexity y Copilot vs DeepSeek: ¿Quién se llevará el pastel?*. Recuperado de <https://impulso06.com/chatgpt-gemini-claude-perplexity-y-copilot-vs-deepseek-quien-se-llevara-el-pastel/>
- Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado. (2024, 31 de diciembre). *Vibe coding | Inteligencia Artificial en Educación*. Recuperado de https://descargas.intef.es/cedec/proyectoedia/guias/contenidos/inteligencia_artificial/vibe_coding

- Kaspersky. (2025, 27 de octubre). *Los peligros ocultos del "vibe coding"*. Recuperado de <https://www.kaspersky.es/blog/vibe-coding-2025-risks/31557/>
- Learn Agents DIY. (s.f.). *Vibe-Coding y Cursor AI*. Recuperado de <https://www.learn-agents.diy/es/not-agents/vibecoding/>
- Logto Blog. (2025, 9 de julio). *Codifica Vibe usando Lovable AI y Logto para construir rápidamente aplicaciones web*. Recuperado de <https://blog.logto.io/es/lovable-logto-auth>
- OpenWebinars. (2025, 28 de agosto). *Vibe Coding: Cómo prototipar aplicaciones con IA sin programar*. Recuperado de <https://openwebinars.net/blog/vibe-coding-como-prototipar-aplicaciones-con-ia-sin-programar/>
- Profesor Productivo. (2024, 22 de septiembre). *Podcast: Comparación de modelos de IA para docentes*. Recuperado de <https://profesorproductivo.com/blog/podcast-comparacion-de-modelos-de-ia-para-docentes/>
- Teknos School. (2025, 27 de enero). *Vibe coding: ¿el futuro de la programación o una moda pasajera?*. Recuperado de <https://www.tokioschool.com/noticias/vibe-coding/>
- Tecnológico de Monterrey. (2025, 4 de noviembre). *Vibe coding para docentes: Crea apps educativas con ayuda de IA*. Recuperado de <https://observatorio.tec.mx/edu-bits-blog/vibe-coding-para-docentes-crea-apps-educativas-con-ayuda-de-ia/>
- Universidad Europea. (2025, 1 de diciembre). *Qué es el vibe coding y qué ventajas presenta*. Recuperado de <https://universidadeuropea.com/blog/vibe-coding/>

CRÉDITOS:

Artículo elaborado por [Boris Sánchez Molano](#), Coordinador de Innovación Pedagógica y TIC del Centro Eduteka, Facultad de Ciencias Humanas, Universidad Icesi, Cali, Colombia.

Publicación de este documento en EDUTEKA: Febrero 17 de 2026.

Última actualización de este documento: Febrero 17 de 2026.