



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



Centro para el
Desarrollo
Tecnológico
Industrial



UNIÓN EUROPEA

Fondo Europeo de
Desarrollo Regional (FEDER)

Una manera de hacer Europa

ENTREGABLE E.2.3.2 RESULTADOS DE LA VALIDACIÓN DE LA APLICACIÓN DESARROLLADA

RESUMEN DEL DOCUMENTO

Participantes de educación primaria y secundaria fueron observados mientras experimentaban con el kit de robótica y Bitbloq. De los resultados se puede concluir que los participantes saben expresar las 4 habilidades estudiadas y cambian la estrategia cuando adquieren práctica. No obstante, se han observado algunas dificultades (más persistentes en los más pequeños)

February 3, 2016

Contents

1	Introducción	2
2	Actividades planteadas en la investigación	3
3	Resultados	5
3.1	Habilidades en programación	5
3.1.1	Estrategias en solución de problemas	5
3.1.2	Algoritmos y secuencias	6
3.1.3	Funciones	6
3.1.4	Condicionales	7
3.1.5	Bucles	7
3.2	Componentes del Kit de robótica	8
4	Conclusiones	17
4.1	Habilidades de programación	17
4.1.1	Resolución de problemas	17
4.1.2	Algoritmos	18
4.1.3	Funciones	18
4.1.4	Condicionales	19
4.1.5	Bucles	19
4.2	Componentes electrónicos	20
4.2.1	Pulsador, sensor IR y sensor ultrasonidos:	20
4.2.2	Componentes digitales vs. analógicos	21
4.2.3	Influencia de los conocimientos previos y la edad	22
5	Discusión	24
6	Anexo I Diario de la Experiencia en el Colegio Balder	27
7	Anexo II Imágenes de las formaciones a alumnos con el Kit de Robótica	34

1 Introducción

En base a la teoría revisada y explicada en el documento E 2.1., se ha realizado un estudio donde se ha probado el kit de robótica en diferentes talleres a una población objetivo entre 10 y 16 años realizando actividades en las que se involucraron el uso de los distintos componentes para observar cómo los participantes son capaces de expresar algoritmos, funciones, condicionales y bucles. Se han realizado diferentes talleres y proyectos para observar la forma en que los participantes de diferentes edades expresan algoritmos, funciones, condicionales y bucles con el fin de determinar cuál puede ser la mejor etapa en la que iniciar la enseñanza de la programación.

2 Actividades planteadas en la investigación

En general, se han planteado muchas actividades y proyectos donde todos los participantes experimentaban con los diferentes componentes del kit de robótica. Realmente, se han usado casi todos los componentes en los diferentes tipos de tareas que debían cumplir donde expresaban algoritmos, funciones, condicionales y bucles.

Un ejemplo de actividad para algoritmos consistía en iluminar una serie de LED que se iluminan de forma alterna. La dificultad de la tarea radicaría en la cantidad de LED usados y la forma de hacer que se alterne su encendido y apagado.

En cuanto a los condicionales, se han propuesto actividades en las que se usan sensores como el IR o el ultrasonidos, el botón o la botonera. Igualmente, la dificultad radica en la forma en que usan los componentes y la cantidad los y tipos de condicionales que son capaces de usar.

En cuanto a las funciones y los bucles, sigue la misma estela el planteamiento, creando diferentes actividades con diferentes componentes en base a unos objetivos para poder observar cómo lo expresan y cómo usan el componente. Ejemplos donde se crean funciones es la elaboración de un sigue líneas que debe cumplir unas características concretas (por ejemplo, una máquina que lleve comida a diferentes puntos cuando ésta se agote).

En conclusión, para medir el nivel de programación de los participantes se usaron diferentes actividades y se propusieron problemas a resolver. De forma cualitativa, se evaluaron los resultados observando el uso que le daban a los componentes del kit de robótica, y la forma en que todos los participantes expresaban algoritmos, condicionales, funciones y bucles. Para los algoritmos se tenían en cuenta aspectos como la longitud del propio algoritmo y la eficacia de éste a la hora de cumplir el objetivo propuesto. En el caso de las funciones se siguió la misma estela que en los algoritmos, observando en qué modo la programación era eficaz para cumplir el objetivo propuesto y la forma de combinar el uso de diferentes componentes. Los condicionales se evaluaron en la forma en que lograban aplicar diferentes bloques control, y cuántos condicionales eran capaces de usar en una programación concreta. Por último, en los bucles, se tuvo en cuenta todos los aspectos pertinentes para la evaluación descritos añadiendo la eficacia en la forma de iterar. Toda la evaluación consistió en la observación sacando conclusiones a

través de la forma en que los participantes abordaban las diferentes tareas, dándole un carácter más cualitativo tanto en la forma de obtener los datos como en la interpretación de los resultados. Igualmente, la validación de las estrategias tomadas por los alumnos, se realizó por observación de la propia conducta para validar si cumplían con un tipo u otro de las estrategias en resolución de problemas descritas en el documento E 2.1.

3 Resultados

A continuación se muestran los datos que se observaron durante el transcurso de las actividades realizadas a los participantes a lo largo de los diferentes talleres:

3.1 Habilidades en programación

3.1.1 Estrategias en solución de problemas

Mientras los participantes realizaban las tareas propuestas durante los talleres, se observaron los siguientes detalles:

- Identificar alguna estrategia.
- Observar si se cambia de un tipo de estrategia a otra.

Validación de estretagias	
Edad	Observaciones
10-16	<p>En las estrategias se ha observado que todos los participantes han experimentado un proceso similar cuando han comenzado a experimentar con los componentes del kit de robótica y Bitbloq. Lo normal es que los alumnos, independientemente de la edad, mantengan una estrategia de búsqueda aleatoria donde buscan de forma desordenada la solución a los problemas que se les exponía, dando por casualidad con el contenido adecuado para dar solución a los primeros problemas con los que se enfrentan. Con la experiencia y la práctica, el cambio que se produce es a una estrategia de ensayo y error, donde igualmente van probando diferentes bloques de programación hasta que dan con el adecuado, pero a través de una búsqueda ordenada y planificada.</p> <p>Algunos alumnos, no pasaban por la primera fase de búsqueda aleatoria y comenzaban directamente en estrategias que requieren una búsqueda con cierta planificación, pero no es lo común, sobre todo en los alumnos de 10 y 11 años.</p>

Tabla 1. Resultados en estrategias.

3.1.2 Algoritmos y secuencias

En la observación que se hizo sobre la forma en que los participantes abordaban la tarea de crear algoritmos con los diferentes componentes del kit de robótica se tuvieron en cuenta los siguientes detalles:

- Observar si existen dificultades al ampliar el algoritmo al aumentar el recorrido.
- Observar los pasos que siguen para completar un mismo recorrido.

Validación de algoritmos	
Edad	Observaciones
10-16	Los alumnos entienden bien el uso de los algoritmos independientemente de la edad. La edad parece influir sólo en la longitud, calidad, o cantidad de componentes utilizados, pero el concepto de algoritmo y establecer secuencias es una tarea fácil de comprender a cualquier edad. Siendo los participantes de mayor edad (a partir de 14 años) los más capaces de establecer algoritmos largos y que incluyan la combinación de varios componentes.

Tabla 2. Resultados en algoritmos.

3.1.3 Funciones

En la observación que se hizo en los diferentes talleres sobre la forma en que los participantes crearon funciones usando los componentes del kit de robótica se tuvieron en cuenta los siguientes detalles:

- Observar si existen dificultades y qué tipo de dificultades.
- Observar la complejidad de la función.

Validación de funciones

Edad	Observaciones
10-16	Las funciones no han supuesto ninguna dificultad, pues es también una secuencia que los participantes construyen con un objetivo más definido.

Tabla 3. Resultados en funciones.

3.1.4 Condicionales

En la observación que se hizo en los diferentes talleres sobre la forma en que los participantes crearon condiciones, se tuvieron en cuenta los siguientes detalles:

- Cantidad de condicionales que saben poner.
- Uso de diferentes bloques de control.

Validación de condicionales	
Edad	Observaciones
10-16	Los condicionales han supuesto la observación de las dificultades más notables. Como bien se dijo en el apartado 2.1., se trata de la conectiva lógica más difícil. Para los más pequeños hace falta una explicación previa que requiera un ejemplo concreto para entender de forma lingüística que tipo de tarea van a ejecutar. Más allá de un IF, los de antes de los 13 años pueden tener dificultades serias, sobre todo usando otros bloques control. Sólo a partir de 14 años han podido usar con mayor facilidad diferentes bloques control para establecer condicionales.

Tabla 4. Resultados en condicionales.

3.1.5 Bucles

En la observación que se hizo en los diferentes talleres sobre la forma en que los participantes creaban bucles, se tuvieron en cuenta los siguientes detalles:

- Longitud y complejidad del bucle.

- Funcionalidad del bucle.

Validación de bucles	
Edad	Observaciones
10-16	Los bucles suponen la puesta en práctica de la expresión tanto de algoritmos, funciones y condicionales. El concepto de que la programación realizada se repite constantemente se entiende perfectamente en cualquiera de las edades de los participantes en los talleres. Las dificultades afloran cuando se comienzan a usar sensores y bloques control, y/o secuencias que requieren hacer una acción con muchos pasos.

Tabla 4. Resultados en bucles.

3.2 Componentes del Kit de robótica

En la observación que se hizo durante los diferentes talleres sobre cómo los participantes experimentaban con los componentes del kit de robótica, se tuvieron en cuenta los siguientes detalles:

- La comprensión que adquieren sobre el funcionamiento del componente que se aprende a usar.
- El uso que son capaces de aplicar al componente.

Validación de componentes del kit de robótica		
Componentes	Edad recomendada	Observaciones
LED	<10	<p>Todos los participantes han identificado fácilmente qué es un componente digital y entender cuales son sus valores. Fácilmente han sabido encenderlo con Bitbloq, y no han tenido dificultad alguna en programar 2 LED. Programando con 3 LED es cuando puede encontrarse alguna dificultad.</p> <p>Los participantes de secundaria que participaron en diferentes talleres y actividades extraescolares vieron que el LED es un elemento sencillo de usar desde su habitual uso como elemento para iluminar. El que sea un diodo ya es algo más complejo. Programarlo y usarlo no les entraña ningún problema, suelen incurrir en el clásico error de no entender que si no programamos su apagado se mantendrá encendido.</p>

Validación de componentes del kit de robótica		
Componentes	Edad recomendada	Observaciones
Pulsador	<10	<p>Los participantes de primaria saben identificar la función en la que deben usar el pulsador, pero ejecutarla les puede resultar más complicado, dando lugar a que sea necesaria una pequeña ayuda.</p> <p>Los participantes de primaria observados en talleres saben identificar la función en la que deben usar el pulsador, pero ejecutarla les puede resultar más complicado, dando lugar a que sea necesaria una pequeña ayuda. La ayuda siempre radica en usar otros tipos de bloques necesarios para su programación como los de control o de variable.</p> <p>En secundaria no tienen ningún problema en usarlo y en meterlo como parte de un condicional para producir un efecto. Encuentran más problemas a la hora de entender a nivel más profundo el funcionamiento que a veces lleva a errores (como por ejemplo pulsar para realizar una única acción y que ocurra más de una vez por el tiempo de ejecución de programa).</p>

Validación de componentes del kit de robótica		
Componentes	Edad recomendada	Observaciones
Zumbador	<10	<p>En primaria, tras la experiencia con los LED y el pulsador, programar el zumbador no supuso ningún tipo de reto.</p> <p>En primaria, tras la experiencia con los LED y el pulsador, programar el zumbador no supuso ningún tipo de reto. El funcionamiento es fácilmente entendible, y en todos los talleres en los que se practicó con niños de menor edad supieron rápidamente hacerlo funcionar y aplicar diferentes tonos.</p> <p>En secundaria el zumbador es igual de sencillo de utilizar que el LED, con la diferencia que el zumbador ya conlleva el tiempo de ejecución y tras ese tiempo deja de emitir sonido y el LED lo tienes que apagar a nivel programación, con lo cual la primera vez buscaban la forma de apagar el zumbador sin entender que sólo sonaría el tiempo programado.</p>

Validación de componentes del kit de robótica		
Componentes	Edad recomendada	Observaciones
Sensor IR	<10	<p>En primaria han tardado un poco en entender el funcionamiento del sensor IR y cómo puede usarse. Haciendo analogía con un botón les ayuda a entender que funcionan detectando blanco o negro en lugar de ser pulsado o no. En primaria, y sobre todo los de menor edad, han tardado un poco en entender el funcionamiento del sensor IR y cómo puede usarse. Haciendo analogía con un botón les ayuda a entender que funcionan detectando blanco o negro en lugar de ser pulsado o no.</p> <p>En secundaria, a nivel uso del mismo entienden que detecta blanco o negro, no entienden bien que en realidad detecta un cambio de estado y tampoco entienden totalmente bien que es un sensor que emite luz por sí solo. Basta hacer la pregunta de si funciona a oscuras para que te contesten “no” y entiendas que no lo tienen tan claro. Tras instruirles en ello entienden mejor cómo funciona a nivel más profundo y cómo usarlo.</p>

Validación de componentes del kit de robótica		
Componentes	Edad recomendada	Observaciones
Servo de rotación continua	<10	<p>Los participantes no han tenido problemas en usar este componente, solo fue necesario explicar que significa “sentido horario” y “sentido antihorario”, , sobre todo con aquellos participantes en talleres dedicados a niños y niñas pertenecientes a primaria o de menor edad en general.</p> <p>En secundaria no tienen problemas para usar los servos de rotación continua aunque si tienen muchos problemas a la hora de programarlos de forma correcta meditando el sentido de giro, lo utilizan y si no es como querían cambian el sentido.</p>
Sensor de Luz	10	<p>En primaria (entre 10 y 12 años), cuando se ha usado este componente, tienen dificultades sobre su funcionamiento por el concepto elemental que tienen de luz. No obstante, cuando se explica que la intensidad de la luz se puede medir, facilita su comprensión rápidamente, pero al tratarse de un componente analógico porta dificultades a la hora de establecer valores.</p> <p>Lo mismo ocurre en secundaria, incluso con los más mayores, pues suele ser el primer componente analógico y eso dificulta un poco su uso. Especialmente a la hora de incluirlo en iteraciones, un fallo típico es poner que una luz se encienda cuando sea de noche con un condicional y en vez de decir “cuando la luz sea menor de XXXX” poner “cuando la luz sea igual a XXXX”, con lo cual no se enciende nada más que en un instante.</p>

Validación de componentes del kit de robótica		
Componentes	Edad recomendada	Observaciones
Potenciómetro	10	En secundaria el uso normal del potenciómetro no tiene mucho problema asociado, pues es simplemente entender los valores que toma y utilizarlos. El problema viene a la hora de mapearlo, les cuesta mucho entender la forma en que se mapea y se establecen los estados por lo que si se quiere mapear un potenciómetro antes de los 14 años vamos a tener problemas de adquisición de concepto por parte de algunos alumnos. Este problema, como es esperable, se ha encontrado también en niños y niñas pertenecientes a los primeros años de la franja de edad observada.
Sensor de temperatura y humedad	10	No usado
Sensor de sonido	10	No usado
Servo (mini servo)	10	Los participantes en general no tienen problemas con él, sobre todo si ya han experimentado anteriormente con otros componentes. No obstante, el uso depende mucho de los conocimientos previos sobre geometría y ángulos. Por lo tanto, el funcionamiento, en términos generales y para todos los grupos de edad, es fácil de entender pero programarlo requiere entender los conceptos de ángulo y grados que son importantes para determinar la eficacia con la que se le da uso.

Validación de componentes del kit de robótica		
Componentes	Edad recomendada	Observaciones
Sensor de ultrasonidos	11	<p>Todos entendieron este sensor de la misma forma que entendieron el sensor IR. La experiencia previa facilitó el uso de este componente, y sabían perfectamente que es capaz de detectar la presencia de objetos y que la distancia puede variar dependiendo de cómo se programe. No obstante, los participantes de 10 años, a pesar de entender el funcionamiento a algunos les cuesta imaginar o entender el concepto de ondas que rebotan y lo aprenden sólo mediante analogía al pulsador. Además, aplicar un uso les costaba más en comparación con aquellos participantes de mayor edad.</p> <p>A partir de 13 o 14 años entienden mejor su funcionamiento al compararlo con la forma de posicionarse de un murciélago. Su uso no les ha supuesto mucho problema. Una programación típica es usarlo como sensor de aparcamiento y aquí a veces tienen problemas al poner un condicional pues ponen la condición de mayor distancia la primera y luego al ir poniendo el resto con un ELSE no se dan cuenta que la primera opción ya incluye las demás, les cuesta entender que hay que ir desde la distancia más pequeña hacia las mayores.</p>

Validación de componentes del kit de robótica		
Componentes	Edad recomendada	Observaciones
Joystick	12	En los grupos de menor edad (10 y 11) no se ha utilizado por su complejidad. En secundaria, o a partir de la edad recomendada, han tenido algunos problemas para usarlo porque intentan que realice funciones simultáneas. El joystick tiene dos potenciómetros y se programan por separado, por lo que no podemos hacer que avance a la vez hacia delante y hacia la derecha, es decir, no podemos avanzar en oblicuo, primero avanzará en una dirección y luego en la perpendicular y la composición de ambas dará la dirección oblicua pero no dará esa sensación salvo que los movimientos sean muy pequeños.
Pantalla LCD	13	No tienen problemas con su uso en secundaria pues las usamos a un nivel muy superficial.

Tabla 6. Resultados en componentes de robótica.

4 Conclusiones

4.1 Habilidades de programación

4.1.1 Resolución de problemas

Cambio de estrategias: En secundaria lo normal al iniciarse en un nuevo contenido o proceso mental es que tiendan a desarrollar una estrategia inexistente, es decir, buscan probar en líneas aleatorias y generalmente no conectadas entre sí. Hoy en día todo elemento tecnológico tiene cierta dosis de desarrollo hacia lo intuitivo por lo que suelen seguir líneas que el propio elemento evoca (por ejemplo conectar elementos de un lenguaje de programación por bloques). Aun así, en un primer acercamiento no se establece un proceso de asimilación del problema y resolución del mismo pensando previamente cómo hacerlo. Cuando la evolución es la correcta y conocen la herramienta o contenido sí que tienden a pensar una forma de abordar el problema y buscan seguir su estrategia replanteando todo aquello que no funcione como habían pensado.

Lo mismo ocurre con los participantes más pequeños, de 10 y 11 años: la estrategia podría considerarse también inexistente pues lo primero que buscan es nada, prueban bloques sin anticipar ningún posible efecto hasta que dan con el resultado planteado en la actividad. Igualmente, cuando conocen la herramienta y van adquiriendo experiencia se puede percibir un cambio en la forma en que abordan los problemas que se les plantea pasando de una estrategia plenamente aleatoria a una más caracterizada por el ensayo y error, donde ya anticipan un planteamiento de lo que necesitan y para seleccionar aquellos elementos que piensan que van a ser útiles para probarlos. La evolución en todas las edades sigue un curso similar, pero la rapidez en la que cambian de estrategia y mejoran la eficiencia puede diferenciarse con la edad, siendo los más mayores los más eficientes. Los participantes de mayor edad, entre 15 y 16 años han disfrutado de un cambio más rápido en el uso de estrategias más eficientes, causadas no sólo por sus mayores capacidades sino también por poseer mayor experiencia en general tanto en tareas de estudio, capacidad de lectura, de análisis y de abstracción.

Un punto importante a tratar en los participantes es el interés y la motivación que han demostrado en la realización de los talleres. Interés y motivación: En

el cambio de estrategia anteriormente mencionado entra a colación la necesidad de estar basada en motivación o interés. Si no hay interés las estrategias frente a nuevos problemas son similares a las iniciales, no se genera una evolución en la forma en que las estrategias mejoran. El cambio en la forma de planificar estrategias y el éxito de las mismas es tremendamente diferente en alumnos con motivación e interés y alumnos que no se vinculan con los objetivos de aprendizaje.

4.1.2 Algoritmos

La base del algoritmo en secundaria la entienden perfectamente. Secuencian bien los sucesos básicos y no tienen problema en establecer eventos que van primero y eventos que van después. Sí que es cierto que les es más difícil el tema de organizarse dentro de un algoritmo que tiene funciones, condicionales y bucles, pero las tareas de organizar secuencias y saber que un orden incorrecto lleva al fracaso la tienen clara desde el principio. Ayuda mucho empezar explicando algoritmos del mundo físico (por ejemplo unas instrucciones de google maps o una receta de cocina) ya que la palabra algoritmo se torna más real y menos “informática”. En alumnos más pequeños (10 y 11 años) se ha visto que manejan fácilmente los algoritmos, pero les cuesta un poco más de tiempo y esfuerzo en lograr visualizar un estado inicial y un estado final, y pueden necesitar realizar algunas pruebas con la programación antes de identificar claramente dichos estados inicial y final.

4.1.3 Funciones

Las funciones sin retorno las entienden perfectamente, no dejan de ser un pequeño algoritmo que puedes ejecutar en cualquier momento y no difiere en absoluto de cualquier algoritmo básico que hayan aprendido a la hora de aprender las funciones. Tienen más problemas en las funciones con retorno, especialmente porque los usos son más complejos de lo que pueden haber adquirido en el momento en que se explican funciones.

4.1.4 Condicionales

Tareas de un sólo condicional: Son tareas que tenemos integradas en nuestro propio lenguaje así que no tienen problemas con ellas. Si llueve me mojo es algo que todo alumno de cualquier edad entiende. Quizá pueden tener mayor problema en el componente que en el uso de un condicional. Los más pequeños (menores a 12 años), con la explicación adecuada, han sabido manejar bien un sólo condicional, dejando las dificultades al componente utilizado o a la exploración de la interfaz en la búsqueda de los bloques adecuados.

Tareas con dos condicionales: cuando los condicionales dan problemas siempre es cuando pueden tener un conflicto al existir dos condicionales o un sólo condicional con varias condiciones (un AND o un OR), ahí a veces no tienen en cuenta que en los ELSE o ELSE IF deben tener en cuenta todas las condiciones iniciales. Los más pequeños han sido los más afectados en este sentido, pues al usar dos condicionales se encontraban con un muro. Esto es algo esperable pues como bien se dijo en la teoría revisada el condicional se trata de la conectiva del lenguaje lógico más difícil de usar para las personas. Además, hay que tener en cuenta que además del esfuerzo mental que supone a menor edad debido a un desarrollo cognitivo menos maduro a la hora de establecer los condicionales, hay que sumar el uso de otros recursos mentales en la exploración, selección y ejecución de bloques que suponen una carga para la memoria operativa. En general, antes de los 13 años de edad es cuando se encuentran mayores dificultades en el desempeño de tareas donde se usan conectivas condicionales.

4.1.5 Bucles

El mayor problema existente es llegar a ese punto donde un bucle es un aporte real frente a un condicional. Al principio tienden a no entender la diferencia sustancial de un bucle y un condicional. Probablemente se deba porque a nivel lenguaje no hay diferencia entre la acción producida por “si llueve, abre el paraguas” y “mientras llueva, abre el paraguas”. Es a partir de los 14 años cuando podemos llevarles al punto de explicar efectos físicos o programaciones donde un bucle tiene más sentido.

Quizá el bucle FOR es más utilizable por su carácter de repetición pero todos los parámetros que hay que poner lo hacen poco útil para niños muy pequeños.

Al nivel primeros pasos las tareas propuestas con la intención de usar bucles suelen ser resueltas con condicionales.

4.2 Componentes electrónicos

Los componentes electrónicos pertenecientes al kit de robótica son muchos y poseen tanto diferencias como similitudes en su funcionamiento y el uso que se le puede aplicar, además de las diferencias encontradas en los participantes en función de la edad.

Existen unos componentes que han resultado ser muy fáciles e intuitivos para todo los participantes sin hacer distinción en la edad. La tabla realizada por expertos en el Entregable E 2.1. ha resultado ser de lo más acertada. El LED es el ejemplo más representativo de un componente fácil e intuitivo: sólo hay que arrastrar el bloque componente “encender LED” para hacer posible que eso ocurra. Lo mismo ocurre con el Zumbador o el servo de rotación continua. Con la simple definición del componente, y el bloque que se puede encontrar para programarlo a cada uno, parece ser más que suficiente para que los alumnos (sobre todo los de 10 años) se hagan una idea muy clara de cómo funciona y cómo ponerlo en marcha. En términos generales, se puede afirmar que lo intuitivo que resulte tanto el componente, los bloques que les corresponden en Bitbloq en el apartado “componentes” como la profundidad de los conocimientos que posee una persona sobre la forma en que funciona, parecen modular la dificultad y el uso que se le va a poder aplicar.

4.2.1 Pulsador, sensor IR y sensor ultrasonidos:

Se podría establecer dentro de una misma categoría de uso de componentes al pulsador, al sensor IR y al sensor de ultrasonidos. Estos tres componentes mantienen una estrecha relación por la forma en que los participantes aprenden a utilizarlos. Empezando por los más pequeños, adquirir el uso de un pulsador es muy sencillo. La única dificultad que pueden encontrar es usar un IF, además que es necesaria tener experiencia con determinados actuadores como el LED.

Al igual que el LED, el zumbador o el servo de rotación continua, el pulsador es sobradamente intuitivo para cualquier participante de cualquier edad. Todo

el mundo pulsa constantemente interruptores en cualquier lugar que encienden y apagan luces. Para programar un pulsador de forma sencilla hay que usar por ejemplo un IF, pero al tratarse de la versión más simple de un condicional en Bitbloq todavía no supone ninguna dificultad seria para programarlo. No obstante, programar un pulsador para encender un LED se trata de una tarea muy sencilla. De la misma forma, y utilizando el pulsador como ejemplo representativo, los alumnos de 10 años no tienen ningún problema en aprender a programar un sensor IR o un sensor ultrasonidos. No obstante, parece que les queda un poco de vacío sobre lo que realmente está ocurriendo pues aún no tienen los conocimientos científicos adecuados para comprender el funcionamiento de ambos sensores totalmente. Éste problema de entendimiento realmente no supuso un problema para hacer una programación sencilla pues tenían el pulsador como referencia, pero sí puede suponer limitaciones esta falta de conocimiento sobre un componente a la hora de hacerles pensar en diferentes usos. Por ejemplo, si están en una sala oscura, es mejor usar un ultrasonidos que un sensor IR, y si un usuario no es capaz de discriminar entre ambos sensores van a producirse limitaciones en su uso.

Por descontado, esta dificultad sobre el entendimiento con mayor profundidad sobre estos sensores no ocurría con los participantes mayores a 13 años pues tienen tanto mayor capacidades como conocimientos previos disponibles que faciliten tanto la comprensión como el uso de los componentes.

4.2.2 Componentes digitales vs. analógicos

Otro detalle a tener en cuenta es el tipo de componente, ya sea sensor o actuador. A la hora de programar, los participantes, en general, han encontrado más facilidades en programar componentes digitales. Según los resultados, y la teoría del Entregable E 2.1. es un resultado esperable. Un componente con dos estados, es mucho más sencillo de entender que uno analógico como el potenciómetro (posee 1023 posiciones).

Un ejemplo representativo son el sensor IR y el sensor de luz. El sensor de luz ha supuesto mayores dificultades en su programación pues posee 800 valores entre máxima oscuridad (0) y máxima luz (800). Además, medir la intensidad de la luz sin un instrumento adecuado puede resultar una tarea bastante subjetiva, es decir, dependiente del propio participante.

4.2.3 Influencia de los conocimientos previos y la edad

Los conocimientos previos son importantes para entender con mayor profundidad el componente que se está utilizando, además de que es un facilitador importante a la hora de pensar un uso para dicho componente. La edad también es influyente, pues va a determinar la cantidad y calidad de los propios conocimientos (una persona con más años escolarizado tendrá un bagaje mayor), además de tener capacidades cognitivas más desarrolladas que permitirá una mayor eficiencia (véase Entregable E 2.1.)

El potenciómetro, es un ejemplo de un componente fácil de entender (aunque sea analógico) pero difícil de dar un uso, sobre todo en personas menores de 14 años. La forma en que se mapea parece crear confusión en general por los diferentes tipos de bloque que son necesarios, y posiblemente por una falta de contextualización. Sólo los más mayores consiguen comprender cómo darle un uso y programarlo con mayor facilidad.

Volviendo al sensor de luz, aunque es un componente que todos los participantes a cualquier edad aprenden a usar, los más mayores, que poseen conocimientos de física más avanzados que permiten una asimilación más rápida. Este mismo argumento se puede usar para diferenciar la forma en que los participantes más pequeños aprenden a usar los servos con la forma en que lo usan a partir de cierta edad. Tanto servo de rotación continua como mini servo son digitales pero a la hora de programar, el primero solo posee dos sencillas posiciones (rotar sentido horario y sentido antihorario) mientras que el segundo se programa para que se posicione entre 0 y 180 grados. Los de diez años, pueden conocer el concepto de ángulo, pero no son todavía muy eficientes a la hora de programar un mini servo comparado con el de rotación continua que no hace falta tener esos conocimientos.

Para finalizar, se ha visto una diferencia importante en función de la edad, y son los usos que son capaces de dar a los componentes y la cantidad de componentes que funcionen de forma conjunta. A mayor edad, los participantes son más capaces de dar usos más variados a todos los componentes en general. Esto puede explicarse por la teoría revisada en el Entregable E 2.1., donde se explica la capacidad de las personas pertenecientes al periodo de las operaciones formales. En este periodo la realidad se concibe como un conjunto de lo posible y las personas tienen una mayor capacidad para generar múltiples abstracciones o hipótesis, y, por lo

tanto, facilitando la capacidad de realizar proyectos donde se incorporen diferentes componentes y establecer diferentes usos en base a unos objetivos propuestos.

5 Discusión

Se ha podido observar que de forma general todos los participantes de todas las edades observadas (entre 10 y 16) han sido capaces de aprender a programar y de expresar de forma adecuada las 4 habilidades descritas (elaboración de algoritmos, funciones, condicionales y bucles). Además, el proceso de aprendizaje se ha visto en otros niveles más generales cómo las estrategias con las que afrontan los problemas a los que se enfrentaban, optimizando sus conductas desde estrategias de índole aleatorio a búsquedas cada vez más ordenadas y planificadas. No obstante existen algunas dificultades que pueden ser generadas por una posible falta de contextualización a la hora de dar un uso concreto y funcional a los componentes del kit de robótica, y una merma en la programación provocada en la capacidad de lectura de los más pequeños que impedía la búsqueda o comprensión de algunos bloques. Por lo que se concluye, es posible que pese a las facilidades que ofrece Bitbloq para programar, puede tener elementos que podrían considerarse programación de bajo nivel debido a las capacidades de la población a la que va destinado.

En cuanto al lenguaje, se ha podido observar que los bloques facilitan el acceso a la programación pero hace a los alumnos dependientes de su nivel de lectura, quedando más expuestos a tener dificultades en el uso de la aplicación a los más pequeños cuyo nivel está menos desarrollado. En un lenguaje por bloques, como es Bitbloq, la similitud de la expresión de programación con las propias del lenguaje facilitan el acceso hasta el punto de realizar una comparativa léxico - programación que facilita mucho la programación, si bien ese símil requiere de cierta destreza en el uso del lenguaje hablado y escrito, razón que dificulta el acceso a los tramos de edad más bajos, encontrando problemas a la hora de leer o escribir los bloques y parámetros necesarios para elaborar un programa sencillo. Si bien todos los niños son capaces de interpretar “si aprieto el botón se enciende” no es tan sencillo estructurar un deseo como tal de una forma concreta y a la hora de redactar la condición para ser entendida por el programa podemos tener una barrera de acceso.

Lo que sí demuestra Bitbloq es que se puede aprender a programar sin aprender un lenguaje de programación de la misma forma que se puede aprender a expresar lo que sentimos sin ubicarlo en un lenguaje concreto. El ser humano es capaz de pensar y elaborar estrategias más allá del medio en el que las expresa y sabe

encontrar la forma de hacerse entender en situaciones complejas para el lenguaje que domina. En lo comprobado al programar con Bitbloq encontramos estrategias que se podrían comparar a las que puede desarrollar una persona para pedir comida en un país donde no consigue establecer comunicación verbal. Sabemos lo que queremos y buscamos la forma de expresarlo. Bitbloq permite bajar la dificultad de expresión para poder centrarnos en resolver el problema y no tanto en la estructura concreta del lenguaje que lo resuelve.

Así mismo, el código (en ocasiones tildado de la verdadera programación) no es más que otra forma de expresar el proceso que sigue la máquina para que el humano sea capaz de entenderlo, siendo más importante entender el proceso que se ejecuta que el modo en que se describe para la comunicación hombre - máquina. Es en un momento concreto del aprendizaje donde la simple interacción interesada en el proceso va a llevar al sujeto a profundizar en la forma de expresiones más concretas, pero la sencillez con que se consiguen expresiones sencillas con un efecto físico inmediato produce el efecto de inmersión con entusiasmo en el proceso por parte del sujeto activo.

Frente a la edad óptima para iniciarse en el mundo de la programación hay una respuesta clara: desde el comienzo de la etapa escolar. El problema está en el acceso a la programación y en el nivel de complejidad, pero las estrategias programadas están en todo momento presentes en el mundo del menor (por ejemplo, voy a montar estas piezas y luego juego con ellas). Al principio la programación se puede realizar de forma más sencilla, física, con juegos, con secuencias lógicas para realizar acciones, etc. Poco a poco podemos llevar ese pensamiento secuenciado hacia un entorno más ubicado en programación computacional. No es necesario trabajar con un ordenador para fomentar el pensamiento que facilita la resolución de problemas programando acciones, pero si se quiere usar un ordenador y una aplicación cuyo aprendizaje pueda transferirse a lenguajes de programación a más bajo nivel, debe adaptarse una interfaz más adaptada a las capacidades de los alumnos al principio de la etapa escolar y un hardware programable que permita crear acciones más intuitivas en la forma en que los más pequeños perciben el mundo. Los más pequeños han demostrado saber programar los componentes pero han tenido dificultades en realizar tareas que requerían establecer una abstracción o una hipótesis para programar un conjunto de componentes de diferente tipo para realizar una acción concreta.

Por lo tanto, podemos encontrar tres puntos de vital importancia que se deben tener en cuenta a la hora de querer facilitar la adquisición de la programación:

- Bloques que permitan el funcionamiento conjunto de varios componentes.
- Selección de unos bloques concretos para facilitar la exploración en la interfaz.
- Un hardware adaptado a las necesidades y capacidades de los usuarios al principio de la etapa escolar.

6 Anexo I Diario de la Experiencia en el Colegio Balder

Diario sesiones

Lunes	Martes	Miércoles	Jueves	Viernes
1º A (sesión 1)	4º B (sesión 2)	4º A (sesión 1)	4º A (sesión 2)	1º B (sesión 3)
3º A (sesión 1)	1º B (sesión 1)	3º A (sesión 3)	1º B (sesión 2)	4º A (sesión 3)
3º B (sesión 1)	3º A (sesión 2)	3º B (sesión 3)	1º A (sesión 2)	1º A (sesión 3)
4º B (sesión 1)	3º B (sesión 2)		4º B (sesión 3)	

SESIÓN	1º ESO	3º ESO	4º ESO
1 (semana 1)	<p>Usar un sensor IR para detectar blanco o negro y que se encienda un LED o suene un zumbador.</p> <p>Usar una fotoresistencia para encender un LED en ausencia de luz.</p> <p>Usar un potenciómetro para encender y apagar un LED y para variar el sonido que emite un zumbador.</p>	<p>Problemas propuestos (luces de navidad con sonido y alarma). Sin problemas para realizarlo, aunque la mayoría crean un solo bucle para el LED y el zumbador, en lugar de crear dos bucles.</p> <p>à No dividen en pequeños eventos diferentes, intentan crearlo todo de una vez.</p>	Tutorial de Khanacademy para aprender a programar por código en lenguaje Javascript.
2	Scratch: Funcionamiento básico del Scratch. Controlar un personaje haciendo que se mueva al pulsar las flechas del teclado. Al terminar, conceptos básicos de cómo cambiar el tamaño y el disfraz. Realizar un videojuego donde un personaje persiga al que manejamos y nuestro personaje tenga que recoger ítems por la pantalla. El personaje que nos persigue aumenta su velocidad según recogemos ítems.	Clase de tecnología tradicional: Energías.	Tutorial de Khanacademy para aprender a programar por código en lenguaje Javascript.
3	Siguen con el videojuego de la sesión anterior	<p>Siguen con el problema de las luces de navidad.</p> <p>Los que terminan: tutorial Khanacademy para aprender Javascript</p>	Tutorial de Khanacademy para aprender a programar por código en lenguaje Javascript.
4 (semana 2)	Realizar un juego en Scratch que consista en una nave espacial que se desplace por la pantalla y aparezcan objetos (a modo asteroides) que avancen de derecha a izquierda en diferentes alturas y a diferentes velocidades, teniendo que esquivarlos con la nave.	Terminar los problemas compartidos (el último es la alarma). Según van terminando van a empezar un tutorial con Khanacademy para programar por código en lenguaje Javascript. El objetivo es familiarizarse con el mismo de cara a usarlo en el futuro (en nuestro caso facilita el uso de openScad para diseñar e imprimir 3D).	Iniciación al dibujo 3D con código con el programa openScad. Realización de un dado (intersección de cubo y esfera y resta a las caras de la esfera de esferas para numerar las caras). Impresión 3D de uno de los dados para que visualicen el resultado.
5	Prueba de destrezas: explicar un videojuego propio comentando cómo funcionaría y que objetos tendrían que crear, explicando qué harían. Para ello se les pone una captura de pantalla del videojuego "fruit ninja" y se les dice que para esa interfaz se inventen un videojuego (que puede ser exactamente el fruit ninja) y expliquen que objetos, fondos y acciones necesitarían programar.	Siguen con la sesión anterior: terminan los problemas hasta el de la alarma. Tutorial con Khanacademy para programar por código en lenguaje Javascript.	Realizar un dado en openscad utilizando código para dibujar la intersección de una esfera y un cubo y quitándole a cada cara semiesferas para hacer los números.
6	<p>Realización de un videojuego con scratch: Fruit Ninja. Problema con el movimiento parabólico, resulta difícil explicarles cómo hacerlo y especialmente que entiendan las variables. Finalmente les queda claro uno de los múltiples</p> <p>procesos para hacer un movimiento similar al parabólico (no estrictamente parabólico pues es una ecuación de segundo grado y son muy pequeños para ello).</p>	<p>Seguir el tutorial de Khanacademy para programar por código. Algunos ya han llegado a variables y objetos móviles. El objetivo es reforzar el fundamento de escribir ideas secuenciadas con el código para estructurar un proceso de cara a</p> <p>trasladarlo al trabajo por bloques, siendo capaces de descomponer el problema propuesto en una sucesión de hechos sencillos.</p>	Dibujar un pinocho en openscad. Introduciendo los giros con un comando como traslación espacial respecto de los ejes (les causa muchos problemas entender si el giro tiene que ser sobre un eje u otro). Creación de variables para que se desplacen objetos de forma continua.
7 (semana 3)	Continuamos con el juego en Scratch simulando al Fruit Ninja. Muchos problemas a la hora de crear variables, no escuchan y siguen creando variables globales cuando programan un objeto por lo que luego todos los objetos se mueven de la misma forma.	Continúan con el tutorial de Khanacademy para aprender Javascript.	Examen consistente en dibujar, en dos sesiones, un muñeco completo en OpenSCAD (similar a un pinocho, muñeco de nieve, etc). Primera sesión de las dos dedicadas a tal fin.

8	Realización de videojuego Fruit Ninja. Algunos consiguen terminarlo a falta de conseguir que al no cortar una fruta te quite una vida al tocar el borde inferior.	Control de destrezas con dos situaciones de bitbloq y explicación de las mismas argumentando si cumplen su propósito así como explicación de “pin analógico” y “pin digital”.	Prueba de diseño en OpenSCAD (realizar un pinocho, muñeco de nieve o similar).
9	Realizar un dado en openSCAD, primera sesión. Siguiendo un videotutorial que he realizado tienen que diseñar un dado en el programa de dibujo por código.	Tarea (ausencia profesor) que sigan con el tutorial de Khanacademy.	Último día para terminar el dibujo en openSCAD (pinocho, muñeco de nieve o algo así). Me envían todos los diseños que han realizado.
10 (sem ana 4)	Segunda sesión siguiendo el videotutorial para realizar el diseño de un dado de 6 caras en OpenSCAD. Algunos lo terminan y empiezan a hacer un muñeco de nieve o similar.	Se comparte con los alumnos el siguiente bloque de problemas de bitbloq y arduino, que empieza con la programación del zumbador con un potenciómetro. Las actividades compartidas van de la 7 a la 12.	Se inicia bitbloq y arduino, explicando cómo funciona bitbloq, cómo se conectan cosas a la placa arduino y pidiéndoles que sigan los tutoriales de LED (I y II). Se hará una pequeña puesta en escena del kit de robótica y bitbloq de cara a que cuando se les proponga el proyecto global tengan unas nociones muy básicas al respecto.
11	Tercera sesión siguiendo el videotutorial para realizar el diseño de un dado de 6 caras en OpenSCAD. Al terminar tienen que realizar algún muñeco (quieren hacer muñecos de mindcraft, muñeco de nieve, pollitos...).	Clase tradicional: energías, central térmica nuclear y central hidroeléctrica	Segunda sesión de introducción breve a bitbloq y arduino. Tras terminar los tutoriales de LED I y LED II realizan el del zumbador y el sensor de luz.
12	Última sesión de dibujo con OpenSCAD, tema libre. Hacen un poco de todo, alguna maza con pinchos, muñecos de mindcraft, muñecos de nieve, etc.	Siguen trabajando los problemas propuestos, están terminando el proyecto del potenciómetro para hacer sonar un zumbador, algunos ya inician el siguelíneas. La motivación es muy grande, lo que más les cuesta es ponerlo en el blog, pero es verdad que el resultado es muy bueno.	Siguen con los tutoriales de diwo para bitbloq (infrarrojos, ultrasonidos, pulsador...). Llegan al nivel adecuado para empezar a plantearse iniciar el proyecto.
13 (sem ana 5)	Clase de tecnología clásica. Trabajamos sobre la electricidad, simbología eléctrica y corrientes continuas y alternas.	Trabajan sobre los problemas. Tenían que traer unas pilas por equipos para ir haciendo el siguelíneas pero sólo un alumno las trae y no las que eran necesarias. Bronca al canto. Están advertidos de que si el lunes no traen el material que les he pedido no trabajamos problemas y damos clase.	Trabajan con algunos elementos de bitbloq. Surgen problemas a la hora de configurar la pantalla LCD, ya he comentado con Alberto el asunto, es un error de programación de bitbloq.
14	Están diseñando en OpenSCAD un poco libremente, usando algunas variables y creando objetos y duplicándolos. Les presento también los tutoriales de vuestro compañero Juan (Obijuan) para que intenten cosas un poco más difíciles.	Clase de tecnología clásica, energías: energía solar y energía eólica, explicación de sistemas de aprovechamiento, pros y contras, etc...	Trabajan un poco en su blog personal explicando en unas entradas el funcionamiento de OpenSCAD, bitbloq y Scratch.
15	NO HAY CLASE: CARNAVAL	Siguen con los problemas que les hemos compartido. Casi todos están haciendo un siguelíneas (con el renacuajo). Problema: las actividades que implican varias sesiones suponen un problema si hay varias líneas por curso, ya que si lo dejan a medias el otro curso no puede usarlo y tienen que desmontarlo... o tener el material por duplicado.	NO HAY CLASE: CARNAVAL
16 (sem ana 6)	Les dejo que sigan diseñando en OpenSCAD, enseñándoles a hacer figuras un poco más complejas como anillas a partir de la rotación de un círculo o columnas de diferentes secciones y desarrollos (por ejemplo de base pentagonal y que al ir subiendo el pentágono vaya girando, creando una especie de espiral).	Siguen con los problemas que les he compartido, la mayoría está haciendo el siguelíneas aún, pero el grupo que más fluido va llega a una actividad que usa la LCD, les digo que sigan con otras actividades ya que esa aún no es posible hacerla (no funcionan los LCD). El siguelíneas es un proyecto que no se puede hacer en una sola sesión.	Trabajan en su blog personal describiendo bitbloq y Arduino, OpenSCAD y SCRATCH.
17	Clase tradicional de electricidad. Ejemplo de un circuito eléctrico y análisis de cómo va variando el voltaje y la intensidad a lo largo del mismo, así como explicación de todos los elementos del mismo.	Clase sobre energías. Energía mareomotriz y debate sobre el uso y gestión de la energía en España (consumos, autosuficiencia energética, compra a otros países de la misma, gestión del sector privado, etc). Les gusta MUCHO este tema.	Se les inicia en el trabajo con FreeCAD, explicando muy brevemente cómo funciona y dejándoles libertad para investigar un poco. Será una herramienta que usen en el proyecto (combinada con OpenSCAD o de manera única).

18	Control de destrezas durante cinco minutos en el que tienen que responder a unas cuestiones básicas de openscad. Al terminar les propongo un juego de Scratch en el que tienen que comerse frutas que caen y evitar comerse otros objetos, al comerse frutas se va mejorando salud y van cambiando los fondos. También cuenta como control.	Continúan con sus problemas, aunque antes hacen un control de destrezas sobre bitbloq en el que tienen que detectar un pequeño fallo de programación y tras corregirlo explicar qué función realiza la programación. Para el ejemplo se usa la programación del siguelíneas.	Continúan aprendiendo el uso de FreeCAD para diseño 3D, les cuesta mucho más que el OpenSCAD.
19 (sem ana 7)	Siguen trabajando en el videojuego de Scratch que les propuse la semana anterior.	Trabajan en los problemas. El ritmo de trabajo es regular pero algunos van un par de problemas por delante de otros.	Les dejo tiempo para trabajar en su blog personal y les propongo una prueba de destrezas: tienen que crear un "control" (y el solucionario) sobre openscad con preguntas que a su gusto demuestren que, en caso de contestar correctamente, el examinado sabe y domina dicho software.
20	Tienen que terminar el videojuego de frutas cayendo. Se propone para los que terminan que realicen una programación que simule el viento en Scratch.	Trabajan sobre un proyecto en twitter que hemos empezado compitiendo con otro colegio en retos creativos, durará un mes y a partir de ahora lo harán en casa, no quitará tiempo de clase, pero como es un punto del temario (redes sociales) lo he trabajado de esta forma y la primera sesión se la hemos dedicado en clase.	Siguen con el trabajo para crear un control sobre openscad (domina OpenSCAD al nivel que lo dominan ellos) y terminar su blog personal.
21	Control sobre electricidad.	Siguen con los problemas, han ido empezando diversos problemas y, al no ser ya todos una sucesión de problemas resueltos en diwo, tienen que pensarlos un poco y les supone un reto mayor. Estoy contento con el resultado, ya veréis algún vídeo.	Trabajan sobre un proyecto en twitter que hemos empezado compitiendo con otro colegio en retos creativos, durará un mes y a partir de ahora lo harán en casa, no quitará tiempo de clase, pero como es un punto del temario (redes sociales) lo he trabajado de esta forma y la primera sesión se la hemos dedicado en clase.
22 (sem ana 8)	Trabajan con el Scratch terminando los proyectos que tenían iniciados. Última sesión para ello.	Trabajan en los problemas generados para ser trabajados con bitbloq. Los resultados son buenos, se pueden ver en los blogs que están trabajando.	Se les entrega el proyecto para realizar un edificio con impresión 3D, robótica, programación, etc. Me resulta imposible que se lean el dossier con calma todos y tengo que parar la clase y explicárselo paso a paso. Son incapaces de prestar atención a algo denso y largo con calma y trabajando en grupo, veremos cómo les ayuda este trabajo.
23	Corrección en la pizarra del control que hicieron sobre electricidad y después les dejo trastear el scratch un rato. Última sesión antes de volver a bitbloq.	Siguen con los problemas, hay un grupo que está a punto de terminar (le quedan la pantalla LCD cuando se pueda usar, el joystick y la botonera).	Segunda sesión del proyecto para decidir opciones y elaborar la tabla de votaciones para decantarse por la idónea. Les explico un poco a usar google drive. Se lo toman bastante en serio aunque como aprendizaje personal creo que hay que hacer una hoja de instrucciones muy concretas y cortas y que la tengan a mano, en el sentido de: "todo se trabaja en google drive", "no se puede acometer directamente un trabajo a lo loco sin realizar una estrategia de trabajo", etc.

24	<p>Pequeño control de destrezas de bitbloq. Detectar un error en una programación y decir hacia dónde se traslada una esfera al poner unas cantidades numéricas en los ejes "x", "y" y "z". Después empezamos a entender cómo funciona un siguelíneas explicando sensor infrarrojo y sensor de luz, diferencias (analógico y digital, uno reconoce dos estados y otro más de dos). Después les explico las dos formas de hacer un siguelíneas (sensores que detectan la línea negra y si se salen de la misma giran y sensores que detectan el blanco que colinda con la línea y si detectan la línea giran, la diferencia es que en el primero paran la rueda contraria al sensor que detecta el cambio y en el segundo la rueda del mismo lado).</p> <p>Programan un sensor infrarrojos que mueve una rueda o la para en función de recibir blanco o negro.</p>	<p>Siguen con los problemas de bitbloq. Tengo que crear actividades nuevas ya que a alguno le queda poco para terminar las 13 que les he compartido. Espero tener un hueco para pasaros las fichas corregidas próximamente, y voy a intentar hacer un video de la mayoría de ellos, puede ser algo que podéis poner en diwo como ejemplo de cómo acometer la robótica con bitbloq en aula por problemas.</p> <p><u>Noto que las guías en diwo y en general están muy enfocadas a poder conseguir algo sin entenderlo y muchos grupos están programando sin aprender la mentalidad que tiene que tener alguien al programar. Iré creando actividades que no tengan tutoriales y tengan que resolver desde el pensamiento crítico y analizando cómo conseguirlo.</u></p>	<p>Tercera sesión para el proyecto por equipos. Este grupo me va a dar problemas, son muy apáticos, pero intentaré que salga con éxito una propuesta y se involucren en hacer algo muy bueno. (4ºB)</p> <p>Trabajo en el proyecto, vamos afinando un poco las propuestas, les resulta complicado abordar algo tan grande y poco secuenciado pero ahí es donde les tengo que enseñar a trabajar en equipo e ir mejorando su posición frente a los retos o problemas. (4ºA)</p>
25 (semana 9)	<p>Programando el siguelíneas. Casi nadie trae las pilas que les he pedido, éste punto podría haberlo solucionado de otra forma pero una de las cosas que más observo en mi alumnado es que no es responsable y me gusta que vean el aspecto negativo de no hacer lo que se les pide, por ello sin pilas no pueden montar el siguelíneas, aunque si lo programan y siguen programando un servo para que gire una cantidad de grados en función de la luz.</p> <p>--> realmente tienen más facilidad que 3ºESO para abordar problemas con bitbloq aunque fallan cosas concretas, por ejemplo llaman a cualquier variable con el mismo nombre aunque su contenido no sea igual, no llegan a entender que el ordenador nunca tiene que encontrarse en la tesitura de decidir, pues no sabe hacerlo, es algo humano. Si dos cosas entran en conflicto el programa resultará en fracaso (por ejemplo que una variable sea el servo en el pin 8 y otra el servo en el pin 9 y ambas se llamen "servo", ¿cual ejecuto?).</p>	<p>Siguen con los problemas, estoy corrigiendo los blogs y hay algunos buenos y otros regulares, pero en general el resultado es muy bueno. Hay un grupo que está con el joystick intentando que haga funcionar un coche para dirigirlo.</p>	<p>Siguen con el proyecto. Empiezan a afinar, les cuesta entrar en materia y serconscientes de que las cosas hay que hacerlas de un modo concreto, por ejemplo no entienden que si les pido que hagan para cada "reto" que vayan a cometer una estrategia para llevarlo a cabo la única forma que tiene sentido es hacerlo antes y aplicarlo y no ir a lo loco y luego intentar rellenar una memoria. Necesitan más entrenamiento para la autonomía y abordar retos que tienen que ser desgajados y secuenciados. También es cierto que es un grupo especial que arrastra muchos problemas a lo largo de la ESO y siempre han sido muy problemáticos.</p>
26	<p>Clase de tecnología, comenzamos con el tema de estructuras.</p>	<p>Tienen control sobre energías.</p>	<p>Trabajo en el proyecto, lo han cogido con ganas y van avanzando, aunque es cierto que es complicado que todos los miembros del grupo se impliquen al mismo nivel.</p>
27	<p>Empiezan a montar el siguelíneas. Les dejo que sigan las instrucciones al respecto de Diwo. Sólo un grupo lo termina, prefiero que les cueste y lo tengan que desmontar, tienen que coger soltura en seguir instrucciones y no tener todo tan claro siempre desde el profesor. (1º B)</p> <p>Les dejo un guión para que hagan un logo en 3D a partir de un 2D. Les cuesta muchísimo seguir el proceso explicado, no lo leen y preguntan todo antes de llegar a ello. Los niños de hoy en día no son capaces de leer textos extensos jejeje. (1º A)</p>	<p>Problemas: están terminando los problemas que les compartí. No van mal, siguen siendo todos de bitbloq y el kit de robótica, pero ya la semana que viene les compartiré cosas de diseño y programación...</p>	<p>Trabajan en el proyecto, no están trabajando mal. Les estoy enseñando a diseñar figuras imprimibles a un miembro de cada grupo para que lo haga y les enseñe al resto. Hay un encargado de cada grupo haciendo un logo imprimible y el resto están trabajando en la forma de su proyecto y empezando los diseños de las partes que tienen que construir.</p>
28 (semana 10)	<p>Por problemas con bitbloq les pongo a diseñar con el openscad figuras en 3D a partir de imágenes o figuras 2D (usando el inkscape y un tutorial de mi web).</p>	<p>Usan el openscad por primera vez. Les dejo que diseñen un dado siguiendo un tutorial de mi web (así el lunes próximo ya les puedo compartir nuevos problemas ya de diseño 3D).</p>	<p>Siguen con el proyecto, ya tienen algunas ideas clave y espero que la semana que viene empiece a consolidarse para poder analizar en semana santa resultados hasta el momento de un proyecto tan libre y con grupos tan amplios.</p> <p>--> se nota ya mucho que algunos se lo toman en serio y otros pasan olímpicamente. Voy a tomar notas puntuales consistentes en coger por banda a alguno del grupo y que me cuente qué está haciendo y qué está haciendo el resto del grupo.</p>

29	--- Excursión ---	Siguen con OpenSCAD. Es un proceso un poco más guiado porque no teníamos otra opción dado que no funcionaba correctamente bitbloq. Algunos detectan que ya funciona bien y prefieren seguir terminando las actividades de robótica.	--- Excursión ---
30	--- Puente día del padre ---	--- Puente día del padre ---	--- Puente día del padre ---
31 (sem ana 11)	Programando y montando un siguelíneas, los que acaban se ponen con el sigueluces.	Siguen con sus problemas, algunos ya han hecho un renacuajo que es dirigido por un joystick, que es el último que les compartí, y ya empiezan con diseño 3D.	Siguen con el proyecto, estoy muy contento con cómo han comenzado a trabajar con ganas, están sacando resultados. Decir que les emocionó el tema de diseñar un logo y verlo impreso, porque han empezado a ser conscientes de que es algo que puede quedar muy chulo. (4ºB) Trabajamos con el proyecto, aunque empezaron un poco peor en términos de autonomía que el otro grupo ya están empezando a coger ritmo. Ya tenemos los logos y están diseñando las partes de la estructura que quieren imprimir. (4ºA)
32	Seguimos montando el siguelíneas y sigueluces. Tienen bien adquirida la dinámica y noto que enfocan mejor los problemas tecnológicos que los de 3º y 4º de la ESO, quizá por haber empezado por Scratch e impresión 3D con openscad.	Trabajan en los problemas.	Siguen trabajando con el proyecto. Recuerdo que un grupo hará un polideportivo y el otro hará un restaurante estilo americano (4ºB) Siguen con el proyecto. Está bastante afinada la idea, el grupo que va a hacer un tren va a hacerlo colgado, no por vías, sino suspendido de un rail. El que va a hacer una pasarela de moda va a montar una estructura impresa en 3D para poner luces imitando focos. (4ºA)
33	--- Vacaciones semana santa ---	Siguen con los problemas.	Seguimos con el proyecto, aunque como entregamos las notas la clase dura poquito.
34,35,36	NO INFO	NO INFO	NO INFO
37	Programan una botonera y un joystick con bitbloq usando los tutoriales de DIWO. Todo bien, tras explicarles un poco el funcionamiento de cada uno se ponen a trabajar y la mayoría termina la botonera con un zumbador y haciendo parpadear un LED.	Siguen con los problemas. Todo correcto.	Hacen un control de destrezas de manera individual consistente en explicar el avance por el proyecto individual y como grupo.
38	Damos clase de estructuras, explicación de cimentaciones de edificios.	Les enseño a utilizar Tinkercad, siguen un tutorial de procesos básicos y luego se ponen a enredar con el programa. Lo usarán seguramente para hacer algunos de sus problemas.	Siguen con el proyecto, estamos imprimiendo ya piezas de la estructura (TRABAJAN EN LA MAQUETA) La realidad es que un proyecto de este estilo tiene que estar muy bien estructurado y yo no he tenido el tiempo de secuenciarlo correctamente con tanta novedad, al final necesitan fechas precisas para llegar a objetivos porque si no no hacen nada. Lo voy solucionando poniéndoles límites temporales a objetivos futuros.

39	<p>Conociendo Tinkercad. Día para trastear con el programa y seguir el tutorial que he hecho de movimientos básicos para conocer como diseñar algo.</p>	<p>Siguen con los problemas de diseño.</p>	<p>Continuamos con el proyecto.</p> <p>Trabajan bien pero no son conscientes del poco tiempo que les queda por mucho que lo repito. Es importante tener en cuenta que proyectos muy largos les desmotivan por la lentitud en obtener cosas tangibles y tienen que estar muy bien diseñados en fechas, tiempos, objetivos a corto plazo, motivación frente al resultado aunque no sea muy concreto inicialmente...</p> <p>Uno de los grupos diseñó una plataforma para imprimirla y al imprimirla han visto que hay varias cosas que no habían pensado y que además no es simétrica como querían, fruto de diseñar a ojo y no con medidas. Un ejemplo claro de las ventajas de la impresión 3D, conectan el diseño virtual con el real y se dan cuenta que hay que ser muy precisos...</p>
----	---	--	---

7 Anexo II Imágenes de las formaciones a alumnos con el Kit de Robótica



