2.9 Programación: Asistencia en tareas seleccionadas

Aprender a programar sin que la IA escriba el código

Asignatura: Programación · **Curso:** 9 (14-15 años) · **Duración:** 1 clase doble (90 min) · **IA:** Asistencia **CON** IA

Competencias relacionadas con la IA (Marco DUAL.AI.TEACHer) – nivel docente	**Objetivos de aprendizaje específicos de la asignatura – nivel del alumnado**
AF-TL-2b *(Fundamentos y aplicaciones de la IA × Enseñanza y Aprendizaje, Nivel 2 – Implementación Reflexiva)*: «El profesorado puede organizar elementos de la clase que integren el uso de la IA con roles pedagógicos claros y estrategias para verificar los elementos del resultado.» **AF-FC-2b** *(Fundamentos y aplicaciones de la IA × Facilitar la Competencia Digital (en IA) del Alumnado, Nivel 2 – Implementación Reflexiva)*: «El profesorado puede aplicar orientaciones que ayuden al alumnado a desarrollar habilidades prácticas de IA, adaptando las tareas a los conocimientos previos del alumnado y a las herramientas de IA disponibles.»	Al final de la clase, el alumnado podrá: • saber cómo utilizar la IA para realizar tareas repetitivas o de apoyo en programación, como sangrar el código o escribir los comentarios • pedir a un modelo de IA que proponga mejoras en el código, sin darle el código en sí, de modo que el alumnado pueda autoevaluar sus conocimientos • evaluar críticamente al modelo de IA cuando se le pregunta cómo optimizaría el código dado

> **Mensaje clave:** El alumnado necesita conocer los fundamentos de la programación si quiere evaluar los resultados que ofrece un modelo. Por eso, en los primeros pasos del aprendizaje de la programación el alumnado debe utilizar los modelos de IA para tareas de apoyo y repetitivas relacionadas con la programación, pero no para producir código. También puede utilizar el modelo para pedir posibles mejoras del código y poner así a prueba sus capacidades de programación.

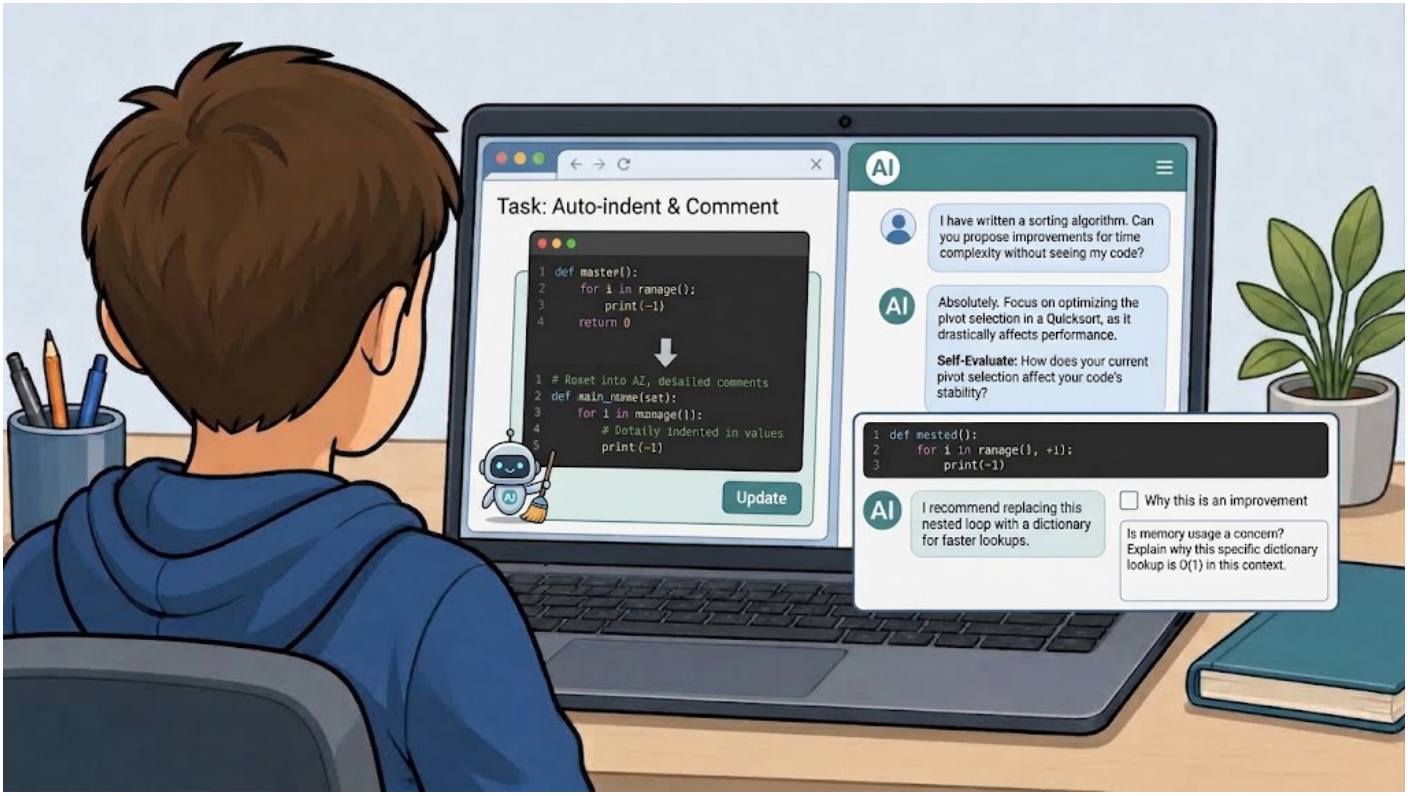


Fig. 1: «Aprender a programar sin que la IA escriba el código» (imagen generada con IA, Google Gemini 3.6 Thinking)

Contenido

La enseñanza de la programación está cambiando en un contexto en el que los sistemas de IA pueden generar, explicar y revisar código. Aunque estas herramientas ofrecen nuevas oportunidades de aprendizaje, también plantean un reto educativo importante: el alumnado necesita conocimientos suficientes de programación para evaluar si una sugerencia generada por IA es correcta, eficiente o siquiera pertinente para la tarea que tiene entre manos. Sin una comprensión básica de los conceptos de programación, el alumnado puede aceptar soluciones incorrectas, pasar por alto errores o no reconocer alternativas mejores. Por este motivo, aprender los fundamentos de la programación sigue siendo esencial, incluso cuando se dispone de herramientas de IA.

En los cursos introductorios de programación, la IA debería situarse como una herramienta de apoyo al aprendizaje y no como programadora sustituta. El alumnado puede utilizar la IA de forma eficaz para tareas repetitivas o de bajo nivel que no reducen el esfuerzo cognitivo que supone aprender a programar. Algunos ejemplos son dar formato y sangrar el código, generar comentarios de documentación, explicar mensajes de error, resumir reglas de sintaxis o ayudar al alumnado a entender la finalidad de determinadas construcciones de programación. Estos usos permiten al alumnado centrar su atención en la lógica subyacente, los algoritmos y los procesos de resolución de problemas, que son el núcleo de la competencia en programación.

Otro uso productivo de la IA es la revisión de código. En lugar de pedir a un modelo de IA que

escriba una solución, el alumnado puede escribir primero su propio código y después pedir al modelo que identifique posibles mejoras. La IA puede sugerir nombres de variables más claros, una mejor organización del código, menos redundancia, mayor legibilidad o enfoques alternativos que el alumnado puede evaluar de forma independiente. Esto convierte a la IA en una compañera de retroalimentación, y permite al alumnado comparar su propio razonamiento con sugerencias externas y valorar los puntos fuertes y débiles de sus soluciones.

Sin embargo, la retroalimentación generada por IA no debería aceptarse de forma acrítica. Los modelos pueden recomendar cambios innecesarios, ineficientes o basados en suposiciones incorrectas sobre la finalidad del código. Por tanto, un objetivo central de esta clase es desarrollar la capacidad del alumnado de evaluar críticamente las sugerencias de la IA. El alumnado aprenderá a preguntar a un modelo de IA no solo qué mejoras propone, sino también por qué considera beneficiosos esos cambios. Al comparar el razonamiento del modelo con principios de programación establecidos, el alumnado puede juzgar si la optimización propuesta mejora realmente el código o si solo representa una decisión de diseño distinta.

Esta clase trata la IA como una herramienta que apoya la reflexión, la autoevaluación y la mejora de la calidad del código, y no como generadora de código. El alumnado practica escribiendo sus propios programas, utilizando la IA para tareas de apoyo, pidiendo retroalimentación sobre su trabajo y analizando críticamente la calidad de las respuestas que recibe. El objetivo último es ayudar al alumnado a convertirse en mejores programadores y programadoras y, a la vez, en personas usuarias de sistemas de IA mejor informadas.

Plan de clase

Fase	Tiempo	Actividad
Introducción	10 min	El profesorado presenta dos soluciones breves de código para el mismo problema: una escrita por un alumno o una alumna y otra generada por un modelo de IA. El alumnado debate qué solución parece más clara, más eficiente o más fácil de entender.
Explicación	10 min	El profesorado presenta los usos adecuados de la IA en la enseñanza de la programación y subraya la distinción entre tareas de apoyo, revisión de código y generación de código.
Exploración	15 min	El alumnado trabaja por parejas en un pequeño ejercicio de programación e identifica qué tareas podrían delegarse adecuadamente en un asistente de IA y cuáles deberían seguir siendo responsabilidad propia.
Ciclo de mejora	35 min	El alumnado escribe su propia solución y completa 2 o 3 rondas de escribir → preguntar → evaluar → revisar, utilizando la IA únicamente para tareas de apoyo y sugerencias de mejora.
Reflexión	20 min	Debate con toda la clase sobre la utilidad y las limitaciones de la retroalimentación de la IA, centrado en cómo determinó el alumnado si las sugerencias eran valiosas o no.

Preguntas guía: añadir complejidad por capas

El alumnado realiza tantas rondas como permita el tiempo durante los 35 minutos del ciclo de mejora. Cada ronda sigue el mismo proceso de cuatro pasos: Escribir → Preguntar → Evaluar → Revisar

1. **Ronda 1 — Utilizar la IA para tareas de apoyo**
   - Escribir: completa de forma autónoma un ejercicio breve de programación.
   - Preguntar: pide apoyo a la IA para una tarea no relacionada con la programación en sí, como:
     - mejorar el sangrado,
     - generar comentarios,
     - explicar un mensaje de error,
     - describir qué hace una función.
   - Evaluar: ¿la explicación de la IA describe con exactitud el código? ¿Los comentarios generados son útiles y comprensibles?
   - Revisar: incorpora los cambios que resulten útiles, asegurándote de que la lógica del programa siga siendo enteramente obra tuya.
2. **Ronda 2 — Pedir sugerencias de mejora**
   - Escribir: revisa el programa que has terminado e identifica las partes que podrían mejorarse.
   - Preguntar: pide sugerencias de mejora sin permitir que la IA reescriba el código. Por ejemplo:

     > «Sugiere tres formas de mejorar la legibilidad y el mantenimiento de este programa, pero no proporciones código de sustitución.»

   - Evaluar: ¿qué sugerencias parecen valiosas? ¿Qué sugerencias tendrían poco impacto en el programa?
   - Revisar: implementa las mejoras con las que estés de acuerdo y comprueba si el programa sigue comportándose correctamente.
2. **Ronda 3 — Evaluar críticamente los consejos de optimización**
   - Escribir: selecciona una sección de tu código que realice una tarea repetitiva o que utilice bucles.
   - Preguntar: pide sugerencias de optimización y exige a la IA que justifique sus recomendaciones.

     > Ejemplo de prompt:
     >
     > «Explica cómo podría optimizarse este código y por qué cada optimización propuesta mejoraría el rendimiento o la legibilidad. No proporciones el código optimizado.»

   - Evaluar: ¿las optimizaciones propuestas son realmente beneficiosas? ¿Cuáles están respaldadas por un razonamiento claro? ¿Cuáles parecen innecesarias o difíciles de justificar?

    - Revisar: decide qué recomendaciones implementar y explica tu razonamiento.

Preguntas de reflexión:

- ¿Qué sugerencias de la IA resultaron más útiles para mejorar tu código?
- ¿La IA dio alguna sugerencia con la que no estuvieras de acuerdo? ¿Por qué?
- ¿Cómo determinaste si una optimización merecía la pena?
- ¿Cuándo resulta más útil la IA durante el proceso de programación?
- ¿Qué tareas de programación debería seguir realizando el alumnado?
- ¿Por qué son necesarios los conocimientos de programación para evaluar la retroalimentación generada por IA?
- ¿Cómo puede el alumnado evitar volverse demasiado dependiente de las herramientas de IA al aprender a programar?

Pregunta de cierre (para la fase de reflexión):

¿Cuántos conocimientos de programación hacen falta para determinar si una sugerencia de la IA mejora realmente un programa, y qué riesgos surgen cuando las personas usuarias se apoyan en consejos generados por IA sin comprender ellas mismas el código?

Materiales

- Un portátil por pareja de alumnado con un entorno de programación
- Acceso a un asistente de chat de IA o a un modelo de lenguaje alojado localmente
- Un ejercicio breve de programación adecuado al nivel de experiencia del alumnado
- Una ficha de trabajo que siga el ciclo escribir → preguntar → evaluar → revisar
- Preguntas de reflexión centradas en la alfabetización en IA y la evaluación crítica
- Ejemplos de código opcionales que ilustren buenas y malas prácticas de programación

---

Revision #3
Created 2026-09-17 20:25:25 UTC by Alexander
Updated 2026-10-29 14:15:19 UTC by Jannik