ISBN: 978-628-96613-3-0. ISSN: 2414-6390. Digital Object Identifier: https://dx.doi.org/10.18687/LACCEI2026.1.1.455

Detection of Non-Formative Use of Generative AI in Computer Science 1: Study Based on Automated Rubric and Individual Oral Assessments

Inés Friss de Kereki
Universidad ORT Uruguay, Uruguay, *kereki i@ort.edu.uy*

Abstract*– The use of generative artificial intelligence (GenAI) tools in introductory computer science courses can support learning, but it also entails the risk that these tools may be used as a substitute for students' own reasoning. This paper presents, as its main contribution, an analysis of the detection of non-formative use of GenAI in a mandatory assignment of a Computer Science 1 course. The course included the use of explicit instructional guidelines, recommendations, and activities for formative GenAI use. The study was exploratory, and was conducted during the second semester of 2025. Throughout the course, activities integrating GenAI were implemented with a focus on formative roles. An automated rubric was designed to identify coding patterns that are not expected in first-semester students. The results of the automated analysis were contrasted with students' performance in oral assessment sessions. The results show that 42% (5 out of 12) of the submitted assignments exhibited strong indications of non-formative GenAI use, evidenced by a mismatch between the characteristics and complexity of the submitted code and the level of understanding demonstrated during the corresponding oral assessment.

Keywords*—Computer Science 1, Artificial Intelligence, Active learning

Detección de uso no formativo de IA generativa en Programación 1: estudio basado en rúbrica automática y defensas orales individuales

Inés Friss de Kereki
Universidad ORT Uruguay, Uruguay, *kereki_i@ort.edu.uy*

Resumen*– El uso de herramientas de inteligencia artificial generativa (IAG) en cursos iniciales de programación puede apoyar el aprendizaje, pero también conlleva el riesgo de que dichas herramientas sean utilizadas como sustituto del razonamiento propio. Este trabajo presenta como principal aporte un análisis de la detección de uso no formativo de IAG en un trabajo obligatorio de un curso de Programación 1. El curso incluyó el uso de guías explícitas, recomendaciones y actividades orientadas al uso formativo de IAG. El estudio fue exploratorio y se desarrolló durante el segundo semestre de 2025. Durante el curso, se integraron actividades que incorporaron el uso de IAG focalizando en roles formativos. Se diseñó una rúbrica automatizada orientada a identificar patrones de codificación no esperables en estudiantes de primer semestre. Los resultados del análisis automático fueron contrastados con el desempeño de los estudiantes en instancias de defensa oral. Los resultados muestran que el 42% (5 de 12) de los trabajos entregados presentó indicios altos de uso no formativo de IAG, evidenciados por una discordancia entre las características y complejidad del código entregado y la comprensión demostrada durante las defensas de esos mismos trabajos.

Palabras clave—Programación 1, Inteligencia artificial, aprendizaje activo.

I. Introducción

Las tecnologías de la información han transformado la educación y, en particular, la inteligencia artificial generativa (IAG) ha abierto nuevas posibilidades para apoyar el aprendizaje [1, 2]. Herramientas como ChatGPT [3] permiten experiencias más personalizadas e interactivas, pero su incorporación en la enseñanza requiere un encuadre responsable desde etapas tempranas de la formación [4].

Aunque la IAG ha sido promovida como una herramienta con alto potencial educativo, la evidencia empírica sobre su impacto en el aprendizaje en cursos iniciales de programación sigue siendo limitada [5, 6]. En particular, se ha señalado el riesgo de dependencia excesiva y reducción del compromiso cognitivo cuando estas herramientas sustituyen el razonamiento propio [7].

La ingeniería de prompts ("prompt engineering"), donde *prompt* se refiere a la instrucción o consigna proporcionada a la inteligencia artificial (IA) y *promptear/prompting* al acto de formularla de manera intencional, es el proceso de estructurar instrucciones que permitan a la IA generar respuestas útiles y controladas [8].

Es importante averiguar como incorporar las herramientas de IA en la pedagogía. Se necesita entender los fundamentos para ser ingenieros de software efectivos. Debe enseñarse el uso responsable de la IA y prevenir el mal uso [8]. Si bien las tecnologías pueden potenciar y apoyar el aprendizaje, introducen el riesgo de que su uso se desplace hacia una sustitución del razonamiento propio [8].

Programación 1 (P1), asignatura del 1er semestre en carreras de Ingeniería en Sistemas y similares, es un curso fundacional, en el que se establecen las bases conceptuales necesarias para abordar algoritmos más complejos en cursos posteriores. En el curso de Programación 1 se propone incorporar herramientas como ChatGPT [3] como parte activa del trabajo en clase, fomentando el uso de la conversación con la IA para pensar, resolver problemas, escribir y revisar código, con el objetivo de que los estudiantes aprendan de forma más dinámica y acompañada, pero sin descuidar el aprendizaje. Implica no sólo saber escribir código, sino saber además cómo pedir ayuda, cómo interpretar lo que sugiere la IA y cómo decidir si una solución es adecuada o no.

Esto promovería un rol más activo y reflexivo por parte del estudiante. El rol del docente es de guía y acompañamiento de este proceso a través de actividades, enseñando a hacer buenas preguntas, a evaluar lo que responde la IA y a aprovechar sus aportes sin depender completamente de ella. Se considera uso formativo cuando la IA es utilizada para comprender el problema, explorar alternativas, revisar errores y reflexionar sobre las decisiones algorítmicas, manteniendo al estudiante como agente activo del proceso. El uso no formativo de IAG sería aquel en el que la herramienta es utilizada como sustituto del razonamiento propio, evidenciado por una falta de correspondencia entre el producto entregado y la comprensión demostrada por el estudiante. Operativamente, se considera indicio de uso no formativo la presencia de construcciones, estilos o decisiones algorítmicas cuya explicación o modificación no puede ser justificada o realizada por el estudiante.

Este trabajo tiene por objetivo colaborar en la detección de uso no formativo de IAG en un curso inicial de Programación, proponiendo un enfoque basado en la triangulación entre formación explícita en uso de IAG, análisis automático de código y defensa individual presencial

(evaluación oral para verificar autoría y comprensión del trabajo).

El artículo está estructurado de la siguiente forma: en la Sección II se describen trabajos relacionados, en la Sección III el curso de Programación 1, sus contenidos y el esquema de evaluación; en la Sección IV se presenta la experimentación realizada, incluyendo la encuesta diagnóstica y el diseño y aplicación de actividades con IAG para contextualizar su uso en el curso; en la Sección V se presenta la evaluación: se detalla la rúbrica automática implementada en AutoGrade [9] y se analizan los resultados obtenidos con las defensas, para finalmente en la Sección VI ofrecer conclusiones y líneas de trabajo futuro.

II. Trabajos Relacionados

A continuación se presentan diversos estudios recientes que analizan el impacto del uso de herramientas de IAG, como ChatGPT, en cursos iniciales de programación, considerando aspectos como su efecto en el desempeño académico, las estrategias de estudio y las prácticas pedagógicas asociadas.

ChatGPT incluye entre sus ventajas la capacidad de proveer aprendizaje adaptativo y personalizado, con la potencial desventaja como el riesgo de confiar en exceso en la tecnología [10]. Se indica en [10] que uno de los desafíos críticos es el potencial para la copia en evaluaciones y para prevenir esto, se sugiere que las instituciones deben incluir políticas y guías de uso, y monitorear la actividad durante las evaluaciones o usar otras medidas anti-plagio [10].

El trabajo de Kosar y colegas [11] analiza el impacto del uso de ChatGPT en un curso universitario inicial de programación orientada a objetos mediante un experimento controlado con 182 estudiantes, divididos en un grupo que utilizó ChatGPT y otro que no. Los resultados muestran que el uso de ChatGPT no tuvo un efecto estadísticamente significativo en el desempeño de los estudiantes, ni en las calificaciones de los trabajos prácticos ni en los exámenes parciales. Los autores concluyen que, bajo ciertas medidas y ajustes pedagógicos, el uso de ChatGPT por parte de programadores novatos puede considerarse viable en el contexto educativo [11].

El estudio de Xue y colegas [12] sobre 56 participantes divididos en un grupo con acceso a GPT y otro sin acceso, concluye que el uso de ChatGPT no tiene un impacto significativo en el desempeño de los estudiantes en tareas típicas de CS1. Además, se observa que los estudiantes que utilizan ChatGPT reducen considerablemente la consulta de otros recursos educativos tradicionales y tienden a depender casi exclusivamente del GPT, sin que ello garantice una mejora en el aprendizaje [12]. Finalmente, la mayoría de los estudiantes mantiene una postura neutral respecto al rol de ChatGPT en cursos introductorios de programación [12].

El artículo de Andleeb y colegas [13] presenta un diseño donde los estudiantes alternaron entre el uso y no uso de ChatGPT en dos tareas de programación en C. Los resultados muestran que el acceso a ChatGPT mejoró significativamente la calidad del código y redujo el tiempo de resolución, aunque los efectos sobre la comprensión conceptual fueron dispares según el tema. Los estudiantes valoraron positivamente la herramienta para depuración y práctica, pero manifestaron preocupaciones sobre su exactitud y su impacto en el desarrollo de habilidades a largo plazo [13]. Como conclusión del trabajo, ChatGPT puede mejorar la eficiencia y la calidad del trabajo en principiantes, pero puede no mejorar uniformemente el entendimiento conceptual [13]. Dicho trabajo aporta evidencia de beneficios operativos, pero también pone de manifiesto que estos no se traducen necesariamente en una comprensión conceptual homogénea.

El estudio de Güner y Er [14] analiza cómo distintas intervenciones didácticas influyen en la interacción entre estudiantes y ChatGPT en el aprendizaje de programación. A lo largo de tres sesiones, los estudiantes utilizaron ChatGPT con niveles crecientes de guía: sin orientación, con entrenamiento en promptear y con una guía de laboratorio con ejemplos de prompts [14]. El entrenamiento en promptear tuvo un impacto positivo en la calidad de las interacciones con ChatGPT. El estudio destaca la importancia de orientar explícitamente el uso de la IA para favorecer un uso más significativo y efectivo [14]. Estos resultados respaldan la importancia de un encuadre pedagógico explícito y guiado, coherente con el enfoque formativo.

En resumen, los trabajos revisados refuerzan la pertinencia del enfoque propuesto para el curso de Programación 1, donde la incorporación de ChatGPT se concibe como una herramienta de apoyo al razonamiento y no como su sustituto. La evidencia empírica muestra que el uso de ChatGPT no garantiza mejoras automáticas en el aprendizaje [11, 12], que puede inducir dependencias y cambios en las estrategias de estudio [12], y que sus beneficios se concentran principalmente en aspectos operativos como la calidad del código o la eficiencia, con efectos desiguales sobre la comprensión conceptual [13]. Asimismo, se destaca la importancia del rol docente y del aprendizaje de habilidades de interacción con la IA para promover un uso más significativo [14]. En este contexto, la distinción entre uso formativo y no formativo de la IAG resultaría central, y pondría de manifiesto la necesidad de contar con mecanismos que permitan detectar situaciones en las que el producto entregado no se corresponde con la comprensión del estudiante, objetivo al que este trabajo busca contribuir.

III. Curso de Programacion 1

El curso de P1 tiene por objetivos fomentar el pensamiento computacional y desarrollar las habilidades básicas de resolución de problemas orientadas al diseño de algoritmos en un lenguaje de programación. El enfoque del curso es predominantemente estructurado, basado en programación imperativa e incluye una introducción a conceptos básicos de programación orientada a objetos. Los temas fundamentales incluyen: pensamiento computacional, variables, estructuras de control, funciones y procedimientos,

estructuras de datos básicas y nociones de objetos. El lenguaje es JavaScript y se incluyen nociones de HTML y CSS. Dura 15 semanas, con 4 horas de teórico y 2 horas en laboratorio por cada semana.

El curso es presencial y sigue el formato invertido: previo a cada clase los estudiantes acceden a materiales introductorios y en la siguiente clase se trabaja en forma activa sobre esos contenidos. Los docentes del curso cuentan con amplia experiencia.

La evaluación consiste en dos parciales individuales en papel (con 15 y 40 puntos) y dos trabajos obligatorios de larga duración en equipos de dos personas. El primer trabajo obligatorio es una página web sencilla, con estilos predefinidos y el segundo obligatorio es continuación del primero e implica agregarle toda la funcionalidad requerida en JavaScript a esa página. Los puntajes son 10 y 35 puntos respectivamente. Para aprobar el curso se requiere 70 o más puntos de los 100 disponibles, además de superar un mínimo preestablecido en cada evaluación.

El trabajo obligatorio del curso de agosto-noviembre de 2025 incluyó el desarrollo de una página local para registro de jugadores, poder jugar a 2 juegos diferentes (uno de calcular sumas generadas al azar y el otro de presionar pares similares en una grilla dinámica), registrar comentarios y editarlos posteriormente, y cargar en la página información en formato de lista con los jugadores que no han jugado, jugadores con la máxima cantidad de comentarios, tablas con el detalle de los comentarios ordenados por hora y búsqueda de textos en los comentarios. Estas funcionalidades requieren la aplicación de lógica básica, estructuras de control y la manipulación simple de los elementos de la página web, a través de la interacción con el DOM (Document Object Model).

Para la corrección de todas las pruebas se usan rúbricas unificadas, que son de conocimiento de los estudiantes. La rúbrica incluye aspectos como el diseño de las clases, formato y estilo de la página y cumplimiento de los diferentes requisitos. El 2do obligatorio tiene además una defensa individual presencial en el laboratorio cuyo formato de documentación se ve en la Fig. 1. La defensa consiste en descargar el obligatorio del repositorio institucional, instalarlo en la máquina del laboratorio, realizar un cambio propuesto en el código en un tiempo máximo de 30 minutos (sin utilizar ningún material de apoyo) y luego responder preguntas orales sobre el código entregado. En cursos previos, la defensa no incluía preguntas y duraba 60 minutos, pero se observó la necesidad de agregar las preguntas para verificar la comprensión y conocimiento del código y también reducir el tiempo de codificación del cambio a media hora, pues se había notado que, en los casos exitosos, con pocos minutos para codificar alcanzaba y los casos no exitosos, las dificultades persistían aún con mayor disponibilidad de tiempo.

P1 DEFENSA PROGRAMACION

Estudiante:

La resolución debe ser realizada exclusivamente por el propio estudiante en el tiempo previsto, SIN AYUDA de ningún tipo. La resolución exitosa del cambio y consultas dentro del tiempo implicará que la defensa es correcta. Cualquier otra situación será analizada e implicará pérdida total o parcial de puntos del obligatorio.

Fecha / / 2025 Hora: Comienzo: Finalización: **Tiempo Máximo: 30 minutos**

Instaló correctamente el sistema en el ambiente: **SI - NO** Cantidad de pruebas erróneas: **0 –1 –2 –3-más**

Cambio solicitado:

Resultado Implementación del cambio solicitado:

Preguntas sobre el código: ubica código requerido	**SI NO**
Explicación de código seleccionado:	**BIEN PARCIAL MAL**

Firmas

Docente

Alumno (firma y aclaración)

Fig. 1 Defensa individual del trabajo obligatorio (formato de documentación)

Es sumamente útil para el docente tener una revisión previa del trabajo antes de la defensa, pues orienta los cambios a pedir y preguntas a realizar. Muchas veces no es posible realizar esa revisión previa porque la defensa es muy próxima a la entrega y no hay tiempo para prepararla en detalle, y, o, la cantidad de grupos a revisar es grande para el tiempo disponible previo a la defensa. En esos casos, se tienen preparados un conjunto de posibles cambios, y en el caso que el estudiante avise que no hizo esa parte del obligatorio, se sustituye el cambio por otro sin penalización, por ejemplo si el cambio implica ordenar los jugadores por otro criterio y el estudiante indica que en su obligatorio no implementó la ordenación, se realiza la sustitución.

Los cambios para realizar en el código son incluir funcionalidades parecidas al obligatorio, por ejemplo, agregar una lista con los jugadores que tienen más de 3 juegos ganados, o permitir ingresar a jugar solamente si hay 2 o más jugadores mayores a 18 años registrados. Generalmente implican lógica sencilla, con cambios menores en la página (agregando por ejemplo un párrafo o una lista nueva), obtener los datos necesarios y cargar lo obtenido en la página. Cuando el estudiante tiene pronto el cambio, avisa al docente, quien lo revisa y prueba el código. De encontrar algún error, lo marca en la hoja respectiva y el estudiante puede corregirlo. Esta parte permite observar el manejo de las estructuras de control y variables, la comprensión general de la solución implementada, y la capacidad de depuración y resolución de errores en el propio ambiente de codificación.

Como continuación de la defensa, se hacen preguntas para detectar autoría e indagar sobre el razonamiento realizado, incluyendo la justificación de decisiones de diseño, la explicación del funcionamiento de ciertas partes del programa y la identificación de posibles errores o limitaciones.

El uso de IAG permite, en muchos casos, obtener código correcto y bien estructurado; sin embargo, el estudiante podría no haber internalizado las opciones tomadas ni desarrollado el razonamiento algorítmico involucrado en la solución. Se trata de verificar que el trabajo es realizado y comprendido por el

propio estudiante, por lo que una defensa exitosa implica que el estudiante comprende el problema, entiende su solución y puede razonar sobre decisiones y cambios y realizarlos. Valida no sólo el resultado sino el proceso.

En caso de defensa exitosa, se obtienen los puntos correspondientes al obligatorio. En caso de problemas, se quitan puntos, eventualmente todos, según la magnitud de los errores o inconsistencias detectadas. Al concluir la defensa, el resultado es documentado y firmado formalmente por el docente y el estudiante.

IV. Experimentacion

Se seleccionó un grupo de estudiantes al azar en el curso de agosto-noviembre de 2025. Inicialmente, el grupo tenía 28 alumnos. Se realizó una encuesta anónima al comienzo del curso. La experimentación consistió en diseñar y aplicar actividades con el uso de IA, con el objetivo de entrenar a los estudiantes en habilidades de prompting y en un uso significativo de la IA, como sugiere [14], y posteriormente analizar su desempeño en instancias de evaluación. Se describe a continuación la encuesta y actividades.

A. Encuesta inicial

A comienzos del semestre se realizó una encuesta diagnóstica anónima, donde se obtuvieron 16 respuestas.

La mayoría de los estudiantes que respondieron indicaron que ya habían cursado previamente la asignatura (68.8%, 11 estudiantes) y solamente uno indicó no tener conocimientos previos de programación. La mayoría de los encuestados ya utilizaban IA, principalmente ChatGPT (93.8%, 15 alumnos).

Ante la consulta: "¿Te sientes capaz de evaluar si lo que te da la IA está bien o mal?", la mayoría responde que "Sí" (62.5%, 10 estudiantes) o "Más o menos" (37.5%, 6 estudiantes) (ver Fig. 2). Se les consultó sobre: "¿Sabes cómo escribir un prompt para una IA?", 62.5% (10 estudiantes) indican "más o menos", 25% (4 estudiantes) indican "Sí" y 12.5% (2 estudiantes) refieren "No" (ver Fig. 3). Estos resultados sugieren una brecha entre la confianza percibida y las competencias reales para evaluar críticamente las respuestas de la IA, lo que podría limitar una validación efectiva de sus resultados.

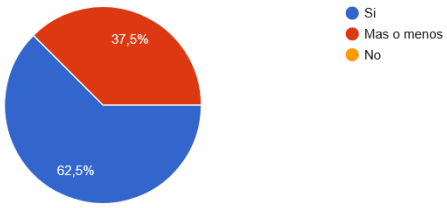

Fig. 2 ¿Te sientes capaz de evaluar si lo que da la IA es correcto?

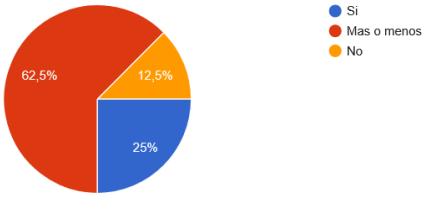

Fig. 3 ¿Sabes cómo escribir un prompt claro y útil para una IA?

Se les consultó: "¿Recomendarías el uso de IA en P1?", ningún estudiante seleccionó opciones de rechazo explícito al uso de IA en el curso (ver Fig. 4). Las respuestas fueron: "Sí, seguro" (31.3%, 5 estudiantes), "Si, puede ser" (56.3%, 9 estudiantes) y quizás (12.5%, 2 estudiantes). Muestra un alto grado de aceptación a su uso.

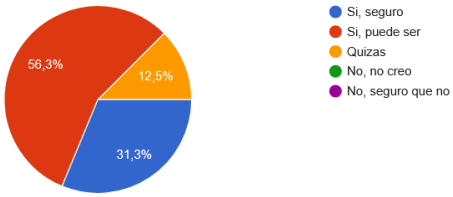

Fig. 4 ¿Recomendarías el uso de IA en P1?

En relación con "¿Qué rol imaginas usar la IA en el curso?, las opciones de selección múltiple más indicadas fueron:

- "Asistente de calidad: para que revise mi código, me diga si hay errores y me explique cómo arreglarlos" (62.5%, 10 alumnos)
- "Evaluador: para que me diga si está bien, me de casos de prueba" (56.3%, 9 alumnos).
- "Tutor: para que me explique conceptos o me enseñe a usar algo" (37.5%, 6 alumnos)

Estos resultados muestran que los estudiantes conciben mayoritariamente a la IA como una herramienta de apoyo al proceso de aprendizaje, en roles coherentes con un uso formativo, más que como un sustituto directo de la resolución de problemas.

En síntesis, la encuesta diagnóstica muestra una alta familiaridad y aceptación del uso de IAG en P1. Si bien la mayoría de los estudiantes declara confianza para evaluar las respuestas de la IA, también reconoce limitaciones en sus habilidades de prompting, lo que refuerza la existencia de una brecha entre percepción y validación crítica efectiva. La IA es concebida principalmente como una herramienta de apoyo: en roles de asistente, evaluador o tutor. Dado el tamaño reducido de la muestra y el carácter voluntario de la encuesta, estos resultados se interpretan como indicios exploratorios.

B. Diseño y uso de actividades

Se diseñaron actividades específicas orientadas a fomentar un uso formativo de la IA. Si bien estaban previstas

para todo el curso, la asistencia no obligatoria y la baja concurrencia a clases llevaron a concentrar su implementación principalmente en las primeras cinco semanas del semestre, etapa clave para la formación de hábitos de razonamiento en cursos iniciales de programación.

Algunas de las principales desarrolladas en esas semanas del curso fueron:

a) Estructuras de control: Luego de la explicación de estructuras de control, se hizo en clase el ejercicio de dibujar una carita sonriente a partir de instrucciones orales difusas. Después, utilizando el GPT propio del curso ProfeP1Chat [15] se hizo ejercicio similar: "Dame las instrucciones para dibujar algo sencillo pero sin decirme que es, yo adivino". Hicieron varios experimentos, algunos estudiantes adivinaron. Luego se hizo el ejercicio al revés: "yo te daré instrucciones y vos dibujas con caracteres ASCII". El dibujo a explicar por el alumno a la IA era un cohete simple, formado por un cuadrado en la base con un círculo dentro, un triángulo en la parte superior y otros dos en la base. Observando lo realizado por los alumnos, uno indicó que "Dibujó mal los triángulos", otros dijeron frases cómo: "la IA interpreta mal los triángulos", pero no siguieron haciendo prompting. Unos pocos, luego de varias interacciones, lograron un dibujo bastante similar. Dos llegaron a algo parecido, pero abandonaron, tomaron por bueno ese dibujo parcial preliminar. Se aprovechó este contexto de ejercicio para explicar en detalle cómo interactuar y cómo preguntar, resaltando el proceso completo.

La versión utilizada por ProfeP1Chat de ChatGPT es la 5. Una limitación práctica fue la disponibilidad restringida de acceso gratuito a ella. Se exploraron alternativas como Claude [16] y Gemini [17], pero sus estilos de respuesta diferían significativamente de los contenidos y convenciones del curso, lo que podía generar confusión. Por este motivo, se decidió limitar el uso de IA a un ejercicio por clase y reforzar el trabajo manual.

b) Seudocódigo: El prompt fue: "Dame ejercicios para hacer yo en pseudocódigo, ya hice el de bañar al perro, salir de casa y pintar un cuarto". La IA ofreció ejercicios cómo cepillarse los dientes, anotarse en la biblioteca y, o, hacer jugo de sobrecito. Participaron 10 alumnos que hicieron la actividad en forma adecuada. Esta actividad permitió utilizar la IA en un rol de tutor generador de consignas, manteniendo al estudiante como responsable de la resolución y discusión del algoritmo.

c) Expresiones aritméticas y lógicas. Con el objetivo de fomentar la depuración, el prompt fue: "Dame ejercicios con expresiones lógicas o aritméticas sencillas y yo encuentro el error". La IA ofreció varios ejercicios en rol de tutor, que los estudiantes resolvieron.

d) Algoritmia sencilla. Se hizo en clase, en forma individual, el ejercicio de cálculo del máximo de un conjunto de datos y cálculo del 2do máximo, en JavaScript. El docente pidió al GPT [15] que genere solución sencilla, que fue discutida en grupo. Se compararon detalladamente las soluciones con las vistas en clase, incluyendo elementos no vistos en el curso. Esta comparación permitió evidenciar diferencias entre soluciones correctas desde el punto de vista funcional y soluciones alineadas con los contenidos y estilos esperados en el curso. Hubo alta participación en la discusión.

e) Strings. Al presentar strings en JavaScript y luego de algunos ejercicios, se pidió que se usara ProfeP1Chat [15] para generar una tabla de datos de prueba de algunos ejercicios específicos vistos en clase. También se indicó analizar la formulación de los prompts, dado que las respuestas iniciales evidenciaban desalineaciones entre lo solicitado y lo esperado. Se discutió que siempre deben revisar las salidas de la IA.

En resumen, en las distintas actividades del curso, la IA fue utilizada de manera formativa con roles diferenciados que mantuvieron al estudiante como agente activo del aprendizaje. Se empleó como generador de consignas y ejercicios (por ejemplo, en seudocódigo y expresiones), como interlocutor para analizar la precisión de instrucciones y la necesidad de un prompting adecuado (estructuras de control y ASCII), y como recurso de contraste para discutir soluciones algorítmicas en forma crítica (máximo y segundo máximo). Asimismo, en el trabajo con strings, la IA permitió evidenciar malentendidos y promover la verificación y validación de resultados. En todos los casos, la herramienta fue utilizada para fomentar el razonamiento, la depuración, la reflexión sobre decisiones y la revisión crítica de resultados, evitando su uso como generadora directa de soluciones finales sin análisis previo. Este marco de actividades permitió establecer expectativas claras sobre un uso formativo de la IA, que luego sirvieron como referencia para identificar desalineaciones entre el código entregado y la comprensión demostrada durante las evaluaciones.

V. Resultados

Para la evaluación del segundo trabajo obligatorio se consideraron conjuntamente la rúbrica de la cátedra, la rúbrica automática implementada en AutoGrade [9] y el desempeño en la defensa individual presencial.

Para detectar con antelación a la defensa un posible uso excesivo de IA (o para detectar plagio, como refiere [10]) y como apoyo al docente, se definió en AutoGrade [9] una rúbrica para tratar de verificar si el programa incluye mayoritariamente elementos avanzados o no. No se trata de analizar la corrección del ejercicio (objetivo original de AutoGrade [9]), sino de identificar patrones de estilo y estructuras de codificación no esperables en estudiantes de primer semestre.

Los docentes hicieron múltiples generaciones de soluciones con IA, donde aparecieron frecuentemente elementos no esperables en el curso, como por ejemplo, uso de funciones y métodos avanzados del lenguaje, enfoques y estilos de programación funcional, convenciones y estilos de documentación y patrones de interacción con el DOM no trabajados en clase.

Así, la rúbrica utilizada en AutoGrade incluyó estos elementos indicadores o criterios equivalentes:

1) Si usa map
2) Si usa filter
3) Si usa forEach
4) Si usa find
5) Si usa reduce o some
6) Si usa arrow functions
7) Si uso de import/export, async/await o regex
8) Tiene múltiples puntos de return en funciones
9) Si usa break y continue en bucles para salidas anticipadas
10) Uso de const y, o var (en el curso se indica let)
11) Usa operadores ternarios
12) Uso de template strings
13) Los comentarios son en inglés, y, o, las funciones tienen nombres genéricos (ejemplos: "handleSubmit", "renderList")
14) Los comentarios tienen formato muy estructurado o sobredocumentado estilo JSDoc
15) Usa cualquiera de: destructuring, spread (...), Array.from, Promises
16) El estilo es diferente en zonas, algunas áreas del código parecen hechas por estudiante de primero y otras son muy profesionales, o la estructura es "demasiado prolija" tipo plantilla (banners, TODOs, ESLint).
17) Mezcla nombres de variables en español y en inglés
18) Usa métodos como firstchild, firstElementChild / lastElementChild, childNodes / children nextSibling / previousSibling, nextElementSibling / previousElementSibling, parentElement / parentNode, hasChildNodes() o similares para interactuar con el DOM
19) Usa QuerySelectorAll o similar
20) Usa submit y, o, preventDefault

Los criterios incluidos fueron revisados por tres docentes, con el objetivo de ajustar su pertinencia respecto a los contenidos y estilos esperados en el curso. Los criterios se consideran equivalentes en tanto apuntan a identificar el uso de elementos que no forman parte de los contenidos esperados ni del estilo de resolución habitual de estudiantes de primer semestre en el curso, independientemente de la función específica utilizada. No se penaliza el uso de estos elementos en sí mismo; sin embargo, su utilización requiere que el estudiante pueda explicarlos, justificarlos y reproducir razonamientos equivalentes durante la defensa. AutoGrade no reemplaza el juicio docente, sino que actúa en forma temprana como herramienta de apoyo para focalizar la defensa y orientar la indagación.

Hubo 12 entregas del 2do. obligatorio que corresponden a 15 alumnos, los demás no entregaron. Las entregas 1, 4 y 5 son de equipos de 2 estudiantes, las demás fueron individuales. Se aplicó AutoGrade y se detectaron 5 trabajos con valores altos de probable uso de IA. En la Fig. 5, se muestra por cada renglón el resultado de cada entrega, cada columna corresponde a un criterio y en cada posición el valor "1" representa cumplir con el criterio. El porcentaje se calcula considerando la cantidad de "1" entre los 20 criterios. Hubo 5 casos con alto porcentaje de coincidencia: uno con 45% y los otros 4 con 70% o más, lo que representa el 42% (5 de 12) de los trabajos entregados. El análisis mostró que los cinco casos con valores elevados en la rúbrica correspondieron a estudiantes que toman el curso nuevamente, lo que sugiere una concentración de casos problemáticos en estudiantes recursantes.

	Elementos																				%
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
Ent. 1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	5%
Ent. 2	1	1	1	1	1	1	1	1	0	1	1	1	0	0	1	0	1	0	1	0	70%
Ent. 3	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	10%
Ent. 4	1	1	1	1	1	1	1	1	0	1	1	1	1	0	0	0	1	0	1	1	75%
Ent. 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	5%
Ent. 6	0	0	1	0	0	1	1	1	0	1	0	1	0	0	1	0	0	0	1	1	45%
Ent. 7	1	1	1	1	1	1	1	1	0	1	1	1	0	0	1	0	1	0	1	1	75%
Ent. 8	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	10%
Ent. 9	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	1	0	20%
Ent. 10	0	1	1	1	1	1	1	1	0	1	1	1	0	0	1	0	1	0	1	1	70%
Ent. 11	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	10%
Ent. 12	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	10%

Fig. 5 Resultado de AutoGrade

Las defensas de esos casos fueron insuficientes:

- el estudiante de la entrega 2 (con 70% en AutoGrade) no asistió;
- de la entrega 4 (75% en AutoGrade), un estudiante puso unas pocas líneas de código erróneas y las explicaciones fueron insuficientes y el otro estudiante pudo hacer el cambio, pero no pudo explicar nada;
- el estudiante de la entrega 6 (45% en AutoGrade) pudo hacer el cambio con algún error, excediéndose del tiempo pero no puedo responder ni justificar los elementos del resto del código; y
- tanto el estudiante de la entrega 7 (75% en AutoGrade) como el de la entrega 10 (70% en AutoGrade) solamente pudieron poner un párrafo o lista en HTML, pero nada del código JavaScript ni dar explicaciones.

En todos los casos, las dificultades observadas evidenciaron una falta de correspondencia entre la complejidad del código entregado y la capacidad del estudiante para explicarlo, modificarlo o razonar sobre él.

No se detectaron casos falsos positivos: todos los casos con valores altos en AutoGrade presentaron defensas insuficientes. Los restantes casos (con valores bajos de AutoGrade) tuvieron defensas razonables, haciendo los cambios y respondiendo las preguntas, excepto en un caso (entrega 8), que tuvo valor bajo de AutoGrade y el estudiante no pudo hacer nada del cambio ni explicar. Aclaró que tuvo ayuda extra no apropiada de otros estudiantes (ver resumen de los casos en Tabla 1).

TABLA I
RESUMEN DE ENTREGAS, AUTOGRADE Y DEFENSAS

Entrega	% AutoGrade	Defensa
1 – estudiante 1	5%	suficiente
1 – estudiante 2	5%	suficiente
2 – estudiante 3	70%	no asistió
3 – estudiante 4	10%	suficiente
4 – estudiante 5	75%	insuficiente
4 – estudiante 6	75%	insuficiente
5 – estudiante 7	5%	suficiente
5 - estudiante 8	5%	suficiente
6 – estudiante 9	45%	insuficiente
7 – estudiante 10	75%	insuficiente
8 – estudiante 11	10%	insuficiente
9 – estudiante 12	20%	suficiente
10 – estudiante 13	70%	insuficiente
11 – estudiante 14	10%	suficiente
12 – estudiante 15	10%	suficiente

Tradicionalmente se usa Moss [18] en los cursos para detectar copias de trabajos y en general se detectaban varios casos. Este curso también se usó dicha herramienta pero no se identificaron similitudes, es decir, no se encontraron trabajos que compartieran código. Esto refuerza la idea de que las dificultades detectadas en las defensas no se vincularon a similitudes entre entregas, ya sea por trabajos colectivos o posibles copias entre estudiantes, sino a un cambio en la forma de producción de código compatible con el uso de herramientas de IA.

En resumen, se observó una concordancia general entre AutoGrade y las defensas presenciales: los casos con valores altos en la rúbrica tendieron a presentar defensas insuficientes, mientras que la amplia mayoría de los estudiantes con valores bajos pudo realizar los cambios y justificar el código, con una sola excepción. Esto respalda al uso de AutoGrade como herramienta de apoyo al criterio docente, cuya validez se ve reforzada por su concordancia con las defensas individuales presenciales, permitiendo además gestionar de forma eficiente grandes cantidades de trabajos.

VI. Conclusiones y Trabajo Futuro

Durante el curso de Programación 1 se incorporaron actividades que integraron la IAG con un enfoque formativo, orientadas a explicitar usos esperados de esta herramienta y a proporcionar un marco de referencia para su utilización. En este trabajo exploratorio se analizó la detección de uso no formativo de IAG en el curso, triangulando la aplicación de dichas actividades orientadas a formar en el uso de IA, con rúbrica automática de análisis de código (que permite detectar tempranamente posibles casos problemáticos) y defensas individuales presenciales.

Los resultados muestran que una proporción significativa de los trabajos entregados (5 de 12, 42%) presentó diferencias entre la complejidad del código y la comprensión demostrada por los estudiantes durante las defensas individuales, lo que constituye un indicio de uso no formativo de la IAG. Estos hallazgos evidencian que, aún en presencia de actividades orientadas al uso formativo, pueden darse situaciones en las que la IAG sustituye el razonamiento propio. Desde una perspectiva pedagógica, también podrían tenerse en cuenta factores tales como la presión académica, falta de confianza y, o, ciertos hábitos de estudio.

Los resultados sugieren que la incorporación de la IAG con un enfoque formativo, si bien es necesaria, no es suficiente por sí sola para garantizar un uso que favorezca el razonamiento propio en cursos iniciales de programación. En este contexto, las defensas individuales son un elemento clave para evidenciar la autoría y la comprensión, permitiendo detectar usos no formativos de la IAG.

Como limitaciones al estudio, se tiene en cuenta el reducido tamaño de la muestra, la baja asistencia a clases y la alta proporción de estudiantes que retoman el curso. Estas condiciones podrían influir tanto en los patrones de uso de la IAG como en las percepciones y desempeños observados. En este marco, los resultados se presentan como evidencia exploratoria y descriptiva.

Como líneas de trabajo futuro, se propone ampliar el conjunto de actividades, el tamaño de la muestra y el análisis estadístico. Asimismo, podría ser de interés profundizar en la comparación entre estudiantes recursantes y noveles en el curso, así como entre estudiantes con experiencia en IA y sin experiencia en IA, y considerar marcos teóricos que enriquezcan el análisis de la relación de los estudiantes con el uso de la IA. También se propone incorporar instancias de defensa simuladas durante el curso, como estrategia preventiva, orientadas a que los estudiantes practiquen la explicación, justificación y modificación de sus propias soluciones en un contexto y ambiente más distendido, así como promover la concientización sobre el uso responsable de herramientas de IA. Estas instancias permitirían anticipar dificultades en la comprensión del proceso de resolución y analizar su impacto en el uso de la IAG, sin carácter evaluativo. Además se propone incorporar instancias de autorreporte donde los estudiantes reflexionen sobre cómo usaron la IA, con qué objetivo y en qué momentos del proceso, contrastando estas percepciones con los resultados observados. Otras posibles líneas son: analizar la adaptación y validación de la rúbrica para otros cursos y, o, lenguajes de programación, y discutir el aporte del trabajo a la mejora curricular en relación con el diseño de evaluaciones para planes de estudio.

Agradecimiento

Este proyecto fue seleccionado y recibe financiamiento en el marco del 2do. llamado del "Fondo de Innovación en Proyectos de Enseñanza utilizando Inteligencia Artificial" de la Universidad ORT Uruguay.

Referencias

[1] R. Sun y X. Deng, “Using Generative AI to Enhance Experiential Learning: An Exploratory Study of ChatGPT Use by University

Students", *Journal of Information Systems Education*, vol. 36, no. 1, pp. 53–64, Winter 2025, https://doi.org/10.62273/ZLUM4022.
[2] P. N. Cortez Herrera, "ChatGPT y la docencia: oportunidades y desafíos", *Latam Revista Latinoamericana de Ciencias Sociales y Humanidades*, vol. 6, Marzo 2025. https://doi.org/10.56712/latam.v6i2.3792.
[3] ChatGPT, https://chatgpt.com.
[4] P. Denny, J. Prather, B. Becker, J. Finnie-Ansley, A. Hellas, J. Leinonen, A. Luxton-Reilly, B. Reeves, E. Santos y S. Sarsa, "Computing Education in the Era of Generative AI", *Communications of the ACM*, Vol 67, No. 2, Feb. 2024.
[5] J. Alvarez, C. Hernández, M. Benítez y S. Ojeda-Ramírez, "La inteligencia más allá de lo artificial: un enfoque sociocultural para la educación en ingeniería y computación". *Rev. Latinoamericana de Educación*, vol. 16, n.2, 2025
[6] J. Prather, B. Reeves, J. Leinonen, S. Macneil, A. Randrianasolo, B. Becker, B. Kimmel, J. Wright y B. Briggs, "The Widening Gap: the Benefits and Harms of Generative IA for Novice Programmers", en: Proc. of the 2024 ACM Conference on Int. Computing Education Research, vol 1, pp 469 – 486, 2024, https://doi.org/10.1145/3632620.3671116
[7] H. Lee, A. Sarkar, L. Tankelevitch, I. Drosos, S. Rintel, R. Banks y N. Wilson, "The impact of generative AI on Critical Thinking: self-reported reductions in cognitive effort and confidence effects from a survey of knowledge workers", en: Proc. of the 2025 CHI Conference on Human Factors in Computing Systems, Japón, 2025, https://doi.org/10.1145/3706598.3713778.
[8] E. Shein, "The impact of AI on Computer Science Education", *Communications of the ACM*, Vol 67, No 8, Set. 2024.
[9] I. F. de Kereki and I. Garrido, "AutoGrade: an AI-Based Assessment Tool for Computer Science 1", en: Proc. of 2025 IEEE Engineering Education World Conference (EDUNINE), Uruguay, 2025, https://doi.org/ 10.1109/EDUNINE62377.2025.10980846.
[10] A. Dwinggo, X. Zhai, K. Aoki, L. Bojic y S. Zikic, "An In-depth Review of ChatGPT's pros and cons for learning and teaching in Education", iJIM *Int. Journal of Interactive Mobile Technologies*, vol 18, No. 2, 2024, https:/doi.org/10.3991/ijim.v18i02.46509.
[11] T. Kosar, D. Ostojic, Y. Liu y M. Mernik, "Computer Science Education in ChatGPT Era: Experiences from an Experiment in a Programming Course for Novice Programmers", *Mathematics 2024*, vol 12. https://doi.org/10.3390/math12050629.
[12] Y. Xue, H. Chen, G. Bai, R. Tairas y Y. Huang, "Does ChatGPT Help with Introductory Programming? An Experiment of Students Using ChatGPT in CS1", en: Proc of 2024 IEEE/ACM 46th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET), 2024.
[13] S. Andleeb, B. Kantorski y J. Carver, "ChatGPT in Introductory Programming: Counterbalanced Evaluation of Code Quality, Conceptual Learning, and Student Perceptions", en: ACM SIGCITE'25, Sacramento, California, USA, 2025.
[14] H. Güner y E. Er, "AI in the classroom: Exploring students' interaction with ChatGPT in programming learning", *Education and Information Technologies*, vol 30, pp 12681–12707, 2025, https://doi.org/10.1007/s10639-025-13337-7.
[15] ProfeP1Chat, https://chatgpt.com/g/g-oswWk4aJB-profep1chat.
[16] Claude, https://claude.ai.
[17] Gemini, https://gemini.google.com.
[18] MOSS, https://theory.stanford.edu/~aiken/moss/.