

Introducción

El avance de las tecnologías digitales y su impacto en el desarrollo de la vida personal, profesional y ciudadana ha generado la necesidad de desarrollar ciertos saberes y habilidades que permitan aprovechar, por parte de los estudiantes, dichas tecnologías en pos de lograr expresarse y hacer cosas significativas con ellas.

En este sentido, desde la sociedad, hay una demanda creciente hacia las instituciones educativas con relación a que deben propiciar experiencias que permitan a los estudiantes comprender e intervenir de manera más rica su propio contexto.

En particular, estamos hablando de promover habilidades para: resolver problemas, ya sea de manera personal y a la vez con otros; aplicar su creatividad a situaciones a las que se enfrenten; planificar y organizar proyectos en diversos contextos; adquirir fluidez en el uso de herramientas para tener usos efectivos de las tecnologías; buscar y recuperar información fidedigna y por sobre todo crear y expresarse con tecnologías.

El pensamiento computacional es un núcleo de aprendizaje prioritario y se define como una forma de resolver problemas que toma experiencias y técnicas de las ciencias de la computación y que se basa en generar soluciones a partir del análisis, la descomposición, la lógica, la abstracción y la algoritmia.

Formalizando el concepto de pensamiento computacional

Resolver problemas comunes de la vida diaria aplicando las experiencias y técnicas de las ciencias de la computación significa pensar como los científicos informáticos y los desarrolladores de software abordan las soluciones para los problemas específicos de su disciplina. Esto es, desarrollar la capacidad de descubrir **patrones** en los problemas, aplicar soluciones que se puedan expresar en forma de **algoritmos**, desarrollar la **abstracción** y aplicar la **lógica** son algunas de las habilidades y conocimientos que más influyen en el Pensamiento Computacional (PC).

En esta concepción del PC, se utilizan los siguientes nuevos conceptos:

Patrones: Un patrón es algo que se repite de manera continua en una secuencia o en conjunto elementos. Cuando se intenta resolver un problema, en la etapa de análisis, descubrir qué patrones asociados existen permite entender mejor las distintas partes que lo componen y que deben ser tenidas en cuenta en la etapa de diseño de soluciones. Es decir, que saber detectar patrones y descubrir las reglas que los definen ayuda a entender mejor los problemas y a diseñar sus posibles soluciones.



Figura 1. Patrones numéricos

La búsqueda de patrones se puede realizar no solo en los elementos de un problema sino también en los procedimientos que usamos para resolverlos



¿Cuál es el patrón que une los cordones con los zapatos?

Debemos recordar que las situaciones que involucran patrones habitan nuestra vida diaria, por ejemplo, se usan patrones climáticos para crear los modelos de pronósticos meteorológicos; las mascotas descubren patrones de respuesta de sus dueños a sus acciones y así adaptan su comportamiento a una manera esperada la próxima vez; cuando manejamos un automóvil observamos, de manera inconsciente, el patrón de conducta del conductor del vehículo que se encuentra delante nuestro, esto nos permite tomar precauciones ante posibles accidentes. Así, al identificar patrones, podemos hacer predicciones, crear reglas y resolver problemas más generales. Esta generalización de la resolución del problema es un camino a la abstracción; que se desarrollará más adelante; por lo tanto, una promueve la otra.

Por ejemplo, una generalización de la figura 1, expresado como un modelo matemático sería el siguiente

$$(n - 1) + 5$$

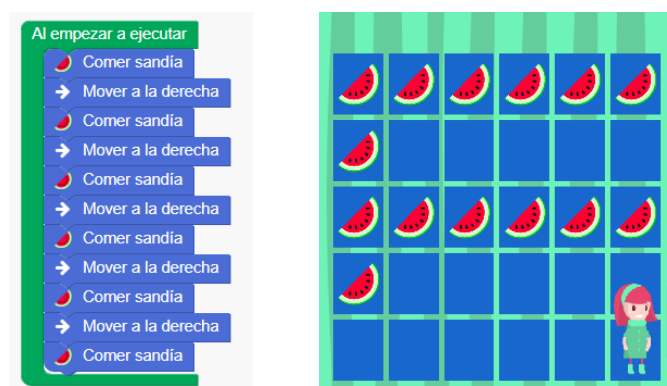
Que expresa la forma de la “Serie numérica” involucrada.

Otra característica importante de los patrones, resaltada por los científicos informáticos es que contribuyen a la rapidez y la eficiencia mediante **la reutilización de los métodos ya creados** (los patrones identificados)

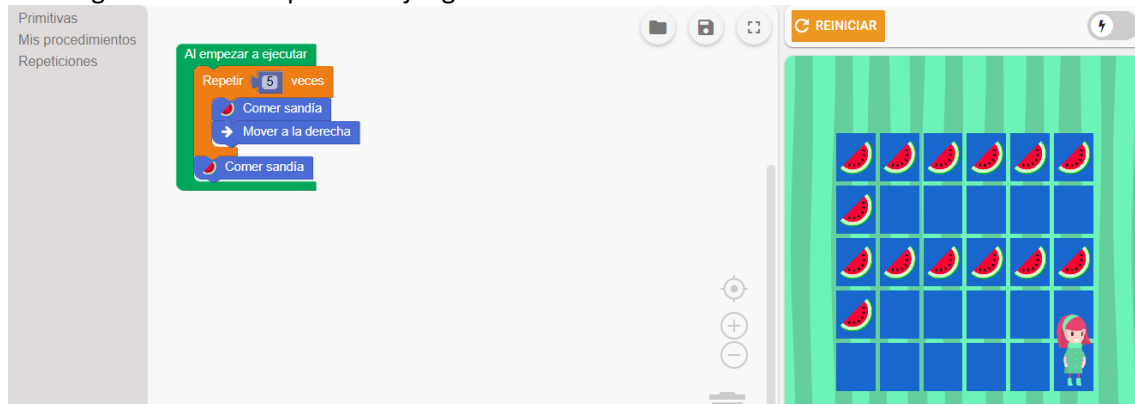
Por ejemplo, en el sitio pilasbloques.programar.ar se proponen diversos juegos que tienen por objetivo desarrollar estos elementos del PC en edades tempranas. Por ejemplo, el Juego María y las Sandías propone que Ud establezca las “primitivas” disponibles para generar un algoritmo que resuelva este problema. Una primitiva es una orden o instrucción **disponible y válida**. Para este ejemplo identificamos el problema, las primitivas y el objetivo es organizar las primitivas para que se cumpla el objetivo:



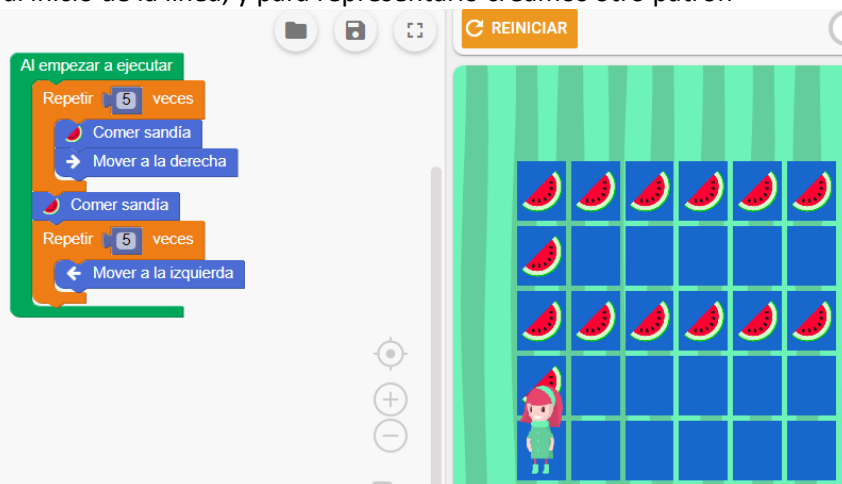
Suponga que entonces resuelve el hecho de comer las sandías de una fila, la siguiente figura muestra la secuencia de primitivas para solucionarlo:



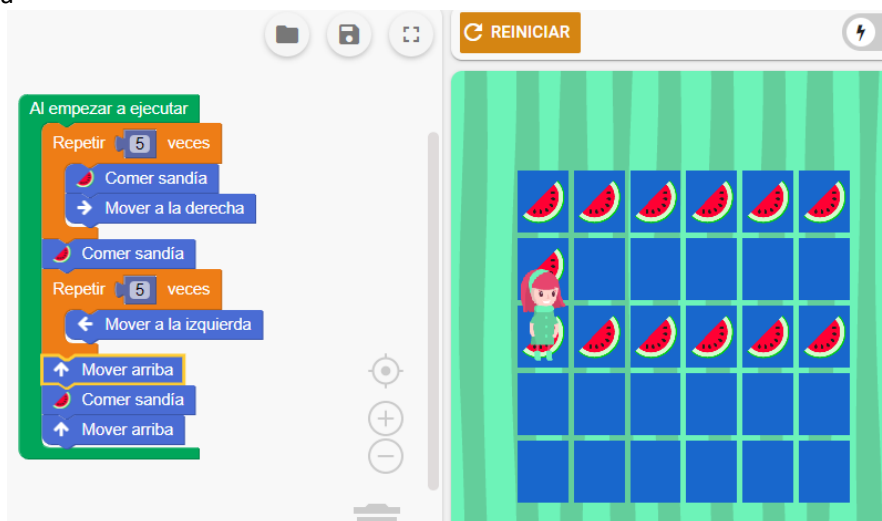
Pero vemos que hay una secuencia de primitivas que se repetirían para realizar esta actividad. Esta repetición es un patrón de movimientos en este caso. Entonces podríamos crear un patrón de la siguiente manera para este juego



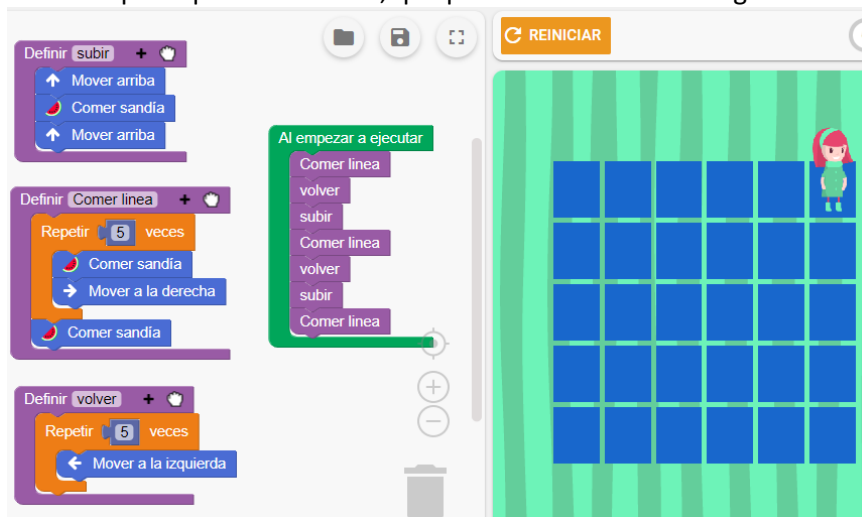
Como puede observar se usa un bloque denominado repetición, se indica el número de veces que se repite el patrón. Esto facilita la comprensión de las primitivas a usar y su aplicación. Siguiendo con el juego vemos que existen varios patrones. Por ejemplo, luego de comer María debe volver al inicio de la línea, y para representarlo creamos otro patrón



Ahora podemos notar que debemos subir dos líneas y además comer en el proceso la sandía intermedia















Ahora notamos que deberíamos repetir diversos patrones para que se complete el objetivo. Aquí es donde entra a jugar la reutilización de patrones. En este caso dentro del pensamiento computacional la técnica de la programación utiliza como mecanismo para poder reutilizar patrones los denominados módulos (en un tipo de programación denominada Programación Estructurada) u operaciones/métodos (denominados así en otro tipo de programación denominado Programación Orientada a Objetos). En este caso el juego nos permite crear estas operaciones con la opción procedimientos, que podría realizarse de la siguiente manera



Cuando se crean estas operaciones, y le asignamos un nombre; luego simplemente se deben “invocarlos” en una secuencia adecuada y entonces los mismos resolverán el problema. Fíjese como se están reutilizando las operaciones; y como dos de esos procedimientos poseen repetición de patrones.

Otra característica de los patrones es que requiere de un nivel de abstracción. Por ejemplo, observe el siguiente cuadro














4	8	16	32	¿Cuál sigue?
				¿Cuál sigue?
				¿Cuál sigue?
				¿Cuál sigue?

Ejemplos de otras actividades con patrones.

¿Cuál es el siguiente?” Observe que el ejercicio se realiza con números o bien con una secuencia de rotaciones, u objetos.

Ya volveremos sobre la abstracción. Ahora piense que los patrones también requieren pensar en actividades de clasificación. Al examinar cada uno de los elementos de un problema puede crear reglas de clasificación. La clasificación es parte de la lógica y el trabajo de conjuntos. Y ciertamente se reconocen patrones.

El uso de la técnica de reconocimiento de patrones para asistir la clasificación es también utilizado frecuentemente por las ciencias de la computación, ya que permiten establecer reglas para trabajar con distintos tipos de elementos que componen un problema. Por ejemplo ud puede responder el planteo del siguiente ejercicio:

<p>¿Se pueden asociar estas imágenes en grupos?, ¿cuáles?</p> <div style="text-align: center;">  </div>	<p>Posibles soluciones:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; padding: 5px;"> Comunicación:  </td> <td style="width: 50%; padding: 5px;"> Comida:  </td> </tr> <tr> <td style="width: 50%; padding: 5px;"> Con personas:  </td> <td style="width: 50%; padding: 5px;"> Sin personas:  </td> </tr> </table>	Comunicación: 	Comida: 	Con personas: 	Sin personas: 
Comunicación: 	Comida: 				
Con personas: 	Sin personas: 				

Ejemplos de clasificaciones

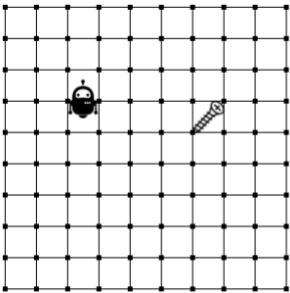
Pero ¿puede realizarlo un ordenador a través de un programa? Ciertamente que sí.

Algoritmos: Un algoritmo es una secuencia **finita**, **ordenada** y lógica de pasos para llegar a hacer una tarea determinada. Ejemplos de algoritmos en la vida cotidiana pueden ser:

- Una receta de cocina
- Un manual de un artefacto que indica cómo ensamblarlo.
- Un listado de acciones a realizar en caso de que se produzca una emergencia en un edificio determinado

Se puede pensar también que los algoritmos son un objeto de comunicación, ya que, como lista de pasos ordenados, comunican terceros como hacer una tarea. En lo cotidiano, el papel de una persona está asociado a entender y “ejecutar” estos algoritmos. Observe que, al indicar la secuencia de patrones, en realidad lo que estamos generando es un algoritmo con ellos. Ahora que, si se dedica a la programación de programas para ordenadores, la regla de oro será siempre: “Antes de codificar diseñe o modele los algoritmos”. Los algoritmos en este caso serán las secuencias de instrucciones que debe realizar el “ordenador” para realizar su tarea. Estos algoritmos codificados forman una parte del software que desarrollador crea (ya veremos más adelante que referirnos a un programa debemos incluir el código y su documentación).

En un ejemplo anterior vimos como se creó un juego para desarrollar el PC. Este juego sigue un algoritmo. Podemos imaginar el siguiente ejemplo como base para crear algoritmos que resuelvan la necesidad de que un Robot recoga un tornillo, y veremos que la metodología que usemos para crear ese algoritmo será similar a la que usó para el juego de comer sandías:

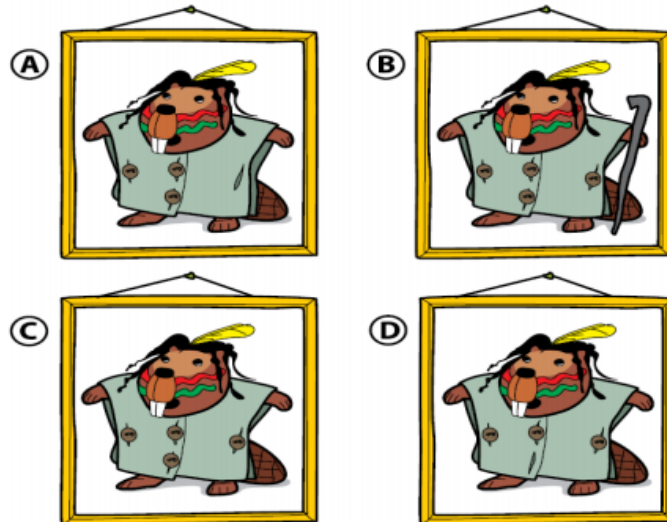
<p>Ejemplo de escenario.</p> <div style="text-align: center;">  </div>	<p>Ejemplo de instrucciones.</p> <ul style="list-style-type: none"> • Izquierda • Derecha • Arriba • Abajo • Recoger tornillo <p>Para resolver el ejercicio de la grilla izquierda bastaría con crear el programa:</p> <pre>Derecha Derecha Derecha Derecha Recoger tornillo.</pre> <p>Se puede pensar en complejizar el conjunto de instrucciones o los escenarios a utilizar tanto como se quiera.</p>
---	---

Lógica: La lógica estudia los principios de la demostración de los argumentos mediante la comprobación de la validez de expresiones, las cuales pueden ser evaluadas como: ciertas/verdaderas o falsas/no verdaderas.

Así, la lógica estudia las inferencias y el pensamiento humano, proporcionando principios para determinar qué tipo de evidencia es apropiada para una situación y su control de validez.

En computación, la lógica se utiliza tanto en el análisis como en la resolución de problemas. Por ejemplo, al momento de diseñar reglas para clasificar o para tomar decisiones.

Observe la siguiente figura



A partir de estas cuatro fotos, se debe elegir una de acuerdo con dos condiciones que se cumplan a la vez, por ejemplo:

1. que el personaje de la foto no tenga ningún palo y
2. que todos los botones de su saco deben estar cerrados

Este tipo de ejercicios es cada vez más utilizado en las entrevistas de trabajos, donde se evalúa las habilidades de PC de los aspirantes. En este caso claramente se trata de medir la capacidad de habilidad lógica de una persona.

En las ciencias de la computación se trata del estudio y aplicación de la **lógica booleana**. Para este ejemplo hay dos condiciones y ambas deben cumplirse, por lo que se aplica el operador booleano "Y", denominado más formalmente conjunción (también escrito con el símbolo " \wedge ") que determina que a partir de dos entradas verdaderas se obtiene una salida verdadera. ¿De donde salen estas afirmaciones?

Las tablas de verdad son herramientas de la lógica formal que asisten a la comprobación del valor de verdad de razonamientos. La tabla de verdad para el operador "Y" de conjunción es:

p	\wedge	q
V	V	V
V	F	F
F	F	V
F	F	F

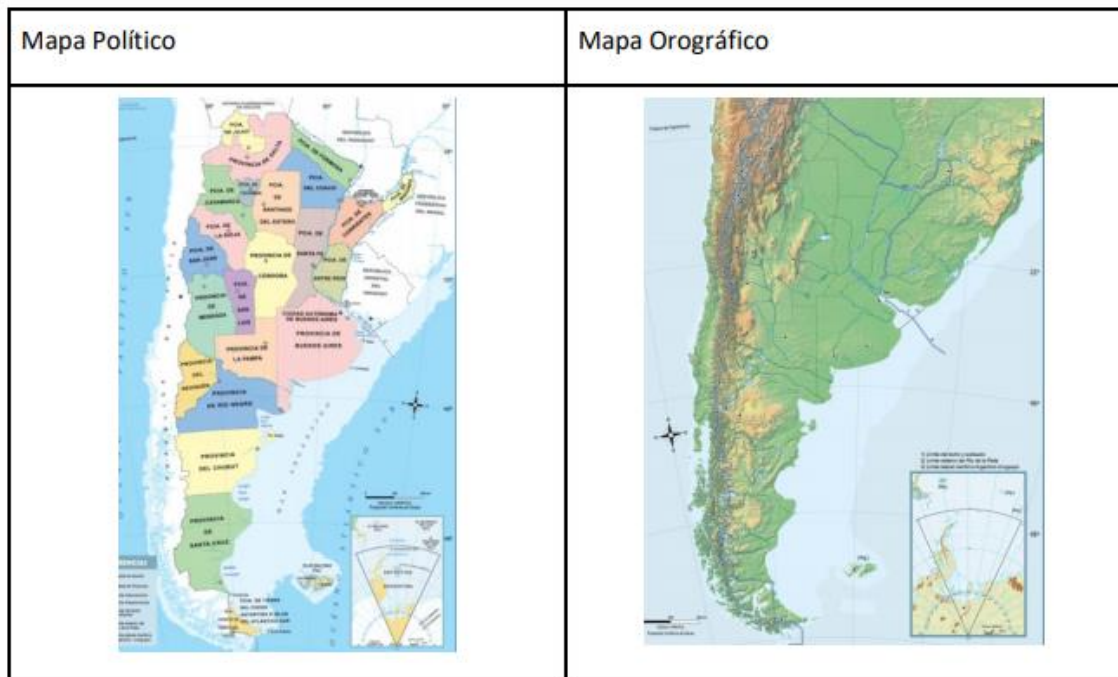
Tabla de verdad para el operador de conjunción " \wedge "

Es evidente que se profundizará en los aspectos de los operadores booleanos que se utilizan a la hora de crear algoritmos.

Abstracción: se refiere a poder crear una representación de una idea sólo teniendo en cuenta los elementos que son importantes a un problema específico. La esencia de la abstracción se basa en preservar la información relevante e ignorar la que no lo es. Aquí anticipamos que dentro de la programación orientada a objetos las capacidades de abstracción son realmente muy importantes.

El proceso de abstracción, aunque pueda sorprenderle es natural en torno a las actividades humanas.

Un ejemplo de abstracción puede darse en los mapas. Cuando el interés está en las divisiones políticas se suele ignorar casi todos los otros posibles detalles. Ahora, sí lo que nos interesa son formaciones rocosas y relieves debemos acudir a una representación diferente, donde se las realce y se escondan otros tipos de atributos.



Ejemplos de mapas de Argentina. La abstracción se centra en aspectos esenciales y se ignoran otros. Así en estos mapas de un mismo país revelan diferentes tipos de información

La capacidad de abstracción es importante para el desarrollo del PC, ya que no solo resulta útil a la hora de analizar un problema sino también para crear soluciones. En particular, al momento de representar objetos del mundo real en un ordenador, se los debe abstraer de muchas de sus características para simplificarlos y hacer posible su representación y manipulación. Por ejemplo, para un sistema de stock de un supermercado, puede no importarle la forma, el color o la variedad de un paquete de papas fritas, pero sí en cambio le puede importar la fecha de vencimiento, el código de barras y el precio de compra. Para este sistema, entonces, un paquete de papas fritas es el conjunto de esos tres datos, mientras que para nosotros es algo completamente diferente.

Tal como ha sido mencionado, las técnicas asociadas al pensamiento computacional suelen ser trabajadas de forma conjunta. En particular la abstracción está íntimamente relacionada con el reconocimiento de patrones y la generalización (o reutilización), como así también lo está al momento de aplicar y utilizar algoritmos.

Adquirir habilidades mediante el pensamiento computacional no implica necesariamente el desarrollo de habilidades relacionadas con la programación de ordenadores.

Como se desarrolla el pensamiento computacional

Es necesario avanzar en prácticas que permitan profundizar su comprensión y ejercitar las habilidades vinculadas al PC. A continuación, se presentan una serie de caminos, basados en distintos enfoques, para continuar trabajando:

- Actividades desconectadas: se basan en ejercicios y problemas que no requieren de pantallas y que, por lo general, se pueden resolver con lápiz y papel. Es ideal pasar por un kiosco de periódicos y solicitar revistas de ejercicios basados en juegos de lógica, otras propuestas pueden incluir actividades kinestésicas (aquellas que involucran el movimiento y el cuerpo) así también como juegos de mesa. Permite que las personas se enfoquen en los problemas, en las capacidades y en los conceptos sin generar distracciones provenientes de posibles problemas tecnológicos y de interfaz. Sin embargo, como desventaja cabe aclarar que estas actividades suelen ser limitadas en cuanto a su potencial para resolver problemas reales y que es recomendado complementarlas con prácticas posteriores que incorporen el uso tecnologías digitales.
- Plataformas de práctica de programación: estas actividades corresponden a un enfoque más tradicional que se basa en resolver ejercicios de programación en plataformas en línea. Una aproximación más popular y moderna está constituida por aquellas que se basan en ambientes de trabajo visual con interfaz de tipo “arrastrar y soltar bloques”. En Argentina el Portal Program.AR de la Fundación Sadosky dispone de una colección de ejercicios en línea y para descargar de este tipo (<http://pilasbloques.program.ar/> que es el lugar donde se presenta el juego de comer sandías usado en un ejemplo anterior). La ventaja de este tipo de actividades se encuentra en la posibilidad de desarrollar la fluidez en el uso de herramientas a partir de diseñar soluciones a problemas (y ponerlas a prueba) o desarrollar artefactos propios.