

BOTBLOQ: Ecosistema integral para el diseño, fabricación y programación de robots DIY

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI)

EXPEDIENTE: IDI-20150289

Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

ACRÓNIMO DEL PROYECTO: BOTBLOQ



UNIÓN EUROPEA

Fondo Europeo de Desarrollo Regional (FEDER)

Una manera de hacer Europa

ENTREGABLE E.2.1. ESPECIFICACIÓN DE REQUISITOS DE LA APLICACIÓN A DESARROLLAR IDENTIFICANDO LAS HABILIDADES.

RESUMEN DEL DOCUMENTO - CAMBIO

Your abstract goes here... ...

January 15, 2016

Contents

1	Introducción	2
1.1	Capacidades cognitivas entre los 6 años hasta los 16 años	2
1.2	Enseñanza de solución de problemas	4
1.3	Uso de algoritmos	8
1.4	Condicionales	9
1.5	Procesos iterativos	10
1.6	Funciones y funciones anidadas	11
2	Metodología	12
2.1	Participantes	12
2.2	Hipótesis	12
2.3	Conclusiones	13
3	Validación de componentes por expertos	14
4	Conclusiones	18
5	Bibliografía	20

1 Introducción

En este trabajo se pretende averiguar en qué edad puede ser óptimo el aprendizaje de la programación usando el lenguaje de bloques Bitbloq, un lenguaje para programar el kit de robótica bq. El hecho de desarrollar una programación en bloques se debe a la necesidad de acercar la programación a niños para que puedan aprender las nociones básicas desde una edad más temprana.

Con Bitbloq se puede programar los componentes electrónicos usados en el kit de robótica (luces LED, zumbador, sensores IR, servos, etc.) para determinar diferentes comportamientos entre los diferentes componentes. Este tipo de tareas conlleva el uso de habilidades típicas del lenguaje lógico usado en programación, las cuales aún están incipientes en las mentes de los niños en el periodo de educación primaria y secundaria, y no se sabe con exactitud la forma en la que los niños pueden utilizarlas realizando actividades de programación en Bitbloq.

En este trabajo se pretende observar la forma en que niños entre 6 y 16 años de edad expresan algoritmos, funciones, procesos iterativos y bucles en Bitbloq para realizar una hipótesis en función de los resultados obtenidos.

1.1 Capacidades cognitivas entre los 6 años hasta los 16 años

Antes de continuar explicando más detalles sobre el proyecto, conviene hacer una importante distinción entre las capacidades cognitivas imperantes entre niños de 6 a 16 años. Martí (2008) describe las capacidades cognitivas en niños de 6 a 12 años de edad, y, Carretero y León (2008) a partir de 12 años y la adolescencia. El primero describe el estadio de las operaciones concretas de Piaget y los segundo el estadio de las operaciones formales (que se empiezan a desarrollar aproximadamente a partir de los 12 años de edad).

Empezando por el estadio de las operaciones concretas, los niños dentro de este periodo aprenden a coordinar el conocimiento que adquieren de forma ordenada y de forma estratégica de cara a la resolución de tareas. No obstante, aún tienen una serie de limitaciones:

- En el estadio de las operaciones concretas, dichas operaciones se refieren a objetos concretos que deben estar presentes cuando realizan una tarea.

- Los niños aún son incapaces de usar hipótesis o enunciados formales desligados de cualquier contenido físico, es decir, toda operación en el pensamiento está subordinada a una realidad concreta y no alcanza la abstracción total. Sólo son capaces de pensar sobre los elementos del problema tal y como éste es presentado.
- Sólo a veces se pueden concebir situaciones posibles adicionales, pero siempre restringidas a una prolongación de lo real.

Sin embargo, en el estadio de las operaciones formales la realidad es concebida como un subconjunto de lo posible. Las personas en esta etapa no sólo son capaces de tener en cuenta todos los datos reales y presentes sino que también pueden prever todas las relaciones causales posibles entre sus elementos. En esta etapa cobra más importancia el carácter hipotético-deductivo del pensamiento al tener disponible la capacidad de realizar abstracciones. Dichas abstracciones se convierten en hipótesis, y se pueden generar varias de forma simultánea. Estas hipótesis están relacionadas para explicar el fenómeno o problema que se tenga que resolver pero no dependen de su presencia.

Además, el carácter proposicional guarda una estrecha relación con el carácter hipotético-deductivo del pensamiento. Para realizar inferencias e hipótesis, se usan proposiciones sobre “lo que puede ser posible” independientemente de la realidad concreta. Por lo tanto, en la etapa adolescente se trabaja no sólo con objetos concretos sino con representaciones proposicionales de los objetos.

Así pues, se puede decir que la diferencia entre un periodo y otro es la presencia, o no, del objeto sobre el cual se está operando (condición esencial para los más pequeños). También habría que atribuir otras diferencias referidas al mayor desarrollo y número de experiencias en cada grupo de edad, siendo los más mayores los más eficientes en el uso de sus capacidades cognitivas.

Ahora, hay que pensar en dos personas, una perteneciente al estadio de las operaciones concretas y otra al de las operaciones formales, que tratan de poner en marcha un servo de rotación continua. La segunda persona, sólo con ver el servo, podría ser capaz de mantener en su mente varias hipótesis sobre la aplicación que puede tener este componente del kit de robótica (ser el motor de un vehículo, girar un molino, subir y bajar un ascensor, etc.) y aplicarlo en contextos que no están presentes mientras experimenta con el servo, mientras que la primera persona

tendría más dificultades para plantear hipótesis dada la necesidad de tener presente los objetos o los contextos sobre los que se está operando.

Con todo lo descrito se podría decir que la potencialidad del kit de robótica dependerá, en parte, de la capacidad de las personas que experimenten él.

1.2 Enseñanza de solución de problemas

Uno de los pilares actuales en el marco educativo es la enseñanza de estrategias en solución de problemas. Existen numerosos trabajos en los que se intenta fomentar el aprendizaje de estas estrategias en el currículo académico en diferentes etapas, desde los primeros cursos escolares hasta titulaciones de grado. En concreto, Rodríguez et al. (2015) habla de la demanda actual sobre formación en la competencia de resolución de problemas en el mundo laboral, además de nombrar la exigencia del Espacio Europeo de Educación Superior (EEES) de incluir dinámicas para el aprendizaje de estrategias en solución de problemas en las titulaciones universitarias.

Queda patente que en la actualidad se está fomentando la enseñanza en solución de problemas, pero este tipo de estrategias se llevan estudiando desde la primera mitad del siglo XX. En uno de los manuales más recientes sobre psicología del pensamiento, Pérez Echeverría (2004) plantea una definición a la solución de problemas a través de la distinción entre resolver problemas y resolver ejercicios. Cuando se resuelve un ejercicio se trata de aplicar una solución inmediata en disposición de una serie de mecanismos conocidos para poder resolverlo (por ejemplo, un niño de primaria resolviendo problemas de matemáticas sabrá que tiene que hacer una multiplicación o una división), pero cuando se hace una tarea de solución de problemas no se dispondría de dichos mecanismos o herramientas que faciliten el proceso de solución.

Esta misma autora menciona también diferentes taxonomías de problemas, siendo interesante aquella en la que se distinguen problemas simples de problemas complejos. La complejidad en este caso dependería de varios factores como la capacidad de las personas de crear una representación del problema, acceder a conocimientos previos, relacionar un problema con otros problemas, su relación con el contexto, clarificar metas y objetivos, etc. (Funke, 1991). Por lo tanto, podemos concluir que el nivel de complejidad de un problema dependería de ciertas

capacidades de las personas y también de la propia naturaleza del problema.

Ahora bien, los participantes de este proyecto van a enfrentarse a una serie de tareas de programación que deberán abordar siguiendo algún tipo de estrategia, y es esta forma de abordar las tareas y cómo expresan determinadas habilidades necesarias en la programación, las que resultan ser el objeto de estudio. En el proyecto que se está desarrollando se quiere dar respuesta a las siguientes preguntas:

- ¿Cómo expresan tareas secuenciales niños entre 6 y 16 años?
- ¿Cómo expresan el uso de condiciones?
- ¿Cómo expresan procesos iterativos?
- ¿Cómo forman funciones y funciones anidadas?

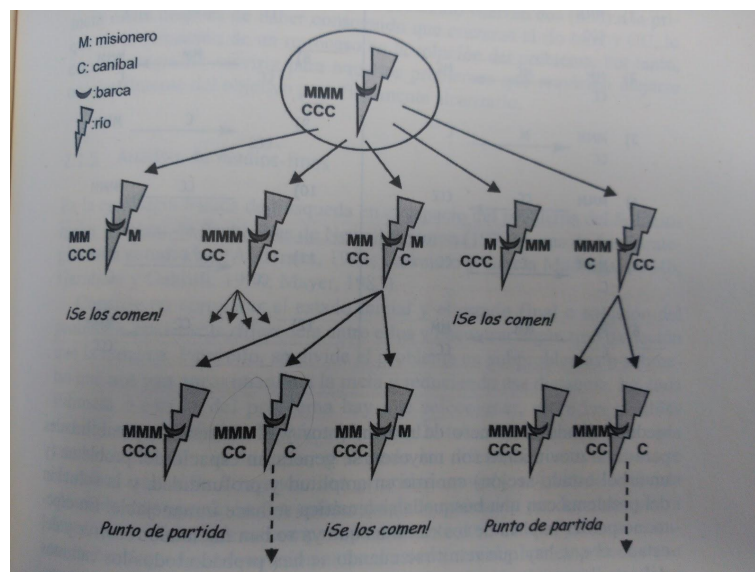
Los participantes del proyecto van a realizar diferentes tareas en las que deben cumplir una serie de objetivos con el kit de robótica que tendrán a disposición, y usando Bitbloq deberán establecer un plan de acción en el que se espera observar cómo usan y expresan las habilidades que se acaban de nombrar.

Si los participantes deben enfrentarse a una tarea nueva (la población escogida no posee conocimientos previos de programación) es interesante observar la forma de abordarlo, lo que sugiere otra cuestión a tener en cuenta: ¿cómo se pone en marcha la solución a un problema? Pérez Echeverría (2004) concluye que la solución de problemas puede ser aplicada a muchos ámbitos, y la forma de establecer las estrategias dependerá de dichos ámbitos (una persona puede ser experta en resolver problemas electrónicos, y puede no serlo en resolver problemas culinarios, y la forma de abordar los diferentes problemas posiblemente sea distinta). Esta misma conclusión la realizan Aguilar et al. (2002), los cuales nombran numerosos estudios de los que se puede concluir la importancia de los conocimientos básicos y específicos para la comprensión de un problema y su resolución. Existen formas concretas de abordar problemas concretos, pero un buen solucionador de problemas tendrá un marco de actuación general para abordar cualquier tipo de problema: analizar el problema, planear un modo de resolverlo, llevar a cabo el plan, revisar el trabajo, o, establecer una secuencia de pasos a seguir para llevar a

cabo un plan de acción. A continuación se expone la descripción de Postigo (2004) sobre las estrategias de tipo general:

- Búsqueda aleatoria: consiste en buscar de manera desordenada la solución a un problema, es decir, se dan respuestas al azar esperando encontrar por casualidad la solución del problema. No requiere seguir ningún tipo de planificación y ver si damos con la combinación correcta.
- Ensayo y error: la búsqueda de la solución es ordenada y planificada. Se seleccionan diferentes caminos para encontrar la solución y se van comprobando los resultados de cada uno hasta solucionar el problema. Los resultados fallidos se registran para no repetirlos.
- Búsqueda exhaustiva o sistemática: se crea un diagrama o árbol estado-acción a partir del estado inicial. Consiste en plantear los posibles resultados a partir de ese estado. A continuación, se plantean los posibles resultados de cada uno de los resultados anotados. Y así sucesivamente.

Ejemplo:



- Subir la montaña: esta estrategia es una modificación de la anterior (búsqueda exhaustiva). La diferencia está principalmente en que sólo desarrollamos el resultado que consideramos más eficaz y/o el que más nos acerca a la meta, es decir, con el que menos pasos necesitaremos.

- **Análisis de medios-fines:** Consiste en comparar el estado inicial y el estado final o solución del problema, analizar la diferencia entre ellos y encontrar algún tipo de acción que se aproxime y reduzca la distancia entre ambos. Para ello, se divide el problema en submetas que nos aproximan a la solución y se seleccionan las acciones más adecuadas para cada estado del problema (estados entre submeta y submeta).
- **División del problema en subproblemas:** Consiste en fraccionar el problema en varias partes y resolver cada pequeña parte de forma independiente. A diferencia de la estrategia de análisis de medios-fines, esta estrategia no implica que las partes sean fases del proceso del problema. Por tanto, hay que tener cuidado al utilizar esta estrategia, ya que la división del problema puede no ser acertada (por ejemplo, por una representación errónea del mismo) lo que no garantiza la resolución del problema.
- **Búsqueda hacia atrás:** Esta estrategia es similar a la de análisis de medios-fines. La diferencia es que, en este caso, el árbol de estado-acción se empieza desde el estado final del problema y se va explorando hacia el estado inicial. Para aplicar esta estrategia, es necesario que la meta del problema esté suficientemente especificada y que sea una solución única, es decir, que el problema no se pueda solucionar de otra forma.
- **Analogías:** Consiste en utilizar la solución de otro problema diferente pero cuya solución sirve como modelo para solucionar el problema existente. Aunque parezca que esta estrategia es bastante común, los estudios dicen que su uso es poco frecuente porque no siempre es fácil detectar el parecido entre los problemas.

Una de las conductas que se quiere observar es si existe una mejora en la capacidad de resolver problemas. Gibbon (2007) encontró que alumnos de 5º y

6º curso en EEUU que utilizaban Lego Mindstorms tenían mejores resultados de pensamiento divergente y convergente en resolución de problemas. Concretamente, matizaba que usuarios frecuentes de Lego Mindstorms poseían mayor capacidad de resolver problemas divergentes, y usuarios poco frecuentes o iniciados en Lego Mindstorms puntuaban más alto en resolución de problemas convergentes. Lego Mindstorms parece ser que estimula la capacidad de resolución de problemas en general, pero en el estudio hecho no se menciona exactamente si hay un cambio conductual en la estimulación de dicha capacidad. En este trabajo se pretende entonces, bajo la premisa de que la programación estimula la capacidad de resolver problemas en niños, si se produce un cambio en las estrategias y de qué forma.

Programar utilizando Bitbloq requiere establecer secuencias de bloques de programación para ejecutar una determinada acción a través de diferentes componentes de robótica, lo que cognitivamente hablando significa establecer una serie de estrategias o un plan de acción (con una determinada elaboración), en el cual, cada persona hace una sucesión de pasos para que los componentes electrónicos mantengan unas funciones que permitan cumplir un objetivo o una serie de objetivos. Así, a través del plan de acción que utilicen los participantes, se pretende observar el modo en que establecen algoritmos, usan condicionales, crean bucles (procesos iterativos) y programan funciones.

1.3 Uso de algoritmos

“Un algoritmo es un procedimiento sistemático que indica paso a paso cómo debe realizarse una tarea, y que garantiza el hallazgo de una solución para el problema” (Gabucio Cerezo, 2011). Los participantes (personas entre 6 y 16 años) experimentarán con los componentes del kit de robótica y una placa ZUM tratando de alcanzar una serie de objetivos a través de los bloques de programación Bitbloq.

Codina et al. (2015) enumera varios estudios de resolución de problemas y establecimiento de estrategias para resolver problemas matemáticos donde se pone de manifiesto que los alumnos tienen más dificultades en las primeras fases (búsqueda de información o analizar la tarea) para resolver problemas o a la hora de implementar un plan de acción. Sin embargo, estos autores mencionan que en algunos estudios el uso de aplicaciones o medios digitales facilita el uso e implementación de estrategias para la resolución de problemas matemáticos.

En Bitbloq, los participantes cuentan con una serie de bloques de programación, por lo tanto, se espera que a través de los bloques de programación (como variable digital facilitadora) los participantes sepan establecer algoritmos (con diferentes grados de complejidad en función de la edad). Martí (2008) plantea que niños y niñas a partir de los 6 años poseen unos recursos cognitivos básicos que les permiten establecer planes de acción para resolver tareas, y dichas habilidades mejorarán en eficacia según vayan pasando los años, por lo que es de esperar que los participantes sepan establecer una serie de algoritmos en la programación con la finalidad de cumplir unos objetivos.

La edad puede ser determinante en la eficacia de establecer algoritmos, pues la capacidad de los más pequeños en principio limitaría el uso de algunos componentes del kit de robótica y determinados bloques de programación. Un LED RGB (en el que se debe decidir en valores numéricos el color que se quiere) tiene un funcionamiento más complejo que un LED (encender o apagar). Además, como se ha dicho antes, los más pequeños necesitan tener presente un objeto concreto en el que van a operar, y usar el LED RGB puede requerir elaborar una hipótesis o una abstracción sobre su aplicación y su funcionamiento, y, sin embargo el LED sólo se enciende y se apaga.

En conclusión, teniendo en cuenta las especificaciones hechas sobre las capacidades cognitivas en diferentes etapas del desarrollo, se espera que los algoritmos de los más pequeños utilicen bloques de alto nivel (generales) y siguiendo secuencias sencillas. Según vaya aumentando la edad, las secuencias podrían ser más eficientes y de más bajo nivel (teniendo en cuenta aspectos concretos de la programación).

1.4 Condicionales

El uso del conocimiento condicional depende del conocimiento declarativo y procedimental de una tarea, añadiendo el análisis de las condiciones de dicha tarea. Las tareas condicionales pueden ser experiencias complejas pues se requiere una serie de conocimientos previos acerca de la naturaleza de una tarea, los medios para resolverla, y los objetivos que se quieren cumplir. Concretamente, Asensio (2004) matiza que el uso del condicional en el pensamiento proposicional puede ser considerado la conectiva más compleja y difícil incluso para adultos instruidos en su uso.

Como ya se ha explicado, los participantes se encuentran en una franja de edad en la cual poseen recursos cognitivos que les permiten organizar la información en diferentes dominios y usarla de forma estratégica (Martí, 2008), pero el uso del condicional para los niños de menor edad va a estar más limitado.

Un dato a tener en cuenta es la teoría de los modelos mentales propuesta por Johnson-Laird (1983) y Jonhson-Laird y Byrne (1992), que versa sobre la capacidad de las personas de razonar elaborando y manipulando representaciones mentales de objetos y situaciones del mundo, en la cual, a mayor número de representaciones supondría mayor carga en la memoria operativa (la memoria RAM de un ordenador sería el equivalente a la memoria operativa, donde se cargaría la información necesaria para resolver una tarea) dificultando el razonamiento (Santamaría, García-Madruga y Carretero, 1996). Según esta predicción de la teoría de modelos mentales, y recordando la necesidad de los niños pertenecientes al estadio de las operaciones concretas de disponer del objeto físico y real sobre el que está operando, los participantes más jóvenes tendrían más dificultades en superar las pruebas sobre condicionales que se van a plantear a través de Bitbloq. Así, se estima que a mayor número de condicionales en uso, mayores dificultades encontrarán por la sobrecarga de memoria operativa, dificultando el razonamiento disponible para realizar las tareas.

1.5 Procesos iterativos

Otro de los aspectos que se tiene interés en observar es la capacidad de programar un proceso iterativo a través de bucles. En la búsqueda bibliográfica no se ha encontrado nada relacionado con este campo de investigación, por lo que sólo podemos observar si los participantes utilizan algún proceso iterativo concreto a través de la funcionalidad de un bucle de programación.

Así pues, basándonos de nuevo en las capacidades de los niños, se observaría que tipo de bucles son capaces de realizar en función de la complejidad de dichos bucles. Vamos a llamar bucles simples a aquellos que alterna la ejecución de uno o dos bloques de la misma categoría, y complejos a aquellos que poseen varios bloques y de diferentes categoría. Esta pequeña categorización se hace para facilitar tanto la observación como la descripción de los resultados.

1.6 Funciones y funciones anidadas

Al igual que los procesos iterativos, no se ha encontrado nada relacionado con la forma que tienen los niños en diferentes etapas de expresar funciones usando programación de bloques. Siguiendo la estela de lo descrito hasta ahora, se espera que la complejidad y la eficacia de la programación de funciones varíe una vez más en función de la edad de los participantes.

2 Metodología

2.1 Participantes

Los participantes en este primer paso fueron alumnos de primaria y secundaria.

Ninguno de los participantes padecía algún tipo de necesidad educativa especial (TDAH, TEA, dislexia, etc.)

2.2 Hipótesis

En este trabajo realmente existen cuatro preguntas a las que se quiere dar respuesta, siendo una por cada habilidad que quiere observarse en los alumnos participantes:

1. ¿Cómo establecen algoritmos niños entre 5 y 16 años de edad?
2. ¿Cómo crean funciones?
3. ¿Cómo usan condicionales en la programación?
4. ¿Cómo establecen bucles o procesos iterativos?

No obstante, existe una pregunta que engloba a estas cuatro: ¿A qué edad es óptimo aprender programación?

Las habilidades de pensamiento lógico nombradas se expresarán de diferente manera en función del nivel de desarrollo, por lo tanto, se pueden hacer las siguientes preguntas: ¿es mejor esperar a la adolescencia para enseñar programación?, o, ¿enseñar a edades más tempranas facilita la programación en edades posteriores?

Estas cuestiones van a tratar de contestarse a través de la observación de los participantes en diferentes tareas de programación y en diferentes edades para poder contestar a la cuestión principal: ¿A qué edad es óptimo aprender programación?

2.3 Conclusiones

Se han descrito 4 habilidades consideradas importantes o básicas para realizar una tarea de programación. Estas habilidades atañen al lenguaje y pensamiento lógico, y el lenguaje de programación por bloques Bitbloq debería facilitar su desarrollo. Las habilidades también deberían desarrollarse independientemente y de forma combinada.

El lenguaje de programación puede facilitar el desarrollo de estas habilidades y su uso combinado, pues programando no sólo se puede elaborar secuencias de órdenes, sino también añadir condiciones, generar bucles lógicos y anidar funciones. Además, la realización de estas tareas debe facilitar y permitir el desarrollo de estrategias de solución de problemas.

En la teoría se ha mencionado la importancia de establecer un plan de acción mediante secuencias de actuación, retroalimentando la habilidad de establecer algoritmos.

Por último, aunque existan diferencias cognitivas en función de la edad de las personas (sobre todo en la población objetivo), quiere observarse la forma en que éstas se expresan y si es posible crear un lenguaje de programación que su uso se adapte a diferentes edades.

3 Validación de componentes por expertos

En base a la experiencia adquirida durante todos los talleres con niños llevados a cabo. Se ha realizado una consulta a los monitores que habitualmente imparten estos talleres, es decir, una validación por expertos para comprobar la percepción de todos ellos.

Para la validación se pidió que marcaran qué componentes de la lista que son adecuados técnicamente para cada edad, teniendo en cuenta principalmente tres aspectos:

1. comprensión general de componente y de su utilidad.
2. cantidad de conexiones por pines con el kit de robótica de BQ.
3. dificultad de la programación del componente.

Esta validación se ha realizado concretamente con 7 expertos: Alberto Valero, Alfredo Sánchez, Ana de Prado, Beatriz Fernández, José Alberca, Julián Caro y Pedro de Oro.

La validación se apoya en una serie de actuaciones durante el año 2015, en talleres y colaboraciones con el mundo educativo tales como:

- Cursos de formación al profesorado de infantil, primaria y secundaria tanto acreditados por la Comunidad de Madrid como por iniciativa propia (en ambos casos con un respaldo en forma de afluencia consolidado y una valoración muy alta por el profesorado tras realizarlo).
- Talleres propios en nuestras oficinas con menores de todas las edades en colaboración con colegios, academias y ludotecas con muy buenos resultados en cuanto a motivación y seguimiento del plan propuesto.
- Talleres por medio de colaboradores con nuestros materiales que siempre han desembocado en reediciones y nuevos talleres. En estos casos la temática ha sido tradicionalmente ubicada en el formato de clases extraescolares.
- Presencia en aula en el contexto de asignaturas curriculares de primaria y secundaria en diferentes colegios e institutos de España asistiendo con producto y experiencia al profesorado para su correcta implantación.

- Experiencia piloto en un colegio de Madrid impartiendo la asignatura de Tecnología, programación y robótica un año antes de su puesta en marcha curricularmente, con resultados medidos y contrastados que refuerzan nuestra idea del modo de abordar el aprendizaje de la tecnología y la programación.
- Formación universitaria a futuros profesores en las áreas de magisterio, diversos másteres TIC y cursos de experto, así como seminarios tanto en universidades públicas como privadas.
- Desarrollo de una respuesta Creative Commons a la propuesta curricular de la Comunidad de Madrid para la asignatura de Tecnología, programación y robótica, ubicada en nuestro espacio virtual y con seguimiento por parte del profesorado de la Comunidad de Madrid.
- Desarrollo de un marco de acción para revertir la tendencia negativa en matriculaciones de un instituto público de la Comunidad de Madrid con el objetivo de convertirlo en un instituto de excelencia tecnológica (I.E.S. Federico García Lorca de Las Rozas de Madrid).

Posteriormente, se ha puesto en común los resultados de esta consulta y se han debatido las respuestas de todos hasta llegar a una conclusión común sobre el manejo de cada uno de los periféricos estudiados.

Los resultados obtenidos se resumen en la siguiente tabla:

Tabla 1. Periféricos adecuados por edad/nivel de conocimiento. Fuente: elaboración propia.

Con estos resultados iniciales se ha planteado un estudio de validación, que se llevará a cabo de forma simultánea al estudio de casos sobre las habilidades de programación en las diferentes edades estudiadas.

Esta validación se realizará mediante observación a los niños de las diferentes edades, comprobando adecuación de los resultados extraídos de la validación por expertos y, en caso de que alguno no se ajuste a lo planteado, se realizarán estudios complementarios para comprobar la edad más adecuada para el uso del periférico.

El tipo de observación será sistematizada de laboratorio, pues se tiene preparado previamente una situación en la que los participantes realizarán la tarea de forma controlada y sin que existan elementos que puedan interferir. En cuanto a la medición, se registrará el modo en que aplican algoritmos, condicionales, bucles

Componente	<10	10	11	12	13	14	15	16	>16
Pulsador	X								
Sensor de luz		X							
Sensor de infrarrojos	X								
Potenciómetro		X							
Joystick			X						
Sensor de ultrasonidos		X							
Sensor de temperatura y humedad		X							
Sensor de sonido		X							
Sensor de color				X					
Reloj en tiempo real (RTC)		X							
LED	X								
Zumbador	X								
Servo de rotación continua	X								
Servo (miniservo)		X							
Pantalla LCD				X					

y programan funciones y de qué forma usan los componentes del kit de robótica.
Para ello se usará la siguiente tabla de observación:

Validación de componentes del kit de robótica

Componente	Edad recomendada	Edad recomendada
LED	<10	
Pulsador	<10	
Zumbador	<10	
Sensor IR	<10	
Servo de rotación continua	<10	
Sensor de Luz	10	
Potenciómetro	10	
Sensor de temperatura y humedad	10	
Sensor de sonido	10	
Reloj en tiempo real (RTC)	10	
Servo (mini servo)	10	
Sensor de ultrasonidos	11	
Joystick	12	
Sensor de color	13	
Pantalla LCD	13	

4 Conclusiones

¿Qué se espera de este proyecto? Se quiere construir una plataforma de acceso universal donde cualquier persona pueda aprender robótica y a programar usando un lenguaje de bloques. Aunque cualquier persona pueda tener acceso, la adaptación en la que se quiere trabajar está destinada a una población infantil, por lo tanto, los bloques

de programación deben ser intuitivos, que resuman las funcionalidades de los componentes del kit de robótica y permita tener flexibilidad a la hora de programar. Por lo tanto, la creación de un sistema de bloques debería tener en cuenta los siguientes puntos:

- Los bloques deben ser intuitivos. Los bloques que pongan en marcha los componentes de robótica deben ser específicos en describir su funcionalidad.
- Los bloques que permitan operar y establecer conectivas lógicas deben poder anidar fácilmente diferentes bloques.
- La interfaz debe ser accesible y discriminativa. Por un lado, los usuarios deben ver de forma clara en la pantalla del ordenador qué componentes están usando, cómo los están programando, y qué relaciones establecen unos con otros.
Por otro lado, deben discriminarse fácilmente los diferentes tipos de bloques, recomendando el uso de colores para facilitar dicha discriminación. Representar de forma visual los bloques de programación permitiría una fácil discriminación sobre la programación que se está realizando.
- Los bloques deben facilitar un aprendizaje progresivo sobre diferentes aspectos de la programación. Por un lado debería poderse realizar programaciones que den resultados con un feedback inmediato de aquello que se esté programando (que se encienda un LED o un servo gire). Por otro lado, debe ser posible poder realizar programaciones complejas que pongan en juego el uso de varios componentes para cumplir un objetivo.

En resumen, los bloques de programación debe promover el uso intuitivo de los componentes de robótica independientemente de su complejidad. La dificultad

sólo debe recaer a la propia naturaleza del componente y no en su programación. Un LED se podría programar usando un bloque que indique “Encender LED”, y un mini servo (en teoría más complejo) también debería programarse con la misma facilidad. En este caso la complejidad de programar un servo es mayor porque éste se mueve en una franja de 180°, obligando a los usuarios a tener conocimientos de matemáticas previos para entender su funcionamiento. Un niño que no posea los conocimientos previos suficientes para entender un componente, tendrá mayor dificultad en entender su uso, pero la programación deberá ser igual de sencilla para todos los componentes.

Así pues, se ha estimado la complejidad de los componentes en función de sus características y los conocimientos previos que pueden facilitar su uso. Por ejemplo, el LED simplemente se enciende, algo que los niños más pequeños ven constantemente en su contexto. Por el contrario el sensor de color ya no es tan sencillo porque no sólo hay que entender la luz como un espectro, o cómo algo que se propaga, sino que también el color de los objetos depende de la reflectancia que tienen. De esta manera, la edad será un aspecto determinante en el uso de los componentes del kit de robótica no sólo por las capacidades cognitivas de los niños sino también por su conocimiento y experiencia en el mundo.

5 Bibliografía

Aguilar, M., Navarro, J. I., López, J. M. y Alcalde, C. (2002). Pensamiento formal y resolución de problemas matemáticos. *Psicothema*, Vol. 14, N° 2, pp. 382-386.

Asensio, M. (2004). Razonamiento proposicional. En Carretero, M. y Asensio, M., *Psicología del pensamiento*. Madrid: Alianza Editorial.

Carretero, M., León, J. A. (2008). Del pensamiento formal al cambio conceptual en la adolescencia. En Palacios, J., Marchesi, A., Coll, C., *Desarrollo psicológico y educación*, 1. *Psicología evolutiva*, Madrid: Alianza Editorial.

Codina, A., Cañadas, M. C., Castro, E. (2015). Mathematical problem solving through sequential process analysis. *Electronic Journal of Research in Educational Psychology*, Vol 13 (1), pp. 73-110.

Edilberto J. Rodríguez Rivero, Aurora E. Rabazo Martín, David Naranjo Gil (2015). Evidencia empírica de la adquisición de la competencia de resolución de problemas. *Perfiles Educativos*, Vol. XXXVII, N° 147.

Funke, J. (1991). Solving Complex Problems: Exploration and Control of Complex Social Systems. En R. J. Sternberg y P. A. Frensch, *Complex Problem Solving. Principles and Mechanisms*. Hillsdale, New Jersey, LEA.

Gabucio Cerezo, Fernando (2011). Miquel Siguan y el murmullo de la conciencia en Lev Vygotsky y James Joyce. *Revista de Historia de la Psicología*, Vol. 32 (2-3) Jun-Sep, 125-140.

Gibbon, L. W. (2007). Effects of lego mindstorms on convergent and divergent problem-solving and spatial abilities in fifth and sixth grade students. *Dissertation Abstracts International Section A: Humanities and Social Sciences*, Vol. 68 (5-A), pp. 1870. Publisher: ProQuest Information & Learning.

Martí, E. (2008). Procesos cognitivos básicos y desarrollo intelectual entre los 6 años y la adolescencia. En Palacios, J., Marchesi, A., Coll, C., *Desarrollo psicológico y educación*, 1. *Psicología evolutiva*, Madrid: Alianza Editorial.

Pérez Echeverría, M. P. (2004). Solución de problemas. En Carretero, M. y Asensio, M., *Psicología del pensamiento*. Madrid: Alianza Editorial.

Postigo Angón, Y. (2004). Estrategias en solución de problemas. En Carretero, M. y Asensio, M., *Psicología del pensamiento*. Madrid: Alianza Editorial.

Santamaría, C., García-Madruga, J. A. y Carretero, M. (1996). Beyond belief bias: Reasoning from conceptual structures by mental models manipulation.

Memory and Cognition, N.º 25, pp. 250-261.