

El impacto de un juego de mesa para introducir conceptos fundamentales de programación en nivel inicial

Marcos J. Gómez^{1,2}, Rocío Gonzalez², Diana López Alvas², Inés Roggi², and Cecilia Martinez¹

¹ Universidad Nacional de Córdoba, Argentina

marcos.gomez@unc.edu.ar, cecimart@gmail.com

² Fundación Sadosky, Argentina

{rgonzalez, dlopez, iroggi}@fundacionsadosky.org.ar

Abstract. Este trabajo analiza la introducción de conceptos fundamentales de computación en el nivel inicial a través de un juego de mesa desarrollado para tal fin denominado Pilas Bloques Desenchufado PBD. La propuesta permite a niños y niñas explorar conceptos como secuencia y repetición de manera desenchufada y tangible a través de una experiencia lúdica inmersiva. El estudio se realizó en una escuela de la provincia de Córdoba, con estudiantes y docentes de salas de 3, 4 y 5 años. Se implementaron diferentes modalidades del juego y se brindó formación específica a las docentes. Para analizar los aprendizajes de los estudiantes se aplicaron exámenes de opción múltiple y observaciones en el aula. Los resultados muestran que los estudiantes lograron incorporar conceptos como secuencia, argumento y, en algunos casos, repetición. Sin embargo, se evidenció que la apropiación conceptual fue mayor en las salas donde las docentes dedicaron más clases y profundizaron en la enseñanza de estos temas con la modalidad tangible, permitiendo a los estudiantes trabajar de manera sostenida con los desafíos del juego. Este estudio reafuerza la importancia de combinar enfoques desenchufados y tangibles en la enseñanza de la programación en edades tempranas y destaca el papel clave del tiempo de dedicación docente para promover los aprendizajes. Asimismo, este estudio contribuye a tres grandes preguntas de la política educativa del sector: ¿en qué nivel educativo introducir la enseñanza de conceptos de programación?, ¿cuánta carga horaria debería tener estos contenidos en la currícula? y ¿es posible enseñar computación sin computadoras?

Keywords: enseñanza de programación, programación desenchufada, programación tangible, Pilas Bloques, Nivel Inicial

1 Introducción

En los últimos quince años a nivel mundial se han hecho esfuerzos para incluir las Ciencias de la Computación (CC) de manera obligatoria en el currículum de

los niveles obligatorios del sistema. Esta iniciativa abre nuevas preguntas: *¿Para quiénes es deseable ofrecer estos saberes? ¿Cuál es la selección de contenidos que debería hacerse para cada nivel? ¿Qué dispositivos de formación docente son necesarios en un escenario donde las docentes no tienen conocimiento previo de la disciplina?* Este trabajo contribuye a estas preguntas. Puntualmente se centra en la enseñanza de la Computación en el Nivel Inicial.

En nuestro país, la introducción de estos contenidos es obligatoria desde 2018 a través de los Núcleos de Aprendizaje Prioritarios en Educación Digital, Programación y Robótica³. En general, la manera de introducir nociones fundamentales de las CC en el nivel inicial es a través de la enseñanza de la programación. Esta tradición está arraigada a las ideas de Papert, creador de LOGO (Papert, 1980) que han sostenido hasta hoy las ventajas de acercar a los niños a un lenguaje que pueda ser entendido por una computadora y que pueda expresar una idea propia. En el nivel inicial, el aprendizaje es promovido cuando el lenguaje puede interactuar con elementos tangibles -tales como un robot- en un contexto de experimentación y sin el uso de pantallas. De esta manera se enfatiza en los conceptos que transmite el lenguaje por sobre los medios de representación (Pugnali et al., 2017). Muchas investigaciones recientes muestran que la introducción de conceptos de programación y robótica en el nivel inicial se relacionan con el desarrollo de la colaboración, y un acercamiento similar entre los géneros a las ciencias exactas y tecnología (Bers et al., 2014; Kazakoff & Bers, 2014). Asimismo estudios de metanálisis muestran que la introducción de la programación en el nivel inicial promueve el desarrollo cognitivo (Montuori et al., 2024). No obstante ello, poco sabemos de la apropiación conceptual que diferentes grupos etarios de los 3 a los 5 años pueden hacer del gran repertorio de posibles conceptos de programación, qué relación tienen la frecuencia de la enseñanza de esos conceptos en su aprendizaje, y qué importancia tiene el diseño de materiales didácticos en la triada enseñanza, aprendizaje, conocimiento.

Para abordar estas áreas de vacancia en el conocimiento didáctico, este trabajo responde a las siguientes preguntas de investigación:

- ¿Qué conceptos de programación logran desarrollar infantes de 3 a 5 años cuando se los expone a actividades de programación y robótica desenchufada?
- ¿Qué relación se puede establecer entre la frecuencia de enseñanza y el aprendizaje?
- ¿Qué rol juegan los materiales didácticos en la promoción de los aprendizajes?

Las respuestas a estas preguntas son imprescindibles para informar a políticas educativas que debaten en relación a la selección de conceptos para cada grupo etario, el espacio que estos contenidos deberían ocupar en el currículum, y los enfoques de enseñanza promovidos por diferentes materiales didácticos.

³ <https://www.educ.ar/recursos/150123/nap-de-educacion-digital-programacion-y-robotica>

2 Trabajo previo

Desde el campo pedagógico se sostiene el derecho de los niños y niñas a desarrollar su espíritu de exploración a través de la tecnología digital y de construir modelos de pensamiento que permitan comprender las tecnologías detrás de las pantallas. Brailovsky et al. (2022) identifican que el discurso pedagógico se centra en el derecho a desarrollar la cognición y la expresión creadora con diversas tecnologías. La enseñanza de la programación a través de la robótica aborda ambas preocupaciones cuando el enfoque didáctico está centrado en la resolución de desafíos computacionales (en contraposición a seguir instrucciones sin oportunidades de proponer alternativas), y permite un trabajo sin pantallas que focaliza en la apropiación de conceptos fundamentales de la computación.

Una línea de investigaciones en este campo, indaga sobre los efectos que tiene el aprendizaje de la programación en el nivel preescolar en habilidades cognitivas que incluyen creatividad, razonamiento y matemática. Este tipo de estudios investiga sobre si las habilidades que se desarrollan aprendiendo programación pueden ser transferidas a otras áreas de la cognición. En un metanálisis de 105 estudios, Scherer et al. (2019) encuentran que las habilidades de programación pueden ser transferidas a algunas situaciones que requieren pensamiento creativo, matemática, y pensamiento espacial. Sin embargo, no se observan transferencias en habilidades relacionadas con la lecto escritura.

Otra línea de investigaciones aborda los aprendizajes de la disciplina Ciencias de la Computación a partir de actividades de programación (Alonso-García et al., 2024; Wei et al., 2024). Wei et al. (2024) realizan una revisión sistemática para encontrar 32 investigaciones que usan diferentes modalidades de enseñanza: gráfica, tangible y desenchufada. La programación gráfica pone el énfasis en la dimensión visual de la programación. Las interfaces desarrolladas para los niños y niñas son intuitivas donde diferentes bloques de programación pueden ser manipulados y combinados para crear un código. Kodu y Scratch Jr. son ejemplos de este enfoque.

La modalidad desenchufada (Bell & Vahrenhold, 2018) y la modalidad tangible (Bers & Horn, 2010) comparten la característica de no requerir pantallas ni dispositivos digitales, lo que las hace accesibles y aplicables en diversos contextos educativos con recursos limitados. Ambas modalidades enfatizan el aprendizaje activo y situado, a través de la manipulación, la exploración y el trabajo colaborativo.

Si bien investigaciones como la de Wei et al. (2024) muestran que la modalidad gráfica puede ofrecer mejores resultados generales en test de pensamiento computacional, otros estudios plantean resultados diferentes. Por ejemplo, Pugnali et al. (2017) encontró que la programación tangible fue más beneficiosa que la gráfica en niños de 4 a 7 años, al favorecer la interacción entre pares y la discusión sobre contenidos, elementos clave para el aprendizaje. Además, es importante considerar que la programación gráfica puede presentar barreras para estudiantes que aún no han desarrollado habilidades motoras finas suficientes para el uso del mouse.

En síntesis, cada modalidad ofrecería diferentes oportunidades de apropiación de los conceptos. Recuperando estas investigaciones, algunas experiencias combinan las modalidades de programación. Canbeldek and Isikoglu (2023) diseñaron para estudiantes de 4 a 5 años una experiencia de programación que incluía un momento desenchufado, seguido de actividades tangibles con Bee Bot y terminaba con programación gráfica. Se llevó a cabo con esta experiencia un estudio experimental con 80 estudiantes de jardines de infantes públicos elegidos al azar y se tomaron pre y post test que evaluaban lenguaje, creatividad, matemática, pensamiento lógico y habilidades propias del pensamiento computacional. Como resultado de 9 semanas de intervención en las aulas, el grupo experimental mostró mejores resultados en todos los test de manera significativa. Sin embargo, la mayor diferencia se dio en los test de razonamiento matemático y pensamiento computacional. Este crecimiento se interpreta como indicador de que la experiencia combinada logra mejorar esas habilidades. Montuori et al. (2023) también combina actividades desenchufadas, tangibles y gráficas obteniendo resultados similares.

En esta misma línea, este estudio propone combinar modalidades de enseñanza y analizar los resultados de aprendizaje con una muestra intencional que permite comprender con mayor detalle las variables de contexto. Para ello desarrollamos una experiencia denominada Pilas Bloques Desenchufada. La relevancia de analizar esta experiencia radica también en la posibilidad de transferir materiales y métodos de enseñanza a contexto donde no es posible -o deseable- adquirir computadoras para los más pequeños. Enseñar conceptos de computación sin computadoras permite pensar políticas de mayor cobertura que aborde el derecho de las niñas y los niños a desarrollar esquemas de pensamiento para comprender el mundo digital.

3 Pilas Bloques Desenchufado

La iniciativa Program.AR de la Fundación Sadosky, ha desarrollado un entorno de enseñanza de programación en bloques. Pilas Bloques (Sanzo et al., 2017) posee desafíos con diversos niveles de dificultad para que estudiantes de nivel primario y secundario, puedan conocer el mundo de la programación dando los primeros pasos en la comprensión de diferentes conceptos fundamentales.

Pilas Bloques cuenta con un laberinto, un desafío y un conjunto de bloques, para crear programas que resuelvan el desafío planteado. Sus personajes fueron diseñados para mostrar la flora y fauna de Argentina y representar a personas de diferentes géneros. En la línea teórica que promueve la combinación de modalidades, durante 2023 Program.ar desarrolló una propuesta desenchufada de Pilas Bloques que permite un acercamiento a los saberes vinculados a la resolución de problemas y la programación especialmente pensado para nivel inicial con variantes que permiten su utilización en el primer ciclo de nivel primario.

Se trata de un juego reglado tangible al estilo de un “**juego de mesa**” que favorece una experiencia corporal inmersiva. La propuesta promueve un aprendizaje situado y significativo, donde las niñas y los niños puedan tener un rol



Fig. 1. Los tableros con los personajes, objetos y los programas para resolver desafíos en las 3 modalidades de PBD.

protagónico. El objetivo principal del juego es que resuelvan desafíos que les permitan incorporar conceptos fundamentales de programación. En este contexto se propone que las niñas y los niños, de acuerdo a sus características y etapa escolar, puedan incorporar conceptos como: lenguaje de programación, noción de programa, comandos primitivos, argumentos, secuencia, repetición e identificación de patrones. La selección de estos contenidos se basó en documentos curriculares específicos (Committee et al., 2016; Seehorn & Clayborn, 2017).

El juego está compuesto por fichas de personajes (Lita o Duba), obstáculos (no se pueden atravesar ni saltar), objetivos, bloques de comandos (acciones y repetición) y cartas de 3 niveles de dificultad que representan distintos desafíos. Estos componentes conviven en un tablero donde se encuentran los espacios para crear la secuencia de bloques para cada personaje y una cuadrícula en el que se ubican los objetos a agarrar, los personajes y los obstáculos, según la carta elegida. En la Fig. 1 podemos observar el juego y sus componentes en la modalidad pizarra.

4 Diseño de estudio

4.1 El contexto de la implementación

El estudio se realizó en horario escolar en una institución de gestión privada en la provincia de Córdoba. La mayoría de las y los niños provienen de familias de clase media con padres profesionales y no se encuentran bajo la línea de pobreza. La currícula de la escuela se basa en el aprendizaje basado en problemas y exploración a partir de proyectos.

El estudio se llevó a cabo en el nivel inicial de la institución, que cuenta con un total de 11 salas: ocho en la mañana (dos salas de 3 años, tres salas de 4 años y tres salas de 5 años) y tres en la tarde (una por edad). Cada sala tiene una docente responsable a cargo del grupo de estudiantes, de la planificación

e implementación de contenidos y proyectos. Además, hay docentes auxiliares que rotan entre salas para acompañar en determinadas actividades o momentos específicos.

Para poder medir la incorporación conceptual propusimos al equipo directivo contar con algunas salas de control. Sin embargo, esto no fue posible ya que todas las docentes querían implementar PBD con sus estudiantes, y veían un inconveniente en que algunos estudiantes se quedaran sin la experiencia. En base a este pedido, organizamos para que todas las docentes pudiesen contar con la oportunidad de usar PBD en el aula. Con el pasar de la experiencia pudimos diferenciar la puesta en acto de cada una de las docentes con el material y el tiempo dedicado a su enseñanza y así compararlos a los diferentes grupos recuperando algunas herramientas de un estudio cuasiexperimental. En estos estudios se comparan grupos en donde no es posible asignar control de variables, sino que se trabaja con los grupos conformados por la institución. En la Sección 5.2 describiremos los grupos de estudiantes, las experiencias, sus diferencias y cómo surgieron diferentes grupos. Las responsables de implementar los contenidos y desafíos en el aula, fueron las docentes de la institución. En la sección siguiente describimos cómo fue el proceso de formación.

4.2 Nuestra Propuesta de Formación Docente

Como parte del proyecto, diseñamos una propuesta que constó de tres elementos complementarios:

1. Las docentes participaron en un taller de un solo encuentro (en esta experiencia, presencial), con una duración de entre 90 y 120 minutos.
2. Se les brindó material didáctico desarrollado específicamente para docentes, que incluyeron clases, actividades y conceptos, siguiendo la estrategia didáctica propuesta en el taller.
3. Una vez finalizado el encuentro de formación y en el proceso previo a la implementación en el aula, las docentes mantuvieron contacto directo con un especialista en didáctica de la programación (el capacitador responsable). Consultaron sobre el material didáctico, conceptos específicos o aspectos particulares de PBD. Para ello, se comunicaron directamente a través de WhatsApp.

El taller: El equipo directivo propuso aprovechar una de las reuniones de equipo completo para hacer una jornada de 90 minutos donde introducir el juego y los conceptos de programación a abordar. La persona encargada de la capacitación docente es un especialista en didáctica de las CC, que hace más de 10 años trabaja en formación docente y desarrollo de contenidos de CC, y en particular de programación. En el taller se presentaron 3 posibles modalidades del juego para poder abordar los contenidos y desafíos.

1. **Juego de piso:** diseñado con una lógica inmersiva de programación desenclufada, en este formato las y los estudiantes asumían el rol de los personajes

del juego. Para ello, utilizaban algún tipo de disfraz y podían desplazarse físicamente sobre un tablero de gran tamaño dispuesto en el suelo, interactuando directamente con la dinámica del juego. Esta versión fue la primera en implementarse, ya que facilitaba la comprensión inicial de las mecánicas sin la necesidad de intermediarios físicos como tableros o fichas.

2. **Juego de pizarra:** en esta versión, con énfasis en la modalidad tangible, el tablero del juego se presentaba en un formato visible para toda la clase, como si fuera un pizarrón. Esto permitía que tanto docentes como estudiantes observaran el desarrollo del juego de manera conjunta y movieran las fichas en un espacio compartido, favoreciendo la reflexión colectiva sobre las estrategias de resolución. Se implementó como un segundo paso, agregando un nivel de abstracción mayor en la representación del juego.
3. **Juego de mesa:** este formato tradicional también con énfasis en lo tangible, consistía en tableros y fichas diseñadas para ser utilizadas en mesas, permitiendo que los estudiantes jugaran en pequeños grupos. Se implementó en la última etapa, cuando los y las estudiantes ya estaban familiarizados con la dinámica del juego y podían abordar desafíos de mayor complejidad de manera autónoma.

El taller se centró en la modalidad de juego de mesa. Organizamos a las docentes por cohorte, entregando a cada grupo un tablero A4 junto con personajes, obstáculos y bloques necesarios para resolver los desafíos.

La formación siguió un enfoque de exploración. En los primeros minutos, se introdujeron los elementos básicos del juego (tablero, personajes, objetos y bloques de movimiento junto con su semántica). Los conceptos de programación —lenguaje, programa, secuencia, argumento y repetición— se presentaron gradualmente, a partir de desafíos que requerían su uso. El capacitador guiaba con preguntas, evitando dar respuestas directas, promoviendo así una experiencia de aprendizaje por indagación.

Por ejemplo, en el primer desafío se proyectó un tablero con obstáculos, Lita la mulita y una lechuga. Las docentes debían replicar el tablero y resolverlo con los bloques disponibles. Intencionalmente no se presentó el bloque **agarrar** en la introducción, para que surgiera como una necesidad durante la resolución.

Se eligió esta estrategia didáctica para intentar que las docentes la pongan en juego en sus aulas. En base a la literatura existente, sabemos que la mayoría de los docentes tienden a enseñar con las mismas estrategias de enseñanza con las que aprendieron, y que el aprendizaje de teorías pedagógicas no puede competir con lo que los docentes experimentaron como estudiantes (Carter, 1990). También sabíamos que la mayoría de los docentes que participaban del curso no tenían experiencia previa en el aprendizaje de programación, por lo tanto, desarrollamos una experiencia de aprendizaje de programación significativa para tener impacto (Windschitl & Sahl, 2002).

5 Recolección de datos y análisis de participantes

En la Sección 5.1 presentamos las diferentes herramientas que utilizamos para recolectar datos durante la experiencia. Luego, en la sección 5.2 describimos a las docentes y realizamos un primer análisis cuantitativo de los grupos de estudiantes que participaron de la experiencia.

5.1 Herramientas de recolección

Para poder medir los aprendizajes y la incorporación conceptual de los conceptos de programación por parte de las niñas y los niños, diseñamos un examen de múltiple opción que debían resolver una vez finalizada la experiencia. Siguiendo la experiencia descrita por Martinez et al. (2015) clasificamos nueve preguntas del examen en simples o compuestas. Una pregunta se considera *simple* cuando es necesario utilizar un solo concepto para resolver la actividad, mientras que se clasifica como *compuesta*, si es necesario combinar conceptos para resolver el desafío planteado. Los conceptos que fueron trabajados durante la experiencia fueron: **secuencia, argumento y repetición**. Los conceptos de Argumento, Secuencia y Repetición se evalúan de manera simple (SArg1, SSec1, SSec2, SSec3, SSec1 y SRep1). En uno de los desafíos se combinan los conceptos de secuencia y argumento (CSArg1) mientras que en dos desafíos se combinan los conceptos de repetición, secuencia y argumento (CRep1 y CRep2).

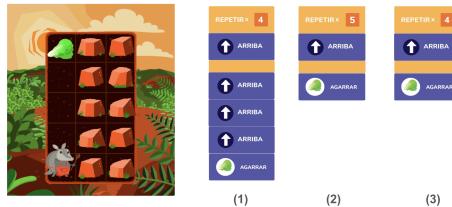


Fig. 2. Uno de los ejercicios que formaron parte de la evaluación de múltiple opción.

En la Fig. 2 se observa uno de los ejercicios. Los estudiantes debían elegir cuál de los tres programas le permitía a Lita llegar hasta la lechuga y agarrarla. En este caso la respuesta correcta es la (3). El examen fue diseñado para incluir respuestas que contengan errores conceptuales comunes de programación. Por ejemplo, en la Fig. 2 las opciones incorrectas (1) y (2) ilustran diferentes errores conceptuales comunes sobre el comportamiento de los ciclos que reportan Gomez et al. (2019) and Weintrop and Wilensky (2015).

Todas las salas programaron desafíos en los cuales los estudiantes tenían que utilizar los conceptos de secuencia y argumento para resolver los tableros propuestos. Con respecto al concepto de repetición fue solamente introducido en una sala de 4 (S4_3) y en todas las salas de 5. En la Sección 5.2 vamos a describir y caracterizar la experiencia de cada una de las salas.

Además de los exámenes de múltiple opción se realizaron observaciones durante las clases en donde fue implementado PBD. Las observaciones fueron realizadas por un investigador especializado en la enseñanza de ciencias de la computación. El investigador realizó anotaciones y envió audios para cada una de las clases observadas. A partir de las mismas pudimos documentar comentarios o situaciones que sucedieron durante la experiencia y que se reportan en las secciones siguientes.

5.2 Análisis de la experiencia

En total participaron 226 estudiantes de la experiencia. Pero, no todos los estudiantes tomaron el examen. La Tabla 1 presenta información detallada sobre la implementación de la experiencia en cada sala. La columna **Edad** indica el nivel (sala de 3, 4 o 5 años), mientras que **ID** corresponde a un identificador asignado a cada sala. La columna **#Evaluados** señala la cantidad de estudiantes que completaron el examen de opción múltiple. **#SA** refiere al número de clases dedicadas a trabajar los conceptos de *secuencia* y *argumento*, y **#Rep** a la cantidad de clases en que se abordó específicamente el concepto de *repetición*. La columna **#Total** indica el total de clases en las que se trabajó con la propuesta. Finalmente, **Modalidades** detalla los formatos de implementación utilizados en cada sala. Las evaluaciones fueron realizadas entre el 20 de noviembre y los primeros días de diciembre. Para cada una de las salas, la evaluación se realizó dos semanas después de que la experiencia haya finalizado en la sala. Por eso las diferencias entre las fechas de evaluación. Como las clases a finales de diciembre están finalizando, muchos estudiantes no asisten todos los días a clases, lo cual redujo la cantidad de estudiantes evaluados por sala.

De la experiencia participaron 11 docentes; 5 de ellas con alguna experiencia previa de enseñanza de robótica.

Edad	ID	#Evaluados	#SA	#Rep	#Total	Modalidades
3	S3.1	11	8	0	8	Piso/Pizarra
3	S3.2	14	10	0	10	Piso/Pizarra/Mesa
Total sala de 3		25				
4	S4.1	11	6	0	6	Piso/Pizarra/Mesa
4	S4.2	15	7	0	7	Piso/Pizarra/Mesa
4	S4.3	17	9	3	12	Piso/Pizarra/Mesa
Total sala de 4		39				
5	S5.1	13	5	1	6	Piso/Pizarra/Mesa
5	S5.2	13	5	2	7	Piso/Pizarra/Mesa
5	S5.3	13	6	2	8	Piso/Pizarra/Mesa
5	S5.4	15	8	3	11	Piso/Pizarra/Mesa
Total sala de 5		54				

Table 1. Cantidad de estudiantes que resolvieron el examen múltiple opción por sala. Cantidad de clases dedicada por sala al contenido y las modalidades utilizadas.

Las observaciones y registros del seguimiento evidenciaron que, si bien las docentes siguieron las secuencias didácticas propuestas, muchas realizaron adapta-

ciones según las necesidades de sus grupos. Estas modificaciones incluyeron desafíos intermedios, incorporación de nuevos elementos al juego, y sugerencias que incluso se integraron en versiones posteriores del recurso.

También se identificaron diferencias en la implementación según sala y docente. En general, todas mostraron interés y compromiso, pese a estar simultáneamente abocadas a otros proyectos institucionales.

En sala de 3, el equipo mostró gran interés, destacándose **S3_2**, que fue la que más tiempo dedicó a la experiencia. En sala de 4, la participación fue más heterogénea: **S4_3** se destacó por su entusiasmo e innovación, adaptando y ampliando contenidos, y promoviendo el uso del formato de mesa. **S4_2** mostró continuidad pero priorizando otros proyectos, mientras que **S4_1** delegó parte del trabajo en una docente auxiliar, lo que redujo la profundidad de la experiencia.

En sala de 5, todas las docentes abordaron los conceptos previstos, aunque con diferencias en profundidad y dedicación. **S5_4** fue referente por su compromiso y por integrar el recurso con propuestas digitales. **S5_3**, en proceso de jubilación, implementó la experiencia pero con menos clases, señalando dificultades para trabajar el concepto de repetición con poco tiempo. **S5_2** dividió tareas con una auxiliar, y **S5_1** abordó el proyecto con menos frecuencia, impulsada principalmente por el interés de sus estudiantes.

6 Resultados

En esta sección presentamos los resultados obtenidos a partir de los exámenes de opción múltiple y las observaciones en el aula. El análisis se realiza por nivel (salas de 3, 4 y 5 años) y también entre docentes del mismo nivel, considerando la cantidad de clases dedicadas y los conceptos abordados. Dado que no todas las salas trabajaron el concepto de repetición, los ejercicios correspondientes fueron resueltos solo por algunos grupos. Además, como los estudiantes no tenían experiencia previa con este tipo de evaluaciones, las docentes acompañaron su implementación para facilitar la comprensión. En la Sección 6.1 se presentan los resultados por nivel, mientras que en la Sección 6.2 se comparan los desempeños entre salas del mismo nivel.

6.1 Análisis por nivel

Como describimos en la sección anterior no todas las docentes pudieron implementar el concepto de repetición. Teniendo en cuenta esa restricción compararemos los resultados de los ejercicios de argumento y secuencia, que son los que todos los estudiantes pudieron resolver.

Como se observa en la Tabla 1, 25 estudiantes de sala de 3, 39 estudiantes de sala de 4 y 54 estudiantes de sala de 5 completaron los ejercicios del examen múltiple opción enumerados anteriormente. En la Fig. 3 se puede observar el porcentaje de estudiantes por sala evaluados que resolvió de forma correcta cada ejercicio.

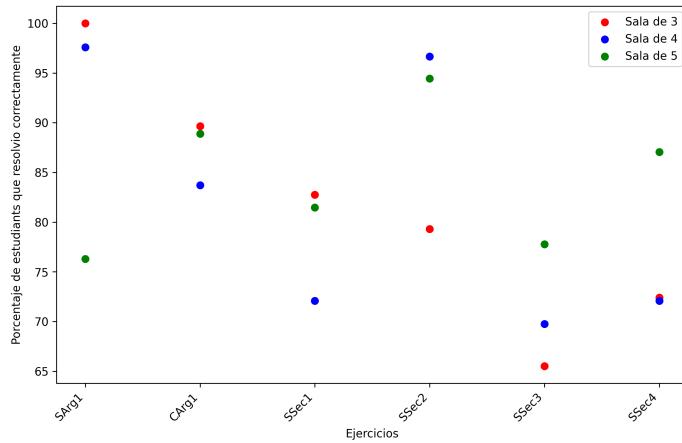


Fig. 3. Porcentaje de estudiantes por nivel evaluados que resolvió de forma correcta cada ejercicio.

Podemos observar que más del 65% de los estudiantes de las 3 salas pudo resolver correctamente la mayoría de los desafíos. En particular hay un rendimiento similar de los estudiantes de 3, 4 y 5 donde está presente el concepto de Argumento (SArg1 y CArg1). Con respecto al concepto de secuencia podemos ver un muy buen desempeño de los estudiantes de las tres salas. Los estudiantes de Sala de 5 resuelven de mejor manera los ejercicios de SSec3 y SSec4, donde los desafíos a resolver son más complejos y largos, los cuál es esperable teniendo en cuenta las diferentes etapas madurativas en las cuales los estudiantes se encuentran. Analizando el gráfico observamos que los estudiantes en las tres salas pueden abordar los conceptos de argumento y secuencia.

6.2 Análisis comparativo entre docentes de sala

Los resultados acumulados por sala (es decir agrupando las cuatro salas de 5, las tres salas de 4 y las tres salas de 3) no son la única variable comparativa a tener en cuenta. Como describimos no todas las docentes implementaron todos los conceptos y dedicaron la misma cantidad de clases y con el mismo nivel de profundidad a la experiencia.

En la Tabla 2 se presenta el porcentaje de estudiantes por sala que resolvieron correctamente cada ejercicio del examen. Los datos permiten comparar desempeños entre docentes de un mismo nivel. Se observa variabilidad dentro de cada grupo etario, lo que sugiere que factores como la cantidad de clases y el enfoque de enseñanza influyen en los resultados.

Comenzamos con el concepto de **argumento**. En general se observa que en las salas en donde menos cantidad de veces se implementó la experiencia, el rendimiento de los estudiantes fue menor. Lo mismo ocurre con los ejercicios vinculados a **secuencia**, donde se destaca un mejor rendimiento en aquellas

Edad	Sala	#SA	#Rep	#Total	SArg1	SSec1	SSec2	CSec1	SSec3	SSec4	CRep1	SRep1	CRep2
3	S3.1	8	0	8	100	73	64	82	64	55	—	—	—
3	S3.2	10	0	10	100	86	86	93	79	93	—	—	—
4	S4.1	6	0	6	91	36	100	73	55	64	—	—	—
4	S4.2	7	0	7	100	80	93	80	60	60	—	—	—
4	S4.3	9	3	12	100	88	100	94	88	88	82	53	53
5	S5.1	5	1	6	85	69	92	77	62	69	8	8	0
5	S5.2	5	2	7	100	92	92	92	62	85	54	0	31
5	S5.3	6	2	8	100	77	92	92	85	100	69	15	38
5	S5.4	8	3	11	100	87	100	93	100	93	93	67	87

Table 2. Total de clases implementadas y desempeño de los estudiantes de cada sala por ejercicio.

salas que dedicaron más clases y emplearon modalidades más cercanas al formato del examen. Por ejemplo, en sala de 3, **S3_2** dedicó más clases y utilizó la modalidad de mesa, obteniendo mejores resultados que **S3_1**. En sala de 4, **S4_3** mostró mejor desempeño en desafíos complejos (SSec3 y SSec4), mientras que **S4_1**, que delegó la implementación a una auxiliar y priorizó otros proyectos, presentó resultados más bajos. En sala de 5, las diferencias también se acentúan en ejercicios de mayor dificultad. Las salas **S5_4** y **S5_2** se destacan por un mejor desempeño, posiblemente asociado tanto a la cantidad de clases como a la forma en que se abordaron los contenidos. Estos datos sugieren que unas seis clases podrían ser suficientes para introducir la secuencia, pero no para resolver desafíos más complejos. A pesar de estas diferencias, en todas las salas se observa una apropiación generalizada de los conceptos de **argumento** y **secuencia**.

Con respecto a repetición, observamos el mismo patrón. Las salas que menos clases dedicaron, menor desempeño tuvieron sus estudiantes. Vamos a sumar a la sala de 4, **S4_3** a la comparación con las salas de 5. La sala **S4_3** y sala **S5_4** tuvieron mucho mejor desempeño con respecto a las otras salas de 5. Esto se vincula no solo a que abordaron más clases al concepto repetición, sino que se abordaron más clases para los conceptos de secuencia y argumento, los cuáles impulsan una mejor comprensión del nuevo concepto.

7 Conclusiones

La implementación PBD en el nivel inicial permitió observar una fuerte apropiación de los conceptos de programación propuestos por parte de las y los estudiantes, así como un alto grado de compromiso y entusiasmo tanto por parte de las docentes y de las y los estudiantes que mostraron, desde el inicio de la experiencia, gran interés en participar, solicitando de manera activa *jugar a PBD*. Un caso particularmente significativo fue el de una estudiante de sala de 3 que eligió festejar su cumpleaños con la temática del juego, decorando incluso su torta con los personajes.

En cuanto a las docentes, se evidenció una apropiación pedagógica significativa del recurso. Muchas de ellas no solo adaptaron la secuencia propuesta a las necesidades de sus grupos, sino que además crearon desafíos intermedios, incorporaron ayudas visuales como fichas en forma de rompecabezas, subrayados

y códigos de colores, y realizaron sugerencias que se incorporaron en futuras versiones del juego. Esto muestra el potencial de la herramienta no solo como recurso didáctico, sino también como catalizador de la creatividad docente.

A nivel conceptual, se evidenció que los estudiantes de salas de 3, 4 y 5 años lograron apropiarse de los conceptos de argumento y secuencia, reforzando hallazgos de estudios previos (Bers et al., 2014; Martinez et al., 2015). Además, los estudiantes de sala de 5 pudieron comenzar a trabajar con el concepto de repetición. Sin embargo, se observó que para una apropiación significativa de este último concepto es necesaria una cantidad previa de clases dedicada al trabajo con secuencia y argumento. Específicamente, se detectó que con al menos 8 clases (una por semana) los estudiantes pueden incorporar los conceptos de argumento y secuencia, y que con al menos 11 clases pueden comenzar a apropiarse del concepto de repetición. Esta información es valiosa para orientar la planificación didáctica y establecer expectativas realistas respecto a qué conceptos pueden abordarse y en qué plazos.

Una de las principales limitaciones metodológicas del estudio fue la imposibilidad de implementar un pre-test para evaluar el conocimiento previo de los estudiantes sobre los conceptos de programación. Esto impide establecer con precisión el progreso conceptual individual de los niños y niñas a lo largo de la experiencia. A futuro, contar con una medición inicial permitiría evaluar con mayor rigurosidad el impacto de la propuesta.

References

- Alonso-García, S., Fuentes, A.-V. R., Navas-Parejo, M. R., & Victoria-Maldonado, J.-J. (2024). Enhancing computational thinking in early childhood education with educational robotics: A meta-analysis. *Heliyon*, 10(13).
- Bell, T., & Vahrenhold, J. (2018). Cs unplugged—how is it used, and does it work? In H.-J. Böckenhauer, D. Komm, & W. Unger (Eds.), *Adventures between lower bounds and higher altitudes: Essays dedicated to juraj hromkovič on the occasion of his 60th birthday* (pp. 497–521).
- Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & education*, 72, 145–157.
- Bers, M. U., & Horn, M. S. (2010). Tangible programming in early childhood. *High-tech tots: Childhood in a digital world*, 49, 49–70.
- Brailovsky, D., De Angelis, S., & Melo, G. S. (2022). Ni malas ni buenas: Esenarios del encuentro entre infancias y pantallas. *Voces de la educación*, 25–51.
- Canbeldek, M., & Isikoglu, N. (2023). Exploring the effects of “productive children: Coding and robotics education program” in early childhood education. *Education and Information Technologies*, 28(3), 3359–3379.
- Carter, K. (1990). Teachers’ knowledge and learning to teach. *Handbook of research on teacher education*, 291–310.

- Committee, K.-1. C. S. F. S., et al. (2016). *K-12 computer science framework*. ACM.
- Gomez, M. J., Moresi, M., & Benotti, L. (2019). Text-based programming in elementary school: A comparative study of programming abilities in children with and without block-based experience. *Proceedings of ITICSE 2019 ACM*, 402–408.
- Kazakoff, E. R., & Bers, M. U. (2014). Put your robot in, put your robot out: Sequencing through programming robots in early childhood. *Journal of Educational Computing Research*, 50(4), 553–573.
- Martinez, C., Gomez, M. J., & Benotti, L. (2015). A comparison of preschool and elementary school children learning computer science concepts through a multilanguage robot programming platform. *Proceedings of ITICSE 2015*, 159–164.
- Montuori, C., Gambarota, F., Altoé, G., & Arfé, B. (2024). The cognitive effects of computational thinking: A systematic review and meta-analytic study. *Computers & Education*, 210, 104961.
- Montuori, C., Pozzan, G., Padova, C., Ronconi, L., Vardanega, T., & Arfé, B. (2023). Combined unplugged and educational robotics training to promote computational thinking and cognitive abilities in preschoolers. *Education Sciences*, 13(9), 858.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.
- Pugnali, A., Sullivan, A., & Bers, M. U. (2017). The impact of user interface on young children's computational thinking. *Journal of Information Technology Education. Innovations in Practice*, 16, 171.
- Sanzo, A., Schapachnik, F., Factorovich, P., & O'Connor, F. (2017). Pilas bloques: A scenario-based children learning platform. *2017 LACLO*, 1–6.
- Scherer, R., Siddiq, F., & Sánchez Viveros, B. (2019). The cognitive benefits of learning computer programming: A meta-analysis of transfer effects. *Journal of Educational Psychology*, 111(5), 764.
- Seehorn, D., & Clayborn, L. (2017). Csta k-12 cs standards for all (abstract only). *Proceedings of the 2017 ACM SIGCSE*, 730.
- Wei, Y., Wang, L., Tang, Y., Su, J., Lei, Y., & Peng, W. (2024). Influence of programming education modalities on the computational thinking in young children: A comprehensive review and meta-analysis. *Journal of Computer Assisted Learning*, 40(5), 2385–2397.
- Weintrop, D., & Wilensky, U. (2015). Using commutative assessments to compare conceptual understanding in blocks-based and text-based programs. *Proceedings of the Eleventh Annual International Conference on International Computing Education Research*, 101–110.
- Windschitl, M., & Sahl, K. (2002). Tracing teachers' use of technology in a laptop computer school: The interplay of teacher beliefs, social dynamics, and institutional culture. *American educational research journal*, 39(1), 165–205.