

A photograph of three young women of diverse backgrounds looking intently at a laptop screen. The woman on the left has long, wavy brown hair and is wearing a purple and black plaid shirt. The woman in the middle has dark hair pulled back, wears glasses, and a white shirt. The woman on the right has dark hair and is wearing a blue denim jacket over a white shirt. They are all focused on the laptop in front of them.

# Enseñar Ciencias de la Computación:

Creando oportunidades para los jóvenes de América Latina

*Preparado por Ignacio Jara y Pedro Hepp para Microsoft América Latina*



# 1. Introducción

Las tecnologías digitales están transformando el mundo en que vivimos, incluyendo la educación. Los sistemas educativos buscan adaptarse a este nuevo contexto incorporando computadores e Internet en la docencia y en la gestión y están preparando a las nuevas generaciones para su aprovechamiento en las diferentes esferas de la vida. Se trata de un proceso de adaptación que está permanentemente ajustando su rumbo ante los continuos cambios en el entorno tecnológico y social al que buscan responder.

La *alfabetización digital* de los jóvenes ha estado, hasta ahora, principalmente orientada a ser buenos usuarios de las aplicaciones de la tecnología. En los últimos años, sin embargo, se ha levantado un creciente interés por ampliar esta formación tecnológica, educando a los jóvenes en la comprensión de cómo funcionan las tecnologías y sus principios fundantes, de manera de convertirlos en agentes creativos del mundo digital y no solo en sus consumidores. Así, muchos países desarrollados están revisando sus currículos escolares para incorporar conceptos de *ciencias de la computación* y desarrollar *pensamiento computacional* en los estudiantes; y diversas empresas y organizaciones sin fines de lucro han desplegado iniciativas para promover la *programación* de dispositivos entre los jóvenes convirtiéndose en agentes de educación no formal de *ciencias de la computación*. Se espera que estos conocimientos potencien el desarrollo cognitivo de los jóvenes, faciliten la empleabilidad y amplíen el interés por carreras tecnológicas, fortaleciendo el crecimiento económico de los países.

Este movimiento a favor de la enseñanza de la computación no es nuevo, pero trae novedades. Desde la década de los 70's ha existido la idea de que enseñar a programar a los niños puede ser una experiencia que, junto con ser provechosa para su desarrollo cognitivo, incentiva que más jóvenes se interesen por las profesiones que se requieren en la industria tecnológica. En esos años, muchas escuelas secundarias comenzaron con cursos de programación utilizando los lenguajes asequibles de la época, como BASIC. Por ejemplo, Inglaterra, país históricamente muy activo en estas materias, lanzó en 1981 su primera política nacional denominada *Micromputers for Schools* enfocada en la enseñanza de este lenguaje (Selwyn, 2002). A comienzo de la década de los 80's, el académico de origen sudafricano y discípulo de Piaget, Seymour Papert, creó junto a su grupo en el MIT el lenguaje LOGO, pensado para que los niños aprendieran matemáticas en el contexto lúdico y riguroso del razonamiento lógico requerido para resolver problemas y programar una computadora (Papert, 1993). Las ideas y herramientas generadas por Papert fueron adoptadas en muchos lugares durante los años siguientes y siguen teniendo una profunda influencia hasta nuestros días. En Latino América, por ejemplo, Costa Rica inició en 1987 una política nacional que sigue hasta hoy para enseñar programación en las escuelas primarias para desarrollar pensamiento lógico y resolución de problemas (Muñoz et al, 2014). Desde esa época, países y escuelas han mantenido cursos de informática, principalmente en secundaria, que sin perjuicio de sucesivos cambios de enfoque, han mantenido algún interés en la programación, al menos como materia opcional para los alumnos más interesados.

En los 90's el foco de atención giró hacia el uso de computadores e Internet como apoyo a la enseñanza y el aprendizaje en forma transversal en el currículum escolar. Ya no se trataba de aprender sobre los computadores sino con los computadores; consecuentemente, la mayoría de los cursos de informática existentes se orientaron a la utilización de las aplicaciones computacionales, como procesadores de texto, planillas electrónicas, bases de datos, etc., para resolver problemas cotidianos y asegurar las habilidades tecnológicas básicas requeridas para el uso de las TIC en el resto de las materias escolares, dejando de lado el énfasis más técnico de la programación. Con el paso de los años, sin embargo, en la medida que los niños crecían cada vez más inmersos en ambientes ricos en tecnologías digitales en sus propios hogares, estos cursos de informática se hicieron crecientemente aburridos e irrelevantes (Cobo, 2014). En paralelo, los círculos vinculados a la industria y la academia tecnológica veían con preocupación que la creciente competitividad de la economía digital no estaba acompañada de un mayor interés de los jóvenes por las profesiones que sostienen esta industria; y que la escuela no estaba acercando a las nuevas generaciones a estas disciplinas.

En este contexto, en la década del 2000 vuelve a emerger con fuerza la idea de que todos los estudiantes debían ser expuestos a las bases teóricas y prácticas de las tecnologías digitales durante su experiencia escolar y que ahora debía hacerse de manera más profunda. En 2006, Jeannete Wing, directora del Departamento de Ciencias de la Computación de la Universidad de Carnegie Mellon en Pittsburg, Estados

Unidos, escribió un influyente artículo planteando que el *pensamiento computacional* debía ser considerado una habilidad central del Siglo XXI que todos los alumnos debiesen aprender, abriendo un rico debate sobre los alcances e implicancias de esta visión (Wing, 2006). En los últimos años, muchos países han comenzado a revisar sus currículums poniendo en el centro de sus cursos de informática la enseñanza de *ciencias de la computación*. El Departamento de Educación de Inglaterra, por ejemplo, hizo un cambio radical en este sentido y desde 2014 esta disciplina se enseña desde los primeros años de primaria; a comienzos de 2016, el presidente de Estados Unidos lanzó una campaña denominada *Computer Science for All* para fortalecer la enseñanza de esta materia en las escuelas americanas; y en 2014 European Schoolnet publicó un estudio que muestra que la mayor parte de los países de ese continente ya consideran o están pensando incorporar la *programación* de computadores en sus currículum (EUN, 2014).

Adicionalmente, se han multiplicado iniciativas privadas impulsadas por empresas del rubro, fundaciones y ONG's que invitan a los jóvenes a aprender a programar como una forma de complementar la escuela y ampliar sus posibilidades de aprender, crear y conseguir trabajo. Asimismo, hay un creciente interés en el movimiento makers y de robótica educativa, que ofrecen ambientes propicios para el desarrollo de proyectos de programación cuyos productos salen de las pantallas para dar vida a artefactos concretos que se mueven, recogen datos del ambiente o controlan dispositivos electro-mecánicos<sup>1</sup>. Para sorpresa de muchos, estas iniciativas han tenido una amplia aceptación en el público.

<sup>1</sup>Estos movimientos utilizan diferentes tipos de dispositivos, tarjetas electrónicas y sensores que se conectan con el computador para ser programados y crear así nuevos artefactos y resolver problemas. Son famosas las tarjetas como Raspberri Pi y Arduino, entre otras.

Ejemplo de ello son los cientos de miles de entusiastas que se reúnen en las ferias de *makers* organizadas desde hace varios años en diversos países<sup>2</sup>; los ocho millones de la tarjeta programable de bajo costo *Raspberry Pi* que se han vendido en los últimos cuatro años<sup>3</sup>; y las más de 200 millones de personas de todo el mundo que han utilizado los recursos y participado en las actividades impulsadas por iniciativas como *La Hora del Código*<sup>4</sup> o *YouthSpark*<sup>5</sup> que promueven la enseñanza de la computación en niños y jóvenes. Esta última iniciativa, que es desarrollada por la empresa Microsoft, tiene una versión Latino Americana denominada *Yo Puedo Programar*<sup>6</sup> que involucra a más de 100 entidades asociadas y ha alcanzado a más de 3 millones de jóvenes de la región. Cabe señalar que esta empresa, junto con otros gigantes del rubro, han sido activos promotores de este movimiento aportando ideas, recursos digitales y fondos semillas para ampliar su difusión e impacto.

En el caso de Microsoft, cuyo interés en estos temas es propio de su quehacer, su trabajo en este campo tiene dos aristas: por un lado, su trabajo filantrópico con la sociedad civil, en particular, el orientado a ampliar las oportunidades de los jóvenes usando tecnología para lograr oportunidades concretas de empleo, emprendimiento y reinserción educativa; y por otro, su trabajo con el sector educativo de los países, especialmente los Ministerios de Educación y las políticas de TIC para escuelas. Fue justamente en la intersección de estas dos miradas que Microsoft América Latina identificó la necesidad de reunir antecedentes sobre los desafíos que se están enfrentando en la región en este ámbito para plasmarlos en un documento que permitiera orientar

avances futuros. Para esto, organizó una serie de encuentros con actores públicos y privados que están trabajando en este campo con el fin de recoger inquietudes y perspectivas que pudieran alimentar la elaboración de este documento<sup>7</sup>

Latinoamérica mira con atención el intenso movimiento en favor de la enseñanza de la computación al mismo tiempo que busca resolver sus problemas sociales y educativos más apremiantes, como son la desigualdad, los serios déficits en la calidad de la educación y la falta de oportunidades de buena parte de la juventud que no asiste a la escuela; así como los enormes rezagos de acceso y alfabetización digital aún existentes (Sunkel & Trucco, 2012). Se trata, sin duda, de un contexto diferente al de los países más desarrollados que lideran estos cambios, lo que obliga a pensar y modular la forma en que estas innovaciones aterrizan en nuestras realidades. Parece oportuno, por tanto, alimentar una discusión informada que ayude a tomar buenas decisiones y a responder las múltiples interrogantes que pudieran emerger de su implementación. En este sentido, la experiencia de las iniciativas que participaron en los encuentros organizados por Microsoft constituyen una buena base de aprendizaje para el futuro (para un listado de los expertos participantes en estos encuentros, ver Anexo).

El presente documento busca presentar los principales elementos del movimiento a favor de la enseñanza de computación, tanto en sus componentes públicas como privadas, tanto en la educación formal como no-formal, de manera de ofrecer una descripción básica de este territorio y sus principales desafíos.

<sup>2</sup>Ver <http://makerfaire.com/>

<sup>3</sup>Raspberry-Pi es en realidad un pequeño computador que puede ser programado para ejecutar las más diversas tareas. Ver [www.raspberrypi.org](http://www.raspberrypi.org)

<sup>4</sup>[www.code.org](http://www.code.org)

<sup>5</sup>[www.microsoft.com/about/philanthropies/youthspark](http://www.microsoft.com/about/philanthropies/youthspark)

<sup>6</sup>[www.yopuedoprogramar.com/](http://www.yopuedoprogramar.com/)

<sup>7</sup>Microsoft realizó tres encuentros –en Argentina, Chile y México– en los meses de Marzo y Abril de 2016, en los que participaron decenas de instituciones públicas y organizaciones de la sociedad civil vinculadas con la educación digital. Ver sección 4.

El documento está dirigido prioritariamente a decisores de política públicas que estén interesados en entender este tema, a los que se espera orientar en las decisiones que deban tomar en este ámbito. Para esto, el documento entrega antecedentes sobre lo que se está haciendo en otros países, por qué lo están haciendo y cómo lo están haciendo. Asimismo, el documento está dirigido a quienes dirigen y trabajan en organizaciones privadas como empresas, fundaciones u ONG's que están desarrollando iniciativas de educación no-formal, a quienes se espera enriquecer su visión y potenciar su trabajo. Para ellos, el documento ofrece una perspectiva regional de los actores e iniciativas con quienes poder conectarse y un levantamiento de los desafíos comunes.

El documento se organiza en cinco secciones principales. En la sección 2 se intentará clarificar de qué estamos hablando exactamente: qué se propone enseñar (qué son las ciencias de la computación y el pensamiento computacional); cómo se propone hacerlo (con qué herramientas y pedagogías); y por qué (cuales son los racionales que justifican su ingreso a la educación de los niños y jóvenes). Por último, se discutirá la evidencia disponible sobre los efectos de este tipo de políticas e iniciativas. En la sección 3 se presentarán las principales experiencias de incorporación de la enseñanza de computación en la educación escolar, así como las preguntas que deben abordarse a la hora de adoptar este tipo de políticas. En la sección 4 se presentarán las principales iniciativas impulsadas desde la sociedad civil para promover la computación, así como los principales desafíos que enfrentan. Por último, en la sección 5 se presentarán las principales conclusiones y recomendaciones que emergen de los antecedentes presentados en las secciones precedentes.

## 2. Antecedentes

En esta sección se intentará aclarar los principales conceptos sobre la enseñanza de la computación a niños y jóvenes. Por simplicidad, en este documento se usa el término computación para referirse a la enseñanza de *ciencias de la computación, pensamiento computacional y/o programación*. Se trata de una terminología relativamente nueva para quienes no pertenecen a esta disciplina, pero incluso entre los expertos no hay todavía completo consenso sobre el alcance de estos conceptos cuando se usan en el contexto de la educación escolar y la formación de jóvenes.

### ¿De qué estamos hablando?

En primer lugar, cabe ubicar esta discusión dentro del marco mayor de la utilización de las TIC en educación. Normalmente, las políticas buscan aprovechar los potenciales educativos de las nuevas tecnologías digitales para mejorar y transformar los procesos de enseñanza y aprendizaje que ocurren en el contexto escolar, para lo cual promueven el uso didáctico de aplicaciones generales, software educativo e Internet en forma transversal en todas las disciplinas del currículum. El sentido y propósitos últimos de estos usos pueden variar entre países, regiones y escuelas, pero, en general, es posible afirmar que todos los profesores en ejercicio a comienzos de este siglo XXI están siendo empujados a adaptar su docencia y utilizar de una u otra forma las TIC. Este es un movimiento muy gradual de enorme envergadura y

relevancia cuyo resultado final nadie sabe con certeza, y sobre el cual hay consenso que es ineludible.

La idea de promover la enseñanza de la computación no pretende reemplazar ni afectar este proceso mayor de adaptación de la educación a los nuevos contextos tecnológicos; en cambio, considera que esta adaptación estará incompleta mientras no se consideren estos nuevos saberes como parte de las competencias básicas que todo estudiante debe adquirir durante su experiencia educativa. El uso de las aplicaciones de oficina, software educativo e Internet que los estudiantes aprenden mientras utilizan la tecnología en su vida diaria o durante sus cursos escolares, serían competencias necesarias pero insuficientes para desempeñarse como ciudadanos plenos en el siglo XXI. Tampoco sería suficiente lo que muchos aprenden en cursos especializados de informática durante la secundaria, como buscar y organizar información, trabajar con bases de datos sencillas y tener nociones básicas de cómo programar páginas web. Todos estos aprendizajes, usualmente referidos como *alfabetización digital* o *fluidez tecnológica*, no estarían a la altura de lo que se necesita en el nuevo milenio. Hoy se requeriría el estudio sistemático de la computación, materia que, por tanto, debería ponerse al centro de los cursos de informática y/o en el seno de las disciplinas escolares asociadas con STEM (como matemáticas o física)<sup>8</sup>, así como también en actividades extraescolares.

<sup>8</sup>STEM refiere a la enseñanza de Ciencias, Tecnología, Ingeniería y Matemáticas.

El movimiento en favor de la computación es, entonces, una demanda por sumar al currículum escolar y la formación de los jóvenes los conocimientos y habilidades de una nueva disciplina, así como son actualmente matemática o historia. Pero ¿Qué es exactamente lo que se propone agregar? ¿Cuáles son estos nuevos conocimientos y habilidades? ¿Cómo se deberían enseñar? ¿Por qué deberían ocupar un lugar tan importante en la educación de las nuevas generaciones? ¿Qué evidencia tenemos de su relevancia? Si bien estamos en un estadio muy inicial de la entrada de esta disciplina en los sistemas educativos como para tener buenas respuestas a todas estas preguntas, en lo que sigue de esta sección se abordarán de manera exploratoria estas inquietudes.

### ¿Qué se propone enseñar?

La Ciencia de la Computación es una disciplina académica que estudia lo que puede ser realizado por un computador y cómo realizarlo (Wing, 2006); estudia los computadores y sus algoritmos, incluyendo sus principios, el diseño del hardware y software, sus aplicaciones prácticas y su impacto en la sociedad (CSTA, 2010). Los algoritmos establecen los procedimientos paso a paso que deben seguir los computadores para poder adquirir, representar, estructurar, procesar y comunicar datos, así como hacer cálculos; todas acciones que están a la base de las aplicaciones computacionales que se usan cotidianamente. Dentro del campo de la ciencia de la computación se estudian temas tales como bases de datos, arquitectura y redes de computadoras, interacción humano-computador, sistemas operativos, lenguajes de programación, seguridad informática, redes de datos, sistemas distribuidos, sistemas expertos, inteligencia artificial y robótica, entre otros.

Los departamentos de *Ciencias de la Computación* en las universidades preparan a los profesionales de la informática quienes son responsables de diseñar, desarrollar y mantener los sistemas informáticos que nos rodean. Estos profesionales analizan problemas y procesos humanos, estructurando y modelando aquellas dimensiones que pueden ser abordadas por computadores, y diseñan e implementan sistemas de hardware y software que materializan soluciones informáticas. Esta disciplina está en el corazón de la industria tecnológica que en las últimas décadas han transformado el mundo desarrollando sistemas informáticos que, por ejemplo, sostienen los mercados globales, decodifican el genoma humano, organizan el tráfico aéreo y operan las licitaciones gubernamentales; así como también nos permite hacer transferencias bancarias desde nuestros hogares; pedir al auto que nos guíe por el camino más rápido al trabajo; mantener contacto con compañeros de colegio que no vemos hace años; y ordenar a nuestro teléfono móvil que nos consiga el taxi más cercano.

El estudio de los fundamentos de esta ciencia en el contexto de la educación de niños y jóvenes se plantea principalmente orientado al desarrollo del *pensamiento computacional* (razón por la cual muchas veces ambos conceptos se utilizan en forma intercambiable). Si bien este último no es un concepto nuevo<sup>9</sup>, todavía no hay total acuerdo entre académicos y educadores sobre su alcance y definición precisa<sup>10</sup>, sin perjuicio de lo cual, se puede ilustrar su significado a partir de la definición que hace la propia Wing: el *pensamiento computacional* es un proceso mental que permite formular problemas de tal forma que sus soluciones puedan ser realizadas con computadores (Wing, 2010). Se trata de la manera de pensar de

<sup>9</sup>El concepto de Pensamiento Computacional fue utilizado por primera vez por Papert a comienzos de los 80's.

<sup>10</sup>Para diferentes perspectivas, ver por ejemplo, Selby & Wollard, 2013; Grover & Pea, 2013.



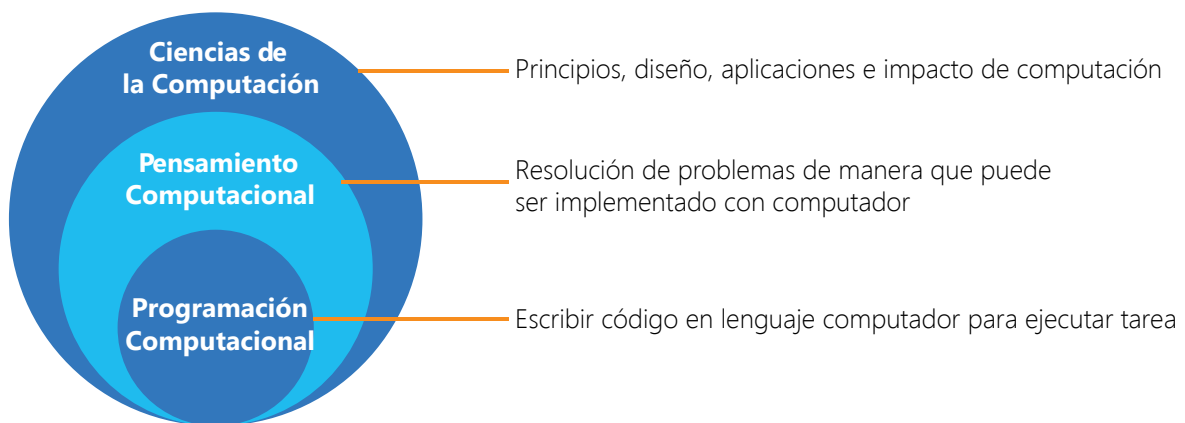
los profesionales de la informática, lo que permitiría entender los usos y limitaciones de las soluciones computacionales y aplicar métodos computacionales a la solución de problemas de diversa índole, sea esta cotidiana, profesional o científica. De esta forma, el *pensamiento computacional* permitiría a los jóvenes hacer una mejor conceptualización, análisis y solución de problemas complejos, por medio de la selección y aplicación de estrategias y herramientas propias de las ciencias de la computación. *El pensamiento computacional* implica pensar en términos de abstracción y generalización; implica modelar y descomponer los problemas en sub-problemas; analizar procesos y datos, así como crear artefactos digitales virtuales y reales; entre otros (CSTA, 2011; Selby & Woollard, 2013).

A veces el término *pensamiento computacional* se usa como sinónimo de programación, probablemente porque ambos refieren finalmente a las capacidades requeridas para resolver un problema definiendo un

conjunto de instrucciones para que un computador ejecute una tarea específica. Sin embargo, al hablar de *ciencias de la computación* y de *pensamiento computacional* lo que se busca es ampliar la enseñanza más allá de la *programación*, incluyendo las bases conceptuales de la computación la que, por cierto, tienen como uno de sus principales resultados la capacidad de programar, esto es, diseñar algoritmos y definir el código que lo lleva a la práctica en lenguaje de computador.

Aunque los tres conceptos nombrados –*ciencias de la computación*, *pensamiento computacional* y *programación*– no son exactamente lo mismo<sup>11</sup> (ver figura nº1), los tres refieren a dotar a los jóvenes de una comprensión profunda de los fundamentos del mundo digital y darle herramientas para ser agentes creativos del mismo y no sólo usuarios y consumidores de tecnología, como se promueve actualmente en la educación.

**Figura N°1: Relación entre conceptos principales de la computación**



Para ilustrar en qué se traducen estas definiciones al momento de establecer un currículum que las lleva a la práctica, el Cuadro N°1 resume los objetivos de aprendizaje descritos para el nuevo curso de informática de Inglaterra y ejemplifica el tipo de contenidos que éste implica.

<sup>11</sup>Cabe señalar, que muchos utilizan estos tres términos en forma intercambiable sin hacer distinciones, generando alguna confusión en el debate sobre este tema.

## Cuadro N°1: Objetivos Ciencias de la Computación de Inglaterra

El Plan de Estudios Nacional de Ciencias de la Computación de Inglaterra tiene como objetivo que los alumnos estén en condiciones de:

- a. Entender y aplicar los principios y conceptos fundamentales de las Ciencias de la Computación, tales como la abstracción, la lógica, los algoritmos y la representación de los datos.
- b. Analizar problemas en términos computacionales, contar con experiencias prácticas de escritura de programas informáticos con el fin de resolver distintas tipologías de problemas.
- c. Evaluar y aplicar tecnologías de la información, incluyendo tecnologías nuevas o desconocidas.
- d. Actuar como usuarios responsables, competentes, seguros y creativos de las tecnologías de información y comunicación.

En base a estos objetivos, el currículum inglés organiza una trayectoria de objetivos más específicos a lo largo de los cuatro niveles de la educación escolar. Por ejemplo, en el primer nivel, los niños debieran ser capaces de sacar los errores de programas sencillos, comprender algoritmos y la manera en que éstos se traducen en programas en los dispositivos digitales, así como comprender que los programas se ejecutan siguiendo secuencias de instrucciones precisas. En el tercer nivel, en cambio, los jóvenes ya deben saber modelar algunos problemas del mundo real; comprender diversos algoritmos claves (por ejemplo, los de clasificación y búsqueda); utilizar lenguajes simples y complejos para hacer programas que utilizan estructuras de datos (por ejemplo, listas, tablas o matrices), procedimientos y funciones; comprender los componentes del hardware y del software que constituyen los sistemas informáticos, cómo se comunican entre sí y con otros sistemas; comprender cómo se pueden representar distintos tipos de información como datos binarios (texto, sonido, imágenes); y crear aplicaciones en dispositivos digitales para un público determinado, con especial atención la confiabilidad, diseño y usabilidad; entre otros.

(Cobo, 2014)

### ¿Cómo se enseña?

Existe un emergente cuerpo de literatura sobre el desarrollo del *pensamiento computacional* en la educación, el que se ha nutrido de la experiencia de la enseñanza de programación en educación secundaria y post-secundaria desde los años 70 y 80. En esa época, posiblemente debido a la complejidad de los lenguajes de programación, hubo mayor énfasis en la codificación de instrucciones que en el diseño de algoritmos, lo que hacía parecer esta tarea como algo muy laborioso e intrincado. La experiencia mostró que no era fácil enseñar a programar ni aprender los conceptos más abstractos de la computación. Con el

tiempo, sin embargo, los métodos han madurado y las herramientas se han abaratado y se han hecho más simples de usar.

Sin perjuicio de estos avances, es aún muy pronto como para contar con un cuerpo sólido y sistematizado de conocimiento sobre la enseñanza y el aprendizaje de esta disciplina en los niños. Hay todavía un importante desafío en torno a quién puede enseñar computación a los jóvenes y cómo debiera hacerlo.

La programación es el principal territorio donde se aplica y desarrolla el *pensamiento computacional*

y es justamente en este ámbito donde hay más experiencia. Programar computadores puede ser algo muy estimulante, pero en ningún caso es algo simple ni que se aprenda rápido. Desde los años 70 se ha intentado facilitar la enseñanza de la programación desarrollando ambientes y herramientas –aplicaciones y lenguajes para programar– especialmente orientadas a este propósito, pero la selección de estas herramientas, así como de las metodologías didácticas en las cuales éstas se insertan, sigue siendo un campo lleno de decisiones complejas, en las que se conjugan múltiples factores, tales como las características y habilidades de los aprendices, los objetivos de aprendizaje y los recursos disponibles, entre otros.

Es una práctica común que en la enseñanza de la programación se utilicen metodologías relacionadas con el aprendizaje basado en proyectos, en problemas o en desafíos –los cuales están siendo también crecientemente utilizados en el aprendizaje de las ciencias en general– en los cuales los jóvenes puedan aplicar los conocimientos adquiridos. Idealmente estos problemas deben ser reales y auténticos, y el foco del trabajo debe estar centrado en el procesamiento de información y en incentivar la reflexión por parte de los alumnos (Lye & Koh, 2014). En el marco de las restricciones temporales de los currículum y horarios escolares es habitual también hacer que los estudiantes sigan instrucciones paso a paso para poder avanzar en las primeras etapas de creación de sus proyectos, pero estas estrategias atentan contra la posibilidad de estimular la creatividad en los niños, la que es mejor lograda cuando se deja que busquen sus propias estrategias para crear o adaptar proyectos para resolver un problema.

Los problemas que se abordan cuando se aprende a programar se plantean en contextos digitales concretos. Por ejemplo, el diseño de juegos es un tipo de problema bastante utilizado por ser particularmente atractivo para los niños y jóvenes. También resulta llamativo la construcción de aplicaciones para dispositivos móviles, especialmente teléfonos celulares. Asimismo, resulta estimulante la construcción de artefactos tangibles a través de tarjetas programables o robots<sup>12</sup>, cuya naturaleza física ofrece una experiencia inspiradora (Garneli et al, 2015).

Los lenguajes de programación gráficos como Scratch, Kodu o SNAP<sup>13</sup>, que permiten programar moviendo bloques de instrucciones en la pantalla en vez de escribir el código, son los más utilizados en la enseñanza de la programación, especialmente en las etapas iniciales, pues son fáciles de utilizar y permiten enfocar la atención en el diseño y creación de las soluciones, evitando los detalles de sintaxis propios de los lenguajes más sofisticados como Python o Java<sup>14</sup>. Si bien los lenguajes gráficos son más simples, han mostrado ser suficientes para desarrollar los conceptos más importantes del pensamiento computacional (Garneli et al, 2015).

Existe también un creciente número de recursos virtuales disponibles en Internet para apoyar la enseñanza y para que los jóvenes aprendan a programar en forma autónoma. Ejemplos de estos recursos son los desarrollados por iniciativas tales como la Hora del Código o YouthSpark mencionadas anteriormente. El enorme alcance que han tenido estos recursos confirma su utilidad y facilidad de uso.

<sup>12</sup>Existen diversas tarjetas programables utilizadas en las escuelas, siendo comunes Arduino y GogoBoards. Kits de robótica también hay varios, siendo el más popular LegoMindstorm.

<sup>13</sup>Hay muchos lenguajes gráficos, siendo quizá los más utilizados Scratch (un descendiente de LOGO), Alice, GameMaker y Kodu, entre otros (Grover & Pea, 2013). Recientemente también se utilizan SNAP (<http://snap.berkeley.edu>) y Blockly (<https://developers.google.com/blockly/>) que han seguido enriqueciendo y simplificando los ambientes de programación, permitiendo incluso el trabajo con niños pequeños, pre-lectores.

<sup>14</sup>Lenguajes más profesionales, como Python, Java o Scheme, entre muchas otras opciones, pueden ser utilizados con estudiantes más avanzados gracias a que les permiten abordar problemas más auténticamente reales.

Según Grover y Pea (2013) los ambientes y herramientas que mejor fomentan el pensamiento computacional cuentan con un “piso bajo” y un “techo alto”, es decir, permiten a los usuarios utilizarlo en un nivel básico, al mismo tiempo que permiten hacer proyectos complejos; cuentan con andamiajes adecuados para apoyar al usuario; permiten la transferencia a otros contextos; son equitativos y sistemáticos.

### ¿Por qué enseñar computación a niños y jóvenes?

Se esgrimen distintos argumentos para justificar la necesidad de educar a los jóvenes en las *ciencias de la computación, el pensamiento computacional y/o la programación*. Los más comunes refieren a los beneficios que tendría para las personas tener una mejor y más profunda comprensión del mundo digital en que estamos inmersos, así como desarrollar la habilidad de resolver problemas aprovechando las potencialidades de las tecnologías digitales. Por ejemplo, parecería razonable que toda persona tuviera los conocimientos que le permitieran tener nociones mínimas del funcionamiento interno de computadores e Internet, de manera de comprender sus posibles alcances y limitaciones.

Estos conocimientos nos permitirían hacer sentido e influir en el mundo digitalizado, computarizado y programable que nos rodea. Una comprensión de los fundamentos de la computación permitiría que las personas innovaran en los diversos ámbitos en que les tocara desempeñarse a partir del diseño de nuevas soluciones que mejoraran la calidad de vida de todos. Sin estos nuevos saberes, estaría en riesgo nuestra participación plena en la sociedad y economía del siglo XXI.

Desde esta perspectiva, el *pensamiento computacional* es visto como una habilidad básica que fomenta estrategias de resolución de problemas que son útiles cualquiera sea el ámbito de estudio o trabajo, más allá del propiamente tecnológico, como sugiere la creciente intersección de esta disciplina con otras

ciencias e ingenierías. Por ejemplo, hoy en día las simulaciones computacionales son esenciales en el descubrimiento de las reglas fundamentales que gobiernan una gran gama de sistemas, desde cómo las hormigas recolectan su alimento, hasta cómo se comportan los mercados globales, pasando por la detección y tratamiento de enfermedades como el cáncer, lo que ha sido posible gracias al desarrollo de métodos computacionales que permiten simular y comprender las mutaciones genéticas involucradas (CSTA, 2011).

Otro argumento importante refiere a las ventajas que tendría para los países atraer a más jóvenes hacia profesiones tecnológicas. Este argumento sostiene que se requiere acercar a los jóvenes a estos temas durante la experiencia escolar –y no después– de manera de aumentar el interés en estudiar las carreras del ámbito tecnológico requeridas por todas las organizaciones e industrias, en especial las del sector que desarrolla las tecnologías de la información. Es de particular preocupación el hecho de que, al mismo tiempo que aumenta la relevancia de este sector tecnológico en el impulso por la innovación y el crecimiento económico, actualmente se esté debilitando la disponibilidad de profesionales con especialización en este ámbito. De consolidarse esta tendencia, en el largo plazo los países podrían quedar fuera del juego tecnológico, sin posibilidad de producir ni tener control sobre los sistemas que se verían obligados a consumir. Por eso parece urgente revertir esta tendencia haciendo que la educación escolar ponga en contacto a los estudiantes con los fundamentos de la revolución tecnológica.

En Latinoamérica, donde muchos países comparten las preocupaciones antes señaladas, hay consideraciones adicionales. En particular, esta región presenta un alto riesgo social por el porcentaje de jóvenes que abandonan la escuela secundaria para trabajar en actividades de baja calificación o quedar expuestos a la cesantía, las drogas y la delincuencia. Para ellos, tener habilidades tecnológicas lo más avanzadas posible,

puede ser un factor que aumente sus oportunidades de empleo o, mejor, evite la deserción escolar, en un contexto de creciente demanda de habilidades tecnológicas en el mercado laboral. Asimismo, puesto que la computación despierta interés en los jóvenes, parece ser un buen vehículo para atraerlos hacia la tecnología y las ciencias. Incluso para los jóvenes de los sectores más vulnerables, la tecnología suele ser atractiva y motivante, y es un medio en el cual también se desenvuelven, aunque sea mínimamente. Lamentablemente, en nuestra región hay todavía muchos países donde el acceso a la tecnología y sus habilidades más básicas está lejos de sectores importantes de la población. En estos contextos, los desafíos son mayores aún, pero son asimismo enormes las ventajas de invertir en iniciativas que aprovechen el interés de los jóvenes por la tecnología para integrarlos en la educación y el empleo. De aquí el valor del trabajo de instituciones que trabajan con los sectores menos favorecidos de este continente entregando acceso, habilidades digitales básicas o nociones de programación, algunas de las cuales serán comentadas en la sección 4.

En suma, la enseñanza de la computación porta hoy la promesa de una educación sintonizada con las necesidades del nuevo siglo, permitiendo que las nuevas generaciones adquieran competencias no solo para comprender y explotar el mundo digital, sino también para ser creativos, innovadores y emprendedores; habilidades consideradas motores del crecimiento y desarrollo de los países. Es quizá demasiado temprano para saber si la enseñanza de la computación podrá responder a este gran anhelo, pero al mismo tiempo parece aún más difícil sostener que su incorporación en la agenda educativa pudiera perjudicarlo.

### **¿Qué evidencia existe?**

Las promesas de la enseñanza de computación se basan en la idea de que esta disciplina desarrolla habilidades cognitivas de alto nivel, como el

razonamiento lógico y la resolución de problemas en el contexto de soluciones computacionales. Como se ha señalado, estas habilidades y conocimientos serían valiosas en sí mismas; y transferibles a los más diversos ámbitos de la vida.

Hay intentos de registrar el impacto de la enseñanza de programación en el desarrollo cognitivo de los niños desde los años 70's, muchos de los cuales han puesto foco justamente en la capacidad de resolución de problemas y su transferencia a otros contextos. Sin embargo, la evidencia recogida es todavía insuficiente para confirmar dicho impacto.

Cabe señalar que la enseñanza de la computación, así como casi todas las políticas e iniciativas de tecnología en educación, ha provisto abundante evidencia circunstancial sobre su impacto en la vida de las personas, principalmente a través del registro de testimonios de beneficiarios; pero a la hora de verificar los resultados de manera sistemática o a mayor escala, la evidencia tiende a ser escasa y controvertida. Mientras hay estudios que confirman los buenos resultados, otros los desmienten y otros afirman que no hay suficiente rigor en ninguno de los anteriores o que en realidad se requiere evaluar otra cosa. Finalmente, no es posible afirmar con certeza que se hayan logrado los impactos prometidos.

Cabe ilustrar esta situación con un ejemplo. A comienzos de la década de los 80's estaba en boga la teoría constructorista de Papert (1980) que planteaba una enseñanza centrada en el alumno y en la construcción de conocimiento a través del diseño y construcción de artefactos que fuesen significativos para quienes los diseñan y construyen. Se promocionaba una educación en torno al desarrollo de proyectos que involucra a toda la persona: su intelecto, sus emociones y su motricidad, así como también sus aficiones y gustos. Papert propuso que la programación computacional era un escenario propicio para poner en práctica esta idea y desarrolló el lenguaje de programación LOGO con

este fin. En este contexto, se generó un gran interés en investigar los beneficios de enseñar programación en el desarrollo cognitivo de los niños. Por ejemplo, Clement y Gullo (1984) diseñaron un experimento para medir el impacto de la programación en el estilo y desarrollo cognitivo de los estudiantes y llegaron a la conclusión de que existen diferencias significativas en el estilo cognitivo (los niños son más reflexivos), en meta-cognición y en pensamiento divergente, pero no en desarrollo cognitivo general. Se suponía que el diseño de Clement y Gullo superaba las limitaciones que ellos habían encontrado en estudios previos, que juzgaban como poco rigurosos: estudios que indicaban algunos efectos sobre la programación en la resolución de problemas (Billings, 1983; Milner, 1973; Soloway, Lochhead & Clement, 1983; Statz, 1974); así como otros que indican que la transferencia de las habilidades desarrolladas mediante la programación son fuertemente dependientes del contexto, por lo que es difícil obtener resultados concluyentes (Pea, Hawkings & Sheingold, 1983). Sin embargo, con posterioridad la mayor parte de los estudios de esa época han sido cuestionados por presentar serios problemas de validez al no ser rigurosamente diseñados (ver, por ejemplo, crítica de Milojkovic, 1984); y, en particular, los test utilizados por Clement y Gullo han sido recurrentemente criticados.

Otro ejemplo más reciente es el caso de la robótica, que ha sido propuesta como escenario de aprendizaje que potencia el pensamiento computacional y desarrolla tanto habilidades transversales como conocimientos específicos de distintas disciplinas. Benitti (2012) hace una recopilación sobre los efectos de la incorporación de la robótica en el aprendizaje, pero al igual que en las otras recopilaciones citadas, esta destaca que existen pocas investigaciones rigurosamente diseñadas para obtener resultados concluyentes.

Actualmente el interés de la investigación sobre el impacto de la enseñanza de la computación está más centrado en el pensamiento computacional. Sin

embargo, aún no se cuenta con evidencia concluyente del efecto de la programación en el desarrollo de *pensamiento computacional* ni de éste sobre otras capacidades cognitivas de los niños. Grover y Pea (2013), por ejemplo, señalan que existe una gran deuda con la investigación empírica desde las ciencias del aprendizaje, referida a determinar el impacto en aspectos socioculturales, emocionales y cognitivos, en las posibilidades de utilizar el pensamiento computacional integrado en otras áreas y en cómo mejorar el aprendizaje mismo de la programación.

Existen otras promesas asociadas al estudio de la computación que aún no han sido confirmadas en forma sistemática. Por ejemplo, algunos sostienen que es un vehículo privilegiado para desarrollar la *creatividad*, ámbito que la escuela tradicional tiende a ahogar dentro de sus estructuras y exigencias, los cuales procuran que los estudiantes elaboren respuestas correctas únicas y ofreciendo pocos grados de libertad para aproximaciones disruptivas a los problemas e interrogantes. En el ejercicio de la programación, en cambio, las soluciones correctas son múltiples, diversas, más o menos elegantes, siempre personales, y los niños tienen espacio para explorar y crear soluciones a problemas que les resulten interesantes, siendo el criterio de realidad –¡que funcione!– su único marco regulatorio. Si bien existen numerosos testimonios que reflejan el desarrollo de la creatividad en niños y jóvenes, existe escasa evidencia rigurosamente medida.

Otro atractivo del estudio de la computación sería su capacidad de inclusión, especialmente mujeres y personas con necesidades educativas especiales; y, por esta vía, ampliar las oportunidades de empleo y progreso de estos grupos que, por diversas razones, son objeto de marginación, especialmente en Latinoamérica. Los testimonios muestran que todos los niños, de diferente género, capacidad y origen social, pueden programar y mostrar altos grados de creatividad, presentando resultados sorprendentes y originales. Esto ha sido posible incluso en contextos

de alta vulnerabilidad en que se ha trabajado con todo tipo de niños y jóvenes, con excelentes resultados. En el caso de las niñas, los testimonios muestran la posibilidad de acercarlas por esta vía al estudio de tópicos científicos y tecnológicos, tanto como parte de su formación escolar como en preparación para el mundo laboral. Lo mismo ha sido posible observar en el caso de niños y jóvenes con necesidades educativas especiales (discapacidades), lo que representa una nueva y esperanzadora oportunidad de inclusión.

La enseñanza de la programación, al igual que otros usos educativos de la tecnología en las aulas y fuera de ellas, también porta la promesa de interesar y motivar de forma muy particular a niños y jóvenes. Prueba de ello son, por ejemplo, los millones de participantes en las actividades online de iniciativas como *la Hora del Código* o *YouthSpark*, incluso en nuestra región. Los testimonios en este sentido son múltiples, tanto dentro como fuera de la escuela. Para muchos, esta motivación ofrece una oportunidad de refrescar partes del quehacer de los estudiantes en las jornadas escolares y de los jóvenes en sus barrios, en especial en aquellos más vulnerables, como alternativa al ocio, la droga o la delincuencia.

Finalmente, cabe señalar que, desde los comienzos de la revolución digital, el principal motor de la integración de la tecnología en la educación ha sido una visión clara sobre el rol estratégico que jugarán las tecnologías en la construcción del futuro de nuestras sociedades. La historia reciente parece confirmar estas predicciones, así como la importancia de una educación tecnológica que fortalezca nuestra capacidad de aprovechar esta revolución para nuestro beneficio. A ratos esta convicción puede parecer suficiente argumento para sostener la enseñanza de la computación a niños y jóvenes, y no se requeriría, por tanto, de mayor evidencia empírica. Sin embargo, es siempre conveniente contar con evidencia sobre los resultados de estas políticas e iniciativas, a fin de introducir mejoras, ajustar el rumbo y convencer a los incrédulos. Lamentablemente, como se ha

señalado anteriormente, aún no es posible contar con evidencia consistente a este respecto, pero se espera que la experiencia e investigación actualmente en curso pueda dar luces sobre este importante campo educativo (Hubwieser et al, 2015).

## 3. Políticas públicas

No son pocos los países que están dando mayor énfasis a la enseñanza de las ciencias de la computación, pensamiento computacional o programación en su educación escolar. En esta sección se revisarán alguna de estas experiencias y se perfilarán las principales preguntas a abordar a la hora de tomar decisiones en este campo.

### 3.1. Experiencias países

Para ilustrar la tendencia de las políticas públicas en relación a la enseñanza de la computación, se han seleccionado algunos ejemplos de países que destacan por su tradición y experiencia en este campo, así como su influencia en nuestra región. A nivel global, se mostrará la situación de Europa, dentro de la cual se destacará la experiencia de Inglaterra; asimismo se comentará el avance de Estados Unidos, Israel y Nueva Zelanda. Dentro de Latinoamérica, se abordará la experiencia señera de Costa Rica y los intentos de Uruguay y Chile por acercar la programación a la población escolar. Cabe aclarar que esta selección de experiencias debe entenderse solo como ejemplos para ilustrar las tendencias; y en ningún caso que estos países sean los únicos que están haciendo algo en esta materia<sup>15</sup>.

#### **Europa**

Un estudio de European Schoolnet publicado en 2014 sobre la enseñanza de programación en las escuelas de ese continente, muestra que este es un foco de interés en la mayoría de sus sistemas escolares (EUN, 2014). El estudio comprendió a 20 países, incluyendo Bélgica, Bulgaria, Chipre, República Checa, Dinamarca, Estonia, España, Finlandia, Francia, Grecia, Holanda, Inglaterra, Irlanda, Italia, Lituania, Luxemburgo, Noruega, Portugal, Polonia y Turquía.

El estudio revela que 12 de los 20 países encuestados ya tienen la programación como parte de sus currículos (Bulgaria, Chipre, República Checa, Dinamarca, Estonia, Grecia, Inglaterra, Irlanda, Italia, Lituania, Polonia, Portugal) y que otros 7 planean integrarlo. La excepción es Noruega que, sin embargo, tiene una asignatura llamada "Tecnología en la Práctica", en la que se permite a los profesores enseñar programación en forma individual si es que lo desean.

La mayoría de estos países, tengan o no ya implementada la programación en sus currículos, consideran que es muy importante desarrollar

<sup>15</sup>Para revisión de otros países, ver CSTA (2005), CAS (2011) y Hubweiser et al (2015).



esta competencia porque, además de tener valor en sí misma, contribuiría con las habilidades de pensamiento crítico y resolución de problemas de los estudiantes. Asimismo, la mitad de los países considera que desarrollar esta competencia<sup>16</sup> es vital para atraer más estudiantes hacia carreras de computación y promover la empleabilidad en el sector tecnológico.

Casi todos los países que ya han integrado la programación en sus sistemas escolares lo hacen a través de un curso especial de informática o tecnología de la información, aunque hay variación de la porción de este currículum que ocupa el desarrollo de esta disciplina. La mayor parte de estos países consideran la programación a nivel de secundaria y muy pocos desde primaria (4 de los 12). En general, la obligatoriedad de esta materia está reservada para los niveles más altos de escolaridad siendo algo opcional en el resto de los niveles, especialmente en primaria. El único país que ha establecido la enseñanza de computación como obligatoria desde primaria es Inglaterra, país que, por esta y otras razones, se verá con más atención a continuación.

### **Inglaterra**

El caso de cómo Inglaterra llegó a integrar la computación en su currículo de K-12 suele ser tomado por muchos como un ejemplo a seguir, tanto por la radicalidad del cambio, como por la articulación de distintos actores de la sociedad civil –académicos e industrias del ámbito tecnológico– con las escuelas, sus docentes y las políticas impulsadas por las autoridades (Brown et al., 2013).

Inglaterra cuenta con un curso de informática desde los años 80's cuando se enseñaba a programar en BASIC, pero en la década siguiente su currículum se orientó principalmente hacia el uso de las herramientas de Office. Sin embargo, desde mediados de los 90's algunos comenzaron a plantear que ese enfoque no estaba dando los resultados esperados (Stevenson Committee, 1997). Una década más tarde, la situación seguía sin cambios, aún cuando voces desde la educación, la industria y el gobierno mostraban creciente insatisfacción frente a la forma en que se estaba abordando el tema (Anderiesz, 2014).

En 2007, Simon Peyton-Jones, investigador de Microsoft Research, comenzó a reunirse con docentes de informática de las escuelas con los que arribó al diagnóstico de que el currículum de dicha materia no estaba logrando instalar las habilidades digitales necesarias en los estudiantes; y, al revés, parecía alejarlos de la computación. De este grupo inicial nació la asociación Computing at Schools (CAS), que a poco andar y con la adhesión de un centenar de miembros, lanzó en 2009 el documento *Computing at School: The state of the Nation*, en el que se explicaban detalladamente los problemas del curso de informática y demandaba un cambio profundo en su currículum abordando la enseñanza de ciencias de la computación.

Cuando asumió un nuevo gobierno en el año 2010, y se comenzó a abrir la puerta para revisar el trabajo con TIC en las escuelas, CAS y otras organizaciones ejercieron una influencia determinante para que se considerara poner la enseñanza de ciencias de la computación en el centro del nuevo curso de

<sup>16</sup>No todos los países europeos se refieren de la misma forma a las competencias por desarrollar: mientras la mayoría la denomina como programación de computadores, algunos las refieren como pensamiento computacional o pensamiento algorítmico.

informática. Por ejemplo, en 2011 la CAS en conjunto con la *British Computer Society* (BCS), lanzaron las bases de una propuesta curricular para materializar el cambio propuesto; y a comienzos de 2012 la *British Royal Society* (BRS) publicó otro influyente reporte titulado *Shutdown or Restart?: The way forward for computing in UK schools*. Asimismo, importantes empresas de tecnología –como Google, Microsoft y BritishTelecom– comenzaron a promover la formación de docentes, agente clave de este tipo de transformaciones. Finalmente, el Departamento de Educación estableció que el nuevo currículum, cuyos objetivos específicos se plasmaron en nuevos programas de estudios<sup>17</sup>, debía implementarse en todas las escuelas inglesas a partir de 2014 (Cobo, 2014).

El objetivo principal del nuevo currículum inglés es equipar a los alumnos para usar el pensamiento computacional y la creatividad para entender y cambiar el mundo; así como utilizar la tecnología para crear programas, sistemas y contenidos y convertirse en alfabetizados digitales en un nivel adecuado para el mundo del futuro<sup>18</sup>.

La particularidad inglesa dice relación con que todo el país está dando un giro completo desde un currículum centrado en Office hacia las ciencias de la computación en una asignatura obligatoria desde los primeros años de primaria. Sin perjuicio de que Inglaterra contaba con profesores especialmente preparados para dictar los cursos de informática, el nuevo currículum implica un enorme desafío en términos de formación y apoyo al docente, así como en la generación

de orientaciones y la provisión de materiales de apoyo (textos, instrumentos de evaluación, software especializado, etc.). Fundar una nueva asignatura escolar como esta, requiere mucho más esfuerzo que mejorar una existente. Para esto, el Departamento de Educación ha contado con la colaboración de muchas otras instituciones comprometidas con esta transformación, como académicos y estudiantes de ciencias de la computación en las universidades, empresas privadas y organizaciones de la sociedad civil, como el mismo CAS mencionado anteriormente (CAS, 2015).

### **Estados Unidos**

A comienzos del año 2016, el presidente Obama lanzó la iniciativa *Computer Science For All* (CS for All), con el objetivo de empoderar a todos los estudiantes de K-12 para aprender computación y estar equipados con las habilidades del pensamiento computacional que les permitan ser creadores en la economía digital y ciudadanos activos en un mundo empujado por la tecnología<sup>19</sup>. De acuerdo con la declaración de esta iniciativa, la ciencia de la computación es una nueva habilidad básica necesaria para aprovechar las oportunidades económicas y promover la movilidad social.

*CS for All* es la primera iniciativa federal que da realce estratégico a la enseñanza de la computación en el sistema educativo americano. Se inyectarán recursos a los estados y distritos escolares para capacitar profesores, elaborar material instruccional y forjar alianzas estratégicas con empresas, medios,

<sup>17</sup><https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study>

<sup>18</sup>Los objetivos del programa de estudios fueron destacados en el Cuadro n°1 en la sección 2.

<sup>19</sup><https://www.whitehouse.gov/blog/2016/01/30/computer-science-all#Need>

fundaciones, así como otras organizaciones de la sociedad civil y profesionales interesados en colaborar, de manera de expandir el aprendizaje de esta disciplina en las escuelas. Asimismo, la iniciativa apoyará a instituciones que han venido desarrollando programas de apoyo a la enseñanza de la computación en secundaria desde mucho antes, como es el caso de *National Science Foundation (NSF)*<sup>20</sup>.

En efecto, Estados Unidos tiene una larga tradición en la enseñanza de computación en las escuelas secundarias, tanto en el marco de los cursos de informática o como parte de los esfuerzos para fortalecer la enseñanza de las disciplinas STEM. Sin embargo, desde comienzos de los 2000 venía retrocediendo el interés en este tipo de cursos. Según un estudio realizado en 2009-2010 por *The Association for Computing Machinery (ACM)* y *The Computer Science Teachers Association (CSTA)*, dos tercios de los estados consideraban conceptos y habilidades de las ciencias de la computación en sus currículos de secundaria, pero de manera muy irregular y opcional. En la mayoría de los casos, el uso de TIC en K-12 estaba enfocado en el uso de la tecnología para apoyar aprendizajes transversalmente en el currículum y para dar una alfabetización digital básica a los estudiantes (CSTA, 2010).

Si bien instituciones como las mencionadas NSF o CSTA tienen una vasta trayectoria en apoyar a profesores de computación en toda la nación, principalmente aportando desarrollo curricular y profesional, la situación general era que la computación no era considerada como parte del currículum obligatorio de la educación escolar. En este contexto, se espera

que la iniciativa presidencial potencie la experiencia previa y facilite la incorporación de las ciencias de la computación en la enseñanza de todos los estudiantes, especialmente en secundaria.

### **Israel**

Este país tiene una larga tradición en la enseñanza de la programación y ciencias de la computación. Desde los años 70's el currículum de secundaria contaba con un curso optativo donde se enseña programación en lenguaje BASIC, el que incorporó programación en LOGO y uso de aplicaciones generales en los 80's. A comienzos de los 90's el Ministerio de Educación formó un comité para repensar este curso, el que fue reformulado para centrarse en los conceptos fundamentales de ciencias de la computación y en el desarrollo del pensamiento algorítmico donde la programación es vista como "la manera de hacer que los computadores ejecuten los algoritmos" (Gal-Ezer, 1995, p73). A lo largo de la década siguiente este nuevo currículum fue gradualmente implementado en las escuelas secundarias. El año 2000 el Ministerio creó el *Centro Nacional para Profesores de Ciencias de la Computación (Machshava)*, organización dedicada a formar y acompañar el desarrollo profesional de los docentes de esta disciplina.

### **Nueva Zelanda**

A mediados de los años 70's, Nueva Zelanda introdujo la enseñanza de programación como parte de un curso de matemática aplicada al final de la secundaria especializada, lo que duró hasta mediados de los 80's, cuando se creó un curso de tecnología para estudiar

<sup>20</sup>Ver por ejemplo la iniciativa CS10K de la NSF en <https://cs10kcommunity.org/>

en forma práctica una diversidad de temas de índole tecnológica –desde alimentaria a digital– lo que dejó a la programación inmersa en un currículum mucho más amplio y sin espacio propio. En 2009 el Ministerio de Educación convocó a un panel de expertos con representantes de la industria, las universidades y las escuelas secundarias para revisar la enseñanza de tecnología. Las recomendaciones de este panel se tradujeron en un nuevo currículum para esta área a partir de 2011. Este nuevo currículum estableció cursos especiales electivos en los últimos años de la secundaria para la enseñanza de las tecnologías digitales que incluyen explícitamente la programación y las ciencias de la computación (Bell et al, 2010).

### **Costa Rica**

Costa Rica fue el primer país Latinoamericano en lanzar una política nacional de incorporación de las TIC a las escuelas en la segunda mitad de los 80's, el Programa Nacional de Informática Educativa (PRONIE), el que está inspirado en la visión de que las tecnologías pueden contribuir al desarrollo cognitivo de los estudiantes, en particular, que la programación puede ser un poderoso medio para desarrollar la capacidad de resolución de problemas y razonamiento lógico en los niños. Luego de casi tres décadas de gradual crecimiento, hoy la mayor parte de las escuelas primarias y secundarias urbanas de este país cuentan con un curso de informática educativa con clases semanales en el laboratorio de computación en las que los estudiantes trabajan proyectos –temáticamente vinculados con el currículum escolar– que conducen a productos de programación (inicialmente utilizaban

el lenguaje LOGO, actualmente usan Scratch). Si bien este curso de informática educativa es obligatorio, su evaluación no se considera en las calificaciones de los estudiantes (Muñoz et al, 2013). Actualmente, la Fundación Omar Dengo, responsable del PRONIE, está proponiendo modificar el currículum de este curso para incluir contenidos propios de las ciencias de la computación, pensamiento computacional, robótica y makers.

### **Chile**

Como parte de su política nacional de integración de TIC al sistema escolar, ENLACES, desde comienzos de los 90's el Ministerio de Educación de este país ha estado fundamentalmente enfocado en usar la tecnología, apoyar la enseñanza de las diferentes materias en forma transversal en el currículum de primaria y secundaria, así como en desarrollar habilidades digitales de gestión de información en Internet. Sin perjuicio de lo anterior, en los últimos años Chile ha promovido las habilidades de programación a través de talleres extra programáticos de robótica en escuelas secundarias<sup>21</sup>. Estos talleres, sin embargo, tienen un alcance limitado, pues cubren anualmente a cerca del 25% de las escuelas secundarias y dentro de estas, solo a grupos de estudiantes interesados y durante el año en que reciben apoyo explícito desde el Ministerio, por lo que no necesariamente alcanzan a constituir una oferta permanente para los estudiantes.

<sup>21</sup>Ver <http://www.enlaces.cl/proyectos/mi-taller-digital/>

## Uruguay

Uruguay fue el primer país del mundo en entregar una laptop a cada estudiante de primaria y secundaria con el propósito estratégico de mejorar la calidad educativa en un marco de equidad. Como parte de esta política, iniciada en 2007 y denominada PLAN CEIBAL, Uruguay desarrolla una serie de iniciativas complementarias para aprovechar esta infraestructura para el aprendizaje de los estudiantes, siendo una de estas los Laboratorios de Tecnologías Digitales (LabTeD). Los LabTeD son talleres extra programáticos que se realizan en cerca de la mitad de las escuelas secundarias y donde los estudiantes interesados realizan proyectos que tiene como resultado final el diseño y construcción de algún artefacto tecnológico de *hardware* y/o *software* basados en programación, robótica y sensores. Se espera que fruto de estas actividades, los estudiantes desarrollen competencias de orden superior como creatividad, colaboración y pensamiento crítico, entre otras. Asimismo, Ceibal promueve que estos LabTeD formen parte del currículum de los cursos de informática existentes en secundaria desde los años 90's y que hasta ahora han estado muy centrados en ofimática (Jara, 2016).

Con la excepción de Costa Rica, las políticas de nuestra región no tienen tradición ni esquemas institucionalizados para la enseñanza de temas vinculados con la computación en las escuelas. Sin embargo, como se ha ilustrado con los casos revisados, es posible encontrar iniciativas acotadas en formato de talleres extra curriculares que capturan el interés de estudiantes secundarios interesados y

hay un creciente interés en incorporar formalmente la computación en los cursos de informática. Prueba de ello es lo señalado para Uruguay y otros casos no abordados en esta revisión, como la Ciudad de Buenos Aires en Argentina, que recientemente modificó el currículum de los últimos años de secundaria para incorporar aspectos de esta disciplina<sup>22</sup>.

Por su parte, los países desarrollados revisados muestran una larga trayectoria con cursos que forman parte del currículum, principalmente en secundaria. Generalmente estos cursos, muchas veces electivos, consideran la enseñanza de la programación, pero están más centrados en el uso de las aplicaciones tecnológicas. En los últimos años, estos currículums están siendo objeto de revisiones para incorporar de manera más decidida las ciencias de la computación y ofrecerlos a todos los estudiantes. Se espera que estos cambios eleven el estatus y calidad de estos cursos y de los docentes que los dictan; y que la incorporación de temas más relevantes y desafiantes permita atraer a más estudiantes al estudio de esta disciplina.

<sup>22</sup><http://www.lanacion.com.ar/1791189-la-informatica-ya-forma-parte-del-plan-en-la-caba>

## 3.2. Decisiones difíciles

Las experiencias revisadas muestran que los procesos de toma de decisiones sobre la incorporación de la enseñanza de la computación en la educación son largos y complejos, y responden a las particulares historias y contextos de cada país. A continuación, se revisan algunas de las principales preguntas a considerar a la hora de incorporar la enseñanza de computación en la educación escolar.

### ***¿Se debe enseñar computación en la educación escolar?***

La principal pregunta que debe responderse es si la computación debe ser materia de la escuela o se sigue dejando como una carrera de la educación post-secundaria. Como se ha visto, los países más desarrollados están convencidos de que si no se comienza en la escuela, su liderazgo tecnológico está seriamente amenazado. Cabe preguntarse si en otros contextos también se justifica incorporar la enseñanza de esta nueva disciplina en los tiempos escolares, incluso al costo de desplazar otros aprendizajes tanto o más relevantes para el desarrollo de los niños (Hubwieser et al, 2015).

En particular, en Latinoamérica aún hay deudas muy importantes en términos de cobertura y aprendizajes básicos que demandan atención de las políticas públicas (TERCE, 2015); y cabe preguntarse con qué prioridad debiera entrar la enseñanza de computación en las agendas educativas de este continente. Es posible que diferentes países tengan distintas respuestas a esta pregunta, incluyendo una intermedia que proponga la instalación gradual de la computación en el sistema escolar al mismo tiempo que se desarrollan las capacidades nacionales que requiere su implementación masiva. Lo que parece ineludible es que, tarde o temprano, los países tendrán que responder esta pregunta mayor y definir una forma de abordaje.

En cualquier caso, los desafíos que se enfrentarían en el camino no son menos complejos que la decisión de tomarlo. Las ciencias de la computación son una disciplina relativamente joven y más lo es su reciente emergencia en el contexto escolar donde se enfrenta con un nuevo escenario muy diferente al de la educación terciaria. Hay todavía muchas preguntas sin una respuesta clara y los países embarcados en este movimiento están apostando y abriendo camino al andar.

### ***¿Es para todos los niños desde temprana edad?***

No hay una respuesta única sobre este tema. Algunos se inclinan por comenzar lo más temprano posible, indicando que hay buenas experiencias al respecto. Otros indican que es mejor cuando se ha logrado en los niños una cierta capacidad de abstracción (por ejemplo, a partir de los 11 años) de modo de estar preparados para entender conceptos más complejos como algoritmo, variables, recursión y otros. Posiblemente la respuesta es que hay un continuum de posibilidades desde lo más simple (programación básica) hasta lo más complejo. La experiencia de la mayoría de los países revisados es que comienzan la enseñanza de computación en secundaria; sin perjuicio de lo cual casos como Inglaterra y Costa Rica la inician desde primaria. Asimismo, para muchos es una materia extracurricular opcional, o bien electiva o de especialización a fines de la secundaria, pero la tendencia observada es ofrecer esta experiencia formativa a todos los estudiantes.

Son, por tanto, varias las opciones de arreglo curricular implicadas en la modalidad elegida para introducir la enseñanza de computación en las escuelas. Lo primero a resolver es si la enseñanza de computación se ofrece a través de cursos obligatorios para todos los estudiantes o en actividades opcionales que solo siguen los interesados. En el caso de tratarse de tiempos curriculares obligatorios, hay que decidir si se aborda en una sola asignatura, probablemente dedicada exclusivamente a la computación como es

la tendencia, o bien en forma transversal insertando diferentes tópicos en varias asignaturas existentes, como matemáticas y las diferentes ciencias (Carvalho et al, 2013).

La opción de incorporar la computación al interior de los cursos de otras disciplinas es un camino más complejo e incierto, debido a que normalmente sus currículos ya están estresados de sus propios contenidos y actividades; y que implicaría formar a una amplia gama de docentes en los fundamentos de ciencias de la computación que se quieren incorporar. Contar con una asignatura propia aparece, en cambio, como una alternativa más segura, aunque no menos desafiante, en particular en aquellos países donde no existe un curso de informática que se pueda utilizar con estos fines. En efecto, siendo el tiempo escolar un recurso limitado, el ingreso de una nueva asignatura implicaría restar tiempos de otras materias no menos importantes, cuyos defensores deben ser convencidos del alto valor que tendrá la nueva disciplina en la experiencia formativa de los estudiantes (Carvalho et al, 2013; Hubwieser et al, 2015).

### ***¿Cuál debe ser el contenido del currículum ofrecido?***

Una cuestión de siempre difícil definición es el contenido mínimo de las asignaturas escolares, aún en disciplinas estables y de larga tradición en la escuela, como matemáticas o ciencias. En el caso de la joven computación el desafío es aún mayor. Sin embargo, ya son varios los países que han realizado completos y detallados diseños curriculares y que llevan varios años probándolos en las escuelas. Esta experiencia es, sin duda, un muy buen punto de partida para la definición curricular que habría que abordar.

Sin embargo, estas definiciones no se han hecho en el aire, sino en países con cierta historia en esto; con docentes especializados en tecnología y programación; y con una red de organizaciones que pueden apoyar su implementación (como universidades y empresas,

entre otras). Es necesario, por tanto, ser conscientes de la necesidad de modular estos modelos a la realidad específica del país, que muchas veces no cuenta con estas mismas condiciones.

La definición del currículum es, además, espacio de disputa de intereses y visiones presentes en cada contexto, que deben ser articulados por la autoridad educativa en función de los intereses estratégicos del país. Cabe mencionar el caso inglés como ejemplo de este tipo de procesos. Como atestiguan varios protagonistas directos de la elaboración del nuevo currículum, la primera propuesta elaborada por un comité de expertos consultados por el Departamento de Educación incluía una mezcla balanceada de contenidos de ciencias de la computación junto con otros más vinculados a la cultura y alfabetización digital menos especializada. Sin embargo, esta propuesta fue reemplazada por otra centrada únicamente en las ciencias de la computación debido a la fuerte influencia de la industria y otros actores fuertemente vinculados con esta disciplina (Cobo, 2014).

### ***¿Cómo preparar a los docentes?***

El principal desafío involucrado en la incorporación de la computación en la educación escolar es la formación de la base docente capaz de convertir en aprendizajes los lineamientos curriculares que se establezcan. Incluso los países con más tradición en este campo, siguen señalando las dificultades implicadas en la adecuada preparación del profesorado. Para apoyar este proceso, Israel, por ejemplo, creó un centro especializado e Inglaterra está invirtiendo enormes sumas en desplegar una diversidad de estrategias para transformar a estos profesionales (CAS, 2015). Debe considerarse que estos países cuentan, en general, con un contingente docente de buena calidad y, en particular, con profesores especializados en tecnología con, al menos, un mínimo de conocimientos sobre programación.

La enseñanza de la nueva disciplina requiere de docentes con conocimientos conceptuales y prácticos muy sólidos junto con una formación pedagógica que permita convertirlos en experiencias de aprendizaje efectiva para los estudiantes (CSTA, 2005). Si bien los fundamentos de la computación están, al menos en teoría, disponibles en las carreras de pregrado universitario del ámbito tecnológico; las didácticas o metodologías y sus recursos (textos, orientaciones, evaluaciones, lenguajes, software) están recién comenzando a madurar en las mismas aulas escolares. Falta mucho aún para contar con un cuerpo consolidado de conocimiento sobre la enseñanza de las ciencias de la computación en el contexto escolar, con claridad de las diferentes estrategias y recursos para cada edad (Hubwieser et al, 2015).

Los países que están avanzando en esta materia están definiendo estándares para la formación de los nuevos docentes de computación, estableciendo requisitos y trayectorias para su formación inicial y sistemas de certificación. Sin embargo, aún no es claro si se podrá interesar a suficientes profesores en esta nueva especialidad; ni cuántas décadas podría tomar tener un nivel aceptable de docentes en cada aula del país.

### ***¿Qué infraestructura se necesita?***

Si bien las ciencias de la computación tienen una componente conceptual que puede prescindir hasta cierto grado del uso de computadores, el aprendizaje de la disciplina se materializa y alimenta de la resolución de problemas programables en dispositivos digitales. Tener acceso fluido a computadores en la escuela es, por tanto, un requisito mínimo para la implementación de estas políticas, ya sea en la forma de laboratorios con suficientes computadores para el trabajo con un curso; u otro tipo de acceso como carros móviles o maletas para llevar laptops a las aulas o simplemente estudiantes que portan sus propios dispositivos móviles.

La realidad latinoamericana es a este respecto muy irregular: mientras hay países donde la gran mayoría de las escuelas cuentan con laboratorios o laptops, hay también un grupo importante donde la mayoría de las escuelas no tienen este acceso básico (Jara, 2015). Lo mismo ocurre a nivel de los hogares de los estudiantes. En este contexto, la enseñanza de computación pasa necesariamente por importantes inversiones en equipamiento.

Pero aún disponiendo de laboratorios con computadoras, como ocurre en varios países de la región, la decisión de utilizarlos en cursos de computación debe ser analizada con cuidado. Estos laboratorios son el resultado de políticas que han invertido décadas en promover el uso transversal de las TIC de manera de apoyar los aprendizajes en las diferentes asignaturas. Al asignarle un nuevo uso a los laboratorios, sin otras inversiones que permitan utilizar las TIC en las aulas, se impediría que los docentes de las otras clases utilicen la tecnología con sus estudiantes.

En suma, la implementación de una política educativa para introducir la enseñanza de la computación en la educación escolar, tal como lo están haciendo países desarrollados, requiere abordar asuntos claves –como su gradualidad, currículum, formación docente e infraestructura– que deben diseñarse cuidadosamente considerando el contexto de aspiraciones y capacidades de cada país. Es recomendable que las decisiones de políticas en este tipo de asuntos estén respaldadas por estudios que consideren todas las variables e inversiones requeridas para que las promesas sean factibles de realizar. De lo contrario, como suele ocurrir en nuestra región, se trataría de otro titular imposible de cumplir.



## 4. Iniciativas independientes

Esta sección recoge parte de la experiencia de numerosas organizaciones privadas y de la sociedad civil que promueven la enseñanza de la computación, entre ellas, aquellas participantes en los encuentros organizados por Microsoft en Argentina, Chile y México a comienzos de 2016 (ver anexo). Estas experiencias involucran empresas privadas, fundaciones, ONG's, organizaciones internacionales y agrupaciones de personas como gremios o asociaciones (asociaciones de software, por ejemplo). También hay iniciativas impulsadas por organismos estatales y que no forman parte de las políticas educacionales para el sistema escolar, sino más bien vinculadas con políticas hacia la juventud o a inclusión digital de la ciudadanía, como es el caso de los proyectos promovidos por la Organización Internacional de la Juventud y las secretarías de la juventud de los países orientadas a los jóvenes que han desertado del sistema educativo formal.

La experiencia de estas organizaciones, junto a muchas otras alrededor del mundo, confirman la existencia de una gran riqueza y variedad de iniciativas concretas en torno a la programación de niños y jóvenes. La evidencia indica que estas iniciativas ya no representan unos pocos hechos aislados y esporádicos de algunos entusiastas, sino que muchas de ellas y los argumentos que las sustentan, constituyen experiencias –en ocasiones masivas– de alfabetización digital y aprendizaje de programación

computacional en niños y jóvenes, principalmente en contextos extra escolares y, en particular, en grupos de jóvenes en riesgo social. Estas iniciativas complementan –y en muchos casos, adelantan– políticas gubernamentales buscando responder a problemas sociales aprovechando las oportunidades que ofrece la tecnología.

### 4.1. Iniciativas y sus motivaciones

En este campo es posible encontrar una gama tan amplia de motivaciones, propósitos y focos, como de iniciativas y organizaciones involucradas. Por ejemplo, hay iniciativas promovidas por empresas que ofrecen en Internet recursos para el autoaprendizaje de jóvenes interesados en la programación y que organizan alianzas y eventos para incentivar su aprovechamiento en escuelas y clubes; hay iniciativas llevadas a cabo por fundaciones y ONG's que trabajan directamente con jóvenes de barrios vulnerables con el propósito de ampliar sus oportunidades laborales; hay algunas que se focalizan en el trabajo con mujeres o estudiantes con necesidades especiales; y también las hay que ofrecen talleres extra curriculares de programación en las escuelas.

A continuación, se describen resumidamente algunas iniciativas que ilustran la amplitud de los esfuerzos desplegados en torno a programación.

En primer término, cabe destacar *YouthSpark*<sup>23</sup>, una iniciativa global de Microsoft en alianza con organizaciones no gubernamentales, gobiernos, empresas y familias, que busca incrementar el acceso de los jóvenes al aprendizaje de ciencias de la computación, de modo de empoderarlos para aspirar a mayores logros para si mismos, sus familias y sus comunidades. A la fecha, más de 300 millones de jóvenes de todo el mundo han participado de YouthSpark.

Los participantes pueden tomar cursos de ciencias de la computación y utilizar recursos digitales a través de los cuales van aprendiendo acerca del pensamiento computacional y desarrollan habilidades de resolución de problemas. Algunos de las numerosas iniciativas que comprende YouthSpark son: YoPuedoProgramar y el Curso de Microsoft Virtual Academy "Aprendiendo a Programar"; Kodu; e Imagine Academy, cuyo objetivo es diseñar y crear juegos y aplicaciones digitales utilizando herramientas informáticas y conocimientos de ciencias de la computación; así como DigiGirlz y Girls Who Code, orientados a niñas, que fomentan su participación en talleres de ciencias de la computación para motivarlas hacia las ciencias y tecnologías, ingeniería y matemáticas a través de la vinculación con mujeres que trabajan en Microsoft y la participación en talleres con otras niñas.

El programa de *YouthSpark* para América Latina, denominado *YoPuedoProgramar*, también ha tenido un impacto considerable con más de 3 millones de participantes. Esta iniciativa permite iniciarse en los primeros pasos de la programación y abre las

puertas para que los interesados sigan formándose y obtengan un diploma especial en la Academia Virtual de Microsoft<sup>24</sup>. Esta iniciativa nació en el 2013 a partir de las recomendaciones de la agenda Post2015 de la Organización Internacional de Juventud (OIJ), que incluía enseñar programación a jóvenes de secundaria y no escolarizados<sup>25</sup>. El alcance de YoPuedoProgramar ha sido amplificado por la colaboración de las secretarías de la juventud que facilitaron el uso de los centros juveniles y la visión de la OIJ que ha recomendado fortalecer este trabajo conjunto para aumentar las oportunidades de los jóvenes.

Cabe destacar, asimismo, la iniciativa *La Hora del Código*<sup>26</sup>, un movimiento global, en que han participado más de cien millones de estudiantes de 180 países. Consiste de una introducción de una hora a las ciencias de la computación, diseñada para mostrar que todo el mundo puede aprender a programar, sin tener experiencia previa y a partir de los 4 años de edad, y así comprender los fundamentos básicos de la disciplina. Los tutoriales, todos de una hora de duración, están disponibles en más de 40 idiomas.

CodeClub<sup>27</sup>, por su parte, es una red mundial de voluntarios que enseñan a programar a estudiantes de 9 a 11 años después de la jornada escolar durante 1 hora a la semana, ya sea en la escuela o en un centro comunitario. A la fecha hay más de 6.000 clubes en el mundo. Los proyectos de programación se relacionan con juegos digitales, animaciones y páginas web.

Por último, ScratchEd<sup>28</sup> nace de la gran popularidad del lenguaje de programación Scratch, cuya

<sup>23</sup>YouthSpark hub: <https://www.microsoft.com/about/philanthropies/youthspark/youthsparkhub/>

<sup>24</sup><https://mva.microsoft.com/>

<sup>25</sup>La elaboración de esta agenda, que forma parte de los Objetivos de Desarrollo Sustentable de la OIJ, contó con el aporte y la experiencia de Microsoft y sus programas de filantropía y educación. Recomendaciones similares han sido señaladas en cumbres iberoamericanas de juventud y en la Cumbre de Presidentes Iberoamericanos. Ver recomendaciones a la Agenda Post 2015 de la OIJ en "AGENDA DE DESARROLLO E INVERSIÓN SOCIAL EN JUVENTUD: UNA ESTRATEGIA POST 2015 PARA IBEROAMÉRICA" (OIJ, 2013)

<sup>26</sup>La Hora del Código <https://hoursofcode.com/es>

<sup>27</sup>Code Club <https://www.codeclubworld.org>

<sup>28</sup>ScratchEd <http://scratched.gse.harvard.edu>

página de usuarios<sup>29</sup> incluye hoy en día más de 15 millones de proyectos que son compartidos y que pueden utilizarse y modificarse. ScratchEd nace el año 2009, y articula una comunidad en línea en la cual los educadores y apoderados interesados en programación con Scratch pueden compartir experiencias de uso curricular de este lenguaje, en el aula o en el hogar, intercambiar recursos, formular preguntas o contactar a otros docentes.

Las iniciativas de alcance global recién mencionadas destacan en un mar de acciones de diverso alcance, foco y duración. Sin perjuicio de esta diversidad, en este campo de rica actividad es posible identificar visiones, diagnósticos y propósitos compartidos. En particular, muchas comparten el interés por aprovechar la actitud positiva y conocimiento tecnológico básico de niños y jóvenes, especialmente de sectores vulnerables, para potenciar sus oportunidades laborales y de emprendimiento. Esta visión se apoya en los siguientes elementos:

En primer lugar, las encuestas, investigaciones y experiencias indican que los jóvenes de hoy están hyper-conectados (de manera transversal a los niveles socio-económicos) a través de las redes sociales. Si bien esta motivación ofrece oportunidades para que aprovechen las tecnologías para su formación y su futuro laboral, también se sabe que el uso de redes sociales no implica el desarrollo de suficientes habilidades para un manejo eficaz y amplio de las oportunidades que ofrece la tecnología. La dinámica de los empleos en todos los sectores productivos y de servicios va requiriendo de nuevas habilidades,

muchas de ellas asociadas con la capacidad de utilizar eficazmente y de crear con tecnologías digitales, de modo que resulta necesario preparar a los jóvenes de mejor forma para insertarse en el nuevo escenario laboral. Esto se relaciona -en algunos países que sufren de altas tasas de deserción escolar- con la necesidad de aplicar estrategias de retención de alumnos en escuelas secundarias y de ofrecerles herramientas de empleabilidad.

En segundo lugar, la tecnología se está posicionando como una componente importante y en ocasiones central en los emprendimientos (start-ups). Los jóvenes que buscan crear nuevas empresas u ONG's, necesitan informarse, construir propuestas y crear redes; tareas en las que las tecnologías son necesarias hoy en día. En esta línea, varias ONG consideran el potencial transformador social de las TIC, al observar sus efectos en jóvenes y adultos que buscan caminos nuevos para acercarse al mundo laboral o para enriquecer sus emprendimientos. Asimismo, los países desarrollados -y gradualmente los que están en vías de desarrollo- están acusando un déficit de mano de obra calificada en términos de alfabetización digital. Esto junto con ser un desafío, es a la vez una oportunidad para nuestros países, de preparar a sus jóvenes para asumir desafíos profesionales de futuro.

## 4.2. Estrategias de trabajo con jóvenes

La experiencia de diversas iniciativas muestra la importancia de generar estrategias para invitar y atraer a los jóvenes hacia la programación, en especial a aquellos menos motivados por la programación

<sup>29</sup>Scratch <https://scratch.mit.edu>

o que no han tenido la oportunidad de acceder a iniciativas en que puedan participar. Algunas de estas estrategias que reportan diferentes organizaciones que trabajan en este ámbito, son las siguientes:

Un factor común parece ser el de implementar los espacios de aprendizaje y de participación en los mismos lugares donde están los jóvenes, en particular en sus barrios y lugares de esparcimiento. La consigna es "ir donde los jóvenes están". Ej: en barrios vulnerables, hay numerosas iniciativas tales como la "Casas del Futuro" de la Sub-Secretaría de Juventud en Argentina, "Casas Poder Joven" del IMJUVE en México, "DesarrolloArte" y "Acá estamos" de INJUV-Chile.

Otro aspecto que resalta como necesario en las organizaciones activas en este ámbito, es el de la difusión. Para los jóvenes y apoderados, resulta importante comunicarles buenas experiencias con programación, contar *buenas historias de vida* – testimonios personales en que se constata un desarrollo personal o grupal, un caso de éxito, etc. También es necesario borrar el estigma que programar es aburrido o solo para adictos a la tecnología a través de testimonios de niñas y niños que desarrollan sus proyectos de programación de manera colaborativa, trabajando en grupos, en ambientes atractivos de aprendizaje que además constituyen espacios lúdicos.

Algunas organizaciones están haciendo un esfuerzo especial por integrar a las niñas y a las jóvenes a programar, en base a proyectos que les sean personalmente atractivos. Otras organizaciones tienen su foco en personas con necesidades educativas especiales y sus testimonios reflejan que este es un ámbito de gran potencial de desarrollo, con aproximaciones hacia áreas de emprendimiento y empleabilidad, en algunos casos incluso relacionadas con discapacidades específicas. Organizaciones sin fines de lucro como CDI, Trust for the Americas a nivel regional, ProAcceso en México, Laboratoria en Perú, Chile y México, Comunidad IT en Argentina, entre

muchas otras, se han sumado a cubrir un espacio educativo en habilidades tecnológicas y programación. Estos esfuerzos buscan crear oportunidades económicas concretas de empleo y emprendimiento a jóvenes de contextos socioeconómicos adversos, incluyendo intermediación laboral en las áreas de tecnología de empresas.

Adicionalmente, cabe tratar en forma destacada el movimiento Makers, que está siendo abordado por diversas iniciativas de la región.

### **El movimiento Makers**

Martin (2015), resume la filosofía de este movimiento y las promesas que ofrece para transformar los aprendizajes. En resumen, se reconocen tres factores convergentes que han dado sustento e impulso al movimiento Makers:

En primer lugar, los nuevos procesadores de bajo costo (tipo Arduino, RaspBerry, Intel Galileo) que han dado vida a lo que se denomina Internet de las Cosas y que han popularizado la programación de todo tipo de artefactos domésticos e industriales, sin requerir de conocimientos avanzados de electrónica o programación. A estos avances tecnológicos se suman las impresoras 3D y máquinas CNC, de costos decrecientes disponibles en los espacios Makers ("Makers Spaces") y que comienzan a verse en las escuelas. En segundo lugar, la programación de computadores por parte de escolares que ya ha sido abordada en este documento, ha sido revitalizada en los últimos años con lenguajes simples de aprender por escolares incluso de primer ciclo de educación básica (Scratch, Snap), gratuitos y en español. Tercero, el renovado impulso a pedagogías relacionadas con aprendizajes basados en proyectos (ABP) que sean significativos para los estudiantes y que ofrecen formas de enfrentar los procesos de enseñanza-aprendizaje abordando de manera interdisciplinaria problemas reales, interesantes y complejos. En esto, los profesores asumen gradualmente un rol de guías

de aprendices y los alumnos un rol activo, de trabajo colaborativo y de construcción colectiva y gradual de conocimientos durante la planificación y desarrollo de sus proyectos.

Estas tres tendencias unidas dan vida al movimiento *Makers* en escuelas, hogares y talleres. Ejemplos son las ferias *Makers*<sup>30</sup> en muchos países, con miles de jóvenes compartiendo sus creaciones a partir de componentes electrónicos simples, programables en lenguajes sencillos. El potencial educativo de los *Makers* se vincula con iniciativas curriculares (en particular en laboratorios) en asignaturas tales como física y mecánica (electricidad, movimiento, sonido, calor, luz, principios de motores, generadores, energías renovables, etc.); biología y botánica (tratamiento de plantas, nutrientes, riego, fotosíntesis, huertos, etc.) e incluso música (generación de sonidos en dispositivos análogo-digitales) y arte (vinculando el valor de sensores con colores y formas).

El interés adicional que atrae a los profesores trabajando con sus estudiantes en la línea *Makers* es la posibilidad de desarrollar su creatividad, la indagación, el espíritu emprendedor y el "aprender haciendo" en torno a proyectos que les son atractivos a los jóvenes. Dados los bajos costos y simplicidad para dar los primeros pasos, se utiliza una estrategia de probar y prototipar de manera progresiva. Esto ayuda a los estudiantes a "atreverse" y arriesgarse con nuevas ideas.

Un aspecto interesante del movimiento *Makers* es la actitud que genera en los jóvenes al sentirse capaces

de crear invenciones que generalmente se ven lejanas a sus capacidades (p.ej. fabricar un mini submarino, controlar la seguridad de un espacio o programar un riego automático). Esto es relevante "especialmente en niñas que tienden a abandonar los dominios de ciencia y matemática en educación secundaria"<sup>31</sup>. Los estudiantes necesitan desafíos que los motiven a indagar en forma genuina (y no solo por cumplir con una guía escolar). Los proyectos que se pueden desarrollar en la línea *Makers* justamente inspiran la indagación y la exploración de ideas utilizando tecnologías modernas y de bajo costo.

### 4.3. Desafíos

De la experiencia de diversas iniciativas también surge un conjunto de desafíos para el desarrollo futuro de las mismas que ameritan atención. En particular:

Es conveniente estimular que otras organizaciones se sumen a estas iniciativas, las que muchas veces no han visto esta oportunidad o han creído que no está en su foco. Sin embargo, la experiencia muestra que hay una gran diversidad de ONG's incorporando programación con los jóvenes y logrando insertarlas en su foco de acción. Para este desafío, es necesario comunicar de manera efectiva el trabajo que se realiza en los diferentes contextos y para diferentes actores (en particular en barrios de contextos vulnerables), en un lenguaje que los interpele. Específicamente, el desafío es ¿Qué y cómo informar de estas iniciativas para sumar voluntades? En este segmento también están los padres y apoderados que debieran conocer los beneficios de que sus educandos aprendan

<sup>30</sup><http://makerfaire.com>

<sup>31</sup><http://www.weareteachers.com/blogs/post/2015/04/03/how-the-maker-movement-is-transforming-education>

computación, pues ellos pueden propiciar la participación en iniciativas en las escuelas. Para las políticas públicas, es necesario articular un conjunto sólido de argumentos, experiencias, evidencias y testimonios que sensibilicen a los tomadores de decisión en Ministerios de Educación y Ministerios Sociales.

La experiencia de quienes trabajan en terreno, indica que en sectores excluidos y de contextos vulnerables existe una brecha de información relativa a las oportunidades que ofrece la tecnología o que pueden aprovecharse a través de la tecnología. Es necesario que quienes viven y trabajan en esos sectores puedan responderse a ¿qué puedo aprender? ¿qué puedo hacer con lo aprendido? Y a la vez, poder conectarse con otras personas de contextos similares para establecer redes informales, resolver problemas y sumarse a oportunidades laborales o formativas.

Si bien ya existe cierta experiencia en inclusión de género en algunas iniciativas, es necesario hacer esfuerzos adicionales para reunir más argumentos y testimonios que interpelen a las niñas y a las jóvenes a participar. Un camino es dar a conocer los testimonios de niñas que se han destacado ¿qué han hecho y cómo lo han hecho? ¿qué dicen de sus experiencias y qué les dicen a otras niñas?

## 5. Conclusiones

Este documento busca contribuir con las decisiones y esfuerzos relativos a la enseñanza de la computación en las nuevas generaciones de niños y jóvenes latinoamericanos. Para eso, se han presentado los principales antecedentes, experiencias y aprendizajes sobre esta materia, recogidos a través de una recopilación de la literatura especializada y del testimonio de responsables de iniciativas reunidas en tres encuentros organizados por Microsoft en Argentina, Chile y México a comienzos de 2016.

Esta revisión ha permitido constatar el creciente interés en promover la enseñanza de la computación—ciencias de la computación, pensamiento computacional y/o programación— en niños y jóvenes tanto dentro como fuera de la región. Prueba de ello, es que las políticas educacionales alrededor del mundo están revisando sus currículums escolares para fortalecer su enseñanza en todos los estudiantes; y que seamos testigos de múltiples iniciativas impulsadas por empresas y organizaciones de la sociedad civil que complementan estos esfuerzos públicos para ampliar las oportunidades de los jóvenes.

Si bien los orígenes de este movimiento se remontan hasta los años 70's, en la última década ha aumentado la preocupación de que la formación tecnológica que se está dando a las nuevas generaciones no sea suficiente para aprovechar los beneficios de la revolución digital. Para responder a este escenario, muchos países están cambiando el foco de sus currículums de tecnología en la educación escolar formal desde la alfabetización digital, hacia el estudio sistemático de la disciplina de la ciencia de la computación, el pensamiento

computacional y la programación. Este cambio está implicando, además, un esfuerzo enorme para formar a los docentes responsables de estos cursos, así como para proveer todos los apoyos necesarios para el aprendizaje de los estudiantes y jóvenes (equipos, textos, acompañamiento, etc.). Se trata de una transformación de largo aliento, cuyos resultados debieran verse reflejados tanto a nivel individual, en mayores habilidades de resolución de problemas, mayor comprensión del mundo digital y mayor interés por carreras tecnológicas; como a nivel social, en un mayor desarrollo de las industrias basadas en nuevas tecnologías y mayor competitividad económica de los países.

Adicionalmente, diversas organizaciones están desplegando iniciativas orientadas a enseñar habilidades digitales y programación a niños y jóvenes, especialmente fuera del contexto escolar y en situación de riesgo social, con la confianza de que estas nuevas capacidades les permitan insertarse en mercados laborales de mayor valor y proyección. Hay iniciativas de diversa índole —mientras algunas ofrecen recursos de autoaprendizaje online en todo el mundo, otras trabajan en barrios urbano-marginales con grupos vulnerables de una pequeña ciudad— pero todas comparten la visión de que un mayor dominio de la tecnología puede abrir nuevas oportunidades a los jóvenes y sus comunidades.

La experiencia internacional de políticas e iniciativas como las revisadas en este documento no permite aún confirmar con evidencia sólida los beneficios prometidos por la enseñanza de la computación; en

particular, su impacto en las capacidades cognitivas de los estudiantes –razonamiento lógico, resolución de problemas–. Sin perjuicio de lo anterior, la necesidad de dotar a los jóvenes con herramientas más eficaces para tener más oportunidades laborales y contribuir con la innovación y el crecimiento en un mundo crecientemente digital, parecen ser por ahora argumentos suficientes para sustentar estas inversiones.

Con todo, los países latinoamericanos deben enfrentar la pregunta de si abordarán de alguna manera la enseñanza de la computación a sus niños y jóvenes, tal como lo están haciendo los países más desarrollados. Existen muchos caminos para esto, algunos más radicales, como el seguido por Inglaterra al instalar un curso de computación obligatorio desde la primaria escolar; hasta estrategias más graduales y opcionales dentro y fuera del sistema educativo formal, como talleres extra-programáticos, olimpiadas o ferias que apoyen y movilicen a los interesados. En este sentido, el trabajo de las organizaciones que desarrollan iniciativas independientes puede ser un recurso muy valioso para promover el desarrollo de estas nuevas capacidades en muchos países de la región. Si bien en este documento las políticas y estas iniciativas han sido presentadas por separado, dando la falsa impresión de que corren por carriles separados, la sinergia entre ambos esfuerzos es un camino muy importante a explorar.

En cualquier caso, la enseñanza de computación requiere abordar diversos desafíos, los cuales han sido revisados en las secciones precedentes de este documento. Entre estos cabe destacar la necesidad de diseñar un currículo pertinente, asegurar una adecuada formación de los docentes, proveer la infraestructura requerida y articular los esfuerzos de las diferentes políticas e iniciativas en redes que le permitan ampliar su impacto y sustentabilidad.



# Referencias bibliográficas

Anderiesz, M. (2014). Erase/Rewind- The Story of Computer Science Education in the UK.

Bell, T., Andreae, P., & Lambert, L. (2010). Computer science in New Zealand high schools. Procedures 12th Australasian Computing Education Conference (ACE 2010), Brisbane, Australia, 15–22.  
Retrieved from <http://dl.acm.org/citation.cfm?id=1862223>

Benitti, F. B. V. (2012). Exploring the educational potential of robotics in schools: A systematic review. *Computers & Education*, 58(3), 978–988. <http://doi.org/10.1016/j.compedu.2011.10.006>

Brown, N. C. C., Kölling, M., Jones, S. P., Ave, J. J. T., Humphreys, S., & Sentance, S. (2013). Bringing Computer Science Back into Schools : Lessons from the UK, 269–274.

BRS (2012). Shutdown or Restart: The way forward for computing in UK schools. The Royal Society: London.  
Carvalho, T., Andrade, D., Silveira, J., Auler, V., Cavalheiro, S., Aguiar, M., Foss, L., Pernas, A., Reiser, R. (2013). Discussing the Challenges Related to Deployment of Computational Thinking in Brazilian Basic Education. In 2013 2nd Workshop-School on Theoretical Computer Science (pp. 111–115). IEEE. <http://doi.org/10.1109/WEIT.2013.27>

CAS (2009). Computing at Schools: the state of the nation, Computing at Schools Working Group Report for the UK Computing Research Comitee: UK.

CAS (2011). Computing at Schools: International Comparison, Version 5: UK.

CAS (2015). Closing the gap to achieve a world class computing teaching workforce, Computing at Schools, British Computer Society, the chartered institute for IT: UK.

Clements, D. H., & Gullo, D. F. (1984). Effects of computer programming on young children's cognition. *Journal of Educational Psychology*, 76(6), 1051–1058. <http://doi.org/10.1037/0022-0663.76.6.1051>

Cobo, C. (2014). Experiencia del caso inglés en la integración de TIC y la definición de estándares de habilidades TIC para docentes (1997-2013), Enlaces: Santiago

Computational Thinking: <https://medium.com/@lorenaabarba/computational-thinking-i-do-not-think-it-means-what-you-think-it-means-6d39e854fa90#.n4h5bmeoh>

CSTA (2005). *The New Educational Imperative: Improving High School Computer Science Education*. Using worldwide research and professional experience to improve U.S. Schools. Computer Science Teacher Association Curriculum Improvement Task Force: New York.

CSTA (2010). *Running on Empty: The failure to teach K-12 Computer Science in the Digital Age*, ACM-Association for Computing Machinery/ Computer Science Teachers Association: USA

CSTA (2011). *CSTA K-12 Computer Science Standards*, Computer Science Teachers Association: New York  
EUN (2014). *Computing our future Computer programming and coding - Priorities, school curricula and initiatives across Europe*. European Schoolnet.

Retrieved from [http://www.eun.org/c/document\\_library/get\\_file?uuid=521cb928-6ec4-4a86-b522-9d8fd5cf60ce&groupId=43887](http://www.eun.org/c/document_library/get_file?uuid=521cb928-6ec4-4a86-b522-9d8fd5cf60ce&groupId=43887)

Gal-Ezer, J., Beerli, C., Harel, D., & Yehudai, a. (1995). A high school program in computer science. *IEEE Computer*, 0198, 1–21. Retrieved from <http://teacher.tchcvs.tc.edu.tw/mhtsai/essay/high-school-program.pdf>

Garneli, V., Giannakos, M.N., Chorianopoulos, K. (2015). *Computing Education in K-12 Schools: A review of Literature*, Conference Paper, IEE Global Engineering Education Conference, March 2015, Tallinn University of Technology: Tallinn, Estonia.

Grover, S., Pea, R. (2013). *Computational Thinking in K-12: A Review of the State of the Field*. *Educational Researcher*, 42(1), 38–43. <http://doi.org/10.3102/0013189X12463051>

Grover, S., Cooper, S., & Pea, R. (2014). *Assessing computational learning in K-12*. *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education - ITiCSE '14*, (April 2016), 57–62. <http://doi.org/10.1145/2591708.2591713>

Havenga, M., Breed, B., Mentz, E., Govender, D., Govender, I., Dignum, F., & Dignum, V. (2013). *Metacognitive and Problem-Solving Skills to Promote Self-Directed Learning in Computer Programming : Teachers ' Experiences*. *SA-eDUC JOURNAL*, 10(2), 1–14. Retrieved from [http://www.nwu.ac.za/sites/www.nwu.ac.za/files/files/p-saeduc/sdl\\_issue/Havenga et al. Metacognitive and problem-solving skills to .pdf](http://www.nwu.ac.za/sites/www.nwu.ac.za/files/files/p-saeduc/sdl_issue/Havenga_et_al.Metacognitive_and_problem-solving_skills_to_.pdf)

Hubwieser, P., Armoni, M., & Giannakos, M. N. (2015). *How to Implement Rigorous Computer Science Education in K-12 Schools? Some Answers and Many Questions*. *ACM Transactions on Computing Education*, 15(2), 1–12. <http://doi.org/10.1145/2729983>

Jara, I. (2016). *Revisión comparativa de iniciativas nacionales de aprendizaje móvil en América Latina. El caso del Plan Ceibal de Uruguay*. UNESCO: Paris

Jara, I. (2015). *Infraestructura digital para educación: Avances y desafíos para Latinoamérica*. IPE-UNESCO: Buenos Aires.

Lye, S. Y., & Koh, J. H. L. (2014). *Review on teaching and learning of computational thinking through*

programming: What is next for K-12? *Computers in Human Behavior*, 41, 51–61. <http://doi.org/10.1016/j.chb.2014.09.012>

Martin, L. (2015). The Promise of the Maker Movement for Education. *Journal of Pre-College Engineering Education Research* *Journal of Pre-College Engineering Education Research*J-PEER *Journal of Pre-College Engineering Education Research*, 5(5), 1–30. <http://doi.org/10.7771/2157-9288.1099>

Microsoft YoutSpark: <http://www.microsoft.com/about/philanthropies/youthspark/>

Muñoz, L., Brenes, M., Bujanda, M.E., Mora, M., Nuñez, O., Zúñiga, M. (2014). Las políticas TIC en los sistemas educativos de América Latina: Caso Costa Rica,; Buenos Aires Unicef.

Papert, S. (1988). A critique of technocentrism in thinking about the school of the future. *Children in the Information Age: Opportunities for Creativity, Innovation and New Activities*, (2), 3–18.

Papert, S. (1993). *Mindstorms: Children, Computers and Powerful Ideas*. Perseus Books: New York.

Selby, C., Woollard, J. (2013). Computational thinking: the developing definition. Retrieved from [http://eprints.soton.ac.uk/356481/7/Selby\\_Woollard\\_bg\\_soton\\_eprints.pdf](http://eprints.soton.ac.uk/356481/7/Selby_Woollard_bg_soton_eprints.pdf)

Selwyn, N. (2002). Learning to Love the Micro: the discursive construction of 'educational' computing in the UK, 1979-89, *British Journal of Sociology of Education*, vol 23, no 3, pp 427-443.

Sunkel, G., Trucco, D. (2012). Las tecnologías digitales frente a los desafíos de una educación inclusiva en América Latina: Algunos casos de buenas prácticas, CEPAL: Santiago de Chile

Stevenson Committee (1997). *Information and Communications Technology in UK Schools: an independent inquiry*, London: The independent ICT in School Commission.

Valverde-Berrocoso, J., Fernández-Sánchez, M. R., & Garrido-Arroyo, M. del C. (2015). El pensamiento computacional y las nuevas ecologías del aprendizaje. *RED - Revista de Educación a Distancia*, 46(46), 1–18. <http://doi.org/10.6018/red/46/3>

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33. <http://doi.org/10.1145/1118178.1118215>

Wing, J. M. (2010). Computational Thinking: What and Why? *The link - The Magazine of the Varnege Mellon University School of Computer Science*, (March 2006), 1–6. Retrieved from <http://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why>

TERCE (2015). *Tercer Estudio Regional Comparativo y Explicativo (TERCE)*. OREALC/UNESCO: Santiago. Disponible en < <http://www.unesco.org/new/es/santiago/terce/>>

# Anexo

En los meses de marzo y abril de 2016, Microsoft Latinoamérica organizó tres encuentros con expertos y organizaciones vinculadas con la promoción de la computación en la juventud de la región. Los encuentros se realizaron en Buenos Aires (Argentina), Santiago de Chile y Ciudad de México. Las conversaciones sostenidas durante estos encuentros ha sido un insumo relevante para la elaboración de este documento. A continuación, se listan los participantes en cada uno de los encuentros.

## Argentina

Alicia Bañuelos – Ministra de Ciencia y Tecnología de la Provincia de San Luis

Pedro Robledo – Subsecretario de Juventud de la Nación – Ministerio de Desarrollo Social de la Nación

Camila Crescimbeni – Directora Nacional de Juventud – Subsecretaría de Juventud de la Nación – Ministerio de Desarrollo Social de la Nación

Silvia Carranza – Presidente - CILSA

Laura Wierzchlo – Directora Ejecutiva - CILSA

Karina Cuzzani – Coordinadora Académica - CILSA

Pablo Godfried – Director - ComunidadIT

Mariel Sabra – Especialista FOMIN - IDB/MIF

Alberto Croce – Director Ejecutivo - Fundación SES

Alejandra Solla – Directora Adjunta - Fundación SES

Natalia Jasin – Responsabilidad Social Empresaria y Sustentabilidad – Intel Argentina

Mariela Relman – Directora - Chicos.net

Marcela Czarny – Directora - Chicos.net

Aníbal Carmona – Presidente - CESSI

Norberto Capellán – Presidente - CICOMRA

Agustín Dellagiovana – Director Nacional – Subsecretaría de Responsabilidad Social - Ministerio de Desarrollo de la Nación

Fifi Palou – Dirección Nacional de Responsabilidad Social - Ministerio de Desarrollo Social

Ludovico Grillo – Secretario de Educación – Municipalidad de Vicente López

Gustavo Cuccuraza – Adicra

Melina Masnatta – Coordinadora de Investigación en Educación – Centro de Implementación de Políticas Públicas para la Equidad y el Crecimiento (CIPPEC)

Adrian Escandarani – Docente – Escuela ORT

Diego Fernandez Slezak – Docente – Facultad de Ciencias de Exactas (Universidad de Buenos Aires)

## Chile

Nicolás Farfán - Director Nacional – Instituto Nacional de la Juventud (INJUV)  
Eugenio Severin – Consultor en Educación y Tecnologías. Director Ejecutivo - Tu Clase mi País.  
Pedro Hepp – Académico de la Pontificia Universidad Católica de Valparaíso.  
Marisol Alarcón – Directora Ejecutiva – Laboratoria Chile.  
Mónica Retamal – Directora Ejecutiva - Fundación Kodea  
Carolina Rivera – Directora Ejecutiva – Innovacien  
Camila Batista – Global Network Manager – CDI  
María Cristina Benavente – Investigadora - Comisión Económica para América Latina y el Caribe CEPAL  
Eugenio Vergara – Director Ejecutivo – CDI  
Hugo Martínez – Director Pedagógico de Colegium/Eduinnova  
Ariel Gringaus – Gerente General - Colegium  
Samuel Delgado - Director de Programa – Panal Chile  
Daniela Huanca – Directora de Desarrollo - Panal Chile  
Juan Luis Ramírez – Director Ejecutivo - Fundación de Vidal Rural  
Claudia Peirano – Fundadora - Grupo Educativo  
Komal Dadlani – CEO y Cofundadora - Lab4you  
Javier Ignacio Errázuriz Araneda – Asesor - Mineduc  
Marcelo Vera. - Director Ejecutivo del Centro de Educación y Tecnologías del Mineduc - Enlaces, Mineduc  
Rodrigo Ferrada – Socio Director – Magenta  
Paulina Rojas –Magenta Consultora  
Macarena Badilla – Analista Responsabilidad Empresarial - Fundación AES Gener  
Daniela Trucco – Oficial de Asuntos Sociales, Division de Desarrollo Social, - CEPAL  
Cristóbal Letelier – Periodista – Instituto Nacional de la Juventud – INJUV.

## México

René Asomoza – Director General, ILCE  
Miguel Angel Cardona – SEP @prende  
César Santos – SEP @prende  
Pilar Muñoz – SNTE  
Alfredo Martínez de la Torre – Director, ANUIES  
Héctor Bernal – Director, Eural  
Carlos Astengo – Director, Tec de Monterrey  
Lorenzo Valle – Director de Vinculación, Tec de Monterrey  
Juan Miguel Pérez Rangel – Fundación Proacceso  
Rosario Jessica Diaz – Fundación Proacceso  
Kristian Salazar – Gerente de Fomento y Promoción, SCT - AEM  
Ignacio Meza – Director de Instituciones y Cultura, Lazos  
Josue García Dávila – Líder de proyecto, Grupo Carso  
Angélica Mora – Directora General, Executive Global System  
Juan Carlos Martínez – Coordinador de Tecnologías Aplicadas, CONALEP

Dr. Carlos León Hinojosa – Director General, CONOCER  
Humberto Merritt Tapia – Director, IPN  
Ana Tamez – UT Santa Catarina  
Ulises Beltrán, IMJUVE  
Juan Carlos Rico Campos, Director de Bienestar y Estímulos a la Juventud, IMJUVE  
Milagros Fernández – Director Ejecutivo, CLASE  
Trust for the Americas – Sergio Pérez  
Daniela Rivera – Directora de los Puntos México Conectado, SCT  
Catalina Demidchuk – Directora de Vinculación, Codeando México  
Miguel Salazar – Director General, Codeando México  
Rafael Linares – Supera México

*(c) 2016 Microsoft Corporation. Todos los derechos reservados. Este documento se proporciona "como-es." La Información y opiniones expresadas en este documento, incluyendo direcciones URL y otras referencias a sitios Web de Internet, pueden cambiar sin previo aviso. Cada cual es responsable y asume el riesgo de uso de la información aquí contenida.*

*Este documento no proporciona ningún derecho legal a cualquier propiedad intelectual de cualquier producto de Microsoft. Usted puede copiar y utilizar este documento siempre y cuando cite la fuente. MICROSOFT NO HACE NINGUNA PROMESA O GARANTÍA, EXPRESA O IMPLÍCITA, EN ESTE DOCUMENTO.*

Esta versión se terminó de editar en agosto de 2016.