

Evaluación de habilidades de pensamiento computacional al inicio de una asignatura de programación en una carrera de Informática

Gladys Dapozo, Cristina Greiner, Raquel Petris, Emanuel Irrazabal, Ana María Company, María Cecilia Espíndola, Yanina Medina

Departamento de Informática. Facultad de Ciencias Exactas y Naturales y Agrimensura
Universidad Nacional del Nordeste (UNNE), Av. Libertad 5450, 3400, Corrientes, Corrientes,
Argentina

{gndapozo, cgreiner, rpetris, eirrazabal, acompany, mcespindola, yanina}@exa.unne.edu.ar

Resumen

En este trabajo se presentan los resultados de la evaluación de habilidades de pensamiento computacional, tomadas de un test diagnóstico realizado al inicio de una asignatura de introducción a la programación. Los estudiantes que participaron de esta experiencia son ingresantes de una carrera de Informática. El dictado de esta asignatura ha regresado a la presencialidad plena y al inicio del curso se les tomó un test diagnóstico con actividades de resolución de problemas que los alumnos tenían que realizar en forma intuitiva, sin ninguna indicación adicional, más que el planteo de la actividad. Adicionalmente se recabó información acerca de la experiencia previa que tenían en programación. Los resultados obtenidos permitirán consolidar estrategias de enseñanza que se enfoquen en la adquisición de habilidades de pensamiento computacional y enfatizar la idea de que la programación no es una actividad meramente técnica, sino que requiere un conjunto de habilidades combinadas de resolución de problemas.

Palabras clave: Habilidades del Pensamiento Computacional. Enseñanza de la programación. Carreras de Informática

Introducción

Enseñanza de la programación

Diversos estudios [1] [2] [3] señalan que es evidente la producción limitada de graduados en ciencias, tecnología, ingeniería y matemática respecto de las necesidades del aparato productivo y, más significativamente, la falta

de masa crítica en capacidades necesarias para trabajar en la innovación digital, entre las que destaca la carencia estructural de capital humano de TIC (Tecnologías de la Información y la Comunicación).

Las carreras de informática contribuyen con la formación de estos recursos, sin embargo, enfrentan diversas dificultades que conducen a una insuficiente producción de profesionales. Entre ellas, se encuentra el desgranamiento temprano, particularmente en los primeros años de la universidad. En este periodo se inicia la enseñanza de la programación, siendo esta un área de estudio con la que la mayoría de los estudiantes no ha tenido un contacto previo, y, por tanto, el aprendizaje conlleva un mayor grado de dificultad [4]. A la hora de planificar el dictado de la programación inicial es común el planteo de diversos interrogantes sobre ¿cómo se enseña?, ¿por dónde se empieza?, ¿qué paradigma se debe utilizar?, ¿qué lenguaje de programación se debe emplear?, ¿cómo se deben orientar los ejercicios?, entre otras cuestiones [5].

En [6] se señala que la investigación existente ha llevado a muchas discusiones e ideas sobre la mejor manera de enseñar la programación introductoria, y de las dificultades que enfrentan los estudiantes en este tipo de cursos. Entre estas, el hecho de que los estudiantes consideran a la programación como una actividad puramente técnica en lugar de un conjunto de habilidades combinadas de resolución de problemas. Por lo tanto, la mayoría de los estudiantes que se inician en la programación tiende a desarrollar un conocimiento superficial y no crean estrategias

de resolución de problemas mediante el uso de construcciones de programación.

En [7] se propone una estrategia de enseñanza de programación en estudiantes universitarios que consiste en el desarrollo de un conjunto de actividades orientadas a estimular el pensamiento computacional mediante herramientas lúdicas con el objetivo de motivar a los alumnos y facilitar el aprendizaje de la programación. Los resultados indican que estas actividades contribuyeron positivamente a incrementar la motivación de los alumnos e incorporar una metodología de resolución de problemas que facilita la comprensión de los conceptos básicos de programación que forman parte de los contenidos de la asignatura.

Pensamiento computacional

Jeannette M. Wing, una de las primeras especialistas en utilizar el término de pensamiento computacional, lo definió de la siguiente manera: “el pensamiento computacional es el proceso de pensamiento involucrado en la formulación de problemas y sus soluciones para que las soluciones estén representadas en una forma que pueda ser efectivamente llevada a cabo por un agente de procesamiento de información” [8].

En otras palabras, se trata del proceso mental a través del cual una persona, planteado un problema, para su posible solución utiliza una secuencia de instrucciones ejecutadas por una computadora, un humano o ambos.

En [9] los autores señalan que el pensamiento computacional es un proceso cognitivo que implica el razonamiento lógico por el cual se resuelven los problemas, y los procedimientos y sistemas se entienden mejor. Abarca:

1. La capacidad de pensar de forma algorítmica: El pensamiento algorítmico es la capacidad de pensar en términos de secuencias y reglas como una forma de resolver problemas. Es una forma de llegar a una solución a través de una definición clara de los pasos.
2. La capacidad de pensar en términos de descomposición: La descomposición es una manera de pensar un problema en términos de sus partes y componentes. Cada parte

debe entenderse, solucionarse, desarrollarse y evaluarse por separado. Esto hace más fácil de resolver problemas complejos, y que los grandes sistemas sean más fáciles de diseñar. Luego las partes se integran para obtener la solución completa del problema.

3. La capacidad de pensar en generalizaciones, identificando y haciendo uso de patrones: La generalización se asocia con la identificación de patrones, similitudes y conexiones, y la explotación de las características similares. Es una forma de resolver rápidamente los nuevos problemas sobre la base de las soluciones de los problemas anteriores, y la construcción en experiencias previas.
4. La capacidad de pensar en términos abstractos, la elección de buenas representaciones: La abstracción es el proceso de hacer un problema más comprensible a través de la reducción de los detalles innecesarios. Una vez determinadas las características relevantes, se crea un “modelo” o representación del problema, que brinda una idea general del problema que se intenta resolver. Una parte fundamental de la abstracción es la elección de una buena representación de un sistema.
5. La capacidad de pensar en términos de evaluación: habilidad para determinar la eficacia y la eficiencia en el uso de recursos.

Lo mismos principios propone Selby [10] como habilidades de Pensamiento Computacional, estableciendo también criterios para desarrollarlos

- Generalización: la habilidad para expresar la solución de un problema en términos genéricos, la cual pueda ser aplicada a diferentes problemas que compartan características como el problema original.
- Descomposición: fraccionar a piezas más pequeñas, fáciles de resolver, partes de un problema.
- Abstracción: habilidad para decidir qué detalles de un problema son importantes y qué detalles se pueden omitir.
- Diseño Algorítmico: habilidad para crear un conjunto de instrucciones que indiquen paso a paso la solución de un problema.

- Evaluación: habilidad para reconocer y determinar los alcances de realizar procesos, en términos de eficiencia y uso de recursos.

El pensamiento computacional está relacionado con otros tipos de pensamiento: el matemático, el lógico y el crítico, entre otros, con los cuales comparte habilidades cognitivas comunes, como ser, reconocimiento de patrones, abstracción, modelado, repetición, entre otras. El pensamiento computacional favorece, a partir del reconocimiento de los aspectos que nos rodean, de problemas reales de las actividades diarias o de las ciencias, la propuesta de soluciones aplicando herramientas informáticas [11].

En [12] se ha realizado una evaluación de habilidades específicas del pensamiento computacional en estudiantes de nuevo ingreso de una carrera de tecnologías de la información con el objetivo de favorecer el desempeño académico de los estudiantes. Se seleccionaron cinco reactivos en correspondencia con las habilidades del pensamiento computacional (abstracción, generalización, descomposición, diseño algorítmico y evaluación) y establecieron una relación con los contenidos temáticos del curso, en base a lo cual determinaron fortalezas y debilidades de los estudiantes respecto de las habilidades del pensamiento computacional.

Por otra parte, la ciencia se desarrolla, crece y avanza por el creciente influjo de la Informática, que contribuye al manejo de cantidades masivas de datos permitiendo extraer conclusiones relevantes en cada área. Este fenómeno se observa aún en ciencias que hasta hace pocos años se habían mantenido al margen de estos desarrollos tecnológicos, como ser las ciencias sociales. El autor propone que, por su aporte a las herramientas mentales disponibles, es necesario incluir los fundamentos computacionales de la Informática entre las ciencias básicas en todos los niveles educativos. El pensamiento computacional representa así un componente irremplazable de las habilidades necesarias para comprender y desempeñarse en el mundo [13]. Todo esto evidencia la necesidad de alcanzar en los estudiantes la comprensión del uso de la

tecnología informática, no como un simple requisito para aprender un lenguaje de programación mediante la elaboración de algoritmos y la generación de código, sino que se busca la apropiación de un proceso metodológico adecuado para resolver problemas y programar.

Basado en los conceptos expuestos, en la asignatura Algoritmos y Estructuras de Datos I de la carrera Licenciatura en Sistemas de Información de la Facultad de Ciencias Exactas y Naturales y Agrimensura de la Universidad Nacional del Nordeste (UNNE), se propuso a los estudiantes la resolución de un conjunto de actividades vinculadas con las habilidades del pensamiento computacional propuestas por Selby [10], con el objetivo de detectar tempranamente los déficits y proponer actividades que refuercen esas habilidades. También se consideraron otras variables como por ejemplo la experiencia previa en programación y el género, a fin de relacionar el perfil del estudiante con sus habilidades computacionales, como una información más a tener en cuenta.

Metodología

Al inicio del dictado de la asignatura Algoritmos y Estructuras de Datos I (marzo 2022), se propuso a los estudiantes que concurrían a la primera clase de contenidos teóricos que respondieran un cuestionario impreso con preguntas acerca de su experiencia previa en programación y la resolución de 5 actividades problemáticas, que tenían que resolver auxiliados de papel y lápiz, y el resultado volcarlo al cuestionario. Se les pidió que resolvieran las actividades intuitivamente, sin darles ninguna indicación adicional.

A continuación, el diseño del cuestionario:

1. **Experiencia previa en programación**
(Marcar el ítem (uno solo) que considere más representativo)
 - 1.1. ¿Tuviste alguna experiencia de programación antes de ingresar a la Facultad?
 - a. Si, en la escuela secundaria o primaria
 - b. Si, estudiando por mi cuenta

- c. Si, en otra carrera (universitaria o terciaria)
 - d. No, ninguna experiencia previa
- 1.2. Si la respuesta anterior fue 1.1.a, ¿qué tipo de actividades realizaste?
- a. Programación con PilasBloques
 - b. Programación con Scratch
 - c. Otras herramientas de programación de tipo lúdica
 - d. Programación con lenguajes de programación (C, Java, Python, otros)

2. Test de habilidades de pensamiento computacional

Se consideraron 5 actividades que evalúan las habilidades de pensamiento computacional propuestas por Selby [10]:

- 1-Abstracción: Canguro
- 2-Diseño Algorítmico: Castores en movimiento
- 3-Generalización o detección de patrones: Espías
- 4-Evaluación: Salto de charcos
- 5- Descomposición: Móviles

- 2.1. **Canguro.** Hay 10 platos en una fila. Hay una manzana en cada plato. Al canguro Tomás le encanta saltar. Primero, el salta desde el plato más a la izquierda con la letra A. En cada salto después de la inicial, salta dos platos hacia adelante, o tres platos hacia atrás (ver ejemplo de la Fig. 1).



Figura 1. Canguro

Tomás sólo salta hacia platos con manzana. Cuando salta, come la manzana. *Pregunta:* Si Tomás recoge todas las 10 manzanas, ¿cuál manzana recoge al final? B, C, D, E, F, G, H, I, o J.

Respuesta correcta: I

- 2.2. **Castores en movimiento.** Una colonia de castores está viajando a través de un bosque oscuro. El camino es estrecho, así que tienen que viajar en una fila sin pasar

uno del otro. Algunas veces hay un hoyo en el camino. Un hoyo es cruzado de la siguiente manera:

- a. Primero saltan tantos castores como sean necesarios para llenar el hoyo.
- b. La colonia entera pasará entonces a través del hoyo.
- c. Los castores que saltaron treparán para salir del hoyo, y unirse al final de la línea.

La Fig. 2 muestra cómo 5 castores pasan un pequeño hoyo que se llena con 3 castores.

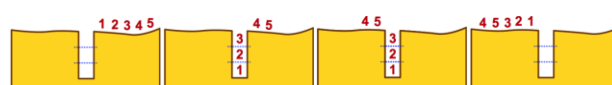


Figura 2. 5 castores que pasan por un hoyo

Una colonia de 7 castores pasa a través del bosque. Cruzan 3 hoyos. El primer hoyo se ajusta a 4 castores, el segundo se ajusta a 2, y el último hoyo se ajusta a 3 castores, como se puede observar en la siguiente figura:

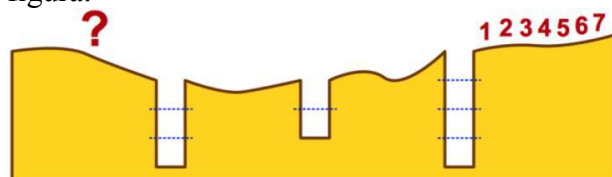


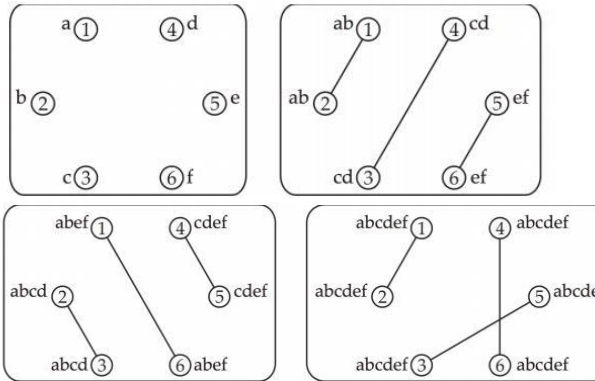
Figura 3. 7 castores cruzan 3 hoyos

Pregunta: ¿En qué orden se encontrarán los castores después de que hayan pasado el tercer hoyo?

Respuesta correcta: 2 1 6 5 3 4 7

- 2.3. **Espías.** Cada viernes, seis espías intercambian toda la información que han reunido durante la semana. Un espía nunca puede ser visto con más de otro espía al mismo tiempo. Así, tienen que tener varios encuentros de reunión en pares y compartir la información que poseen. El grupo de 6 espías sólo necesitan 3 encuentros para distribuir todos sus secretos. Antes del encuentro cada espía mantiene una sola pieza de información (espía 1 conoce 'a', espía 2 conoce 'b', etc.). En el primer encuentro espía 1 y 2 se encuentran y comparten información entonces ahora ambos conocen 'ab'. La Fig. 4 muestra

cuales espías se encuentran en cada reunión a través de una línea. También muestra cuáles piezas de información tienen todos. Después de 3 encuentros toda la información ha sido distribuida.



Pregunta: Después de un incidente internacional un espía ha dejado de atender los encuentros. ¿Cuál es el número mínimo de reuniones necesitadas por los cinco espías restantes para intercambiar toda la información?

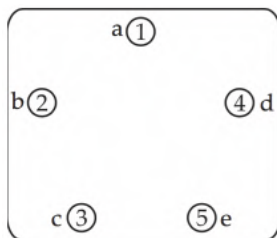


Figura 5. encuentro de 5 espías

Respuesta correcta: 4

2.4. **Salto de charcos.** Ana (edad 7), Berta (edad 8), Carlos (edad 9), Dora (edad 10) y Luisa (edad 11) están jugando un juego donde saltan de un charco a otro. Ellos han ubicado flechas entre los charcos, y todos inician del lado izquierdo como se indica en la Fig. 6.

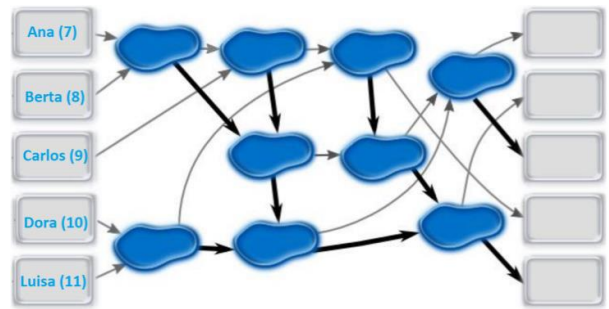


Figura 6. Salto de charcos

Cuando un niño salta dentro de un charco, espera la llegada de un segundo niño. Seguidamente el niño mayor en el charco saltará de acuerdo con la flecha gruesa, el más joven sigue la flecha delgada.

Pregunta: ¿Cuál es el orden (de arriba hacia abajo) en el cual los niños terminarán a la derecha?

Respuesta correcta: Berta, Dora, Carlos, Ana y Luisa (8-10-9-7-11)

2.5. **Móviles.** Un móvil es una pieza de arte que cuelga del techo, generalmente en los dormitorios. Un móvil consiste de varillas y figuras. Cada varilla tiene unos cuantos puntos donde figuras u otras varillas pueden ser atados. Además, cada varilla tiene un punto para colgar, donde se cuelga a una varilla hacia abajo (o hacia el techo). Un móvil puede ser descrito usando números y paréntesis.

Ejemplo: La expresión $(-3 (-1 1) (1 1)) (2 3)$ describe el móvil de la Fig. 7.



Figura 7. Móvil ejemplo

Pregunta: ¿Cuál de los móviles de la Fig. 8 puede ser construido usando las siguientes instrucciones? $(-3 (-1 4) (2 (-1 1) (1 1))) (2 (-1 6) (2 3))$

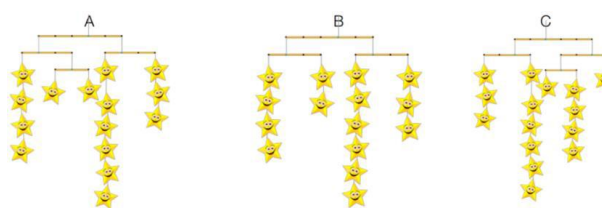


Figura 8. Móviles ejemplos

Respuesta correcta: A

Los ejercicios planteados de la pregunta 5 a la 9 fueron extraídos de la competencia “Desafío del pensamiento computacional 2015” de Bebras (<http://www.bebas.org>) y la Olimpiada de computación de Búsqueda de talento 2015 (<http://www.olympiad.org.za>).

Resultados

450 estudiantes de los 700 que componen la totalidad de los que cursan la asignatura en el ciclo lectivo 2022 resolvieron el cuestionario. A continuación, los resultados destacados:

1. Composición de género

Mujeres: 26% y Varones: 74%.

Esta composición de género es propia de la carrera, en años anteriores los porcentajes fueron similares.

2. Experiencia previa de los estudiantes

Del total de respuestas, el 54% de los estudiantes señala que tiene experiencia previa en programación. En la Fig. 9 se puede apreciar que el 26% tiene experiencia adquirida en la escuela primaria o secundaria, un 3% en otra carrera universitaria o terciaria y un 25% indica que ha estudiado programación por su cuenta. Este último porcentaje da cuenta de la motivación de estos estudiantes y de la gran difusión que se le da actualmente a la programación, dada la fuerte demanda laboral de recursos humanos con esta formación.

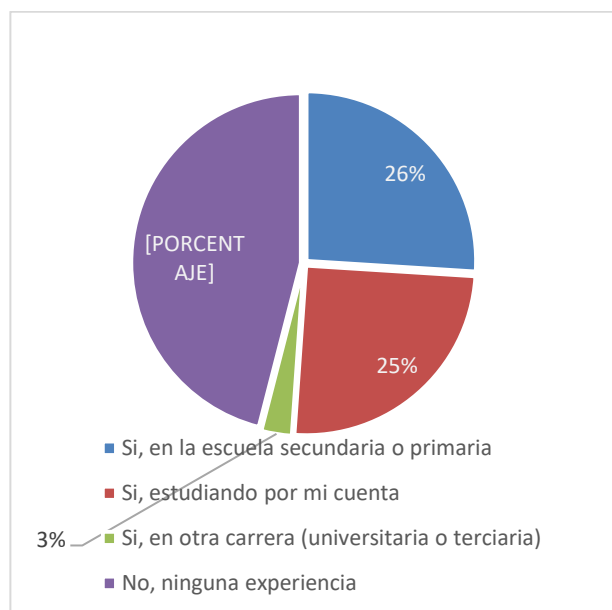


Figura 9. Experiencia en programación

A los que indicaron tener experiencia previa en programación se les pidió que señalen qué herramientas conocen. La mayoría no contestó la pregunta, pero se puede observar que los que adquirieron experiencia en la escuela primaria o secundaria señalaron que realizaron actividades con PilasBloques, Scratch y otras herramientas lúdicas. Estas cifras dan cuenta del avance en las escuelas de la propuesta formativa que la Fundación Sadosky lleva adelante a través de su iniciativa Program.AR. Los que estudiaron por su cuenta conocen PilaBloques y lenguajes de programación convencionales (C, Java, Python, otro).

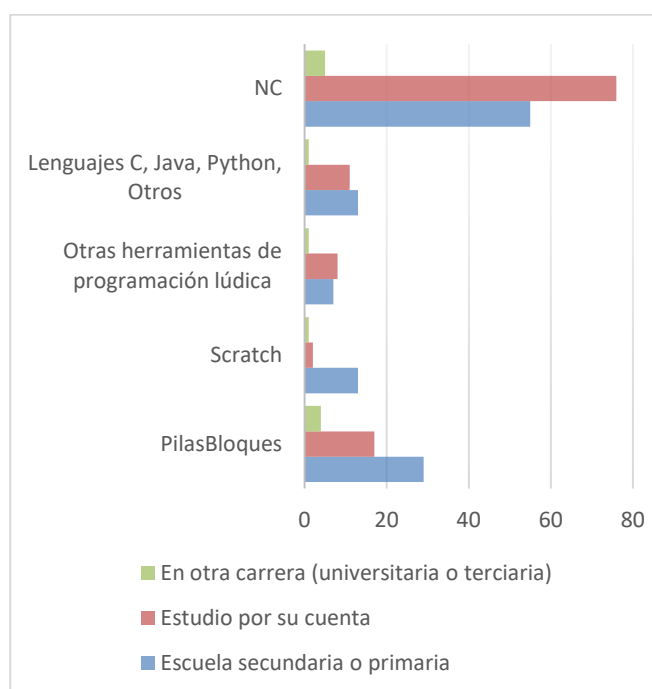


Figura 10. Herramientas de programación

3. Desafíos vinculados con las habilidades del pensamiento computacional

Con relación a los desafíos planteados se observan los resultados obtenidos.

1. Canguro

Este desafío requiere en gran medida de la habilidad de **abstracción**, dado que el estudiante debe analizar y determinar la información relevante para lograr resolver el problema planteado. En la tabla 1 se muestra la frecuencia de las distintas respuestas. Un 45% logró resolver este desafío correctamente.

Tabla 1: Respuestas de la actividad Canguros

Respuesta	Cantidad	%
A	10	2%
B	50	11%
C	19	4%
D	14	3%
E	7	2%
F	16	4%
G	10	2%
H	15	3%
I	203	45%
J	97	22%
N/C	9	2%
Total general	450	100%

2. Castores en movimiento

Para evaluar la habilidad del **diseño algorítmico** se propuso el desafío castores en movimiento, dado que es necesario organizar las instrucciones, es decir las indicaciones para que los castores superen los obstáculos. Se obtuvo una amplia variedad de combinaciones de números como respuesta. En la tabla 2 se detallan las respuestas que se presentaron con mayor frecuencia. Un 43% logró resolver con éxito la actividad.

Tabla 2: Respuestas de la actividad Castores

Respuesta	Cantidad	%
2-1-6-5-3-4-7	192	43%
3-4-5-6-7-1-2	36	8%
7-6-5-4-3-2-1	19	4%
2-1-6-5-7-4-3	14	3%
2-1-5-6-3-4-7	11	2%
N/C	6	1%
Otras respuestas incorrectas	172	38%
Total general	450	100%

3. Espías

Este desafío permite evaluar la habilidad de **generalización**, o **detección de patrones**, ya que, en base a la solución indicada para la situación inicial, es posible aplicar la misma en problemas con similares características. Un 34% logró resolver este desafío correctamente.

Tabla 3: Respuestas de la actividad Espías

Respuesta	Cantidad	%
4	154	34%
3	93	21%
5	62	14%
2	37	8%
6	33	7%
Más de 7	52	12%
N/C	19	4%
Total	450	100%

4. Salto de charcos

El propósito de este desafío es determinar la habilidad de **evaluación**, dado que se deben reconocer y determinar la realización de procesos, en términos de eficiencia y uso de recursos

En la tabla 4 se detallan las soluciones que se presentaron con mayor frecuencia. Un 54% de los estudiantes logró resolver este ejercicio.

Tabla 4: Respuestas de la actividad Salto de charcos

Respuesta	Cantidad	%
8-10-9-7-11	244	54%
8-9-10-7-11	17	4%
11-10-9-8-7	11	2%
9-10-8-7-11	10	2%
Combinaciones incorrectas elegidas por menos de 10 estudiantes	132	29%
N/C	36	8%
Total general	450	100%

5. Móviles

En el desafío de los móviles se pretende evaluar la habilidad de **descomposición**, la cual es necesaria para determinar cada componente y subcomponente del móvil. El 44% de los estudiantes resolvió correctamente este desafío.

Tabla 5: Respuestas de la actividad Móviles

Respuestas	Cantidad	%
A	199	44%
B	112	25%
C	86	19%
N/C	53	12%
Total	450	100%

Análisis global de las actividades

Haciendo un análisis completo, en la tabla 6 se muestra el resultado de cada actividad. El porcentaje de las respuestas correctas está calculado sobre los 450 participantes.

Tabla 6: Respuestas a los desafíos planteados

Desafío	Habilidades	Correctas	%
Canguro	Abstracción	203	45%
Castores	Diseño de Algoritmo	192	43%
Espías	Generalización o detección de patrones	154	34%
Salto de charcos	Evaluación	244	54%
Móviles	Descomposición	199	44%

En promedio, el 44% de los estudiantes pudo revolver las actividades propuestas. La actividad vinculada con la habilidad de Evaluación tuvo el mayor porcentaje de respuestas correctas (54%) y el menor valor el desafío vinculada con la habilidad de Generalización o Detección de patrones (34%).

Relación del desempeño con el sexo

En la Tabla 7 se puede ver la cantidad de respuestas correctas obtenidas entre distintos sexos, donde se observa una diferencia de media de 0.49 entre ambos grupos. Para definir si existe una diferencia entre estas categorías en la población, primero fue necesario determinar si la misma sigue una distribución normal y en función de ello elegir el test estadístico apropiado. Para ello se empleó el test de normalidad de Shapiro Wilk en la muestra reunida de respuestas correctas, y el mismo resultó significativo, rechazando la hipótesis de normalidad. Una vez determinada la distribución no normal de las respuestas correctas, se recurrió al test no paramétrico Mann-Whitney, obteniéndose un p-valor de 0.002 con un nivel de confianza de 95% demostrando una diferencia significativa entre los sexos.

Tabla 7: Estadísticas descriptivas respuestas correctas entre sexos.

Sexo	Cantidad	Media	Desvío estándar
Masculino	334	2.33	1.43
Femenino	116	1.84	1.43

Relación del desempeño con la experiencia previa en programación

Se contabilizó para cada estudiante la cantidad de respuestas correctas, en un rango de 0 a 5, y se relacionó con la experiencia declarada por el estudiante. En la Tabla 8 se muestran los resultados.

Considerando que los que tuvieron 4 o 5 respuestas correctas presentaron un buen desempeño, estos valores se analizaron en función de la experiencia. En la tabla 9 se muestra la relación.

Tabla 8: Experiencia y desempeño

Experiencia previa	Respuestas correctas					
	0	1	2	3	4	5
Si, en la escuela secundaria o primaria	15	21	27	29	14	11
Si, estudiando por mi cuenta	12	29	23	27	17	5
Si, en otra carrera	1	3	3	2	4	

(universitaria o terciaria)						
No, ninguna experiencia	35	42	46	43	29	12

Tabla 9: Experiencia y mejor desempeño

Experiencia previa	Rpta. 4y5	%
Si, en la escuela secundaria o primaria	25	27%
Si, estudiando por mi cuenta	22	24%
Si, en otra carrera (universitaria o terciaria)	4	4%
No, ninguna experiencia	41	45%
Total	92	100%

Un total de 92 estudiantes (20% del total) presentaron un buen desempeño contestando correctamente 4 o 5 de los desafíos.

El dato llamativo es que el 45% dijo no tener ninguna experiencia previa, un valor que supera significativamente a los que manifestaron tener experiencia en otras carreras de nivel superior. Por lo que la experiencia, si bien se muestra como condición relevante, dado que tiene el mayor porcentaje, no está muy alejado del valor obtenido de los que no tienen experiencia previa en programación.

Profundizando en este análisis se incluyeron técnicas de estadística descriptiva e inferencial. En la Tabla 10 se presenta la cantidad de respuestas correctas obtenidas en distintos niveles de experiencia de programación. Es observable que la media de respuestas correctas mínima fue registrada por los sujetos sin experiencia (1.99) y la máxima por el grupo con experiencia universitaria o terciaria (2.90). Dada la no normalidad de las respuestas correctas, se utilizó el test no paramétrico Kruskal-Wallis. El mismo registró un p-valor de 0.001 con un nivel de confianza de 95% demostrando una diferencia significativa entre los cuatro grupos. En cambio, al excluir el grupo con experiencia universitaria no se encontró esta misma diferencia (p-valor de 0.056 con 95% de confianza). Aquí se puede concluir que la experiencia universitaria o

terciaria previa fue un motivo para obtener mejores resultados en actividades.

Tabla 10: Estadísticas descriptivas respuestas correctas entre sujetos con distinta experiencia en programación.

Experiencia previa	Cant.	Media	Desv. Estándar
Si, en la escuela secundaria o primaria	98	2.07	1.44
Si, estudiando por mi cuenta	119	2.39	1.44
Si, en otra carrera (universitaria o terciaria)	41	2.90	1.50
No, ninguna experiencia	188	1.99	1.38

Conclusiones

Estos resultados dan cuenta de que es necesario fortalecer en los estudiantes las habilidades vinculadas con el pensamiento computacional. Surge también que la experiencia previa en programación no fue un factor determinante, excepto en el caso de la experiencia adquirida en otra carrera universitaria o terciaria.

Con lo cual, es importante adoptar metodologías de enseñanza que contribuyan a la adquisición de habilidades computacionales, mediante actividades diseñadas para tal fin.

A futuro se propone continuar con la evaluación de las habilidades del pensamiento computacional con otras actividades a fin de comparar los resultados.

Referencias

- [1] Katz, "TIC, digitalización y políticas públicas," *Entornos Digitales y Políticas Educativas*. IIPE-UNESCO, pp. 17-58. 2016.
- [2] A. Rabosto, M. Zukerfeld. "El sector argentino de software: desacoples entre empleo, salarios y educación". *Ciencia, Tecnología y Política*. Año 2-Nº2. Enero-Junio 2019. ISSN 2618-3188.
- [3] I. Sáenz Córdoba. "Déficit del capital humano en el área informática en Costa Rica". *Technology Inside*. Vol.4-Nº 4. Agosto-Diciembre 2019: pp. 21-28. ISSN: 2215-5392.
- [4] I. Miliszewska y G. Tan. "Befriending

- computer programming: a proposed approach to teaching introductory programming,” *Issues in Informing Science and Information Technology*, vol 4, 2007.
- [5] P. Compañ-Rosique, R. Satorre-Cuerda, F. Llorens-Largo, y R. Molina-Carmona, “Enseñando a programar: un camino directo para desarrollar el pensamiento computacional”. *RED. Revista de Educación a Distancia*, vol. 46. http://www.um.es/ead/red/46/faraon_et_al.pdf. 2015.
- [6] C. Kazimoglu, M. Kiernan, L. Bacon y L. Mackinnon, “A serious game for developing computational thinking and learning introductory computer programming,” *Procedia - Social and Behavioral Sciences* 47, pp. 1991–1999, 2012.
- [7] Dapozo, G. N.; Greiner, C. L.; Petris, R. H. (2017). Introduction To Programming Based On Playful Activities In The University. *Proceedings XLIII CLEI-46 JAIIO*, 1 (2017): 1 - 15.
- [8] Wing, J. M. (2011). *Research Notebook: Computational Thinking--What and Why?* Obtenido de *The Link*, The magazine of the Carnegie Mellon University School of Computer RED. *Revista de Educación a Distancia*. Núm. 63, Vol. 20. Artíc. 4, 30-04-2020 DOI: <http://dx.doi.org/10.6018/red.409991>
- [9] A. Csizmadia, P. Curzon, M. Dorling, S. Humphreys, T. Ng, C. Selby, J. Woollard. “Pensamiento Computacional Guía para profesores”. (2015). Guía traducida al español por Codemas.org Fuente original *COMPUTING AT SCHOOL*. Disponible en: <https://www.codemas.org/wp-content/uploads/2016/04/Pensamiento-computacional-Gu%C3%ADa-para-profesores.pdf>
- [10] Selby, C. C. (2015). Relationships: computational thinking, pedagogy of programming, and Bloom's Taxonomy. *WiPSCE '15 Proceedings of the Workshop in Primary and Secondary Computing Education* (págs. 80-87). London, United Kingdom: ACM New York, NY, USA. doi:10.1145/2818314.2818315
- [11] H. Pérez Narváez y R. Roig-Vila, “Entornos de programación no mediados simbólicamente para el desarrollo del pensamiento computacional. Una experiencia en la formación de profesores de Informática de la Universidad Central del Ecuador”, *RED-Revista de Educación a Distancia*, 46(9), 2015.
- [12] Rojas-López, A, García-Peñalvo, F.J. (2020). Evaluación de habilidades del pensamiento computacional para predecir el aprendizaje y retención de estudiantes en la asignatura de programación de computadoras en educación superior. *RED. Revista de Educación a Distancia*. Núm. 63, Vol. 20. Artíc. 4, 30-04-2020 DOI: <http://dx.doi.org/10.6018/red.409991>, Página 38 de 39
- [13] G. Simari, “Los fundamentos computacionales como parte de las ciencias básicas en las terminales de la disciplina Informática,” *VIII Congreso de Tecnología en Educación y Educación en Tecnología*. ISBN 978-987-1676-04-0, 2013.