Pensamiento computacional y ciencias de la computación en la primera infancia



En los últimos años, investigadores, responsables políticos y educadores han reconocido cada vez más la importancia de fomentar el desarrollo de las habilidades, prácticas y procesos de pensamiento computacional en los niños pequeños. El pensamiento computacional es una forma de pensar universal que resulta relevante para el aprendizaje de los niños, con numerosas aplicaciones en el mundo real y conexiones significativas con otras áreas de contenido, como las matemáticas y las ciencias. Las habilidades de pensamiento computacional también sirven de base para el aprendizaje de las ciencias de la computación. Por ello, los sistemas educativos nacionales y estatales —incluido el de California— han publicado estándares y marcos centrados en la integración del pensamiento computacional y la CI en el aprendizaje temprano (Junta de Educación del Estado de California, 2018; Marco de Ciencias de la Computación para K–12, 2016).

En respuesta a la creciente necesidad de apoyar a los educadores y a las familias en la implementación del pensamiento computacional y las ciencias de la computación en la educación de la infancia temprana, WestEd ha elaborado este informe, en colaboración con sus socios, como parte de la iniciativa «Count Play Explore» (CPE), un esfuerzo estatal en el que participan múltiples organismos y que está liderado por la Oficina del Superintendente de Escuelas del Condado de Fresno, Departamento de Cuidado y Educación Temprana. CPE tiene como objetivo sensibilizar sobre la importancia de las matemáticas, las ciencias y las ciencias de la computación en la primera infancia, así como desarrollar la capacidad de los educadores y las familias para apoyar el aprendizaje y el desarrollo tempranos de los niños.

Este informe aborda el desarrollo del pensamiento computacional y las ciencias de la computación en los niños desde el nacimiento hasta el tercer grado de primaria. Comienza definiendo el pensamiento computacional y las ciencias de la computación, y ofrece una visión general de las habilidades clave de pensamiento computacional adecuadas para el desarrollo de los niños pequeños. A continuación, resume estrategias basadas en la investigación sobre cómo los educadores pueden incorporar el pensamiento computacional y las ciencias de la computación en sus entornos de aprendizaje temprano y participar en experiencias de formación profesional de alta calidad en estas áreas. Por último, el informe concluye con conclusiones clave y recomendaciones para los facilitadores, capacitadores y entrenadores de formación profesional a la hora de poner en práctica experiencias de aprendizaje de pensamiento computacional y ciencias de la computación con niños, educadores y familias en sus comunidades locales.

Definición del pensamiento computacional y las ciencias de la computación

El pensamiento computacional es un conjunto de habilidades, prácticas y procesos de pensamiento crítico y resolución de problemas que cualquier persona, incluidos los expertos en computación, puede aprender y utilizar en diversos contextos (Bers, 2018; Grover y Pea, 2013; Wing, 2006). El pensamiento computacional implica habilidades como seguir instrucciones paso a paso para completar una tarea, reconocer patrones y desglosar tareas complejas en pasos más pequeños.

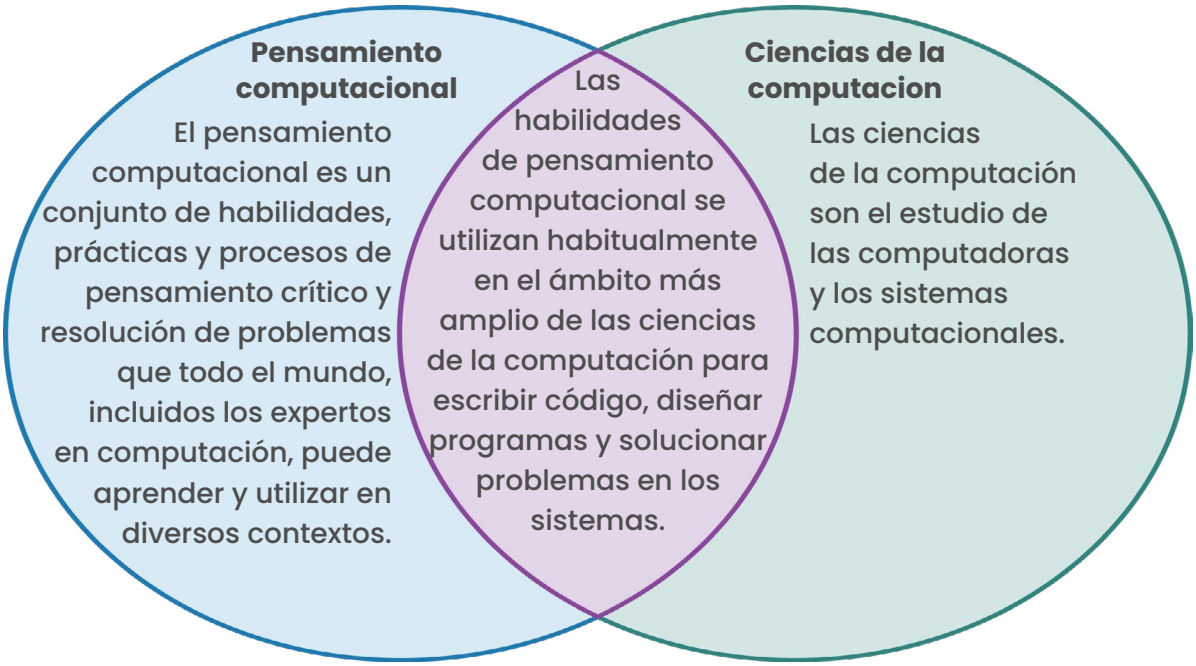
Considere estos ejemplos cotidianos que implican el pensamiento computacional:

- Puede que logre llevar a cabo una tarea compleja, como organizar una fiesta de cumpleaños, desglosándola en tareas más pequeñas, como enviar invitaciones, encargar la tarta y colocar globos. **(descomposición)**
- Si extravía algo, puede volver sobre sus pasos para averiguar dónde y cuándo lo tuvo por última vez. **(depuración)**
- Puede planificar sistemáticamente su recorrido por el supermercado comprando primero la fruta y la verdura, luego los productos secos y, por último, los lácteos. **(secuenciación)**

Las ciencias de la computación son el estudio de las computadoras y los sistemas computacionales (Bers et al., 2022). Los expertos en computación son personas que estudian los algoritmos y los procesos de computación y utilizan la tecnología para resolver problemas del mundo real. Realizan diversas tareas, como escribir y ejecutar código de computación, desarrollar y probar programas, así como solucionar problemas y mejorar los sistemas de computación (California State Board of Education, 2018; Google Inc. y Gallup Inc., 2016 a, 2016b; Garfield County School District, 2023). Todas estas tareas requieren competencias fundamentales en pensamiento computacional y habilidades avanzadas de programación. Sin embargo, los niños de tan solo tres años pueden aprender a programar y participar en el aprendizaje de las ciencias de la computación. Por ejemplo, pueden utilizar tarjetas o bloques codificados por colores que sirven como instrucciones de programación para hacer que un robot se mueva y emita sonidos (Monteiro et al., 2021; Bers, 2018).

Comprender los puntos en común y las diferencias entre el pensamiento computacional y las ciencias de la computación es esencial para apoyar las experiencias de aprendizaje temprano de los niños (véase la figura 1). No todos los niños acabarán convirtiéndose en expertos en computación. Sin embargo, todo el mundo —incluidos los niños— puede aprender a pensar de forma crítica, diseñar sistemas y resolver problemas del mundo real que se basen en habilidades, prácticas y procesos fundamentales del pensamiento computacional.

Figura 1. Definición del pensamiento computacional y las ciencias de la computación



Habilidades clave del pensamiento computacional en la primera infancia

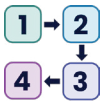
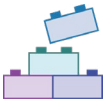
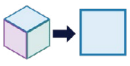
A lo largo de los años, los investigadores en educación en ciencias de la computación han desarrollado marcos de referencia para describir cómo los niños pequeños se involucran en el pensamiento computacional (por ejemplo, Barr y Stephenson, 2011; Bers, 2018; Grover y Pea, 2013). Aunque los marcos difieren en su lenguaje y alcance, todos ellos se centran en aspectos similares del pensamiento computacional. Por ejemplo, el Marco de Ciencias de la Computación para K–12 (2016) se centra en los patrones, la resolución de problemas, la representación y la secuenciación. Las «siete ideas fundamentales del pensamiento computacional» de Bers (2018) utilizan un vocabulario diferente, pero hacen hincapié en habilidades y prácticas similares. Por ejemplo, la secuenciación se denomina «algoritmos».

A pesar de las diferencias entre los marcos, hay seis habilidades de pensamiento computacional que se mencionan de forma sistemática en la bibliografía (consulte la tabla 1 para ver las definiciones y los ejemplos del mundo real):

- Algoritmos y secuenciación
- Reconocimiento de patrones
- Descomposición y modularidad
- Abstracción y representación
- Pruebas y depuración
- Lógica condicional y bucles

El pensamiento computacional implica habilidades de pensamiento crítico y resolución de problemas de las cuales todos, incluidos los científicos de la computación, pueden beneficiarse al aprenderlas y utilizarlas.

Tabla 1. Habilidades clave del pensamiento computacional en la primera infancia

Habilidad de pensamiento computacional	Ejemplos de la vida cotidiana
 **Algoritmos y secuenciación** • **Algoritmo:** conjunto de instrucciones paso a paso necesarias para resolver un problema o completar una tarea • **Secuenciación:** acción de ordenar objetos o instrucciones en un orden específico	• Una rutina matutina paso a paso **(algoritmo)** • Identificar el orden de los pasos para cepillarse los dientes **(secuenciación)**: (1) Aplique pasta de dientes en el cepillo. (2) Pase el cepillo bajo la llave. (3) Cepíllese los dientes durante dos minutos. (4) Escupa la pasta de dientes en el lavabo. (5) Enjuáguese la boca con agua.
 **Reconocimiento de patrones** Identificar o crear una secuencia de elementos que se repiten, como objetos, formas y números	• Confeccionar una pulsera con un patrón repetitivo de cuentas de colores (por ejemplo, siguiendo un patrón AB) • Cantar y dar palmas al ritmo de «Pat-a-Cake» con el educador, repitiendo un patrón de palmas
 **Decomposición y modularidad** • **Descomposición:** dividir un problema o una tarea grande en varios pasos más pequeños • **Modularidad:** combinar pasos más pequeños, o módulos, de diferentes maneras para resolver un problema o completar una tarea	• Desglosar todos los pasos necesarios para ordenar una habitación desordenada antes de decidir por dónde empezar **(descomposición)**: hay que tirar la basura, recoger los juguetes, pasar la aspiradora por el suelo, etcétera • Poner la mesa para seis niños a la hora de la merienda montando primero un cubierto con un plato, una taza y una cuchara, y repitiendo luego esta disposición para los otros cinco niños **(modularidad)**
 **Abstracción y representación** • **Abstracción:** extraer información clave sobre un objeto o concepto complejo • **Representación:** mostrar o reconocer un objeto o una idea mediante un símbolo	• Encontrar un plátano buscando una fruta amarilla con forma de C **(abstracción)** • Identificar una seña octogonal roja en la calle **(abstracción)** y reconocer que dicha seña indica que hay que detenerse **(representación)**

Habilidad de pensamiento computacional	Ejemplos de la vida cotidiana
Pruebas y depuración • **Pruebas:** comprobar si algo funciona tal y como se desea • **Depuración:** detectar un error y utilizar estrategias de resolución de problemas para solucionarlor	• Comprobar si una linterna funciona, identificar que no funciona porque le faltan pilas **(depuración)**, cambiar las pilas y volver a probarla para asegurarse de que la linterna funciona **(pruebas)** • Darse cuenta de que una pieza de un rompecabezas no está en el lugar correcto y girarla en diferentes orientaciones para averiguar dónde encaja **(depuración)**
Lógica condicional y bucles • **Lógica condicional:** utilizar sentencias «si... entonces» para tomar decisiones sobre qué hacer a continuación o para controlar el flujo de un algoritmo • **Bucle:** conjunto recurrente de pasos que se utiliza para determinar si una acción se repite y de qué manera	• Tomar decisiones sobre qué ropa ponerse en función del tiempo; por ejemplo: «SI hace sol, ENTONCES me pondré pantalones cortos» o «SI llueve, ENTONCES me llevaré el paraguas» **(lógica condicional)** • Crear una coreografía que consista en tres palmadas y un giro, y luego repetir la secuencia de acciones dos veces más **(bucles)**

Cómo aprenden y desarrollan los niños las habilidades de pensamiento computacional y científico

El desarrollo del pensamiento computacional comienza en las primeras etapas de la vida, a medida que los niños exploran su entorno y dan sentido al mundo que les rodea. Los bebés y los niños pequeños comienzan a identificar patrones y estructuras repetitivas (por ejemplo, una rutina diaria para irse a dormir) y a pensar de forma condicional (por ejemplo, «Si pulso este botón, entonces sonará un ruido»). A medida que los niños se desarrollan, practican el pensamiento crítico y el razonamiento como parte de sus rutinas diarias y de su juego. Aprenden a resolver problemas más complejos y a seguir instrucciones paso a paso para completar tareas. Por ejemplo, los niños de cuatro años



pueden cantar una canción para recordar los pasos que deben seguir para atarse los cordones de los zapatos.

Aunque los niños desarrollan habilidades de pensamiento computacional a través de las experiencias de aprendizaje cotidianas,

la exposición específica a experiencias de aprendizaje en ciencias de la computación ofrece beneficios. Las investigaciones más recientes indican que la exposición temprana a las ciencias de la computación puede ayudar a los niños a desarrollar una actitud positiva hacia esta disciplina. Por ejemplo, un estudio realizado con niños de primaria de entre 7 y 10 años reveló que la participación en un programa extraescolar de robótica y programación mejoraba la comprensión de los niños sobre las ciencias de la computación y la robótica, y aumentaba su motivación para seguir una carrera profesional relacionada con las ciencias de la computación (Ragusa y Leung, 2023). La exposición temprana a las ciencias de la computación también puede romper los estereotipos negativos que los niños puedan tener sobre quiénes son los expertos en computación y a qué se dedican (Hansen et al., 2017; De Wit et al., 2021; Master et al., 2021). Por ejemplo, numerosos estudios han revelado que las experiencias positivas con robots programables pueden aumentar el interés de las niñas por las ciencias de la computación, lo que puede contribuir a promover la igualdad de género en los campos de las ciencias, la tecnología, la ingeniería y las matemáticas (Girshin et al., 2023; Sullivan, 2019).

El desarrollo del pensamiento computacional comienza en las primeras etapas de la vida, a medida que los niños exploran su entorno y dan sentido al mundo que les rodea. A medida que los niños crecen, resulta beneficioso que se les exponga de forma específica a experiencias de aprendizaje en ciencias de la computación.

Los Estándares de Ciencias de la Computación de California para los niveles K–12 (California State Board of Education, 2018) y el Marco de Ciencias de la Computación para los niveles K–12 (K12CS, 2016) —en consonancia con otros marcos estatales e internacionales de ciencias de la computación— destacan la importancia de la exposición temprana a experiencias de aprendizaje relacionadas con el pensamiento computacional y las ciencias de la computación. Aunque estos marcos difieren en muchos aspectos, comparten algunas similitudes en su enfoque sobre cómo los niños aprenden y desarrollan habilidades de pensamiento computacional. Por ejemplo, reconocen que los niños pueden desarrollar habilidades fundamentales de pensamiento computacional a través de experiencias de aprendizaje activas y lúdicas. Especialmente en el caso de los alumnos más pequeños, estas experiencias suelen ser **«sin tecnología»** (lo que significa que no implican el uso de tecnología) y están integradas con otras áreas de contenido. Por ejemplo, cuando los niños pequeños agrupan calcetines por talla y color, no solo desarrollan habilidades matemáticas como la agrupación y la ordenación, sino que también demuestran la habilidad de pensamiento computacional de la abstracción, al comprender que unos calcetines son grandes y otros pequeños.

A medida que los niños se desarrollan, continúan explorando las habilidades de pensamiento computacional de forma activa y lúdica. Por ejemplo, cuando los niños en edad preescolar y de Primaria temprana juegan al juego «Simón dice», aprenden el concepto de lógica condicional («SI digo "Simón dice", ENTONCES realice esta acción»). A estas edades más avanzadas, los niños también pueden tener la oportunidad de participar en actividades de pensamiento computacional y ciencias de la computación **con tecnología integrada**. Las actividades con tecnología implican el uso de algún tipo de tecnología, como un robot o una aplicación de programación, que involucra a los niños en la codificación o la programación. **La codificación** o **la programación** es la «práctica de desarrollar un conjunto de instrucciones (código) que una computadora pueda comprender y ejecutar, así como depurar, organizar y aplicar dicho código a contextos adecuados de resolución de problemas» (Mills et al., 2021). La programación sumerge a los niños en una experiencia de aprendizaje activo basada en la teoría cognitiva del constructivismo

(aprender haciendo), con especial énfasis en el construccionismo (aprender creando). Al convertirse en creadores de artefactos tecnológicos, como el código informático, los niños pueden desarrollar habilidades de pensamiento crítico y de resolución de problemas, al tiempo que aprenden a expresar sus ideas de nuevas formas.

Implementación del pensamiento computacional y las ciencias de la computación en entornos de educación infantil

Los educadores pueden llevar a cabo actividades de pensamiento computacional y ciencias de la computación de diversas formas para apoyar el desarrollo y el aprendizaje de los niños. Estas actividades pueden realizarse con o sin tecnología, facilitarse a través de experiencias tanto estructuradas como abiertas, e impartirse de forma independiente como experiencias autónomas de pensamiento computacional y ciencias de la computación o integrarse en otras lecciones y áreas de contenido.

Actividades sin tecnología y con tecnología

Las actividades sin tecnología son muy recomendables en los entornos de educación infantil. En estas actividades, los niños exploran conceptos de pensamiento computacional de forma activa y lúdica. Por ejemplo, pueden utilizar tarjetas con historias para practicar la secuenciación, jugar a juegos que impliquen seguir o crear secuencias mediante el movimiento (como indicar a un amigo que se dirija a un lugar concreto utilizando instrucciones sencillas) o utilizar colecciones de objetos para practicar el agrupamiento y el reconocimiento de patrones. Las actividades sin tecnología introducen a los niños en el pensamiento computacional sin necesidad de pasar tiempo frente a una pantalla. Además, las actividades sin tecnología son más accesibles, ya que suponen una menor barrera económica. Los educadores no necesitan tener acceso a tabletas, computadoras o robots para llevar a cabo actividades de pensamiento computacional sin tecnología (Bers y Sullivan, 2019; Relkin y Strawhacker, 2021). Para obtener más ideas de actividades de pensamiento computacional sin tecnología y recursos curriculares, visite y consulte el folleto **«Recursos de programación para educadores y familias»**.

Las actividades «con conexión» implican el uso de la tecnología. Algunas tecnologías están compuestas por piezas tangibles o **físicas**. El kit de robótica KIBO, por ejemplo, no incluye pantallas (véase la figura 2a). Los niños ensamblan bloques de madera codificados por colores que contienen códigos de barras, utilizan el robot para escanear los códigos de barras en un orden concreto y observan cómo el robot realiza las acciones programadas. Otras tecnologías de programación son interfaces gráficas, o **basadas en pantallas**, como la aplicación de programación ScratchJr (véase la figura 2b). Con ScratchJr, los niños arrastran bloques codificados por colores para crear programas que controlan personajes, fondos y sus interacciones. Por ejemplo, los bloques azules de ScratchJr hacen que los personajes se muevan, los bloques morados cambian la apariencia de los personajes, y así sucesivamente. Los lenguajes de programación basados en bloques, como KIBO y ScratchJr, utilizan símbolos en lugar de texto, lo que los hace accesibles para los niños que aún no saben leer. Por último, otras tecnologías de programación combinan elementos físicos y basados en pantalla, y se denominan interfaces **híbridas**. Por ejemplo, el robot Blue-Bot (véase la figura 2c) puede programarse mediante los botones situados en la parte superior del robot o mediante una aplicación basada en pantalla. Para consultar más ejemplos de herramientas de programación físicas, basadas en pantalla e híbridas para niños, visite y explore el folleto **«Recursos de programación para niños»**.



Figura 2a. Kit de robótica KIBO



Figura 2b. Aplicación ScratchJr



Figura 2c. Robot Blue-Bot

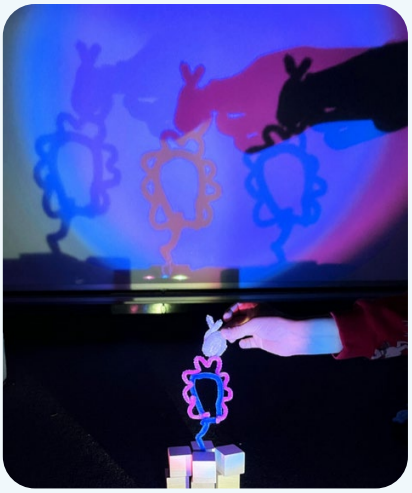
Los educadores pueden facilitar el aprendizaje del pensamiento computacional y las ciencias de la computación de los niños mediante una combinación de actividades sin computadora y con computadora. Las actividades de pensamiento computacional sin computadora pueden proporcionar experiencias con materiales concretos que ayuden a desarrollar la comprensión conceptual y las habilidades fundamentales de los niños, a las que podrán recurrir más adelante cuando participen en actividades de ciencias de la computación con computadora, de carácter más abstracto. Por ejemplo, la enseñanza de algoritmos a los niños puede comenzar con una actividad sin computadora realizada por parejas. Un niño es el programador y el otro actúa como un robot. El programador decide una secuencia de acciones y se las comunica paso a paso, utilizando tarjetas direccionales, al otro niño, o robot. En esta actividad, el programador está desarrollando un algoritmo para que lo ejecute el «robot». A continuación, esta actividad puede dar paso a una actividad con tecnología, como que los niños programen un robot KIBO para que se desplace de un lugar a otro.

Experimentación computacional

Las actividades de pensamiento computacional y de ciencias de la computación tienen mayor éxito en los entornos de aprendizaje temprano cuando los educadores utilizan estrategias eficaces de enseñanza y aprendizaje, tales como:

- crear oportunidades para el juego con un propósito,
- proporcionar materiales y espacios adecuados para la exploración activa,
- dar prioridad a los intereses y metas de los alumnos, y
- fomentar el aprendizaje colaborativo.

Estas cuatro estrategias constituyen componentes fundamentales de la experimentación («tinkering» en inglés) computacional. La experimentación consiste en manipular, construir, crear o desmontar materiales con el fin de poner a prueba ideas (Tinkering Studio, s. f.). El «tinkering» computacional es un término más reciente en este ámbito que se utiliza para describir cómo los educadores pueden involucrar a los niños en experiencias de aprendizaje de pensamiento computacional y ciencias de la computación de manera social, relevante, física e interdisciplinar. Por ejemplo, los educadores pueden ofrecer a los niños una amplia variedad de materiales para explorar, observar sus interacciones con los materiales y el entorno de aprendizaje, y plantear preguntas abiertas para potenciar su curiosidad natural. Como resultado, las exploraciones de los niños resultan significativas, están impulsadas por el propio niño y son flexibles en función de sus intereses y niveles de capacidad. Las metas finales de las experiencias de experimentación computacional son la alegría, la autonomía y el sentido de pertenencia.



Experimentación computacional mediante exposiciones interactivas

El Centro AIMS para la Educación en Matemáticas y Ciencias, en colaboración con los socios de Count Play Explore (CPE), ha elaborado guías para facilitadores sobre las exposiciones interactivas destinadas a niños y familias en torno a temas relacionados con STEAM, como los juegos de luces y sombras, los tubos de viento y la arena sensorial. Estas exposiciones constituyen ejemplos paradigmáticos de cómo fomentar la experimentación computacional de los niños. Por ejemplo, en una exposición sobre luz y sombras, los niños pueden experimentar con diversos materiales (como linternas, papel translúcido de colores, utensilios de cocina con agujeros, etc.) y explorar cómo la luz atraviesa los materiales de forma diferente para proyectar distintos tipos de sombras. Para ampliar esta experiencia de aprendizaje lúdico, los niños podrían trabajar en equipo para introducir en el espacio robots con sensores de luz y observar cómo estos interactúan con los materiales de la exposición. Estas exposiciones interactivas ilustran cómo los educadores pueden crear entornos de aprendizaje en los que los niños participen tanto en actividades sin tecnología como con tecnología, de manera que se fomente su curiosidad lúdica por los fenómenos científicos naturales.

Actividades independientes e integradas

Los educadores pueden llevar a cabo exploraciones lúdicas de pensamiento computacional y ciencias de la computación como actividades independientes, tal y como ilustran muchos de los ejemplos de este resumen. Las actividades independientes de pensamiento computacional permiten a los niños trabajar habilidades específicas de resolución de problemas, como resolver rompecabezas en Code.org o programar un robot para que se desplace de un lugar a otro. Si bien las actividades independientes pueden ofrecer valiosas oportunidades para el aprendizaje de los niños, integrar el pensamiento computacional y las ciencias de la computación en las materias troncales constituye un enfoque especialmente eficaz y práctico para los educadores. Al entrelazar las investigaciones de pensamiento computacional y ciencias de la computación con áreas de contenido como las matemáticas, el lenguaje y la alfabetización, las ciencias y las ciencias sociales, los educadores pueden hacer que el pensamiento computacional resulte más relevante y significativo para las experiencias de aprendizaje cotidianas de los niños (Rich et al., 2017). Por ejemplo, un niño podría volver a contar la historia de cómo se rompió el brazo desglosándola en lo que ocurrió primero, después, a continuación y por último (un algoritmo). Otro ejemplo consiste en poner en marcha un proyecto de robótica al final de un estudio de varias semanas sobre un tema concreto. Por ejemplo, los niños de segundo curso podrían aprender sobre diferentes danzas culturales de todo el mundo y, a continuación, programar robots para que interpreten una secuencia de danza cultural. Incluso podrían decorar su robot para que se pareciera a un bailarín de la comunidad cultural que hayan elegido.

Adoptar un enfoque integrado para el aprendizaje del pensamiento computacional y las ciencias de la computación no solo fomenta experiencias de aprendizaje significativas, sino que también aprovecha al máximo el tiempo lectivo sin restar importancia a otras áreas de contenido. Las investigaciones más recientes con niños pequeños demuestran que, cuando las

clases de ciencias de la computación se integran de forma reflexiva con las de matemáticas y alfabetización, los niños mejoran sus habilidades computacionales sin que ello afecte a su rendimiento en matemáticas y alfabetización (Bers et al., 2022; Yang y Bers, 2024). Además, la integración posiciona el pensamiento computacional como una forma universal de razonar con amplias aplicaciones en el mundo real, lo que ayuda a los educadores a reconocer que ya están fomentando el pensamiento computacional dentro de su plan de estudios habitual (Rich et al., 2017). Por lo tanto, aunque las actividades independientes pueden resultar eficaces, las exploraciones integradas de pensamiento computacional y ciencias de la computación ofrecen una estrategia potente y flexible para desarrollar pensadores computacionales en entornos de educación infantil.

Formación profesional en pensamiento computacional y ciencias de la computación para educadores de la educación de la infancia temprana

Son muchos los factores que influyen en la actitud, los conocimientos, la confianza y la capacidad de los educadores para aplicar el pensamiento computacional y las ciencias de la computación sus entornos de educación infantil. Las investigaciones muestran que muchos educadores no están familiarizados con el pensamiento computacional y las ciencias de la computación y pueden sentirse nerviosos a la hora de impartirlos (Ketelhut et al., 2020). Los educadores también pueden sentirse poco preparados para enseñar ciencias de la computación, ya que esta implica nuevos conceptos que no se les presentaron durante su formación inicial ni en sus experiencias de formación continua (California State Board of Education, 2018; Love et al., 2022; Sands et al., 2018).



Una formación profesional de alta calidad puede reforzar los conocimientos de los educadores sobre el pensamiento computacional y las ciencias de la computación y mejorar su confianza a la hora de impartirlos. La formación profesional de alta calidad consta de los siguientes siete elementos clave: (1) aprendizaje activo y lúdico, (2) enfoque claro y coherente en los contenidos, (3) oportunidades de colaboración, (4) modelar prácticas eficaces, (5) entrenamiento y apoyo individualizado, (6) retroalimentación y reflexión, y (7) oportunidades de aprendizaje sostenido (Darling-Hammond et al., 2017).

La tabla 2 ofrece un resumen general de tres estudios clave realizados en entornos de educación infantil (Marcella-Burdett et al., 2020; Rich et al., 2017; Yadav et al., 2018). En cada estudio, los educadores de la infancia temprana participaron en experiencias de formación profesional de alta calidad para desarrollar sus conocimientos sobre el pensamiento computacional y las ciencias de la computación, así como para aumentar su confianza a la hora de poner en práctica experiencias de aprendizaje relacionadas con el pensamiento computacional y las ciencias de la computación con los niños. El cuadro destaca las formas en que los siete elementos clave de una formación profesional de alta calidad quedaron patentes en la puesta en práctica de la formación profesional en cada uno de los tres estudios. En conjunto, estos estudios sirven de ejemplo de cómo los facilitadores, capacitadores y entrenadores de formación profesional podrían estructurar su formación para apoyar a los educadores en sus respectivos entornos.

El aprendizaje profesional de alta calidad puede reforzar los conocimientos de los educadores sobre el pensamiento computacional y las ciencias de computación, y mejorar su confianza a la hora de impartirlos.

Tabla 2. Ejemplos de estudios sobre formación profesional de alta calidad en las ciencias de la computación y pensamiento computacional en entornos de educación infantil

Elemento clave de un aprendizaje profesional de alta calidad	Yadav et al. (2018)	Rich et al. (2017)	Marcella-Burdett et al. (2020)
Aprendizaje activo y lúdico	Educadores que participaron en actividades de pensamiento computacional sin tecnología para adultos (por ejemplo, redactar instrucciones sobre la mejor forma de llegar a un destino popular desde su ubicación actual)	Docentes que participan en retos de diseño de ingeniería (por ejemplo, «Máquinas simples») y en clases de ciencias de la computación sin tecnología que utilizan Code.org y Scratch	Docentes que participan en actividades de pensamiento computacional y las ciencias de la computación (por ejemplo, programación de pingüinos, movimientos corporales con tarjetas direccionales y alfombrillas, y robots Sphero y Matatalab)
Enfoque de contenido claro y coherente	Las sesiones de desarrollo profesional se centraron específicamente en el pensamiento computacional	Las sesiones de desarrollo profesional se centraron específicamente en las ciencias de la computación	Las sesiones de desarrollo profesional se centraron específicamente en las ciencias de la computación

Elemento clave de un aprendizaje profesional de alta calidad	Yadav et al. (2018)	Rich et al. (2017)	Marcella-Burdett et al. (2020)
Oportunidades de colaboración	Los educadores colaboraron para diseñar lecciones y discutir estrategias adecuadas al nivel de desarrollo de los niños para llevar a cabo actividades de pensamiento computacional	Los educadores colaboraron para diseñar clases y discutir estrategias adecuadas al nivel de desarrollo de los niños para llevar a cabo actividades de las ciencias de la computación	Los educadores colaboraron para diseñar clases y discutir estrategias adecuadas al nivel de desarrollo de los niños para llevar a cabo actividades de las ciencias de la computación
Demostración de prácticas eficaces	Los facilitadores mostraron ejemplos de actividades de pensamiento computacional de nivel adulto para involucrar a los educadores como alumnos y compartieron estrategias comunes para llevar a cabo una enseñanza de las ciencias basada en la investigación	Los facilitadores mostraron ejemplos de lecciones utilizando el plan de estudios «Engineering is Elementary» (La ingeniería es elemental), las actividades de «Hour of Code» (La hora de codificar) en code.org y las actividades de programación con Scratch	Los facilitadores mostraron actividades prácticas que los educadores podrían llevar a cabo posteriormente con los niños
Entrenamiento y apoyo individualizado	Reuniones quincenales moderadas por un investigador en educación científica	Los investigadores en las ciencias de la computación y pensamiento computacional también actuaron como entrenadores para apoyar al personal docente y a los educadores	Los facilitadores del programa de desarrollo profesional (PL) ofrecieron diversas formas de entrenamiento a los docentes que participaron en la serie de sesiones de desarrollo profesional sobre programación

Elemento clave de un aprendizaje profesional de alta calidad	Yadav et al. (2018)	Rich et al. (2017)	Marcella-Burdett et al. (2020)
Comentarios y reflexión	Los talleres y las reuniones de seguimiento brindaron a los docentes la oportunidad de reflexionar sobre los conceptos y habilidades del pensamiento computacional	Los educadores y el profesorado reflexionaron conjuntamente sobre la visión de su centro educativo respecto a las competencias básicas en ingeniería y en ciencias de la computación	Las sesiones ofrecieron a los educadores la oportunidad de discutir cómo llevarían a cabo de forma e e las actividades en sus entornos de aprendizaje
Oportunidades para un aprendizaje continuo	Un taller inicial de nueve días seguido de sesiones quincenales	Sesiones semanales por cursos a lo largo de un año	Cuatro sesiones de entrenamiento ofrecidas a lo largo del año

Observación: Los siete elementos clave de un desarrollo profesional de alta calidad se basan en el marco de Darling-Hammond et al. (2017) para un desarrollo profesional eficaz de los educadores.

Apoyo a las experiencias de aprendizaje de pensamiento computacional y CS de las familias

Aunque gran parte de este informe discute el papel de los educadores, las familias también desempeñan un papel importante a la hora de apoyar las experiencias de aprendizaje de los niños en materia de pensamiento computacional y ciencias de la computación. Varios estudios examinan el aprendizaje de pensamiento computacional y ciencias de la computación en espacios de aprendizaje informales, como bibliotecas, museos y otros espacios comunitarios. Las exposiciones interactivas y los talleres para familias ofrecen valiosas oportunidades para que los niños participen en actividades de pensamiento computacional y ciencias de la computación junto con sus familiares (Govind, 2021). Al colaborar en proyectos de programación creativa, las familias pueden desarrollar nuevas habilidades, pasar tiempo de calidad juntas y conocer mejor los intereses de cada uno. Campana y Mills (2023) analizaron cómo los educadores de bibliotecas y museos incorporaban el pensamiento computacional a su trabajo con niños desde el nacimiento hasta los ocho años y sus familias. Los investigadores descubrieron que los educadores llevaban a cabo diversas actividades de pensamiento computacional, entre las que se incluían experiencias sin tecnología, con tecnología, integradas y de experimentación. Por ejemplo, los educadores organizaron puestos con las aplicaciones de programación Scratch y ScratchJr, programas de participación libre en los que las familias construían ciudades de cartón utilizando cartones reciclados de diversos tamaños y tipos, y actividades manuales en las que las familias elaboraban pulseras con código binario y diseñaban motivos coloridos.

Las familias también pueden participar en experiencias de aprendizaje de pensamiento computacional y las ciencias de la computación en casa. Por ejemplo, pueden jugar a juegos de lógica, planificar los pasos de una receta o participar en actividades de narración de historias y retos de construcción que animen a los niños a utilizar el pensamiento crítico y a practicar la resolución colaborativa de problemas. Las familias también pueden introducir herramientas de programación y robótica adecuadas a la edad de los niños, lo que les permite experimentar el aprendizaje del pensamiento computacional de una forma divertida y atractiva. Los educadores pueden apoyar a las familias en estas experiencias fomentando oportunidades para la indagación y la exploración. Por ejemplo, los educadores pueden animar a las familias a plantear a los niños preguntas abiertas, en lugar de darles las respuestas directamente, y a seguir la curiosidad y los intereses naturales de los niños. Para obtener más recursos con los que involucrar a las familias en experiencias de aprendizaje de pensamiento computacional y de ciencias de la computación, consulte el folleto **«Recursos de programación para educadores y familias»**.

Conclusiones clave y recomendaciones para facilitadores y responsables

Establecer una visión para la implementación de la educación en pensamiento computacional y ciencias de la computación en entornos de aprendizaje temprano.

Dada la variedad de enfoques para implementar el aprendizaje de las ciencias de la computación y el pensamiento computacional, los facilitadores deben colaborar con los responsables para establecer una visión de cómo se ve, se percibe y se experimenta el aprendizaje de las ciencias de la computación y el pensamiento computacional para los niños desde el nacimiento hasta los ocho años. Esta visión puede ayudar a los facilitadores a plantearse formas de adaptar los recursos para los educadores que trabajan con edades o cursos específicos, como bebés y niños pequeños, niños de kindergarten y niños de la Primaria temprana. Reflexione sobre cómo los facilitadores y los educadores podrían integrar el pensamiento computacional y la ciencia de la computación con otras áreas de contenido para enriquecer sus experiencias de aprendizaje en una amplia variedad de temas. La forma en que los responsables aborden la educación en pensamiento computacional y ciencia de la computación determinará el tipo de experiencias de desarrollo profesional que los facilitadores pondrán en práctica con los educadores, los niños y las familias en sus comunidades locales.

Identificar formas en las que los educadores y las familias pueden integrar el pensamiento computacional y la ciencia de la computación en sus prácticas actuales.

¡El pensamiento computacional está en todas partes! Todos los niños y educadores practican el pensamiento computacional como parte de sus rutinas diarias, sus experiencias y su plan de estudios. Por lo tanto, los educadores no necesitan aprender ni enseñar una nueva materia. En su lugar, pueden identificar áreas en las que resulte natural integrar el pensamiento computacional y las ciencias de la computación. Ayudar a los educadores a integrar el pensamiento computacional y las ciencias de la computación de forma activa y lúdica puede contribuir a aumentar su confianza y comodidad con estos temas, al tiempo que se aprovecha su experiencia previa en las materias que ya imparten. El mismo principio se aplica a las familias. Fomente la concienciación de las familias sobre cómo se pueden integrar el pensamiento computacional y las ciencias de la computación en las experiencias de aprendizaje cotidianas. Identifique recursos que ayuden a las familias a desarrollar una relación saludable con la tecnología, incluyendo formas de utilizarla

para promover un aprendizaje activo y lúdico del pensamiento computacional y las ciencias de la computación.

Explorar recursos y estrategias para fomentar una actitud favorable al pensamiento computacional y las ciencias de la computación entre los educadores y las familias.

Los educadores y las familias pueden experimentar sentimientos de ansiedad o miedo en relación con el pensamiento computacional y las ciencias de la computación. Identifique recursos que apoyen la actitud de los adultos, tales como actividades lúdicas de pensamiento computacional y las ciencias de la computación para adultos, actividades introductorias para llevar a cabo con los niños o vídeos de educadores y familiares que pongan en práctica experiencias de aprendizaje de pensamiento computacional y las ciencias de la computación junto a los niños. Fomente oportunidades para reflexionar sobre cualquier experiencia previa en pensamiento computacional y las ciencias de la computación y establezca conexiones con otras áreas de contenido. Destaque los ámbitos en los que el pensamiento computacional y las ciencias de la computación pueden integrarse en las rutinas diarias y en el juego puede mostrar a los educadores y a las familias que ya están abordando algunos de estos conceptos y prácticas.

La iniciativa CPE cuenta con varios conjuntos de recursos en línea para ayudar a los facilitadores a implementar el desarrollo profesional en pensamiento computacional y ciencias de la computación con los educadores:

- Pensamiento computacional
- Programación lúdica y creativa

Cada conjunto contiene recursos, entre los que se incluyen una presentación de diapositivas, una guía de facilitación, actividades para adultos y material impreso.

Apoyar a todos los educadores, familias y niños para que dispongan de oportunidades equitativas de aprendizaje en pensamiento computacional y programación.

Tal y como se ha descrito anteriormente en este informe, existen numerosas formas de implementar el pensamiento computacional y las ciencias de la computación en entornos de educación infantil (por ejemplo, actividades sin tecnología y con tecnología; herramientas de programación físicas, basadas en pantallas e híbridas). Tenga en cuenta todas estas opciones diferentes (y los costos asociados a ellas) a la hora de difundir recursos para que los educadores los implementen en sus entornos de aprendizaje. Por ejemplo, considere dar prioridad a los recursos gratuitos y de bajo costo para maximizar el alcance, al tiempo que ofrece opciones con distintos niveles de precio. Además, dedique tiempo a conocer los puntos fuertes y los contextos particulares de los educadores, los niños y las familias. Considere formas de promover oportunidades de aprendizaje de pensamiento computacional y las ciencias de la computación para comunidades con poblaciones de es cultural, étnica y socioeconómicamente diversas. Basarse en los principios del diseño universal para el aprendizaje y en pedagogías inclusivas puede ayudar a fomentar un aprendizaje equitativo de pensamiento computacional y las ciencias de la computación para todos.

Referencias

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads, 2*(1), 48–54.

Bers, M., Strawhacker, A., & Sullivan., A. (2022). *The state of the field of computational thinking in early childhood education.* OECD Education Working Papers, No. 274. OECD Publishing.

Bers, M. U. (2018). Coding and computational thinking in early childhood: The impact of ScratchJr in Europe. *European Journal of STEM Education, 3*(3).

Bers, M. U., & Sullivan, A. (2019). Computer science education in early childhood: The case of ScratchJr. *Journal of Information and Technology Education: Innovations in Practice, 18*, 113–138.

California State Board of Education. (2018). *Computer science standards for California public schools.* https://www.cde.ca.gov/be/st/ss/documents/csstandards.pdf

Campana, K., & Mills, J. E. (2023). Playing, tinkering, and problem solving: Understanding early computational thinking in libraries and museums. *Journal of Early Childhood Research, 21*(3), 369–383.

Darling-Hammond, L., Hyler, M. E., & Gardner, M. (2017). *Effective teacher professional development.* Learning Policy Institute.

De Wit, S., Hermans, F., & Aivaloglou, E. (2021). Children's implicit and explicit stereotypes on the gender, social skills, and interests of a computer scientist. In A. J. Ko & J. Vahrenhold (Eds.), *Proceedings of the 17th ACM Conference on International Computing Education Research* (pp. 239–251).

Garfield County School District. (2023). *What is computer science?* https://www.garfk12.org/what-is-computer-science/

Girshin, R. Z., Rosenberg, N., & Kukliansky, I. (2023). Early childhood robotics: Children's beliefs and objective capabilities to read and write programs. *Journal of Research in Childhood Education, 38*(2), 317–335.

Google Inc., & Gallup Inc. (2016a). *Diversity gaps in computer science: Exploring the underrepresentation of girls, Blacks and Hispanics.* http://goo.gl/PG34aH

Google Inc., & Gallup Inc. (2016b). *Trends in the state of computer science in U.S. K-12 schools.* https://services.google.com/fh/files/misc/trends-in-the-state-of-computer-science-report.pdf

Govind, M. (2021). Fostering computational thinking in homes and other informal learning spaces. In M. U. Bers (Ed.), *Teaching computational thinking and coding to young children.* IGI Global.

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher, 42*(1), 38–43.

Hansen, A. K., Dwyer, H. A., Iveland, A., Talesfore, M., Wright, L., Harlow, D. B., & Franklin, D. (2017). Assessing children's understanding of the work of computer scientists: The draw-a-computer-scientist test. In M. E. Caspersen (Ed.), *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education,* pp. 279–284.

K12CS. (2016). *K–12 Computer Science Framework.* http://www.k12cs.org

Ketelhut, D. J., Mills, K., Hestness, E., Cabrera, L., Plane, J., & McGinnis, J. R. (2020). Teacher change following a professional development experience in integrating computational thinking into elementary science. *Journal of Science Education and Technology, 29*(1), 174–188.

Love, T. S., Bartholomew, S. R., & Yauney, J. (2022). Examining changes in teachers' beliefs toward integrating computational thinking to teach literacy and math concepts in grades K-2. *Journal for STEM Education Research, 5*, 380–401.

Marcella-Burdett, J., Roth, A., Savelkouls, S., & Zur, O. (2020). *California statewide early math initiative in local communities: Building educator math capacity.* WestEd. https://www.wested.org/resources/california-statewide-early-math-initiative-local-communities/

Master, A., Meltzoff, A. N., & Cheryan, S. (2021). Gender stereotypes about interests start early and cause gender disparities in computer science and engineering. *PNAS, 18*(48).

Mills, K., Coenraad, M., Ruiz, P., Burke, Q., & Weisgrau, J. (2021, December). *Computational thinking for an inclusive world: A resource for educators to learn and lead.* Digital Promise. https://doi.org/10.51388/20.500.12265/138

Monteiro, A. F., Miranda-Pinto, M., & Osório, A. J. (2021). Coding as literacy in preschool: A case study. *Education Sciences, 11*(5). https://doi.org/10.3390/educsci11050198

Ragusa, G., & Leung, L. (2023). The impact of early robotics education on students' understanding of coding, robotics design, and interest in computing careers. *Sensors, 23*(23). https://doi.org/10.3390/s23239335

Relkin, E., & Strawhacker, A. (2021). Unplugged learning: Recognizing computational thinking in everyday life. In M. Bers (Ed.), *Teaching computational thinking and coding to young children* (pp. 41–62). IGI Global.

Rich, P. J., Jones, B. L., Belikov, O., Yoshikawa, E., & Perkins, M. (2017). Computing and engineering in elementary school: The effect of year-long training on elementary teacher self-efficacy and beliefs about teaching computing and engineering. *International Journal of Computer Science Education in Schools, 1*(1), 1–20.

Sands, P., Yadav, A., & Good, J. (2018). Computational thinking in K-12: In-service teacher perceptions of computational thinking. In M. S. Khine (Ed.), *Computational thinking in the STEM disciplines: Foundations and research highlights* (pp. 151–164). Springer International Publishing.

Sullivan, A. (2019). *Breaking the STEM stereotype: Reaching girls in early childhood.* Rowman & Littlefield Publishers.

Tinkering Studio. (n.d.). *Our work at the tinkering studio.* Exploratorium. https://www.exploratorium.edu/tinkering/about

Wing, J. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35.

Yadav, A., Krist, C., Good, J., & Caeli, E. N. (2018). Computational thinking in elementary classrooms: Measuring teacher understanding of computational ideas for teaching science. *Computer Science Education, 28*(4), 371–400.

Yang, D., & Bers, M. U. (2024). *Coding as another language: Impact on math and literacy achievement in early CS* [Conference session]. American Educational Research Association (AERA) Annual Meeting, Philadelphia, PA.

Count Play Explore (CPE) y el Superintendente de Escuelas del Condado de Fresno conservan todos los derechos exclusivos para reproducir, distribuir, exhibir y ejecutar la obra. Cualquier uso, reproducción o distribución no autorizada de esta obra, ya sea en su totalidad o en parte, está estrictamente prohibida sin el permiso por escrito del titular de los derechos de autor. Acceda a los *términos de uso de los materiales de CPE* en el enlace proporcionado.