Herramienta de traducción de lenguaje natural a lenguaje Python

Natural Language to Python Programming Language Translation Tool

Ofelia Gutiérrez Giraldi[*1], Lizbeth Sanchez Ferrer[1], Martha Martínez Moreno[2], Jorge Ángel Estudillo Silva[1], Efrén Ernesto Pérez Nicanor[1], Esteban Santos Angulo[1] y David León Salas[1]

[1]Tecnológico Nacional de México. Instituto Tecnológico de Veracruz. Ingeniería en Sistemas Computacionales Calz. Miguel Angel de Quevedo 2779, Col. Formando Hogar, 91897, Veracruz, Ver. México
[2]Tecnológico Nacional de México. Instituto Tecnológico de Toluca, Ingeniería en Sistemas Computacionales Av Tecnológico S/N, Col. Agrícola Bellavista, 52149, San Salvador Tizatlalli, Estado de México, México
[*]ofelia.gg@veracruz.tecnm.mx

PALABRAS CLAVE:

Programación, traducción, asistente inteligente

RESUMEN:

Desde una perspectiva académica y tecnológica, esta investigación se centra en el desarrollo de una herramienta educativa basada en inteligencia artificial generativa cuyo propósito es apoyar el aprendizaje de la programación de manera didáctica y transparente. Surge como respuesta a las limitaciones pedagógicas de los asistentes de IA (Inteligencia Artificial) actuales, los cuales suelen proporcionar soluciones completas sin favorecer el razonamiento lógico ni la comprensión profunda del código. La propuesta consiste en una aplicación web Full Stack que integra un modelo de lenguaje especializado (CodeLlama), ejecutado en una infraestructura autoalojada, capaz de traducir instrucciones escritas en lenguaje natural en español a código funcional en el lenguaje Python. A través de una interfaz de chat interactiva con transmisión de resultados en tiempo real, la herramienta permite observar de forma progresiva la construcción del código, simulando el acompañamiento de un tutor humano. El sistema está orientado a estudiantes de programación de nivel principiante e intermedio, priorizando el fortalecimiento de la lógica algorítmica, la comprensión sintáctica y el uso crítico de la IA como apoyo educativo, sin sustituir los procesos formales de enseñanza.

KEYWORDS:

Programming, translation, intelligent assistants

ABSTRACT:

This research presents the analysis, design, and implementation of an educational web-based tool aimed at supporting programming learning through a pedagogically guided use of generative artificial intelligence. Responds to the growing gap between traditional algorithm-teaching tools and modern AI assistants, which often provide immediate solutions without fostering cognitive engagement. The proposed system integrates a self-hosted generative language model (CodeLlama) capable of translating natural language instructions written in Spanish into functional Python code. Built upon a decoupled Full Stack architecture, the application features an interactive chat interface that streams code generation in real time, allowing learners to observe the logical and syntactic construction process step by step. This approach transforms the AI from a mere problem solver into an instructional tutor. The tool is primarily intended for beginner and intermediate programming students, emphasizing transparency, algorithmic reasoning, and active learning, while serving as a complementary educational resource rather than a replacement for formal instruction.

• Recibido: 20 de enero de 2026 • Aceptado: 12 de mayo de 2026 • Publicado en línea: 2 de octubre de 2026

Copyright:© 2026 por los autores
Creative Commons Atribución 4.0 CC-BY

1. INTRODUCCIÓN

En los últimos años, la enseñanza de la programación ha experimentado transformaciones significativas impulsadas por el avance acelerado de la inteligencia artificial y, en particular, de los modelos de lenguaje generativo. Estas tecnologías han modificado la manera en que los estudiantes interactúan con el conocimiento, permitiendo nuevas formas de apoyo automatizado en el proceso de aprendizaje. Sin embargo, su incorporación en el ámbito educativo ha generado tanto oportunidades como desafíos, especialmente en lo referente al desarrollo de la lógica algorítmica y la comprensión profunda del código fuente. En este contexto, surge la necesidad de diseñar herramientas que aprovechen el potencial de la inteligencia artificial sin sustituir el razonamiento crítico ni fomentar la dependencia tecnológica.

2. FUNDAMENTOS TEÓRICOS

La investigación sobre la inteligencia artificial (IA) no es un fenómeno reciente. Sus fundamentos teóricos se remontan a mediados del siglo XX, cuando los pioneros de la computación comenzaron a explorar la posibilidad de que las máquinas pudieran simular el pensamiento humano [3]. Sin embargo, el panorama actual ha sido drásticamente redefinido por los avances en el Aprendizaje Automático (Machine Learning) y, más recientemente, por la irrupción de la IA generativa.

Un informe de la UNESCO señala que, aunque la IA ha formado parte del discurso educativo durante décadas, ha sido en los últimos años, y particularmente con la popularización de modelos de lenguaje extensos en 2023 y 2024, que su implementación práctica en las aulas ha pasado de ser una teoría a una realidad tangible que demanda análisis y regulación [11].

2.1. Definición y Tipos de Inteligencia Artificial

En el contexto educativo, la inteligencia artificial "se refiere a sistemas informáticos que realizan tareas que requieren inteligencia humana, como aprender, resolver problemas y comprender el lenguaje" [4]. No se trata de una sola tecnología, sino de un conjunto de herramientas que se pueden clasificar según su aplicación en el aprendizaje.

Un estudio de revisión identifica cuatro aplicaciones clave de la IA en la educación: "el perfilado y la predicción, los Sistemas de Tutoría Inteligentes (ITS), la automatización, y los robots y chatbots educativos" [14].

De manera más específica para las aplicaciones modernas, Fullmind las categoriza en tres tipos principales [4]:

- Sistemas de Aprendizaje Adaptativo: Ajustan la dificultad y el tipo de contenido en tiempo real según el rendimiento del estudiante.
- IA Generativa: Herramientas (como ChatGPT o Copilot) que pueden crear contenido nuevo, como borradores de planes de lecciones, preguntas de práctica personalizadas o la generación de código.
- Análisis Predictivo: Utiliza la IA para analizar patrones de datos de los estudiantes y pronosticar resultados, permitiendo identificar a los alumnos en riesgo de reprobación.

Para complementar la categorización de Fullmind, es útil analizar la IA desde una perspectiva funcional en el aula. López y Mendoza proponen una distinción clave entre "IA de apoyo" (Assistive AI) e "IA generativa" (Generative AI) [16]. La primera actúa como un andamio cognitivo (ej. correctores gramaticales avanzados o asistentes de búsqueda), mientras que la segunda es capaz de producir contenido original, lo que implica una reevaluación de la autoría y la originalidad en las tareas académicas.

Además, la categoría de "Sistemas de Tutoría Inteligentes (ITS)", mencionada por Zaïane, merece una profundización. Un ITS no es un simple chatbot; su arquitectura se basa en modelar el conocimiento del estudiante. Un estudio reciente sobre el desarrollo de ITS en 2025 explica que estos sistemas mantienen un "modelo del estudiante" que rastrea las concepciones correctas y erróneas. "El sistema compara el modelo del estudiante con el modelo de un experto en la materia y genera dinámicamente el siguiente problema o explicación óptima para cerrar esa brecha de conocimiento". Esta capacidad de diagnóstico y remediación en tiempo real es lo que los distingue de las plataformas de aprendizaje adaptativo más simples [10].

2.2. Aplicaciones de la IA en la Educación

La integración de la IA está transformando las instituciones educativas a múltiples niveles. Un reporte especial de Microsoft indica que el 86% de las organizaciones educativas ya reportan el uso de IA generativa, la tasa más alta de cualquier industria [9]. Los líderes educativos la utilizan para tareas administrativas y eficiencias operativas, mientras que los docentes la emplean para crear materiales de clase y responder preguntas comunes.

Para los estudiantes, la IA ofrece herramientas de apoyo disponibles 24/7, como tutores virtuales y chatbots, y facilita entornos de aprendizaje interactivos y personalizados.

Más allá de las aplicaciones administrativas y el apoyo general [12], las aplicaciones pedagógicas específicas están ganando terreno. Un área de gran impacto es la evaluación formativa escalable. Tradicionalmente, la retroalimentación detallada consume una enorme cantidad de tiempo docente. Los "Sistemas de Puntuación Inteligente" (Intelligent Scoring Systems) utilizan IA para analizar respuestas abiertas, ensayos cortos e incluso código de programación, proporcionando retroalimentación formativa instantánea [10].

Otra aplicación avanzada es la creación de Rutas de Aprendizaje Personalizadas (PLR). Un informe del Observatorio de Innovación Educativa describe estos sistemas como la siguiente evolución del aprendizaje adaptativo [10]. En lugar de simplemente ajustar la dificultad, estos sistemas:

Analizan el historial completo de un estudiante, sus objetivos de carrera declarados y sus estilos de aprendizaje inferidos para trazar un mapa de competencias. El motor de IA recomienda una secuencia de módulos, lecturas, simulaciones y proyectos de múltiples fuentes, creando un plan de estudios verdaderamente individualizado.

Esto permite que dos estudiantes en la misma clase tomen caminos muy diferentes para alcanzar el mismo objetivo de aprendizaje, optimizando el compromiso y la eficiencia.

2.3. El Papel de la IA en la Enseñanza de la Programación

El impacto de la IA en el aprendizaje específico de la programación es uno de los campos más dinámicos. Un estudio sobre estudiantes universitarios concluyó que "si bien la IA puede transformar la enseñanza de la programación informática, su implementación debe ir acompañada de estrategias pedagógicas... que fomenten la autonomía y el aprendizaje significativo" [13].

La llegada de la IA generativa ha provocado un debate sobre la pertinencia de seguir enseñando programación de manera tradicional. Barrios del Valle analiza esta cuestión, argumentando que estas herramientas no eliminan la necesidad de aprender a programar, sino que reorientan la habilidad hacia el "desarrollo de soluciones" y la supervisión del código generado, en lugar de la simple escritura de sintaxis [1].

Además, la IA se está utilizando para fomentar habilidades clave en los futuros programadores. León Morejón proponen el uso de herramientas de IA para "dinamizar el proceso de autorregulación del aprendizaje de la programación", ayudando a los estudiantes a gestionar su propio proceso de aprendizaje [6].

Ampliando la idea de "autorregulación", una de las aplicaciones más transformadoras de la IA en la programación es la Retroalimentación Automatizada. Históricamente, los AFS (Automated Feedback Systems) en programación se limitaban a verificar si la salida de un programa era correcta o incorrecta (pruebas unitarias). Sin embargo, los AFS modernos impulsados por IA van mucho más allá. Un estudio de Chen sobre AFS para programadores novatos encontró que las herramientas de IA actuales no solo identifican errores de sintaxis, sino que pueden diagnosticar errores lógicos y problemas de eficiencia. Por ejemplo, el sistema puede detectar que el código de un estudiante, aunque funcional, utiliza un algoritmo $O(n^2)$ (como un bucle anidado) cuando existe una solución $O(n)$ (más eficiente). El sistema de retroalimentación de IA no solo marcó el error lógico, sino que proporcionó una pista en lenguaje natural que guiaba al estudiante a reconsiderar su caso base en la función recursiva, un tipo de andamiaje que imita la intervención de un tutor humano experto. Este nivel de retroalimentación instantánea y conceptual es crucial en la enseñanza de la programación, ya que permite a los estudiantes depurar su pensamiento lógico, no solo su sintaxis, en el momento preciso en que surge el problema.

2.4. El Lenguaje Natural y el Procesamiento de Lenguaje Natural (PLN)

Para comprender cómo una herramienta puede traducir la intención humana en código formal, primero se debe definir el Lenguaje Natural. A diferencia de los lenguajes de programación (como C++), que son formales, inequívocos y basados en reglas sintácticas estrictas, el lenguaje natural (como el español) es el medio de comunicación humana, caracterizado por su flexibilidad, ambigüedad y dependencia del contexto [19].

El puente tecnológico que permite a las máquinas interpretar este lenguaje es el Procesamiento de Lenguaje Natural (PLN) [17, 18], un subcampo de la IA. El PLN se define como "el conjunto de técnicas computacionales que permiten a las máquinas comprender, interpretar y generar lenguaje humano". Esto se logra mediante tareas como la tokenización (dividir texto en palabras), la normalización (reducir "corría" a "correr") y el análisis sintáctico (entender la gramática de la oración).

El Desafío de la Ambigüedad en la Programación

El mayor desafío del PLN aplicado a la programación es la ambigüedad [15]. Una instrucción humana como "haz una lista de estudiantes" es ambigua: ¿significa (a) crear una variable de tipo list, (b) imprimir en pantalla una lista existente, o (c) diseñar una interfaz gráfica con una lista? Los lenguajes de programación no pueden tolerar esta ambigüedad.

Por esta razón, la investigación en PLN para la educación no se centra solo en modelos que lo "entiendan todo", sino en cómo usar el PLN para reducir la carga cognitiva del estudiante. El objetivo es que el estudiante se concentre en el que (la lógica algorítmica) mientras la herramienta de IA maneja el cómo (la sintaxis exacta), una habilidad clave que se discute para los programadores en 2025.

PLN Estructurado vs. PLN de Formato Libre

En la enseñanza de la programación, el PLN se manifiesta principalmente de dos maneras: PLN de Formato Libre (Conversacional): Herramientas como GitHub Copilot Chat o ChatGPT permiten al usuario escribir una

instrucción completa ("escríbeme una función que ordene una lista"). Aunque potente, esta puede actuar como una "caja negra" que entrega una solución sin que el estudiante entienda el proceso.

PLN Estructurado (o Lenguaje Natural Controlado): Este es el enfoque que utiliza tu proyecto, similar a herramientas como PSeInt. No permite al usuario escribir cualquier cosa, sino que ofrece "bloques y palabras clave" en lenguaje natural (ej. "Escribir", "Leer", "Para cada"). Este enfoque restringe la ambigüedad de forma deliberada. Su beneficio pedagógico es que guía al estudiante a través de la sintaxis correcta, actuando como un "andamio semántico" y forzando al estudiante a estructurar su pensamiento de manera algorítmica [16].

Esta herramienta se sitúa en esta segunda categoría, utilizando los principios del PLN para crear un puente directo y sin ambigüedades entre el pensamiento lógico en español y la sintaxis formal de C++.

2.5. Modelos de Aprendizaje Asistido con IA

Las plataformas de aprendizaje impulsadas por IA están diseñadas para personalizar la educación. A diferencia de los Sistemas de Gestión de Aprendizaje (LMS) tradicionales que ofrecen contenido estático, estas plataformas "utilizan el aprendizaje automático para personalizar el aprendizaje, impulsando el compromiso, la eficiencia y la retención".

Los modelos específicos de IA para programadores que facilitan este aprendizaje incluyen:

- Asistentes de Código (ej. GitHub Copilot, Tabnine): Se integran en el editor (IDE) y sugieren líneas de código o funciones completas en tiempo real. Son útiles para la productividad y el aprendizaje de sintaxis.
- Modelos Conversacionales (ej. ChatGPT-4, Claude): Permiten al estudiante "dialogar" sobre un problema, pedir explicaciones de código, depurar errores lógicos o traducir código de un lenguaje a otro.
- Entornos de Aprendizaje Integrados (ej. Replit Ghostwriter): Diseñados específicamente para estudiantes, combinan un editor de código con un asistente de IA que explica los conceptos.

2.6. Análisis Comparativo de Soluciones Existentes vs. Propuesta Didáctica

Si bien los modelos mencionados anteriormente representan el estándar en la industria del desarrollo de software, es fundamental distinguir su utilidad profesional de su eficacia puramente pedagógica en etapas iniciales de formación. Herramientas comerciales como ChatGPT o Blackbox AI priorizan la velocidad y la resolución inmediata, lo cual puede resultar contraproducente para el desarrollo de la lógica algorítmica en estudiantes novatos.

A continuación, se presenta un análisis comparativo entre los asistentes de IA más predominantes en el mercado actual y la herramienta educativa propuesta en este proyecto ("Tu Web"), evaluando características clave como la especialización, la metodología de entrega de información y el control del proceso de aprendizaje.

3. DESARROLLO DE LA APLICACIÓN

Describe el funcionamiento, arquitectura y ejecución de la Herramienta de Traducción de Lenguaje Natural a Python, una aplicación web Full Stack diseñada para convertir instrucciones escritas en lenguaje natural (en español) en código fuente Python de manera automática.

El sistema hace uso de un modelo de inteligencia artificial especializado en generación de código (Codellama), el cual se ejecuta en un servidor privado mediante el motor Ollama. La interacción con el usuario

se realiza a través de una interfaz tipo chat que transmite las respuestas generadas en tiempo real.

La aplicación está construida bajo una arquitectura desacoplada, compuesta por un frontend estático y un backend independiente expuesto mediante una API (Application Programming Interface) REST.

3.1. Funcionamiento General de la Aplicación

El uso de la aplicación es intuitivo. El usuario introduce una instrucción en lenguaje natural dentro de la interfaz de chat, por ejemplo:

- "Crea una función que sume dos números".
- "Genera una clase llamada Perro con un método para ladrar".

El backend recibe la solicitud, la envía al modelo Codellama configurado para generación de código Python y transmite la respuesta progresivamente al frontend, donde se muestra el código generado en tiempo real.

3.2. Características Principales

- **Generación de Código en Tiempo Real:** El código aparece progresivamente, palabra por palabra, a medida que el modelo lo genera.
- **Resaltado Automático de Sintaxis:** El código Python se colorea utilizando una librería especializada para facilitar su lectura.
- **Botón de Copiado Rápido:** Al pasar el cursor sobre un bloque de código, aparece un botón que permite copiar el contenido.
- **Asistente Interactivo "Clippy":** Un ícono ubicado en la esquina inferior derecha brinda consejos aleatorios sobre Python y fundamentos de programación.

3.3. Estructura del Repositorio

```
├── .idea/
└── inspectionProfiles/
│       └── profiles_settings.xml # Configuración de perfiles de inspección del IDE (validaciones, análisis estático)
│   ├── LNAP.iml                  # Configuración del módulo del proyecto en PyCharm; Lenguaje Natural A Programas (intérprete, rutas y dependencias)
│    ├── jsLibraryMappings.xml     # Mapeo de bibliotecas JavaScript reconocidas por el IDE (ej. highlight.js)
│       ├── material_theme_project_new.xml  # Preferencias visuales del plugin Material Theme UI
│    ├── misc.xml                  # Ajustes generales del proyecto y definición del SDK de Python
│   ├── modules.xml                # Registro de los módulos del proyecto utilizados por el IDE
│   ├── vcs.xml                    # Configuración del sistema de control de versiones (Git)
│    └── workspace.xml              # Estado de trabajo del entorno: historial, ventanas, tareas y configuraciones temporales
├── backend/
│    └── app.py          # Servidor Flask con la integración de Ollama
├── docs/
│   ├── index.html    # Interfaz del chat y módulo Clippy
│   ├── style.css     # Estilos de la aplicación
│    └── script.js      # Lógica del frontend y consumo del API
└── README.md          # Documentación general del proyecto
```

Directorio ".idea"

El directorio .idea/ es generado automáticamente por **PyCharm** (o cualquier IDE basado en IntelliJ) y contiene la configuración interna del proyecto. Estos archivos no forman parte del funcionamiento de la aplicación en producción; sin embargo, son relevantes para documentar el entorno de desarrollo.

LNAP.iml: Define el módulo principal del proyecto dentro del entorno PyCharm. Incluye la configuración del intérprete de Python, la asociación con el entorno virtual (.venv) y las bibliotecas utilizadas durante el desarrollo. Asimismo, especifica los directorios incluidos y excluidos, así como las dependencias internas empleadas por el IDE para el análisis del código (Fig. 1).

1 Declara el módulo como un proyecto Python.
2 Establece el SDK de Python 3.13 asignado.
3 Excluye la carpeta .venv de los análisis del

IDE.

4 Registra bibliotecas como highlight.js y las dependencias del intérprete virtual.

jsLibraryMappings.xml: Registra las bibliotecas JavaScript externas que PyCharm debe reconocer dentro del proyecto. En este caso, asocia la librería highlight.js al directorio raíz del proyecto para habilitar funciones de autocompletado, análisis estático y referencia cruzada en los archivos JS (Fig. 2).

material_theme_project_new.xml: Incluye las preferencias visuales específicas del plugin *Material Theme UI*, utilizado en el IDE. Este archivo almacena metadatos de la configuración estética aplicada al proyecto, como el esquema de colores, migraciones internas y ajustes personalizados por el desarrollador (Fig. 3).

misc.xml: Abarca las configuraciones generales del proyecto, incluyendo el intérprete global de Python asignado (Python 3.13 (LNAP)) y parámetros adicionales de herramientas automáticas. Es un archivo clave para que PyCharm reconozca el entorno base de ejecución (Fig. 4).

modules.xml: Lista los módulos que conforman el proyecto. En este caso, solo se registra un módulo: LNAP.iml. PyCharm utiliza este archivo para gestionar la estructura modular interna y cargar la configuración correspondiente al iniciar el proyecto (Fig. 5).

```
<?xml version="1.0" encoding="UTF-8"?>
<module type="PYTHON_MODULE" version="4">
  <component name="FacetManager">
    <facet type="Python" name="Python facet">
      <configuration sdkName="Python 3.13 virtualenv at ~/PycharmProjects/LNAP/.venv" />
    </facet>
  </component>
  <component name="NewModuleRootManager">
    <content url="file://$MODULE_DIR$">
      <excludeFolder url="file://$MODULE_DIR$/.venv" />
    </content>
    <orderEntry type="jdk" jdkName="Python 3.13 (LNAP)" jdkType="Python SDK" />
    <orderEntry type="sourceFolder" forTests="false" />
    <orderEntry type="library" name="highlight.js" level="application" />
    <orderEntry type="library" name="Python 3.13 virtualenv at ~/PycharmProjects/LNAP/.venv interpreter library" level="application" />
  </component>
</module>
```

Figura 1. Código "LNAP.iml"

```
<project version="4">
  <component name="JavaScriptLibraryMappings">
    <file url="file://$PROJECT_DIR$" libraries="{highlight.js}"/>
  </component>
</project>
```

Figura 2. Código "jsLibraryMappings.xml"

```
<project version="4">
  <component name="MaterialThemeProjectNewConfig">
    <option name="metadata">
      <MTProjectMetadataState>
        <option name="migrated" value="true"/>
        <option name="pristineConfig" value="false"/>
        <option name="userId" value="593ee508:19a1408352a:-7ffe"/>
      </MTProjectMetadataState>
    </option>
  </component>
</project>
```

Figura 3. Código "material_theme_project_new.xml"

```
<project version="4">
  <component name="Black">
    <option name="sdkName" value="Python 3.13 (LNAP)"/>
  </component>
  <component name="ProjectRootManager" version="2" project-jdk-name="Python 3.13 (LNAP)" project-jdk-type="Python SDK"/>
</project>
```

Figura 4. Código "misc.xml"

vcs.xml: Especifica la configuración del sistema de control de versiones (Version Control System). Aquí se indica que el proyecto utiliza **Git** y se asigna el directorio raíz como repositorio principal. Esta configuración permite al IDE habilitar herramientas como commits, merges, diff y logs (Fig. 6).

workspace.xml: Es el archivo más extenso y contiene toda la información relacionada con el estado de trabajo del desarrollador. Incluye:

- Historial de tareas (TaskManager)
- Cambios recientes del repositorio
- Configuración de ventanas y paneles del IDE
- Preferencias de autocompletado
- Datos de GitHub vinculados (pull requests, cuentas, repositorio)
- Recientes archivos modificados
- Branches visitadas
- Configuraciones temporales de ejecución

Este archivo es personal del entorno de desarrollo y nunca se utiliza en producción (Fig. 7).

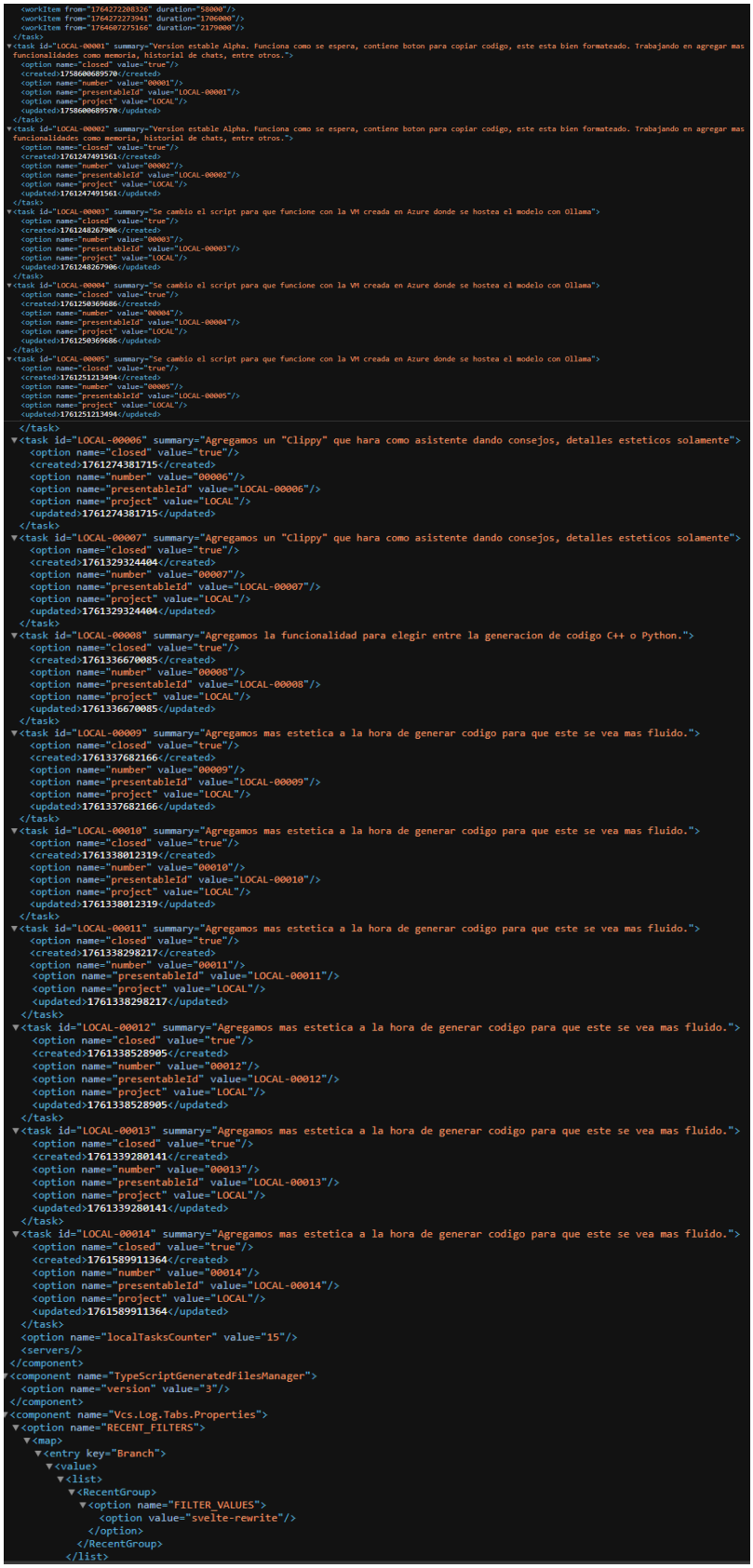
```
<project version="4">
  <component name="ProjectModuleManager">
    <modules>
      <module fileurl="file://$PROJECT_DIR$/.idea/LNAP.iml" filepath="$PROJECT_DIR$/.idea/LNAP.iml"/>
    </modules>
  </component>
</project>
```

Figura 5. Código “modules.xml”

```
<project version="4">
  <component name="VcsDirectoryMappings">
    <mapping directory="$PROJECT_DIR$" vcs="Git"/>
  </component>
</project>
```

Figura 6. Código “vcs.xml”

Figura 7. Código “workspace.xml”



```
    <workItem from="1764272208326" duration="58000"/>
    <workItem from="1764272273941" duration="1706000"/>
    <workItem from="1764607275166" duration="2179000"/>
  </task>
  <task id="LOCAL-00001" summary="Version estable Alpha. Funciona como se espera, contiene boton para copiar codigo, este esta bien formateado. Trabajando en agregar mas funcionalidades como memoria, historial de chats, entre otros.">
    <option name="closed" value="true"/>
    <created>1758600689570</created>
    <option name="number" value="00001"/>
    <option name="presentableId" value="LOCAL-00001"/>
    <option name="project" value="LOCAL"/>
    <updated>1758600689570</updated>
  </task>
  <task id="LOCAL-00002" summary="Version estable Alpha. Funciona como se espera, contiene boton para copiar codigo, este esta bien formateado. Trabajando en agregar mas funcionalidades como memoria, historial de chats, entre otros.">
    <option name="closed" value="true"/>
    <created>1761247491561</created>
    <option name="number" value="00002"/>
    <option name="presentableId" value="LOCAL-00002"/>
    <option name="project" value="LOCAL"/>
    <updated>1761247491561</updated>
  </task>
  <task id="LOCAL-00003" summary="Se cambio el script para que funcione con la VM creada en Azure donde se hostea el modelo con Ollama">
    <option name="closed" value="true"/>
    <created>1761248267906</created>
    <option name="number" value="00003"/>
    <option name="presentableId" value="LOCAL-00003"/>
    <option name="project" value="LOCAL"/>
    <updated>1761248267906</updated>
  </task>
  <task id="LOCAL-00004" summary="Se cambio el script para que funcione con la VM creada en Azure donde se hostea el modelo con Ollama">
    <option name="closed" value="true"/>
    <created>1761250369686</created>
    <option name="number" value="00004"/>
    <option name="presentableId" value="LOCAL-00004"/>
    <option name="project" value="LOCAL"/>
    <updated>1761250369686</updated>
  </task>
  <task id="LOCAL-00005" summary="Se cambio el script para que funcione con la VM creada en Azure donde se hostea el modelo con Ollama">
    <option name="closed" value="true"/>
    <created>1761251213494</created>
    <option name="number" value="00005"/>
    <option name="presentableId" value="LOCAL-00005"/>
    <option name="project" value="LOCAL"/>
    <updated>1761251213494</updated>
  </task>
  <task id="LOCAL-00006" summary="Agregamos un "Clippy" que hara como asistente dando consejos, detalles esteticos solamente">
    <option name="closed" value="true"/>
    <created>1761274381715</created>
    <option name="number" value="00006"/>
    <option name="presentableId" value="LOCAL-00006"/>
    <option name="project" value="LOCAL"/>
    <updated>1761274381715</updated>
  </task>
  <task id="LOCAL-00007" summary="Agregamos un "Clippy" que hara como asistente dando consejos, detalles esteticos solamente">
    <option name="closed" value="true"/>
    <created>1761329324404</created>
    <option name="number" value="00007"/>
    <option name="presentableId" value="LOCAL-00007"/>
    <option name="project" value="LOCAL"/>
    <updated>1761329324404</updated>
  </task>
  <task id="LOCAL-00008" summary="Agregamos la funcionalidad para elegir entre la generacion de codigo C++ o Python.">
    <option name="closed" value="true"/>
    <created>1761336670085</created>
    <option name="number" value="00008"/>
    <option name="presentableId" value="LOCAL-00008"/>
    <option name="project" value="LOCAL"/>
    <updated>1761336670085</updated>
  </task>
  <task id="LOCAL-00009" summary="Agregamos mas estetica a la hora de generar codigo para que este se vea mas fluido.">
    <option name="closed" value="true"/>
    <created>1761337682166</created>
    <option name="number" value="00009"/>
    <option name="presentableId" value="LOCAL-00009"/>
    <option name="project" value="LOCAL"/>
    <updated>1761337682166</updated>
  </task>
  <task id="LOCAL-00010" summary="Agregamos mas estetica a la hora de generar codigo para que este se vea mas fluido.">
    <option name="closed" value="true"/>
    <created>1761338012319</created>
    <option name="number" value="00010"/>
    <option name="presentableId" value="LOCAL-00010"/>
    <option name="project" value="LOCAL"/>
    <updated>1761338012319</updated>
  </task>
  <task id="LOCAL-00011" summary="Agregamos mas estetica a la hora de generar codigo para que este se vea mas fluido.">
    <option name="closed" value="true"/>
    <created>1761338298217</created>
    <option name="number" value="00011"/>
    <option name="presentableId" value="LOCAL-00011"/>
    <option name="project" value="LOCAL"/>
    <updated>1761338298217</updated>
  </task>
  <task id="LOCAL-00012" summary="Agregamos mas estetica a la hora de generar codigo para que este se vea mas fluido.">
    <option name="closed" value="true"/>
    <created>1761338528905</created>
    <option name="number" value="00012"/>
    <option name="presentableId" value="LOCAL-00012"/>
    <option name="project" value="LOCAL"/>
    <updated>1761338528905</updated>
  </task>
  <task id="LOCAL-00013" summary="Agregamos mas estetica a la hora de generar codigo para que este se vea mas fluido.">
    <option name="closed" value="true"/>
    <created>1761339280141</created>
    <option name="number" value="00013"/>
    <option name="presentableId" value="LOCAL-00013"/>
    <option name="project" value="LOCAL"/>
    <updated>1761339280141</updated>
  </task>
  <task id="LOCAL-00014" summary="Agregamos mas estetica a la hora de generar codigo para que este se vea mas fluido.">
    <option name="closed" value="true"/>
    <created>1761589911364</created>
    <option name="number" value="00014"/>
    <option name="presentableId" value="LOCAL-00014"/>
    <option name="project" value="LOCAL"/>
    <updated>1761589911364</updated>
  </task>
  <option name="localTasksCounter" value="15"/>
  <servers/>
</component>
<component name="TypeScriptGeneratedFilesManager">
  <option name="version" value="3"/>
</component>
<component name="Vcs.Log.Tabs.Properties">
  <option name="RECENT_FILTERS">
    <map>
      <entry key="Branch">
        <value>
          <list>
            <RecentGroup>
              <option name="FILTER_VALUES">
                <option value="svelte-rewrite"/>
              </option>
            </RecentGroup>
          </list>
```


Figura 7 (continuación). Código “workspace.xml”

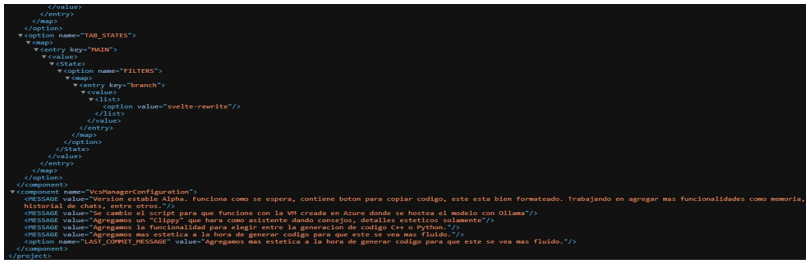


Figura 7 (continuación). Código “workspace.xml”

3.4. Frontend

El frontend está construido utilizando estándares modernos de desarrollo web. Cada tecnología cumple un rol específico dentro de la arquitectura:

HTML5 (HyperText Markup Language 5)

HTML5 se emplea como la base estructural de la aplicación. Sus funciones principales incluyen:

- Definir la estructura general de la interfaz tipo chat.
- Organizar los contenedores para los mensajes del usuario y las respuestas generadas por el backend.
- Incluir elementos interactivos como botones, íconos y el módulo adicional “Clippy”.
- Facilitar la integración de recursos externos mediante etiquetas <script>, <link>, e importaciones de librerías.

HTML5 permite una estructura semántica clara, lo que mejora la accesibilidad, la mantenibilidad y la compatibilidad con navegadores modernos.

CSS3 (Cascading Style Sheets 3)

CSS3 se utiliza para definir la apariencia visual de la aplicación. Dentro del proyecto cumple funciones como:

- Establecer el diseño responsivo de la interfaz, permitiendo que la aplicación se adapte a distintos tamaños de pantalla.
- Aplicar estilos visuales para diferenciar mensajes del usuario y mensajes generados por la IA.
- Diseñar el panel de conversación, botones interactivos y el módulo Clippy.
- Incorporar animaciones sutiles que mejoran la experiencia del usuario.
- Personalizar los colores y tipografías que acompañan al resaltado de código.

CSS3 garantiza una experiencia visual coherente y profesional, manteniendo la estética de un entorno de programación moderno.

JavaScript (ES6+)

JavaScript, en su versión ES6 en adelante, constituye la capa lógica del frontend. Dentro de esta aplicación, su rol es indispensable para:

- Capturar la instrucción escrita por el usuario en el chat.
- Enviar las solicitudes al servidor backend mediante fetch() utilizando el método HTTP POST.
- Recibir la respuesta en forma de *streaming* y mostrarla progresivamente en la pantalla, simulando una escritura en tiempo real.
- Invocar highlight.js para colorear automáticamente el código generado.
- Administrar eventos como clics, hover y acciones del asistente Clippy.
- Evitar bloqueos de la interfaz mediante el uso de promesas y asincronía.
- Gestionar el contenido dinámico sin necesidad de recargar la página.

JavaScript es el puente que conecta la experiencia visual del usuario con la lógica del backend, garantizando interactividad y reactividad.

Hospedaje del Frontend: GitHub Pages (lnap.dev)

El frontend se encuentra hospedado en *GitHub Pages*, un servicio de alojamiento estático que permite publicar sitios web directamente desde un repositorio GitHub. En este proyecto:

- El dominio personalizado lnap.dev redirecciona al contenido alojado en GitHub Pages.
- El sitio funciona de manera completamente estática, lo que implica que:
  - No se ejecuta lógica del lado del servidor en esta capa.
  - Todas las interacciones que requieren procesamiento son enviadas al backend.
- GitHub Pages ofrece:
  - Despliegue rápido y gratuito.
  - Alta disponibilidad debido a su infraestructura global.
  - Integración automática con GitHub Actions si se requiere automatizar el despliegue.

El hospedaje estático es adecuado para este proyecto, ya que todo el procesamiento complejo se realiza del lado del backend, permitiendo mantener el frontend ligero y altamente accesible.

Biblioteca externa: highlight.js

Para mejorar la visualización del código generado por la IA, el proyecto utiliza la biblioteca *highlight.js*, que es una herramienta ampliamente utilizada para el resaltado de sintaxis en aplicaciones web. *highlight.js* se emplea para:

- Analizar automáticamente el código recibido en la respuesta del backend.
- Aplicar formatos de color y estilo basados en las reglas del lenguaje Python.
- Facilitar la lectura e interpretación del código generado.
- Permitir que el resaltado se actualice dinámicamente conforme se recibe el *streaming* del texto.

Las ventajas técnicas son

- No requiere configuración compleja.
- Identifica el lenguaje automáticamente.
- Soporta más de 190 lenguajes, lo que facilita futuras ampliaciones del proyecto a otros lenguajes de programación.
- Puede integrarse en tiempo real, lo que es ideal para flujos de texto generados progresivamente como en esta aplicación.

Integración

La biblioteca se incluye en el proyecto mediante enlaces CDN (Content Delivery Network) desde el archivo index.html, y su ejecución se activa desde script.js cada vez que se recibe y renderiza código generado por la IA.

3.5. BACKEND

El backend está construido utilizando *Python* 3, un lenguaje ampliamente adoptado en el ámbito académico e industrial debido a su sintaxis clara, robusta comunidad y extenso ecosistema de librerías. Para la construcción de la API, se emplea Flask, un microframework ligero y modular que permite desarrollar servicios web de manera eficiente. Flask destaca por:

- Su arquitectura minimalista.
- Su alta extensibilidad mediante “blueprints” y extensiones oficiales.
- Su facilidad de integración con herramientas de inteligencia artificial.

En esta ocasión, Flask actúa como el servidor principal desde el cual se gestionan todas las solicitudes provenientes del frontend.

Inteligencia Artificial: motor Ollama ejecutando el modelo Codellama

El backend integra un componente de

inteligencia artificial a través de Ollama, un motor diseñado para ejecutar modelos de lenguaje grandes (LLMs) de manera local o remota. Ollama permite:

- Cargar y ejecutar modelos de IA sin depender de servicios externos.
- Asegurar mayor privacidad al mantener los datos dentro del servidor propio.
- Optimizar tiempos de inferencia al utilizar hardware dedicado.

El modelo utilizado es Codellama, un LLM especializado en generación de código y razonamiento sintáctico. Su rol dentro del proyecto es transformar instrucciones textuales en código Python coherente y funcional.

API: ruta principal /api/generate que recibe peticiones POST

El backend expone una API muy simplificada pero altamente funcional, compuesta por una única ruta principal:
/api/generate
Esta ruta acepta únicamente solicitudes POST y es responsable de:

- Recibir un objeto JSON con un prompt en lenguaje natural.
- Enviar dicho prompt al modelo Codellama ejecutado en Ollama.
- Transmitir la respuesta al cliente como un flujo (*stream*) continuo de texto.

La implementación de streaming permite que el usuario observe el código siendo generado en tiempo real.

Seguridad CORS: implementada con Flask-CORS

Para permitir la comunicación entre el frontend (hospedado en un dominio distinto) y el backend, se incorpora Flask-CORS (Cross-Origin Resource Sharing), una extensión que habilita el manejo seguro de políticas de intercambio de recursos entre orígenes cruzados.

Esto garantiza que el backend acepte solicitudes solamente desde:

- El dominio oficial del proyecto.
- Servidores autorizados para desarrollo local.

El control CORS evita accesos no autorizados y protege la API contra solicitudes maliciosas provenientes de otros dominios.

3.6. API

Este endpoint permite al usuario enviar una instrucción en lenguaje natural. El backend recibe dicho prompt, lo envía al modelo Codellama a través de Ollama y devuelve un flujo continuo de texto (stream) con el código Python generado.
El uso de streaming mejora la experiencia del usuario al mostrar la salida en tiempo real, sin necesidad de esperar a que el modelo finalice por completo la generación.

Endpoint Principal

Método: POST
Ruta: /api/generate
Host: https://api.lnap.dev

Cuerpo de la Solicitud (JSON)

```
{
"prompt": "Tu instrucción de texto aquí",
"model": "codellama"
}
```

Parámetros

- prompt: instrucción en texto natural escrita por el usuario.
- model: especifica el modelo a utilizar, en este caso codellama.

Respuesta

- *Content-Type*: text/plain
- *Formato*: *streaming*
- *Comportamiento*: El servidor transmite fragmentos de texto a medida que el

modelo genera el código.

Este diseño permite una respuesta inmediata y progresiva.

3.7. Infraestructura

La infraestructura de despliegue constituye el entorno operativo donde el backend se ejecuta de forma estable, segura y accesible desde Internet. Este proyecto utiliza una arquitectura optimizada y distribuida que separa el hosting del frontend y del backend.

Backend alojado en una máquina virtual de Digital Ocean (Ubuntu)

El backend se encuentra desplegado en una *Máquina Virtual* (VM) *en Digital Ocean*, uno de los proveedores de nube más robustos del mercado. Se utiliza un sistema operativo *Ubuntu*, conocido por su estabilidad, seguridad y amplia documentación.
Este entorno ofrece:

1) Recursos dedicados para ejecutar el modelo de IA.
2) Alta disponibilidad y escalabilidad.
3) Control total sobre el entorno de ejecución del servidor.

Servidor web: Nginx como proxy inverso

Para gestionar el tráfico entrante se utiliza Nginx configurado como *reverse proxy*. Esta herramienta permite:

1 Redirigir solicitudes HTTPS hacia el servidor Flask ejecutado internamente.
2 Manejar la carga de forma eficiente.
3 Servir contenido de manera segura y rápida.

Nginx también actúa como primera capa de seguridad, protegiendo el puerto del backend y administrando certificados SSL.

Certificados SSL generados por Let's Encrypt (Certbot)

El dominio del backend (https://api.lnap.dev) se encuentra asegurado mediante certificados SSL emitidos por Let's Encrypt, una autoridad certificadora reconocida internacionalmente. La gestión se realiza con Certbot, lo que permite:

• Renovación automática de certificados.
• Cifrado de extremo a extremo para todas las comunicaciones.
• Conexiones HTTPS seguras entre el frontend y la API.

Esto garantiza la integridad y confidencialidad de los datos transmitidos.

DNS administrado en name.com

La resolución de nombres de dominio se gestiona a través de name.com. Se utilizan registros A que apuntan a:

1 GitHub Pages → para el frontend (lnap.dev).
2 La VM de Digital Ocean → para el backend (api.lnap.dev).

Este esquema permite separar responsabilidades, optimizar tiempos de carga y facilitar el mantenimiento del sistema.

3.8. Ejecución

Para la ejecución local se requieren los siguientes componentes:

- *Python 3.10+ y pip*: necesarios para instalar dependencias y ejecutar el servidor Flask.
- *Ollama instalado y en ejecución*: motor encargado de correr el modelo de IA.
- *Modelo Codellama descargado*:

  • ollama pull codellama
  • Ollama debe estar activo antes de iniciar el backend.

Ejecución del Backend

Acceder a la carpeta del backend:

• cd backend

- (Opcional) Crear un entorno virtual:
  - python3 -m venv venv
  - source venv/bin/activate
- Instalar las dependencias necesarias:
  - pip install Flask flask-cors ollama
- Iniciar el servidor Flask:
  - python3 app.py
- El backend estará disponible en:
  - http://127.0.0.1:5001

Ejecución del Frontend

- Abrir en el navegador el archivo:
  - docs/index.html
- Modificar el archivo docs/script.js para apuntar al backend local:

```
// Valor original
const BACKEND_URL = "https://api.lnap.dev/api/generate";
// Valor para desarrollo local
const BACKEND_URL = "http://127.0.0.1:5001/api/generate";
```

- Actualizar backend/app.py para permitir solicitudes desde archivos locales (origin "null"):

```
# Configuración original
origins = ["https://lnap.dev", "http://localhost:5173"]
# Configuración para entorno local
origins = ["https://lnap.dev", "http://localhost:5173", "null"]
```

- Reiniciar el servidor Flask para aplicar los cambios.

4. ANÁLISIS DE RESULTADOS

1. ¿En qué rango de semestres te encuentras actualmente en tu carrera universitaria?

Tabla 1. Rango de semestres

Respuestas	Cantidad	Porcentaje
1° a 3° semestre	23	41.8%
4° a 6° semestre	8	14.5%
7° a 9° semestre	21	38.2%
10° semestre o superior	3	5.5%
Total	**55**	**100%**

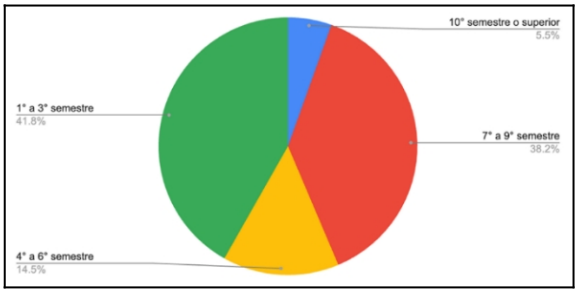


Figura 8. Rango de semestres en alumnos

Interpretación: Un significativo 41.8% son estudiantes de los primeros semestres, seguidos de un 14.5% de los semestres intermedios.

2. ¿Qué carrera cursas?

Tabla 2. Carrera que cursan.

Carrera	Cantidad	Porcentaje
Ing. Bioquímica	0	0.0%
Ing. Eléctrica	0	0.0%
Ing. Electrónica	7	12.7%
Ing. en Energías Renovables	0	0.0%
Ing. en Gestión Empresarial	0	0.0%
Ing. Industrial	0	0.0%
Ing. Mecánica	0	0.0%
Ing. Mecatrónica	0	0.0%
Ing. Química	0	0.0%
Ing. Sistemas Computacionales	46	83.6%
Lic. en Administración	0	0.0%
No contestó	2	3.6%
Total	**55**	**100%**

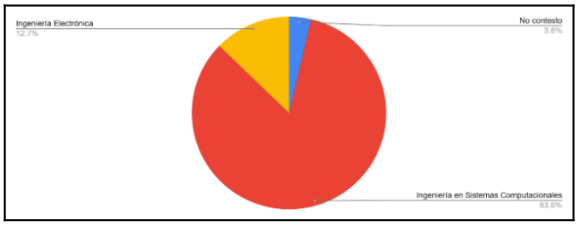


Figura 9. Carreras que cursan los alumnos.

Interpretación: El 86.8% de los encuestados son del área de Ingeniería en Sistemas Computacionales, el resto corresponden a la carrera de Ingeniería Electrónica.

3. ¿Cuál es tu nivel de conocimientos en programación previo al uso de esta herramienta?

Tabla 3. Nivel de conocimiento en programación.

Respuestas	Cantidad	Porcentaje
Nulo	5	9.1%
Básico	26	47.3%
Intermedio	21	38.2%
Avanzado	3	5.5%
Total	**55**	**100%**

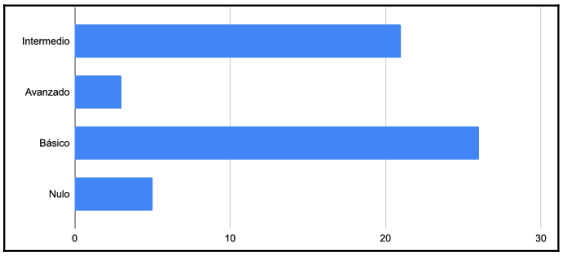


Figura 10. Rango de entendimiento en programación.

Interpretación: En cuanto al nivel de conocimientos en programación, el 47.3% de los encuestados manifiestan tener un nivel básico, pero también existe un 9.1% de estudiantes que no programan.

4. De manera general, ¿qué tan frecuente tuviste que corregir o depurar el código generado por la herramienta de IA para que funcionara correctamente?

Tabla 4. Apoyo en la herramienta.

Respuestas	Cantidad	Porcentaje
Nunca o casi nunca	15	27.3%
A veces	26	47.3%
En raras Ocasiones	8	14.5%
Frecuentemente	6	10.9%
Total	**55**	**100%**

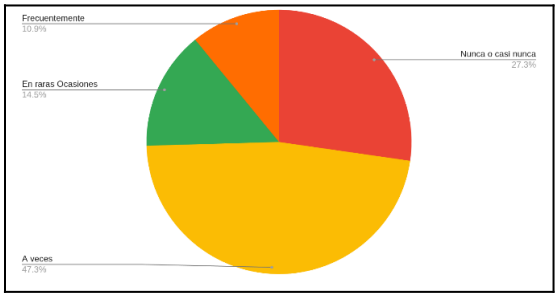


Figura 11. Uso de herramientas IA en códigos de programación.

Interpretación: Un muy significativo 89.1% declara que nunca, casi nunca, a veces o en raras ocasiones, tuvieron que corregir el código generado por la herramienta.

5. ¿Cómo calificarías tu experiencia general de interacción con la herramienta de IA?

Tabla 5. Experiencia general

Respuestas	Cantidad	Porcentaje
Muy mal	0	0%
Mal	5	9%
Indiferente	14	25.5%
Bien	22	40%
Excelente	14	25.5%
Total	**55**	**100%**

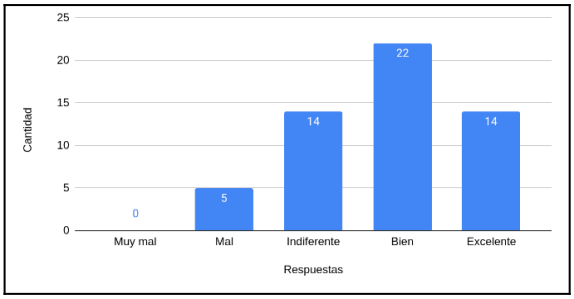


Figura 12. Experiencia general en su uso.

Interpretación: Un significativo 65.5% opinan que la experiencia con la herramienta va de Excelente a bien, sin embargo, existe un 9% que la calificó como una mala experiencia.

6. ¿Cómo te identificas con la siguiente frase: "Ver la generación del código en tiempo real facilitó mi comprensión de la lógica"?

Tabla 6. Comprensión lógica del código en tiempo real.

Respuestas	Cantidad	Porcentaje
Muy de Acuerdo	13	23.6%
De acuerdo	35	63.6%
Neutro	0	0%
En Desacuerdo	6	11%
Muy en Desacuerdo	1	1.8%
Total	**55**	**100%**

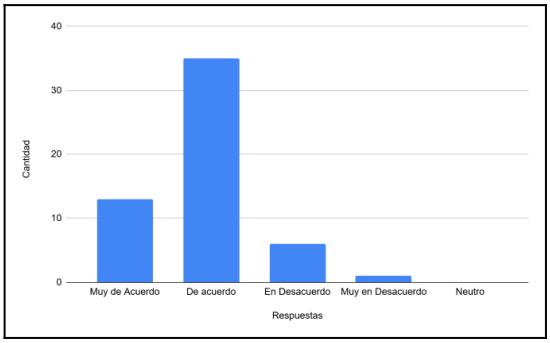


Figura 13. Lógica en programación

Interpretación: El 87.2% de los estudiantes están de acuerdo o muy de acuerdo en la facilidad de la herramienta como apoyo a la comprensión de la lógica. Un 12.8% manifiestan que no les facilitó dicha comprensión.

7. ¿Qué aspecto de la programación sentiste que la IA te facilitó más?

Tabla 7. Aspectos de programación en IA

Respuestas	Cantidad	Porcentaje
Entender la Lógica	8	14.5%
Corregir errores	15	27.3%
Escribir la sintaxis	17	30.9%
Comprobar código	5	9.1%
Todas las anteriores	10	18.2%
Total	**55**	**100%**

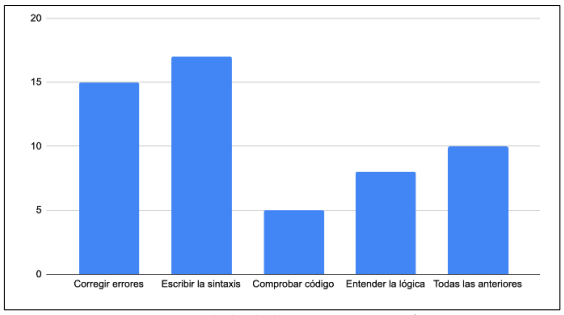


Figura 14. Facilidad de programación en IA.

Interpretación: En cuanto a los aspectos de la programación, un significativo 30.9% considera que la herramienta los apoyó en la sintaxis del lenguaje. Sobresale también la corrección de errores con el 27.3%. El 18.2% opinó que la herramienta apoya en todos los aspectos declarados (lógica, corrección de errores, sintaxis, comprobación de código).

8. En comparación con aprender viendo videos o leyendo tutoriales, ¿cómo calificarías tu experiencia con esta IA?

Tabla 8. Experiencia con la herramienta.

Respuestas	Cantidad	Porcentaje
Mucho más efectiva	13	23.6%
Un poco más efectiva	23	41.8%
Igual de efectiva	16	29.1%
Menos efectiva	3	5.5%
Total	**55**	**100%**

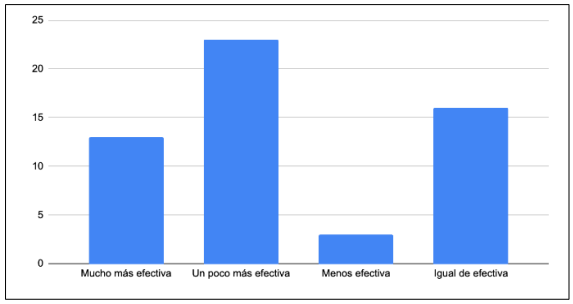


Figura 15. Eficiencia en alumnos y profesores con la herramienta

Interpretación: El 65.4% de los encuestados expresan que la herramienta es más efectiva que otras opciones de aprendizaje como videos o tutoriales.

9. En una escala del 1 al 5, ¿qué tan claros y comprensibles fueron los segmentos de código generados por la IA?

Tabla 9. Comprensión en los segmentos de código

Respuestas	Cantidad	Porcentaje
Nada Claros	1	1.8%
Poco claros	3	5.5%
Neutro	10	18.2%
Claros	20	36.4%
Muy Claros	21	38.2%
Total	55	100%

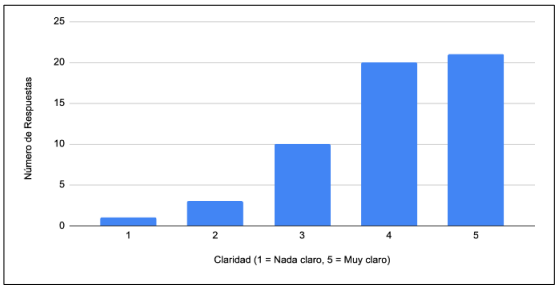


Figura 16. Entendimiento en profesores y alumnos sobre segmentos de código en la IA

Interpretación: Un significativo 74.6% opinó que los segmentos de código fueron Claros o muy claros.

10. Del 1 al 10 donde 1 es no lo recomendaría y 10 si lo recomendaría, ¿qué tanto recomendarías esta herramienta a otros estudiantes?

Tabla 10. Recomendaciones a otros estudiantes

Respuesta	Cantidad	Porcentaje
10	17	30.9%
9	7	12.7%
8	17	30.9%
7	5	9.1%
6	2	3.6%
5	4	7.3%
4	1	1.8%
3	0	0.0%
2	1	1.8%
1	1	1.8%
Total	**55**	**100%**

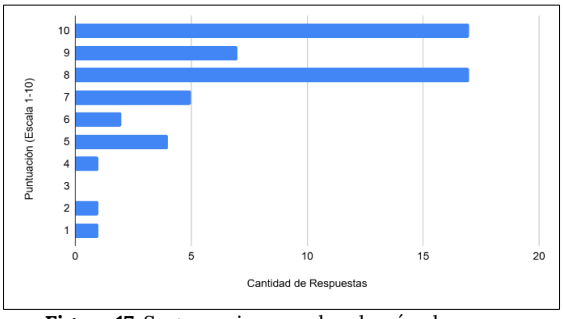


Figura 17. Sugerencias para los demás alumnos

Interpretación: La mayoría de los encuestados (74.5%) otorgó una calificación alta (entre 8 y 10), mostrando una fuerte disposición a recomendar la herramienta.

11. ¿Consideras que la herramienta de IA fue más útil que un buscador tradicional (como Google) para resolver tus dudas de programación específicas?

Tabla 11. Búsqueda de dudas en programación

Respuestas	Cantidad	Porcentaje
Si	47	85.5%
No	8	14.5%
Total	**55**	**100%**

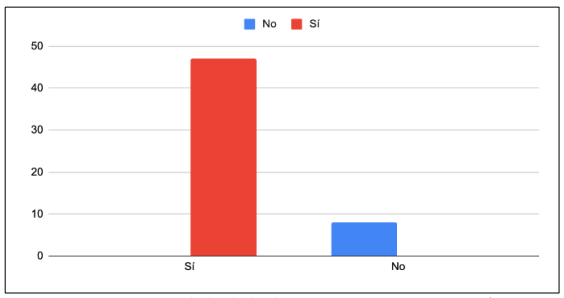


Figura 18. Utilidad de la IA para programación

Interpretación: La gran mayoría de encuestados opinó que si les fue de utilidad en contraste con un buscador tradicional.

12. ¿Consideras que el utilizar la herramienta te ayudaría a mejorar tu experiencia en el aprendizaje de la lógica de programación?

Tabla 12. Mejora aprendizaje de la lógica en programación

Respuestas	Cantidad	Porcentaje
Si	51	92.7%
No	4	7.3%
Total	**55**	**100%**

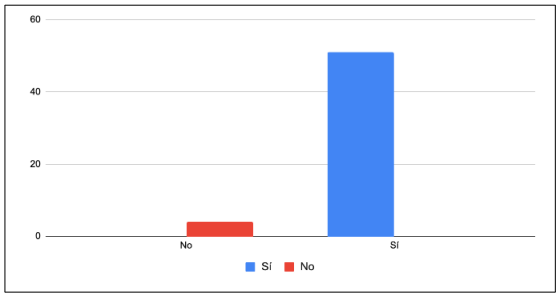


Figura 19. Mejora la experiencia de aprendizaje

Interpretación: El 92.7% considera que la herramienta mejora su experiencia en el aprendizaje de la lógica de programación.

13. ¿Qué función o característica le agregarías para mejorar su utilidad didáctica?

Tabla 13. Opiniones para mejorar

Respuesta	Cantidad	Porcentaje
No contestaron	11	20.0%
Mejorar servidores	11	20.0%
Explicación del código	10	18.2%
Diseño responsivo	9	16.4%
Errores en el código	4	7.3%
Ninguna	3	5.5%
Otros lenguajes	2	3.6%
Agregar características	2	3.6%
Está bien	2	3.6%
Traducción al inglés	1	1.8%
Total	**55**	**100%**

Interpretación: De los 55 encuestados, sobresalen los que opinan sobre la mejora a servidores (11), explicación del código (10), Diseño responsivo (9); de tomarse en cuenta 9 encuestados que opinan se haga traducción al inglés, se agreguen características y otros lenguajes, así como errores en código.

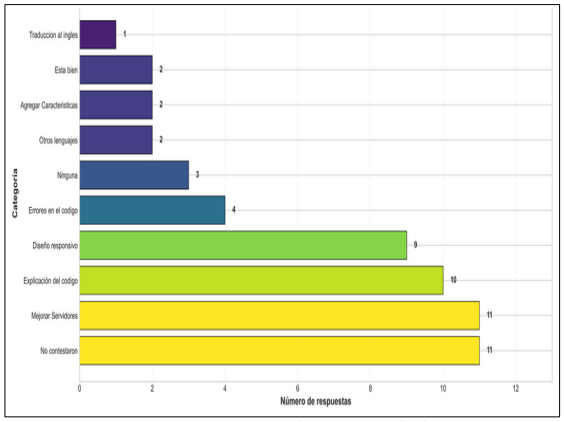


Figura 20. Criterios a considerar para la herramienta

5. CONCLUSIONES

A partir del desarrollo, implementación y evaluación de la herramienta propuesta, se concluye que el uso de inteligencia artificial generativa con un enfoque pedagógico guiado puede constituir un apoyo significativo en el aprendizaje de la programación, particularmente en estudiantes de nivel principiante e intermedio. Los resultados obtenidos evidencian que la visualización progresiva de la generación de código, junto con la traducción de instrucciones en lenguaje natural a Python, favorece la comprensión de la lógica algorítmica, la sintaxis y los procesos de razonamiento subyacentes, reduciendo la dependencia de soluciones inmediatas y opacas. La alta aceptación por parte de los usuarios, reflejada en los niveles de satisfacción, claridad percibida y disposición a recomendar la herramienta, confirma su utilidad como complemento educativo frente a métodos tradicionales y asistentes de IA comerciales.

REFERENCIAS

[1] Barrios del Valle G. Enseñar programación en la era de la IA generativa. Revista Mexicana de Bachillerato a Distancia (2025) 17(34). doi: 10.22201/cuaed.20074751e.2025.34.92845

[2] Codalista. Los mejores modelos de IA para programadores en 2025. Codalista.com; 2025 [citado 2025 Nov 6]. https://codalista.com/los-mejores-modelos-de-ia-para-programadores-en-2025/

[3] Educ.ar. Inteligencia artificial. Portal educativo de la Secretaría de Educación; 2025 [citado 2025 Nov 6]. https://www.educ.ar/IA

[4] Fullmind. AI in education: 10 real applications of EdTech in 2025. Fullmind; 2025 Jul 21. https://www.fullmindlearning.com/blog/ai-in-education-applications

[5] InvestGlass. Las 7 mejores soluciones de plataformas de aprendizaje de IA para 2025. InvestGlass; 2025 May 25. https://www.investglass.com/es/top-7-ai-learning-platform-solutions-for-2025-4/

[6] León Morejón Y, Maia F, Alvarez Riol L de la C. Impactos de la inteligencia artificial en la autorregulación del aprendizaje de la programación. Mendive. Revista de Educación (2025) 23(2):e4002. https://mendive.upr.edu.cu/index.php/MendiveUPR/article/view/4002

[7] Maestrías y Diplomados Tec.mx. IA para programadores: las mejores herramientas de 2025. Blog; 2025 May 28. https://blog.maestriasydiplomados.tec.mx/ia-para-programadores

[8] MENTOR. Inteligencia artificial: ventajas y desventajas de su uso en el proceso de enseñanza aprendizaje. MENTOR Revista de Investigación Educativa y Deportiva (2025). https://revistamentor.ec/index.php/mentor/article/view/7081

[9] Microsoft. 2025 AI in education: a Microsoft special report. Microsoft; 2025. https://cdn-dynmedia-1.microsoft.com/is/content/microsoftcorp/microsoft/bade/documents/products-and-services/en-us/education/2025-Microsoft-AI-in-Education-Report.pdf

[10] Pineda Sánchez VY. La inteligencia artificial como herramienta de optimización en la docencia pública: retos y oportunidades para el futuro de la educación. RIESED - Revista Internacional de Estudios sobre Sistemas Educativos (2025) 3(16):803-822. doi: 10.5281/zenodo.15697704

[11] UNESCO. La IA y el futuro de la educación: disrupciones, dilemas y orientaciones. UNESCO; 2025 Sep 29. https://www.unesco.org/es/articles/el-uso-de-la-ia-en-la-educacion-decidir-el-futuro-que-queremos

[12] University Canada West. Advantages and disadvantages of AI in education. University Canada West; 2025 May 21. https://www.ucanwest.ca/blog/education-careers-tips/advantages-and-disadvantages-of-ai-in-education

[13] Vera Paredes DA, Franco Arias OO, Córdova Martínez LC. Impacto de la inteligencia artificial en el aprendizaje de la programación informática en los estudiantes universitarios. Ciencia y Educación (2025) 6(3):33-50. doi: 10.5281/zenodo.15056104

[14] Zaïane OR, Gârbacea IA, Bârsan EI. AI applications in education: a review. International Journal of Education and Information Technologies (2025) 19:33-38.

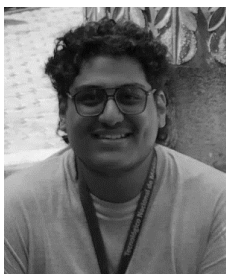
https://www.researchgate.net/publication/389971525_AI_Applications_in_Education_A_review
[15] CampusMVP. ¿Merece la pena aprender a programar en 2025, en la era de la IA? CampusMVP; 2025 May 6. https://www.campusmvp.es/recursos/post/merece-la-pena-aprender-a-programar-en-2025-en-la-era-de-la-ia.aspx
[16] Santos R, Silva B. Semantic scaffolding: the role of Natural Language to Code (NL-to-Code) systems in novice programming education. En: Proc. of the 2025 IEEE Conf. on Education and Training (EDUCON). IEEE; 2025. p. 7-12.
[17] IBM. ¿Qué es el procesamiento del lenguaje natural (PLN)? IBM México; 2025 [citado 2025 Nov 11]. https://www.ibm.com/mx-es/think/topics/natural-language-processing
[18] KeepCoding. Procesamiento del lenguaje natural: guía esencial para entender y aplicar PLN. KeepCoding Blog; 2025 Sep 3 [citado 2025 Nov 11]. https://keepcoding.io/blog/procesamiento-del-lenguaje-natural-guia/
[19] Muñoz Guillena R. Procesamiento del lenguaje natural como eje central de la inteligencia artificial generativa [lección inaugural del Curso Académico 2024-2025]. Universidad de La Rioja, Servicio de Publicaciones; 2024. https://dialnet.unirioja.es/descarga/libro/985766.pdf

ACERCA DEL AUTOR

Ofelia Gutiérrez Giraldi. Docente del Instituto Tecnológico de Veracruz con 46 años de antigüedad. Actualmente tiene un nivel de profesor-investigador, trabajando en este rubro hasta antes de participar directamente en puestos directivos/administrativos: Jefa de proyecto de investigación, Jefa de la División de Estudios de Posgrado e Investigación y Subdirectora Académica. De vuelta al ámbito de profesor-investigador, he participado nuevamente en proyectos de investigación y reanudado mis labores como docente del Depto. de Sistemas y Computación.

Lizbeth Sanchez Ferrer. Originaria de Veracruz, Veracruz, México. Las áreas en las que se ha desarrollado es la configuración y administración de redes de computadoras, administración y gestión de proyectos, metodologías de investigación, de programación. Le gusta la leer y disfruto muchas películas sobre todo basadas en hechos reales, misterio y las que te hacen pensar. Práctico yoga y pilates.

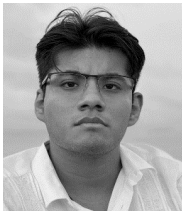
Martha Martínez Moreno. Graduada de Licenciatura en Informática en el Instituto Tecnológico de Orizaba en 1994. Obtuvo su Maestría en Gestión Educativa en el Instituto Universitario Veracruzano 2024. Docente en CONALEP ORIZABA en 1993, Universidad del Valle de Orizaba en 1993, CONALEP Veracruz II en 1996, Instituto Tecnológico de Veracruz en 1996, actualmente en el Instituto Tecnológico de Toluca desde 2020. Además, secretaria de academia en el Instituto Tecnológico de Toluca en 2021 y jefe de Departamento de Sistemas y Computación en el mismo instituto.



Jorge Angel Estudillo Silva. Estudiante de Ingeniería en Sistemas Computacionales con vasta experiencia en desarrollo con Python. Especializado en inteligencia artificial, ciencia de datos y desarrollo de software. Experiencia comprobada en investigación académica, habiendo desarrollado modelos de IA para clasificación médica en el CIMAT. Participante activo en programación competitiva ACM ICPC con sólidas habilidades en resolución de problemas algorítmicos.

Esteban Santos Angulo. Estudiante universitario en Sistemas Computacionales del Instituto Tecnológico de Veracruz con especialidad en Ciencia de Datos y Gestión de Proyectos de Software. Actualmente cursando el décimo semestre de la carrera.

Efren Ernesto Pérez Nicanor. Estudiante de Ingeniería en Sistemas Computacionales en el Instituto Tecnológico de Veracruz, con especialidad en Ciencia de Datos. Ha complementado su formación participando en competencias como Coding Cup e ICPC, y en el programa de especialización Oracle ONE (2024). Actualmente, colabora en proyectos de investigación sobre Inteligencia Artificial Generativa y modelos LLM (CodeLlama), enfocándose principalmente en el análisis de resultados, la documentación técnica y la integración de informes de investigación. Sus áreas de interés comprenden los algoritmos matemáticos y la programación en Java.



David León Salas. Estudiante del Instituto Tecnológico de Veracruz, en la carrera de Ingeniería en Sistemas Computacionales. Cuenta con una certificación en Inteligencia Artificial otorgada por Samsung Innovation Campus. Actualmente se desempeña como residente en la empresa Tenaris Tamsa. Sus intereses académicos y profesionales se centran en la Ciencia de Datos, particularmente en el análisis, modelado y visualización de información para la extracción de conocimiento, así como en el estudio y aplicación de las matemáticas como fundamento para el desarrollo de modelos y soluciones computacionales. Además, muestra un especial interés en el diseño y uso de bases de datos NoSQL para el manejo eficiente de grandes volúmenes de datos.