

Aprendizaje y Enseñanza de Programación: El desafío de herramientas de Inteligencia Artificial como ChatGPT

Laura De Giusti^{1,3}, Gonzalo Lujan Villarreal^{3,4},
Eduardo Ibañez¹, Armando De Giusti^{1,2}

¹ Instituto de Investigación en Informática LIDI (III-LIDI),
Facultad de Informática, Universidad Nacional de La Plata –

Comisión de Investigaciones Científicas de la Provincia de Buenos Aires

² CONICET – Consejo Nacional de Investigaciones Científicas y Técnicas

³ CICPBA – Comisión de Investigaciones Científicas de la Provincia de Buenos Aires

⁴ PREBI – SEDICI – Universidad Nacional de La Plata

{ldgiusti, eibanez, degiusti}@lidi.info.unlp.edu.ar; gonzalo@prebi.unlp.edu.ar

Abstract. En este trabajo se analiza el empleo de herramientas de IA y en particular ChatGPT en la enseñanza y el aprendizaje inicial de programación a nivel universitario. Se discuten cuestiones conceptuales y se sintetiza el análisis a soluciones con ChatGPT de enunciados clásicos que resuelven los alumnos en los cursos de primer año, en un lenguaje imperativo, otro orientado a objetos y un tercero que permite expresar algoritmos concurrentes. El análisis crítico del código generado permite reflexionar sobre la utilidad actual de estas herramientas en el ámbito universitario, las posibilidades que ofrecen y algunos desafíos que plantean, tanto para los alumnos como para los docentes de asignaturas relacionadas con Programación.

Keywords: ChatGPT, Enseñanza inicial, Paradigmas.

1. Introducción

Es interesante iniciar este artículo con la respuesta que da ChatGPT a la pregunta: Podremos reemplazar los programadores utilizando herramientas de Inteligencia Artificial como chatGPT? La respuesta se sintetiza a continuación [1]:

“Es poco probable que las herramientas de Inteligencia Artificial, como ChatGPT, reemplacen por completo a los programadores en un futuro cercano. Aunque los avances en la IA han sido significativos en los últimos años, todavía existen ciertas limitaciones y desafíos que impiden que estas herramientas sustituyan completamente la labor humana en el desarrollo de software.

ChatGPT y otras aplicaciones similares son impresionantes en su capacidad para generar texto coherente y responder preguntas, pero su enfoque actual es más útil para tareas específicas, como proporcionar información, ayudar en la redacción de contenido o incluso asistir en algunas partes del proceso de desarrollo de software.

Sin embargo, hay varias razones por las cuales es poco probable que estas herramientas reemplacen a los programadores:

1. Complejidad de la programación: La programación es una tarea compleja que implica la resolución de problemas, la toma de decisiones y el diseño de algoritmos. Aunque las herramientas de IA pueden ayudar en ciertos aspectos, la

- naturaleza creativa y lógica de la programación requiere habilidades humanas.
2. Creatividad e innovación: La programación no se trata solo de seguir instrucciones, sino también de ser creativo e innovador para encontrar soluciones óptimas y eficientes. La creatividad humana es difícil de replicar completamente mediante algoritmos de IA.
 3. Interpretación del contexto y la intención: Los programadores no solo deben escribir código, sino también entender el contexto y las intenciones detrás de los requisitos del proyecto. La interpretación y la comunicación efectiva con los clientes o usuarios son tareas en las que la inteligencia humana es esencial.
 4. Errores y depuración: La depuración de código y la corrección de errores son aspectos fundamentales del desarrollo de software. La IA podría ayudar en algunas partes, pero identificar y solucionar problemas complejos a menudo requiere habilidades humanas y conocimientos técnicos profundos.
 5. Ética y responsabilidad: La programación a menudo involucra tomar decisiones éticas y comprender las implicaciones sociales y legales de las soluciones desarrolladas. Estas cuestiones requieren una comprensión humana más profunda y una consideración consciente.

En lugar de reemplazar a los programadores, las herramientas de IA y la automatización están más orientadas a complementar su trabajo. Pueden ayudar a acelerar ciertas tareas, mejorar la productividad y proporcionar asistencia en el proceso de desarrollo. Sin embargo, la programación seguirá siendo una disciplina en la que la intervención humana y el juicio crítico serán fundamentales durante mucho tiempo.”

Esta respuesta es coherente con algunos conceptos que se conocen:

- ChatGPT es una aplicación de chatbot de inteligencia artificial desarrollado en 2022 por OpenAI que se especializa en el diálogo. Se trata de un gran modelo de lenguaje, con más de 150 millones de parámetros y entrenado con un gran volumen de datos (en particular texto) que se ha ajustado con técnicas de aprendizaje tanto supervisadas como de refuerzo. Ha sido pensado inicialmente para realizar tareas relacionadas con el lenguaje, desde la traducción hasta la generación de texto. [2,3,4].
- Desde el punto de vista de la formación de alumnos, en particular en temas de Informática, es importante remarcar la diferencia entre “formarse para resolver problemas utilizando programación” y “poder generar código en un lenguaje de programación”. Naturalmente en las asignaturas iniciales de las carreras de Informática buscamos el primer objetivo [5,6], que consiste en desarrollar aptitudes tales como la modelización de problemas del mundo real, la abstracción en enunciados de los mismos para convertirlos en algoritmos ejecutables en computadora, así como la expresión de soluciones en diferentes lenguajes de programación. Claramente las herramientas de IA como ChatGPT pueden cumplir más adecuadamente con el segundo punto, ya que dado un enunciado especificado correctamente y un lenguaje de programación que esté en su base de conocimiento, es muy posible que genere un código útil. [7,8]
- Las herramientas de IA “tipo ChatGPT” pueden ser útiles para el alumno y el docente en varios aspectos complementarios al aprendizaje y la enseñanza de la programación, tales como sugerencias sobre un código ya desarrollado incluyendo detección de errores o soluciones tentativas a un problema simple que sirvan para

- su análisis y perfeccionamiento por el docente y/o el alumno. [9,10]
- Naturalmente en el ámbito profesional (donde el usuario ya está capacitado como programador) y en un lenguaje determinado, estas herramientas pueden acelerar el proceso de desarrollo de software al generar rápidamente código que luego el programador formado puede perfeccionar. [11,12,13]
 - Obviamente el alcance y calidad de las soluciones que ofrecen hoy estas herramientas de IA como generadoras de código, está acotado al conjunto de lenguajes que manejen y la variedad de soluciones (y su evolución) que tienen en su base de conocimiento. Como veremos más adelante, si el lenguaje le resulta poco conocido o si la complejidad del problema obliga a combinar recursos de un lenguaje y otro (por ejemplo buscando mejorar el tiempo de respuesta o la eficiencia o el consumo energético) es más difícil “reemplazar un programador” por una herramienta de IA tipo ChatGPT. [14]

2. Análisis de Ejemplos

En la Facultad de Informática de la UNLP las asignaturas Conceptos de Algoritmos Datos y Programas (CADP) y Taller de Programación (TP) son comunes del primer año de los planes de estudio vigentes tanto de las Licenciaturas cortas de Analista.

En la primera de ellas se abordan conceptos iniciales de programación, utilizando programación estructurada, incluyendo temas como el análisis de un problema, la modelización de este, la expresión del algoritmo que lo resuelve, la modularización en el caso de la programación procedural y la definición de las estructuras de datos necesarias para resolver el mismo.

Se discuten en el curso conceptos de importancia en la formación del alumno como la elección correcta de las estructuras de control y de datos, la documentación adecuada del código, su corrección funcional y también su eficiencia relativa a otros modos de implementar la solución.

En esta primera asignatura se utiliza el lenguaje Pascal como una guía para que el alumno pueda implementar los algoritmos y eventualmente probar en una máquina sus soluciones. Si bien Pascal normalmente no es un lenguaje utilizado comercialmente en programación en la actualidad, resulta claro que se lo emplea en el ámbito educativo por ser simple, estructurado, fácil de entender y ejecutable en cualquier modelo de máquina que el alumno disponga (en el aula o en su casa). Por otro lado se busca que el alumno se concentre en la expresión correcta de los algoritmos y no en las complejidades del lenguaje. En el segundo semestre como correlativa de CADP los alumnos cursan Taller de Programación en donde se ven tres paradigmas de programación: imperativo, orientado a objetos y concurrente. En el primer modelo de programación imperativa, se profundizan los conceptos de CADP y se introducen los conceptos de recursividad y estructuras de datos no lineales como árboles; en el segundo modelo, programación orientado a objetos, se presentan los conceptos de clases e instancias, herencia, jerarquía y polimorfismo, y se utiliza el lenguaje de programación Java; finalmente, en el tercer modelo se abordan de manera inicial los conceptos de la programación concurrente, vinculados a la división del problema en procesos que se comunican y sincronizan utilizando memoria compartida o pasaje de

mensajes, y se utiliza el entorno CRME [15], desarrollado por investigadores del instituto III-LIDI, para implementar los algoritmos en un lenguaje concurrente básico (R-Info). Es importante mostrar en estos ejemplos la sincronización por memoria compartida y por mensajes y su potencial combinación en un problema real.

En este punto la idea de este trabajo ha sido “preguntarle” a ChatGPT cuál es la solución a diferentes problemas planteados y analizar la misma. Este análisis incluye evaluar si existe respuesta, si el código generado es correcto y si tiene en cuenta conceptos de importancia en la formación de los alumnos.

Ejemplo 1: Aquí se abordan temas relacionados con la programación procedural incluyendo análisis del problema, modularización del mismo, utilización de las estructuras de control (while, repeat until, for e if) y las estructuras de datos (registros, arreglos y listas) vistas en el curso.

Usuario>> *La compañía Canonical Llt. desea obtener estadísticas acerca del uso de Ubuntu Linux en La Plata. Para ello, debe realizar un programa que lea y almacene información sobre las computadoras con este sistema operativo (a lo sumo 10.000). De cada computadora se conoce: código de computadora, la versión de Ubuntu que utilizan (18.04, 17.10, 17.04, etc.), la cantidad de paquetes instalados y la cantidad de cuentas de usuario que poseen. La información debe almacenarse ordenada por código de computadora de manera ascendente. La lectura finaliza al ingresar el código de computadora -1, que no debe procesarse. Una vez almacenados todos los datos, se pide:*

- *Informar la cantidad de computadoras que utilizan las versiones 18.04 o 16.04.*
- *Informar el promedio de cuentas de usuario por computadora.*
- *Informar la versión de Ubuntu de la computadora con mayor cantidad de paquetes instalados.*
- *Eliminar la información de las computadoras con código entre 0 y 500.*

Respuesta ChatGPT>> *¡Perfecto! El enunciado del ejercicio es claro. Este programa cumple con los requerimientos del enunciado.*

Primero, se define el tipo de registro `Computadora` para almacenar los datos de cada computadora. Luego, se declara un arreglo `computadoras` para almacenar hasta 10,000 computadoras. El programa utiliza un bucle repetitivo para leer los datos de las computadoras hasta que se ingrese el código -1 o se alcance el límite máximo. Después de la lectura, se procesan los datos para obtener los resultados requeridos.

Finalmente, se muestran los resultados en la salida. Ten en cuenta que este código asume que los datos ingresados son válidos y no realiza una validación exhaustiva. Si deseas agregar verificaciones adicionales, puedes modificar el código según tus necesidades.

Espero que esta solución te sea útil. Si tienes alguna otra pregunta, no dudes en preguntar.

En este momento chatGPT brinda una solución que se puede ver de manera completa en:

<https://github.com/gonetil/chatGPT-cadp-taller/blob/main/cadp/solucion1.pas>

En dicha solución se puede observar lo siguiente:

- No incluye modularización, concepto fundamental y obligatorio desde el curso inicial de nivelación de la carrera. Modularizar correctamente es una exigencia

para los alumnos en las evaluaciones normales de la asignatura.

- El procesamiento de los datos se realiza mientras se leen los mismos. Si bien esto no es incorrecto desde el punto de la solución, no responde al enunciado que pedía procesar luego de leer los datos.
- La estructura de control utilizada es una estructura postcondicional (repeat until), cuando se esperaba un precondicional (while). Si bien la solución funciona, la elección de la estructura de control más adecuada es uno de los puntos importantes en este tipo de ejercicios para evaluar.
- La eliminación de los datos en el vector se realiza poniendo valores en 0 en las posiciones que se deben eliminar. En este punto es importante aclarar que en la asignatura se trabaja con dimensiones físicas (cantidad máxima de elementos que se pueden almacenar) y lógicas (cantidad real de elementos almacenados) en un vector, realizando corrimientos de los elementos cada vez que uno debe ser eliminado.

Por lo tanto, analizando estos errores desde el punto de vista de los conceptos abordados en la materia se hace un refinamiento sobre el pedido para el mismo ejercicio, pero indicando algunas pautas:

Usuario>> *Ahora quiero ajustar el programa para que utilice Procedures y Functions, y para que primero se cargue la información en un vector y luego se procese dicho vector para resolver los distintos puntos. Para eliminar valores del vector, no se deben poner en cero, sino que se deben realizar los corrimientos necesarios entre posiciones y la correspondiente actualización de la dimensión lógica*

En respuesta ChatGPT brinda una nueva solución que se puede ver de manera completa en:

<https://github.com/gonetil/chatGPT-cadp-taller/blob/main/cadp/solucion2.pas>

En dicha solución se puede observar lo siguiente:

- La solución no modulariza la lectura de datos. Esto significa que el código para leer los datos está mezclado con el código para generar la estructura de datos. Esto puede hacer que el código sea más difícil de entender, reusar y depurar, y es un concepto que se remarca durante la cursada: la lectura de datos debe ser independiente de la estructura de datos en la cual serán almacenados.
- La solución sigue utilizando una estructura de control postcondicional.
- Se antepone la palabra clave CONST en los parámetros de entrada, indicando así que no van a ser modificados. Si bien existe este concepto en el lenguaje Pascal, es preferible (y así se explica en la asignatura) que los parámetros sean por valor o copia y parámetros por referencia o entrada/salida.
- Al incluir la modularización se recorre la estructura en cada módulo, lo que reduce la eficiencia general de la solución.
- El corrimiento de los valores guardados en el vector no se realiza de manera correcta debido al mal manejo de los índices. Es importante notar que este tipo de errores no es fácil de detectar por alguien que no tiene claros estos conceptos.

Ejemplo 2: este ejemplo aborda los temas relacionados con la programación procedural incluyendo árboles binarios de búsqueda y por ende soluciones recursivas.

Usuario >> *Quiero resolver un ejercicio del módulo Programación Imperativa de la materia Taller de Programación de la Facultad de Informática de la UNLP. El código debe estar en Pascal, y el enunciado dice:*

Escribir un programa que:

a. Implemente un módulo que lea información de ventas de un comercio. De cada venta se lee código de producto, fecha y cantidad de unidades vendidas. La lectura finaliza con el código de producto 0. Un producto puede estar en más de una venta.

Se pide:

- i. Generar y retornar un árbol binario de búsqueda de ventas ordenado por código de producto.*
- ii. Generar y retornar otro árbol binario de búsqueda de productos vendidos ordenado por código de producto. Cada nodo del árbol debe contener el código de producto y la cantidad total vendida. Nota: El módulo debe retornar los dos árboles.*
- b. Implemente un módulo que reciba el árbol generado en i. y un código de producto y retorne la cantidad total de unidades vendidas de ese producto.*
- c. Implemente un módulo que reciba el árbol generado en ii. y un código de producto y retorne la cantidad total de unidades vendidas de ese producto.*

En este momento chatGPT brinda nueva solución que se puede ver de manera completa en:

<https://github.com/gonetil/chatGPT-cadp-taller/blob/main/imperativo>

En dicha solución se puede observar lo siguiente:

- La solución no encapsulaba los datos del árbol en un tipo de datos registro dedicado, sino que todos los campos estaban dentro del mismo nodo del árbol. Esto atenta contra la legibilidad del código, y limita su reutilización.
- La solución utiliza una función para generar los nodos del árbol. Si bien es correcto, no resulta claro para el alumno por el modo en que el tema se explica conceptualmente en el curso. VER
- Al igual que las soluciones del ejemplo 1, no se modularizaba la lectura de datos mezclando el código para leer la información con el código que la procesa. Esto puede hacer que el código sea más difícil de entender y depurar, y reduce su reusabilidad.
- Uno de los problemas más importantes es que la solución cometía dos errores graves: por un lado, en uno de los módulos, confundía el tipo de datos del parámetro real con respecto al parámetro formal al invocarlo, lo que implica que el código no funciona correctamente. Por el otro lado, al recorrer el árbol binario, le faltaba evaluar el caso en que encontró el elemento en el árbol (para lo cual debería acumular un valor). Esto significa que el código no sólo no resuelve correctamente lo pedido, sino que este error puede ser particularmente complejo de identificar para un lector no entrenado (como un alumno de los primeros cursos de programación), ya que requiere comprender muy bien cómo

se ejecuta el algoritmo recursivo, en qué punto debería cortarse la recursión y en qué punto debería invocarse recursivamente.

Ejemplo 3: este ejemplo aborda los temas relacionados con la programación orientada a objetos.

Usuario >> Quiero resolver un ejercicio del módulo Programación Orientada a Objetos de la materia Taller de Programación. El código debe estar en lenguaje Java. El enunciado dice:

Queremos representar a los empleados de un club: jugadores y entrenadores.

- *Cualquier empleado se caracteriza por su nombre, sueldo básico y antigüedad.*
- *Los jugadores son empleados que se caracterizan por el número de partidos jugados y el número de goles anotados.*
- *Los entrenadores son empleados que se caracterizan por la cantidad de campeonatos ganados.*

A- Implemente la jerarquía de clases declarando atributos, métodos para obtener/modificar su valor y constructores que reciban los datos necesarios.

B- Cualquier empleado debe responder al mensaje calcularEfectividad. La efectividad del entrenador es el promedio de campeonatos ganados por año de antigüedad, mientras que la del jugador es el promedio de goles por partido.

C- Cualquier empleado debe responder al mensaje calcularSueldoACobrar. El sueldo a cobrar es el sueldo básico más un 10% del básico por cada año de antigüedad y además:

- *Para los jugadores: si el promedio de goles por partido es superior a 0,5 se adiciona un plus de otro sueldo básico.*
- *Para los entrenadores: se adiciona un plus por campeonatos ganados (5000\$ si ha ganado entre 1 y 4 campeonatos; \$30.000 si ha ganado entre 5 y 10 campeonatos; 50.000\$ si ha ganado más de 10 campeonatos).*

D- Cualquier empleado debe responder al mensaje toString, que devuelve un String que lo representa, compuesto por nombre, sueldo a cobrar y efectividad.

F- Realizar un programa que instancie un jugador y un entrenador. Informe la representación String de cada uno.

NOTA: para cada método a implementar piense en qué clase/s debe definir el método.

ChatGPT brinda nueva solución que se puede ver de manera completa en: <https://github.com/gonetil/chatGPT-cadp-taller/tree/main/objetos>.

En dicha solución se puede observar lo siguiente:

- La solución en Java está muy bien. Si bien puede mejorarse, sobre todo por la falta del uso de un método abstracto en la superclase Empleado, la solución es correcta de acuerdo con los objetivos de la materia.
- Se observó que todas las variables de instancia se declaraban de forma protegida (protected), lo que evitaba el uso de getters y setters, pero atentaba contra el encapsulamiento y ocultamiento de información.

Ejemplo 4: este ejemplo aborda los temas relacionados con la programación concurrente.

Usuario >> *Quiero resolver un ejercicio del módulo Programación Concurrente, de la materia Taller de Programación de la Facultad de Informática de la UNLP. El programa debe realizarse en lenguaje R-Info / CMRE. El enunciado dice*

Tres robots deben recorrer el perímetro de su cuadrante, como se indica a continuación: El robot 1 comienza la esquina (2,2) y debe realizar un cuadrante de 6x6 juntando todas las flores que encuentre El robot 2 comienza en la esquina (5,5) y debe realizar un cuadrante de 10x10 juntando todas las flores y los papeles que encuentre El robot 3 comienza en la esquina (9,9) y debe realizar un cuadrante de 7x7 juntando todos los papeles que encuentre Cada robot que finalice su cuadrante deberá avisar al robot fiscalizador. Al recibir el aviso, el robot fiscalizador indicará inmediatamente una calle a la que deberá dirigirse el robot recolector, considerando que el robot que finalizó primero irá a la calle 20, el segundo a la 21 y el tercero a la 22. Cuando los robots recolectores reciben un número de calle, deberán posicionarse en la avenida 1 de dicha calle, y avanzar a lo largo de la calle depositando en cada esquina un papel, una flor o ambos, según lo que cada robot haya juntado. El recorrido finalizará al completar la calle o vaciarse las bolsas.

ChatGPT >> *Lamentablemente, no estoy familiarizado con el lenguaje de programación R-Info mencionado en el enunciado. Como modelo de lenguaje de IA, mi conocimiento se limita a los lenguajes de programación más comunes como Python, Java, C++, entre otros.*

Sin embargo, puedo ofrecerte una solución general utilizando un enfoque pseudocódigo que puedes adaptar al lenguaje de programación R-Info o cualquier otro lenguaje que estés utilizando.

- *La solución no está programada en lenguaje R-Info, y por lo tanto no permite evaluar el buen uso de las herramientas del lenguaje. Por ejemplo, qué información se envía al fiscalizador, o si se considera que el envío es asíncrono y la recepción es síncrona. El pseudocódigo propuesto no considera estos elementos de manera explícita.*

2.1 Síntesis de los resultados obtenidos de las respuestas generadas por ChatGPT

Analizando cada una de las soluciones a los problemas planteados podemos observar lo siguiente:

Programación Imperativa	Programación Orientada a Objetos	Programación Concurrente
<p>En general la solución funciona, pero no es correcta en conceptos fundamentales para la formación en programación.</p> <p>En CADP dos conceptos importantes son la modularización que involucra el pasaje de parámetros y también la interpretación correcta del enunciado por el alumno.</p> <p>Ambos conceptos no son considerados en soluciones iniciales de chatGPT.</p> <p>En el caso de soluciones recursivas el problema detectado es peor, ya que las soluciones brindadas contenían errores, los cuales en algunos casos son difíciles de detectar por un usuario no entrenado o formado.</p>	<p>En este tipo de problemas las soluciones presentadas por chatGPT son correctas y las que más se aproximan a la manera en que se enseñan los temas en la asignatura.</p>	<p>No existen soluciones en el lenguaje que se utiliza en la materia y las aproximaciones brindadas no pueden evaluarse.</p> <p>Esto refleja una limitación natural de las herramientas de IA utilizadas para generar código: su conocimiento de lenguajes determinados depende de haber sido entrenadas en los mismos.</p>
<p>Estos problemas observados en las soluciones a ejemplos en los distintos paradigmas hacen difícil que un usuario no formado completamente o que se está empezando a formar pueda:</p> <ul style="list-style-type: none"> ▪ Identificar determinados errores. ▪ Encontrar alternativas mejores (o más eficientes) a la solución presentada. ▪ Asimilar la solución generada por ChatGPT a los conceptos vistos en las asignaturas. 		

3. Impacto en la Metodología de Enseñanza y Aprendizaje de Programación

Al momento es muy difícil tener conclusiones sobre el empleo sistemático y planificado de este tipo de herramientas en la Metodología de enseñanza y aprendizaje en las asignaturas iniciales de Programación. De todos modos, hay que considerar que los alumnos emplean (y emplearán cada vez más) estos recursos en la generación de código.

Mencionaremos algunos aspectos en los que se está trabajando para implementarlos en los cursos iniciales de programación en la Facultad de Informática de la UNLP:

A. Mecanismos de interacción con el alumno, utilizando los desafíos del código generado por ChatGPT (o herramientas similares). Entendemos que si el alumno comprende su rol de analizar críticamente la solución que ofrece la herramienta de IA y mejorarla según las pautas que ha aprendido en el curso, se mejora su motivación para el aprendizaje de criterios de optimización de algoritmos. En particular la cátedra trabaja con los alumnos que recurren con una metodología denominada “Autoaprendizaje Guiado” que tiende a un modelo de “Aula Invertida”, en la cual este tipo de desafíos encuadra perfectamente.

B. Insistir en aspectos de fondo en la formación en las capacidades propias de un programador presentando cuestiones conceptuales importantes para este nivel de cursos iniciales, como la modularización, la eficiencia comparada de soluciones, el empleo o no de parámetros, la reutilización de patrones de programación, etc. utilizando las soluciones de ChatGPT (o herramientas similares) para destacar estos aspectos, a partir de su análisis crítico y de cómo perfeccionar un código buscando (por ejemplo) mayor legibilidad o eficiencia o facilidad de mantenimiento.

C. Reforzar el tema de interpretación de enunciados y mostrar como enunciados funcionalmente equivalentes pueden interpretarse de modo diferente por una herramienta automática, para lograr que el alumno haga un análisis profundo del problema que se plantea. El tema es crítico en los cursos iniciales de Programación, más allá del empleo o no de herramientas de IA y mostrarle al alumno que NO debe hacer una lectura apresurada de los enunciados y que es necesario que desarrolle capacidad de abstracción sobre el enunciado del problema es de gran importancia. Para esto emplear la comparativa con interpretaciones parciales o incorrectas de enunciados por ChatGPT es útil.

D. Discutir el empleo de estas herramientas en temas especialmente dificultosos para el alumno de los cursos iniciales de programación, tales como Recursividad / Estructuras dinámicas como Listas, Arboles y Grafos y el concepto de eficiencia y calidad en el código que se genera en un lenguaje dado.

E. Analizar los modelos de evaluación práctica que se utilizan en forma sincrónica y asincrónica (en cursos presenciales y a distancia) y considerar el impacto que puede tener el empleo de herramientas “tipo ChatGPT” por los alumnos en las evaluaciones. Aquí hay un trabajo de sensibilización en aspectos éticos para el alumno (que le serán útiles en su vida profesional) y también en el desarrollo de capacidades tecnológicas desde la asignatura para detectar situaciones de “plagio” basadas en el empleo de herramientas como ChatGPT. Al mismo tiempo esto exige que los docentes planteen evaluaciones que permitan valorar los reales aprendizajes del alumno.

F. Insistir en lo conceptual: Que el alumno comprenda que “tener una solución a un problema correctamente especificado” NO es lo mismo que “analizar un problema del mundo real y encontrar un modo de resolverlo en forma eficiente utilizando una computadora”. Para esto se trabajan en enunciados incrementales de problemas que conducen al mismo algoritmo y se muestra que no siempre el “código automático” es correcto, porque la fase de análisis del problema es más importante que la generación de código en sí.

4. Reflexiones, conclusiones y líneas de trabajo futuro

En este trabajo se analizaron cuestiones metodológicas para aprovechar constructivamente el empleo de herramientas de IA tipo ChatGPT en el aprendizaje y enseñanza inicial de Programación. Al mismo tiempo se estudiaron casos simples de algoritmos y su expresión en lenguajes utilizados en primer año en cursos de la Facultad de Informática de la UNLP, de modo de ver la potencialidad de ChatGPT para resolverlos correctamente y qué dificultades/errores se encontraron.

Incorporar herramientas de IA en el Aula es un proceso en pleno desarrollo, en todos los niveles educativos y este trabajo nos muestra potencialidades para favorecer el tiempo de aprendizaje del alumno e incrementar la interacción con las tareas que puede realizar durante los trabajos prácticos y fuera del ámbito de las clases.

Al mismo tiempo nos presenta desafíos como los cambios metodológicos, la capacitación de los docentes y la adaptación de los mecanismos de evaluación de modo de asegurar el aprendizaje real del alumno y evitar que el uso de herramientas de IA lo confundan pensando que la “generación de código que resuelve un algoritmo” cubre los aspectos formativos que se buscan en los cursos iniciales de programación.

Quedan abiertas muchas líneas de trabajo, entre las que mencionaremos:

- Ampliación de los casos de estudio, incrementando la complejidad de los enunciados y de los problemas a resolver.
- Considerar aspectos a optimizar en las soluciones (tiempo, eficiencia, consumo).
- Analizar problemas de concurrencia y sus soluciones automáticas y óptimas.
- Ver comparativamente otras herramientas de generación automática de código en los lenguajes utilizados en la formación inicial en Programación.
- Discutir mecanismos de evaluación y autenticación de los trabajos de los alumnos.

Referencias

- [1] <https://chat.openai.com/> Julio 2023
- [2] Augmented intelligence in programming learning: Examining student views on the use of ChatGPT for programming learning R Yilmaz, FGK Yilmaz - Computers in Human Behavior: Artificial Humans, 2023 – Elsevier. journal homepage: www.journals.elsevier.com/computers-in-human-behavior-artificial-humans
- [3] ChatGPT for education and research: Opportunities, threats, and strategies. MM Rahman, Y Watanobe - Applied Sciences, 2023 - mdpi.com
- [4] Inteligencia artificial al servicio de la educación: Chat GPT para Docentes Guillermo Antón Pardo. Ediciones Rodio- ISBN 9788418794346. Mayo 2023
- [5] Programa de Conceptos de Algoritmos, Datos y Programas (2023). Disponible en: <https://www.info.unlp.edu.ar/wp-content/uploads/2023/03/CADP-2023.pdf> . Último acceso 8 de agosto de 2023.
- [6] Programa de Taller de Programación. Disponible en <https://www.info.unlp.edu.ar/wp-content/uploads/2023/03/Taller-de-Programacion-Redictado.pdf> . Último acceso 8 de agosto de 2023
- [7] AI and Machine Learning for Coders: A Programmer's Guide to Artificial Intelligence de Laurence Moroney. Editorial O'Reilly Media ISBN 978-1492078197.
- [8] Is ChatGPT the Ultimate Programming Assistant--How far is it?. H Tian, W Lu, TO Li, X Tang, SC Cheung, J Klein, TF Bissyandé. arXiv preprint [arXiv:2304.11938](https://arxiv.org/abs/2304.11938), 2023•arxiv.org
- [9] GPTutor: a ChatGPT-powered programming tool for code explanation. Eason Chen, Ray Huang, Han-Shin Chen, Yuen-Hsien Tseng, Liang-Yi Li. International

Conference on Artificial Intelligence in Education 2023.

- [10] Use chat gpt to solve programming bugs. NMS Surameery, MY Shakor – International Journal of Information technology and Computer Engineering (IJITC) ISSN : 2455-5290. Vol. 3 No. 01 (2023): Dec 2022-Jan 2023
- [11] ChatGPT as a Support Tool for Online Behavioral Task Programming Laurent Avila Chauvet, Diana Mejía Cruz & Christian Oswaldo Acosta Quiroz. Sonora Institute of Technology. Available at SSRN ..., 2023 - papers.ssrn.com
- [12] Pair Programming with ChatGPT: AI-Enhanced Coding for the Modern Developer de Michael D Callaghan.ISBN 979-8394911040
- [13] How Coders Can Survive—and Thrive—in a ChatGPT World . 4 Tips for programmers to stay ahead of generative AI Rina Diane Carballar.IEEE Spectrum – Julio 2023.
- [14] Augmented intelligence in programming learning: Examining student views on the use of ChatGPT for programming learning. R Yilmaz, FGK Yilmaz – Computers in Human Behavior: Artificial Humans, 2023 – Elsevier. www.journals.elsevier.com/computers-in-human-behavior-artificial-humans
- [15] De Giusti, Laura; Leibovich, Fabiana; Chichizola, Franco; Naiouf, Marcelo (2016). Teaching Concurrency and Parallelism Concepts with CMRE. Journal of Computer Science & Technology; vol. 16, no. 2. ISSN 1666-6038.