

Introduciendo BlocksCAD como recurso didáctico en matemáticas

Pablo Beltrán-Pellicer
Carlos Rodríguez-Jaso
José María Muñoz-Escolano

SUMA núm. 93
pp. 39-48

Artículo recibido en *Suma* en diciembre de 2018 y aceptado en septiembre de 2019

BlocksCAD es un entorno web de programación visual por bloques, de aspecto similar a Scratch, para el software libre de modelado en 3D OpenSCAD. Con esta herramienta, los cuerpos en el espacio se describen mediante algoritmos que incluyen la utilización de unas pocas primitivas, como prismas y esferas; transformaciones geométricas y operaciones lógicas y conjuntistas. En este artículo se introduce el recurso y se proponen ejemplos de tareas que se han puesto en práctica con alumnado de segundo ciclo de ESO.

Palabras clave: Recurso didáctico, Pensamiento computacional, Geometría.

El interés en utilizar herramientas tecnológicas para la enseñanza y aprendizaje de las matemáticas y, específicamente, de la geometría, se puede rastrear desde la aparición del lenguaje LOGO y su célebre tortuga hasta el día de hoy, donde nadie cuestiona las sugerentes aplicaciones didácticas de los entornos de geometría dinámica, cuyo principal exponente es GeoGebra. Si bien la mayoría de las publicaciones existentes hasta ahora se refieren a programas de geometría dinámica plana, Gutiérrez y Jaime (2015) se

Introducing BlocksCAD as an educational resource in mathematics // BlocksCAD is a scratch-like programming environment to create 3D objects, which offers a web-based interface for the open-source OpenSCAD 3D modeling software. With this tool, bodies in space are described by algorithms that include the use of a few primitives, such as prisms and spheres; geometric transformations and logical and set operations. This article introduces the resource and proposes examples of tasks that have been implemented with students in the second cycle of secondary education in Spain.

Keywords: Educational resource, Computational thinking, Geometry.

hacen eco de la reciente aparición de programas de geometría dinámica tridimensional y cómo estos pueden suplir la carencia en los centros educativos de materiales manipulativos apropiados para el aprendizaje de contenidos de geometría espacial.

Por otro lado, hay ciertas herramientas tecnológicas sin una orientación a priori tan específica hacia las matemáticas, pero con un gran potencial educativo, que están encontrando rápidamente un hueco en los

centros educativos y que merecen nuestra atención. Entre ellas, se encuentran muchas de las que forman parte del flujo de trabajo de la fabricación e impresión en 3D. Estas máquinas han visto disminuir su coste de tal manera (Jones y otros, 2011) que ya son muchos los colegios e institutos que disponen de alguna para realizar actividades en asignaturas como Tecnología (González-Gómez y otros, 2012) y Matemáticas (Beltrán-Pellicer, 2017; Beltrán-Pellicer y Rodríguez-Jaso, 2017, 2018).

Una impresora 3D fabrica los objetos de forma aditiva. Esto quiere decir que, tras un proceso software de descomposición laminar, los imprime capa a capa, de abajo arriba. Dichos objetos pueden haber sido descargados de alguno de los repositorios que existen por Internet, como GitHub o Thingiverse en el formato habitual STL (que describe la superficie o mallado del objeto) o haber sido creados por el usuario con un software de diseño asistido por ordenador (CAD).

Dentro de los CAD existen múltiples opciones. Por un lado, hay entornos interactivos con mayor o menor grado de complejidad, tales como el 3D Builder de Microsoft, Rhinoceros 3D, Inventor, TinkerCAD (aplicación web gratuita) o la opción libre FreeCAD. En ellos se trabaja a partir de alguna perspectiva tridimensional, pudiendo establecer acotaciones y, en algunos casos, imponer restricciones o parametrizar piezas. Por otro lado, hay alternativas como OpenSCAD, que podría describirse como una herramienta que lee un fichero de texto con la descripción de los objetos 3D para generar el modelo y renderizarlo, de forma que en la pantalla se pueda ver su imagen. En otras palabras, con OpenSCAD se *programa* el modelo tridimensional a partir de unas primitivas que definen cuerpos básicos y una serie de transformaciones. Si en los entornos interactivos la parametrización aparece como una funcionalidad que proporciona un valor añadido, en OpenSCAD forma parte de su núcleo. Esto permite al diseñador efectuar cambios en cualquier paso del proceso de modelado, haciendo que los diseños sean fácilmente configurables. Pensemos, por ejemplo, en una rueda dentada. En OpenSCAD escribiríamos un programa que modelaría una rueda de un diámetro d , con n dientes.

Coincidimos con autores (Schelly y otros, 2015) en que OpenSCAD resulta sencillo de utilizar, y que realmente bastan 30 minutos de instrucción para aprender el manejo básico. También es verdad que hay algunos aspectos que requieren de una mayor profundización, y que no es trivial, por ejemplo, diseñar objetos complejos de forma recursiva.

El hecho de que OpenSCAD sea software libre ha posibilitado que surjan entornos de modelado basados en él. Es el caso de BlocksCAD, que simplemente añade una interfaz de programación visual por bloques similar a la del archiconocido Scratch. Recientes investigaciones señalan el potencial de la programación por bloques de Scratch como recurso didáctico en el aula de matemáticas en secundaria, tanto para la adquisición de contenidos matemáticos como de procesos propios de Pensamiento Matemático Avanzado (abstracción, generalización y formulación y validación de hipótesis) (Baeza-Alba, Claros-Mellado y Sánchez-Campaña, 2016; Ferrer, 2011) o para la mejora de la afectividad de los estudiantes hacia las matemáticas (Jorge-Pozo, Jiménez-Gestal y Murillo, 2017).

Descripción del recurso

A continuación, vamos a tratar de describir brevemente el entorno desde una perspectiva didáctica, relacionando lo que se puede hacer con posibles condicionantes. Como hemos dicho, BlocksCAD permite modelar cuerpos en 3D mediante la utilización de bloques que tienen su correspondencia directa con instrucciones textuales en OpenSCAD. La ventaja de emplear estos bloques visuales es que no se pueden cometer errores de sintaxis. Es decir, no hay que estar pendiente de si falta un paréntesis, o los parámetros se introducen separados por comas en lugar de por punto y coma. Así, es posible que el programa final no haga lo que se pretendía, pero por lo menos funciona siempre. Por este motivo, a pesar de que los programadores experimentados perciban estos entornos como menos potentes y capaces, su transparencia y facilidad de uso permite que el alumnado los emplee directamente (Weintrop y Wilensky, 2015).

Los diferentes bloques se agrupan en las siguientes categorías: primitivas (3D y 2D), operaciones con conjuntos, transformaciones geométricas, operaciones matemáticas, lógicas, algorítmicas y herramientas de texto. Nos referiremos a ellos con sus nombres originales en inglés, aunque desde la versión 1.8.3 ya hay una versión traducida al español. Esto nos permitirá distinguir claramente en el presente artículo entre las primitivas, modelos y procedimientos de BlocksCAD y los correspondientes cuerpos en el espacio.

PRIMITIVAS (3D SHAPES, 2D SHAPES)

En primer lugar, hay una serie de bloques 3D básicos, que denominaremos primitivas, con los que se pueden modelar cuerpos simples. Son *cube*, *cylinder*, *sphere* y *torus*. El término cubo designa en geometría un caso concreto de prisma; es decir, un prisma cuadrado y de altura igual al lado de la base. Sin embargo, en BlocksCAD con la primitiva *cube* se modela un prisma rectangular de dimensiones arbitrarias (figura 1). Observaremos que, además de especificar las dimensiones en milímetros, se puede elegir entre modelarlo centrado (con centro en el origen de coordenadas) o no centrado (con un vértice en el origen, de manera que el modelo se extiende en la dirección positiva de cada eje).

Podríamos sentirnos tentados de emplear primitivas con nombres acordes a los cuerpos geométricos que representan. Aunque siempre es posible cambiar el

nombre de cada primitiva por otros más acordes con la definición de referencia (como prisma), esto es algo que repercutiría sobre la reusabilidad de modelos y funciones realizados por otros usuarios, lo cual es uno de los alicientes del entorno. Además, el hecho de que el significado de *cube* sea diferente al de la palabra *cubo* que se emplea en geometría, es una ocasión para profundizar en el aula en las propiedades que definen a cubos y prismas. Por otro lado, BlocksCAD no es el único contexto en el cual la palabra cubo presenta otro significado, ya que en un huerto ese término se utiliza para designar a un recipiente, cuya forma además no suele ser cúbica. Las experiencias que hemos realizado con alumnos muestran que esta posible preocupación por la terminología apenas sale a la luz más allá de la primera vez que se utilizan estas primitivas, resultando en una mera convención pragmática.

En la primitiva *cylinder* observamos que tiene más parámetros de los necesarios para especificar un cilindro. Además del radio y la altura, aparece un segundo radio, un candado y la posibilidad de centrar o no el modelo. Si el candado está cerrado, ambos radios serán iguales, mientras que si el candado está abierto podremos especificar el radio de la base inferior (*radius1*) y el radio de la base superior (*radius2*) por separado. Con esto se consigue modelar un tronco de cono y, si uno de los radios es nulo, un cono. Tanto en la opción centrado como no centrado, el centro de los círculos que forman las bases

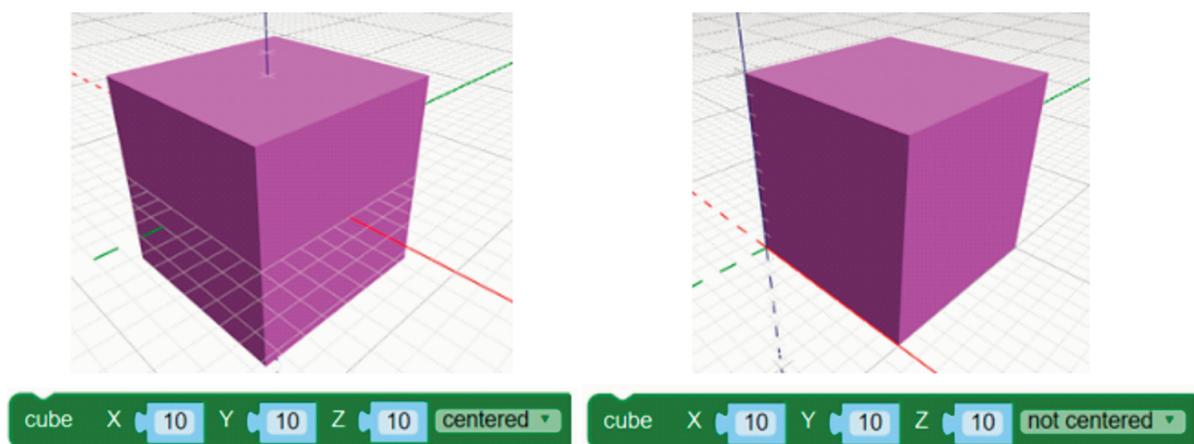


Figura 1. *cube* (10,10,10) para modelar un cubo de 10 mm de arista

pertenece al eje Z. Si lo centramos, la altura se reparte a partes iguales entre el semieje Z positivo y el semieje Z negativo, mientras que en el caso no centrado la altura se extiende solo a lo largo del semieje positivo.

Las otras primitivas 3D son *sphere* y *torus*, que permiten modelar una esfera y un toro, respectivamente.

Finalmente, en BlocksCAD nos encontramos con *circle* y *square*. La mayoría de las transformaciones y operaciones que describiremos a continuación pueden usarse también sobre ellas. Estas primitivas en 2D resultan de utilidad, por ejemplo, para modelar cuerpos 3D a partir de la extrusión de la forma 2D a lo largo de un eje o camino determinado. *Extrusión* es un término del ámbito de la tecnología que hace referencia a la acción y al proceso de extruir; esto es, crear objetos que tengan una sección transversal definida y constante. Un ejemplo cotidiano es la máquina de hacer churros, donde se introduce la masa y, posteriormente, se le obliga a pasar por un orificio con una forma establecida, creando un cuerpo con esa sección constante. En los programas de modelado 3D, cuando se *extruye* una forma 2D a lo largo de un eje, se obtiene un cuerpo (el modelo de un cuerpo) con esa misma sección.

OPERACIONES CON CONJUNTOS (SET OPS)

Dentro de la categoría de operaciones con conjuntos nos encontramos la unión (*union*), intersección (*intersection*), diferencia (*difference*) y el operador *hull* que comentaremos al final. Todas ellas actúan como operadores sobre objetos creados ya con anterioridad, considerando que las primitivas crean cuerpos macizos. En el caso de las tres primeras se obtiene un nuevo conjunto de puntos en el espacio, que conforma un nuevo objeto resultante de aplicar el operador sobre una serie de objetos preexistentes, respectivamente:

- *Union*: todos los puntos que están en alguno de los objetos.
- *Intersection*: los puntos que están en todos los objetos.
- *Difference*: crea un nuevo objeto compuesto por todos los puntos que están en el primer objeto y no están en los demás.

De esta forma, las operaciones con conjuntos y las primitivas 3D permiten modelar diferentes cuerpos: prismas rectangulares, cilindros, conos, esferas, esferas con agujeros con forma de prisma rectangular, etc. Sin embargo, llegados a este punto el lector posiblemente se pregunte cómo modelar otros cuerpos tan elementales como prismas rectos que no sean rec-

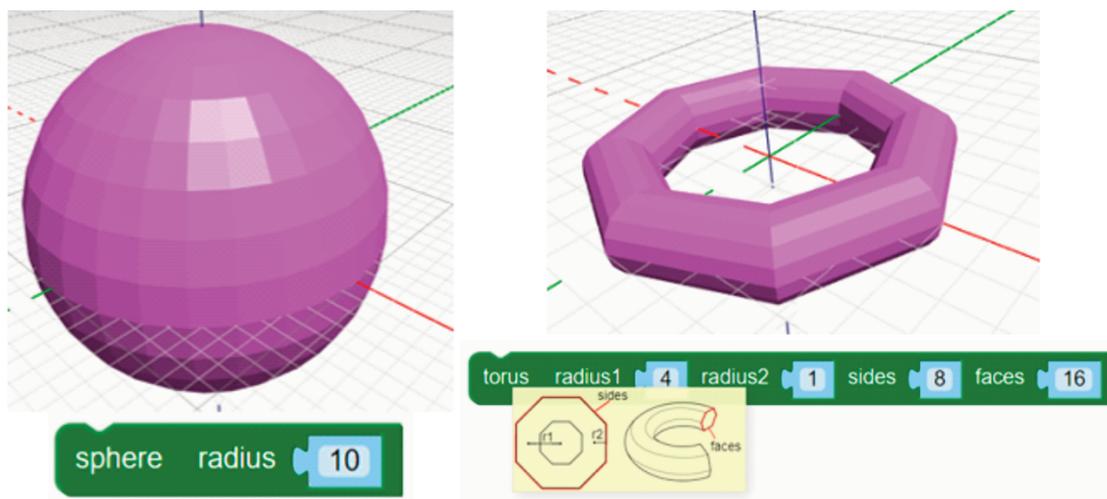


Figura 2. *sphere* (10) para modelar una esfera de radio 10 mm y *torus* (4, 1, 8, 16) para modelar un toro de 8 lados y 16 caras

tangulares, o cómo hacer pirámides. Una primera respuesta la encontramos en la operación *hull*, y otras, en el bloque de transformaciones.

El operador *hull* (que en la versión traducida se denomina *envoltura convexa*) genera el cuerpo convexo mínimo que incluye a los objetos que lo componen. En la figura 3 se muestra un ejemplo en el que se modela un cubo a partir de sus 8 vértices, los cuales se modelan como pequeñas esferas. Y aquí es conveniente detenerse para remarcar que estamos ante un software de modelado. Como ya se ha mencionado, por un lado, resulta evidente que las esferas que modelan puntos no aparecen como esferas ideales a nuestros ojos, sino que es posible apreciar el mallado poligonal. Por otro lado, el propósito principal de este software es el de generar modelos para, posteriormente, ser impresos en 3D. Es decir, que una vez se tiene el mallado que representa el modelo, el proceso de fabricación del objeto tiene que pasar por otro software, que en la jerga se conoce como *de fileteado* (*slicing*), el cual convierte el modelo en instrucciones que pueden ser entendidas por una máquina de impresión 3D (movimientos de motores, posiciones, extrusiones de plástico, temperaturas, etc.). Algunos ejemplos de este software son Slicer o Cura, y como parámetros de fabricación incluyen otros aspectos, como el porcentaje de relleno (para ahorrar plástico, el

interior de estos objetos no es completamente sólido, sino que es una retícula parametrizada según los requerimientos mecánicos de la pieza). La fabricación finaliza con la impresión 3D del modelo, que pasa a convertirse en un cuerpo físico.

Por todo esto, realmente, para los propósitos de trabajo con impresoras 3D, no importa si modelamos los puntos como esferas o como cubos. Basta con que estos objetos sean lo suficientemente pequeños como para que la curvatura al realizar el *hull* sea indistinguible del borde físico que se imprimiría. Estrictamente, conviene observar que en ninguno de los dos casos estaríamos ante un cubo. No obstante, esto se podría conseguir si se modelan los puntos con cubos, calculando las coordenadas de cada uno de ellos para que la envoltura resultante fuera cúbica.

De todas maneras, para un cuerpo tan elemental como el cubo ya hay una primitiva que lo modela perfectamente. Sin embargo, la actividad que consistiría en modelar un cubo de esta forma (u otros cuerpos en el espacio) exige calcular de forma previa las coordenadas de los vértices, lo que conlleva otra forma de abordar el modelado de la pieza, distinta al uso de otras primitivas y operadores. De esta forma, el conocimiento matemático puesto en juego es diferente.

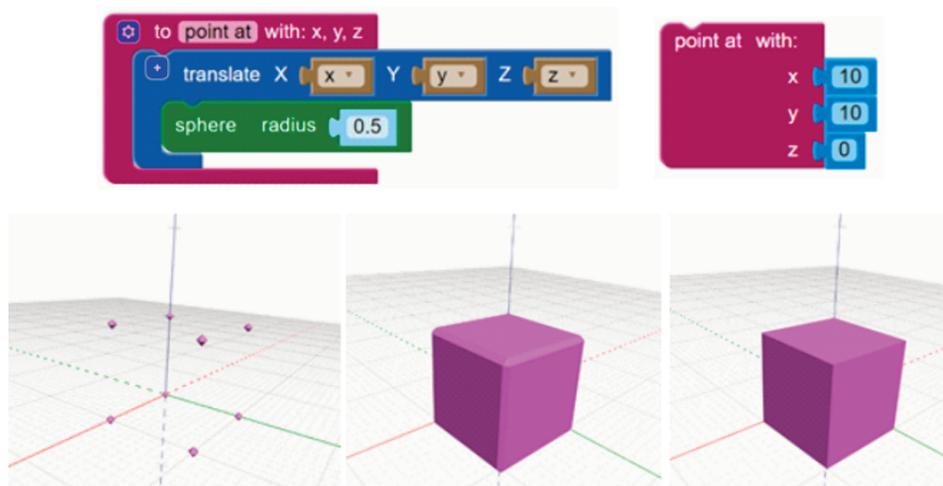


Figura 3. Envoltura convexa o *hull* de los vértices de un cubo, modelados como *spheres* de radio 0,5 mm (izquierda y centro) y radio 0,01 mm (derecha)

MÓDULOS

En el modelado del cubo que se mostraba en la figura 3 se ha hecho uso de un módulo (que hemos denominado *point at*) que generaba una esfera de un radio parametrizable en unas coordenadas determinadas. Los módulos evitan tener que repetir una serie de bloques que se va a utilizar una y otra vez, encapsulándolos en un bloque específico que puede tener parámetros de entrada y devolver resultados en forma de variables o generar un modelo.

Desde la perspectiva del profesor que quiere trabajar un contenido concreto, los módulos permiten proporcionar bloques de código ya hechos para ser utilizados por el alumnado. Por ejemplo, se puede preparar un módulo para hacer una pirámide regular, cuyos parámetros de entrada sean el número de lados de la base, la longitud del lado y la altura.

TRANSFORMACIONES

Aquí encontraremos bloques para hacer traslaciones, rotaciones, simetrías, escalados, escalados a lo largo de un eje y cambios de color. A estas transformaciones hemos de añadir extrusiones lineales o rotacionales y la elección del número de lados. Nos centraremos en estas dos últimas, ya que las primeras son autodescriptivas. Una extrusión, en este contexto, permite modelar objetos con sección transversal definida y fija a lo largo de una línea (extrusión lineal) o en torno a un eje (extrusión rotacional). Se usará, por ejemplo, si se pretende modelar una pieza a partir de su planta. De esta manera, se dibuja en 2D la planta y luego se extruye.

Por otro lado, el bloque *sides* es esencial para poder aumentar el rango de cuerpos que se pueden modelar con las primitivas. Si aplicamos el bloque *sides* a un cilindro, este cilindro ideal pasará a ser modelado por

un prisma poligonal inscrito con el número de lados que especifique *sides*. La figura 4 muestra, a modo de ejemplo, cómo obtener un prisma hexagonal.

MATEMÁTICAS, LÓGICA, VARIABLES Y BUCLES

Cuando resulta necesario realizar cálculos para establecer medidas, parámetros de traslaciones, etc., recurriremos a los bloques que proporciona la categoría *math*, donde se incluyen operaciones aritméticas, funciones trigonométricas y generadores de números aleatorios. Los resultados de estas operaciones pueden introducirse directamente en los parámetros de las primitivas 3D o de las transformaciones o, por otro lado, asignarse a variables que luego pueden ser utilizadas para diferentes propósitos. La categoría *loops* permite introducir bucles en los algoritmos, que pueden ser recursivos, mientras que la categoría *logic* completa la paleta de instrucciones que se pueden añadir a los algoritmos.

Objetivos de aprendizaje y relación con el currículo

Esta forma de modelar cuerpos en el espacio conecta el pensamiento computacional (Wing, 2006) con la visualización y el conocimiento de propiedades de cuerpos y transformaciones en el espacio, así como con las operaciones conjuntistas. Dependiendo de lo que se quiera modelar y cómo, BlocksCAD también enlaza con la geometría analítica, ya que también permite definir trayectorias a partir de ecuaciones paramétricas. Estamos entonces ante un recurso con grandes posibilidades para la enseñanza y el aprendizaje de las matemáticas, con un enfoque muy competencial.

Las tareas que se pueden proponer con BlocksCAD persiguen objetivos de aprendizaje que se pueden rastrear en los contenidos especificados en el currículo

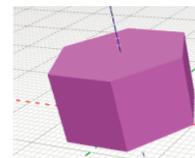
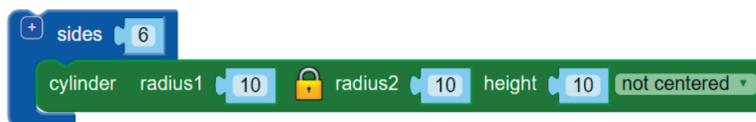


Figura 4. Modelado de un prisma recto hexagonal aplicando *sides*(6) sobre un *cylinder*

(MECD, 2015). Sin ir más lejos, en el bloque de procesos, métodos y actitudes en matemáticas, en todos los cursos de la ESO se alude a la utilización de medios tecnológicos para facilitar la comprensión de propiedades geométricas o funcionales y la realización de cálculos de tipo numérico, algebraico o estadístico.

No se trata aquí de emplear un medio tecnológico porque sí, sino que su uso viene motivado porque el usuario ha de poner en juego su conocimiento sobre geometría, especialmente lo relativo a transformaciones en el espacio. Como valor añadido, cada diseño puede verse como una simulación, puesto que una vez descrito el modelo del cuerpo mediante los bloques de primitivas y transformaciones, el usuario debe pulsar el botón de renderizado para visualizar el objeto resultante. El término pensamiento computacional no aparece explícitamente en el currículo, pero nos alineamos con Baeza-Alba y otros (2016), cuando lo relacionan con la resolución de problemas y el planteamiento de investigaciones matemáticas, especialmente aquellas que involucran procesos de matematización y modelización.

Aunque se pueden proponer ejemplos de tareas que sirven para abordar contenidos de otros bloques, es en la geometría donde la relación resulta más evidente. Finalmente, es un recurso que posibilita el desarrollo de elementos transversales y la interdisciplinariedad. No debemos olvidar que el propósito principal de este software es la elaboración de modelos para ser impresos en 3D. La puerta a planificar proyectos conjuntos con los departamentos de Tecnología queda más que abierta. De hecho, es algo que puede facilitar el primer paso: la familiarización con el artefacto.

Una propuesta sobre el modelado de cuerpos

Elaborar un modelo para el cuerpo representado en la figura 5.

El cuerpo en cuestión se puede proporcionar físicamente, impreso, o como un archivo STL, el cual se puede manipular virtualmente (girar, ampliar, etc.) con

cualquier software de diseño 3D. Observemos que el cuerpo elegido se puede modelar de distintas formas, en función de la capacidad de análisis y de pensamiento computacional del diseñador. Por ejemplo, si pensamos en el cuerpo anterior como un cubo con pirámides de base cuadrada en el centro de cada una de las caras, que podría ser una descomposición fácil de visualizar, nos veremos en la tesitura de utilizar traslaciones y rotaciones, o simetrías, para colocar cada una en el lugar que le corresponde. Si, por el contrario, nos percatamos de que puede modelarse como la unión de un cubo y un octaedro, ambos centrados en el origen, el algoritmo se reduce considerablemente. El código de la figura 6 muestra la unión de un cubo de arista a , fijada inicialmente a 15 mm, y un octaedro. El octaedro, a su vez, se modela como la unión de dos *cylinders* de radio superior 0, a los que se le aplica la operación *sides* con parámetro 4 lados, y uno de ellos se rota en torno al eje Y.

Nótese también que otra persona que haya visualizado de la misma forma este cuerpo, también podría modelarlo con un algoritmo algo distinto. Así, en lugar de utilizar una rotación para una de las pirámides que forman el octaedro, podría haber fijado a 0 el radio de la base inferior.

El Taller de Talento Matemático de la Universidad de Zaragoza (de la Cueva, 2016) ofrece un contexto interesante para plantear este tipo de tareas. En la sesión de noviembre de 2018, los autores propusimos como tarea el modelado de este cuerpo <<https://mat3d.github.io/>>, y la diversidad de soluciones ha sido mayor de la esperada. Fueron 54 los estudiantes que

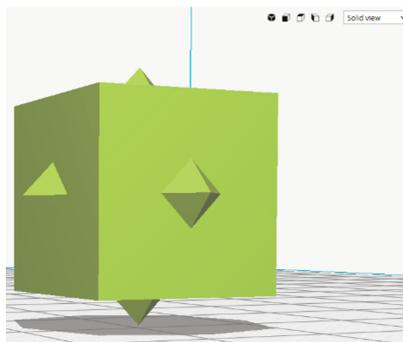


Figura 5. Cuerpo de referencia para la actividad

participaron en la sesión, repartidos en 40 ordenadores. De forma voluntaria, 18 de ellos entregaron sus producciones, que revelan las estrategias de modelado que se muestran en la tabla 1.

Estrategia de modelado	Número de producciones
Unión de dos cilindros de 4 lados, rotando uno de ellos para formar un octaedro, y un cubo.	11
Unión de esfera de 4 lados y cubo.	4
Unión de un cubo y 6 conos (modelados como cilindros sin centrar de lado 4), a partir de rotaciones de estos últimos.	2
Unión de esfera de 3 lados y cubo de 7 lados.	1

Tabla 1. Estrategias de modelado observados para el cuerpo de referencia de la actividad

Es interesante observar que 4 de estas 18 se basaron en la estrategia de la figura 7, modelando un octaedro como una *sphere* sobre la que se aplica la transformación *sides*; es decir, como una «esfera de cuatro lados».

Discusión y conclusiones

El interés de BlocksCAD como recurso didáctico radica en que, al modelar un cuerpo, no solo se ha de pensar en una descomposición de este en otros más simples, sino en modelarlo mediante las primitivas y operaciones de BlocksCAD. Esto se relaciona con el pensamiento computacional, pues exige pensar en el algoritmo necesario para llevar a cabo esa composición. De esta forma, ante la posible complicación del algoritmo para modelar un cuerpo a partir de una visualización determinada, el propio entorno fomenta buscar otra visualización que facilite la programación.

Otro aspecto a destacar es que los modelos diseñados con BlocksCAD se pueden imprimir, con lo que obtenemos el objeto físico, sobre el que se pueden elaborar otras propuestas didácticas interdisciplinares en el marco de proyectos STEM. Actualmente, GeoGebra 3D incorpora también esta funcionalidad, sin duda, atraído por las posibilidades que ofrece el disponer del objeto impreso. Una actividad directa consistiría simplemente en medir y comprobar que el modelo se corresponde con la idea original.

Un aspecto llamativo de BlocksCAD es la nomenclatura de las primitivas. Por ejemplo, con la primitiva *cube* resulta que se pueden hacer prismas rectan-

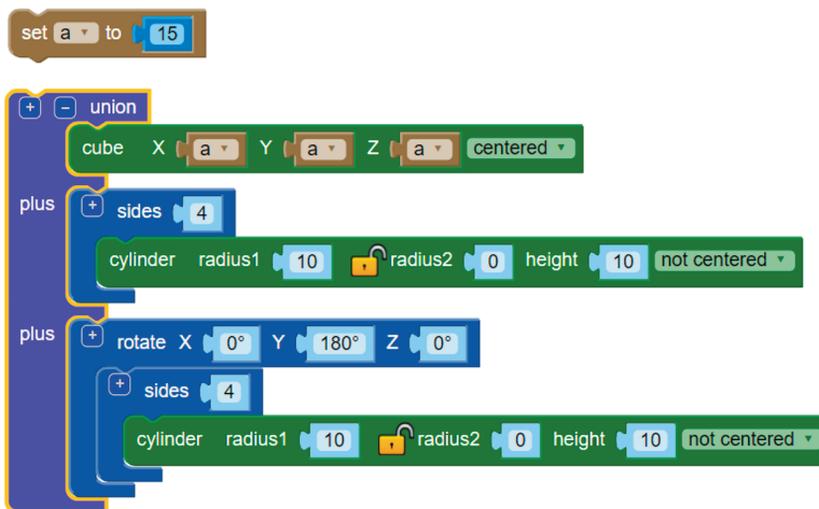


Figura 6. Código que modela el cuerpo de la figura 5 mediante la unión de un cubo y un octaedro

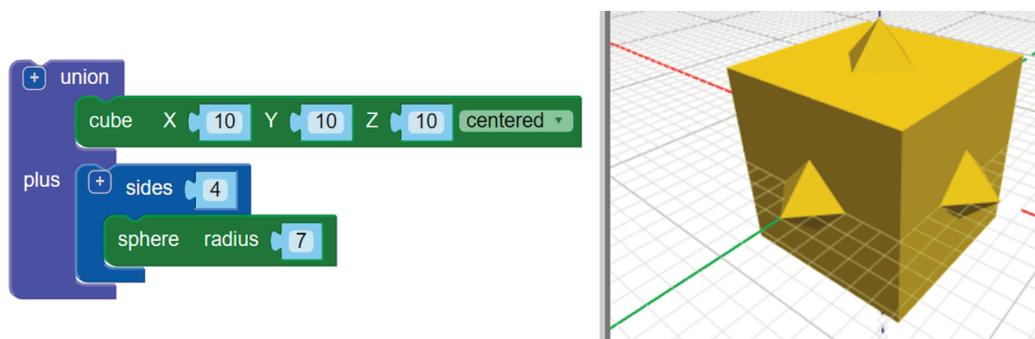


Figura 7. Modelado del cuerpo de la actividad (figura 5) como la unión de un *cube* y una *sphere* de 4 lados

gulares y con la primitiva *cylinder* prismas, conos y pirámides. ¿Estamos entonces ante un posible conflicto de interpretación? Creemos que sí, pero en el sentido de que el conflicto posibilita una oportunidad de aprendizaje hacia la profundización de propiedades que tienen en común los cuerpos que pueden modelarse con cada primitiva. En el fondo, se trata de la clásica distinción entre clasificaciones inclusivas y exclusivas de figuras y cuerpos geométricos: ¿un cuadrado es un rectángulo? Esta cuestión permite abordar la importancia de las definiciones en geometría, a la vez que incide en diferenciar entre el objeto matemático y sus representaciones. Godino y otros (2012, p. 128) remarcan esta necesidad:

Es necesario tener en cuenta la naturaleza no ostensiva, inmaterial, de los objetos matemáticos y las relaciones dialécticas complejas que se establecen entre estos objetos y sus representaciones materiales. Al mismo tiempo se tiene que saber que no hay objeto matemático sin sus diversas representaciones, porque tal objeto no es otra cosa que las reglas de uso de dichas representaciones.

Por otro lado, la composición de modelos mediante operaciones conjuntistas permite introducir un significado para la unión, la intersección y el complementario, contenidos que aparecen, casi de improviso, en el currículo español de matemáticas para la descripción de sucesos aleatorios y el estudio de la probabilidad. La sintaxis propia de estas operaciones ocasiona no pocas dificultades a los estudiantes de secundaria, por lo que creemos que

un trabajo previo como el que proporciona BlocksCAD podría allanar el camino en ese sentido.

La distinción entre artefacto e instrumento (Rabardel, 2001) ayuda a comprender el papel de este tipo de recursos en la enseñanza y aprendizaje de las matemáticas. BlocksCAD, como tal, es un mero artefacto, que solamente se convierte en instrumento en conjunción con ciertas habilidades cognitivas. Esto conforma un marco teórico del que se valen diversas investigaciones que analizan el uso de software en matemáticas, como GeoGebra (Iranzo y Fortuny, 2009) y que permite plantear futuras líneas de trabajo. Lo realmente interesante es analizar cómo afecta el uso del artefacto al razonamiento de los estudiantes y en qué se diferencian estas tareas de otras que no hagan uso del mismo.

Agradecimientos: Esta investigación se ha desarrollado dentro del proyecto EDU2016-74848-P (FEDER, AEI) y dentro del grupo S36_17D - Investigación en Educación Matemática (Gobierno de Aragón y Fondo Social Europeo).

Referencias bibliográficas

- BAEZA-ALBA, M. A., F. J. CLAROS-MELLADO y M. T. SÁNCHEZ-CAMPAÑA (2016), «Una propuesta didáctica en 3.º ESO para trabajar el pensamiento matemático avanzado haciendo uso de Scratch», *Épsilon*, n.º 33(2), 31-46.

- BELTRÁN-PELLICER, P. (2017), «Modelado e impresión 3D como recurso didáctico en el aprendizaje de la probabilidad», *Épsilon*, n.º 34(95), 99-106.
- BELTRÁN-PELLICER, P., y C. RODRÍGUEZ-JASO (2017), «Modelado e impresión en 3D en la enseñanza de las matemáticas: un estudio exploratorio», *ReiDoCrea*, n.º 6, 16-28.
- (2018), «Construcciones en BlocksCAD para analizar el conocimiento en geometría». *Primer Workshop sobre Entornos Tecnológicos en Educación Matemática*, Valencia, España.
- DE LA CUEVA, F. (2016), «El Taller de Talento Matemático», *Entorno Abierto*, n.º 12, 5-7.
- FERRER, T. (2011), «Usando Scratch en secundaria. Competencia matemática y aprender a aprender», *Aula de Innovación Educativa*, n.º 206, 20-23.
- GODINO, J. D., M. GONZATO, J. A. CAJARAVILLE y T. FERNÁNDEZ (2012), «Una aproximación onto-semiótica a la visualización en educación matemática», *Enseñanza de las Ciencias*, n.º 30(2), 109-130.
- GONZÁLEZ-GÓMEZ, J., A. VALERO-GÓMEZ, A. PRIETO-MORENO y M. ABDERRAHIM (2012) «A new open source 3D-printable mobile robotic platform for education», *Advances in autonomous mini robots*, Springer, Berlin, 49-62.
- GUTIÉRREZ, Á., y A. JAIME (2015), «Análisis del aprendizaje de geometría espacial en un entorno de geometría dinámica 3-dimensional», *PNA*, n.º 9(2), 53-83.
- IRANZO, N., y J. M. FORTUNY (2009), «La influencia conjunta del uso de GeoGebra y lápiz y papel en la adquisición de competencias del alumnado», *Enseñanza de las Ciencias*, n.º 27(3), 433-446.
- JONES, R., P. HAUFE, E. SELLS, P. IRAVANI, V. OLLIVER, C. PALMER y A. BOWYER (2011), «RepRap – the replicating rapid prototyper», *Robotica*, n.º 29, 177-191.
- JORGE-POZO, D., C. JIMÉNEZ-GESTAL y J. MURILLO (2017), «Influencia de un entorno virtual de enseñanza aprendizaje en la afectividad hacia las matemáticas de estudiantes de secundaria: estudio de casos», *Investigación en Educación Matemática XXI*, SEIEM, Zaragoza, 325-334.
- RABARDEL, P. (2001), «Instrumented mediated activity in situations», *People and computers XV-Interactions without frontiers*, Springer-Verlag, Berlín, 17-30.
- Real Decreto 1105/2014, de 26 de diciembre, por el que se establece el currículo básico de la Educación Secundaria Obligatoria y del Bachillerato*, Ministerio de Educación Cultura y Deporte, Boletín Oficial del Estado.
- SHELLY, C., G. ANZALONE, B. WIJNEN, y J. M. PEARCE (2015), «Open-source 3-D printing technologies for education: Bringing additive manufacturing to the classroom», *Journal of Visual Languages & Computing*, n.º 28, 226-237.
- WEINTROP, D., y U. WILENSKY (2015), «To Block or not to Block, That is the Question: Students' Perceptions of Blocks-based Programming». *Proc. IDC'15. ACM*, 199-208.
<<http://doi.org/10.1145/2771839.2771860>>.
- WING, J. M. (2006), «Computational thinking», *Communications of the ACM*, n.º 49(3), 33-35.

Pablo Beltrán-Pellicer

Universidad de Zaragoza
<pbeltran@unizar.es>

Carlos Rodríguez-Jaso

IES Pedro Cerrada, Utebo
<carlos.rodriguez.jaso@gmail.com>

José María Muñoz-Escolano

Universidad de Zaragoza
<jmescola@unizar.es>