



Programación competitiva como estrategia de enseñanza aprendizaje en módulos introductorios de programación

Competitive programming as a teaching-learning strategy in introductory programming modules

Programação competitiva como estratégia de ensino-aprendizagem em módulos introdutórios de programação

ARTÍCULO ORIGINAL



Escanea en tu dispositivo móvil
o revisa este artículo en:

<https://doi.org/10.33996/revistahorizontes.v9i38.1044>

Ariel Ramiro Mamani Tola 

arielr.mat@gmail.com

Universidad Privada Domingo Savio. Potosí, Bolivia

Artículo recibido 3 de mayo 2023 | Aceptado 8 de junio 2025 | Publicado 1 de abril 2025

RESUMEN

La enseñanza de la programación, en carreras como la Ingeniería de Sistemas y la Ingeniería Informática, enfrenta retos crecientes en Bolivia, donde la industria de software demanda profesionales altamente capacitados. El objetivo del estudio es evaluar el impacto de la incorporación de estrategias de programación competitiva en la enseñanza de módulos de programación en carreras de Ciencias de la Computación, específicamente en los módulos de Programación. Se seleccionó una muestra intencionada de 60 estudiantes. Se utilizaron registros observacionales y entrevistas semiestructuradas para recoger información. Los resultados que la programación competitiva en la educación superior mejora la motivación y fortalece las habilidades del razonamiento lógico. Facilitando su preparación para competencias nacionales e internacionales como la International Collegiate Programming Contest (ICPC). Se concluye que integrar elementos de competencias de programación universitaria boliviana contribuye significativamente a mejorar el proceso de enseñanza-aprendizaje y prepara a los futuros profesionales para los desafíos del mercado tecnológico local y global.

Palabras clave: Aprendizaje; Enseñanza; Programación Competitiva; Tecnología

ABSTRACT

The teaching of programming in programs such as Systems Engineering and Computer Engineering faces growing challenges in Bolivia, where the software industry demands highly trained professionals. The objective of this study is to evaluate the impact of incorporating competitive programming strategies in the teaching of programming modules in Computer Science programs, specifically in the Programming modules. A purposive sample of 60 students was selected. Observational records and semi-structured interviews were used to collect data. The results show that competitive programming in higher education improves motivation and strengthens logical reasoning skills, facilitating their preparation for national and international competitions such as the International Collegiate Programming Contest (ICPC). It is concluded that integrating elements of Bolivian university programming competitions significantly contributes to improving the teaching-learning process and prepares future professionals for the challenges of the local and global technology market.

Key words: Learning; Teaching; Competitive Programming; Technology

RESUMO

O ensino de programação em cursos como Engenharia de Sistemas e Engenharia da Computação enfrenta desafios crescentes na Bolívia, onde a indústria de software demanda profissionais altamente qualificados. O objetivo deste estudo é avaliar o impacto da incorporação de estratégias de programação competitiva no ensino de módulos de programação em cursos de Ciência da Computação, especificamente nos módulos de Programação. Uma amostra intencional de 60 alunos foi selecionada. Registros observacionais e entrevistas semiestructuradas foram utilizados para a coleta de dados. Os resultados mostram que a programação competitiva no ensino superior melhora a motivação e fortalece as habilidades de raciocínio lógico, facilitando a preparação para competições nacionais e internacionais, como o Concurso Internacional de Programação Universitária (ICPC). Conclui-se que a integração de elementos das competições universitárias de programação bolivianas contribui significativamente para a melhoria do processo de ensino-aprendizagem e prepara os futuros profissionais para os desafios do mercado de tecnologia local e global.

Palavras-chave: Aprendizagem; Ensino; Programação Competitiva; Tecnologia

INTRODUCCIÓN

Las ciencias de la computación han sido uno de los campos científicos que más rápido ha evolucionado desde finales del siglo XX y a lo largo del siglo XXI, impulsadas por avances tecnológicos que han generado nuevos paradigmas y aplicaciones informáticas en áreas como las aplicaciones móviles, la computación en la nube, el big data y, más recientemente, la inteligencia artificial (IA) (Ourense na Rede, 2024). Este contexto tecnológico demanda profesionales cada vez más competitivos y versátiles, capaces de adaptarse a las exigencias de una industria en constante transformación y alta competitividad (Galileo, 2023).

Por lo tanto, el proceso de enseñanza y aprendizaje en módulos introductorios de programación debe actualizarse para incorporar estos avances tecnológicos, sin perder la esencia formativa que permite a los estudiantes desarrollar competencias sólidas en ciencias de la computación. La programación, como disciplina fundamental dentro de esta área, es reconocida como una de las más desafiantes para los estudiantes, debido a su naturaleza teórico-práctica y a la necesidad de integrar conceptos matemáticos con habilidades de resolución de problemas (Piteira y Costa, 2012).

Históricamente, la enseñanza de la programación se basaba principalmente en métodos teóricos, donde el diseño de algoritmos

se resolvía con lápiz y papel, relegando la práctica computacional a un papel complementario (Piteira y Costa, 2012). Sin embargo, este enfoque resulta insuficiente frente a las demandas actuales, que requieren profesionales con habilidades tanto teóricas como prácticas, capaces de enfrentar retos complejos en entornos reales de desarrollo de software (Galileo, 2023).

Si bien hoy en día los programas académicos han sido diseñados acorde a la realidad actual de la programación y el desarrollo de software, es en la aplicación de dichos programas en aula donde queda en evidencia una brecha entre la forma en que se desarrolla la programación y lo que las industrias de software esperan de un programador. Aunque el campo laboral de un profesional en ciencias de la computación y sus áreas de aplicación como la Ingeniería de Sistemas o la Ingeniería Informática se extiende más allá del desarrollo de software y aplicaciones, es innegable que la programación contribuye al desarrollo del pensamiento lógico y mejora las habilidades de resolución de problemas (Ramírez y Yudy, 2019).

Por otra parte, la importancia de la programación trasciende la currícula educativa en las áreas específicas de las ciencias de la computación, y se ha convertido en una competencia profesional cada vez más demandada por empresas en diversos sectores, incluso en la educación secundaria, la programación se considera una materia más, en la que los

estudiantes, más allá de aprender a construir programas, utilizan herramientas sencillas basadas en programación (por ejemplo; Scratch) para fortalecer sus habilidades lógicas y de resolución de problemas (Vázquez y Ferrer, 2015).

En ese sentido, carreras como la Ingeniería de Sistemas y la Ingeniería Informática, al ser áreas de aplicación de las ciencias de la computación, figuran entre las más demandadas por los bachilleres del sistema educativo boliviano. Al respecto, el boletín informativo del Comité Ejecutivo de la Universidad Boliviana (CEUB) sitúa el área de conocimiento de la Ingeniería y Tecnología entre los primeros lugares de preferencia para los nuevos postulantes a las diferentes universidades (CEUB, 2022).

Esta evidente demanda respecto a las carreras relacionadas al desarrollo tecnológico es generalizada, tanto en Universidades públicas y privadas, donde la Ingeniería de Sistemas y la Ingeniería Informática acogen cada vez a más estudiantes quienes aspiran a convertirse en profesionales competentes, con las habilidades necesarias para poder hacer frente a los desafíos de una sociedad cada vez más dependiente de la tecnología.

Por otro lado, la innata naturaleza competitiva del ser humano le ha llevado a utilizar la competencia como un mecanismo para mejorar sus habilidades, y la programación no queda exenta de esa realidad. En este contexto,

en 1970 surgió el concepto de 'Competencia de programación' dentro de las aulas de la Universidad de Texas, que posteriormente, en 1977, dio origen a la competencia más importante y prestigiosa de programación a nivel mundial, la ACM-ICPC (Association for Computing Machinery - International Collegiate Programming Contest) actualmente conocida simplemente como ICPC. Para la región de Latinoamérica, esta competencia se concentra en un evento único denominado "ICPC Programadores de América", que funciona como clasificatoria para el mundial de programación (Wikipedia, 2024; EcuRed, 2024).

Si bien en un principio esta disciplina mental surgió como una forma de motivar a los estudiantes de programación a mejorar sus habilidades a través de la competencia, hoy en día se ha consolidado como una referencia fundamental para la industria del software, donde las empresas buscan talentos capaces de construir soluciones informáticas óptimas (Ripoll, 2017).

Este fenómeno se explica en gran medida porque los estudiantes que se preparan para estas competencias desarrollan, durante su entrenamiento, habilidades clave tales como: razonamiento lógico avanzado para la solución de problemas, dominio de diversos tipos de algoritmos y su aplicación práctica, capacidades superiores en la depuración de código, mayor velocidad y precisión en la escritura, manejo fluido de múltiples lenguajes de programación,

capacidad para responder eficazmente bajo presión, y optimización de soluciones algorítmicas para mejorar su eficiencia temporal y espacial (Troomes, 2024; Cubadebate, 2023).

En cuanto al contenido de la competencia de programación ICPC, algunos de los tópicos considerados son la algoritmia, estructuras de datos y librerías, búsqueda completa, técnicas divide y vencerás, algoritmos greedy, programación dinámica, teoría de grafos, teoría matemática y procesamiento de cadenas (Halim y Halim, 2013).

Aunque muchos de estos temas son abordados en diferentes asignaturas dentro de la currícula de Ingeniería de Sistemas e Informática, no siempre son desarrollados de forma que permitan al estudiante una preparación adecuada para competencias de programación. Si bien participar en estas competencias no es un objetivo principal para todos los profesionales en ciencias de la computación, es evidente que hacerlo mejora significativamente sus habilidades de programación, convirtiéndolos en profesionales competentes.

Considerando lo anterior, la incorporación de componentes del entrenamiento en programación competitiva en módulos introductorios de programación ofrece amplias posibilidades para potenciar el desarrollo del potencial de los estudiantes en Ciencias de la computación. En este sentido, diversas universidades de la región han

orientado sus esfuerzos hacia las competencias de programación como estrategia para fortalecer las habilidades de sus futuros profesionales tanto en Ingeniería de Sistemas como en Ingeniería Informática. Ejemplos de ello, se encuentran estudios como “Uso de la programación competitiva como aporte a la enseñanza de la Lógica Algorítmica. Experiencia Universidad de la Amazonia” (Molina et al., 2016) y “Programación competitiva como estrategia didáctica para potenciar el aprendizaje colaborativo en estudiantes de ingeniería” (Lora et al., 2020).

La relevancia del presente estudio radica en su aporte a la comunidad científica, dado que se ha evidenciado una carencia de investigaciones en este ámbito. Además, desde una perspectiva práctica, sienta las bases para futuras investigaciones y la posibilidad de ampliar los elementos evaluativos al aplicar la programación competitiva como estrategia didáctica.

Por lo tanto, el objetivo principal de este estudio es evaluar el impacto de la incorporación de estrategias de programación competitiva en la enseñanza de módulos de programación en carreras de ciencias de la computación, específicamente en Programación I y II, con el fin de mejorar la motivación, las habilidades de razonamiento lógico y la capacidad de resolución de problemas de los estudiantes, preparándolos para competencias como la ICPC y su inserción en el mercado laboral.

MATERIALES Y MÉTODO

Este estudio se enmarca en un enfoque mixto, combinando métodos cualitativos y cuantitativos, orientado a la investigación aplicada y descriptiva. El propósito principal fue analizar la implementación de estrategias de programación competitiva en módulos de enseñanza en carreras de Ciencias de la Computación, en los módulos de Programación I y II, con el fin de evaluar su impacto en la motivación y las capacidades de los estudiantes en la resolución de problemas algorítmicos.

El diseño del estudio fue de tipo exploratorio e interventivo, ya que se aplicaron las estrategias de programación competitiva durante varios períodos académicos (marzo 2021, marzo 2022 y enero 2023). Este diseño permitió no solo observar sino modificar activamente el proceso formativo mediante la introducción de estrategias de programación competitiva. En cada ciclo, se incorporaron elementos como problemas de programación, uso del juez BOCA, y el entorno Linux, lo que permitió observar cambios en las capacidades y actitud de los participantes respecto al aprendizaje de la programación.

La población objeto de estudio estuvo conformada por 120 estudiantes inscritos en los módulos de Programación I y II en una universidad boliviana ubicada en Potosí. De esta población, se seleccionó una muestra intencionada de 60 estudiantes, basada en criterios

de participación activa y asistencia regular a clases y actividades complementarias. Esta muestra incluyó estudiantes de distintas gestiones y niveles de rendimiento previo, buscando representar la diversidad del grupo y garantizar la validez del análisis.

Para la recogida de información, se utilizaron diversas técnicas e instrumentos, entre ellas: (i) registros observacionales durante el desarrollo de las actividades, que permitieron evaluar la participación y el rendimiento de los estudiantes; (ii) la aplicación de cuestionarios y entrevistas semiestructuradas diseñadas para medir su motivación, percepción y autoevaluación de sus habilidades previas y mejoradas; y (iii) el análisis de los resultados obtenidos en las tareas de programación y en las competencias ICPC en las que los estudiantes participaron posteriormente. Además, se realizó una revisión documental sobre el uso de la programación competitiva como estrategia didáctica en ciencias de la computación.

Para la recopilación de datos, se emplearon técnicas e instrumentos como registros observacionales durante las sesiones, cuestionarios y entrevistas semiestructuradas para medir motivación y autoevaluación de habilidades, análisis de resultados en tareas de programación y en las competencias ICPC, y entrevistas abiertas con docentes y estudiantes. La población objeto de estudio estuvo conformada por 120 estudiantes inscritos en los módulos de Programación I

y II, de los cuales se seleccionó una muestra intencionada de 60 estudiantes, mediante criterios de participación activa y asistencia regular. La muestra abarcó estudiantes de distintas gestiones y niveles de rendimiento previo.

Las dimensiones analizadas incluyeron la motivación en el aprendizaje de la programación, la capacidad de razonamiento lógico y resolución de problemas, el nivel de habilidades técnicas en algoritmia y programación, la persistencia en la participación en competencias como ICPC, y las brechas entre el contenido impartido en los módulos y los requerimientos de estas.

El análisis cuantitativo incluyó estadística descriptiva y pruebas de comparación de medias para evaluar cambios significativos, mientras que el análisis cualitativo se basó en codificación y categorización sistemática de respuestas, asegurando triangulación y confiabilidad.

La metodología aplicada consistió en la implementación, durante los períodos académicos de marzo 2021, marzo 2022 y enero 2023, de estrategias relacionadas con la programación competitiva en los módulos de Programación I y II. Estas estrategias incluyeron el uso de problemas del juez BOCA, actividades en entornos Linux y sesiones de entrenamiento en resolución de problemas algorítmicos.

RESULTADOS Y DISCUSIÓN

Proponer y emplear estrategias en el proceso de enseñanza-aprendizaje de módulos de

programación debe partir de objetivos educativos claros y de una concepción pedagógica sólida. En este estudio, los fundamentos del aprendizaje desarrollador se asumen como eje central para diseñar una estrategia que garantice la formación integral de futuros profesionales en Ciencias de la Computación. Según Zilberstein et al. (1999): el proceso de enseñanza aprendizaje desarrollador constituye la vía mediatizadora esencial para la apropiación de conocimientos, habilidades, normas de relación emocional, de comportamiento y valores, legados por la humanidad, que se expresan en el contenido de enseñanza, en estrecho vínculo con el resto de las actividades docentes y extradocentes que realizan los educandos. Las estrategias educativas por tanto deben contribuir a conformar y a reafirmar en los estudiantes, los hábitos de auto preparación, favorecer su disposición a incrementar constantemente sus conocimientos y a motivar su actividad creadora con el fin de desarrollar su capacidad para resolver los problemas prácticos (Molinet et al., 2022)

El rol de la programación competitiva

La programación competitiva se alinea perfectamente con estos principios, pues fomenta la auto preparación y la capacidad de resolver problemas prácticos. A través de la resolución de problemas con un contenido guiado y preparado así como la técnica de programación por pares y en equipos ya que una cualidad de algunas competencias de programación como ser la ICPC,

una característica es el trabajo en equipo ya que a los participantes se les presenta un set o conjunto de problemas complejos para que lo comprendan y planteen una solución algorítmica, por lo general, es necesario de diferentes puntos de vista para plantear soluciones a los problemas y presentar un programa eficiente. Es en esa etapa de la competencia donde un adecuado trabajo colaborativo cobra relevancia ya que todos los integrantes del equipo deben aportar, con sus ideas de solución a una construcción integrativa de la solución.

Características de los problemas de la ICPC

Una cualidad destacable de la preparación de los estudiantes en programación competitiva, es el uso de los problemas auténticos presentados

en competencias como la ICPC. Estos problemas no son típicos ejercicios de libros de texto, sino desafíos que integran razonamiento lógico y habilidades de programación en escenarios ficticios pero que podrían ser aplicados en problemas reales de la sociedad. Además, la diversidad temática de los problemas exige a los estudiantes aplicar conocimientos multidisciplinarios, enfrentando verdaderos retos de programación más allá de una sola temática. A modo ilustrativo, la figura 1, muestra un ejemplo de un problema utilizado en la competencia de programación ICPC “Latin American Regional – 2022” que es clasificatoria al mundial de programación. En esta fase, los problemas se presentan en inglés, lo que añade un componente adicional de desafío para los participantes.

ICPC Latin American Regional – 2022

9

Problem E – Empty Squares

Martín has a board of $1 \times N$ squares. He also has N tiles of $1 \times 1, 1 \times 2, \dots, 1 \times N$ squares, one of each type. He has placed one of the tiles on the board. His friend, Nico, wants to place some of the remaining tiles in such a way that as many squares as possible are covered. How many squares will remain empty if he succeeds?

Tiles placed on the board cannot overlap each other. Besides, each placed tile must be located completely within the board and must cover whole squares.

Input

The input consists of a single line that contains three integers N ($1 \leq N \leq 1000$), K ($1 \leq K \leq N$) and E ($0 \leq E \leq N - K$), indicating that the board has $1 \times N$ squares, and a tile of $1 \times K$ squares is placed leaving E empty squares to its left.

Output

Output a single line with an integer indicating the number of squares that will remain empty if Nico covers as many squares as possible with the remaining tiles.

Sample input 1	Sample output 1
6 2 2	3
Sample input 2	Sample output 2
1000 1 1	1

Figura 1. Problema de programación competitiva usado en la competencia ICPC.

Evaluación automatizada y el Juez BOCA

Otra característica esencial de la ICPC es el uso de sistemas de evaluación automatizada, conocidos como jueces en línea o jueces automáticos, se muestra en la figura 1, que no es más que un “sistema de evaluación de programas automatizados, los cuales también se conocen como jueces en línea o jueces automáticos. Este tipo de jueces evalúan los programas que realizan los equipos durante el concurso. La evaluación es automática e inmediata. Cada uno de los equipos envía la solución de los problemas a través de este juez automático y recibe una calificación de forma casi inmediata (Gutiérrez, et al., 2015). Entre los jueces más reconocidos se encuentra el Contest Management System (CMS) utilizado en la Olimpiada de Informática OIO, orientado a estudiantes de secundaria.

En el caso de la ICPC, el sistema de evaluación empleado en etapas nacionales y latinoamericanas es el juez BOCA. Este sistema web fue desarrollado para la Maratona de la Programación de la Sociedad Brasileña de Computación y en la actualidad se utiliza en numerosas regiones, incluyendo todas las universidades del sistema universitario boliviano (de Campos y Ferreira, 2004).

La Figura 2 presenta una captura del scoreboard generado por el juez BOCA, donde se visualiza la cantidad de problemas resueltos y evaluados automáticamente, lo que permite un seguimiento objetivo y transparente del desempeño de los equipos durante la competencia.

#	User/Site	Name	A	B	C	D	E	F	G	H	I	J	K	L	M	Total
1	teamsoar001/AR	[UBA - FCEN] una ma y no incho ma	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	12 (1485)
2	teambrr005/BR	[UFRJ] Lebenslangerschicksalsschatz	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	11 (1431)
3	teambrr002/BR	[UFMG] Humuhumunukunuaapua'a	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	11 (1514)
4	teammoco01/CO	[UNAL Bogotá] phiUN	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	10 (1339)
5	teambrr001/BR	[UNICAMP] Você beijaria Matheus Leal Viana?	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	10 (1346)
6	teambrr003/BR	[UNICAMP] Inimigos do Hungaro	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	9 (873)
7	teamsoar005/AR	[UBA - FCEN] Prim Floyd	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	9 (1182)
8	teamsoch014/CL	[PUC-Ing] Laranjas.clear()	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	9 (1425)
9	teamcbcu001/CU	[UH] UH Top	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	1/100	9 (1445)

Figura 2. Captura del scoreboard generado por el juez BOCA.

Entorno de programación en la ICPC

Un aspecto fundamental de la competencia ICPC es el entorno de programación estandarizado, ya que garantiza una competencia equitativa, todos los participantes utilizan una computadora por equipo con el mismo hardware y la misma configuración de software donde se utiliza por completo aplicaciones de software libre, es decir con una licencia “Open Source”. Las características principales del entorno de programación son:

- SO: Ubuntu 22.04.1 LTS Linux (64-bit).
- Desktop: GNOME
- Editores de código: vi/vim, gvim, emacs, gedit, geany, kate
- Lenguajes de programación:
 - Java (Openjdk version 17.0.5 2022-10-18)
 - C (gcc 11.3.0 (Ubuntu 11.3.0-1ubuntu1 22.04))
 - C++ (g++ 11.3.0 (Ubuntu 11.3.0-1ubuntu1~22.04))
 - Python 3 (PyPy 7.3.10 con GCC 11.3.0)

Para cada lenguaje, si la compilación es exitosa, el sistema ejecuta el archivo generado para producir el resultado que será evaluado automáticamente. Este entorno, además de ser robusto y confiable, refleja el compromiso de la ICPC con el uso de tecnologías abiertas y accesibles para todos los participantes.

Elementos clave en la formación basada en programación competitiva

En la actualidad, la implementación de la programación competitiva en módulos educativos suele centrarse en tres elementos principales:

- Uso de problemas con características de la competencia ICPC
- Uso del juez virtual BOCA para las actividades evaluativas
- Uso del entorno de programación de la ICPC

Diseño de problemas con enfoque competitivo

Es importante, resaltar que los problemas debe ser diseñados desde cero, es decir, creados a partir de un análisis del contenido del módulo de programación que se debe abordar y la estructura de programación que se desea trabajar. Por ejemplo, un ejercicio tradicional sobre estructuras repetitivas podría ser:

“Construya un programa para sumar los números pares del 1 al 10”

En este caso, el enunciado del ejercicio muestra tanto el problema como su solución, y en este caso, el estudiante solo debe repetir el desarrollo de una estructura repetitiva de forma casi mecánica

sin construir ningún razonamiento simplemente repetir como el docente explico el desarrollo de una estructura repetitiva.

Este tipo de ejercicios ocasiona que el estudiante tenga la impresión de que ha dominado la estructura repetitiva cuando solamente ha

aplicado el mismo proceso mediante el cual el docente ha desarrollado la estructura repetitiva sin entender realmente lo que significa una estructura repetitiva. Por otro lado, un ejercicio con temática similar pero con características de un problema de programación competitiva sería así, Figura 3.

Conejos

Archivo: conejos.cpp

Los conejos (de fibonacci) es un problema muy interesante que dice así: ¿Cuántas parejas de conejos tendremos a fin de año, si comenzamos con una pareja que produce cada mes otra pareja que procrea a su vez a los dos meses de vida? Considerando las siguientes reglas:

- En primer lugar, tenemos una pareja de conejos el primer mes.
- El segundo mes, la pareja envejece (todavía no procrea)
- El tercer mes, la pareja procrea otra pareja, o sea que ya tenemos dos.
- El cuarto mes, la pareja más vieja vuelve a procrear, mientras que la segunda envejece. En total, tenemos 3 parejas.
- El quinto mes, las dos parejas más viejas procrean de nuevo, y la tercera envejece. En total, tenemos $3+2=5$
- El sexto mes, las tres parejas más viejas procrean, y las dos más nuevas envejecen, de manera que tenemos $5+3=8$.

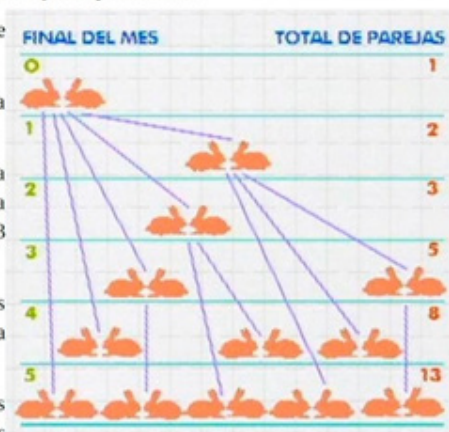


Figura 3. Ejercicio de programación usado en un módulo de Programación I.

Problemas de trasfondo y razonamiento lógico

En la Figura 3, el problema en cuestión, aunque sencillo en su solución técnica (solo el uso de operaciones aritméticas y estructuras repetitivas), está diseñado para que el estudiante identifique el trasfondo conceptual y aplique razonamiento lógico antes de programar. Este enfoque se fundamenta

en la teoría del aprendizaje significativo y desarrollador, que promueve la construcción activa del conocimiento a partir de la resolución de problemas auténticos y contextualizados. Así, el estudiante no solo replica procedimientos, sino que desarrolla habilidades cognitivas superiores, alineadas con las competencias profesionales requeridas en Ciencias de la Computación.

Evaluación automatizada: juez de programación

Para apoyar la formación en programación en estudiantes de Ciencias de la Computación es el uso de un juez de programación para la evaluación de los programas presentados por los estudiantes, en ese sentido, cuando un estudiante se enfrenta al juez de programación logra desarrollar dos habilidades fundamentales que lo convertirán en un buen programador:

Capacidad de Adaptación a las respuestas del juez.

El juez de programación al recibir un problema lo evaluó y presenta 5 posibles respuestas de las cuales solo una corresponde a la respuesta correcta y las demás respuestas pueden ser: El estudiante debe interpretar los distintos tipos de retroalimentación (respuesta incorrecta, tiempo límite excedido, error de presentación, error de compilación) y ajustar su código hasta lograr una solución correcta.

En todos los casos, el juez otorga un puntaje válido solo si la respuesta es correcta, en otro caso el puntaje es cero, en otras palabras, no se considera el procedimiento, solo el resultado. Esta forma de evaluar a pesar de ser algo estricta, representa en términos simples la mentalidad que debe tener un programador al construir soluciones informáticas, es decir, no hay soluciones a medias y un problema siempre se debe resolver por completo.

De esa forma si un estudiante no recibe la respuesta correcta, entonces debe adaptar su código para encontrar sus fallas y en algunos casos encontrar como optimizar ese código para recibir la respuesta correcta. En estos procesos el docente puede colaborar dando indicios de cuál es la falla del programa orientando a una solución correcta trabajando sobre la zona de desarrollo próximo del estudiante que plantea Vygostky.

Pro-actividad para anticiparse a los casos de prueba

La necesidad de anticipar múltiples escenarios y contraejemplos estimula el razonamiento lógico y la creatividad. Preguntas como “¿Qué pasa si la entrada es muy grande?” o “¿Qué sucede si los datos son atípicos?” promueven la capacidad de generalización y la transferencia de conocimientos, habilidades esenciales para la depuración y el aseguramiento de la calidad del software.

Este tipo de evaluación, aunque estricta, refleja la realidad profesional: en la industria, los programas deben ser correctos y eficientes, sin soluciones parciales. El enfoque fomenta la resiliencia y la mejora continua, competencias valoradas en el ámbito laboral.

Entorno de Trabajo: Software Libre y Linux

Finalmente, el último elemento considerado actualmente en la formación de programadores es el entorno de trabajo, aunque este elemento es

un poco más difícil de aplicarlo por el hecho de que considera el uso de software libre mediante un sistema operativo tipo Linux, y por lo general los laboratorios de computación siempre están preparados con software privativo (Windows), es de gran utilidad ya que el uso de software libre representa un desafío en sí mismo a los estudiantes ya que la diferencia de programar en un entorno Windows y un Linux es considerable por el mismo hecho de que Windows está diseñado para los usuarios comunes y en un proceso de formación de un futuro programador, no se lo puede considerar como un usuario más.

Por otro lado, los sistemas Linux están pensados más en desarrolladores y profesionales en cualquier área tecnológica incluidas las áreas de aplicación de las Ciencias de la Computación. Enfrentar a los estudiantes a un sistema operativo tipo Linux para que construyan sus programas, les da un mayor control sobre el programa y les permite comprender realmente cómo funciona un programa y como se ejecuta dentro del Sistema Operativo.

Es importante señalar que, aunque la estrategia basada en programación competitiva ha mostrado beneficios en los módulos iniciales de programación, aún queda pendiente su integración en módulos de programación avanzada o especializada. La literatura sugiere que la continuidad y progresión de estas estrategias a lo largo del currículo puede potenciar el desarrollo

de competencias profesionales de manera más integral.

Resultados de la aplicación de elementos de programación competitiva en la formación de programadores

La incorporación de elementos de programación competitiva en la formación de futuros programadores, no se puedan evaluar en algún periodo definido ya que independientemente de su aplicación, se debe considerar que el módulo en el que ha sido aplicado está regido por un plan académico con objetivos definidos, en ese sentido, la estrategia de aplicar elementos de programación competitiva se convierte en un apoyo para lograr los mencionados objetivos. Sin embargo, entrevistas semiestructuradas y observaciones empíricas realizadas al progreso de estudiantes que han participado en competencias de programación:

El 85% de los entrevistados manifestó una mejora significativa en la motivación hacia el aprendizaje de la programación. El 78% reportó mayor facilidad para asimilar contenidos complejos en módulos posteriores, como algoritmos avanzada y estructuras de datos.

Además, el 70% expresó que la preparación con elementos de programación competitiva les permitió enfrentar con mayor confianza las competencias ICPC y otras actividades extracurriculares. En este contexto, se han

logrado identificar algunos resultados que reflejan los beneficios de las estrategias descritas en su formación, algunos casos específicos son:

De los estudiantes participantes en la competencia de programación en las gestiones 2016-2018, que han sido los primeros en tener

preparación anticipada con las mencionadas estrategias, dos de ellos actualmente trabajan en empresas de software reconocidas a nivel nacional, la Figura 4, a continuación muestra el perfil en LinkedIn de una de las estudiantes participantes:



Figura 3. Perfil en LinkedIn de una estudiante participante en la competencia ICPC.

Durante las gestiones 2020-2022, los estudiantes participantes manifestaron que, tras su proceso de preparación y participación en competencias de programación, experimentaron una disminución significativa en las dificultades para asimilar contenidos de módulos posteriores, especialmente en áreas relacionadas con algoritmia avanzada y estructuras de datos.

En lo que se refiere a las gestiones más recientes, las mencionadas estrategias pudieron ser aplicadas en los módulos de Programación I y Programación II, en los periodos de marzo 2021, marzo 2022 y enero 2023, se evidenció una mejora en la motivación de los estudiantes al contenido impartido y también mejorando su capacidad de razonamiento lógico en la resolución de problemas. Los estudiantes que cursaron los mencionados módulos actualmente son participantes de la competencia ICPC y siguen con su proceso de formación.

Sin embargo, es necesario mencionar limitaciones de las estrategias implementadas. Actualmente, estas están diseñadas para el ciclo formativo inicial y se centran principalmente en la formación algorítmica básica y su aplicación práctica. Sin embargo, el temario completo de la competencia ICPC abarca contenidos más avanzados que van más allá de los módulos básicos de programación, incluyendo áreas como teoría de grafos, geometría computacional y optimización, entre otros. Esta diferencia genera una brecha importante respecto al potencial que representaría integrar el temario completo de la ICPC en la formación de futuros profesionales en Ciencias de la Computación, dado que no todos los módulos curriculares pueden cubrir la totalidad de dicho contenido.

Discusión

La programación competitiva, consolidada como un deporte mental y actividad formativa en carreras de Ingeniería de Sistemas e Ingeniería Informática en el sistema universitario boliviano, aún se encuentra parcialmente desvinculada del contenido formal de los módulos curriculares. Sin embargo, incorporar elementos del entrenamiento propios de la programación competitiva en la formación académica puede mejorar significativamente las habilidades de programación y razonamiento lógico en dichos estudiantes, mejorando además su preparación para competencias como la ICPC.

Estos hallazgos se alinean con los resultados reportados por Zambrana (2019), quién destaca que la metodología de programación competitiva fomenta valores fundamentales para la formación profesional, tales como el trabajo colaborativo, la gestión del estrés y la investigación constante de soluciones técnicas. Asimismo, se observó la mejora en la motivación y el razonamiento lógico, lo que coincide con lo señalado por Molina et al. (2016), quienes evidencian que la programación competitiva potencia el aprendizaje colaborativo y el desarrollo de habilidades algorítmicas en estudiantes de ingeniería.

De igual forma, aunque los resultados cuantificables de aplicar las estrategias descritas en el presente estudio no son verificables de forma inmediata, debido a la naturaleza formativa y a

largo plazo de la intervención, sin embargo, se identificó un impacto positivo en la percepción de los estudiantes sobre la facilidad para asimilar contenidos posteriores y en su motivación hacia la programación. Para muchos estudiantes que están ya en la etapa final de su formación como profesionales en Ciencias de la Computación, la simple participación en competencias de programación los hacen más visibles para programas de entrenamiento que tienen diversas empresas de software en Bolivia mejorando sus posibilidades de conseguir un trabajo como programadores en dichas empresas. Este hallazgo es consistente con la teoría del aprendizaje desarrollador (Zilberstein et al., 1999), que sostiene que la auto preparación y la resolución práctica de problemas fortalecen la capacidad cognitiva y la autonomía del estudiante.

Un aspecto relevante es la visibilidad que la participación en competencias de programación otorga a los estudiantes en el mercado laboral. Esto se pudo observar en casos documentados en Bolivia, estudiantes formados bajo esta metodología han logrado acceder a programas de entrenamiento en empresas de software reconocidas (Zambrana, 2019). No obstante, esta evidencia es cualitativa y anecdótica, lo que representa una limitación para la veracidad y generalización de los resultados, aspecto que debe ser abordado con estudios longitudinales y cuantitativos más rigurosos.

Finalmente, aunque la aplicación de estrategias de programación competitiva ha demostrado beneficios claros en las etapas iniciales de formación, persiste la necesidad de extender su implementación a áreas más especializadas como el desarrollo de sistemas y aplicaciones. En estos campos, la algoritmia es una base subyacente, pero el contenido técnico y práctico es más complejo y específico, lo que dificulta la integración directa de las estrategias competitivas sin una adaptación curricular adecuada.

Si bien la incorporación de elementos de programación competitiva representa una mejora significativa en las primeras fases de formación, su aplicación en módulos avanzados aún es limitada. Esto se debe a que, en áreas especializadas, la algoritmia queda subyacente a tecnologías y metodologías más técnicas y específicas, lo que requiere enfoques didácticos distintos que integren tanto la creatividad como la rigurosidad técnica.

En resumen, esta investigación contribuye a la comprensión del valor formativo de la programación competitiva en la educación superior en Bolivia, reafirmando la necesidad de un enfoque integral que articule competencias técnicas, cognitivas y profesionales para formar ingenieros de sistemas y computación más competentes y preparados para los retos del mercado global.

CONCLUSIONES

El proceso de formación de los estudiantes de las diferentes carreras en Ciencias de la Computación enfrenta muchos desafíos crecientes debido a la exigencia de una sociedad y una industria de software cada vez más competitiva. En ese contexto, la programación competitiva se posiciona como una herramienta estratégica que no solo contribuye a la mejora de habilidades técnicas, sino que también facilita la identificación y el reclutamiento de talentos por parte de empresas del sector.

La enseñanza de la programación requiere que los docentes trasciendan del contenido preparado y diseñado en el plan curricular, incorporando estrategias y alternativas innovadoras que integren elementos de competencias reales y prácticas. La inclusión de componentes propios de competencias de programación, como problemas desafiantes, la evaluación mediante jueces automáticos (BOCA), y el uso de entornos de programación en Linux, representa una oportunidad concreta para fortalecer la formación de los estudiantes en programación, volviéndolos más competentes y competitivos en el ámbito académico y profesional.

Los resultados del estudio evidencian que, la aplicación de estos elementos de programación competitiva, genera impactos positivos en la motivación y en la capacidad de razonamiento

lógico de los estudiantes durante los módulos introductorios de programación. Sin embargo, el verdadero alcance de esta estrategia se manifiesta a largo plazo, cuando los estudiantes consolidan sus habilidades superiores de resolución de problemas, adaptabilidad y desempeño bajo presión; competencias esenciales para enfrentar los retos de una industria del software en constante evolución.

CONFLICTO DE INTERESES. El autor declara que no existe conflicto de intereses para la publicación del presente artículo científico.

REFERENCIAS

- Comité Ejecutivo de la Universidad boliviana (CEUB). (2022). Boletín Estadístico CEUB (Comité Ejecutivo de La Universidad Boliviana, Relaciones públicas y Protocolo). https://ceub.edu.bo/doc/sndi/BOLETIN_estadistico2022.pdf
- Cubadebate. (2023). Concurso de programación competitiva ACM-ICPC. Recuperado de <http://www.cubadebate.cu/etiqueta/competencia-internacional-universitaria-de-programacion-acm-icpc/>
- De Campos, C., y Ferreira, C. (2004). BOCA: um sistema de apoio a competições de programação. *Sociedade Brasileira de Computação*. <https://research.tue.nl/files/134832343/decampos2004d.pdf>
- EcuRed. (2024). Competencia internacional universitaria de programación. Recuperado de https://www.ecured.cu/Competencia_internacional_universitaria_de_programacion
- Galileo. (2023, agosto 17). Las ciencias de la computación y su importancia en la actualidad. Recuperado de <https://www.galileo.edu/pdh/noticias/las-ciencias-de-la-computacion-y-su-importancia-en-la-actualidad/>
- Gutiérrez, L., Morales, D., y Martínez, F. (2015). LOS CONCURSOS DE PROGRAMACIÓN COMO DETONANTE DEL APRENDIZAJE. *REVISTA ELECTRÓNICA ANFEI DIGITAL*, 2. <https://anfei.mx/revista/index.php/revista/article/view/175/609>
- Halim, S., y Halim, F. (2013). Competitive Programming 3, the new lower bound of programming contests. lulu.com. <https://es.scribd.com/document/687854799/Competitive-Programming-3-the-New-Lower-Bound-of-Programming-Contests-by-Sтивен-Halim-Z-lib-org-1-300>
- Collegiate Programming Contest. ICPC. (2024). 2024 ICPC World Finals Astana Programming Environment. <https://docs.icpc.global/worldfinals-programming-environment/>
- Lora, G., Suaza, H., Rodríguez, P. y Taborda, G. (2020). Programación competitiva como estrategia didáctica para potenciar el aprendizaje colaborativo en estudiantes de ingeniería. *Revista Ibérica de Sistemas e Tecnologías de Informação*, 39. <https://dialnet.unirioja.es/servlet/articulo?codigo=8597366>
- Molina, Y., Espinosa, D., y Sánchez, E. (2016). Uso de la programación competitiva como aporte a la enseñanza de la Lógica Algorítmica. *Experiencia Universidad de la Amazonia. Innovación tecnológica para el desarrollo sustentable*, 26. <https://www.redalyc.org/pdf/5043/504373193012.pdf>
- Molinet, M., Lopez, L., y Ruiz, R. (2022). So de las tecnologías en el trabajo independiente y el perfeccionamiento del aprendizaje desarrollado. *Edumed Holguin*. <https://edumedholguin.sld.cu/index.php/edumedholguin22/2022/paper/download/297/169>
- Ourense na Rede. (2024, abril 26). Viaje a través de la historia de la computación desde los primeros ingenios hasta la revolución digital del siglo XXI. Recuperado de <https://ourensenarede.com/viaje-traves-historia-computacion-desde-los-primeros-ingenios-hasta-revolucion-digital-siglo-xxi/>

- Piteira, M. y Costa, C. (2012). Programación informática y programadores novatos. <https://doi.org/10.1145/2311917.2311927>
- Ramírez, W., y Yudy, A. (2019). Estrategia didáctica basada en TIC para enseñanza de programación: Una alternativa para el desarrollo del pensamiento lógico. *Universidad Pedagógica y Tecnológica de Colombia*. https://repositorio.uptc.edu.co/bitstream/001/2946/1/TGT_1548.pdf
- Ripoll, D. (2017). ACM-ICPC: Los muchachos que marcaron la historia. Recuperado de <https://campcarib.wordpress.com/2017/06/08/acm-icpc-los-muchachos-que-marcaron-la-historia/>
- Troomes. (2024). ¿Qué es la programación competitiva? Recuperado de <https://troomes.com/viewtopic.php?t=791>
- Vázquez E. y Ferrer, D. (2015). La creación de videojuegos con scratch en educación secundaria. *Ciudades creativas – cp*, 2015, n°6, pp. 63-73. ISSN 20146752. Girona (Catalunya). Recibido: 25/06/2015 - aceptado: 10/07/2015 n°6 63communication Papers –Media Literacy & Gender STUDIES. <https://dialnet.unirioja.es/servlet/articulo?codigo=5182831>
- Wikipedia. (2024). Competición Internacional Universitaria de Programación. Recuperado de https://es.wikipedia.org/wiki/Competici%C3%B3n_Internacional_Universitaria_de_Programaci%C3%B3n
- Zambrana Camberos, A. M. (2019). Programación competitiva y su aporte en la formación de ingenieros de sistemas. *Paradigma*, XIX, 73-77. Recuperado de <https://paradigma.emi.edu.bo/index.php/aa/article/download/25/28/56>
- Zilberstein, J., Portela, R., y MacPherson, M. (1999). *Didáctica integradora de las Ciencias*. Academia. <https://scholar.google.com/citations?user=gRD9i4YAAAAJ&hl=es>