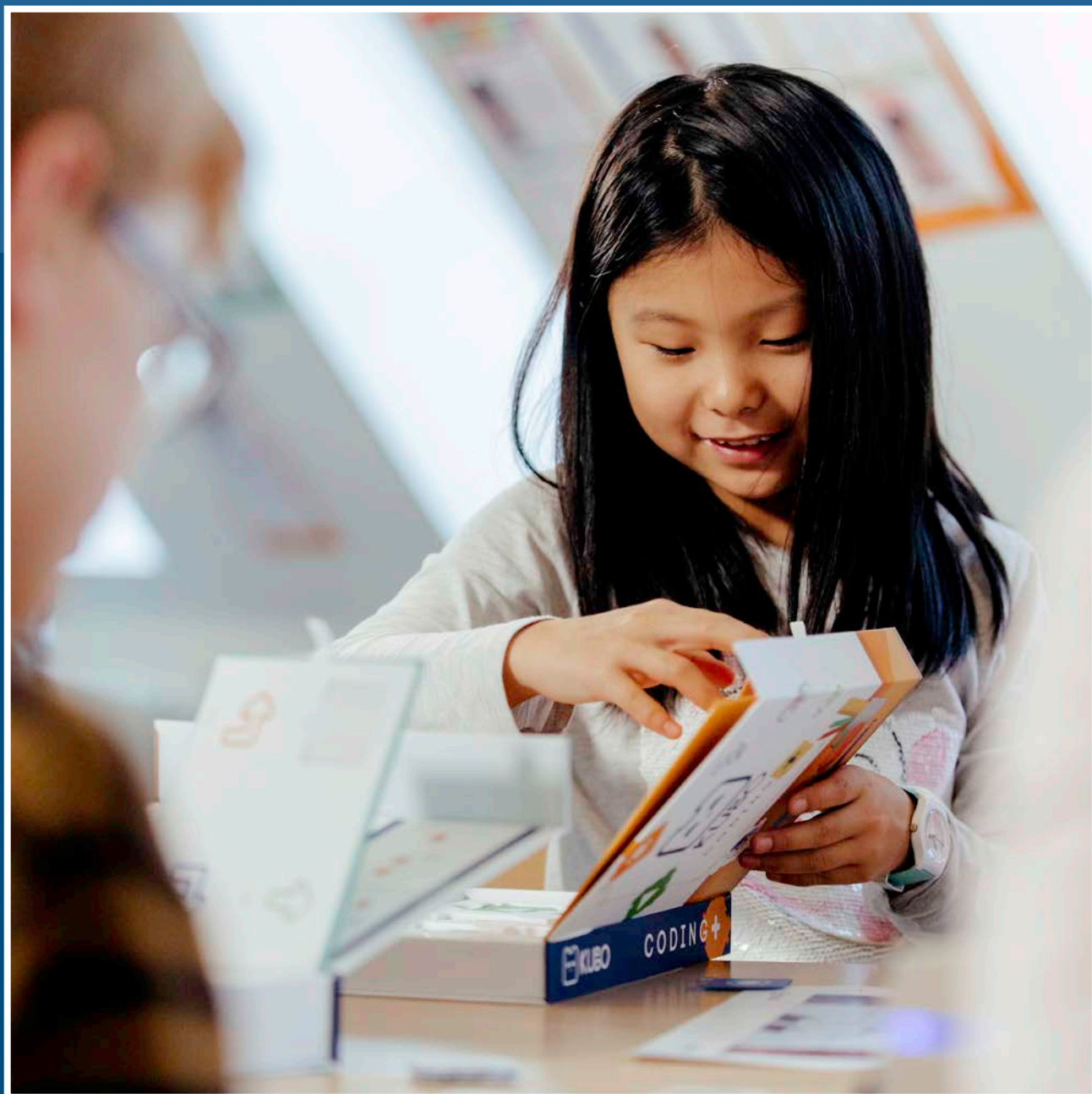




CREANDO
IMPACTO SIGNIFICATIVO

Contenidos

EMPODERANDO A NUESTROS CHICOS	3
ENSEÑA A LA NUEVA GENERACION	4
Ofrézcanle confianza para tomar el control.	5
Comiencen desde chicos a dominar las habilidades	6
La alfabetización digital es el futuro.	7
Enseñen a los niños como pensar	8
Piensa de forma innovadora e independiente	9
PROGRAMACIÓN EN EL AULA	10
Ampliar las fronteras	11
Apagar las computadoras	12
Los errores son buenos	13
Manténganlo simple	14
Referencias	15





Empoderando a nuestros chicos

No creo que la tecnología sea nuestra salvación. Lamentablemente, no podemos codificar nuestra salida de la crisis económica, ambiental o de la crisis humanitaria. Pero la tecnología en manos de seres humanos capaces contribuirá a un cambio positivo, y el pensamiento y la programación computacional van a contribuir a entender cómo y cuándo es apropiada la tecnología.

En KUBO, creemos que el empoderamiento de nuestros niños para prepararlos para el futuro requiere una forma lúdica y focalizada de igual a igual para un aprendizaje que sea relevante e inclusivo. Queremos contribuir a un ambiente de aprendizaje donde los chicos puedan “brillar” porque experimentan cómo es trabajar en colaboración dando espacio a las ideas de los demás y así construir la confianza de los demás a medida que aprenden juntos.

Aprendí a programar a los ocho años y en la medida que me di cuenta de todas las cosas asombrosas que podía crear con la programación, todo un mundo nuevo se abrió ante mí. Quiero que todos los niños experimenten el mismo sentimiento de maravilla y empoderamiento.

El plan de estudios de ciencias de la computación de hoy ofrece poco más que una imagen instantánea de lo que la

tecnología ya está haciendo basada en lo que ya sabemos. Para que el aprendizaje tecnológico tenga un valor real, debemos empujar continuamente los límites de lo que es posible hacerse dentro del aula. Como diseñadores de EdTech, es nuestra función proporcionar las herramientas y estimular la mentalidad que permite a los estudiantes explorar posibilidades imprevistas.

Quiero que los niños se conviertan en agentes en la búsqueda de un impacto creativo. La tecnología no debería ser una realidad virtual, sino una realidad mejor, y por esa razón nos comprometemos a ofrecer tecnología educativa que no elimine a los niños del mundo que los rodea, sino que les abre las mentes al mundo y a las personas que los rodean.

En este documento queremos compartir nuestra pasión con colegas educadores, responsables de las estrategias y a todos los socios del mundo educativo; queremos contribuir a una creciente comprensión del propósito y la visión de la tecnología educativa. Compartimos nuestros puntos de vista sobre el por qué la programación debe ser parte de la educación primaria y cómo creemos que puede integrarse en el aula. Es una invitación a tomar en serio la programación como una habilidad futura que permita a los niños participar de manera activa y creativa en la sociedad, en un mundo cada vez más digitalizado



Daniel Lindegaard

Fundador y Jefe de Tecnología



ENSEÑA A LA NUEVA
GENERACION A
PROGRAMAR



Ofrézcanle confianza para tomar el control.

Cuando el futuro esté conformado en gran medida por la tecnología, aquellos que sean capaces de entenderla, usarla e influir en ella recibirán un poder de decisión mucho mayor. Al proporcionar a todos los niños conocimientos y habilidades tecnológicas a través de la educación formal, el control se comparte y la enseñanza de códigos se convierte en un acto de democratización.

Existen predicciones generalizadas de una brecha digital cada vez más significativa entre los ciudadanos. Esta brecha se refiere a la división entre aquellos que tienen las habilidades para participar en una sociedad de alta tecnología y aquellos que no [1]. En el pasado la brecha ha sido una referencia para acceder a la tecnología, pero con los desarrollos tecnológicos más baratos y accesibles, la brecha en la cuarta revolución industrial se referirá a aquellos que tienen la capacidad de manipular datos, aquellos que están alfabetizados digitalmente y aquellos que no.

Creemos que la alfabetización digital debe ser enseñada con una dosis saludable de ética, y que los estudiantes deben ser alentados a pensar sobre el impacto que la creación tecnológica tiene en su comunidad y en la sociedad. Se les debe enseñar a pensar sobre un propósito ético y sostenible..

Incluso con tecnologías complejas cada vez más presentes en nuestra vida cotidiana, muchas personas no las entienden, ni creen que tienen mucho poder de influencia. No es necesariamente por miedo, sino por la impotencia que las personas experimentan cuando se enfrentan a la tecnología. El empoderamiento digital consiste en cambiar esa dinámica de poder para que, en lugar de experimentar que la tecnología tiene poder sobre la vida de uno, estamos facultados para llevar el control. Como dijo Rushkoff [2]: programe o sea programado.



Comiencen desde chicos a dominar las habilidades

Los niños parecen dominar la tecnología a una edad cada vez más temprana e intuitivamente exploran y utilizan una amplia variedad de tecnologías en sus vidas cotidianas. Sin embargo, si bien el hecho de ser educado con la informática hace que la mayoría de los niños sean maestros del uso de dispositivos, no significa que los entiendan o que puedan reflexionar críticamente sobre su uso.

La mayoría de las iniciativas de programación se han dirigido a niños mayores, pero existen buenas razones para introducir a la programación a niños de apenas cuatro años, para permitirles desarrollar los modelos

cerebrales para un mayor pensamiento abstracto. Una introducción temprana también permite que más niños desarrollen entusiasmo por la programación y significa que se pueden prevenir los estereotipos de carrera basados en el género dentro de STEM (Ciencia, Tecnología, Ingeniería y Matemáticas) [3].

Varios estudios sugieren que la introducción a la programación a una edad muy temprana les brinda a los niños la oportunidad de ser verdaderamente fluidos [4], similar a la enseñanza de otras habilidades básicas como la alfabetización y las matemáticas.

La alfabetización digital es el futuro.

La alfabetización digital no es simplemente la capacidad de leer y escribir código [5]. La alfabetización y el proceso de entender un lenguaje completamente nuevo incluyen mucho más que leer y escribir,... incluye contextos sociales y culturales para poder navegar con confianza y ser verdaderamente fluidos [6], [7].

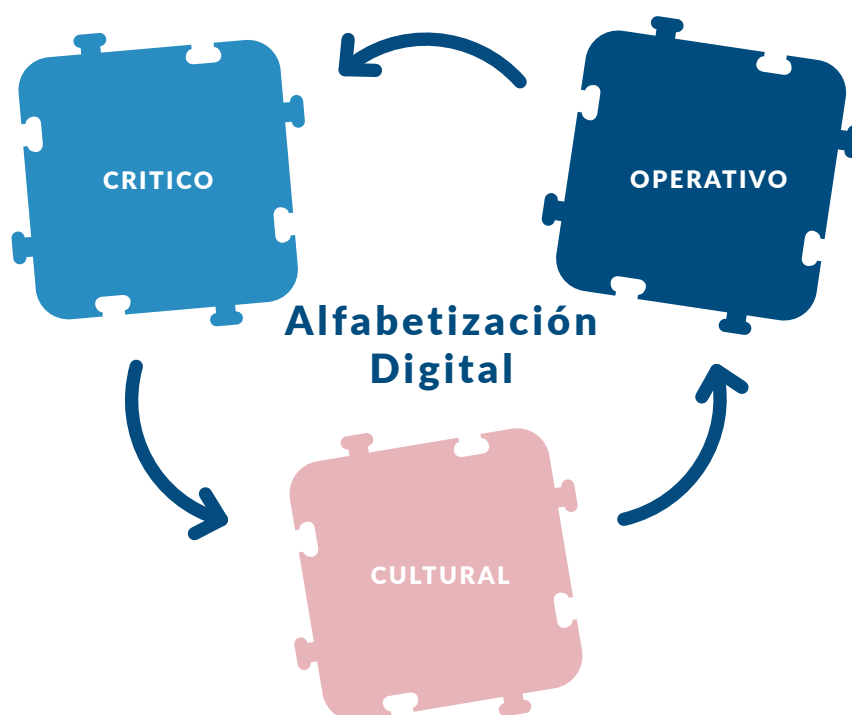
En pocas palabras, una computadora solo entiende los ceros y los binarios. De hecho, no es tan inteligente.

La programación es esencialmente el lenguaje que traduce el lenguaje humano al lenguaje informático. Existen varios lenguajes de programación, todos destinados a programar una computadora para ejecutar una serie de operaciones. Aunque cada uno es ligeramente diferente, los idiomas o dialectos de programación están basados en el mismo vocabulario y conceptos subyacentes.

La alfabetización, vista de manera holística, tiene aspectos operativos, culturales y críticos, y estos tres aspectos también son relevantes cuando se habla de alfabetización digital [6], [8].

El aspecto operativo de la alfabetización se refiere a las habilidades de lectura, escritura y creación de significados, y poder acceder a la información. Es esencialmente comunicación. El aspecto cultural se centra en la formación del sentido y el aspecto crítico sobre la capacidad para interactuar de manera crítica con textos y dispositivos digitales, así como con cuestiones de poder y de representación [6].

Las palabras no existen en un vacío, y tampoco el código, y las tres dimensiones de la alfabetización son importantes para una comprensión completa de la programación. Por lo tanto, la programación en educación no solo debe percibirse como la habilidad de comunicación, sino que también debe reconocerse por sus aspectos culturales y críticos.



Enseñen a los niños como pensar

Vamos a enseñar a los niños cómo pensar, ...no qué pensar. Vamos a desarrollar la forma de pensar computacional del siglo XXI.

La definición del siglo XXI que nos ofrece Jeanette Wing definiendo el pensamiento computacional como “resolver problemas, diseñar sistemas y comprender el comportamiento humano, basándose en los conceptos fundamentales de la informática” [9]. También ella afirmó que el pensamiento computacional involucra los procesos de pensamiento necesarios para encontrar soluciones a los problemas, formulados de tal forma que puedan ser efectivamente implementados por las computadoras.

En contextos educativos K-12, el concepto de computación fue introducido ya en los años 80 por Seymour Papert y su programa pionero LOGO [10]. Se abandonó de forma rápida y errónea, pero la importancia de la programación ha vuelto a entrar en terreno de la educación. Se utilizan varias definiciones, pero para ponerse de acuerdo sobre los conceptos centrales que incluyen abstracción, pensamiento algorítmico, automatización, descomposición y generalización,

relacionadas con las prácticas de depuración, colaboración, creatividad y la capacidad de lidiar con el planteamiento de problemas[11].

La programación y el pensamiento computacional están estrechamente relacionados. El aprendizaje de conceptos de programación se puede ver como una serie de habilidades prácticas, y los procesos de pensamiento computacional como habilidades cognitivas [12]. Si bien la programación sería más fácil de aprender si ya tuvieras un buen conocimiento del pensamiento computacional, aprender a programar es, al mismo tiempo, una muy buena forma de practicar el pensamiento computacional.

Por lo tanto, aprender a programar no consiste solo en aprender a codificar, sino que también se trata de abstracción y pensamiento algorítmico, dividir un problema en un proceso paso a paso para guiar a alguien o algo a completar una tarea o resolver un problema [13]. Esta es la razón por la que el pensamiento computacional no solo es relevante para las computadoras, sino que se puede transferir a una variedad de situaciones y problemas en la vida, como hornear un pastel, seguir paso a paso una receta de cocina o armar un mueble.

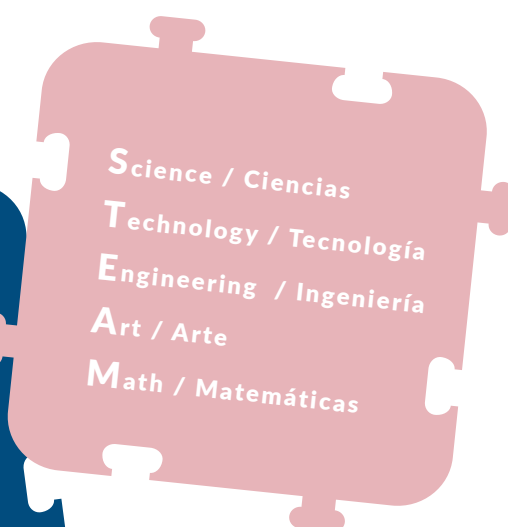
CONCEPTOS DE PROGRAMACIÓN



PENSAMIENTO COMPUTACIONAL



DISCIPLINAS





Piensa de forma innovadora e independiente.

Nadie puede esperar que uno sepa lo que va a suceder en el futuro, pero las tendencias actuales sugieren que existe un creciente deseo de ir más allá de las ciencias STEAM para nutrir, practicar y desarrollar habilidades que los robots, como máquinas, no son capaces de llevar a cabo [14], [15]. Entre otras cosas, esto significa ser creativo; capaces de pensar de forma innovadora e independiente.

Para utilizar las palabras de Stephan Turnipseed, vicepresidente ejecutivo y director de estrategias de Pitsco Education:

“Debemos preparar a los niños para que sean únicamente humanos en un mundo cada vez más inhumano”. [16]

Las competencias de STEAM son, sin duda, importantes para el éxito en el mundo de mañana, pero cada vez es más probable que vayan acompañadas de más habilidades para la vida basadas en el carácter. El informe del Foro Económico Mundial sobre el futuro del empleo, incluye un resumen de las habilidades que serán las más buscadas

en el futuro [14]. Esa visión general muestra que la creatividad ha saltado de la décima habilidad más buscada en el 2015, a la tercera más necesaria para el año 2020. Del mismo modo, la Inteligencia Emocional, que ni siquiera figura en la lista del 2015, ahora es la sexta más buscada.

Programar no solo es aprender esa habilidad en particular, sino también darles a los niños un conjunto más amplio de competencias: esto incluye pertenencia social y valores, inteligencia moral, ser creativo y proponer nuevas ideas, desarrollar habilidades con énfasis en lo humano y tomar decisiones éticas. Según un estudio reciente, los beneficios educativos de la programación en el aula durante los primeros años no solo están relacionados con las habilidades STEM, como las matemáticas y la resolución de problemas, sino también con el razonamiento verbal, causal y social [17]

Importantes Habilidades

Fuente: Future of Jobs Report, World Economic Forum

2020

1. Solución de problemas complejos
2. Pensamiento crítico
3. Creatividad
4. Administración de personal
5. Coordinación con otros
6. Inteligencia Emocional
7. Juicio y toma de decisiones
8. Orientación al servicio
9. Negociación
10. Flexibilidad cognitiva

2015

1. Solución de problemas complejos
2. Coordinación con otros
3. Administración de personal
4. Pensamiento crítico
5. Negociación
6. Control de calidad
7. Orientación al servicio
8. Juicio y toma de decisiones
9. Escuchar activamente
10. Creatividad



PROGRAMACIÓN EN EL AULA DE CLASE



Ampliar las fronteras

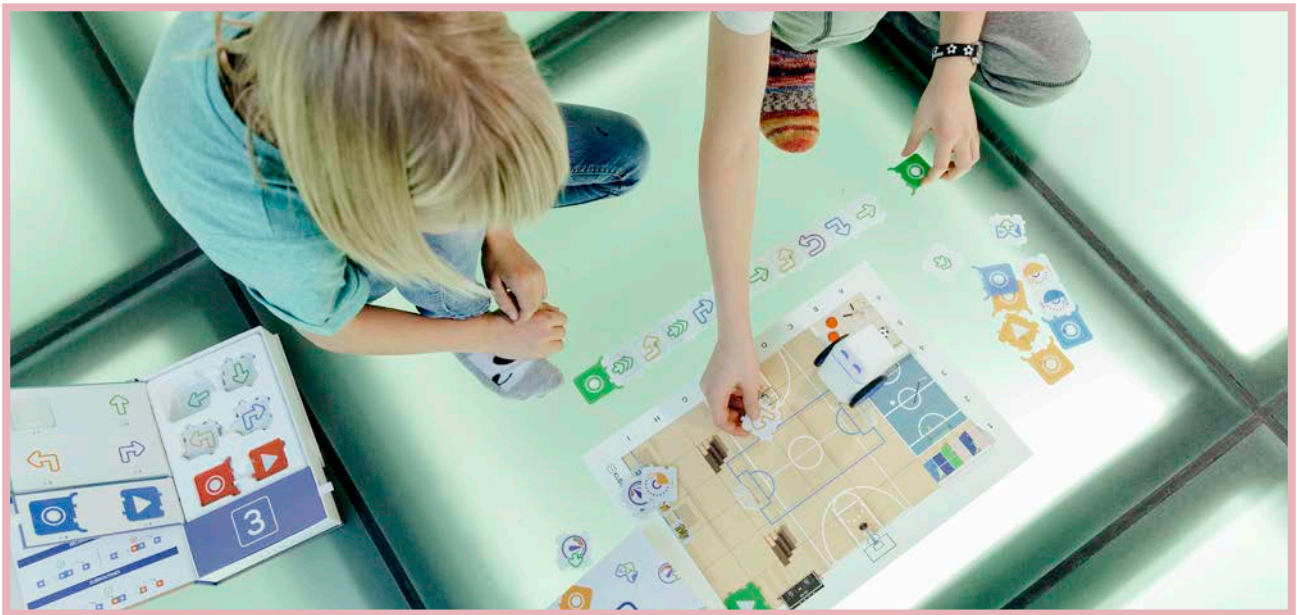
Teniendo en cuenta la responsabilidad de la educación formal para preparar a los niños no solo para una carrera profesional sino también para una vida con sentido y significado, [15] las competencias de programación y pensamiento computacional han ganado su lugar en la educación primaria. La pregunta es, ¿cómo se entrega mejor este avance?

Creemos en los principios de pisos bajos y techos altos; que las cosas deben ser relativamente fáciles de entender, pero bastante difíciles de dominar, haciéndolas

accesibles y atractivas. Mitchel Resnick y su equipo de MIT también destacan la importancia de los “fronteras ampliadas”, lo que permite un amplio conjunto de

intereses para motivar y aplicar conocimiento [18]. Una buena práctica de entregar la programación es integrarla en las asignaturas existentes y relacionarla con lo que los maestros ya hacen en su aula [19].

Con el ritmo actual del desarrollo tecnológico, es primordial que nosotros, como proveedores de Educación tecnológica, ayudemos a los maestros, al hacer productos que sean fáciles de configurar y que estén realmente diseñados para el entorno del aula. También debemos proporcionar capacitación a los maestros, junto con la inspiración del currículo y el apoyo estructurado que guíe a los estudiantes a una participación más profunda y un pensamiento de nivel superior [19], [20].



Apagar las computadoras

La enseñanza no conectada del pensamiento computacional y la programación está ganando reconocimiento [4], y aunque puede parecer contraria a la intuición eliminar de la ecuación a las pantallas de video PC, existe evidencia suficiente para respaldar que esto fortalece el proceso de aprendizaje. Debido a las crecientes preocupaciones sobre la cantidad de tiempo de exposición a las pantallas a la que están expuestos los niños y un mercado abarrotado de recursos, también es difícil para los maestros determinen qué recursos son los más efectivos.

El conocido trabajo de Seymour Papert, “Mindstorms” y su teoría del construccionismo, ha establecido firmemente los beneficios de tener “objetos con los que pensar” [10]. Los objetos facilitan los modelos mentales al proporcionar una representación concreta para mejorar la comprensión de lo abstracto.

En los primeros años del desarrollo de un niño, el aprendizaje se basa en sentir y manipular el mundo que los rodea. Sólo más tarde aprenden a pensar de manera abstracta [21], [22]. Por lo tanto, las herramientas concretas de aprendizaje son esenciales en los primeros años para adaptarse a la forma de aprendizaje de los niños.

Al quitar las pantallas de la ecuación de aprendizaje en el aula de los primeros años, los niños no solo se benefician del aprendizaje práctico, sino también de una mejor interacción entre pares. Esto significa que, en lugar de tener líneas de visión paralelas apuntando a una pantalla, tienen la oportunidad de conectarse, compartir, hablar e inspirar o inspirarse de una manera más directa e íntima con sus compañeros.

Los errores son buenos

La programación es una de las pocas áreas donde es aceptable cometer errores y esto ayuda a crear un entorno de aprendizaje que acepta los errores como parte natural del aprendizaje. Un código rara vez funciona la primera vez, y uno de los elementos esenciales para convertirse en un buen programador es la capacidad de identificar y corregir errores en el código. Este proceso se llama depuración y es uno de los procesos en el pensamiento computacional [9], [13], [20].

En una sociedad que se centra cada vez más en la perfección, la programación es una oportunidad para aprender a sentirse cómodos con los procesos de prueba y error en un proceso interactivo. Al programar, los niños no deben necesariamente esforzarse por alcanzar la perfección, sino que se les debe alentar a experimentar con varias soluciones creativas.

Los niños deben practicar la articulación e identificación de errores en el código, y deben hacer sugerencias sobre cómo solucionarlo. En programación, a menudo hay más de una solución.

a un problema, que también brinda a los niños el beneficio de las comparaciones y discusiones sobre cómo y por qué un algoritmo puede ser mejor que otro.

Enfrentar dificultades o desafíos en el camino cuando se aprende programación es casi inevitable, pero al hacer que la programación sea interesante y al crear una cultura de aprendizaje donde se permiten los errores, los niños estarán más dispuestos y decididos a intentar, intentar e intentar nuevamente hasta que encuentren una solución al problema.



Manténganlo simple

Creemos que la programación no debe estar maquillada, sino que debe enseñarse en su sentido más puro, de la misma manera que se enseña el aprendizaje del alfabeto para llegar al objetivo de poder leer y escribir.

La programación no debe ser sometida a procesos de gamificación o tratada como un elemento secundario, por ejemplo, al ser introducida en una tarea que tiene un mayor enfoque en la construcción de un robot u otros mecanismos. Los conceptos básicos de los conceptos de programación y la terminología de programación deben ser el enfoque principal del aprendizaje en las etapas iniciales. Esto no significa que el aprendizaje no pueda ser lúdico, pero la actividad no debe disfrazar o diluir el objetivo principal del aprendizaje.

La programación debe ser conversacional. Los ejemplos de la vida real y aquellos más cercanos a los intereses existentes de los niños ayudan a hacer que los conceptos y términos abstractos sean más comprensibles.

Y estos no necesitan estar centrados en los conceptos de la enseñanza de STEM. Muy a menudo, los ejemplos de programación en contextos del mundo real están relacionados con funciones tradicionales, como la construcción de puentes, la navegación de barcos o la

programación de máquinas para procesar oficios. Existen muchas más formas de garantizar que la programación y la computación tengan más sentido para todos los niños. Los ejemplos del uso del código en la vida real pueden ser desde la instalación de luz hasta la decoración o interacción estética, hasta el diseño de productos y entornos que influyen en el entorno y el bienestar humano.

Al comenzar con las actividades diarias normales, como seguir una receta o planificar una ruta en un mapa, los niños aprenden a comprender los conceptos básicos de los algoritmos. Ellos los experimentan

como secuencias visibles, tangibles y lógicas que se pueden manipular y reutilizar de nuevas maneras, abriendo nuevas posibilidades.

La programación se utiliza en todos los ámbitos de la vida, y la verdadera oportunidad para nuestros jóvenes aprendices es comprender plenamente su uso e imaginar cómo se puede utilizar en el futuro; para vivir en un mundo donde no hay suficiente empleo, para luchar en un mundo que dominarán, la programación y el pensamiento computacional son habilidades esenciales.



Referencias bibliográficas

- [1] A. Carvin, 'Mind the Gap: The Digital Divide as the Civil Rights Issue of the New Millennium.', *Multimed. Sch.*, vol. 7, no. 1, pp. 56–58, 2000.
- [2] D. Rushkoff, *Program or be programmed: Ten commands for a digital age*. Or Books, 2010.
- [3] S. S. Metz, 'Attracting the engineers of 2020 today', *Women Minor. Sci. Technol. Eng. Math. Upping Numbers*, pp. 184–209, 2007.
- [4] M. U. Bers, *Coding as a playground: Programming and computational thinking in the early childhood classroom*. Routledge, 2017.
- [5] A. A. DiSessa, *Changing minds: Computers, learning, and literacy*. Mit Press, 2001.
- [6] S. Bulfin and K. McGraw, 'Digital literacy in theory, policy and practice: old concerns, new opportunities', *Teach. Digit. Technol. Big Issues Crit. Quest.*, pp. 266–281, 2015.
- [7] A. Vee, 'Understanding computer programming as a literacy', *Lit. Compos. Stud.*, vol. 1, no. 2, pp. 42–64, 2013.
- [8] B. Green and C. Beavis, *Literacy in 3D: An integrated perspective in theory and practice*. Australian Council for Educational Research, 2012.
- [9] J. M. Wing, 'Computational thinking', *Commun. ACM*, vol. 49, no. 3, p. 33, Mar. 2006.
- [10] S. Papert, *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc., 1980.
- [11] S. Bocconi and A. Chiocciariello, 'The Nordic Approach to Introducing Computational Thinking and Programming in Compulsory Education', Jan. 2018.
- [12] K. Tsarava, K. Moeller, N. Pinkwart, M. Butz, U. Trautwein, and M. Ninaus, 'Training computational thinking: Game-based unplugged and plugged-in activities in primary school', in *European Conference on Games Based Learning*, 2017, pp. 687–695.
- [13] S. Grover and R. Pea, 'Computational thinking in K–12: A review of the state of the field', *Educ. Res.*, vol. 42, no. 1, pp. 38–43, 2013.
- [14] World Economic Forum, 'The future of jobs: employment, skills and workforce strategy for the Fourth Industrial Revolution | VOCEDplus, the international tertiary education and research database'. [Online]. Available: <https://www.voced.edu.au/content/ngv:71706>. [Accessed: 01-Feb-2019].
- [15] OECD, 'The future of education and skills: Education 2030', 2018.
- [16] 'Preparing Young Learners for Life Beyond the Fourth Industrial Revolution', *Education Reimagined*, 23-Oct-2018. [Online]. Available: <https://education-reimagined.org/life-beyond-fourth-industrial-revolution/>. [Accessed: 25-Mar-2019].
- [17] A. Strawhacker and M. U. Bers, 'What they learn when they learn coding: investigating cognitive domains and computer programming knowledge in young children', *Educ. Technol. Res. Dev.*, pp. 1–35.
- [18] M. Resnick and K. Robinson, *Lifelong kindergarten: Cultivating creativity through projects, passion, peers, and play*. MIT Press, 2017.
- [19] A. Yadav, H. Hong, and C. Stephenson, 'Computational thinking for all: pedagogical approaches to embedding 21st century problem solving in K-12 classrooms', *TechTrends*, vol. 60, no. 6, pp. 565–568, 2016.
- [20] V. Barr and C. Stephenson, 'Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community?', *Inroads*, vol. 2, no. 1, pp. 48–54, 2011.
- [21] P. Marshall, 'Do tangible interfaces enhance learning?', in *Proceedings of the 1st international conference on Tangible and embedded interaction*, 2007, pp. 163–170.
- [22] Y. B. Kafai and M. Resnick, *Constructionism in practice: Designing, thinking, and learning in a digital world*. Routledge, 2012.



districalc

Distribuidores oficiales de Kubo en Chile
www.districtalc.com



All rights reserved ©2019 · KUBO Robotics ApS · Niels Bohrs Allé 185 - 5220 Odense SØ · Denmark · SE/CVR-nr.: 37043858

www.kubo.education