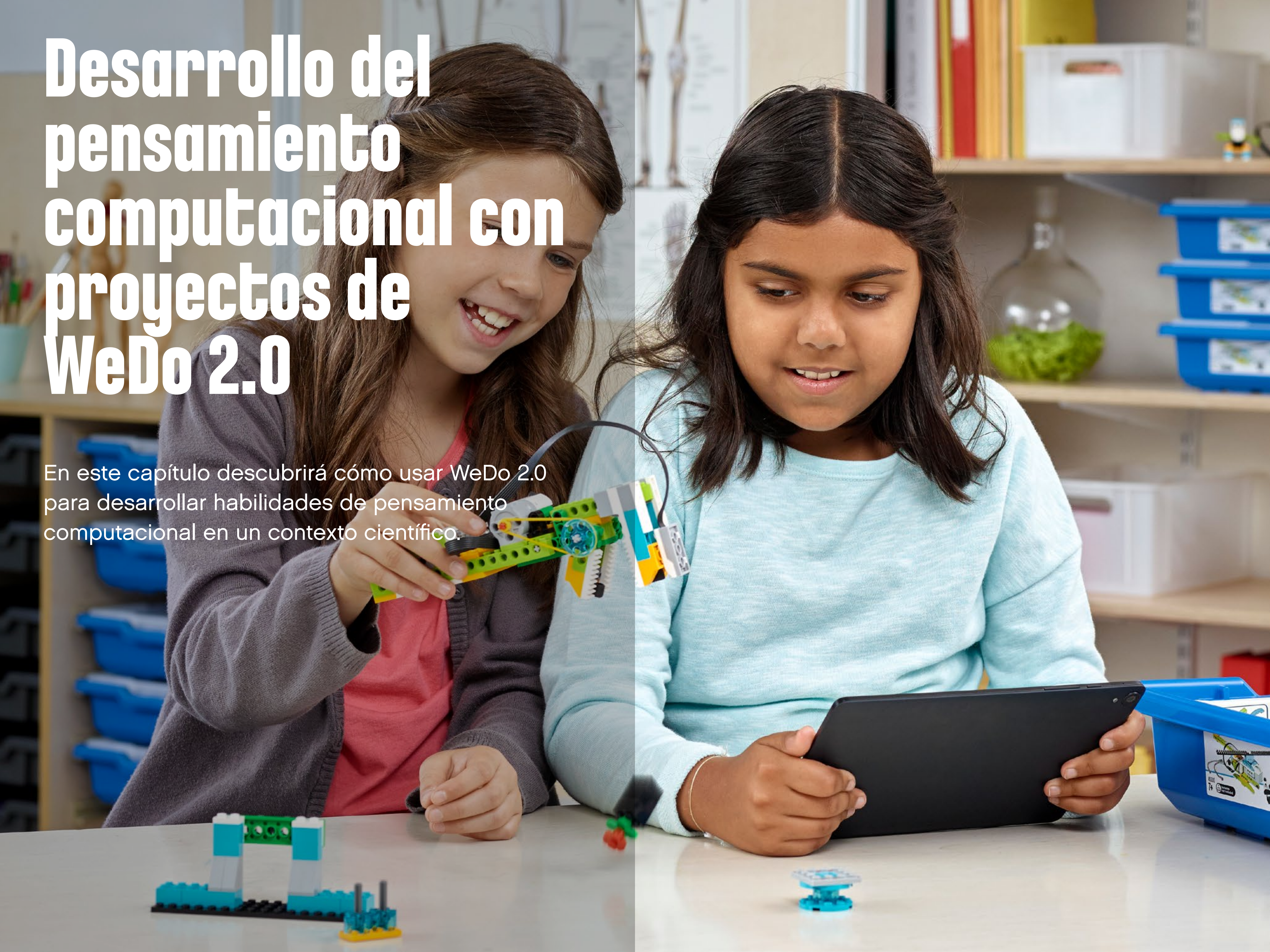


Desarrollo del pensamiento computacional con proyectos de WeDo 2.0

En este capítulo descubrirá cómo usar WeDo 2.0 para desarrollar habilidades de pensamiento computacional en un contexto científico.





Desarrollar el pensamiento computacional con proyectos de LEGO® Education WeDo 2.0

A LEGO® Education le complace presentar este proyecto, que se ha diseñado especialmente para utilizarse en las aulas de colegios de educación primaria con el fin de desarrollar las habilidades de pensamiento computacional de los estudiantes.

El pensamiento computacional es un conjunto de habilidades de la que cualquier persona puede hacer uso para resolver problemas cotidianos. En WeDo 2.0, estas habilidades se desarrollan a lo largo de las distintas fases de cada proyecto. Se han identificado todas las oportunidades de desarrollo en cada uno de los proyectos, de manera que puede centrarse en las que considere más importantes para usted y sus alumnos.

Todos los proyectos de WeDo 2.0 combinan el uso de ladrillos de LEGO® con un lenguaje de programación basado en iconos, lo que permite a los estudiantes encontrar soluciones a problemas mientras se familiarizan con los principios de programación.

WeDo 2.0 desarrolla el pensamiento computacional mediante actividades de codificación que dan vida a las creaciones de los estudiantes, felices y emocionados por descubrir más cosas.





Informática, pensamiento computacional, codificación

Mientras que la ciencia y la ingeniería surgieron en épocas tempranas de la humanidad, la informática tiene una historia mucho más reciente. Sin embargo, además de influenciar la manera en la que abordamos la ciencia y la ingeniería, esta joven disciplina también ha cambiado nuestro estilo de vida.

La informática es una disciplina CTIM que comparte características con la ciencia, la tecnología, la ingeniería y las matemáticas.

Todas las disciplinas CTIM ofrecen oportunidades para desarrollar una mentalidad y un conjunto de prácticas permanente. Entre estas prácticas encontramos la capacidad para formular preguntas, diseñar soluciones y comunicar resultados.

El pensamiento computacional es otra de estas prácticas. Se trata de una manera de pensar mediante la que cualquier persona puede resolver problemas.

El pensamiento computacional puede definirse como un conjunto de habilidades, entre las que se incluye el pensamiento algorítmico. “Codificar” o “codificación” son términos que pueden usarse para describir la acción de crear un algoritmo.

Por tanto, la codificación es un vehículo para desarrollar el pensamiento computacional en un contexto CTIM.

Disciplinas CTIM

Ciencia, tecnología, ingeniería,
matemáticas, informática

Desarrollar una mentalidad y un conjunto de prácticas permanente

1. Formular preguntas y solucionar problemas.
2. Usar modelos.
3. Diseñar prototipos.
4. Investigar.
5. Analizar e interpretar datos.
6. Usar el pensamiento computacional.
 - a. Descomponer
 - b. Abstraer
 - c. Utilizar un pensamiento algorítmico (código)
 - d. Evaluar
 - e. Generalizar
7. Implicarse en un argumento a partir de la evidencia.
8. Obtener, evaluar y comunicar información.



¿Qué es el pensamiento computacional?

El primero en usar la expresión “pensamiento computacional” fue Seymour Papert, aunque la profesora Jeannette Wing es famosa por haber divulgado la idea. Ella define el pensamiento computacional como:

“los procesos mentales involucrados en la formulación de problemas y sus soluciones, de tal modo que estas últimas se representen de una forma que puedan llevarse a cabo eficazmente por un agente que procese información”.
(Wing, 2011)

El pensamiento computacional se utiliza en diferentes campos y situaciones, así como en nuestra vida diaria. Las habilidades de pensamiento computacional están presentes en la ciencia, la ingeniería y las matemáticas. Estas habilidades pueden definirse de la siguiente forma:

Descomposición

La descomposición es la habilidad para simplificar un problema en partes más pequeñas para hallar una solución más fácilmente. De este modo, resulta más fácil explicar el problema a otra persona o separarlo en tareas. A menudo la descomposición conduce a la generalización.

Ejemplo: A la hora de irse de vacaciones, la preparación (o el proyecto) puede dividirse en subtareas: reservar billetes de avión, reservar una habitación en un hotel, hacer la maleta, etc.

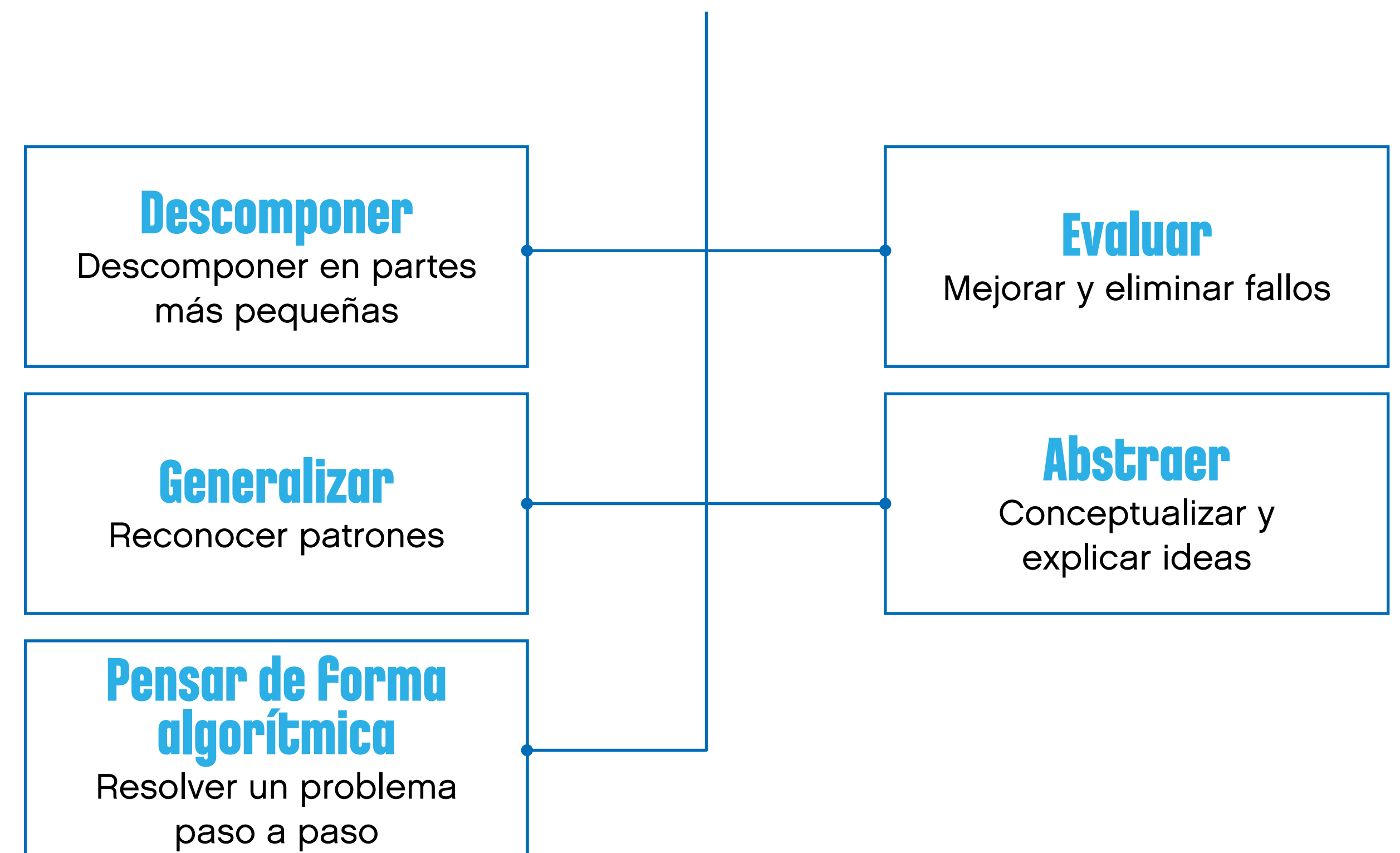
Generalización (reconocimiento de patrones)

La generalización es la habilidad para identificar las partes de una tarea que se conocen o se han visto en algún otro sitio. A menudo la generalización conduce a formas más sencillas de diseñar algoritmos.

Ejemplo: Los semáforos funcionan mediante la repetición continua de la misma serie de acciones.

Pensamiento computacional

Formas de resolver problemas





¿Qué es el pensamiento computacional?

Pensamiento algorítmico

El pensamiento algorítmico es la habilidad para crear una serie ordenada de pasos con el propósito de resolver un problema.

Ejemplo uno: al cocinar una receta, seguimos ordenadamente una serie de pasos para preparar un plato.

Ejemplo dos: al jugar con un ordenador, podemos codificar una secuencia de acciones que le digan al equipo qué hacer.

Evaluación o valoración

Esta es la habilidad para verificar si un prototipo funciona correctamente y, si no es el caso, la habilidad para identificar qué debe mejorarse. También se trata del proceso por el que pasa un programador informático para encontrar y corregir errores en un programa.

Ejemplo uno: al cocinar, probamos constantemente la comida para comprobar si está correctamente condimentada.

Ejemplo dos: cuando buscamos errores ortográficos y de puntuación en una redacción estamos valorándola para que pueda leerse correctamente.

Abstracción

La abstracción es la habilidad para explicar un problema o una solución omitiendo los detalles poco importantes. En otras palabras: ser capaz de conceptualizar una idea.

Ejemplo: Al describir una bicicleta, solo incluimos algunos detalles. Puede que mencionemos el color o el tipo de bicicleta que es y añadir más detalles para alguien a quien le interesen las bicicletas.



Un proceso para desarrollar habilidades de pensamiento computacional

Uso de un proceso de diseño técnico

A la hora de buscar soluciones para un problema, los ingenieros recurren a un proceso de diseño. Pasan por una serie de fases que les guían hasta dar con una solución. Durante cada una de las fases utilizan y desarrollan algunas de sus habilidades. Estas son las habilidades a las que hacemos referencia cuando hablamos de “habilidades de pensamiento computacional”.

En WeDo 2.0, los estudiantes siguen un proceso similar:

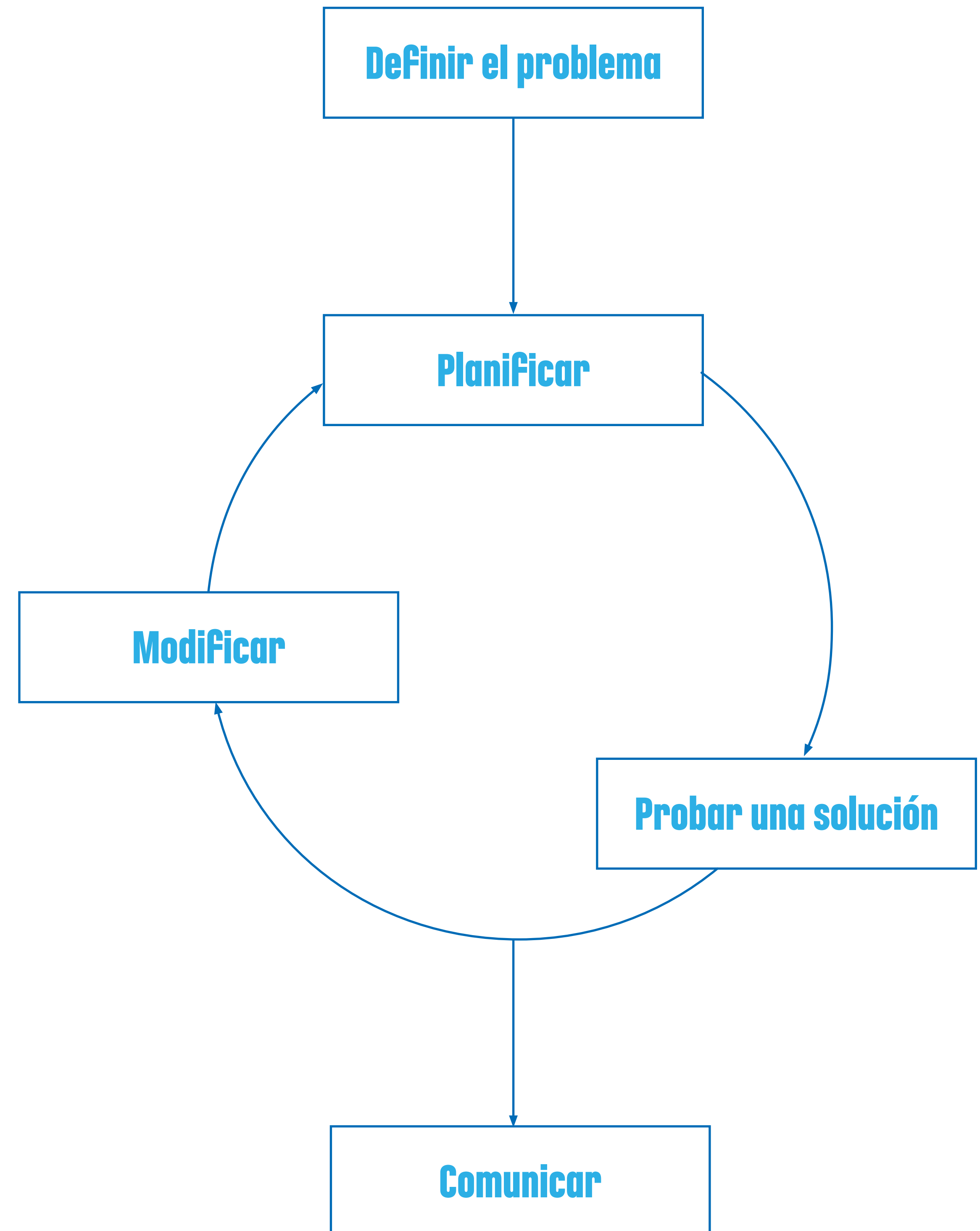
Definición del problema

Se presenta un tema a los alumnos que les conducirá a un problema o una situación que les gustaría mejorar. A veces un problema puede tener muchos detalles. Para resolverlo más fácilmente, el problema puede descomponerse en partes más pequeñas.

Mediante la definición del problema de una manera sencilla y la identificación de algunos criterios de éxito, los estudiantes desarrollarán una habilidad llamada “descomposición”.

En otras palabras:

- ¿El estudiante es capaz de explicar el problema por sí mismo?
- ¿El estudiante es capaz de describir cómo evaluará si ha tenido éxito la resolución del problema?
- ¿El estudiante es capaz de descomponer el problema en partes más pequeñas y manejables?





Un proceso para desarrollar habilidades de pensamiento computacional

Planificación

Los estudiantes deberían invertir tiempo en imaginar diferentes soluciones para el problema y elaborar un plan detallado para ejecutar una de sus ideas. Definirán los pasos que tendrán que llevar a cabo para llegar a la solución. Mediante la identificación de las partes de la tarea que posiblemente hayan visto anteriormente, los estudiantes desarrollarán una habilidad llamada “generalización”.

En otras palabras:

- ¿El estudiante es capaz de elaborar una lista de acciones para programar?
- ¿El estudiante es capaz de identificar partes de programas que podría usar?
- ¿El estudiante es capaz de reutilizar partes de programas?

Prueba

A continuación, se le encarga a los alumnos que creen la versión definitiva de su solución. En esta fase del proceso utilizan un lenguaje de programación basado en iconos para activar sus modelos LEGO®. Dado que los estudiantes codifican sus ideas, desarrollan sus habilidades de pensamiento algorítmico.

En otras palabras:

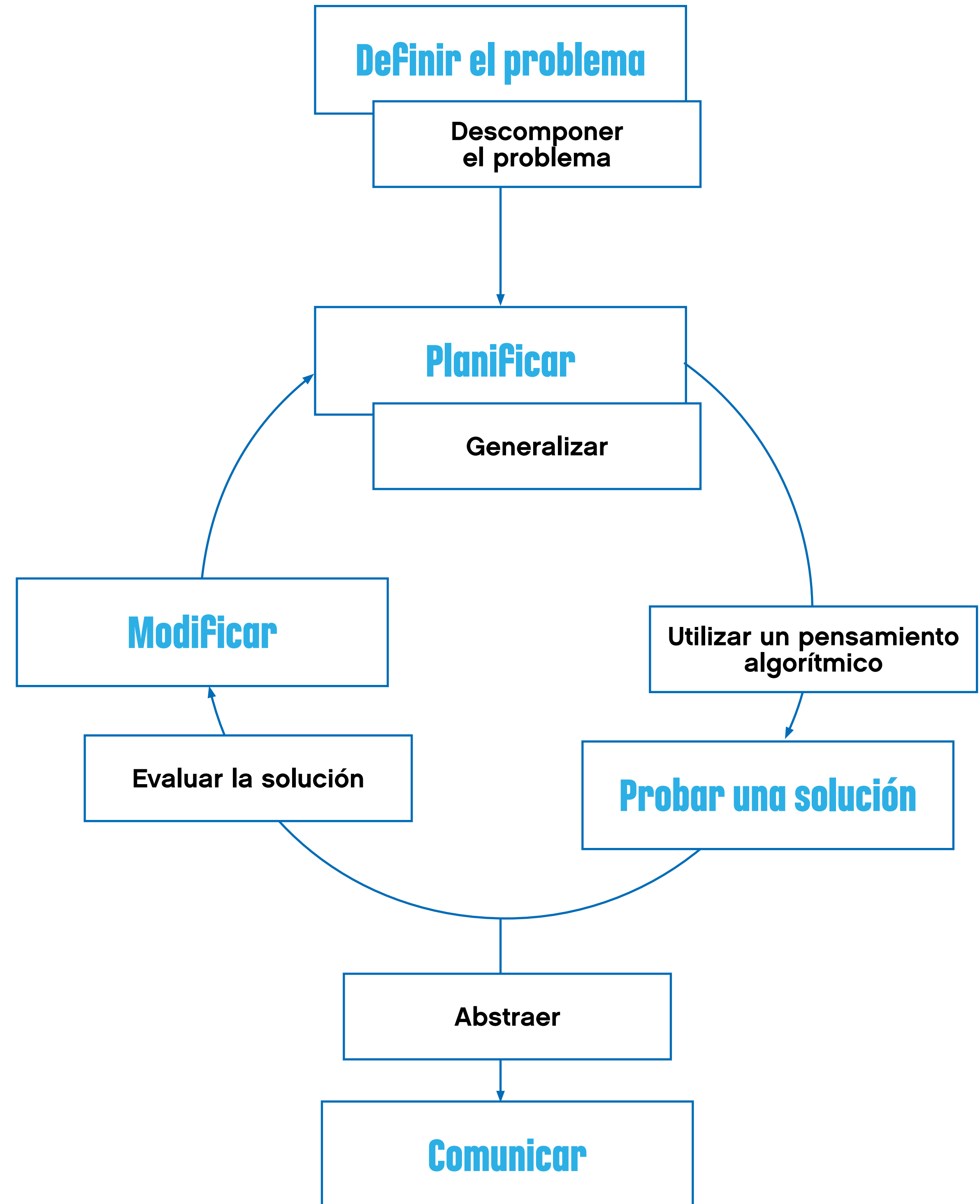
- ¿El estudiante es capaz de programar una solución para un programa?
- ¿El estudiante es capaz de usar secuencias, bucles, declaraciones condicionales, etc.?

Modificación

Los estudiantes evaluarán sus soluciones según si sus programas y sus modelos cumplen con los criterios de éxito. Haciendo uso de sus habilidades de evaluación, determinarán si tienen que cambiar, reparar, depurar o mejorar algunas partes de sus programas.

En otras palabras:

- ¿El estudiante realiza iteraciones de su programa?
- ¿El estudiante repara los problemas de su programa?
- ¿El estudiante es capaz de juzgar si la solución está relacionada con el problema?





Un proceso para desarrollar habilidades de pensamiento computacional

Comunicación

Los estudiantes presentarán a la clase la versión final de su solución y explicarán de qué manera cumple los criterios de éxito. Mediante la explicación de su solución con el nivel de detalle adecuado, desarrollarán sus habilidades comunicativas y de abstracción.

En otras palabras:

- ¿El estudiante explica la parte más importante de la solución?
- ¿El estudiante aporta los detalles necesarios para mejorar la comprensión?
- ¿El estudiante se asegura de explicar cómo cumple su solución con los criterios de éxito?





Desarrollo del pensamiento computacional mediante codificación

Para desarrollar su pensamiento algorítmico, los estudiantes se familiarizarán con algunos principios de programación. A medida que desarrollan sus soluciones, organizarán una serie de acciones y estructuras que darán vida a sus modelos.

Los principios de programación más comunes de WeDo 2.0 que usarán los estudiantes son:

1. Salida

Una salida es algo que puede controlarse mediante un programa escrito por los estudiantes. Algunos ejemplos de salidas para WeDo 2.0 son la activación y la desactivación de sonidos, luces, pantalla y motores rotatorios.

2. Entrada

Una entrada es una información recibida por un ordenador o un dispositivo. Puede introducirse mediante el uso de sensores en forma de valor numérico o textual. Por ejemplo, un sensor que detecta o mide algo (como la distancia) convierta este valor en una señal de entrada digital para que pueda usarse en un programa.

3. Eventos (a la espera de)

Los estudiantes pueden indicar a su programa que espere a que ocurra algo antes de continuar la secuencia de acciones. Los programas pueden esperar un periodo de tiempo concreto o esperar a que el sensor detecte algo.

4. Bucle

Los estudiantes pueden programar acciones que se repitan para siempre o durante un periodo de tiempo específico.

5. Funciones

Las funciones son un grupo de acciones que debe usarse conjuntamente en situaciones específicas.

Por ejemplo, el grupo de ladrillos que podría utilizarse para crear un parpadeo de luz se llamaría “la función de parpadear”.

6. Condiciones

Los estudiantes usan las condiciones para programar acciones que sólo deben ejecutarse en determinadas circunstancias. Crear condiciones en un programa implica que alguna parte de este no se ejecutará jamás, a menos que se cumplan las condiciones. Por ejemplo, si el sensor de inclinación se inclina hacia la izquierda, el motor arrancará; si el sensor se inclina hacia la derecha, el motor se detendrá. Si el sensor de inclinación no se inclina nunca hacia la izquierda, el motor nunca arrancará y, si nunca se inclina hacia la derecha, el motor nunca se detendrá.

