

Study of the impact of ChatGPT in the teaching of introductory subjects to programming

MSc. Ing. Ambrosini Alejandro 
alejandro.ambrosini@unc.edu.ar

Bottini Franco Nicolas , Robledo Valentín , Lavezzari Fausto  y Lencina Aquiles Benjamín 
francobottini99@mi.unc.edu.ar, valentin.robledo@mi.unc.edu.ar, fausto.lavezzari@unc.edu.ar,
aquiles.lencina@mi.unc.edu.ar

*Laboratorio de Computación, Departamento de Computación, Facultad de Ciencias Exactas, Físicas y Naturales,
Universidad Nacional de Córdoba, Argentina*

Abstract– *The mass use of artificial intelligence tools is a reality that is currently going through and seems to trigger an important paradigm shift in work and educational fields. This article exposes the impact generated by one of these new artificial intelligence tools, ChatGPT, in teaching the basic concepts of programming. To achieve this goal, the behavior of this technology is analyzed in the resolution of evaluations of an introductory subject to programming in C++ of the Facultad de Ciencias Exactas, Física y Naturales de la Universidad Nacional de Córdoba, Argentina. With this analysis, it is revealed that ChatGPT is capable of solving the evaluative exercises with an effectiveness percentage of 97%, therefore, a student without the necessary knowledge can pass an exam. From this, it can be concluded that the reliability of current evaluation methods is in jeopardy. In addition, it is possible to infer that the misuse of this technology by students in the early stages of learning has a negative impact on their training. On the other hand, the potential for validation and tutoring provided by this tool is highlighted when it is used honestly.*

Keywords– *ChatGPT, programming, assessment, programming education, artificial intelligence.*



Estudio del impacto de ChatGPT en la enseñanza de materias introductorias a la programación

MSc. Ing. Ambrosini Alejandro 
alejandro.ambrosini@unc.edu.ar

Bottini Franco Nicolas , Robledo Valentín , Lavezzari Fausto  y Lencina Aquiles Benjamín 
francobottini99@mi.unc.edu.ar, valentin.robledo@mi.unc.edu.ar, fausto.lavezzari@unc.edu.ar,
aquiles.lencina@mi.unc.edu.ar

Laboratorio de Computación, Departamento de Computación, Facultad de Ciencias Exactas, Físicas y Naturales,
Universidad Nacional de Córdoba, Argentina

Resumen– La masificación en el uso de herramientas de inteligencia artificial es una realidad que actualmente se está atravesando y aparenta desencadenar un cambio de paradigma importante en ámbitos laborales y educativos. En este artículo se expone el impacto que genera una de estas nuevas herramientas de inteligencia artificial, ChatGPT, en la enseñanza de los conceptos básicos de la programación. Para lograr este objetivo, se analiza el comportamiento de esta tecnología en la resolución de evaluaciones de una asignatura introductoria a la programación en C++ de la Facultad de Ciencias Exactas, Física y Naturales de la Universidad Nacional de Córdoba, Argentina. Con este análisis, se revela que ChatGPT es capaz de resolver los ejercicios evaluativos con un porcentaje de efectividad del 97%, por lo tanto, un estudiante sin los conocimientos necesarios puede aprobar un examen. A partir de esto, se puede concluir que la fiabilidad de los métodos de evaluación actuales está en peligro. En adición, se logra inferir que el mal uso de esta tecnología por parte de los estudiantes en las etapas tempranas de aprendizaje, supone un impacto negativo en su formación. Por otra parte, se destaca el potencial de validación y tutoría que proporciona esta herramienta cuando se realiza un uso honesto de la misma.

Palabras clave– ChatGPT, programación, evaluación, enseñanza de programación, inteligencia artificial.

I. INTRODUCCIÓN

La inteligencia artificial está teniendo una fuerte y acelerada irrupción en nuestra vida cotidiana. Aplicaciones como DALL-E, FakeYou o FaceSwap están a disposición del común de los usuarios y permiten generar imágenes, audios e incluso videos haciendo uso de IA. La masificación y el auge en el desarrollo de estas tecnologías, además de ventajas o beneficios, trae consigo un conjunto de problemáticas asociadas a la creación de contenido original, suplantación de identidad, fraude, manipulación de campañas electorales, privacidad, seguridad de los datos, discriminación algorítmica, falta de transparencia en los procesos de toma de decisiones, entre muchos otros problemas éticos y sociales [1].

Dentro de todo este nuevo espectro de herramientas que utilizan inteligencia artificial, existe una en particular que,

debido a su amplia difusión, motiva el desarrollo de este artículo: ChatGPT de OpenAI [2].

ChatGPT (Generative Pre-training Transformer), es una inteligencia artificial desarrollada por OpenAI basada en el modelo de lenguaje natural GPT-3.5 y entrenado para generar todo tipo de texto. Esta herramienta, es capaz de mantener conversaciones con el usuario mediante un chat, siendo sus respuestas destacables por su alto nivel de realismo y similitud con el lenguaje humano. El sistema está basado en un modelo de lenguaje generativo creado con 175 mil millones de parámetros entrenados a partir de 8 millones de documentos [3].

Por otra parte, cabe destacar, que ChatGPT cuenta con una versión de acceso libre y gratuita a través de la web, lo que la pone al alcance de cualquier persona interesada.

Por estos motivos, la herramienta, comienza a despertar inquietudes en el ámbito de la educación, debido a que brinda a los estudiantes la posibilidad de generar ensayos, resolver problemas e incluso generar código fuente utilizando entradas de lenguaje natural.

El impacto real que pueden llegar a tener estas nuevas herramientas en la educación a largo plazo todavía es un enigma. Aunque el conocimiento de este tema ha avanzado de manera vertiginosa en los últimos años, todavía existe una gran incertidumbre en cuanto a su verdadero potencial en la enseñanza y aprendizaje, sin embargo, se observan diversas publicaciones que abordan este tema [4 - 8].

En estas publicaciones, se plantea la utilización de herramientas de IA para la creación de explicaciones y devoluciones personalizadas, disponibles para el usuario en todo momento y lugar. Pero por otra parte, también expone sus limitaciones, se puede decir que el principal limitante de este tipo de sistemas es su entrenamiento, ya que su capacidad de respuesta depende del material fuente con el que lo hayan alimentado. Un set de entrenamiento limitado o sesgado da como resultado una IA que genera respuestas limitadas y sesgadas.

El temor al mal uso de estos sistemas por parte de los estudiantes es una realidad latente en el ámbito educativo, la posibilidad de cometer fraudes en instancias evaluativas es un hecho factible [9 - 12].

La enseñanza de la programación, es a todas luces, una de las áreas más afectadas por el surgimiento de estas herramientas. Este modelo de lenguaje es perfectamente capaz de generar código fuente a partir de entradas sencillas sin necesidad de altos conocimientos técnicos, lo que resulta útil para los profesionales en el área del desarrollo de software, ya que posibilita un aumento en la productividad. Sin embargo, el abuso de esta herramienta por parte de estudiantes en etapas tempranas de aprendizaje, puede afectar la comprensión de los conocimientos fundamentales de la programación y significar un problema en su formación a futuro.

Con estos apartados en mente, en esta publicación se busca puntualizar el análisis y explorar el impacto que esta herramienta supone en la enseñanza de los conceptos fundamentales de la programación. Se tiene como objetivo visibilizar y dar a conocer las implicaciones de esta tecnología en este campo en particular de la educación. Para ello, se utilizará como material de estudio evaluaciones tomadas en la materia de Informática, asignatura introductoria a la programación en C++ para las carreras de ingeniería de la Facultad de Ciencias Exactas, Física y Naturales de la Universidad Nacional de Córdoba, Argentina. Teniendo en cuenta que, estas evaluaciones tienen un tiempo límite de desarrollo de dos horas para el alumno. Además, durante estas evaluaciones los docentes no permiten la utilización de ChatGPT para su resolución. Con todo esto, se analizará en profundidad la resolución brindada por *ChatGPT* a estos problemas.

II. MATERIAL DE ANÁLISIS

Como se mencionó anteriormente, el material de análisis son evaluaciones dadas en la materia de Informática, esta asignatura tiene como objetivo brindar fundamentos básicos de programación en C++ a estudiantes de primer año de diversas ingenierías. Puntualmente, se analiza un tipo de evaluación en concreto, las cuales se denominan “especificación de algoritmos”, y son realizadas por los alumnos a través de una plataforma de enseñanza virtual en la web llamada *Moodle*.

Moodle es un sistema de gestión de aprendizaje gratuito y de código abierto que se utiliza para el aprendizaje combinado, la educación a distancia, el aula invertida y otros esquemas de aprendizaje en línea en escuelas, universidades, lugares de trabajo y otros sectores.

La “especificación de algoritmos”, consiste en la implementación de un programa de manera escalonada. El programa a implementar se subdivide en varios subprogramas o funciones, y se le pide al estudiante que desarrolle uno a uno estos subprogramas. Cada uno de estos subprogramas implementados se evalúa utilizando una serie de pruebas automatizadas mediante *CodeRunner* [13] que determinan si la resolución es correcta. Los estudiantes pueden ver el resultado de cada prueba inmediatamente y recibir comentarios sobre sus soluciones.

Finalmente, la evaluación cuenta con un ejercicio integrador que requiere que los estudiantes combinen todos los subprogramas realizados para obtener el programa completo. A diferencia de los subprogramas individuales, este punto no se evalúa automáticamente y requiere la intervención de un docente.

El docente, una vez finalizado el examen, evalúa la capacidad del estudiante para integrar los diferentes subprogramas de manera efectiva y verificar que el programa completo funcione correctamente. La puntuación de la evaluación se basa en la cantidad de pruebas automáticas que se han superado en cada subprograma y en el resultado de la corrección por parte del docente del ejercicio final de integración. Se espera que los estudiantes demuestren su comprensión de los fundamentos de programación y su capacidad para aplicarlos de manera efectiva en la creación de programas completos.

Se considera que la metodología de evaluación utilizada en la asignatura de informática es efectiva y suficiente para evaluar el conocimiento y las habilidades de programación básicas de los estudiantes en un entorno de aprendizaje práctico y en línea.

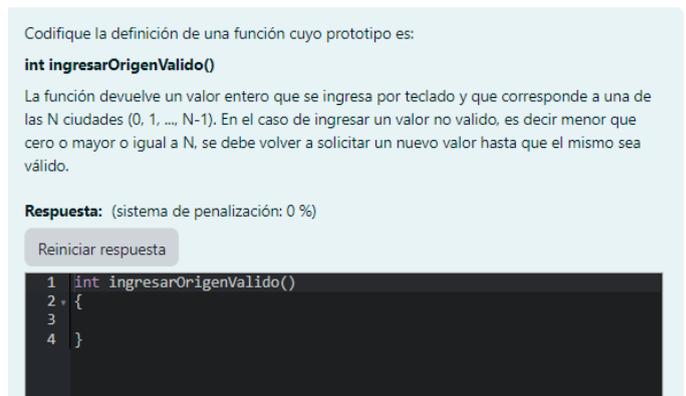


Fig. 1 Visión del estudiante de una consigna de la “especificación de algoritmos”.

III. METODOLOGÍA DE TRABAJO

Para la realización de este trabajo se utilizó la versión web gratuita de *ChatGPT*, dado que es la de mayor accesibilidad para los estudiantes. Si se quiere ingresar a esta, basta con registrarse en la página oficial de *OpenAI*.

Con el material de análisis a disposición, la metodología de trabajo utilizada fue la siguiente:

1. Primero, se inició la conversación con el chatbot especificando que en las siguientes entradas de texto, los programas solicitados, deberán ser codificados en el lenguaje de programación C++.
2. Una vez se le ha especificado el lenguaje y la tarea que debe realizar, se le dio la consigna, tal cual está dada en el entorno de *Moodle*, para el desarrollo de un subprograma.
3. Se tomó la respuesta brindada por el chatbot y sin realizar modificación alguna, se la introdujo en el entorno de evaluación web para posteriormente ejecutar todos los test automatizados sobre la misma.
4. Una vez ejecutadas las pruebas, se registró el número de test automáticos superados con el primer código generado por *ChatGPT*.

- En caso de no haber superado alguna de las pruebas automatizadas, se entabló un diálogo con el chatbot buscando obtener una versión del código que supere todos los test definidos por la cátedra.
- Ya obtenida la nueva respuesta, se volvió a repetir el proceso de testeo.
- Nuevamente, se registró el número de test superados.
- Si aún siguen fallando algunos test, se repite una última vez el proceso de diálogo con el chatbot para obtener una nueva versión del código y testear.

Con estos pasos, se busca replicar el comportamiento que tendría el estudiante al realizar un ejercicio con *ChatGPT*.

Una vez realizado el experimento con todos los ejercicios a disposición, se efectúa una recopilación estadística de los resultados obtenidos. Con esto, se presentan dos gráficos, el primero describe el porcentaje de ejercicios en los cuales se logró superar en su totalidad los test automatizados en intentos de uno, dos y hasta tres pasos, se considera un paso a una interacción con *ChatGPT*. Por otra parte, el segundo gráfico muestra la calificación correspondiente a cada una de las evaluaciones realizadas teniendo en cuenta los puntajes obtenidos en los ejercicios del examen, primero en el único paso y luego, la mejor calificación que se logró obtener en intentos de hasta tres pasos.

Finalmente, teniendo en cuenta el análisis de resultados y la experiencia obtenida de la interacción con el chatbot, se logra dar una visión detallada acerca del comportamiento general de *ChatGPT* en la resolución de este tipo de problemas introductorios a la programación.

IV. EJEMPLO DE APLICACIÓN

Se ejemplifica a continuación, la aplicación de la metodología descrita anteriormente:

- Especificación del lenguaje:

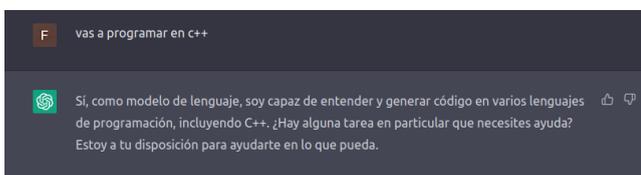


Fig 2. Captura del inicio de la conversación con *ChatGPT*.

- Se pasa como prompt (conjunto de palabras que desencadenan la generación de contenido a través de un software de inteligencia artificial) la consigna tal cual está dada por la cátedra, en este caso es la siguiente:

Codifique la definición de una función con prototipo: void CargarDistanciasCiudades(double distancias[N][N])

La función recibe como argumento una matriz de NxN que representa las distancias en kilómetros entre las ciudades denominadas 0, 1, ..., y N-1. Por lo tanto, el elemento en la fila i y columna j representa la distancia desde la ciudad 'i' hasta la ciudad 'j'. La función debe solicitar al usuario ingresar las distancias entre ciudades por teclado y almacenarlas en la matriz.

- No necesariamente la distancia entre dos ciudades debe ser la misma desde una de ellas o desde la otra, por lo que se debe ingresar la distancia tanto desde la ciudad i a la ciudad j, como también desde la ciudad j hasta la ciudad i.*
- La función NO DEBE solicitar la distancia entre una ciudad y la misma ciudad. Es decir, no debe solicitar la distancia desde la ciudad i a la ciudad i, y debe asignar directamente el valor de distancia 0 a esos casos.*
- En el caso que se ingrese una distancia menor o igual a cero, la función deberá indicar mediante un mensaje que el valor no es válido y pedir que se ingrese nuevamente hasta que el mismo sea mayor que cero.*

La respuesta obtenida de *ChatGPT* es:

Aquí está el código de la función cargarDistanciasCiudades en C++ que cumple con las especificaciones indicadas en el enunciado:

```
void cargarDistanciasCiudades(double distancias[N][N]) {
    for (int i = 0; i < N; i++){
        for (int j = i+1; j < N; j++){
            double distancia;
            do {
                cout<<"Ingrese la distancia entre la ciudad " << i << " y la ciudad "<<j<<" ";
                cin >> distancia;
                if (distancia <= 0) {
                    cout<<"La distancia ingresada no es valida. Por favor ingrese un valor mayor a cero.
                }
            } while (distancia <= 0);
            distancias[i][j] = distancia;
            distancias[j][i] = distancia; // Distancia de i a j es igual a la de j a i
        }
        distancias[i][i] = 0; // Distancia de una ciudad a si misma es 0
    }
}
```

Fig 3. Primer código generado por *ChatGPT*.

La función utiliza dos bucles anidados para recorrer la matriz de distancias y solicitar al usuario que ingrese las distancias entre cada par de ciudades. El bucle externo itera sobre las filas de la matriz, y el bucle interno itera sobre las columnas, pero sólo tomando en cuenta las columnas a la derecha de la diagonal principal, para evitar que se pidan las distancias entre una ciudad y sí misma o que se ingresen distancias duplicadas.

Dentro del bucle interno, se utiliza un bucle do-while para pedir al usuario que ingrese la distancia entre la ciudad 'i' y la ciudad 'j', y verificar que sea mayor a cero. Si el usuario ingresa una distancia inválida, se muestra un mensaje de error y se pide que se ingrese el valor nuevamente.

Una vez que se valida la distancia, se asigna el valor correspondiente tanto a 'distancias[i][j]' como a 'distancias[j][i]', ya que la distancia entre 'i' y 'j' es igual a la distancia entre 'j' e 'i'.

Finalmente, fuera del bucle interno, se asigna el valor 0 a la distancia entre una ciudad y sí misma, es decir, 'distancias[i][i]'.

Se observa que además del código, *ChatGPT*, nos entrega una explicación detallada del mismo.

- El código obtenido se introdujo textualmente en el entorno de ejecución de Moodle y se lanzaron las pruebas automatizadas.

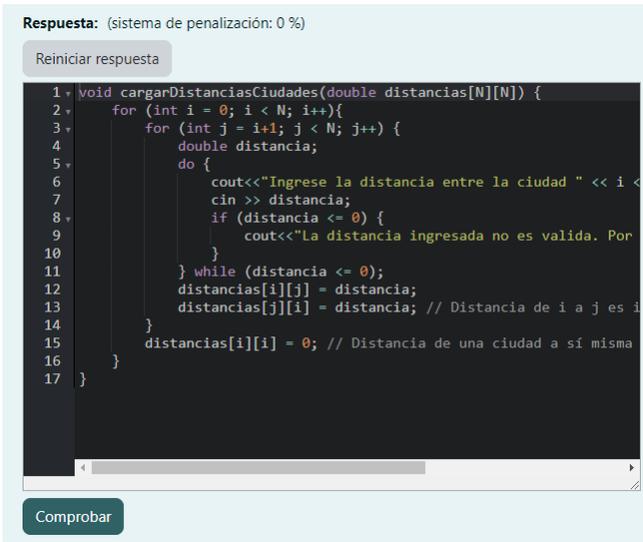


Fig 4. Visión del entorno de ejecución de la EA en el aula virtual de Moodle.

- El código dado tiene un error debido a una mala comprensión de la consigna. La herramienta, interpretó erróneamente la siguiente petición: “No necesariamente la distancia entre dos ciudades debe ser la misma desde una de ellas o desde la otra, por lo que se debe ingresar la distancia tanto desde la ciudad *i* a la ciudad *j*, como también desde la ciudad ‘*j*’ hasta la ciudad ‘*i*.’” y considero la matriz como simétrica. Por esto, no logré superar todos los tests y obtuve 6/15 puntos.

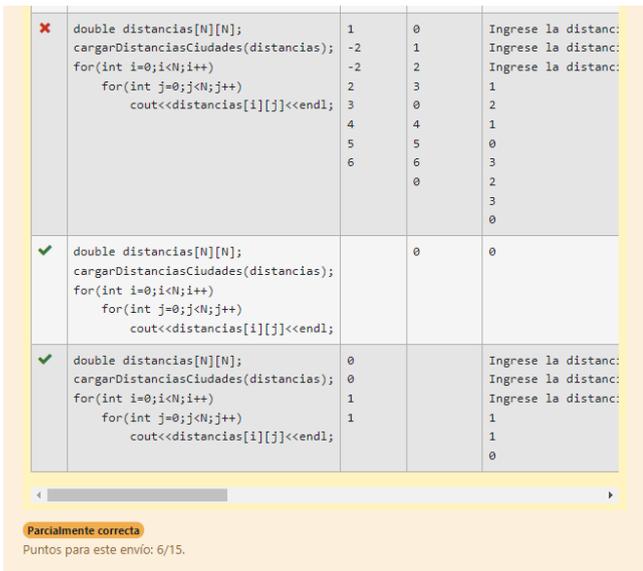


Fig 5. Parte del resultado de la primera ejecución de las pruebas automatizadas.

- Se le remarco este error a la IA utilizando como prompt: “Estas asumiendo erróneamente que la matriz es simétrica”. Con esto se obtuvo una nueva respuesta.

Tienes razón, lo siento por la confusión en mi respuesta anterior. En caso de que la matriz de distancias no sea simétrica, el enfoque adecuado sería solicitar al usuario que ingrese la distancia de una ciudad a otra sin reflejarla en la posición opuesta. Aquí está una posible implementación de la función `cargarDistanciasCiudades` en C++ que maneja correctamente una matriz de distancias que no es simétrica:

```

void cargarDistanciasCiudades(double distancias[N][N]) {
  for (int i = 0; i < N; i++) {
    for (int j = 0; j < N; j++) {
      if (i != j) { // No solicitamos la distancia de una ciudad a sí misma
        double distancia;
        do {
          cout << "Ingrese la distancia entre la ciudad "<<i<<" y la ciudad "<<j<<": ";
          cin >> distancia;
          if (distancia <= 0) {
            cout << "La distancia debe ser mayor que cero. Intente nuevamente.\n";
          }
          while (distancia <= 0);
          distancias[i][j] = distancia;
        } else {
          distancias[i][j] = 0;
          //Asignamos directamente el valor 0 a la distancia de una ciudad a sí misma
        }
      }
    }
  }
}

```

Fig 5. Nueva versión del código generado por ChatGPT.

La función utiliza dos bucles `for` anidados para recorrer la matriz de distancias y solicitar al usuario que ingrese las distancias entre ciudades. El bucle externo itera sobre las filas de la matriz, mientras que el bucle interno itera sobre las columnas. Dentro del bucle interno, la función utiliza un condicional `if` para verificar si la ciudad actual es diferente de la ciudad actual del bucle externo. Si es así, se solicita al usuario que ingrese la distancia entre las dos ciudades y se almacena en la matriz de distancias en la posición correspondiente $[i][j]$. Si la ciudad actual es la misma que la del bucle externo, se asigna directamente el valor 0 a la distancia de una ciudad a sí misma $[i][i]$. Espero que esto solucione el problema. Si tienes alguna otra pregunta o inquietud, no dudes en preguntar.

- Este nuevo código obtenido se introdujo nuevamente en el entorno de ejecución.
- Finalmente, se superaron todos los tests, obteniendo 15/15 puntos.



Fig 6. Parte del resultado de la nueva ejecución de las pruebas automatizadas.

IV. RESULTADOS OBTENIDOS

El experimento se realizó con un total de seis evaluaciones, en donde entre todas contemplan 27 ejercicios sobre los cuales se ejecutaron 335 pruebas automatizadas.

De esta manera, se obtuvo que: en 20 ejercicios *ChatGPT* entregó en un intento de un solo paso la respuesta que superó todas las pruebas, en cuatro ejercicios necesitó intentos de dos pasos para entregar un código que lograra superar el total de las pruebas y en uno se necesitó un intento de tres pasos para conseguir el resultado buscado. Finalmente, en dos ejercicios *ChatGPT* no logró entregar una respuesta que complete la totalidad de las pruebas.

De esta manera, se resume el porcentaje de resolución de ejercicios en el siguiente gráfico:

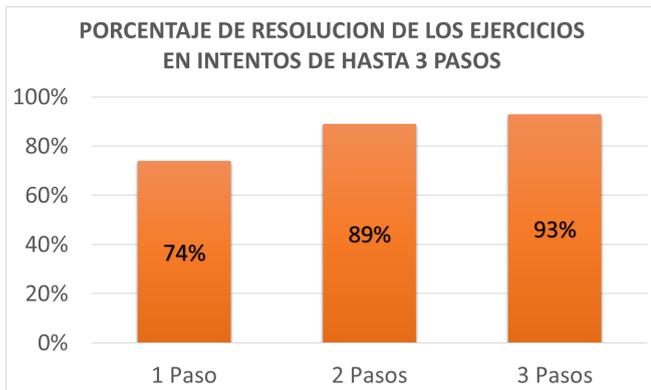


Fig 7. Resultados de la resolución de *ChatGPT* a los ejercicios de las evaluaciones. Fuente: experimentos propios.

Realizando un análisis de la resolución individual de cada evaluación y teniendo en cuenta los resultados obtenidos para los ejercicios resueltos en un solo paso, *ChatGPT* obtiene una calificación mayor al 60% en cuatro evaluaciones y logra calificación perfecta en dos de estas cuatro.

Por otra parte, tomando la mejor solución obtenida para los ejercicios, se tiene que en los seis exámenes, *ChatGPT*, supera la calificación del 60% mientras que en cuatro de los seis obtiene la calificación perfecta.

Estos resultados se resumen en el siguiente gráfico:

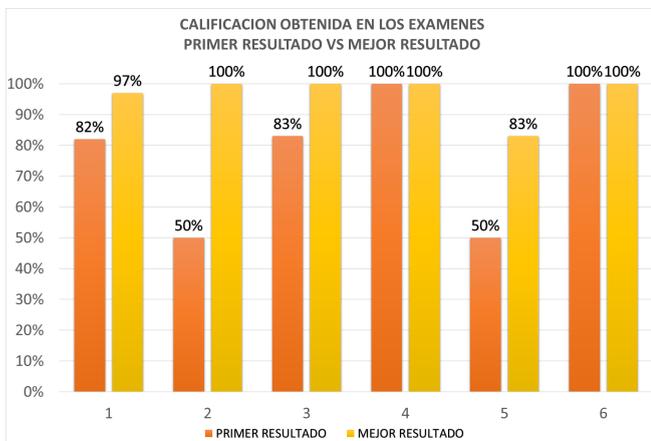


Fig 8. Calificaciones obtenidas por *ChatGPT* en las distintas evaluaciones. Fuente: experimentos propios.

Cabe recalcar que esta es la calificación obtenida solamente con los ejercicios de corrección automática, no se tiene en cuenta el ejercicio integrador, contenido en el último paso de la “especificación de algoritmos”, que requiere la intervención de un docente para su cálculo.

Con esto, promediando los resultados obtenidos para el mejor de los casos, podemos concluir que la IA es capaz de resolver los ejercicios planteados con una efectividad de hasta el 97%.

También se observó que, utilizando esta herramienta, el tiempo de realización de las EA resulta visiblemente menor a las dos horas establecidas por la cátedra para completar la evaluación.

V. COMPORTAMIENTO OBSERVADO

Después de examinar las respuestas y resultados dados por *ChatGPT* se pueden remarcar algunas observaciones sobre su comportamiento típico.

A. Instrucciones claras

Al ser exámenes introductorios a la programación, no se evalúa la capacidad del alumno de llegar a una solución creativa e inteligente, sino que se busca que este comprenda y sepa utilizar las herramientas principales de un lenguaje de programación (condicionales, bucles, etc). Es por ello que las consignas suelen ser muy claras y concretas sobre lo que debe hacer el código solicitado.

Este aspecto, favorece enormemente a un sistema como *ChatGPT*, que cuanto mayor información y contexto tenga sobre la tarea a desarrollar, mejores salidas otorga. En este tipo de ejercicios no se da lugar a la interpretación.

B. Corrección de errores

Si bien son pocas, en algunas ocasiones las respuestas brindadas por *ChatGPT* poseen errores, en estos casos, cuando el defecto está en la interpretación del propio enunciado, al momento de solicitarle a la herramienta que corrija la respuesta, se le dificulta abandonar el concepto erróneamente asumido y vuelve a repetir el mismo fallo en reiteradas ocasiones, muchas veces incluso llegando a informar que ya lo corrigió cuando en realidad no fue así. Para lograr que solucione este tipo de equivocaciones, en la mayoría de casos, se debe indicar explícitamente que modificar.

También, se puede mencionar, que el chatbot tiende a reincidir en errores ya solucionados. Cuando se corrige un fallo y se sigue utilizando ese mismo chat para depurar otros apartados de la respuesta, es probable que en futuras salidas vuelva a introducir un error corregido anteriormente.

C. Nivel de detalle

Uno de los aspectos diferenciadores de las respuestas de *ChatGPT* es el nivel de detalle que se plasma en ellas, por lo general el código obtenido es comentado minuciosamente e incluso implementa un control de errores. Esto es un comportamiento muy inusual en una persona que está aprendiendo a programar.

También, cabe destacar que además de brindar un código fuente en respuesta a una petición dada, *ChatGPT*, en la mayoría de ocasiones, entrega una explicación muy completa del código generado, de esta manera, resulta mucho más sencillo para el usuario entender el funcionamiento del programa que nos está generando y poder modificarlo en caso de que lo requiera.

D. *Tecnicismos en la entrada*

En este análisis, la primer entrada que recibe la herramienta es el enunciado del problema, pero a partir de la corrección de errores y de la experiencia recolectada durante la realización de los experimentos, se pudo observar como esta es capaz de entender y dar una respuesta coherente a partir de indicaciones poco técnicas. Entonces, se puede sostener que no es necesario tener altos conocimientos del lenguaje de programación en el que se está solicitando una implementación para poder interactuar de forma efectiva con la herramienta y obtener una respuesta satisfactoria de la misma, al menos esto es cierto cuando los problemas a tratar son “sencillos” en términos generales.

Como se mencionó anteriormente, pueden surgir complicaciones ya que la herramienta no es capaz de encontrar errores concretos o realizar cambios muy específicos por sí sola sin las indicaciones adecuadas.

VI. CONCLUSIONES

A partir de los experimentos realizados, se puede observar que *ChatGPT* posee la capacidad de resolver las evaluaciones planteadas a los estudiantes con creces. En la mayoría de los casos, solamente brindando como entrada el enunciado del problema, las respuestas obtenidas superan todos las pruebas automatizadas.

Se evidencia, por lo tanto, que pelagra la integridad y fiabilidad de estos métodos de evaluación, ya que un estudiante sin conocimiento alguno de programación puede aprobar el examen, incluso con nota sobresaliente en una gran cantidad de casos.

Se intuye que el mal uso de esta herramienta como por ejemplo, resolver exámenes de manera ilegítima u obtener soluciones completas a problemas sin tratar de interpretarlas y entenderlas, puede tener consecuencias negativas en el aprendizaje de la programación, especialmente en las etapas tempranas. Teniendo en cuenta que el objetivo de este tipo de asignaturas introductorias es brindar al estudiante conocimientos básicos de la disciplina, como la sintaxis del lenguaje y el pensamiento lógico. Estos son aspectos que *ChatGPT* resuelve de muy buena manera quitándole la posibilidad al alumno de desarrollarlas por sí mismo.

Se considera necesario que los docentes y responsables académicos sean conscientes de esta herramienta y sus posibilidades, ya que su abuso por parte de los estudiantes puede ser perjudicial para su aprendizaje y formación.

Como se expuso en este artículo, si bien la efectividad de esta tecnología es muy alta en ejercicios iniciales, en ocasiones se equivoca, es en este apartado donde los conocimientos del usuario toman un papel más preponderante, dado que, para solicitarle una corrección efectiva a la herramienta, es necesario cierto grado de conocimientos técnicos. Esto se debe a que la mejor

alternativa para corregir el error es indicarle al chat el lugar concreto donde se encuentra el mismo o al menos exactamente en que se está equivocando.

Con estas observaciones no se busca menoscabar a *ChatGPT* y su aplicación en la enseñanza y aprendizaje de la programación. Es destacable el trabajo de explicación que realiza la IA al momento de otorgar una respuesta, ya que no devuelve simplemente el código fuente del programa a implementar, sino que este se encuentra comentado y acompañado de una descripción sobre el desarrollo y funcionamiento del mismo. Además, es posible consultarle por dudas concretas y que este indique de qué manera abordarlas, o bien, usarlo como herramienta de validación de las soluciones desarrolladas por uno mismo, entre otras.

Esto describe la aplicación *ChatGPT* como una herramienta útil para ser usada como tutor o asistente que ayude al aprendizaje de los temas de una forma mucho más rápida que una búsqueda ordinaria por internet o tener que esperar a que un docente esté disponible para consulta. Por tanto, si todo se produce bajo el supuesto de que el estudiante haga un uso honesto de la misma, esta herramienta resulta de utilidad para promover el crecimiento intelectual.

De todas formas, hay que considerar que *ChatGPT* puede no dar explicaciones acertadas, por esta razón, es necesaria una mirada crítica que permita identificar errores y actuar en consecuencia.

Como continuación del presente trabajo, se buscará obtener resultados más fiables y representativos. Actualmente, la muestra de estudio todavía no es significativa y no se ha experimentado como influencia el factor humano en el proceso de corrección de las evaluaciones. Por lo tanto, se cree necesario realizar una comparación entre el desempeño de *ChatGPT* y estudiantes reales de la materia analizada, como así también la forma en que ellos responden y se desenvuelven mediante la utilización de la herramienta.

La inclusión de los docentes en el análisis es sumamente relevante para poder evaluar su capacidad de distinguir entre respuestas realizadas por los estudiantes y las generadas con la herramienta.

El desafío de la investigación en curso consistirá entonces en experimentar y visualizar las ventajas y desventajas que trae consigo la IA, generalizando para un mayor conjunto de herramientas, y de este modo visualizar con mayor certidumbre cómo influye el cambio disruptivo de paradigma provocado por la inteligencia artificial en la enseñanza de la programación y en la educación en general.

REFERENCIAS

- [1] González Arencibia, M., & Martínez Cardero, D. (2020). Dilemas éticos en el escenario de la inteligencia artificial. *Economía y Sociedad*, 25(57), 93-109.
- [2] Hughes, A. (2023). *ChatGPT: Everything you need to know about OpenAI's GPT-3 tool*. BBC Science Focus.
- [3] OpenAI © 2015–2023 <https://openai.com/about>
- [4] Jofre, C. M. *ChatGPT, Inteligencia Artificial y Universidad*.
- [5] Qadir, J. (2022). Engineering education in the era of ChatGPT: Promise and pitfalls of generative AI for education.

- [6] Muhammad Shidiq. (2023). THE USE OF ARTIFICIAL INTELLIGENCE-BASED CHAT-GPT AND ITS CHALLENGES FOR THE WORLD OF EDUCATION; FROM THE VIEWPOINT OF THE DEVELOPMENT OF CREATIVE WRITING SKILLS.
- [7] Mhlanga, David, Open AI in Education, the Responsible and Ethical Use of ChatGPT Towards Lifelong Learning (February 11, 2023).
- [8] Juca-Maldonado, F. (2023). El impacto de la inteligencia artificial en los trabajos académicos y de investigación. Revista Metropolitana de Ciencias Aplicadas, 6(S1), 289-296.
- [9] Carrasco, J. P., García, E., Sánchez, D. A., Porter, E., De La Puente, L., Navarro, J., & Cerame, A. (2023). ¿ Es capaz “ChatGPT” de aprobar el examen MIR de 2022? Implicaciones de la inteligencia artificial en la educación médica en España. Revista Española de Educación Médica, 4(1).
- [10] Gilson A, Safranek C, Huang T, Socrates V, Chi L, Taylor R, Chartash D. How Does ChatGPT Perform on the United States Medical Licensing Examination? The Implications of Large Language Models for Medical Education and Knowledge Assessment.
- [11] de Winter, Joost. (2023). Can ChatGPT pass high school exams on English Language Comprehension?.
- [12] Kamil Malinka y Martin Perešini, Anton Firc, Ondřej Hujňák y Filip Januš. (2023). On the Educational Impact of ChatGPT: Is Artificial Intelligence Ready to Obtain a University Degree?.
- [13] Lobb, R., & Harlow, J. (2016). Coderunner: A tool for assessing computer programming skills. ACM Inroads, 7(1), 47-51.