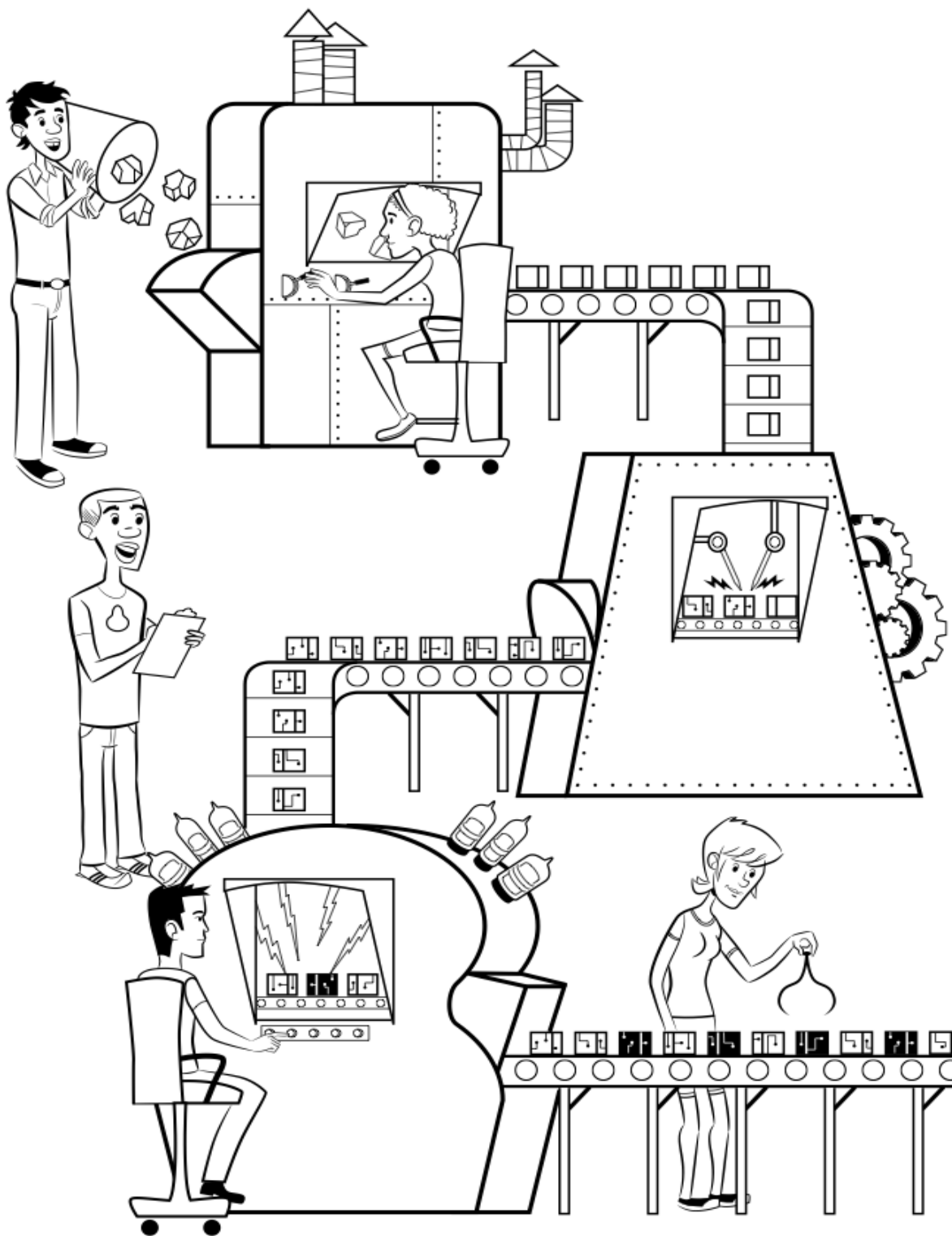




Una guía de dibujos animados para solucionar problemas,
diseñar sistemas y comprender el comportamiento humano.

Un proyecto de
Benjamin Chun
&
Tim Piotrowski

<http://www.ctillustrated.com/>
<http://www.eduteka.org/PensamientoComputacionalIlustrado.php>



Analizar los efectos de la computación

Práctica #1 del Pensamiento Computacional: Analizar los efectos de la computación

La computación está en todas partes. Desde los motores de búsqueda que nos ayudan a encontrar información, pasando por las cajas registradoras de los mercados, para llegar al software que se usa para diseñar puentes, vivimos en un mundo construido con los resultados de la computación.

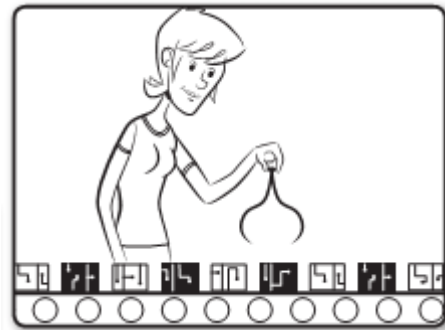
La computación no es simplemente una palabra diferente para referirse a la tecnología. Por ejemplo, un teléfono celular contiene muchas tecnologías diferentes; un transmisor y receptor de radio, un procesador, memoria y partes electromecánicas como botones y pantallas táctiles. Cuando estudiamos los efectos de la computación, no estamos tratando de aprender cómo la física gobierna estas tecnologías. Analizar los efectos de la computación específicamente quiere decir observar lo que sucede cuando recopilamos, guardamos y procesamos datos.



La computación realizada por un teléfono celular incluye grabar su voz como dato, comprimir y transmitir ese dato e interactuar con un sistema mayor que direcciona los datos de su llamada para que lleguen a su destino. Este mismo proceso computacional se hace a la inversa de manera que su compañero de conversación pueda contestarle. Esto parece mucho trabajo de computación para que lo realice su celular, pero lo anterior solo es una parte de lo que sucede cuando usted hace una llamada.

Todo el envío y la recepción de datos se sucede por medio de ondas de radio. Cuando Nikola Tesla inventó la tecnología de

radiocomunicación, esta se podía usar solamente para comunicación masiva en forma de emisoras. La intervención de la computación transformó esa incipiente capacidad tecnológica en la forma más refinada que utilizamos hoy en nuestros teléfonos celulares. Uno de los efectos de la computación es que actualmente el radio puede usarse para comunicarse entre personas, con muchas conversaciones simultáneas que suceden en la misma área física.



Cuando analizamos los efectos de la computación, anotamos y medimos cómo se transforman los datos. Observamos cómo se procesa la información y que se logra con ese procesamiento. Podemos reflexionar lo que sucedería si ese poder de computación no estuviera disponible. También comenzamos a imaginar cosas nuevas que podríamos lograr usando la computación.

Parte considerable del trabajo de analizar los efectos de la computación es la observación cuidadosa, como lo están haciendo en esta ilustración Blas, Ada, Carlos, Alan y Gracia. En su mundo, como en el nuestro, la computación está en todas partes. Si miramos más detenidamente, comenzaremos a ver lo que la computación, no la tecnología hecha por otros, hace por nosotros.



Producir artefactos computacionales

Práctica # 2 del Pensamiento Computacional

Producir artefactos computacionales

Cuando hablamos de crear artefactos computacionales nos referimos a hacer cosas. Programar es una de las formas más visibles en las que hacemos artefactos computacionales. En ese caso, los artefactos son tanto los programas que construimos como los resultados obtenidos con estos. Pero el término artefacto computacional no se limita a los programas de computador. Puede referirse a una gran cantidad de cosas que abarcan desde microprocesadores a códigos de barras o al sistema de navegación de un avión.

En esta ilustración los personajes están construyendo, probando y explorando artefactos computacionales. El proceso de crear no está limitado a la mera producción de ideas o de piezas de ensamble. Las maquinas que se ven en estas caricaturas son simbólicas, diseñadas para diversas interpretaciones o para imaginar. A continuación encuentre algunas maneras de observarlas para ayudarle a comenzar:

Gracia está creando algo nuevo. En el momento usa una llave por ser la herramienta correcta para el trabajo que está realizando. Ella no está simplemente utilizando una máquina que otro creó; está haciendo algo nuevo. A veces crear cosas es un proceso no solo difícil sino demandante en tiempo, pero mejora con la experiencia.



resolver un problema complejo, como el de Alan que está parado de cabeza! Muchos conceptos computacionales, como la idea de la cinta de Moebius, pueden retar nuestras creencias de lo que es posible y revelarnos verdades más profundas sobre las propiedades de los sistemas que estamos creando o utilizando. Inicialmente esto puede parecer tan difícil como caminar parado de cabeza por el cielo raso, pero después de un tiempo posiblemente lo encontremos entretenido.



Blas usa un guante que controla una mano más grande y fuerte. Esa mano puede hacer muchas cosas, incluso levantar a Blas. El guante que él usa es una metáfora para artefactos computacionales que nos permiten utilizar todo el poder de las máquinas para realizar gran cantidad de cálculos. Cuando redirigimos ese



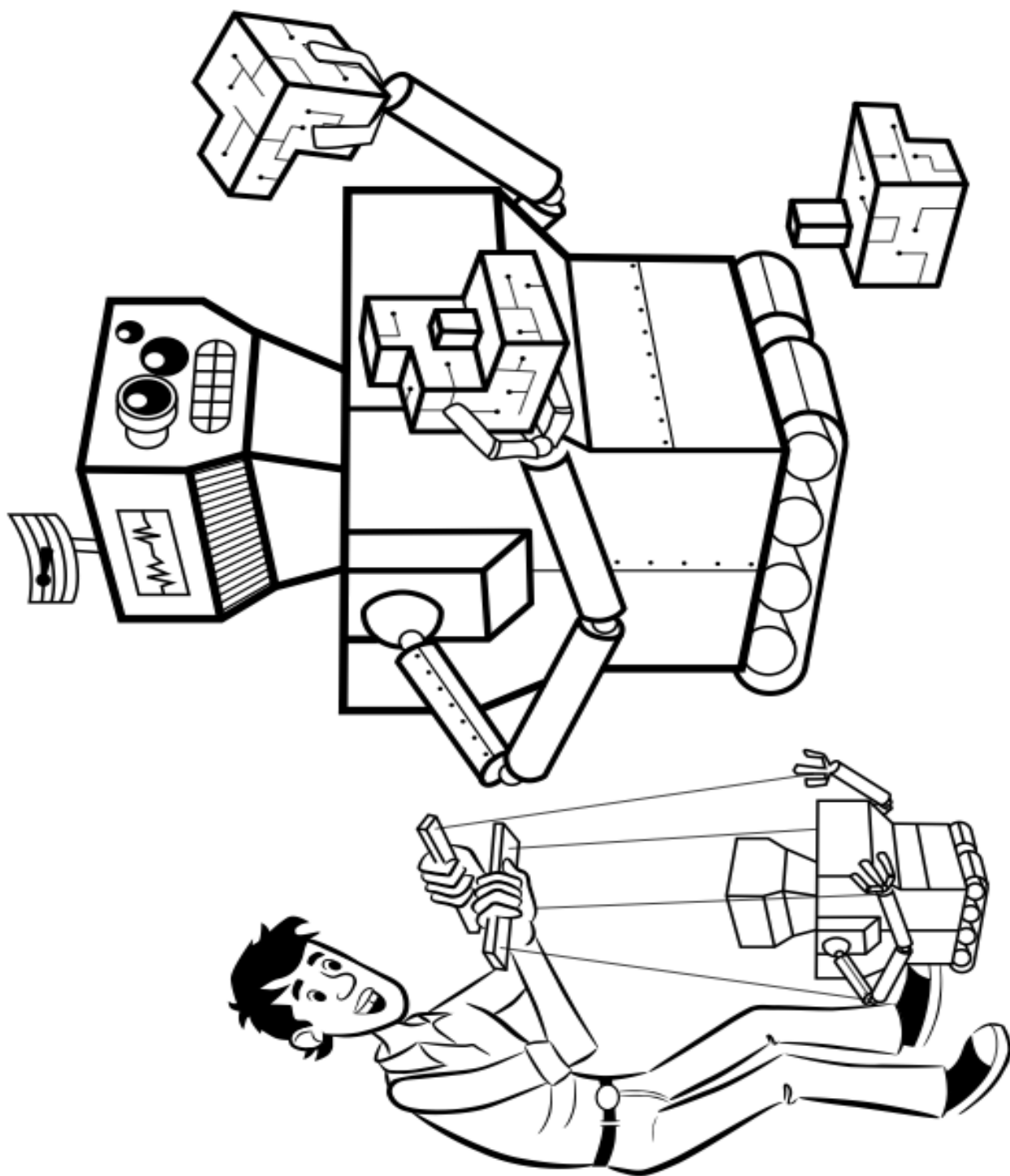
poder hacia sí mismo, como sucede cuando usamos recursión, funciones de orden superior o escribir un compilador para un lenguaje en ese mismo lenguaje, lo que sucede puede ser muy emocionante.

Alan está caminando por el cielo raso. Sostiene una cinta de Moebius, una superficie topológica de un solo lado. Cuando se voltea o se rota para pegarla a ella misma, un rectángulo plano regular puede transformarse en una cinta de Moebius. Usando el Pensamiento Computacional, podemos cambiar nuestra perspectiva para

Carlos está sosteniendo un globo terráqueo cubierto con lo que parecen pequeños discos de radio. Los artefactos computacionales no necesitan diseñarse para que trabajen aislados. Pueden trabajar juntos y comunicarse para realizar una tarea, como vemos en los procesadores de varios núcleos o en la computación en paralelo. Quizá los discos de radio le ayudan a Carlos a oír cosas que los otros personajes no oyen. Se igual manera, los algoritmos para reconocer patrones, el procesamiento de señales, la corrección de errores y la reducción de ruido, acrecientan



nuestra habilidad para extraer información de los datos. Con la ayuda de artefactos computacionales, adquirimos nuevos poderes.



El uso de la abstracción y de los modelos

Práctica # 3 del Pensamiento Computacional

El uso de la abstracción y de los modelos

“Todos los modelos están equivocados, pero algunos son útiles”

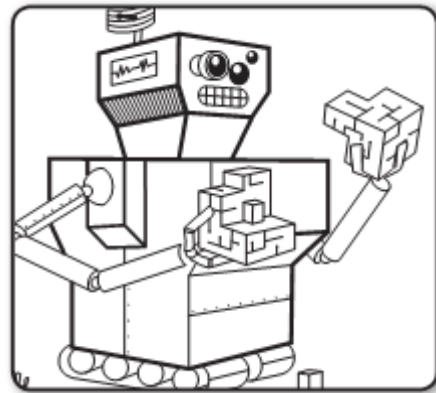
--Geroge E. P. Box

Uno de los significados que tiene la palabra modelo es: Una versión más pequeña o simple del artículo original. El modelo puede ser un objeto físico como el pequeño robot que aparece en la ilustración. Observe que Blas no está tratando de mover los brazos del robot grande, ni de mover él los bloques grandes. En cambio está tratando de trabajar con un robot modelo lo suficientemente pequeño que le permita asirlo con sus propias manos. Esto es, simplificar el trabajo físico que necesita hacer, de la misma manera que el modelo simplificado de una idea facilita pensar.



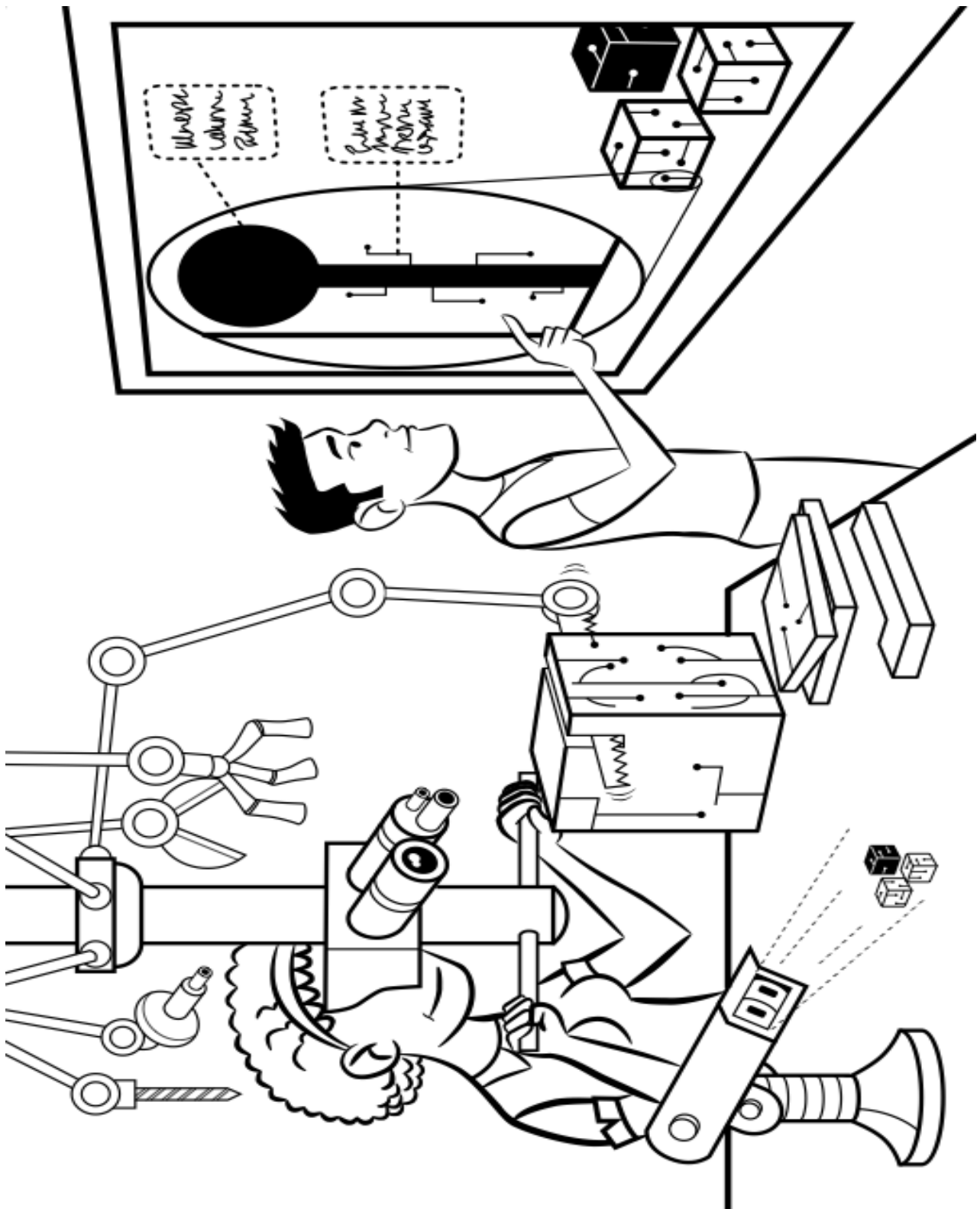
Por ejemplo, la mecánica clásica es un modelo: Es la aproximación a la computación sencilla de Newton de la realidad más compleja del movimiento. En ciencia de la computación, cada vez que escribimos un programa, hacemos un modelo. Debemos escoger la información y el nivel de detalle que representaremos en nuestro programa. Algunos detalles deben dejarse por

fuera. Si en un modelo o programa tratamos de incluirlo todo, ¡terminaremos simulando el mundo entero!



En un sistema complejo, debemos utilizar muchos y muy diferentes modelos y lograr que funcionen juntos. No debemos preocuparnos si una parte del sistema se desconecta y se reemplaza por otra cosa que puede cumplir el mismo objetivo. Podemos decir que abstraemos esa parte del sistema. Seleccionar cuidadosamente las cualidades que más nos interesan e ignorar el resto de detalles es la clave de la abstracción. Cuando deliberadamente separamos nuestro sistema en partes que pueden entenderse, probarse, reusarse y reemplazarse individualmente, estamos creando nuevas abstracciones.

Ver además: Modularidad, relación mapa – territorio, marioneta.

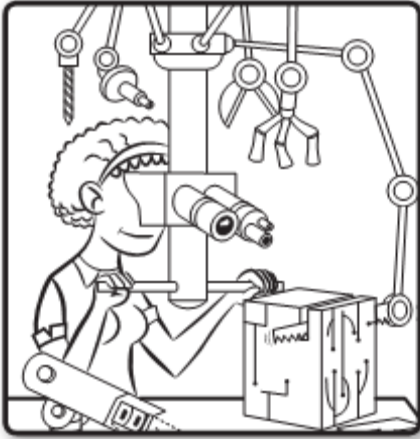


Analizar problemas y artefactos

Práctica # 4 del Pensamiento Computacional

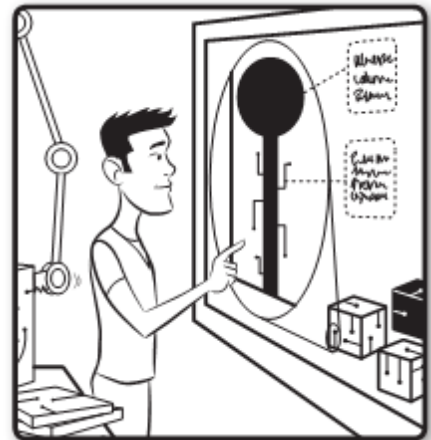
Analizar problemas y artefactos

Dice Wikipedia, “Analizar es el proceso de partir o descomponer un tema o sustancia complejos en partes más pequeñas para poder entenderlos mejor”



En esta ilustración, **Ada** usa una herramienta con muchos acoples, para representar la idea de que con frecuencia necesitamos probar múltiples enfoques y diferentes herramientas, antes de “dar” con la solución del problema. Diferentes problemas y diferentes maneras de enfocarlos tienen debilidades distintas. Con frecuencia no podremos solucionar un problema hasta que ensayemos diferentes vías para descomponerlo. Por eso, cuando se trabaja en un problema, es tan valioso tener a disposición una variedad de herramientas conceptuales.

Hacia la derecha, **Alan** controla una vista de acercamiento de los cubos que hay en la mesa. Esto le permite ver y entender desde el exterior no solo cómo se ve y se comporta el cubo, sino además, cómo sus componentes internos contribuyen a su comportamiento general. Los programadores realizan este tipo de análisis cuando realizan una depuración, hacen lo mismo los ingenieros eléctricos cuando utilizan un osciloscopio para visualizar señales.





Comunicar procesos y resultados

Práctica # 5 del Pensamiento Computacional

Comunicar procesos y resultados

Rara vez un artefacto computacional se explica por sí mismo. Una CPU hecha con transmisores microscópicos en silicio o un programa de compilación binaria de unos (1) y ceros (0), son ambos bien difíciles de entender. Sus formas se optimizan para el desempeño computacional no para la comprensión humana. El plan de diseño para la CPU o el código fuente para el programa son más fáciles de entender. Pero aún estos precursores no necesariamente explican cómo fueron hechos o por qué funcionan.

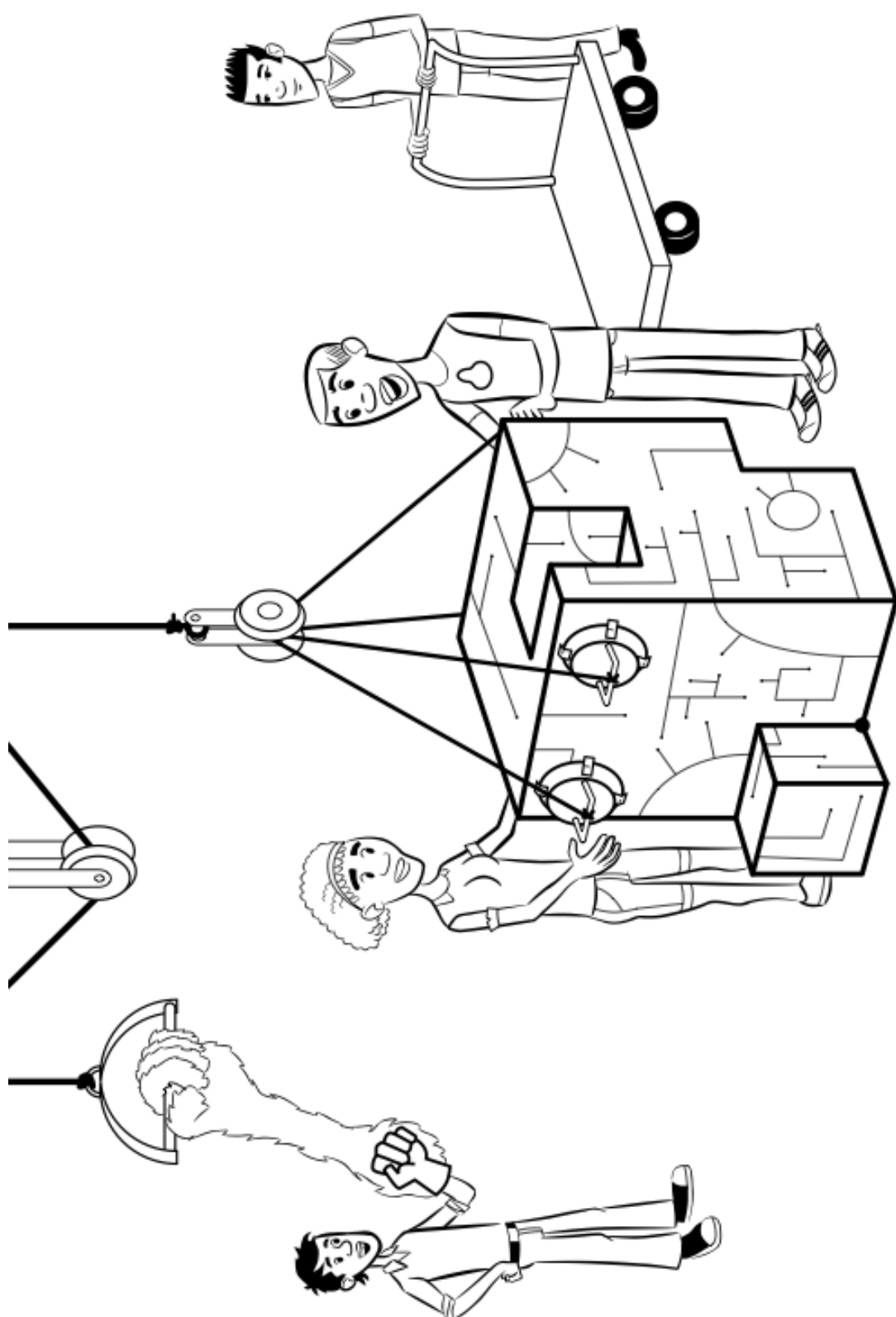
El pensamiento computacional requiere que discutamos procesos que deben atender tanto las personas como las máquinas, además de la intención de estos procesos de producir resultados específicos. Por ejemplo, cuando un programador está aprendiendo a escribir programas, se le debe enseñar a depurar imprimiendo el valor de una variable. Cuando usted descubre una nueva técnica matemática para manipular eficientemente formas en 3D, debe escribirla de manera que otras personas puedan entenderla y utilizarla. La comunicación es la forma en que contamos al mundo nuestros conocimientos.

Cuando usamos la computación para resolver un problema, la respuesta que obtenemos no necesariamente tiene significado automático para otros. Debemos comunicar ese resultado de manera que revele tanto su importancia como su origen.



En la ilustración, Carlos está capturando los sonidos que produce un loro en su hábitat natural y los está transmitiendo a Gracia que se encuentra en otro lugar. Esto lo podemos interpretar literalmente como una comunicación de datos de audio, similar a la voz de alguien durante una llamada telefónica. Sin embargo, para otro loro, el canto del primero representa un proceso (como una técnica que se usa para encontrar frutas y semillas, o el plan para una migración estacional) o algunas noticias importantes (como el ganador del concurso anual de discursos). La comunicación sobre procesos y resultados nos permite beneficiarnos de los descubrimientos hechos por otros pensadores computacionales.





El trabajo efectivo en equipos

Práctica # 6 del Pensamiento Computacional

El trabajo efectivo en equipos

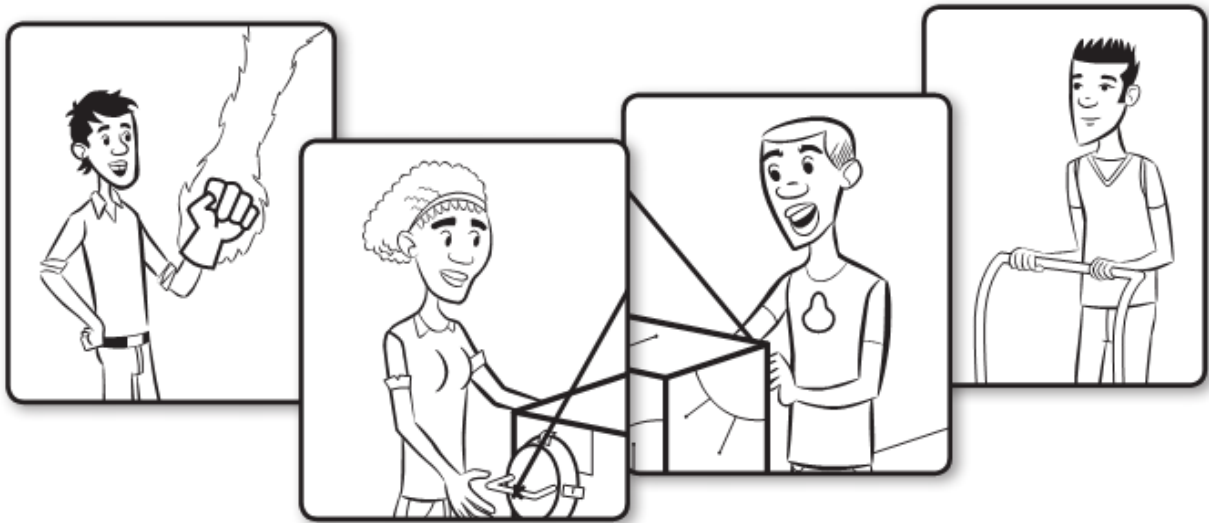
La habilidad para trabajar dentro de un equipo puede ser el diferenciador entre el éxito y el fracaso. Construir cualquier sistema complejo, software o hardware, requiere que se haga más trabajo en menos tiempo de lo que cualquier persona puede llevar a cabo. Pero aumentar el número de personas no necesariamente significa que el trabajo estará listo más rápido.

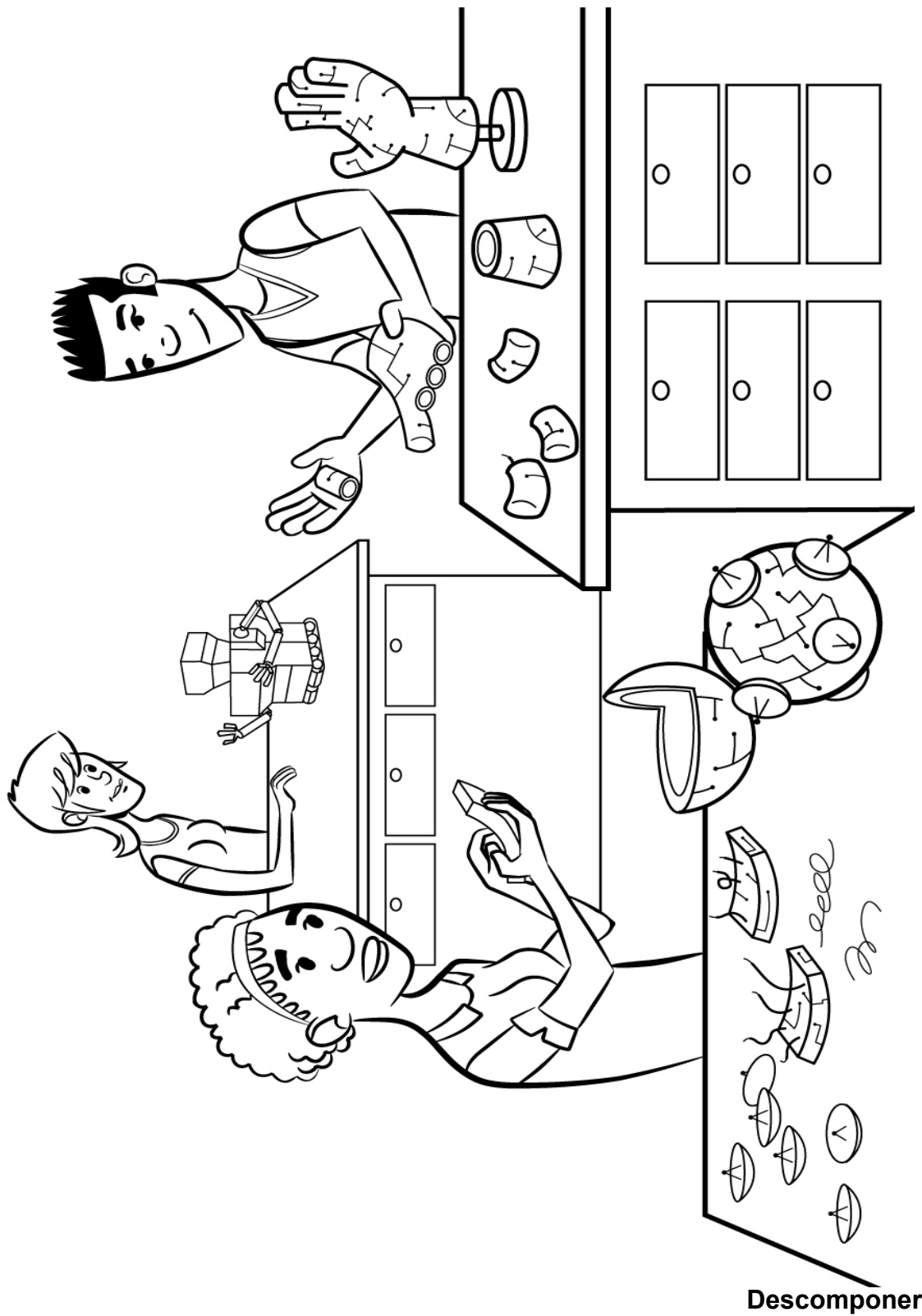
Para que el trabajo en equipo verdaderamente sea efectivo, los individuos necesitan habilidades interpersonales y de comunicación, así como conocimiento sobre diferentes metodologías y procesos grupales. A medida que el equipo aumenta su tamaño, el papel de la cultura y la administración se vuelven cada vez más importantes. El trabajo en equipo, como cualquier otra habilidad, requiere práctica.

Diversas estrategias para repartir el trabajo tienen distintas fortalezas y debilidades. Planear la mejor manera de trabajar juntos no siempre es fácil, pero es importante para el pensamiento computacional.

En la medida en que los procesadores de varios núcleos y la computación distribuida se vuelven más comunes, veremos a los computadores mismos trabajar en equipos. A la mayoría de los sitios Web que usted visita los atienden centros de datos, en los que cientos o miles de computadores individuales trabajan juntos para realizar tareas sorprendentes. ¡Nosotros los humanos podemos hacer lo mismo!

Ver además: Ley de Brooks, programación en pareja, sistema de control de versiones.





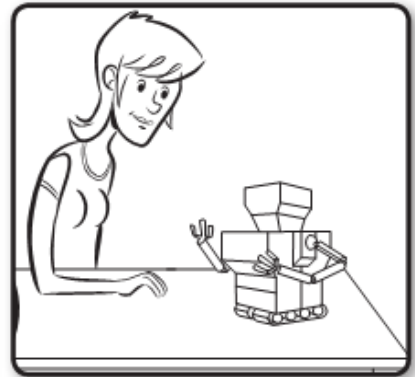
Descomponer

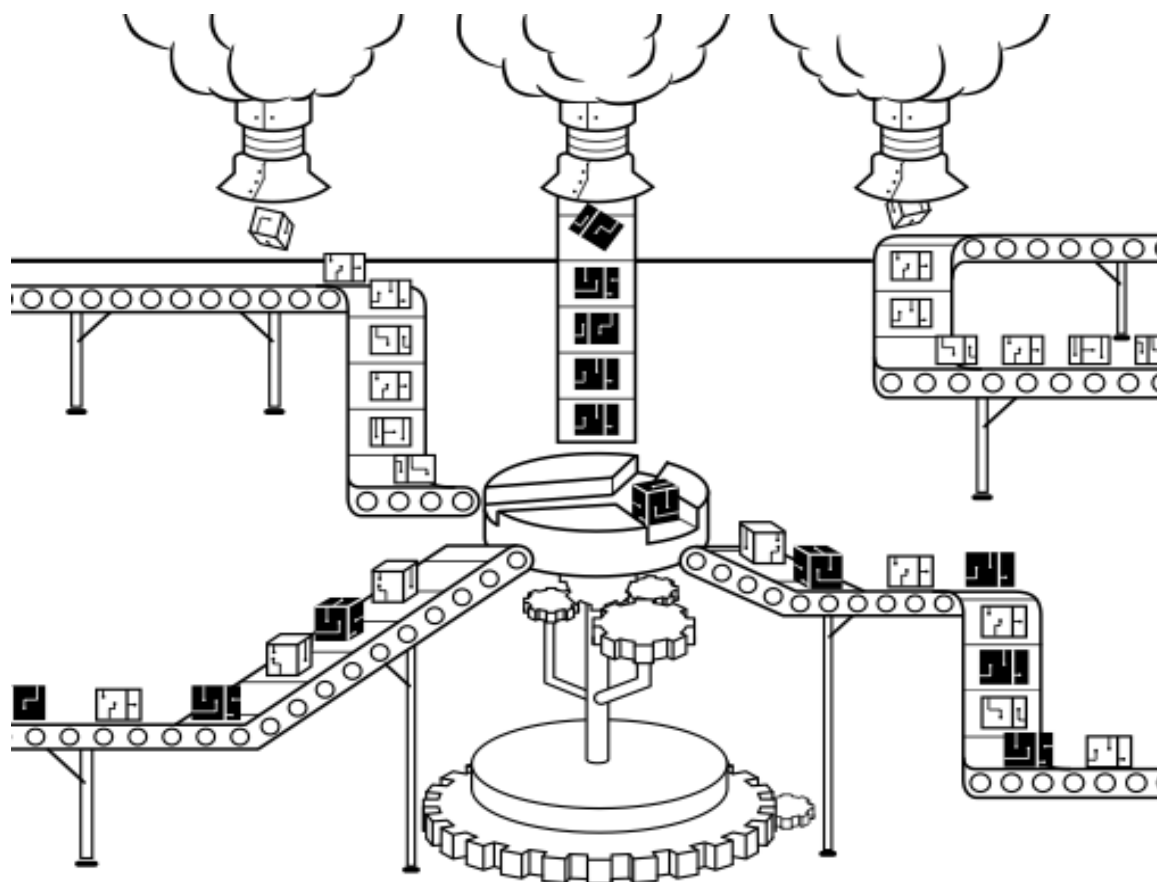
Explorando el pensamiento computacional # 1

Descomponer

En ésta ilustración, **Ada, Alan y Gracia**, están cada uno descomponiendo algunas de las máquinas que hemos visto en otras escenas. Pero descomponer no es solamente desarmar objetos. También es desagregar los pasos de un proceso. Muchas de las cosas que pensamos requieren una única acción, realmente están compuestas por muchas pequeñas acciones. Por ejemplo, decimos que vamos a preparar la comida. Pero cuando a esto le aplicamos la descomposición, encontramos que preparar la comida significa realmente, abrir la nevera, sacar las verduras, cortar la cebolla, prender la estufa y muchos otros pequeños pasos.

Un problema computacional difícil puede muchas veces resolverse si pensamos en él como si fuera una tarea compuesta por muchas otras más pequeñas y sencillas. Descomponer incluye identificar esas pequeñas tareas y cómo encajan entre ellas. Mientras más veces se realice este proceso, más fácil se vuelve. Pregúntele a Ada, que está desarmando un globo terráqueo. Aunque los globos terráqueos tienen ligeras diferencias, ella tiene una idea bastante buena de las piezas que va a encontrar al desarmar uno.





Reconocimiento de patrones

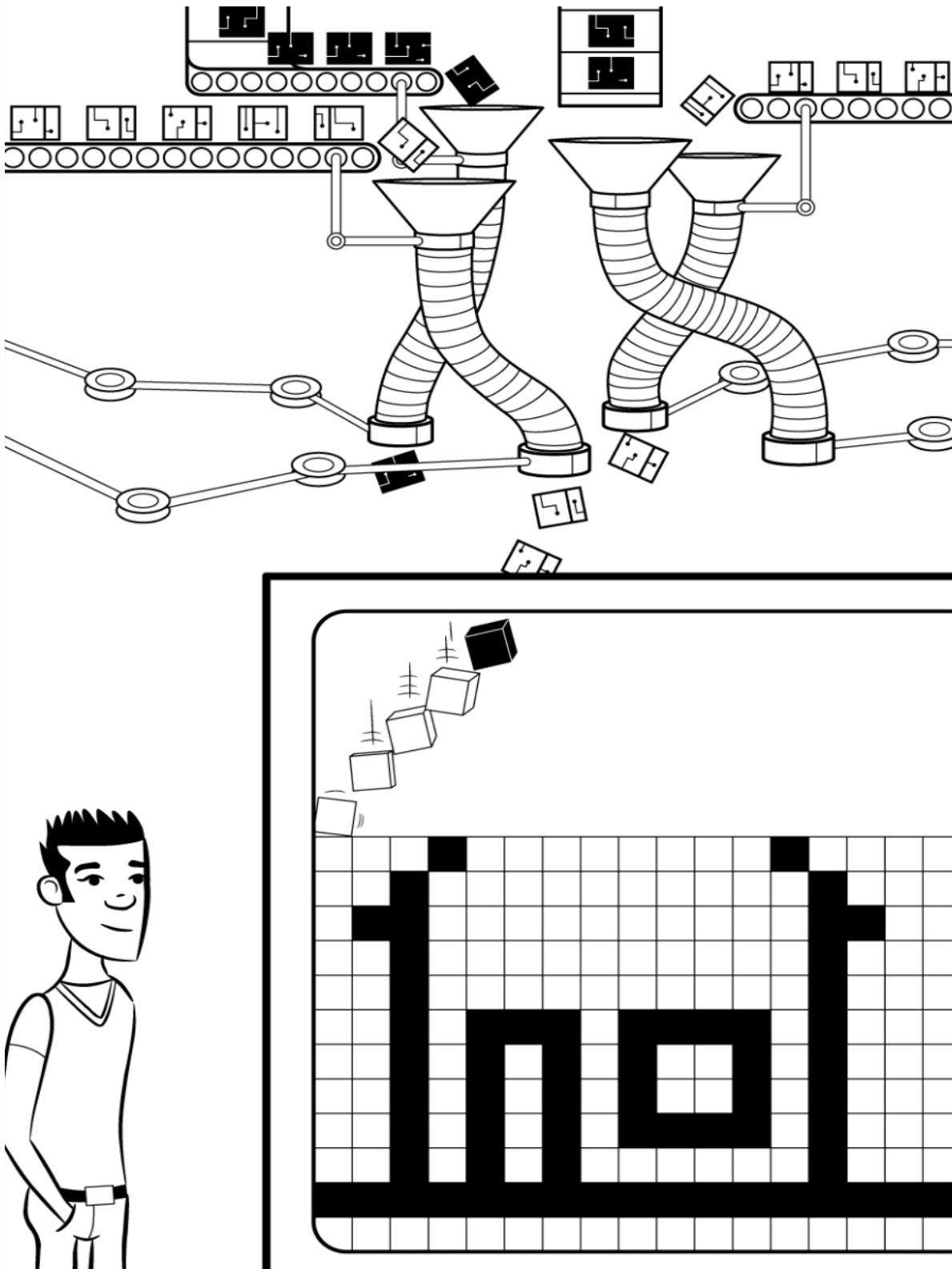
Explorando el pensamiento computacional # 2

Reconocimiento de patrones

Hay algo extraño respecto al patrón de estos bloques y **Gracia** se lo está señalando a **Ada**. Aunque no están mirando toda la compleja máquina que produce este patrón de bloques, pueden identificar lo que es inusual. Esto no significa que algo esté mal, pero les indica que pueden estar sucediendo más cosas de las que ellas inicialmente pensaron.

Hacerse una idea de lo que usted espera, es una manera de encontrar un patrón. Mientras más observe usted, más patrones encontrará en la naturaleza, en artefactos computacionales y en procesos. Cuando reconocemos un patrón, podemos utilizar nuestras otras habilidades de pensamiento computacional para ayudarnos a entender su importancia.





Generalización de patrones y abstracción

Explorando el pensamiento computacional #3

Generalización de patrones y abstracción

Después de haber visto varias veces el mismo patrón, puede comenzar a pensar en diferentes maneras de describirlo.

Alan está viendo cómo algunos bloques caen en determinado sitio para formar un dibujo. Si la máquina suelta nuevamente el mismo patrón de bloques, harán el mismo dibujo.

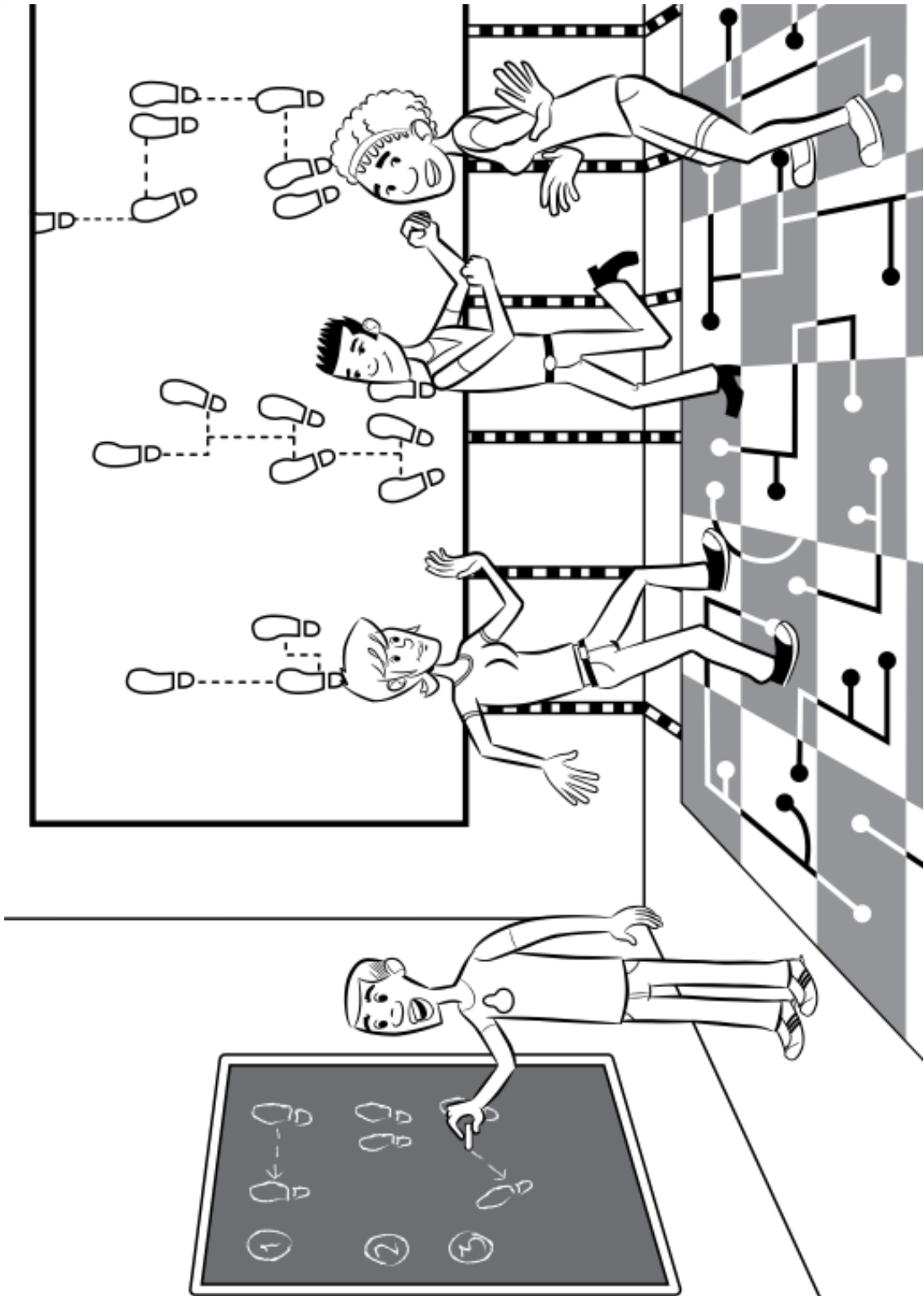
Viendo caer los bloques, **Alan** tiene mucho para pensar al respecto. Hay muchísimos patrones posibles – mire si puede calcular la cantidad. Existen también muchas formas para describir éstos patrones.

Si queremos que la máquina haga el dibujo de una casa con la puerta ubicada al lado derecho y no al izquierdo, la

instrucción será muy similar. ¿Qué sucedería si en lugar de darle cada vez a la máquina nuevas instrucciones, le dijéramos solamente lo qué debe cambiar en algunas de ellas?

Necesitaríamos instrucciones que describieran cómo hacer cambios en otras instrucciones.

Pensar de esta manera indica algo sobre el trabajo que realizamos cuando tratamos de generalizar patrones. Buscamos lo que es igual en un grupo de patrones y tratamos entonces de describirlo de tal forma que sea a la vez clara y eficiente. Si de una vez podemos describir todo el grupo de patrones, un patrón de patrones, tendremos entonces una abstracción.



Diseño del algoritmo

Explorando el pensamiento computacional # 4

Diseño del algoritmo

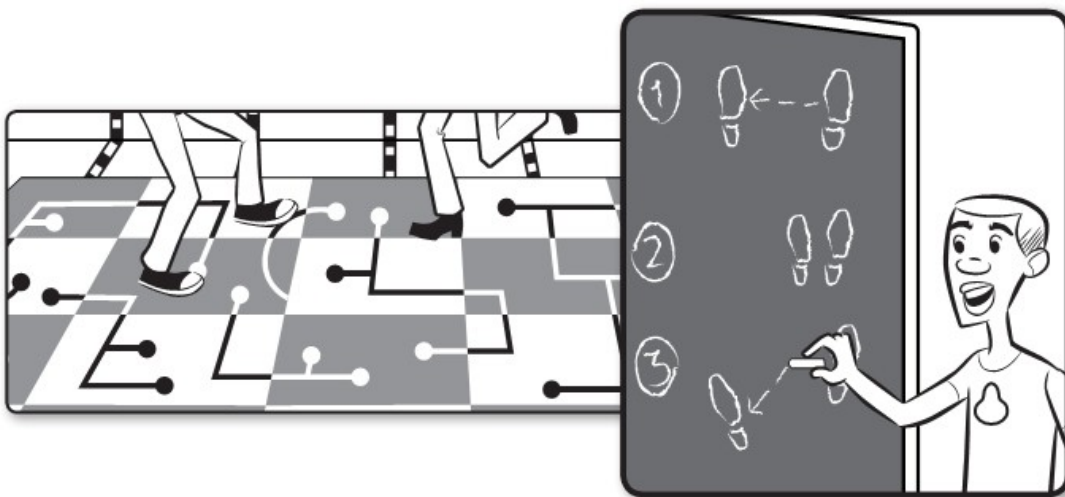
¡Estamos en una fiesta bailable que tiene como tema el pensamiento computacional! El piso sobre el que se baila, y que se muestra en la ilustración, puede tanto grabar los pasos de los danzantes como iluminarse con algunas instrucciones de baile.

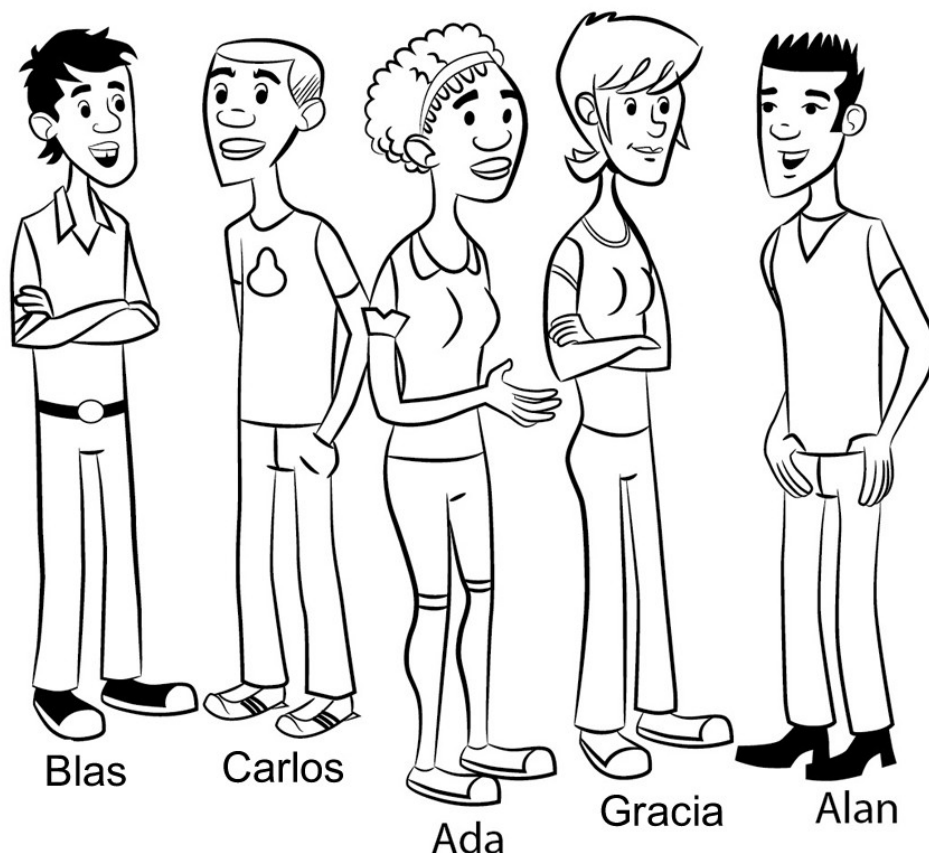
Pero mientras **Gracia**, **Alan** y **Ada** bailan libremente, **Carlos** está diseñando un baile nuevo. Tal como en el caso de un algoritmo, un baile tiene una serie o conjunto de pasos que otros pueden seguir, para alcanzar el mismo resultado.

A veces pensamos que los algoritmos se escriben como un programa de computador, pero un algoritmo se asemeja más a una idea. El mismo algoritmo puede escribirse en muchos lenguajes de computador diferentes. Son

los pasos que se siguen en el proceso los que hacen que un algoritmo sea lo que es.

Para diseñar un algoritmo o un baile, usted necesita comprender su objetivo y las restricciones del sistema. Los seres humanos solo tienen dos pies, así que los bailes diseñados para personas tienen que funcionar con esa limitación. Los sistemas computacionales tienen diferentes tipos de limitaciones, tales como la velocidad del procesador o la capacidad de la memoria o la cantidad de electricidad que estos consumen. Diseñar un algoritmo que realice tareas específicas dentro de las restricciones del sistema es como crear un baile atractivo que todo el mundo quiere aprender.





Estos son los niños del pensamiento Computacional Ilustrado! A ver si puedes averiguar qué pensadores computacionales famosos llevaron sus nombres.

CRÉDITOS:

Traducción al español por Eduteka¹ del documento “Counputational Thinking Illustrated”² elaborado por Benjamin Chun³ y Tim Piotrowski⁴.

Todos los dibujos de Pensamiento Computacional Ilustrado están disponibles en formato vectorial, lo que significa que se ven bien en cualquier tamaño. Este proyecto tiene licencia “Creative Commons BY-SA”⁵. Usted es libre de usar estos dibujos animados: imprimirlos como carteles gigantes o folletos, remezclarlos para adaptarlos a sus necesidades, o usarlos en un libro de texto. Los únicos requisitos son que se cite la fuente y se comparta la nueva creación bajo la misma licencia.

¹ <http://www.eduteka.org/PensamientoComputacionalIlustrado.php>

² <http://www.ctillustrated.com/>

³ <http://benchun.net/>

⁴ <http://glitchworks.com/>

⁵ <http://creativecommons.org/licenses/by-sa/3.0/>