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

ENTREGABLE E.2.1. Especificación de requisitos de la aplicación a desarrollar identificando las habilidades.

RESUMEN DEL DOCUMENTO

El objeto de esta investigación consiste en identificar las principales capacidades de programación de niños entre 10 y 16 años, y de menor edad, identificando cuándo podría ser óptimo aprender a programar. En este documento se plantean las bases pedegógicas para observar cómo niños y adolescentes establecen estrategias de resolución de problemas y expresan las siguientes habilidades: crear algoritmos, funciones, condicionales y bucles.

February 8, 2016

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

Contents

1 Introducción **2**

1.1 Capacidades cognitivas entre los 6 años hasta los 16 años 2

2 Programación de electrónica en niños **5**

2.1 Enseñanza de solución de problemas 5

2.2 Uso de algoritmos 9

2.3 Funciones y funciones anidadas 11

2.4 Condicionales 11

2.5 Procesos iterativos 12

3 Metodología **13**

3.1 Componentes 13

3.2 Participantes 13

3.3 Planteamiento de la investigación 13

3.4 Habilidades 14

4 Estudios realizados **16**

5 Conclusiones y Objetivos del estudio **22**

6 Bibliografía **24**

7 Anexo I Maletín de electrónica de BQ **26**

8 Anexo II Lista de formaciones con Bitbloq y el Kit de robótica, durante 2015 **30**

9 Anexo III Guiones sesiones de formación **31**

10 Anexo IV Guía taller con el Kit **36**

11 Anexo V Imágenes de formaciones con Bitbloq y el Kit de robótica, durante 2015 **52**

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

1 Introducción

En este trabajo se pretende averiguar a 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 este trabajo se pretende observar la forma en que niños entre 10 y 16 años de edad (aunque también se quieren plantear posibilidades a partir de 5 años) 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 segundos 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

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

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.

Para ilustrar estos conceptos vamos a 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

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

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, del desarrollo de las personas que experimenten él.

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

2 Programación de electrónica en niños

2.1 Enseñanza de solución de problemas

Uno de los pilares actuales en el marco educativo es la enseñanza de estrategias en solución de problemas. Existen algunos 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 el manual de psicología del pensamiento de Carretero y Asensio, 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. Según Colombo de Cudmani (1998), en las diferencias entre novatos y expertos se puede encontrar la automatización de procesos en los segundos suprimiría la toma de decisiones que debe tomar para resolver tareas, convirtiéndose entonces dichas tareas en ejercicios.

Volviendo a Pérez Echeverría, 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 prob-

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

lema 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). A esta misma conclusión llegan 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

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

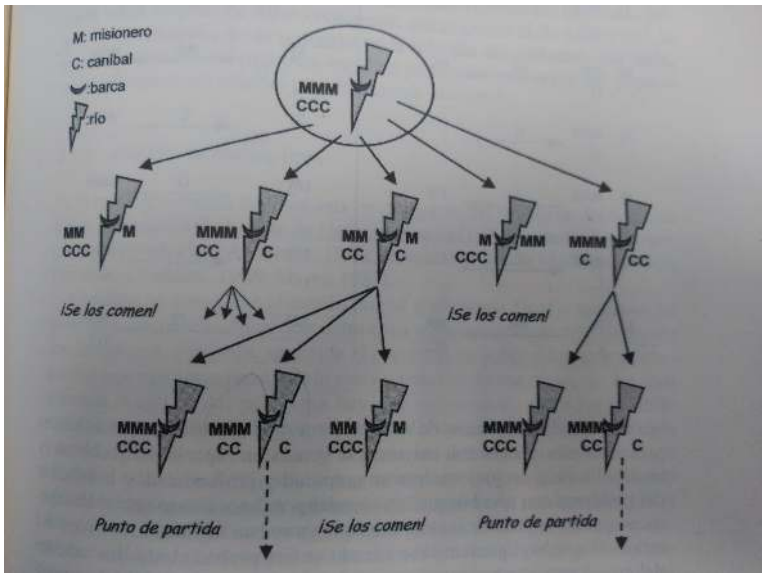
plan, revisar el trabajo, o, establecer una secuencia de pasos a seguir para llevar a cabo un plan de acción. A continuación se expone la descripción de Postigo (2004) sobre las estrategias de tipo general:

- 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

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

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.

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

El objeto de este estudio es observar cómo los participantes expresan algoritmos, funciones, condicionales y bucles mientras realizan tareas de programación, pero es interesante también observar la capacidad de resolver problemas pues consideramos una estrecha relación en la expresión de las habilidades recién nombradas (algoritmos, funciones, condicionales y bucles). 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.

2.2 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 10 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, A., Cañadas, M. C. y Castro, E. (2015) enumeran varios estudios de

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

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 y 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 con componentes electrónicos, 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 puedan ser expresados siguiendo secuencias sencillas (al menos) en todas las edades gracias a los bloques de alto nivel. Según vaya aumentando la edad, las secuencias podrían ser más eficientes y de más bajo nivel (teniendo en cuenta aspectos concretos de la programación).

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

2.3 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, y por los conocimientos previos que puedan facilitar la expresión de funciones.

2.4 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.

Una teoría 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 dificultando el razonamiento (Santamaría, García-Madruga y Carretero, 1996). Para entender bien el funcionamiento de la memoria operativa, haciendo una analogía, la memoria RAM de un ordenador sería el equivalente a la memoria operativa, donde se cargaría la información necesaria para resolver una tarea. 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

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

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.

2.5 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 qué 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 el estudio de cara a facilitar tanto la observación como la descripción de los resultados.

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

3 Metodología

3.1 Componentes

No se ha encontrado nada que resulte útil, o explique de qué forma es necesaria hacer que los participantes usen los diferentes componentes del kit de robótica. Tampoco se conoce la edad recomendada en entender el funcionamiento de los componentes y el uso que van a ser capaz de aplicar. Para ello, se ha hecho una validación por parte de expertos en robótica, en la cual, dichos expertos establecieron a que edad es recomendable usar los diferentes componentes del kit de robótica. Se ha utilizado el Kit de Electrónica de BQ, cuyos componentes se detallan en el Anexo I Maletín de electrónica de BQ.

3.2 Participantes

Los participantes en este primer paso serán alumnos de educación primaria y secundaria, comprendiendo una franja de edad entre 10 y 16 años.

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

3.3 Planteamiento de la investigación

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

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

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

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 una determinada edad 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 las habilidades definidas hasta ahora, es decir, expresar tareas secuenciales, usar condiciones, expresar procesos iterativos y el uso de funciones en los participantes a través de diferentes tareas de programación y en diferentes edades para poder contestar a la cuestión principal: ¿A qué edad es óptimo aprender programación?

3.4 Habilidades

Para medir las 4 habilidades descritas (expresión de algoritmos, funciones, condicionales y bucles) se realizará una observación de forma controlada, ya que se preparará previamente una situación de aprendizaje en la que se eviten o controlen variables extrañas, como por ejemplo: los alumnos consulten en internet cómo hacer una actividad en lugar de hacerlo por sí mismo, o no use correctamente la aplicación o los componentes del kit de robótica. En cuanto a la medición se registrará, además de las estrategias con las que abordan las tareas y si existen cambios con la experiencia, el modo en que los participantes aplican algoritmos, condicionales, bucles y programan funciones. Para ello se usarán las siguientes tablas de observación:

Validación de estretagias	
Edad	**Observaciones**

Tabla 1. Validación de algoritmos.

Validación de algoritmos	
Edad	**Observaciones**

Tabla 2. Validación de estrategias.

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

Validación de funciones	
Edad	**Observaciones**

Tabla 3. Validación de funciones.

Validación de condicionales	
Edad	**Observaciones**

Tabla 4. Validación de condicionales.

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

4 Estudios realizados

En base a la experiencia adquirida durante todos los talleres de robótica educativa llevados a cabo por el equipo del Departamento Educativo de BQ. 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 realización de este estudio y confirmación del uso de componentes en diferentes edades y ámbitos educativos, también se ha contado con la experiencia en aula de Alfredo Sánchez en el marco de la colaboración Colegio Balder - BQ, en el Anexo I del documento E.2.3.2, que posteriormente se incorporó al Departamento de Educación, así como con la experiencia de los profesores asistentes a las formaciones realizadas por BQ, que han dado sus impresiones de los diferentes componentes y prácticas en el entorno de programación Bitbloq.

Esta validación se ha realizado concretamente con 8 expertos del Departamento Educativo de BQ

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.

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.

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

- 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).

La lista completa de formaciones junto con las fechas, se encuentra en el Anexo II Lista de formaciones con Bitbloq y el Kit de robótica, durante 2015. El contenido de las formaciones se encuentra en el Anexo III Guiones sesiones de formación.

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:

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

Componente	Descripcion	<10	10	11	12	13	14	15	>=16
Pulsador	Botón que permite el paso de corriente eléctrica al pulsarlo y no la permite al soltarlo (cierra y abre el circuito).	X							
Sensor de luz	Detecta la cantidad de luz que hay en el entorno		X						
Sensor de infrarrojos	Sensor de infrarrojos: Detecta la diferencia entre colores claros y oscuros. En ocasiones son digitales para diferenciar entre blanco y negro.	X							
Potenciómetro	Permite regular el paso de corriente eléctrica, pudiendo ampliar o reducir el voltaje de la misma.		X						
Joystick	Dispositivo que incorpora 2 potenciómetros y 1 pulsador para regular movimientos en dos direcciones y servir como botón.			X					

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

Componente	**Descripcion**	**<10**	**10**	**11**	**12**	**13**	**14**	**15**	**>=16**
Sensor de ultrasonidos	Detecta la distancia a la que se encuentran los objetos emitiendo una onda ultrasónica y calculando el tiempo que tarda en regresar tras rebotar con algún objeto.		X						
Sensor de temperatura y humedad	Es capaz de medir la temperatura y humedad del ambiente.		X						
Sensor de sonido	Detecta o mide el sonido o ruido ambiental.		X						
LED	Sólo deja pasar la corriente eléctrica en un sentido y la bloquea en el contrario. Cuando deja pasar corriente emite luz.	X							
Zumbador	Zumbador: Emite sonido en la frecuencia elegida, pudiendo reproducir notas musicales, melodías, etc...	X							
Servo de rotación continua	Transforma la corriente eléctrica en movimiento circular.	X							

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

Componente	**Descripcion**	**<10**	**10**	**11**	**12**	**13**	**14**	**15**	**>=16**
Servo (mini-servo)	Transforma la corriente eléctrica en movimiento circular, pudiéndose elegir el ángulo de rotación, entre 0 º a 180 º		X						
Pantalla LCD	Permite mostrar información como imágenes o texto.				X				

Tabla 6. 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. La observación se registrará en la siguiente tabla, donde se anotará la comprensión que adquieren de los propios componentes y del uso que son capaces de dar:

Tabla 7. Validación de componentes de robótica.

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

Componente	**Edad recomendada**	**Observaciones**
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	
Pantalla LCD	13	

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

5 Conclusiones y Objetivos del estudio

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, 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.

Tras el análisis realizado, ¿cuáles son los objetivos que se proponen para este proyecto? Se quiere construir una plataforma de acceso universal donde cualquier persona pueda aprender robótica y programación 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 a ojos de los usuarios de la población objetivo, que resuman las funcionalidades de los componentes del kit de robótica y permita adquirir la lógica de programación. 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

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

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 180º, 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.

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

6 Bibliografía

Aguilar, M., Navarro, J. I., López, J. M. y Alcalde, C. (2002). Pensamiento formal y resolución de problemas matemáticos. Psicothema, Vol. 14, Nº 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, Nº 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 form conceptual structures by mental models manipulation.

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

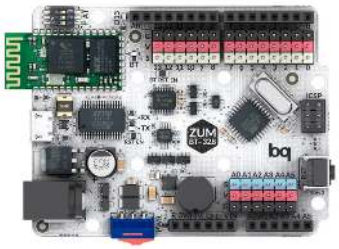
Memory and Cognition, N.º 25, pp. 250-261.

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

7 Anexo I Maletín de electrónica de BQ

El matelín de electrónica de BQ está especialmente preparado para el entorno educativo:

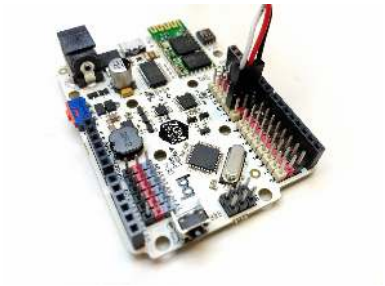
Incluye una placa controladora ZUB BT-328:



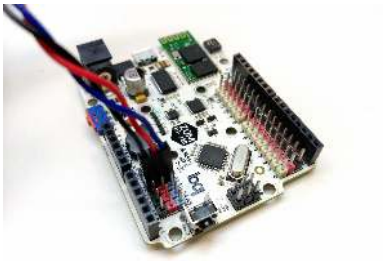
La placa, basada en Arduino, tiene preparadas las conexiones para conectar directamente los pines de los sensores y actuadores tanto digitales como analógicos:

Ejemplo de conexión de componentes digitales:

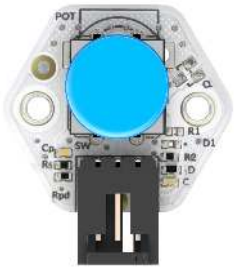
Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020



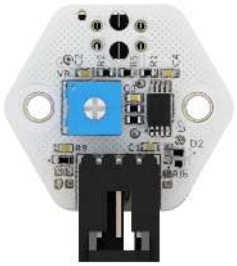
Ejemplo de conexión de componentes analógicos:



El maletín incluye los siguientes componentes que serán objeto de estudio:

Componente	**Imagen**
LED	
Pulsador	

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

Componente	Imagen
Zumbador	
Sensor IR	
Servo de rotación continua	
Sensor de Luz	

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

Componente	Imagen
Potenciómetro	
Servo (mini servo)	
Sensor de ultrasonidos	
Joystick	
Pantalla LCD	

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

8 Anexo II Lista de formaciones con Bitbloq y el Kit de robótica, durante 2015

Fecha	Descripción	Provincia
2015/3/2	Formación robótica Comunidad de Madrid	Madrid
2015/4/9	Introducción a la robótica y la programación	Madrid (Sur)
2015/5/4	Experto Universitario UNIR (PER1)	Global
2015/5/11	Introducción a la robótica y la programación	Madrid
2015/6/8	Formación robótica Comunidad de Madrid	Madrid
2015/7/6	Formación robótica Comunidad de Madrid	Madrid
2015/7/20	Formación robótica Comunidad de Madrid	Madrid
2015/9/21	Formación robótica Comunidad de Madrid	Madrid
2015/10/6	Formación robótica en IES Federico García Lorca	Madrid
2015/10/16	Formación robótica Alumnos Master UAM	Madrid
2015/10/19	Formación robótica Comunidad de Madrid	Madrid
2015/11/5	Formación Empresas Formadoras de Robótica	Madrid
2015/11/9	Experto Universitario UNIR (PER2)	Global
2015/11/16	Formación robótica Comunidad de Madrid	Madrid
2015/11/21	Sábado Tecnológico en IES Federico García Lorca	Madrid
2015/11/23	Formación robótica Profesorado Galicia	La Coruña
2015/11/27	Formación Facultad Magisterio Leioa	Leioa
2015/11/30	Formación robótica Comunidad de Madrid	Madrid
2015/12/1	Club Rononauta Diciembre IES FGL	Madrid
2015/12/11	Formación Facultad Magisterio Leioa	Leioa
2015/12/11	Taller alumnos Colegio el Prado	Madrid
2015/12/15	Formación Empresas Formadoras de Robótica	Madrid
2015/12/22	Curso de robótica y programación con Arduino y Protocoder	Bilbao

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

9 Anexo III Guiones sesiones de formación

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

SESIÓN 1. INICIACIÓN A LA ROBÓTICA Y PROGRAMACIÓN.

CONTENIDOS

Introducción.
Programando placas controladoras con Bitbloq 2.
Uso de actuadores en Bitbloq 2 (I): LED y zumbador.
Uso de sensores en Bitbloq 2 (I): pulsador y sensor de luz.
Programación de sentencias condicionales y bucles (I): Condicional "Si... Ejecutar".
Comunicación por el puerto serie.

Links Importantes:

http://diwo.bq.com/
http://bitbloq.bq.com/
http://diwo.bq.com/course/aprende-robotica-y-programacion-con-bitbloq-2/

Contactos:

educacion@bq.com
comercial.educacion@bq.com

Ejercicios:

1 LED:

Nivel Básico: Crear un programa para encender el LED y otro para apagarlo.

Nivel Medio: Hacer parpadear un LED cada 2 segundos.

Nivel Avanzado: Hacer parpadear un LED con una frecuencia variable. Empezará en 2 segundos, luego la mitad y así sucesivamente hasta que el parpadeo sea tan rápido que parezca que siempre está encendido.

Documentación: *http://diwo.bq.com/y-la-luz-se-hizo-programando-leds-en-bitbloq-2/*

2 LED:

Nivel Básico: Hacer parpadear los dos LEDs a la vez.

Nivel Medio: Hacer parpadear los dos LEDs alternativamente.

Nivel Avanzado: Continuación del ejercicio avanzado de "1 LED". Mejorarlo para que lo haga simultáneamente en ambos LEDs.

Documentación: *http://diwo.bq.com/y-la-luz-se-hizo-programando-leds-en-bitbloq-2/*

PULSADOR, VARIABLES Y CONDICIONAL IF::

Nivel Básico: Crear una variable para almacenar el valor del pulsador y encender los LEDs cuando pulsemos el botón.

Nivel Medio: Hacer que los LEDs parpadeen cuando pulsemos el botón.

Nivel Avanzado: Realizar un programa que al pulsar el botón, encienda un LED y apague otro y al soltar el botón encienda y apague el contrario.

Documentación: *http://diwo.bq.com/si-tu-me-dices-ven-sentencias-condicionales-y-el-pulsador/*
http://diwo.bq.com/programando-con-variables-en-bitbloq-2/

SENSOR DE LUZ:

Nivel Básico: Automatizar una farola de la calle: hacer que un LED se encienda cuando haya poca luz y se apague cuando vuelva a haber luz.

Nivel Medio: Igual que el ejercicio del nivel básico, pero sólo funciona si estamos pulsando el botón.

Documentación: *http://diwo.bq.com/ve-hacia-la-luz-robotin-programando-el-sensor-de-luz/*

ZUMBADOR:

Nivel Básico: Hacer que suene una nota.

Nivel Medio: Realizar una melodía.

Nivel Avanzado: Hacer una gaita electrónica. Consiste en usar el sensor de luz para decirle qué nota tocar al zumbador, de forma que según qué luz haya sobre el sensor, el zumbador tocará una nota diferente.

Documentación: *http://diwo.bq.com/piruru-piii-programando-el-zumbador/*
http://latecladeescape.com/h/2015/08/frecuencia-de-las-notas-musicales

SERIAL MONITOR:

Nivel Básico: Visualizar el valor del sensor de luz o del pulsador por el puerto serie.

Nivel Medio: Al enviar “A” por el puerto serie se encienda un LED y al enviar “B” se apague.

Documentación: *http://diwo.bq.com/comunicandose-con-puerto-serie-bitbloq2/*

SESIÓN 2. INICIACIÓN A LA ROBÓTICA Y PROGRAMACIÓN. .

CONTENIDOS

Repaso del tema anterior.
Uso de sensores en Bitbloq 2 (II): Potenciómetro y Sensor de ultrasonidos.
Uso de actuadores en Bitbloq 2 (II): Servo de rotación continua y Miniservo.
Programación de sentencias condicionales y bucles (II): Bucle "Contar".
Máquina de estados y Funciones.

Links Importantes:

http://diwo.bq.com/
http://bitbloq.bq.com/
http://diwo.bq.com/course/aprende-robotica-y-programacion-con-bitbloq-2/

Contactos:

educacion@bq.com
comercial.educacion@bq.com

Ejercicios:

SERVO DE ROTACIÓN CONTINUA:

Nivel Básico: Mover en alguna dirección el servo de rotación continua.

Nivel Medio: Mover el servo de rotación continua primero en una dirección, luego parar y luego en la otra dirección, en el mismo programa.

Nivel Avanzado: Cambiar el estado del servo de rotación continua mediante el pulsador. Cada vez que se pulsa, cambia de estado.

Documentación: *http://diwo.bq.com/muevete-el-servo-de-rotacion-continua/*

POTENCIÓMETRO:

Nivel Básico: Crear una variable para almacenar el valor del potenciómetro y modificar la velocidad de parpadeo de un LED.

Nivel Medio: Variar el tono con el que suena el zumbador mediante el potenciómetro.

Nivel Avanzado: Encender 5 LEDs colocados en línea progresivamente mediante el potenciómetro, de tal forma que cuando el potenciómetro esté en un extremo no haya ningún LED encendido y si está en el extremo contrario, los 5 LEDs estén iluminados, encendiéndose proporcionalmente al giro efectuado.

Documentación: *http://diwo.bq.com/programando-el-potenciometro-con-bitbloq-2/*

MINISERVO Y MAPEO DE VARIABLES:

Nivel Básico: Mover el miniservo a distintos ángulos con diferentes programas (uno para cada ángulo).

Nivel Medio: Usar el potenciómetro para mover el miniservo, mapeando la variable.

Nivel Avanzado: Hacer el ejercicio del nivel medio con un miniservo y controlar el otro de forma similar mediante el sensor de luz.

Documentación: *http://diwo.bq.com/la-potencia-sin-control-el-miniservo/*

BUCLE "CONTAR":

Nivel Básico: Hacer un programa que se la ligue al escondite. Tiene que parpadear 20 veces (1 por segundo) y después emitir un pitido con el zumbador.

Nivel Medio: Construir una línea de LEDs (compuesta por 5 LEDs) e ir encendiendo uno a uno mediante un bucle "contar".

Documentación: *http://diwo.bq.com/bucles-bucles-bucles-bucles-bu-los-bucles-de-control/*

SENSOR DE ULTRASONIDOS:

Nivel Básico: Ver mediante el puerto serie el valor del sensor de ultrasonidos.

Nivel Medio: Crear un avisador de choque de marcha atrás de los coches. Al pulsar el botón (que representa la marcha atrás), el zumbador emitirá pitidos que sonarán más rápido cuanto más cerca estemos del sensor de ultrasonidos.

Nivel Avanzado: Hacer un Theremin.

Documentación: *http://diwo.bq.com/abre-los-ojos-el-sensor-de-ultrasonidos/*

FUNCIONES:

Nivel Básico: Definir una función que haga sonar una melodía mediante el zumbador. Esta melodía deberá tener una duración variable, que se especificará mediante un argumento de la función.

Nivel Medio: Definir una función que calcule la hipotenusa recibiendo como argumentos los catetos. Probar que funciona enviando el valor de retorno por el puerto serie.

Documentación: *http://diwo.bq.com/funciones-sin-retorno-bitbloq-2/*

MÁQUINA DE ESTADOS:

Nivel Básico: Realizar un programa que al realizar una pulsación del botón (pulsar y soltar) encienda el LED, al realizar otra pulsación parpadee, a la siguiente pulsación se apague, y a la siguiente vuelva a empezar (es decir, lo encienda de nuevo y continúe igual).

10 Anexo IV Guía taller con el Kit

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

Kit de robótica, cómo hacer un taller de 3 horas

bq

El contenido de este documento es de apoyo a la formación.

Contenido

- Objetivos generales
- Objetivos pedagógicos
- ¿Qué necesito definir?
- Listado de actividades
- Definición de actividades
- Recomendaciones para un taller perfecto
- Uso básico del kit de robótica para el taller
- Preguntas frecuentes del producto

El contenido de este documento es de apoyo a la formación.

Objetivos generales

- Despertar la curiosidad por la electrónica y los productos electrónicos
- Despertar el interés por la programación

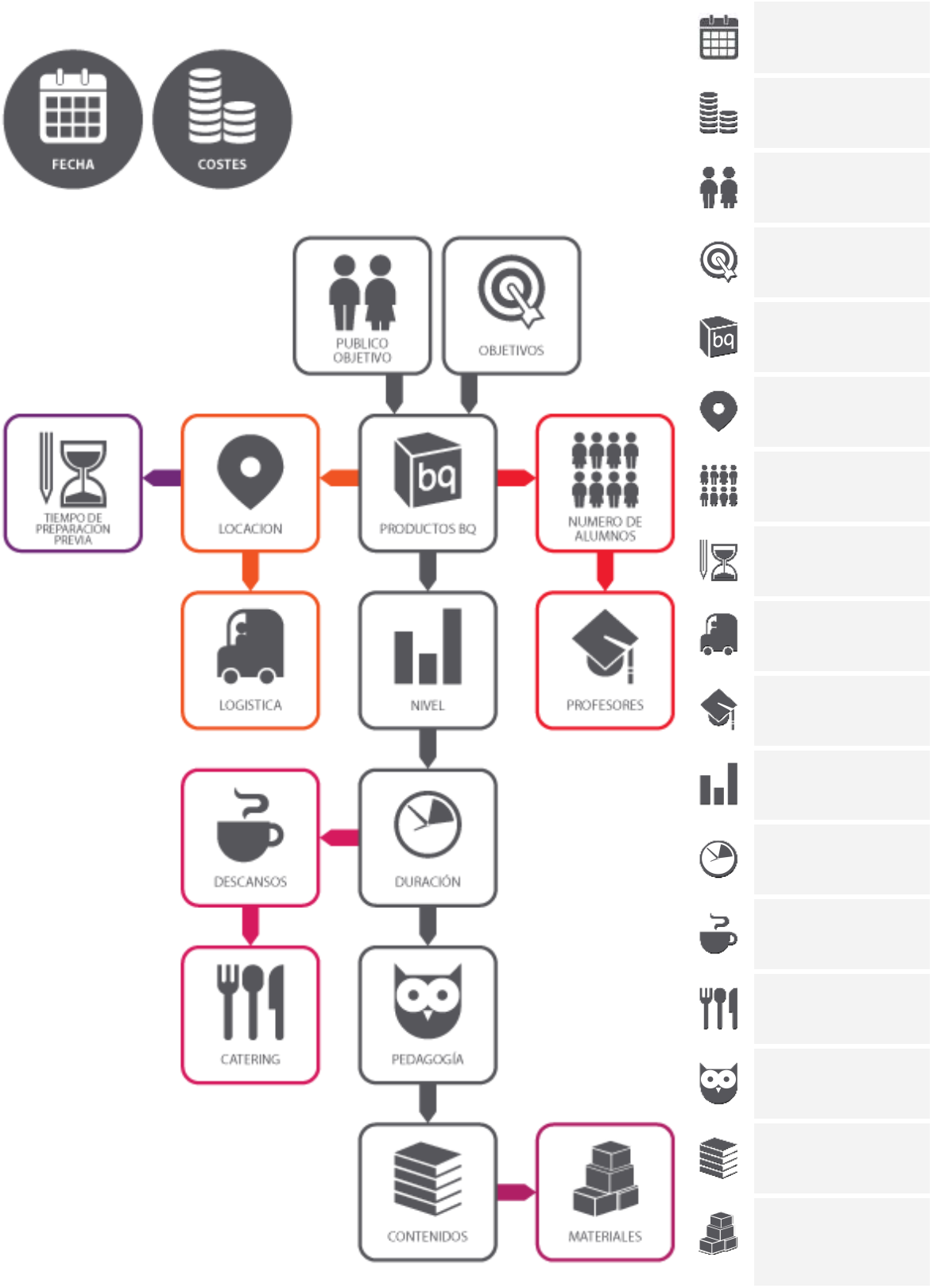
Objetivos pedagógicos

- Aprender a programar placas controladoras con Bitbloq 2
- Uso de actuadores en Bitbloq 2: LED, zumbador, servo de rotación continua y mini servo
- Uso de sensores en Bitbloq 2: pulsador, sensor de luz, potenciómetro y sensor de ultrasonidos
- Programación de sentencias condicionales y bucles: Condicional “Si... Ejecutar” y bucle “Contar”
- Comunicación por el puerto serie
- Máquina de estados y Funciones

El contenido de este documento es de apoyo a la formación.

¿Qué necesito definir?

En este documento encontrarás una guía de todos los aspectos que debes definir cuando plantees realizar un taller.



El contenido de este documento es de apoyo a la formación.

Listado de actividades

Detalle de actividades con tiempos para 3 horas:

Entorno	Actividad	Duración
Desarrollo del taller	Cortesía	5 minutos
Desarrollo del taller	Presentación del taller	5 minutos
Desarrollo del taller	Nos presentamos todos	5 minutos
Robótica general	Sensores y actuadores, software, hardware, diseño 3D	10 minutos
Placa controladora	Placa controladora: pines analógicos y digitales	5 minutos
Kit de robótica	LED	15 minutos
Kit de robótica	Pulsador, variables y condicional IF	15 minutos
Kit de robótica	Sensor de luz	15 minutos
Kit de robótica	Serial monitor	15 minutos
Desarrollo del taller	Descanso	10 minutos
Kit de robótica	Zumbador	15 minutos
Kit de robótica	Servo de rotación continua	15 minutos
Kit de robótica	Potenciómetro	15 minutos
Kit de robótica	Miniservo y mapeo de variables	15 minutos
Kit de robótica	Bucle ‘contar’	15 minutos
Desarrollo del taller	Resumen taller	5 minutos

En caso de disponer de más tiempo para la realización del taller, se pueden programar también los siguientes ejemplos en Bitbloq

Kit de robótica	Sensor de ultrasonidos	15 minutos
Kit de robótica	Funciones	15 minutos
Kit de robótica	Máquina de estados	15 minutos

El contenido de este documento es de apoyo a la formación.

Definición de las actividades

Robótica general

Entre todos los componentes del kit tenemos sensores (o periféricos de entrada) y actuadores (o periféricos de salida), encargados de interactuar con el entorno.

Sensor: dispositivo capaz de leer condiciones del entorno y transformarlas en señales eléctricas.

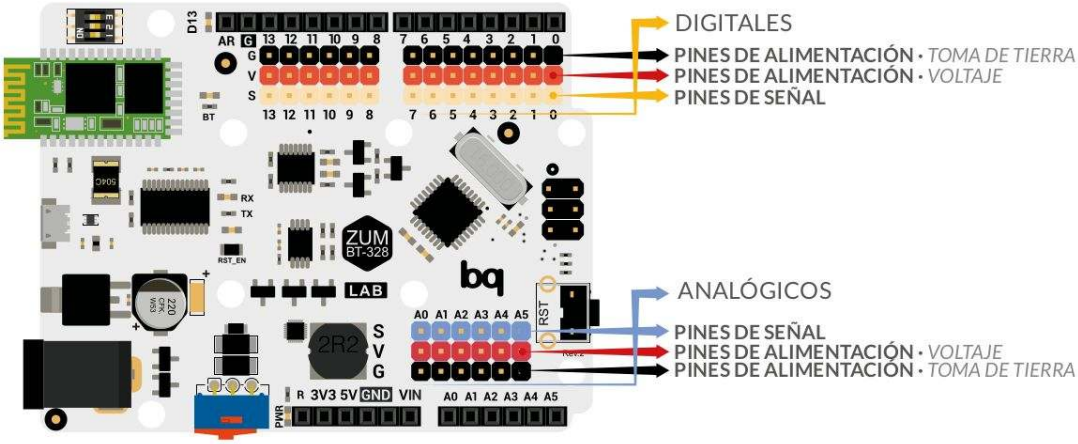
Actuador: dispositivo capaz de transformar señales eléctricas en un proceso físico, como luz, sonido o movimiento con la finalidad de realizar un cambio. Los actuadores sólo podrán funcionar a través de los pines digitales. Para hacerlo, controla los actuadores modulando la salida digital, es decir, el tiempo pasa a ser un factor importante en los ciclos que se den.

LISTADO COMPONENTES KIT EDUCATIVO	
SENSORES	**ACTUADORES**
Pulsador/botón Potenciómetro Sensor IR Sensor de luz Sensor Ultrasonidos Joystick Botonera	LED Miniservos Zumbador Servo Rotación Continua Pantalla LCD

Placa controladora

Pines digitales y pines analógicos. Los pines son las entradas y salidas de casi toda la información que pasa por la placa. Sirven para conectar los periféricos (sensores o actuadores) con los que vamos a crear nuestros proyectos.

El contenido de este documento es de apoyo a la formación.



Hay 2 tipos de pines: digitales y analógicos.

1- Pines digitales: Pueden funcionar como entradas (conectando un sensor) o como salidas (conectando un actuador). Los pines digitales se corresponden con la ristra larga de pines de la placa, van del 0 al 13.

En el caso de conectar un sensor, éste mandará un valor de 0 Voltios (LOW o 0 lógico) o 5 Voltios (HIGH o 1 lógico), dicho valor será leído por el microcontrolador.

En caso de conectar un actuador, será el micro el que escriba el valor de 0 Voltios o 5 Voltios, encendiendo o apagando el actuador correspondiente. Dado que sólo pueden leerse o escribirse dos valores se suele decir que escriben 0 o 1, TRUE o FALSE, LOW o HIGH. En cualquier caso, siempre se refiere a los valores de 0 Voltios o 5 Voltios.

Se recomienda evitar usar los pines 0 y 1, ya que son los mismos que se usan para la comunicación de la placa con el ordenador y podría darse un comportamiento inesperado.

2- Pines analógicos: Los pines analógicos en las placas de la familia Arduino sólo se utilizan como entradas, es decir, sólo podremos conectar sensores. En lugar de poder leer 0 Voltios o 5 Voltios, por ejemplo, podrían leer un valor de 3 Voltios. Estas lecturas se traducen al microcontrolador con un valor entre 0 y 1023. Es decir, si lee 0 Voltios, al micro llegaría un 0. En cambio, si lee 5 Voltios llegaría 1023. En las placas Arduino UNO o ZUM BT 328 existen 6 pines analógicos numerados del 0 a 5.

Los pines de señal (**S**=Signal) son los blancos (en los digitales) y los azules (en los analógicos). Los pines de alimentación son los rojos (**V**=Voltaje) y los negros (**G**=Ground=toma de tierra).

El contenido de este documento es de apoyo a la formación.

Las placas controladoras Arduino también se pueden comunicar con el exterior usando el puerto USB o el Bluetooth. Estos sirven para cargar el programa en la memoria del microprocesador o para comunicarnos con la placa a través de comandos.

Resumiendo:

- ***Microprocesador:*** *es el microchip que realiza todas las operaciones necesarias para que tu placa ejecute tu programa.*
- ***Pines digitales:*** *solo pueden tener dos valores: 1 o 0, todo o nada, encendido o apagado.*
- ***Pines analógicos:*** *pueden tener múltiples valores: 0, 1, 34, 255, etc.*
- ***Entrada micro-USB:*** *a través de esta entrada la placa se comunica con el ordenador.*
- ***Entrada de baterías:*** *la placa recibe energía de las baterías o la fuente de alimentación para poder funcionar.*
- ***Bluetooth:*** *permite a la placa comunicarse sin cables con un ordenador, smartphone u otros.*
- ***Botón ON/OFF:*** *quita la energía a los pines de tu placa de forma que puedas programar sin que el robot esté continuamente moviéndose.*
- ***Reguladores de tensión:*** *adaptan la tensión de alimentación al voltaje en que trabaja la placa: 5V.*

Kit de robótica

Las actividades de programación planteadas en el listado de actividades se explican a continuación de manera detallada:

1 LED:

Nivel Básico: Crear un programa para encender el LED y otro para apagarlo.

Nivel Medio: Hacer parpadear un LED cada 2 segundos.

Nivel Avanzado: Hacer parpadear un LED con una frecuencia variable. Empezará en 2 segundos, luego la mitad y así sucesivamente hasta que el parpadeo sea tan rápido que parezca que siempre está encendido.

El contenido de este documento es de apoyo a la formación.

Documentación:
http://diwo.bq.com/y-la-luz-se-hizo-programando-leds-en-bitbloq-2/

2 LED:

Nivel Básico: Hacer parpadear los dos LEDs a la vez.

Nivel Medio: Hacer parpadear los dos LEDs alternativamente.

Nivel Avanzado: Continuación del ejercicio avanzado de "1 LED". Mejorarlo para que lo haga simultáneamente en ambos LEDs.

Documentación:
http://diwo.bq.com/y-la-luz-se-hizo-programando-leds-en-bitbloq-2/

PULSADOR, VARIABLES Y CONDICIONAL IF::

Nivel Básico: Crear una variable para almacenar el valor del pulsador y encender los LEDs cuando pulsemos el botón.

Nivel Medio: Hacer que los LEDs parpadeen cuando pulsemos el botón.

Nivel Avanzado: Realizar un programa que al pulsar el botón, encienda un LED y apague otro y al soltar el botón encienda y apague el contrario.

Documentación:
http://diwo.bq.com/si-tu-me-dices-ven-sentencias-condicionales-y-el-pulsador/
http://diwo.bq.com/programando-con-variables-en-bitbloq-2/

SENSOR DE LUZ:

Nivel Básico: Automatizar una farola de la calle: hacer que un LED se encienda cuando haya poca luz y se apague cuando vuelva a haber luz.

Nivel Medio: Igual que el ejercicio del nivel básico, pero sólo funciona si estamos pulsando el botón.

Documentación:
http://diwo.bq.com/ve-hacia-la-luz-robotin-programando-el-sensor-de-luz/

SERIAL MONITOR:

Nivel Básico: Visualizar el valor del sensor de luz o del pulsador por el puerto serie.

El contenido de este documento es de apoyo a la formación.

Nivel Medio: Al enviar “A” por el puerto serie se encienda un LED y al enviar “B” se apague.

Documentación:
http://diwo.bq.com/comunicandose-con-puerto-serie-bitbloq2/

ZUMBADOR:

Nivel Básico: Hacer que suene una nota.

Nivel Medio: Realizar una melodía.

Nivel Avanzado: Hacer una gaita electrónica. Consiste en usar el sensor de luz para decirle qué nota tocar al zumbador, de forma que según qué luz haya sobre el sensor, el zumbador tocará una nota diferente.

Documentación: *http://diwo.bq.com/piruru-piii-programando-el-zumbador/*

http://latecladeescape.com/h/2015/08/frecuencia-de-las-notas-musicales

SERVO DE ROTACIÓN CONTINUA:

Nivel Básico: Mover en alguna dirección el servo de rotación continua.

Nivel Medio: Mover el servo de rotación continua primero en una dirección, luego parar y luego en la otra dirección, en el mismo programa.

Nivel Avanzado: Cambiar el estado del servo de rotación continua mediante el pulsador. Cada vez que se pulsa, cambia de estado.

Documentación: *http://diwo.bq.com/muevete-el-servo-de-rotacion-continua/*

POTENCIÓMETRO:

Nivel Básico: Crear una variable para almacenar el valor del potenciómetro y modificar la velocidad de parpadeo de un LED.

Nivel Medio: Variar el tono con el que suena el zumbador mediante el potenciómetro.

Nivel Avanzado: Encender 5 LEDs colocados en línea progresivamente mediante el potenciómetro, de tal forma que cuando el potenciómetro esté en un extremo no haya ningún LED encendido y si está en el extremo

El contenido de este documento es de apoyo a la formación.

contrario, los 5 LEDs estén iluminados, encendiéndose proporcionalmente al giro efectuado.

Documentación:
http://diwo.bq.com/programando-el-potenciometro-con-bitbloq-2/

MINISERVO Y MAPEO DE VARIABLES:

Nivel Básico: Mover el miniservo a distintos ángulos con diferentes programas (uno para cada ángulo).

Nivel Medio: Usar el potenciómetro para mover el miniservo, mapeando la variable.

Nivel Avanzado: Hacer el ejercicio del nivel medio con un miniservo y controlar el otro de forma similar mediante el sensor de luz.

Documentación: *http://diwo.bq.com/la-potencia-sin-control-el-miniservo/*

BUCLE "CONTAR":

Nivel Básico: Hacer un programa que se la ligue al escondite. Tiene que parpadear 20 veces (1 por segundo) y después emitir un pitido con el zumbador.

Nivel Medio: Construir una línea de LEDs (compuesta por 5 LEDs) e ir encendiendo uno a uno mediante un bucle "contar".

Documentación:
http://diwo.bq.com/bucles-bucles-bucles-bucles-bu-los-bucles-de-control/

SENSOR DE ULTRASONIDOS:

Nivel Básico: Ver mediante el puerto serie el valor del sensor de ultrasonidos.

Nivel Medio: Crear un avisador de choque de marcha atrás de los coches. Al pulsar el botón (que representa la marcha atrás), el zumbador emitirá pitidos que sonarán más rápido cuanto más cerca estemos del sensor de ultrasonidos.

El contenido de este documento es de apoyo a la formación.

Nivel Avanzado: Hacer un Theremín.

Documentación: *http://diwo.bq.com/abre-los-ojos-el-sensor-de-ultrasonidos/*

FUNCIONES:

Nivel Básico: Definir una función que haga sonar una melodía mediante el zumbador. Esta melodía deberá tener una duración variable, que se especificará mediante un argumento de la función.

Nivel Medio: Definir una función que calcule la hipotenusa recibiendo como argumentos los catetos. Probar que funciona enviando el valor de retorno por el puerto serie.

Documentación: *http://diwo.bq.com/funciones-sin-retorno-bitbloq-2/*

MÁQUINA DE ESTADOS:

Nivel Básico: Realizar un programa que al realizar una pulsación del botón (pulsar y soltar) encienda el LED, al realizar otra pulsación parpadee, a la siguiente pulsación se apague, y a la siguiente vuelva a empezar (es decir, lo encienda de nuevo y continúe igual).

El contenido de este documento es de apoyo a la formación.

Recomendaciones para un taller perfecto

Aquí encontrarás un plan de sugerencias generales de un taller, útil para todo el que vaya a dar una clase.

- ☐ Cargar todo el material previo al taller
- ☐ Regletas
- ☐ Llevar siempre cargadores por si acaso
- ☐ Tener a mano claves de wifi
- ☐ Llevar cables HDMI o VGA
- ☐ Saber si hay proyectores
- ☐ Tiempo de preparación
- ☐ Condiciones de sonido
- ☐ sala sin otras formaciones al lado
- ☐ Los padres cerca desconcentran a los niños
- ☐ Descansos si es muy largo
- ☐ Si es amenaza, llevar personas que contesten preguntas mientras los ponentes dan el taller, para que no les molesten preguntando
- ☐ Cómo solucionar peleas por compartir material entre los niños
- ☐ Espacio suficiente en las mesas para manipular el material
- ☐ Llevar toda la documentación preparada
- ☐ Capacidad de adaptarse a cambios inesperados de número de niños, nivel, falta de internet
- ☐ Cómo tratar a un alumno que no presta atención o molesta a los demás
- ☐ Lleva un móvil con datos para hacer en puente wifi en caso de que no haya internet

El contenido de este documento es de apoyo a la formación.

El contenido de este documento es de apoyo a la formación.

Uso básico del Kit de robótica para el taller

Antes de ponerte a programar con el Kit de electrónica, deberás saber:

Antes de empezar a programar con Bitbloq 2 es importante asegurarse que tenemos todo lo necesario para que éste funcione correctamente en nuestro ordenador.

Sigue los siguientes pasos:

1- Conecta tu placa al ordenador con el cable USB.

2- Abre Bitbloq 2 en tu navegador con Google Chrome.

3- Pulsa sobre la pestaña Placas y arrastra tu placa al recuadro blanco. Si tienes una placa distinta a la que aparece en el listado compatible con la familia UNO, utiliza la placa Arduino UNO.

4- Pulsa en Verificar y verás cómo compila correctamente el programa.

5- Pulsa sobre el botón Cargar, aparecerá un cuadro de diálogo pidiéndote descargar la aplicación Web2board. Selecciona tu sistema operativo, se bajará un archivo a tu ordenador.

6- Si estás utilizando Windows, descomprime el archivo descargado y ejecuta el archivo *web2board.exe*. En el caso de que estés utilizando Linux descomprime el archivo y ejecuta el archivo Web2Board, se abrirá una aplicación que solicitará que introduzcas tu contraseña de usuario, introdúcela y pulsa intro.

7- Una vez que hayas instalado el programa Web2board es recomendable que reinicies el ordenador.

Tras reiniciar el ordenador, abre Bitbloq 2, selecciona tu placa y pulsa sobre el botón Cargar. Se empezará a cargar el programa a tu placa y tras unos segundos, aparecerá en pantalla el mensaje que te indicará que todo ha funcionado correctamente: Programa subido con éxito.

El contenido de este documento es de apoyo a la formación.

11 Anexo V Imágenes de formaciones con Bitbloq y el Kit de robótica, durante 2015

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