

Trabajo Fin de Grado

Estudio, diseño y evaluación de actividades para introducir conceptos básicos de informática a niños de primaria.

Autor

Jorge Berdún Udina

Directores

Pedro Javier Álvarez Pérez-Aradrós
Sandra Baldassarri

ESCUELA DE INGENIERÍA Y ARQUITECTURA
Zaragoza, Junio de 2017



DECLARACIÓN DE
AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./D^a. Jorge Berdún Udina

con nº de DNI 72992164 - P en aplicación de lo dispuesto en el art.

14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo

de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la

Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster)

Grado _____, (Título del Trabajo)

"Estudio, diseño y evaluación de actividades para introducir conceptos básicos de informática a niños de primaria"

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 23 de Junio del 2017

Fdo: _____

Estudio, diseño y evaluación de actividades para introducir conceptos básicos de informática a niños de primaria.

Resumen

Actualmente resulta necesario que los niños se relacionen de una forma responsable con la tecnología que les rodea. Además, resulta infrecuente desarrollar una actividad profesional, o hallar una situación de la vida cotidiana, que no necesite de una interacción con elementos tecnológicos.

Cierto grado de conocimientos informáticos resultan por tanto útiles, y es por esto que las instituciones educativas comienzan a incluir en sus programas educativos, actividades orientadas a enseñar conceptos básicos de informática niños. Sin embargo, existen multitud de herramientas y actividades disponibles que están diseñadas específicamente con este propósito.

El objetivo principal de este trabajo consiste en diseñar, realizar, y evaluar, una serie de actividades destinadas a enseñar conceptos de programación e informática a niños de entre 6 y 12 años. Para ello, primero se recopilará información acerca de conceptos pedagógicos que resultan útiles a la hora de enseñar informática a niños (educación STEM y pensamiento computacional), y se hará un estudio de las herramientas existentes.

Con el fin de obtener más información práctica acerca de las herramientas y el tipo de actividades educativas que se realizan actualmente, se entrevistarán a varios docentes para conocer todos los aspectos de una actividad desde un punto de vista práctico, es decir, identificar el perfil del docente que imparte la clase, el del alumnado asistente, el contexto de la clase, y las herramientas empleadas para el desarrollo de la misma.

Combinando esta información práctica y el estudio de mercado inicial de las herramientas, se realiza una selección final de aquellas herramientas que se utilizan en las actividades, en nuestro caso son Code, LearnToMod y Bitbloq.

Partiendo de esta lista de herramientas, e incluyendo los conceptos pedagógicos que se han considerado útiles para enseñar informática a niños, se procede al diseño de 4 actividades que posteriormente se ejecutan y evalúan.

Durante la ejecución de cada una de estas 4 actividades, el profesor toma notas acerca de su desarrollo, incidencias, completitud consecución de objetivos, etc. Al final de la sesión se realiza una entrevista individual a cada uno de los alumnos asistentes.

De estas dos fuentes de información se extraen una serie de conclusiones que nos acaban aportando dos listas con las características que se consideran ideales de, por un lado, las herramientas dedicadas a enseñar conceptos de informática a niños, y por otro, de las actividades que hacen uso de estas herramientas con el mismo objetivo.

Agradecimientos

Me gustaría comenzar agradeciendo a mis directores de proyecto, Pedro y Sandra, su ayuda e implicación en el desarrollo del trabajo, y el haberse atrevido con un TFG algo más atípico que la norma.

También agradecer a mi familia y amigos por su apoyo y preocupación constantes.

Además, agradecer enormemente a todos los profesionales y docentes entrevistados: Jesús Mínguez, David Bordonada, Daniel Espinosa y Raquel García, Pablo Aliaga, Beatriz García, Andreu Camps, Ana Quintana y María Tosat y a los trabajadores de Ibercaja Clip; por dedicarme parte de su tiempo, y en muchas ocasiones recursos de distinto tipo, con los que trabajar.

Y finalmente, aunque no por ello menos importante, me siento especialmente agradecido con Adrián Blasco y con la academia Enseñalia San Francisco. Al primero por introducirme y ser mi mentor en lo que a educar a niños en cursos relacionados con la informática y la tecnología se refiere. Y a Enseñalia, por darme la oportunidad de trabajar con ellos, y prestarme sus materiales e instalaciones para realizar y evaluar las actividades necesarias para este TFG.

Índice

| | |
|---|-----------|
| 1. Introducción | 9 |
| 1.1 Contexto y Motivación..... | 9 |
| 1.2 Objetivos | 9 |
| 1.3 Estructura de la memoria..... | 10 |
| 2. Conceptos educativos..... | 12 |
| 2.1 STEM y Educación STEM..... | 12 |
| 2.2 Pensamiento computacional..... | 12 |
| 3. Estudio de herramientas para la enseñanza de informática..... | 15 |
| 3.1 Herramientas de programación visual..... | 16 |
| 3.2 Herramientas relacionadas con la robótica..... | 21 |
| 3.3 Herramientas sin ordenador | 23 |
| 3.4 Conclusiones | 25 |
| 4. Metodología | 27 |
| 5. Información práctica, Entrevistas..... | 29 |
| 5.1 Diseño de entrevistas..... | 29 |
| 5.2 Resultados de las entrevistas | 30 |
| 5.2.1 Perfil del docente..... | 30 |
| 5.2.2 Perfil del alumnado | 32 |
| 5.2.3 Contexto de la clase..... | 33 |
| 5.2.4 Opinión herramientas | 35 |
| 6. Diseño de actividades, Clases Tipo..... | 38 |
| 6.1 Selección de herramientas | 38 |
| 6.2 Requisitos comunes de las Clases Tipo..... | 39 |
| 6.3 Aspectos de diseño comunes de las Clases Tipo | 39 |
| 6.4 Programas educativos..... | 41 |

| | |
|---|-----------|
| 6.4.1 Clases Tipo – Code – Programas educativos | 41 |
| 6.4.2 Clase Tipo – LearnToMod – Programa educativo | 44 |
| 6.4.3 Clase Tipo – Bitbloq – Programa educativo..... | 47 |
| 7. Evaluación Clases Tipo..... | 50 |
| 7.1 Consideraciones sobre el sistema de evaluación..... | 50 |
| 7.2 Resultados evaluaciones Clases Tipo..... | 51 |
| 7.2.1 Evaluación Code, Hora del Código – Programa con Ana y Elsa | 51 |
| 7.2.2 Evaluación Code, Hora del Código – Escribe tu primer programa | 53 |
| 7.2.3 Evaluación LearnToMod, Introduction To Loops | 55 |
| 7.2.4 Evaluación Bitbloq, Difunto Robonauta..... | 56 |
| 7.3 Conclusiones | 58 |
| 8. Conclusiones | 60 |
| 8.1 Conclusiones | 60 |
| 8.2 Valoración personal..... | 62 |
| 8.3 Trabajo futuro..... | 62 |

Bibliografía:

Anexo A – Transcripciones de las entrevistas

Índice de Figuras

| | |
|--|----|
| Ilustración 1 - Code - Flappy lvl 9..... | 17 |
| Ilustración 2 - LearnToMod - Skills and Drills (Using the Old Interface), Parameters: Monster House (Scramble)..... | 18 |
| Ilustración 3 – Scratch – Programación libre | 19 |
| Ilustración 4 - Minecraft Education Edition - Inicio Tutorial | 19 |
| Ilustración 5 - Code Builder for Minecraft: Education Edition..... | 20 |
| Ilustración 6 - Bitbloq - Programación libre..... | 21 |
| Ilustración 7 - Piezas y componentes del robot Lego MindStorms EV3..... | 22 |
| Ilustración 8 - LEGO MindStorms - Programación robot EV03 Storm, paso 6 | 22 |
| Ilustración 9 - Robot Turtles - Instrucciones | 24 |
| Ilustración 10 - Cartas guía Code Monkey Island | 24 |
| Ilustración 11 - Tablero Code Monkey Island..... | 24 |
| Ilustración 12 - Diagrama metodología | 27 |
| Ilustración 13 Hora del Código - Programa con Ana y Elsa, nivel 19 | 43 |
| Ilustración 14 Hora del Código – Escribe tu primer programa | 43 |
| Ilustración 15 Actividad Introduction To Loops | 44 |
| Ilustración 16 - LearnToMod - Solución a los problemas 1, 2 y 3..... | 45 |
| Ilustración 17 - LearnToMod - Resultado al ejecutar la solución del problema 4 | 46 |
| Ilustración 18 - LearnToMod – Solución problema 4 | 46 |
| Ilustración 19 - Bitbloq - Difunto Robonauta..... | 48 |
| Ilustración 20 - Bitbloq - Montaje HW | 48 |
| Ilustración 21 - Bitbloq - Código problema final Difunto Robonauta | 49 |

Índice de Tablas

| | |
|--|----|
| Tabla 1 Comparativa Herramientas de programación visual | 20 |
| Tabla 2 Comparativa Herramientas relacionadas con la robótica | 23 |
| Tabla 3 Comparativa Herramientas sin ordenador | 25 |
| Tabla 4 - Entrevistas a Profesionales – Perfil del Docente 1de2..... | 30 |
| Tabla 5 - Entrevistas a Profesionales - Perfil del Docente 2de2 | 31 |
| Tabla 6 - Entrevistas a Profesionales - Perfil del Alumado 1de2..... | 32 |
| Tabla 7 - Entrevistas a Profesionales - Perfil del Alumnado 2de2..... | 32 |
| Tabla 8 - Entrevistas a Profesionales - Contexto de la clase 1de2 | 33 |
| Tabla 9 - Entrevistas a Profesionales - Contexto de la clase 2de2 | 34 |
| Tabla 10 - Entrevistas a Profesionales - Opinión herramientas 1de2..... | 36 |
| Tabla 11 - Entrevistas a Profesionales - Opinión herramientas 2de2..... | 37 |
| Tabla 12 - Entrevistas Alumnos - Clase Code, Hora del Código – Programa con Ana y Elsa | 52 |
| Tabla 13 - Entrevistas Alumnos - Clase Code, Hora del Código – Escribe tu primer programa..... | 54 |
| Tabla 14 - Entrevistas Alumnos - Clase LearnToMod -Introduccion To Loops | 55 |
| Tabla 15 - Entrevistas a los Alumnos – Clase Bitbloq Difunto Robonauta | 57 |

1. Introducción

En este primer capítulo, se desarrollan el contexto y los motivos que han llevado a realizar este trabajo. A continuación, se indican los objetivos que se persiguen y cómo se pretende alcanzarlos, y finalmente, se describe la estructura general del trabajo y se resume brevemente el contenido de cada uno de sus capítulos.

1.1 Contexto y Motivación

Desde el auge de las nuevas tecnologías e Internet, cada vez es más común, y a su vez necesario, que los niños se relacionen de una forma responsable y con conocimiento con la tecnología que les rodea. Esta necesidad se acentúa conforme se empieza a entrar en la vida adulta ya que, en la actualidad, es poco frecuente desarrollar una actividad profesional o, simplemente, hallar una situación de la vida cotidiana, que no requiera de una pequeña interacción con elementos tecnológicos, para lo que cierto grado de conocimientos informáticos resultan útiles o incluso pueden llegar a ser necesarios.

Por estos motivos, la gran mayoría de instituciones educativas, academias y centros de ocio, están comenzando a incluir en sus programas educativos actividades orientadas a enseñar conceptos básicos de programación, algoritmia, aplicaciones ofimáticas, manejo básico de un ordenador e internet, etc. En este contexto, y cuando se trata de transmitir esos conocimientos a niños, existen multitud de herramientas y actividades distintas diseñadas específicamente con este propósito. Sin embargo, no hay una línea clara a seguir, un programa educativo o herramientas estandarizadas a emplear, ya que se trata de actividades que se han comenzado a implantar, o incluso desarrollar, recientemente.

Por esta razón, existe una creciente necesidad de hallar los métodos más adecuados para transmitir estos conocimientos sobre Informática y nuevas tecnologías. Y es por esto que se desea analizar el estado actual de la cuestión realizando un conjunto de actividades, o clases, diseñadas para trabajar con niños y haciendo uso de un determinado subconjunto de las herramientas disponibles.

1.2 Objetivos

El objetivo principal de este trabajo consiste en realizar y evaluar una serie de actividades destinadas a enseñar conceptos de programación e informática a niños de entre 6 y 12 años.

Para alcanzar este objetivo principal, se abordarán a lo largo del trabajo varios objetivos secundarios:

- Recopilar aquellas ideas y estrategias pedagógicas, útiles o necesarias, para enseñar conceptos relacionados con la informática a niños.
- Realizar un estudio para descubrir qué herramientas, y de qué tipo, se utilizan actualmente para enseñar Informática a niños.

- Diseñar una metodología a seguir que permita, en primer lugar, seleccionar qué herramientas emplear en las actividades, y en segundo, diseñar, implementar y evaluar dichas actividades.
- Obtener más información relevante a partir de la práctica y no solo de la teoría sobre: qué tipo de actividades, y en qué contexto, se están realizando actualmente, por quién, a qué tipo de alumnado están orientadas, y qué herramientas se utilizan en las mismas.
- Diseñar una serie de actividades destinadas a enseñar conceptos relacionados con la informática y la programación, haciendo uso de un subconjunto final herramientas escogidas.
- Impartir dichas clases y realizar una evaluación posterior del desarrollo y resultados de cada una de ellas.

1.3 Estructura de la memoria

La estructura de esta memoria consta de ocho capítulos y un único anexo:

- **Capítulo 1 – *Introducción***: breve introducción sobre el trabajo, motivaciones para su realización, y el objetivo del mismo.
- **Capítulo 2 – *Conceptos educativos***: breve descripción de los conceptos de “Educación STEM” (*Science, Technology, Engineering and Mathematics*) y “Pensamiento Computacional”.
- **Capítulo 3 – *Estudio de herramientas para la enseñanza de informática***: estudio inicial sobre las herramientas orientadas a la formación de niños en conceptos relacionados con la informática y la programación que existen y se utilizan actualmente.
- **Capítulo 4 – *Metodología***: descripción de la metodología de trabajo que se sigue para la selección de las herramientas que emplear en las actividades y el diseño, implementación y evaluación final de estas.
- **Capítulo 5 – *Información práctica, Entrevistas***: diseño, y resultados, de las entrevistas realizadas a profesionales relacionados con la enseñanza a niños de conceptos informáticos, para la obtención de información práctica sobre las actividades que imparten.
- **Capítulo 6 – *Diseño de actividades, Clases Tipo***: selección de las herramientas que se emplean en las actividades, requisitos y aspectos de diseño comunes a todas las clases y programa educativo de cada una de las actividades que se realizan.
- **Capítulo 7 – *Resultados evaluaciones Clases Tipo***: recopilación de los datos obtenidos de las clases tipo, notas del profesor y entrevistas a los alumnos, y conclusiones.

- **Capítulo 8 – Conclusiones:** Conclusiones finales del trabajo, valoración personal y trabajo futuro.
- **Anexo A - Transcripciones de las entrevistas:** recopilación de las transcripciones de las entrevistas realizadas a profesionales del sector y cuyo resumen y análisis se encuentra en el capítulo 5 de este TFG.

2. Conceptos educativos

A la hora de diseñar e implementar actividades docentes relacionadas con la enseñanza de conceptos informáticos a niños, es necesario apoyarse en métodos educativos que estén orientados a transmitir, de una forma eficiente y adecuada, dichos conocimientos.

En este capítulo, se describen los conceptos de “STEM”, “Educación STEM” y “Pensamiento Computacional” desde un punto de vista pedagógico e informático. Estos conceptos están específicamente diseñados para integrar la informática dentro del currículo educativo estándar, y enseñar un método de resolución de problemas relacionado con el razonamiento informático, respectivamente.

2.1 STEM y Educación STEM

El acrónimo STEM se acuñó originalmente en los años 90 por la *National Science Foundation* (NSF) [1], aunque no fue hasta el 2001 que, la ex directora de la división de educación y recursos humanos de la NSF, Judith A. Ramaley, inició el programa de educación STEM [2].

La educación STEM es multidisciplinar, abarcando todos los conceptos de las matemáticas, las ciencias y la ingeniería, y aprovechando la relación natural que tienen entre sí. Esta filosofía permite que la enseñanza de estas disciplinas se complete con el componente de la tecnología, que permite la resolución de problemas y la aplicación de los aprendizajes de una forma práctica, creativa e innovadora.

Se trata de transformar la clásica clase magistral que gira en torno al profesor, en una clase basada en un plan de estudios orientado a la resolución de problemas y el descubrimiento y aprendizaje exploratorio, por parte del alumno. La participación de los estudiantes a lo largo de la clase resulta vital, dado que será esta participación activa la que les permitirá hallar las soluciones a los problemas planteados, aunque siempre contando con la guía y el apoyo del profesor [3].

Con este método, se pretende provocar en los alumnos la generación de procesos de investigación científica que permitan el aprendizaje, incluyendo nuevos conceptos dentro de un proceso práctico de diseño y resolución de problemas, tal y como se hace en la ingeniería en el mundo real.

La educación STEM está permitiendo la incorporación de la ingeniería y de sus métodos en la educación primaria y secundaria del siglo XXI, como ya ocurriera, a lo largo del siglo XX con la ciencia y su método.

2.2 Pensamiento computacional

El término “pensamiento computacional” tiene sus orígenes en las ideas de Seymour Papert (matemático, informático y profesor del *Massachusetts Institute of Technology*, MIT), sobre educación y la relación de los niños con los ordenadores [4]. Sin embargo, fue Jeannette Wing (presidenta del departamento de ciencias de la computación en la *Carnegie Mellon University*), quien desarrolló el concepto [5].

El pensamiento computacional se define como una nueva competencia que debe ser incluida en la formación de todos los niños, ya que representa un ingrediente vital del aprendizaje de la ciencia, la tecnología, la ingeniería y las matemáticas (las STEM) [6]. Es decir, “El pensamiento computacional implica resolver problemas, diseñar sistemas y comprender el comportamiento humano, haciendo uso de los conceptos fundamentales de la informática”¹. Por tanto, la esencia del pensamiento computacional consiste en pensar como lo haría un científico informático cuando se enfrenta a un problema [7] [8] [9] [10].

El pensamiento computacional consta de cuatro fases fundamentales, que están interrelacionadas entre sí: (1) descomposición, (2) reconocimiento y generalización de patrones, (3) abstracción y (4) algoritmia. La resolución de un problema complejo a través de la teoría del pensamiento computacional requiere el uso acumulativo de las cuatro fases:

1. **Descomposición**

La fase de descomposición trata de fraccionar un problema o un sistema complejo en problemas más pequeños y, por tanto, de más fácil comprensión.

De este modo, las fracciones reducidas del problema pueden ser examinadas y resueltas independientemente. Cabe incluso repetir el proceso de descomposición sucesivamente hasta obtener problemas lo suficientemente simples como para que resulte cómodo trabajar con ellos.

2. **Reconocimiento y generalización de patrones**

La etapa de reconocimiento y generalización de patrones permite, en la mayoría de los casos, identificar aspectos compartidos entre los problemas más simples que se han generado en el anterior paso.

En este caso, los patrones podrían definirse como “las similitudes o características comunes de los problemas hallados tras la fase de descomposición”.

Cuanto mayor sea el número de patrones detectados, más fácil y más rápida será la resolución del problema. Ya que, si dos o más partes de un mismo problema complejo comparten un mismo patrón, se podrá aplicar a todas ellas una metodología similar, o incluso idéntica, para alcanzar su solución.

3. **Abstracción**

La abstracción es el proceso de filtración de ciertos patrones y detalles específicos.

En esta fase deberán eliminarse todos aquellos detalles específicos del problema (características no generalizadas y no compartidas con otros fragmentos del problema) y cualquier patrón que no ayude a su resolución. Esto permite crear una representación general de cuál es el problema y cómo resolverlo, un “modelo”.

¹ Cita original [6]: “*Computational thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science*”

4. Algoritmia

Esta cuarta y última fase y consiste en desarrollar un algoritmo que solucione los modelos que se han generado en las fases previas y, en última instancia, el problema complejo original.

Un algoritmo es un plan, un conjunto de instrucciones sucesivas que permiten la resolución de un determinado problema. Un algoritmo permite identificar cada una de las instrucciones que lo componen, así como el orden en que deberán ser aplicadas. Por ello, la estructura de un algoritmo debe ser clara y constar de (i) un punto de partida, (ii) un conjunto de instrucciones plasmadas de forma limpia y ordenada y (iii) un fin.

Los algoritmos se utilizan frecuentemente como un punto de partida para la creación de programas informáticos y pueden representarse como diagramas de flujo o en pseudocódigo.

En la práctica, la aplicación de la teoría del pensamiento computacional en procesos educativos o como método resolutivo de cualquier problema (y, especialmente, de los informáticos) resulta sencilla, ya que consiste en la realización de procesos lógicos básicos.

En una clase con niños en la que se pretende resolver problemas complejos usando pensamiento computacional, tras explicar de forma comprensible las cuatro fases que lo componen, el profesor debe guiarles en la aplicación de las mismas. Los alumnos deben llevar a cabo este proceso de forma libre, y siendo conscientes de que no existe una única solución para un mismo problema. De hecho, para la evaluación de actividades desarrollados con base en la teoría del pensamiento computacional, se recomienda usar técnicas tales como:

- Autoevaluación: es el propio alumno el que valora su trabajo.
- Coevaluación (evaluación entre pares): los alumnos se evalúan entre sí.
- El establecimiento de objetivos como medio para el fomento del aprendizaje independiente, el uso de preguntas abiertas como, por ejemplo: “*¿Por qué has elegido esta solución y no otra?*” o “*¿Puedes explicar cómo funciona?*”.
- Enfoques “KWL” (*Know/Want/Learn*) [11], que consisten en preguntar a los estudiantes, al inicio de la clase, (i) qué es lo que ya conocen y (ii) qué desean aprender y, a la finalización de la misma, (iii) qué han aprendido, como medio para el establecimiento de objetivos y de autoevaluación.

3. Estudio de herramientas para la enseñanza de informática

A la hora de realizar una clase o actividad destinada a enseñar informática a niños, es necesario utilizar algún recurso, alguna herramienta, que ayude a los alumnos a comprender, aprender, y usar los conceptos informáticos que se les están explicando.

En este tercer capítulo, se realiza una breve categorización y estudio de las herramientas existentes orientadas a enseñar informática a niños, describiendo en mayor detalle aquellos ejemplos más destacables. Finalmente, se exponen las conclusiones sobre los datos recopilados acerca de estas herramientas. La última de ellas es la lista de herramientas seleccionadas para ser utilizadas en las actividades que se diseñan, realizan y evalúan posteriormente en este TFG.

Actualmente, existe una gran variedad de herramientas disponibles en el mercado. Todas ellas cuentan con multitud de características distintas, como son: plataforma sobre la que se ejecutan, si necesitan de un ordenador o no, si requieren de una conexión a internet, si son de pago único, suscripción o gratuitas, edad para la que están adaptadas, principios informáticos o de otras disciplinas que utilizan y tratan de enseñar, etc.

Al disponer de tantas características distintas, existen distintos métodos de clasificación: por edad para la que están adaptadas [12], por precio final de la herramienta (contando licencia Software y Hardware adicional, en caso de necesitarlo) [13], por nivel general de dificultad de la herramienta, etc. [14] [15]. Sin embargo, estas clasificaciones son solo una enumeración de herramientas informando de sus características principales, de modo que se opta por hacer uso de la siguiente clasificación que diferencia entre (1) herramientas de programación visual, (2) herramientas relacionadas con la robótica y (3) herramientas que trabajan sin ordenador.

Se decide usar esta clasificación porque proporciona una idea general de qué tipo de herramientas existen según con que materiales y recursos trabajan, lo que implica una estructura y contexto de las actividades en las que se aplican muy distinto. No es lo mismo desarrollar una actividad sin ordenadores, que otra con ordenador y robots que otra con sólo el ordenador.

1. **Herramientas de programación visual:** permiten a los usuarios crear programas manipulando elementos gráficos en lugar de tener que especificar una serie de órdenes escritas, en algunos casos también se denominan como herramientas de programación por bloques [16] [17].
2. **Herramientas relacionadas con la robótica:** si bien estas herramientas también pueden trabajar con programación visual, necesitan de material extra para funcionar correctamente, un hardware adicional que consta de distintos elementos relacionados con la robótica, tales como servos, sensores de luz o infrarrojos, leds, etc.
3. **Herramientas sin ordenador:** aquellas herramientas que enseñan conceptos de carácter informático (programación, lógica booleana, pensamiento computacional, etc.) pero sin hacer uso del ordenador ni otros elementos tecnológicos.

Puesto que resulta imposible citar todas las herramientas pertenecientes a estas tres categorías, se nombran aquellos ejemplos más destacados que, además, cumplen las siguientes condiciones:

- Están diseñadas para alumnos entre 6 y 12 años.
- En el desarrollo de una actividad educativa o clase, permiten una aplicación de las teorías y conceptos de educación STEM y pensamiento computacional.
- Disponen de elementos que permitan enseñar, al menos, los siguientes conceptos básicos de informática y programación [18] [19] [20]:
 - Qué es un programa y su estructura básica.
 - Declaración y uso de variables.
 - Lógica booleana y operadores relacionales [21].
 - Instrucciones condicionales.
 - Instrucciones iterativas (Bucles).

3.1 Herramientas de programación visual.

La mayoría de las herramientas de programación orientadas a niños hacen uso de esta técnica. Esto se debe a que, para los niños, resulta más sencillo y ameno arrastrar y trabajar con elementos visuales que escribir una serie de instrucciones. Adicionalmente, resulta menos complicado entender y relacionarse con estos elementos que el tener que trabajar con una programación escrita tradicional, ya que no es necesario mecanografiar las instrucciones o trabajar siempre en inglés.

De este tipo de herramientas, cientos de ellas están basadas en el proyecto de código abierto de Google Blockly [22] [23] [24], que comenzó a desarrollarse en 2011. Blockly es una librería de JavaScript que permite crear editores y lenguajes de programación por bloques de código y que puede generar código JavaScript, Python y PHP entre otros.

A continuación, se presentan cuatro ejemplos de herramientas de programación por bloques, dos primeros que trabajan con la librería Blockly, y otros dos restantes que trabajan con otros tipos de tecnología:

- **Code** [25]: Ofrece una herramienta de programación visual por bloques con multitud de cursos y programas educativos, orientados para edades comprendidas desde los 4 hasta los 18 años. Todas estas actividades pueden encontrarse en su web [26], y algunas de ellas, no todas, disponen de una versión de escritorio para trabajar sin conexión a internet [27] [28].

Las actividades siguen un orden de dificultad creciente, primero se plantean problemas de resolución casi inmediata, que sirven como presentación de un nuevo bloque o instrucción, después, en cada nivel siguiente, se va complicando el uso de este bloque, explotando sus posibilidades y mezclándolo con otros bloques ya conocidos por el alumno. Siempre se trabaja con distintas temáticas que resulten llamativas para los niños, como son personajes de películas y juegos infantiles.

En la Ilustración 1, se observa un problema típico planteado por Code para la actividad de “*Make a Flappy Game: An Hour of Code Tutorial*” [29] en el nivel 9 [30] de dicha actividad. De izquierda a derecha, primero se dispone del juego que se está programando y que se puede ejecutar para comprobar que se ha completado el nivel correctamente. A continuación, se tiene la lista de bloques que se pueden utilizar y finalmente el panel sobre el que disponer los bloques en el orden correcto y crear el programa.



Ilustración 1 - Code - Flappy lvl 9

- **LearnToMod** [31]: Es una herramienta web de pago por suscripción que permite desarrollar pequeños programas (a partir de ahora “*mods*”) que se ejecutan dentro del videojuego Minecraft² y que está orientada para niños de 7 años en adelante [32].

Dispone tres secciones: una primera que permite el desarrollo de estos *mods*, una segunda destinada a generar estructuras (haciendo uso de una aplicación de modelado en 3D) y, finalmente, una tercera que sirve para crear packs de texturas (que alteran los gráficos de los distintos elementos que componen el videojuego Minecraft). Haciendo una combinación de estas tres secciones existe la posibilidad de crear proyectos que modifiquen en su totalidad la temática y objetivos del videojuego original de Minecraft.

Para su aplicación como herramienta docente, resulta interesante la posibilidad de programar *mods* que realicen distintas acciones dentro del juego, como generar estructuras, invocar entidades, reaccionar a eventos, etc. Para ello, LearnToMod incluye tutoriales con ejercicios diseñados para enseñar cada una de estas posibilidades utilizando programación por bloques.

En la Ilustración 2, se tiene un ejemplo de ejercicio, “*Parameters: Monster House (Scramble)*” dentro del tutorial “*Skills and Drills - Learning To Program - Functions With Parameters*”. En este tipo de ejercicio (*Scramble*), al principio el alumno dispone

² Videojuego desarrollado por Mojang de construcción usando bloques de distintos materiales, más información: <https://minecraft.net>

de todas las instrucciones que componen el programa, pero desordenadas, y ha de posicionarlas en el orden correcto para avanzar al siguiente ejercicio del tutorial.

LearnToMod cuenta también con otros tres tipos de actividades en sus tutoriales: actividades tipo “missing” en las que se encuentran todos los bloques del programa en el orden correcto, pero faltando algunos parámetros, cuestionarios, y enunciados que piden hacer un programa desde cero guiando al alumno paso a paso.



Ilustración 2 - LearnToMod - Skills and Drills (Using the Old Interface), Parameters: Monster House (Scramble)

- **Scratch** [33]: Aplicación web y de escritorio [34] gratuita desarrollada por el grupo de investigación “Lifelong Kindergarten” [35] del MIT en el 2007 [36]. A diferencia de las anteriores, esta herramienta no está basada en la librería Blockly, aunque el MIT está colaborando actualmente con Google para sacar una nueva versión de Blockly conjunta, Scratch Blocks [37].

Scratch está diseñado especialmente para niños con edades comprendidas entre los 8 y 16 años [38] y permite crear historias interactivas, juegos y animaciones haciendo uso de la programación por bloques. Además, se pueden importar todo tipos de sonidos e imágenes que posteriormente animar, y pueden compartirse públicamente todos los proyectos que se desee (siempre que se disponga de una cuenta, que son de creación y uso gratuito).

La interfaz de trabajo de Scratch es similar a la empleada por la herramienta Code. Tal y como se observa en la Ilustración 3, a la izquierda se dispone de un área donde se encuentran los elementos que estamos programando/animando (en este caso, el gato básico de Scratch), en el centro se tienen, clasificados por categorías, todos los distintos bloques que se pueden utilizar para su animación y programación. Finalmente, a la derecha, un área de trabajo donde colocar estos bloques en el orden que se desee para crear un programa o animación.

En este caso particular (Ilustración 3), se anima el gato haciendo que mande al principio un mensaje de “Hello” y, acto seguido, entre un bucle infinito en el que emitirá un

sonido, después girara un número determinado de grados en sentido de las agujas del reloj, y finalmente cambiará sus gráficos.

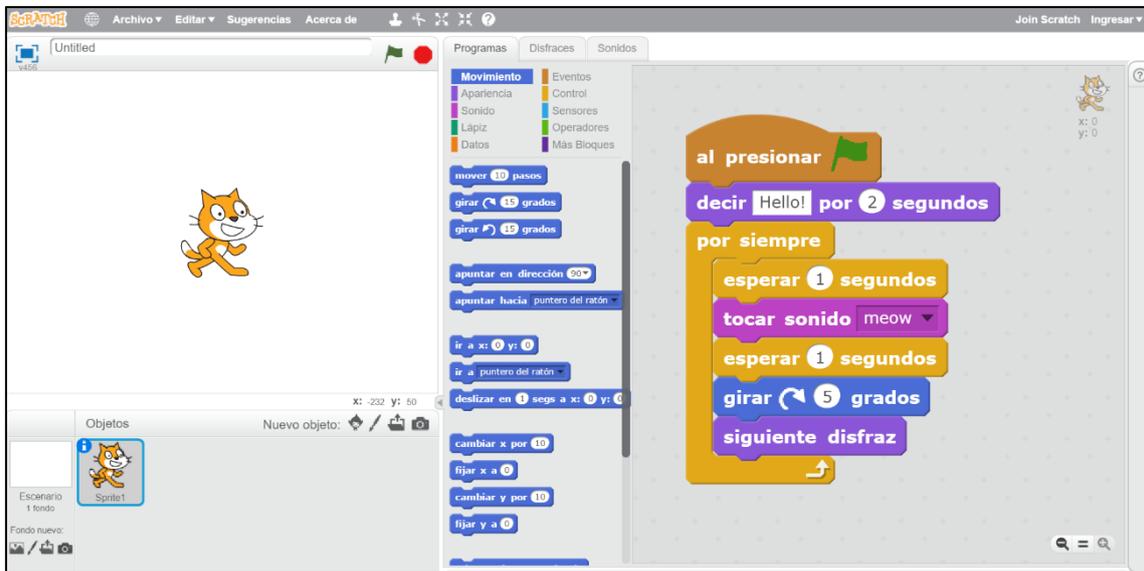


Ilustración 3 – Scratch – Programación libre

- **Minecraft Education Edition** [39]: Es una versión modificada del videojuego Minecraft que está específicamente diseñada para la docencia. Fue desarrollada por Mojang AB y Microsoft Studios y, aparte de los elementos comunes del videojuego Minecraft, dispone de extras que facilitan su aplicación como herramienta educativa. El juego completo fue lanzado como aplicación de escritorio [40] en 2016, y dispone actividades preparadas para niños de 5 años en adelante [41] [42] [43] [44].

En la propia web de Minecraft Education Edition es posible acceder a multitud de clases y actividades preparadas por la comunidad o por la empresa [45]. En la Ilustración 4 se observa la herramienta dentro del tutorial inicial [46] diseñado para que tanto los alumnos, como el profesor, se familiaricen con el control y elementos básicos del videojuego de Minecraft.



Ilustración 4 - Minecraft Education Edition - Inicio Tutorial

Aparte de las posibilidades existentes, desde mayo de 2017 [47], esta herramienta cuenta con la extensión “Code Builder for Minecraft” (Véase Ilustración 5), que permite trabajar con código dentro de la propia herramienta de una forma similar a como trabajan las tres herramientas anteriores [48].

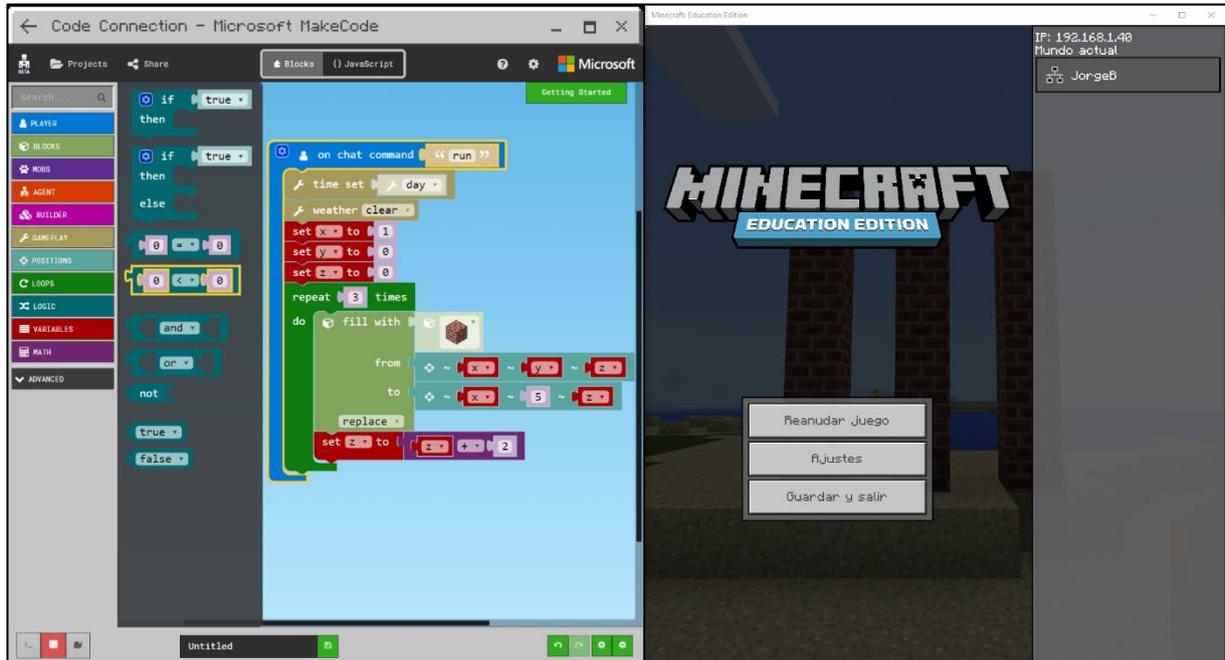


Ilustración 5 - Code Builder for Minecraft: Education Edition

Finalmente, en la Tabla 1 quedan recogidas las principales características de las 4 herramientas descritas en esta sección.

Tabla 1 Comparativa Herramientas de programación visual

| Características | | |
|---|--|--|
| Edad recomendada: | 8 años o más. | 10 años o más. |
| Nivel de experiencia requerido: | Principiante. | Principiante. |
| ¿Multi idioma? | Si. | Si. |
| Precio: | 76€ cada kit "BQ Mi primer Kit de Robótica" [66]. | 399.99€ cada robot "LEGO MindStorms EV3" [67]. |
| Requisitos herramienta | | |
| ¿Necesita de Ordenador? | Si. | Si. |
| Sistemas operativos: | Windows y Mac OS. | Windows y Mac OS. |
| ¿Necesita de conexión a internet? | Si. | No. |
| Necesita de material extra, ¿Cual? | Si, kits de robótica o robots arduino. | Si, robots "Lego MindStorms". |
| Características trabajo con la herramienta | | |
| ¿Trabajo individual, por parejas o por grupos? | Individual y por parejas. | Individual y por parejas. |
| ¿Más conceptos informáticos? ¿Cuáles? | Funciones, funciones con parámetros, mapear valores y más. | Programar características componentes, números aleatorios y más. |
| ¿Dispone de programas educativos/actividades propias? | Si. | Si. |
| ¿Cuál es la duración estimada de las actividades? | Desde media hora. | Desde 1 hora. |
| ¿Permite publicar o compartir los trabajos/proyectos? | Si. | Si. |

3.2 Herramientas relacionadas con la robótica

A la hora de emplear herramientas relacionadas con la robótica, se ha de tener en cuenta que, aunque el software que se utiliza para programar, y con el que interaccionan y trabajan los niños y el profesor, sí que es gratuito, el hardware no, por lo que siempre que se quiera trabajar con herramientas de este tipo es necesaria una inversión en material.

Seguidamente, se dispone de dos ejemplos de herramientas relacionadas con la robótica:

- **Bitbloq** [51]: Herramienta web de programación por bloques creada por BQ [52], que permite programar distintos modelos de placas Arduino aparte de ser compatible con los kits de robótica y robots [53] ofertados por esta misma empresa.

La herramienta está orientada para niños a partir de los 6 u 8 años [54], y trabaja con los siguientes elementos de hardware: leds, servos, zumbadores, botones, potenciómetros, sensores de infrarrojos, sonido, temperatura y humedad, sensores de luz y sigue-líneas, y dispositivos que permiten la comunicación bluetooth y puerto serie. Bitbloq cuenta con unos tutoriales básicos para aprender a trabajar con cada uno de estos componentes [55] y que permite compartir proyectos y buscar otros realizados por la comunidad [56].

Como se observa en la Ilustración 6, cuenta con varias secciones, una primera que permite realizar un modelo virtual del ensamblado de las piezas del robot, otra que permite la programación del mismo, que cuenta con una interfaz similar a las descritas para las herramientas anteriores (Code, Scratch y LearnToMod), y una última para añadir información adicional al proyecto en caso de que se desee publicarlo.

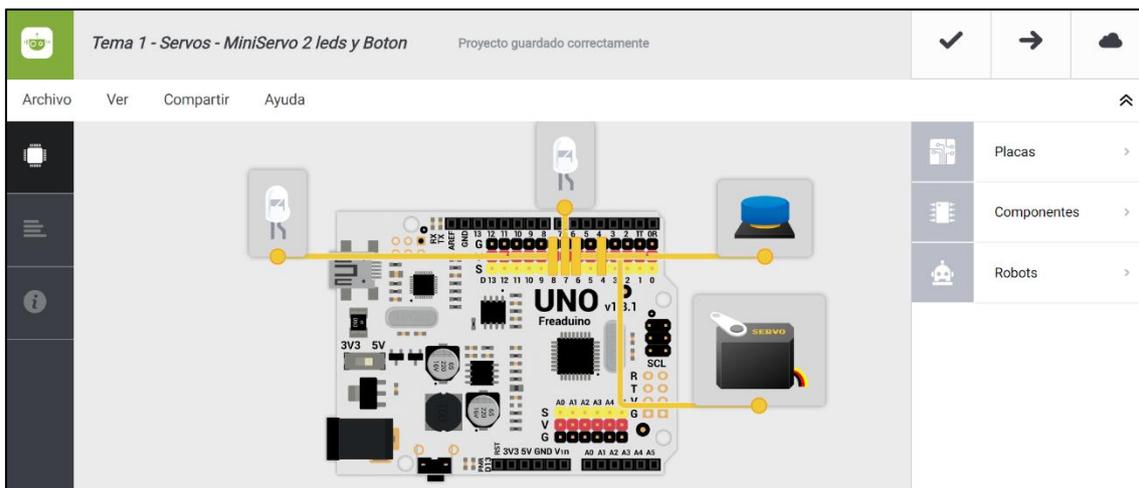


Ilustración 6 - Bitbloq - Programación libre

- **LEGO MindStorms** [57]: Herramientas de robótica para niños fabricadas por LEGO que están basadas en el “Programmable Brick” del MIT [58] [59]. La primera versión fue comercializada en 1998 y su versión actual cuenta con 4 sensores distintos, 2 servos, el bloque programable que actúa como placa base del robot y piezas de LEGO, tal y como se ve en la Ilustración 7 [60], además de una aplicación de escritorio que permite programar estos componentes [61]. Esta herramienta cuenta con programas

predefinidos por LEGO para cada uno de sus robots, y que están explicados pasos a paso, aunque también se puede realizar programas propios usando los componentes que deseemos y publicarlos [62]. LEGO MindStorms está recomendada para niños a partir de 10 años [63] [64].

Tal y como se observa en la Ilustración 8, LEGO sigue un enfoque de programación visual distinto a los descritos hasta ahora. En este caso no todos los bloques se identifican con una instrucción de programación, sino que también se dispone de bloques identificativos de cada uno de los componentes de robotica de los que hace uso. Estos bloques permiten modificar los parametros característicos de cada componente que reprensetan, por ejemplo, en la Ilustración 8 (perteneciente a las instrucciones del robot *EV3RSTORM03* v6 [65]) se ve como en el bloque “Sensor IR” se modifican uno de sus parametros para que funcione a una proximidad de “70”.



Ilustración 7 - Piezas y componentes del robot Lego MindStorms EV3

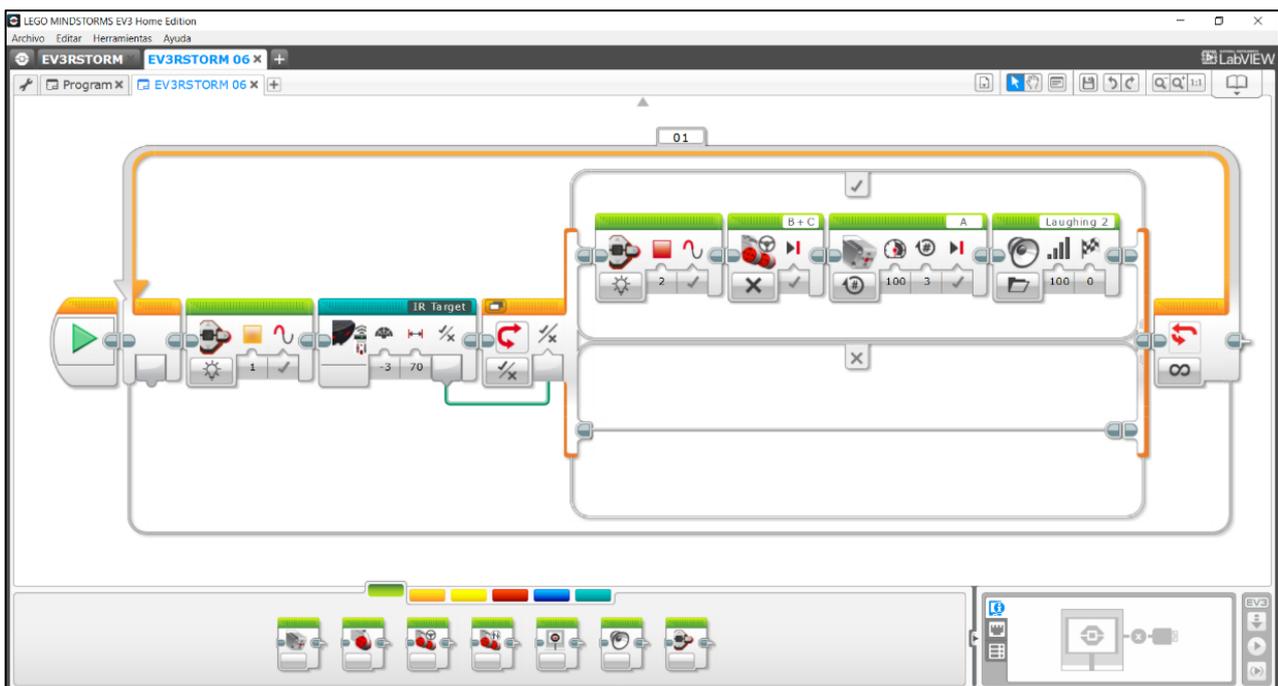


Ilustración 8 - LEGO MindStorms - Programación robot EV03 Storm, paso 6

Al igual que en la sección previa, en la Tabla 2 se dispone de información resumida acerca de las dos herramientas, Bitbloq y LEGO MindStorms, descritas en esta sección.

Tabla 2 Comparativa Herramientas relacionadas con la robótica

| | Bitbloq | LEGO MindStorms |
|---|--|--|
| Características | | |
| Edad recomendada: | 8 años o más. | 10 años o más. |
| Nivel de experiencia requerido: | Principiante. | Principiante. |
| ¿Multi idioma? | Si. | Si. |
| Precio: | 76€ cada kit "BQ Mi primer Kit de Robótica" [66]. | 399.99€ cada robot "LEGO MindStorms EV3" [67]. |
| Requisitos herramienta | | |
| ¿Necesita de Ordenador? | Si. | Si. |
| Sistemas operativos: | Windows y Mac OS. | Windows y Mac OS. |
| ¿Necesita de conexión a internet? | Si. | No. |
| Necesita de material extra, ¿Cual? | Si, kits de robótica o robots arduino. | Si, robots "Lego MindStorms". |
| Características trabajo con la herramienta | | |
| ¿Trabajo individual, por parejas o por grupos? | Individual y por parejas. | Individual y por parejas. |
| ¿Más conceptos informáticos? ¿Cuáles? | Funciones, funciones con parámetros, mapear valores y más. | Programar características componentes, números aleatorios y más. |
| ¿Dispone de programas educativos/actividades propias? | Si. | Si. |
| ¿Cuál es la duración estimada de las actividades? | Desde media hora. | Desde 1 hora. |
| ¿Permite publicar o compartir los trabajos/proyectos? | Si. | Si. |

3.3 Herramientas sin ordenador

A esta categoría pertenece toda aquella herramienta que, aunque esté diseñadas para enseñar conceptos relacionados con la informática y la programación, no cuenta con ningún elemento tecnológico (ni trabajo con ordenadores, robots u otro tipo de hardware).

En su mayoría se trata de juegos de mesa, como lo son las siguientes dos herramientas que se describen en mayor detalle:

- **Robot Turtles** [68]: Es un juego de mesa orientado a enseñar programación a niños pequeños que comenzó siendo un proyecto de Kickstarter y que está orientado para niños a partir de los 3 años [69].

El funcionamiento del juego es sencillo, cada jugador dispone de una carta tortuga y un mazo con cartas de movimientos, el objetivo es hacer llegar las cartas tortuga a las cartas joya de su mismo color, desplazándolas por el tablero, y esquivando o destruyendo los posibles obstáculos que se encuentren en él camino, tal y como se ve en la Ilustración 9 (Imagen 1, hipotética situación inicial del tablero, imagen 2, ejemplo

de cartas de movimiento, imagen 3 resolución del tablero de la imagen 1 usando cartas de movimiento) [70].

A lo largo del desarrollo del juego, cada uno de los niños está elaborando un programa, un algoritmo compuesto de distintas instrucciones de movimiento y acción que son necesarias para alcanzar el objetivo final. Si algún jugador se da cuenta de que ha cometido un fallo, puede gritar “Bug” y deshacer el error.



Ilustración 9 - Robot Turtles - Instrucciones

- **Code Monkey Island** [71]: Es otro proyecto financiado por Kickstarter que está orientado a enseñar programación a niños con edades a partir de 4 años. Las partidas tienen un tiempo estimado de 45 min de duración, y está diseñado para jugar de 2 a 4 jugadores [72].

El juego consta de un tablero, 12 figuras de mono (de tres colores distintos) y 80 cartas de guía, fruta y aumentos (Véase Ilustración 11). Las cartas guía sirven para desplazar los monos por el tablero, los otros dos tipos de cartas son ventajas que se pueden obtener si se alcanzan determinadas casillas. El objetivo del juego es completar el recorrido de todo el tablero con los tres monos [73] [74].

Tal y como se ve en la Ilustración 12, las cartas de movimiento incluyen conceptos de programación como son condicionales, lógica booleana y operadores booleanos, y bucles [75].

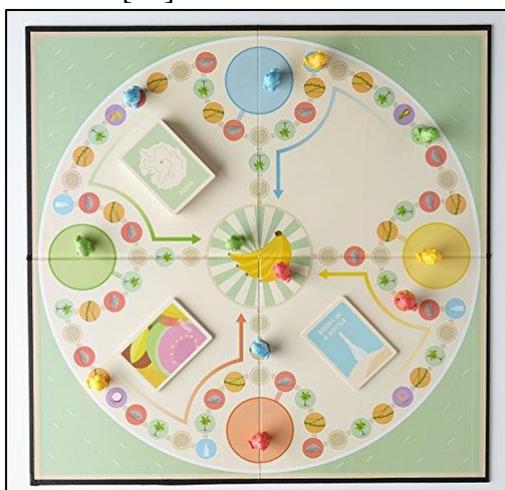


Ilustración 11 - Tablero Code Monkey Island



Ilustración 10 - Cartas guía Code Monkey Island

Finalmente, en la Tabla 3 queda recopilada información resumida y relacionada con estas dos herramientas, Robot Turtles y Code Monkey Island.

Tabla 3 Comparativa Herramientas sin ordenador

| | Robot Turtles | Code Monkey Island |
|---|----------------------------------|----------------------------------|
| Características | | |
| Edad recomendada: | A partir de 3 años. | A partir de 4 años. |
| Nivel de experiencia requerido: | Ninguno. | Ninguno. |
| ¿Multi idioma? | No. | No. |
| Precio: | 22.76\$ [76]. | 19.99\$ [77]. |
| Requisitos herramienta | | |
| ¿Necesita de Ordenador? | No. | No. |
| Sistemas operativos: | | |
| ¿Necesita de conexión a internet? | | |
| Necesita de material extra, ¿Cual? | No, solo el juego de mesa. | No, solo el juego de mesa. |
| Características trabajo con la herramienta | | |
| ¿Trabajo individual, por parejas o por grupos? | Por grupos. | Por grupos. |
| ¿Más conceptos informáticos? ¿Cuáles? | No. | No. |
| ¿Dispone de programas educativos/actividades propias? | Si, las instrucciones del juego. | Si, las instrucciones del juego. |
| ¿Cuál es la duración estimada de las actividades? | Desde media hora. | Desde 45 minutos. |
| ¿Permite publicar o compartir los trabajos/proyectos? | | |

3.4 Conclusiones

Teniendo en cuenta la información recogida de cada una de las herramientas estudiadas y descritas en las secciones anteriores, y el resumen de sus características indicado en las Tablas 1, 2 y 3, se concluye que:

1. En las **herramientas de programación visual**, destaca la existencia de varias dedicadas a trabajar en conjunto con el videojuego Minecraft, aprovechando el interés que despierta esta característica en los niños. Se considera interesante explotar este interés y, de ser posible, se utilizará una herramienta de este tipo.

Adicionalmente, se comprueba que las herramientas de Code y Scratch, de uso gratuito y gran popularidad, resultan muy similares, y es por ello que se intentará recabar más información acerca de ellas, para decidir cuál usar en las actividades.

2. De las **herramientas orientadas a robótica**, queda descartada la opción de trabajar y evaluar LEGO MindStorms por su alto precio y por la gran cantidad de piezas de las que dispone (véase Ilustración 7), que provocarían que en una clase

se dedique demasiado tiempo a una actividad de “montaje del robot” que no se desea evaluar.

La herramienta Bitbloq resulta más atractiva por disponer en sus kits, sólo de componentes relacionados con la robótica, aparte de tener un precio más asequible. Igualmente, salvo poder alquilar o tomar prestado el material, se tendrá que descartar esta opción.

3. Deben descartarse para futuros análisis y evaluaciones aquellas herramientas pertenecientes a la categoría de **herramientas sin ordenador**, ya que se desea realizar un análisis de aquellas que sí que lo utilizan. Además, las pertenecientes a estas categorías son juegos de mesa o juguetes preparados para 2 o 4 jugadores, lo que dificultaría aplicarlas a una clase con un número mayor de alumnos, y, generalmente, mayores que la edad objetivo ideal de estas herramientas.

De modo que, una vez realizado este estudio previo, se concluye que las siguientes herramientas son las que pueden ser empleadas en las actividades que se diseñan y evalúan más adelante: Code, LearnToMod, Scratch, Minecraft Education Edition y Bitbloq, en caso de poder asumir los costes de material (por comprar, alquilar o por préstamo).

4. Metodología

En este capítulo queda descrita la metodología que se sigue en el resto del documento, y que puede observarse en la Ilustración 12, “Diagrama metodología”.

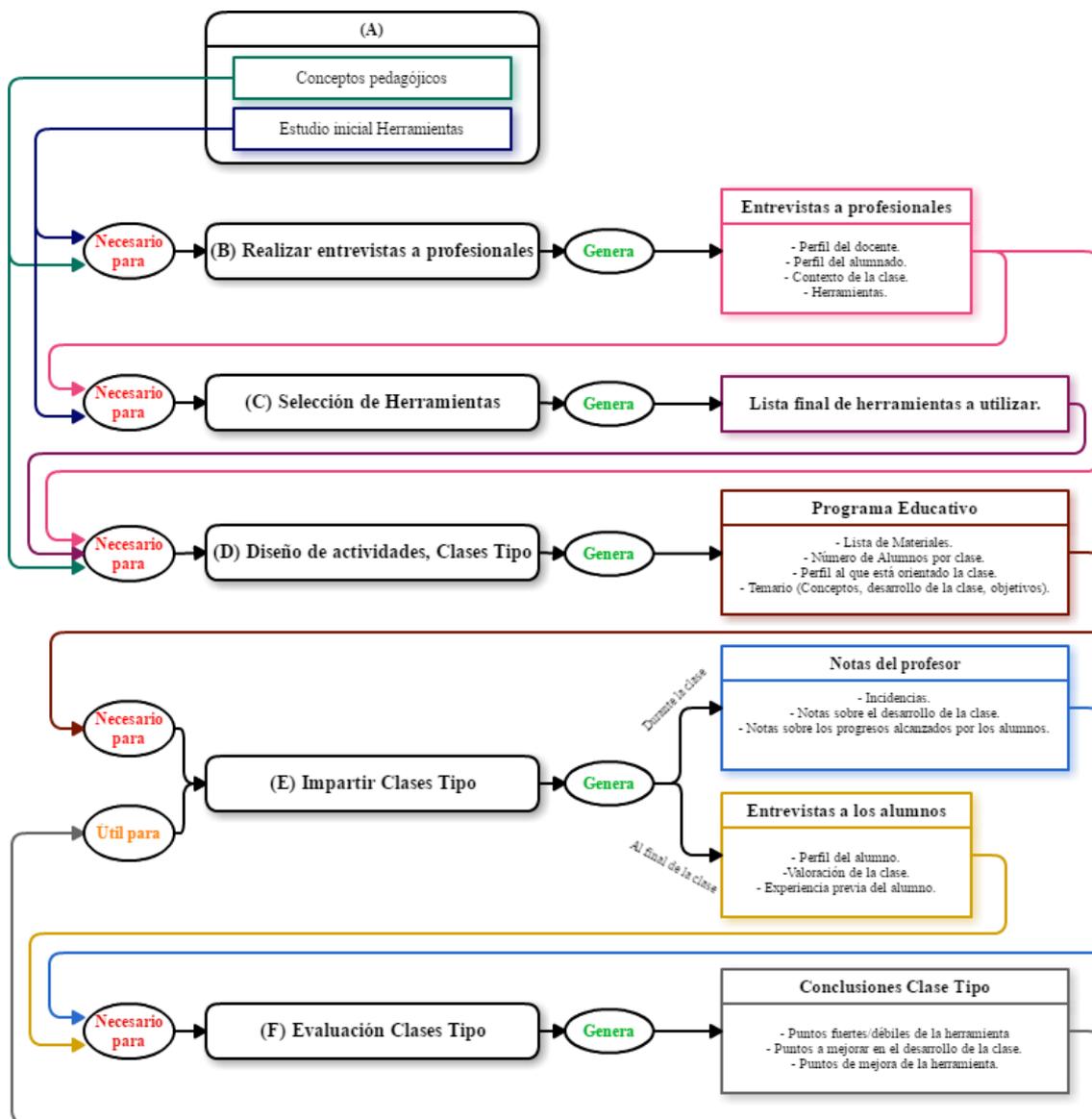


Ilustración 12 - Diagrama metodología

En los capítulos 2 y 3 de este documento se dispone de la información recopilada acerca de qué estrategias y conceptos educativos utilizar en las actividades, y del estudio previo sobre las herramientas existentes en el mercado y que se pueden usar en las actividades, respectivamente. Estos dos capítulos son el estado inicial del diagrama, el **paso A**.

Resulta necesario delimitar aún más la lista de herramientas aspirantes a ser utilizadas en las actividades, y para ello se recurre a la opinión de profesionales del sector (entiéndase por profesional, toda aquella persona que trabaje como docente en el ámbito de la informática con niños).

A cada uno de estos profesionales, se le realiza una entrevista individual acerca del tema (**Paso B del diagrama**), de las que se espera recopilar información práctica acerca de los cuatro elementos que componen una de sus actividades educativas, es decir: (1) perfil del profesional que imparte la actividad, (2) el perfil del alumnado que participa en la actividad, (3) el contexto de la actividad, y (4) las herramientas y recursos que utiliza en la actividad.

Tras el análisis de los datos obtenidos a través de estas entrevistas, y recordando también la información del estudio inicial, se escogen el subconjunto final de herramientas que se utilizarán en las actividades (**Paso C del Diagrama**).

Una vez que se tiene este subconjunto final de herramientas se procede a:

- 1º) Diseñar un conjunto de actividades, en adelante “Clases Tipo”, (**Paso D del Diagrama**) teniendo en cuenta:
 - i. Los datos extraídos de las entrevistas, acerca de los perfiles del docente y su alumnado, y el contexto de sus clases.
 - ii. Que se harán uso de las herramientas finalmente seleccionadas.
 - iii. Que se desea enseñar conceptos básicos relacionados con la informática y la programación a niños pequeños, aplicando, en la medida de lo posible, los conceptos de educación STEM y pensamiento educacional.
- 2º) Impartir las Clases Tipo (**Paso E del Diagrama**) siguiendo el programa educativo especificado y tomando notas de cualquier incidencia o aspecto que el profesor considere interesante a lo largo de su desarrollo. Además, al final de cada una de las sesiones, se evalúan -a través de una pequeña entrevista individual- la experiencia personal de cada uno de los alumnos en la utilización de la herramienta y, en la medida de lo posible, los conocimientos adquiridos mediante el uso de la misma.
- 3º) De los datos obtenidos en las entrevistas y de las notas del profesor, se espera poder extraer una evaluación general de cada una de las Clases Tipo (**Paso F del Diagrama**), a través de la identificación de características comunes, aspectos positivos y negativos, deficiencias y detalles que deben ser objeto de mejora, etc. Además, en las primeras sesiones, es posible que se obtengan conclusiones de la experiencia como profesor que, aunque no modifiquen el diseño de las Clases Tipo que aún no se hayan impartido, sirvan como mejora a lo largo de la ejecución de las mismas.

Una vez realizados todos los pasos de la metodología de este capítulo, se espera haber alcanzado los siguientes objetivos. y desde este paso inicial se comienza a recorrer los siguientes pasos con objetivo de:

1. Determinar qué subconjunto final de herramientas utilizar en las actividades, ya que, pese al estudio inicial realizado, las posibilidades siguen siendo demasiado numerosas (con los pasos B y C).
2. Tener datos e información acerca de cómo elaborar y realizar el diseño, ejecución y evaluación de las actividades con las herramientas finalmente seleccionadas (con los pasos B, D, E y F).

5. Información práctica, Entrevistas

En este capítulo, primero se detalla el diseño que se sigue, respecto al contenido y forma, de las entrevistas realizadas a los profesionales del sector. A continuación, se exponen los resultados de las mismas en distintas tablas, datos y conclusiones, expuestos en cada una de las correspondientes subsecciones (5.2.1 a 5.2.4).

5.1 Diseño de entrevistas

Como se indica en el capítulo previo, se decide realizar un conjunto de entrevistas a profesionales que estén relacionados, o que trabajen, en el ámbito de la docencia enseñando conceptos informáticos o de programación a niños.

Estas entrevistas están orientadas a la obtención de datos e información útil y práctica con respecto a los ítems que componen una actividad de este tipo, es decir, acerca del perfil de los (1) docentes y (2) sus alumnos, así como (3) del contexto de las clases, y (4) sobre qué herramientas emplean en las mismas:

1. **El perfil del docente:** se abordan las siguientes cuestiones acerca del educador; (i) espacio en el que desarrolla su actividad docente y si ha trabajado en algún otro centro, (ii) cargo profesional, (iii) su formación previa, (iv) su experiencia académica en actividades de enseñanza a niños de conocimientos informáticos y de programación (años y número de cursos que imparte), (v) motivos que le llevaron a desempeñar su labor docente relacionada con la informática y (vi) su opinión sobre la relevancia de este tipo de enseñanza y sobre los aspectos clave para su éxito.
2. **El perfil del alumnado:** se trata de determinar, (i) el rango de edades que tienen los alumnos, (ii) las motivaciones para asistir a este tipo de cursos, (iii) porcentaje de alumnos masculinos y femeninos y (iv) si tienen conocimientos previos sobre informática.
3. **El contexto de la clase:** se analizarán principalmente datos relativos a los siguientes parámetros (tanto de su estado real como el del ideal considerado por el docente): (i) número de alumnos por clase, (ii) duración y (iii) frecuencia de las clases, (iv) información sobre su desarrollo, dinámica y metodología de trabajo en las clases y (v) que conceptos informáticos se imparten.
4. **Opinión herramientas:** finalmente, se desea conocer: (i) qué herramientas se emplean a lo largo de la clase, (ii) por qué utilizan esas herramientas y (iii) si desearían emplear alguna herramienta que no usan actualmente. Además, se quiere disponer de información adicional sobre las herramientas Bitbloq, Code y Scratch, herramientas clasificadas como posibles (sección 3.4 y de las que se desea obtener información adicional con el objetivo de descartar o elegir alguna de ellas para el subconjunto final. Se trata de averiguar sobre cada una de ellas: (iv) el conocimiento que tienen de la misma, (v) su uso y (vi) sus eventuales propuestas de mejora.

5.2 Resultados de las entrevistas

En esta sección se muestra, de manera resumida, la información extraída de las entrevistas, teniendo en el Anexo A, sus transcripciones completas.

A la hora de mostrar esta información, se decide hacerlo usando tablas dedicadas a cada uno de los cuatro apartados principales de las entrevistas. La decisión de disponer de dos tablas, en lugar de una por cada uno de los apartados, atiende solamente a una decisión de maquetación. Cada pareja de tablas dispone de los mismos campos y preguntas, solo variando los datos aportados por los distintos profesionales que se entrevistan. Esta decisión se mantiene en el resto de secciones en las que se recoge información de las entrevistas (secciones de la 5.2.1 a la 5.2.4).

5.2.1 Perfil del docente

La información recogida acerca del perfil de los docentes entrevistados queda reflejada y resumida en las tablas 4 y 5.

Tabla 4 - Entrevistas a Profesionales – Perfil del Docente 1de2

| Perfil del Docente | | | |
|---|--|--|---|
| | Adrián Blasco | Jesús Mínguez | David Bordonada |
| Perfil Personal | | | |
| Centro en el que imparte clase actualmente el docente | Academia, Enseñalia. | Colegio, Santa María del Pilar Marianistas. | Colegios. |
| ¿Ha trabajado o imparte formación en otros centros? | Sí, asociación Dlabs. | No. | No. |
| Profesión | Profesor y Coordinador de formaciones STEM de la empresa Enseñalia. | Profesor y Administrador de Sistemas. | Profesor/Asesor de kits de robótica de Libelium y comercial de Libelium. |
| Estudios | Universitarios, Ingeniería Industrial. | Universitarios, Ingeniería Electrónica Industrial. | Universitarios, Ciencias Empresariales. |
| Tiempo impartiendo (años) | 3 | 15 | 3 |
| Número de cursos distintos que imparte | 7 | 9 | 3 |
| ¿Por qué empezaste a educar en STEM? | Buscaba trabajo. | Se requería por parte del colegio. | Se requería por parte de la empresa. |
| Opinión Educación STEM | | | |
| ¿Por qué consideras importante de impartir educación relacionada con las STEM en edades tempranas? | Si que creo que es importante porque conceptos que se explican en la universidad se podrían introducir de una forma más reducida antes. Actualmente los niños tienen contacto directo con la tecnología desde muy pequeños y se debe enseñar a manejarla responsablemente. | Si que creo que es importante porque creo que se debe empezar a fomentar el pensamiento computacional lo antes posible. | Si que creo que es importante porque creo que está bien ir fomentando en nuevas tecnologías a los niños desde que son pequeños. |
| ¿Cuál consideras que es la edad ideal en la que empezar a recibir educación relacionada con la informática? | Con 6 años, primero de primaria. Aunque creo que no se debería enseñar por edades, sino por desarrollo personal del niño y de su interés por aprender. | Con 2-3 años, cuanto antes mejor. Con esa edad ya existen herramientas adaptadas para enseñar pensamiento computacional. | Con 6 años, desde primero de primaria. |

Tabla 5 - Entrevistas a Profesionales - Perfil del Docente 2de2

| Perfil del Docente | | | | |
|---|--|---|--|---|
| | Daniel Espinosa y Raquel Garcia | Pablo Aliaga | Beatriz Garcia | Andreu Camps |
| Perfil Personal | | | | |
| Centro en el que imparte clase actualmente el docente | Colegios y academia Crom Developer. | Asociación, Dlabs. | Colegios y academia NiuBit. | Colegios y academia Avanza Tiempo libre. |
| ¿Ha trabajado o imparte formación en otros centros? | No, solo cuando nos han contratado otros colegios para dar clases extra escolares. | Si, en otros colegios, academias y empresas dedicadas a la educación. | No, solo cuando nos han contratado otros colegios para dar clases extra escolares. | Si, en la asociación de Altas Capacidades de Aragón. |
| Profesión | Profesores y fundadores de la academia Crom Developer. | Profesor autónomo y presidente de la asociación Dlabs. | Profesora y co-fundadora de la academia NiuBit. | Profesor y coordinador de la empresa Avanza Tiempo Libre. |
| Estudios | Grado Superior, Desarrollo de Aplicaciones Informáticas y FP, Desarrollo de Aplicaciones Multiplataforma. | Universitarios, Ingeniería Informática. | Universitaria, Ingeniería Informática. | Bachillerato y cursos (monitor de tiempo libre etc.) no relacionados con la informática. |
| Tiempo impartiendo (años) | 5 | 1 | 4 | 4 |
| Número de cursos distintos que imparte | 1 (genérico y adaptado) | Más de 4 | 2 (genéricos y adaptados) | 2 (genéricos y adaptados) |
| ¿Por qué empezaste a educar en STEM? | Porque vimos que había mercado y decidimos crear la empresa. | Porque ahora hay oferta de profesor para niños, aunque prefiero impartir formación a adultos. | Porque vimos que había mercado y decidimos crear la empresa. | Comprobamos que con la robótica se puede enseñar informática fácilmente y decidimos desarrollarla. |
| Opinión Educación STEM | | | | |
| ¿Por qué consideras importante de impartir educación relacionada con las STEM en edades tempranas? | Si que creemos que es importante, pero también creemos que se está pasando la gente apuntando a niños muy jóvenes que no son capaces ni de interaccionar correctamente con un ordenador, manejar el ratón etc. | Si que creo que es importante porque se debe descubrir a los niños el mundo de la informática y las tecnologías y que trabajen con ellas. También para enseñarles que hay detrás de una web o el funcionamiento de un ordenador, que sepan ver el esfuerzo que implica hacer un programa. | Si que creo que es importante, creo que la robótica y la programación deberían de entrar en el plan de estudios estándar. Es algo que ayuda a los niños a organizarse y entenderse mejor con la tecnología. En el futuro, es muy probable que se necesite interactuar con la tecnología a diario. | Si que creo que es importante. Tratamos hacer una robótica "de campaña", para niños, para niñas, para el que le guste y para el que no. Que en el futuro si se quieren dedicar a estudiar una carrera STEM estupendo, pero que, si optan por otra, cuenten con recursos tecnológicos que ya conozcan. |
| ¿Cuál consideras que es la edad ideal en la que empezar a recibir educación relacionada con la informática? | Con 8 años en adelante, con esa edad los niños tienen suficiente capacidad de razonamiento y psicomotriz para interactuar con un ordenador y aprender informática y robótica. | Con 6 años, enfoco la formación para que sea como un juego y que aprendan jugando. Se puede enseñar a cualquier edad siempre que no se pierda ese enfoque, aunque también habría que tener en cuenta las capacidades de cada niño. | Con 6 años, primero de primaria. En edades más tempranas, como en infantil, si que hay algunos conceptos de robótica que podrían ir introduciéndose y podrían llegar a entender, pero tiene otra complejidad añadida porque los niños aún no saben leer ni escribir. Sería otro tipo de formación. | Con 6 años, primero de primaria. Más pequeños, hemos hecho pruebas con infantil, pero es inviable, son demasiado pequeños para que puedan hacer las actividades. |

Acerca del **perfil personal** del docente destacan las siguientes cuestiones:

- Todos los entrevistados trabajan como docentes impartiendo clases o colaborando en el desarrollo de las mismas, pudiendo asumir en algunos casos, roles extra en sus trabajos.
- Cuentan con una media de 4 años de experiencia en el sector.
- Que el 71% de los entrevistados contaba con una formación universitaria.
- Que solo el 71% de todos ellos habían cursado estudios que estuviesen relacionados con la informática.

Además, acerca de la **educación STEM** y la edad ideal en la que comenzar a impartirla, los entrevistados consideran que:

- El 100%, que impartir formación de este tipo a niños pequeños resulta importante o muy importante, llegando a reclamar que se empiece a incluir en los planes de estudio estándar.
- El, 71%, cree que se debería empezar con 6 años, en primero de primaria.

5.2.2 Perfil del alumnado

A continuación, se pasa a recopilar los datos (visibles en las tablas 6 y 7) sobre el perfil de los alumnos que tienen los docentes entrevistados.

Tabla 6 - Entrevistas a Profesionales - Perfil del Alumnado 1de2

| Perfil Alumnado | | | |
|--|---|---|---|
| | Adrián Blasco | Jesús Mínguez | David Bordonada |
| Rango de edad | Entre 7 y 13 años. | Entre 10 y 12 años. | Entre 6 y 16 años. |
| ¿Los alumnos asisten a clase por iniciativa propia o por iniciativa de los padres? | Iniciativa propia. | Forma parte del currículo del colegio. | Iniciativa propia. |
| ¿Cual es la relación de niños/niñas? | 33% son niñas. | 50% son niñas. | 33% son niñas. |
| ¿Los alumnos cuentan con conocimientos previos relacionados con la informática? | Mayoría de alumnos que no tienen conocimientos previos. | Mayoría de alumnos que no tienen conocimientos previos. | Igualdad entre número de alumnos con y sin conocimientos previos. |

Tabla 7 - Entrevistas a Profesionales - Perfil del Alumnado 2de2

| Perfil Alumnado | | | | |
|--|---|---|---|---|
| | Daniel Espinosa y Raquel Garcia | Pablo Aliaga | Beatriz Garcia | Andreu Camps |
| Rango de edad | Entre 8 y 16 años. | Entre 6 y 14 años. | Entre 6 y 16 años. | Entre 6 y 16 años. |
| ¿Los alumnos asisten a clase por iniciativa propia o por iniciativa de los padres? | Iniciativa propia y de los padres. | Iniciativa propia y de los padres. | Iniciativa propia y de los padres. | Iniciativa de los padres. |
| ¿Cual es la relación de niños/niñas? | 44% son niñas (y menos conforme se aumenta la edad). | 44% son niñas. | 33% niñas en la academia, 50% en los colegios. | 50% son niñas. |
| ¿Los alumnos cuentan con conocimientos previos relacionados con la informática? | Igualdad entre número de alumnos con y sin conocimientos previos. | Mayoría de alumnos con conocimientos previos o que han participado actividades similares. | Mayoría de alumnos con conocimientos previos o que han participado actividades similares. | Mayoría de alumnos que no tienen conocimientos previos. |

Sobre el **perfil del alumnado** se observa que:

- Un 42% de los entrevistados enseñan a niños entre 6 y 16 años de edad, que es a su vez, las edades mínima y máxima con las que trabajan.
- De media, existe una proporción de 40% de alumnas por clase.
- Un 28% de los entrevistados dice que sus alumnos acuden a actividades de este tipo (clases relacionadas con la informática, robótica, etc.) por iniciativa propia, y un 42% indica que tanto por iniciativa propia como por iniciativa de sus padres.
- Sobre los conocimientos previos, el 42% declaran que tienen alumnos de todo tipo, teniendo alumnos que ya han participado en clases similares y alumnos que son completamente nuevos a este tipo de actividades.

Acerca de la **asistencia, duración y frecuencia** de las clases, se han obtenido los siguientes resultados:

- De media, en las clases participan de 14 a 18 alumnos, y la idoneidad, según los entrevistados, sería de entre 8 y 10 alumnos por clase.
- La duración media de las clases varía entre 50 y 240 minutos, aunque la fórmula mayoritaria, 42%, es de 60 minutos, que a su vez coincide con la que consideran la más adecuada el 57% de los profesionales.
- El 71% de los entrevistados imparte una clase semanal, periodicidad considerada idónea también por ese mismo porcentaje de entrevistados.

Con respecto a la **dinámica de las clases**, destacan las siguientes consideraciones:

- El 57% imparte sus clases con un enfoque tanto lúdico como educacional.
- El 57% realiza habitualmente o siempre, actividades de trabajo en grupo en las que los alumnos comparten el material.
- En el 100% de los casos, los ordenadores con los que trabajan los alumnos los proporciona el propio centro donde se realizan las clases.
- El 57% utiliza siempre el ordenador en sus clases, y el 43% restante lo utiliza muy frecuentemente.
- Ninguno de los entrevistados pone deberes o actividades para realizar en casa. Tampoco hacen ningún tipo de examen o evaluación a sus alumnos.

Y, finalmente, sobre los **conceptos** que se emplean y tratan de transmitir, se constata que:

- El 71% enseñan conceptos básicos de programación, como bucles, condicionales, etc. El resto de profesionales utiliza otro enfoque para enseñar estos conceptos u otros que estén intrínsecamente relacionados (pensamiento computacional, etc.)
- La mayoría, 71%, están contentos con los conceptos que enseñan y no incluirían ninguno más.

5.2.4 Opinión herramientas

Finalmente, se pregunta a los encuestados acerca de que herramientas utilizan en sus clases, y particularmente, acerca de la opinión que les merecen las herramientas Bitbloq, Code y Scratch. Véase tablas 10 y 11.

En lo que se refiere a la **opinión de las herramientas** que utilizan los profesionales entrevistados en sus clases, se observa que:

- El 85% utilizan solo herramientas que tengan una versión de escritorio, es decir, que no necesiten de una conexión a internet para su funcionamiento.
- Todos los entrevistados utilizan al menos una herramienta que trabaje con kits de robótica o robots y una que trabaje con programación por bloques.
- Solo el 28% trabajan con una herramienta relacionada con el videojuego Minecraft.
- El 71% está satisfecho con las herramientas que usa actualmente y no desearía probar/utilizar ninguna otra.

Acerca de la opinión que tienen sobre **Bitbloq**:

- Es conocida por el 100% de los entrevistados.
- Utilizada habitualmente por el 14%, y probada por el 42%.
- Echan en falta que fuese una aplicación de escritorio y tuviese mejor soporte por BQ.

Acerca de la opinión que tienen sobre **Code.org**:

- Es conocida por el 85% de los entrevistados.
- Utilizada habitualmente por el 28%, y probada por el 42%.
- Echan en falta que fuese una aplicación de escritorio.

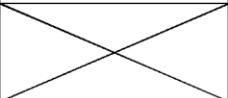
Acerca de la opinión que tienen sobre **Scratch**:

- Es conocida por el 100% de los entrevistados.
- Utilizada siempre por un 28%, habitualmente por el 42%, y probada por el 28%.
- Echan en falta que estuviese mejor integrada con Arduino y una mejor organización de los proyectos públicos (que han subido otros usuarios de la herramienta).

Tabla 10 - Entrevistas a Profesionales - Opinión herramientas 1de2

| Opinión herramientas | | | |
|--|--|---|---|
| | Adrián Blasco | Jesús Mínguez | David Bordonada |
| ¿Qué herramientas utilizas? | BitBloq, TinkerCAD, LearnToMod, Snap for Arduino, Scratch y Code.org. | El IDE de Arduino y Scratch. | Scratch y el fork Snap for Arduino (S4A). |
| ¿Por qué utilizas esas herramientas? | Porque son las que considero mejores en sus "categorías", y son herramientas web multiplataforma y abiertas. | El IDE porque acapara todo lo de programación básica. Y Scratch y Code son sencillos y cubren todas las necesidades para los cursos de Primaria. | Porque son los que funcionan mejor con nuestros kits. |
| ¿Alguna herramienta adicional que te gustaría usar? | Me gustaría disponer de kits de robótica o componentes especializados en comunicaciones. | Me gustaría probar los kits de robótica de Lego. | No. |
| BitBloq | | | |
| ¿Conoces esta herramienta? | Si. | Si. | Si. |
| ¿Utilizas esta herramienta? | Si, habitualmente. | La he probado/la he usado poco. | No. |
| ¿Qué hechas en falta/qué mejorarías de esta herramienta? | Soporte por parte de BQ, ya que a veces no funciona correctamente. Y que si pudieran utilizar mas componentes. | Prefiero trabajar con el IDE de Arduino porque no falla y siempre conecta bien. Además preferiria que fuese una aplicación de escritorio offline. | |
| Code.org | | | |
| ¿Conoces esta herramienta? | Si. | Si. | No. |
| ¿Utilizas esta herramienta? | Si, habitualmente | No. | |
| ¿Qué hechas en falta/qué mejorarías de esta herramienta? | Nada, es una herramienta muy completa. | | |
| Scratch | | | |
| ¿Conoces esta herramienta? | Si. | Si. | Si. |
| ¿Utilizas esta herramienta? | La he probado/la he usado poco. | Si, habitualmente. | Si, habitualmente (usando el fork S4A). |
| ¿Qué hechas en falta/qué mejorarías de esta herramienta? | Hay tantas proyectos disponibles, que encontrar proyectos y actividades útiles resulta complicado, ya que se mezclan con proyectos más amateurs. | Nada. | Nada. |

Tabla 11 - Entrevistas a Profesionales - Opinión herramientas 2de2

| Opinión herramientas | | | | |
|--|---|--|---|---|
| | Daniel Espinosa y Raquel García | Pablo Aliaga | Beatriz García | Andreu Camps |
| ¿Qué herramientas utilizas? | Scratch, S4A, Mblock y SculptGL. | Scratch y S4A. | Scratch, IDE de Arduino, kits de Lego y Minecraft para diseño 3D y programación. | Kits de Lego y Lego MindStorms. |
| ¿Por qué utilizas esas herramientas? | Intentamos usar software libre que sea multiplataforma y que no dependa de una conexión a Internet. | Porque tiene todas las ventajas y potencia de Scratch y además puedes programar Arduino. | Kits de Lego porque llevan muchísimos años trabajando con el tema de la robótica orientada a niños y las otras herramientas son fáciles de enseñar con ellas. | Que sea una aplicación de escritorio en lugar de una web. Hemos probado todo, pero las herramientas de Lego son, en nuestra opinión, las mejores. |
| ¿Alguna herramienta adicional que te gustaría usar? | No. | No. | No. | No. |
| BitBloq | | | | |
| ¿Conoces esta herramienta? | Si. | Si. | Si. | Si. |
| ¿Utilizas esta herramienta? | La he probado/la he usado poco. | La he probado/la he usado poco. | No | La he probado/la he usado poco. |
| ¿Qué hechas en falta/qué mejorías de esta herramienta? | Que fuese una herramienta libre. | No me gusta que sea una plataforma online, debería ser una aplicación de escritorio. |  | Que fuese una aplicación de escritorio. |
| Code.org | | | | |
| ¿Conoces esta herramienta? | Si | Si | Si | Si |
| ¿Utilizas esta herramienta? | La he probado/la he usado poco. | La he probado/la he usado poco. | Si, habitualmente. | La he probado/la he usado poco. |
| ¿Qué hechas en falta/qué mejorías de esta herramienta? | Preferiría que fuese una aplicación de escritorio offline. | No me gusta que sea una plataforma online, debería ser una aplicación de escritorio. | Nada. | Que fuese una aplicación de escritorio. |
| Scratch | | | | |
| ¿Conoces esta herramienta? | Si. | Si. | Si. | Si. |
| ¿Utilizas esta herramienta? | Si, siempre. | Si, habitualmente. | Si, siempre. | La he probado/la he usado poco. |
| ¿Qué hechas en falta/qué mejorías de esta herramienta? | Nosotros nada, pero otros profesores nos han comentado que preferirían que tuviesen más opciones de programación. | Que integre las características que tiene S4A. Mejor comunicación con placas Arduino. | Nada. | Que fuese una aplicación de escritorio. Que tuviese algún apoyo externo, un kit de robótica. |

6. Diseño de actividades, Clases Tipo

En este capítulo se relata el proceso de diseño de las actividades, que de aquí en adelante denominaremos Clases Tipo.

Para diseñar estas Clases Tipo, se consideran los cuatro elementos principales que componen una actividad, es decir: qué perfil ha de cumplir el profesor que la imparte, a qué tipo de niños está orientada, características relacionadas con su contexto (duración, número de alumnos, metodología educativa a seguir, conceptos a enseñar, etc.) y qué herramientas emplear en ellas. Para tomar estas decisiones se ha tenido en cuenta la información recogida en los capítulos 2, 3 y 5 de este trabajo, que incluyen los conceptos educativos que se emplean para enseñar informática a niños (2), el estudio previo de herramientas disponibles en el mercado (3), y los resultados de las entrevistas (5).

Además, a la hora de plantear y ejecutar estas clases, se cuenta con la colaboración de un centro académico que se ofrece a aportar, tanto los alumnos, como los espacios y materiales (ordenadores, licencias, kits de robótica, etc.) de los que disponen.

De este modo, este capítulo incluye la selección de herramientas finales, a continuación, se detallan los requisitos comunes de las Clases Tipo, los materiales que son necesarios en todas las actividades, las características de la localización donde impartirlas, etc. Después se indican las cuestiones de diseño comunes a las actividades, como son su duración, edad objetivo, número de alumnos, etc. Finalmente cuenta con una sección dedicada a recoger los programas educativos de cada una de las actividades que se realizan y cuestiones individuales de diseño de las mismas.

6.1 Selección de herramientas

Teniendo en cuenta (i) los materiales disponibles (facilitados por el centro académico), (ii) los resultados obtenidos en las entrevistas del capítulo previo, y (iii) el estudio inicial del capítulo 3, cuya conclusión es la selección de 5 posibles herramientas (véase sección 3.4), se ha decidido analizar las siguientes: **Code**, **LearnToMod** y **Bitbloq**.

Se han optado por estas herramientas por los siguientes motivos:

- Se deseaba evaluar una herramienta a elegir entre Code y Scratch, y, tras los resultados obtenidos en las entrevistas, se observa que Code no era tan conocida y usada como Scratch. Es por ello que se opta por Code, con ánimo de trabajar con la herramienta menos popular de las dos.
- Se deseaba evaluar una herramienta que interaccionase con el videojuego Minecraft, siendo las dos opciones en este caso, LearnToMod y Minecraft Education Edition. Puesto que solo se dispone de licencias para la herramienta LearnToMod, se elige esta opción.
- Finalmente, como se tiene acceso a los kits de robótica y materiales necesarios, se decide diseñar y realizar una clase con la herramienta relacionada con la robótica Bitbloq.

6.2 Requisitos comunes de las Clases Tipo

En todas las clases son necesarios materiales y recursos con los que los alumnos puedan trabajar y desarrollar las actividades. A continuación, se procede a enumerar aquellos que son comunes a todas las Clases Tipo.

Los materiales básicos para poder llevar a cabo la actividad son los correspondientes a los de un aula de informática, es decir, contar con un equipo (con Windows o Mac)³ por cada alumno asistente, más uno para ser usado por el profesor.

Adicionalmente, para facilitar la tarea del docente, son necesarios:

- Un televisor o proyector que permita al profesor conectar su equipo para que los alumnos puedan ver cómo se soluciona paso a paso alguno de los problemas, o mostrar algún ejemplo o solución completa cuando se considere necesario.
- Una pizarra, en la que el profesor pueda dibujar y explicar conceptos adicionales.

Finalmente, en cuanto al aula, es preciso que cuente con las siguientes características:

- Una conexión a internet que permita que estén trabajando al mismo tiempo todos los equipos (los de los alumnos y el del profesor) sin desconexiones, ya que en todas las actividades se usan herramientas que requieren de una conexión a internet.
- Suficiente espacio de trabajo en el aula para que se pueda realizar la actividad cómodamente.

6.3 Aspectos de diseño comunes de las Clases Tipo

Pese a que se van a emplear distintas herramientas en el desarrollo de las Clases Tipo, existen ciertas cuestiones de diseño que se han decidido aplicar a todas ellas con el fin de homogeneizar, en la medida de lo posible, el tipo de actividades que se van a realizar, y facilitar así su posterior evaluación.

En todas ellas **el profesor** que imparte la actividad deberá contar con un perfil similar al que poseen la gran mayoría de los entrevistados, es decir, universitario relacionado con la informática, y con cierta experiencia en la docencia con niños.

Además, **sobre el perfil del alumnado** al que están orientadas todas las actividades planteadas se decide que van a ser:

- Actividades diseñadas para trabajar con niños con edades comprendidas entre los 7 y 10 años. Se establecen estos límites porque se desea trabajar con niños en un rango de edades lo más amplio posible dentro de una misma actividad.

³ Descartados aquellos equipos que funcionen bajo una distribución Linux debido a los requisitos de las herramientas que se van a utilizar, véanse secciones 6.1 y 3.4.

- Clases en las que no es necesario estar familiarizado con la herramienta que se va a usar, o tener conocimientos previos acerca de conceptos de informática o programación, es decir, los alumnos asistentes no tienen porqué haber participado en actividades similares con anterioridad.

Adicionalmente, se toman las siguientes decisiones de diseño comunes **respecto al contexto de la clase**:

- Las clases están orientadas para un mínimo de 6 alumnos y un máximo de 10. Se trata de grupos menos numerosos de los que trabaja la media de los profesionales entrevistados, esto se debe a que se desea atender de una forma más personalizada a cada uno de los niños participantes, por lo que es necesario trabajar con un número relativamente reducido.
- Se crean actividades diseñadas para tener una duración estimada de entre 60 y 90 minutos. Estos eran los márgenes de tiempo más comunes entre las clases de los profesionales entrevistados, y también los que la mayoría de ellos consideraban ideales.
- Las actividades tendrán una orientación lúdica y educacional, como la mayoría de las clases impartidas por los entrevistados.
- En todas las actividades cada alumno trabajará de forma individual e independiente del resto, disponiendo de todo el material necesario para ello. Se decide trabajar así para conseguir que todos los alumnos realicen al completo todos los aspectos y problemas de las actividades planteadas, y puedan dar su opinión acerca de ellas.

Se definen tres **objetivos educativos** comunes a todas las Clases Tipo propuestas, de modo que los alumnos:

1. Se familiaricen con el uso de una herramienta de programación visual (como son Code, LearnToMod y Bitbloq, que aparte de estar relacionada con la robótica, cuenta con una parte importante de programación por bloques), es decir, que aprendan a:
 - a. Arrastrar, eliminar, duplicar y recolocar en el orden que se desee bloques de código.
 - b. Identificar correctamente para que sirve cada bloque y cuáles de ellos requieren especificar información extra.
2. Conozcan los siguientes conceptos a un nivel básico:
 - a. Qué es el código.
 - b. Qué es programar.
 - c. Qué es debuggear.
 - d. Qué es un programa.
 - e. Qué es un algoritmo.
3. Aprendan el concepto de bucle:
 - a. Qué es un bucle.
 - b. Qué es una iteración.
 - c. Para qué sirven los bucles.

Finalmente, es necesario remarcar los siguientes puntos acerca del **enfoque pedagógico** y los conceptos educativos que se emplean en las clases:

- En todas las actividades propuestas se sigue un método de educación STEM. Todas las actividades cuentan con objetivos educacionales centrados en conceptos relacionados con la informática o la programación, pero también con ideas o conceptos pertenecientes a otras disciplinas, como son las ciencias, la ingeniería o las matemáticas, es decir, se aprovecha la programación y los recursos tecnológicos disponibles para enseñar competencias de otras áreas STEM.

Estos objetivos educacionales no relacionados con la informática se recogen en los programas educativos individuales de cada actividad, ya que no son comunes a todas las clases.

- Además, en todas ellas se utilizará un método de evaluación KWL, preguntando al inicio de la actividad qué es lo que conocen y qué es lo que desean aprender y, a la finalización de la actividad, qué es lo que han aprendido.
- Por otra parte, una vez se avance en la actividad y se afronten los primeros problemas de cierta dificultad, se explica el concepto de pensamiento computacional y cómo se emplea éste para resolver problemas complejos.

Primero se explicará brevemente, y en un lenguaje cercano a los alumnos, qué es el pensamiento computacional, los pasos que lo componen y para qué sirve, después, y a lo largo del desarrollo del resto de la clase, se animará a los alumnos a que empleen este método, guiándolos en el proceso si es necesario.

6.4 Programas educativos

Se ha decidido impartir cuatro Clases Tipo, dos de ellas con la herramienta Code, otra con la herramienta LearnToMod y una final con la herramienta de robótica Bitbloq. Se ha optado por realizar cuatro clases tipo porque se considera un número adecuado de sesiones para disponer de suficiente información con la que trabajar.

A continuación, se dispone de los programas educativos de cada una de estas clases.

6.4.1 Clases Tipo – Code – Programas educativos

Se decide realizar dos Clases Tipo de una hora de duración en las que se utiliza y evalúa la herramienta Code, en concreto dos de las actividades “*Hora del Código*” propuestas por esta herramienta [78]. En ambas clases se ha hecho uso de la herramienta a lo largo de toda la sesión.

En este caso, no se trata de diseñar dos programas educativos desde cero, ya que Code cuenta con programas educativos muy completos para todas sus actividades, incluyendo las actividades de “*Hora del Código*”. Se trata, por tanto, de elegir, o, de ser necesario, adaptar, dos programas educativos que cumplan con las decisiones de diseño descritas en la sección anterior (6.3).

A – Objetivos Educativos

Además de los objetivos educativos comunes, estos dos programas educativos abordan las siguientes cuestiones o conceptos extra, que se pretende que el alumno haya entendido y aprendido para el final de la sesión [79] [80]:

1. Concepto de instrucción condicional (Sólo en la actividad de “Escribe tu primer programa”):
 - a. Qué es una condición.
 - b. Qué es una instrucción condicional “si => haz”.
 - c. Qué es una instrucción condicional “si => haz / sino”.
2. Conceptos relacionados con la orientación espacial en 2 dimensiones, en los dos casos, y manejo de grados en una circunferencia (Sólo en la actividad de “Programa con Ana y Elsa”).

B - Programas educativos

En cada una de estas dos actividades se planten un total de 20 niveles de dificultad, cada nivel propone un problema distinto a resolver, siendo los primeros niveles una simple introducción a la herramienta y a los bloques/instrucciones que se usan.

Los niveles adquieren dificultad conforme se van superando, encontrando el problema más difícil de la actividad en el nivel 19 (véase Ilustración 13 y 14). El nivel 20 es un nivel de programación libre en los que se anima a los alumnos más aventajados que han terminado antes el resto de niveles, a explorar y experimentar libremente las instrucciones que han aprendido a lo largo de la sesión.

- La primera Clase Tipo hace uso de la actividad “**Hora del Código – Programa con Ana y Elsa**” [79] [81]. Esta actividad consiste en realizar una serie de figuras geométricas moviendo el personaje de “Ana” o “Elsa” de la película “Frozen”. Para ello los alumnos tienen que programar los movimientos que ha de realizar el personaje hasta conseguir dibujar una figura propuesta en cada nivel.

En el nivel 4 de la actividad se les pide que realicen un bucle de “X” iteraciones por primera vez, y en los siguientes niveles se complica el uso del mismo para crear distintas figuras. (Recordar que en esta clase no se enseñan los conceptos de bucle condicional o de instrucciones condicionales).

- En la segunda Clase Tipo se realiza la actividad “**Hora del Código – Escribe tu primer programa**” [80]. A diferencia de la actividad anterior, en este caso intervienen figuras de tres mundos distintos (“Angry Birds”, “Plants vs Zombies” e “Ice Age”) con ánimo de mantener la atención del alumno variando el escenario de la actividad en función del nivel de dificultad del que se trate. El objetivo en este caso es guiar a la figura a través de pequeños laberintos proporcionándole instrucciones de movimiento y orientación.

En esta actividad, el primer bucle de “X” iteraciones aparece en el nivel 7, el primer bucle condicional en el nivel 10, la primera instrucción condicional estándar en el nivel 14 y la primera instrucción condicional “si => haz / sino” en el nivel 18.

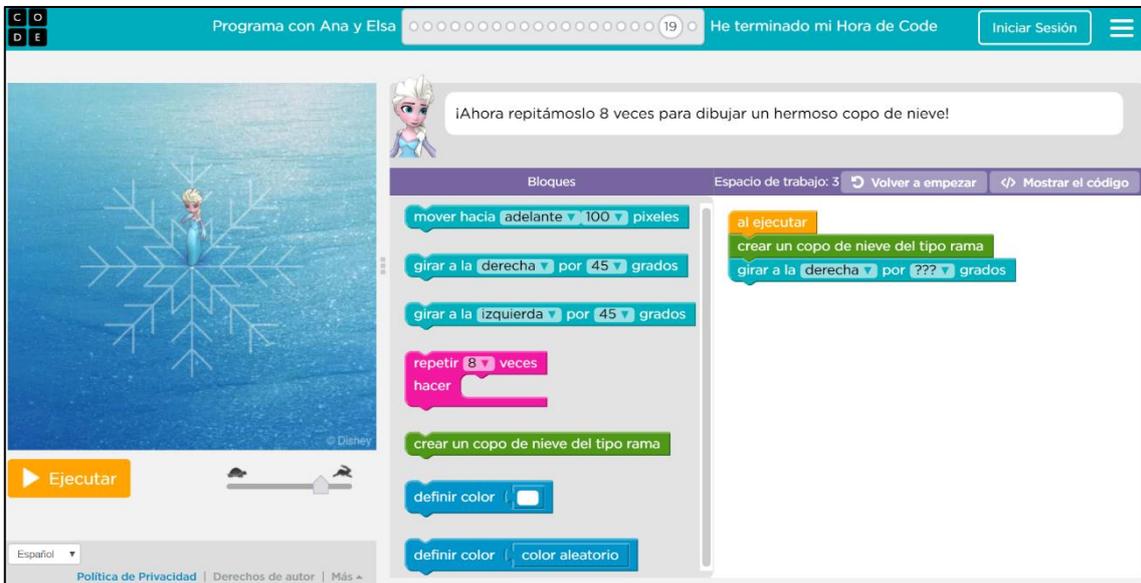


Ilustración 13 Hora del Código - Programa con Ana y Elsa, nivel 19



Ilustración 14 Hora del Código – Escribe tu primer programa

6.4.2 Clase Tipo – LearnToMod – Programa educativo

En este caso se diseña una actividad con la herramienta LearnToMod. Si bien esta herramienta cuenta con multitud de tutoriales que explican paso a paso sus posibilidades, no dispone de un “programa educativo” al uso, como sí disponía Code.

Es por esto que, teniendo en cuenta que es una clase orientada a alumnos que nunca hayan trabajado con esta herramienta, se ha escogido uno de los problemas de uno de sus tutoriales para desarrollar y completar a lo largo de una hora y media.

El problema elegido es el “1 – Introduction To Loops”, dentro de la actividad del mismo nombre, en el tutorial “Learning To Program” del curso “Skills and Drills (Using the Old Interface)” (Véase Ilustración 15, ya que no puede consultarse como recurso web puesto que se requiere una licencia de LearnToMod).

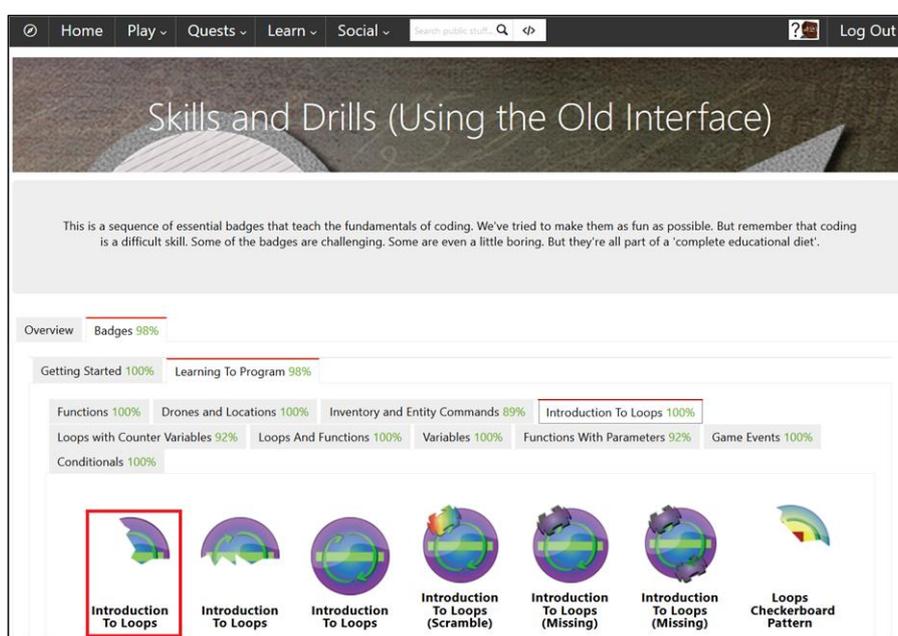


Ilustración 15 Actividad Introduction To Loops

A – Objetivos Educativos

Una vez finalizada la actividad, se espera que los alumnos comprendan los siguientes conceptos:

1. Concepto de función:
 - a. Qué es una función.
 - b. Para qué sirve una función.
 - c. Utilizar funciones.
2. Concepto de Drone:
 - a. Qué es un Drone.
 - b. Para qué sirve un Drone.
 - c. Cómo utilizar un Drone.
3. Conceptos relacionados con la orientación espacial en 3 dimensiones.

B - Programa educativo

En este caso se adapta el ejercicio de “*Introduction To Loops*” para enseñar desde cero todos los conceptos incluidos en los objetivos educacionales de esta Clase Tipo y en los objetivos comunes (sección 6.3). Es por ello que se plantea un avance en 3 pasos, enseñando, y practicando con pequeños problemas, cada nuevo concepto que se introduce a los alumnos y que resulte necesario para llegar a la resolución final del problema “*Introduction To Loops*”:

1. Primero se presentan los distintos bloques disponibles que estén relacionados con un Drone, es decir, aquellos bloques que crean un Drone, que lo mueven en el espacio, y que hacen que coloque, bloques de distinto tipo en la posición en la que se encuentra. Para ello se plantean dos ejercicios en los que los alumnos programen el Drone para colocar, a 3 posiciones hacia delante y 3 a la izquierda, un bloque de ladrillo. En el primer caso, se les enseña a hacerlo sin cambiar, en los bloques de movimiento, la distancia a recorrer, y en el segundo caso, modificándola (Véase Ilustración 16 imagen 1 y 2). Al final se explican las diferencias y las ventajas de la segunda opción frente a la primera [82] [83].
2. Una vez finalizados estos ejercicios introductorios, se presentan las instrucciones de bucle y como pueden ser utilizadas para ahorrarnos trabajo. A continuación, se pide al alumno que haga uso de los bucles para crear una torre de 10 pisos de ladrillos (Véase Ilustración 16 imagen 3) [84].
3. Finalmente, se explica el concepto de función y para qué sirven. Después, se solicita a los alumnos que, reciclando el código que han hecho en el anterior ejercicio (colocándolo en una función), creen una estructura que conste de 3 torres de 10 bloques de ladrillo de altura y separadas por un espacio entre si (Véase Ilustración 17). Una vez que todos los alumnos lo hayan conseguido, se procede a ejecutar el mod, ya finalizado, dentro del videojuego Minecraft (Véase Ilustración 18), ya que en los problemas anteriores se emplea sólo el simulador del que dispone LearnToMod [85].

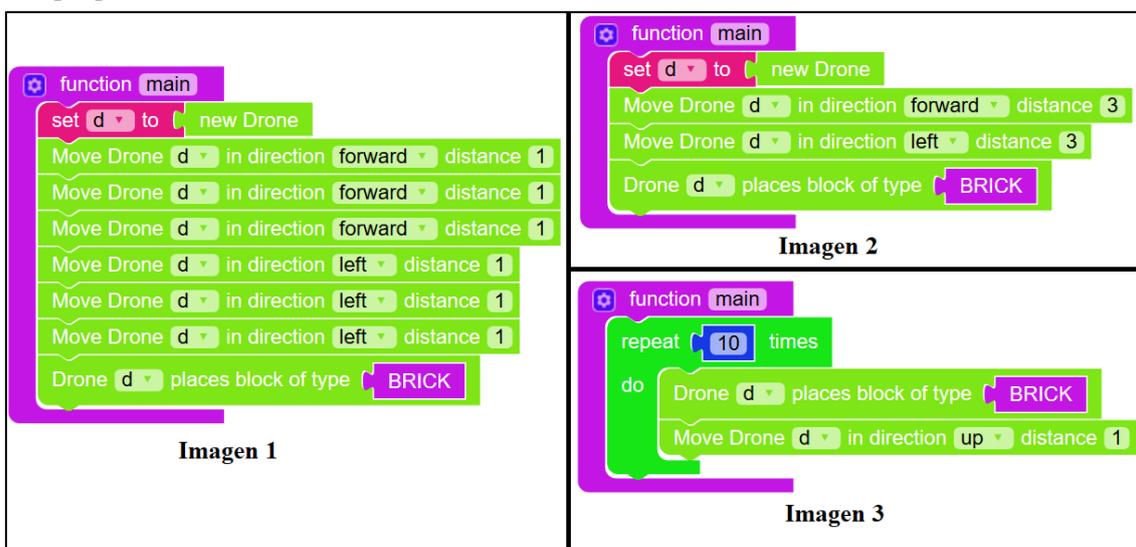


Ilustración 16 - LearnToMod - Solución a los problemas 1, 2 y 3

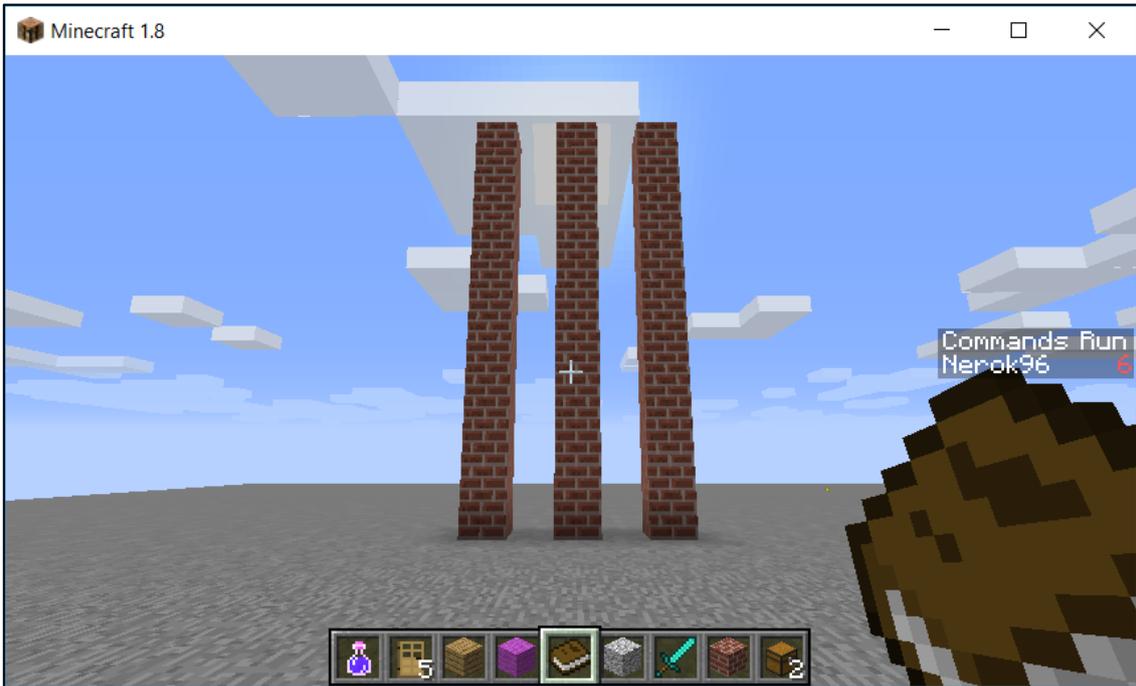


Ilustración 17 - LearnToMod - Resultado al ejecutar la solución del problema 4



Ilustración 18 - LearnToMod – Solución problema 4

6.4.3 Clase Tipo – Bitbloq – Programa educativo

Finalmente, tenemos el programa educativo de la actividad en la que se usa Bitbloq. Esta herramienta, como la anterior, dispone de tutoriales y pequeñas actividades pre-diseñadas por la empresa, BQ, para comprender el funcionamiento básico de los elementos que posee, pero no de programas educativos al uso.

Puesto que esta actividad, al igual que las demás, está diseñada para alumnos que nunca hayan trabajado con herramientas relacionadas con la robótica, se ha optado por adaptar una actividad sencilla publicada por BQ por motivo con Halloween llamada “*El Difunto Robonauta*” [86], realizando primero pequeños problemas que sirvan de introducción a los conceptos e instrucciones necesarios para completar la actividad. Además, destacar que el tiempo estimado de esta clase, al igual que en la anterior, será de 90 minutos.

A – Objetivos Educativos

Una vez finalizada la actividad, se espera que los alumnos comprendan los siguientes conceptos:

1. Concepto de instrucción condicional:
 - a. Qué es una condición.
 - b. Qué es una instrucción condicional “si => haz”.
 - c. Qué es una instrucción condicional “si => haz / sino”.
2. Que conozcan qué son, para qué sirven y el funcionamiento básico de los siguientes componentes robóticos:
 - a. Placa base.
 - b. Zumbador.
 - c. Mini Servo.
 - d. Sensor de Infrarrojos.

B - Programa educativo

Esta Clase Tipo basada en la actividad del Difunto Robonauta de BQ (Véase Ilustración 19) [86], constará de los siguientes pasos:

1. Presentación de los componentes robóticos que vamos a utilizar en la actividad: cuáles son, para qué sirven y cómo se conectan entre si para funcionar correctamente.
2. Explicación de cómo hacer el montaje virtual del Hardware dentro de la herramienta de Bitbloq (Véase Ilustración 19).
3. A continuación, descripción de la ventana de software de la herramienta, indicando para que sirve cada una de las partes de un programa y cómo se utilizan los bloques encargados de accionar el servo y el zumbador. Se realiza un pequeño ejercicio de ejemplo en el que los alumnos experimentan durante 10 minutos a mover el mini servo a distintas posiciones y a hacer sonar el zumbador [87].
4. Una vez familiarizados con cargar programas en la placa base y accionar el zumbador y el servo, se pasa a explicar cómo funciona un sensor de infrarrojos, lo que es una instrucción condicional y para qué sirve. Se plantea un pequeño ejercicio en el que

los alumnos programen al robot para que, en caso de que se active el sensor infrarrojo, suene el zumbador, como si de una alarma casera se tratase [88].

5. Montaje del “*Robonauta*”, recortando la plantilla facilitada por BQ para realizar un pequeño descanso y tener un cambio de ritmo en la clase [89].
6. Finalmente, una vez que ya se tiene montado al “*Robonauta*”, se le agregan los componentes robóticos y se acaba desarrollando el programa completo del “*Difunto Robonauta*”, que consiste en una pequeña calavera de papel que se cerrará cuando se active el sensor infrarrojo que tiene en su base, haciendo sonar también el zumbador (Véase Ilustración 20) [90].



Ilustración 19 - Bitbloq - Difunto Robonauta

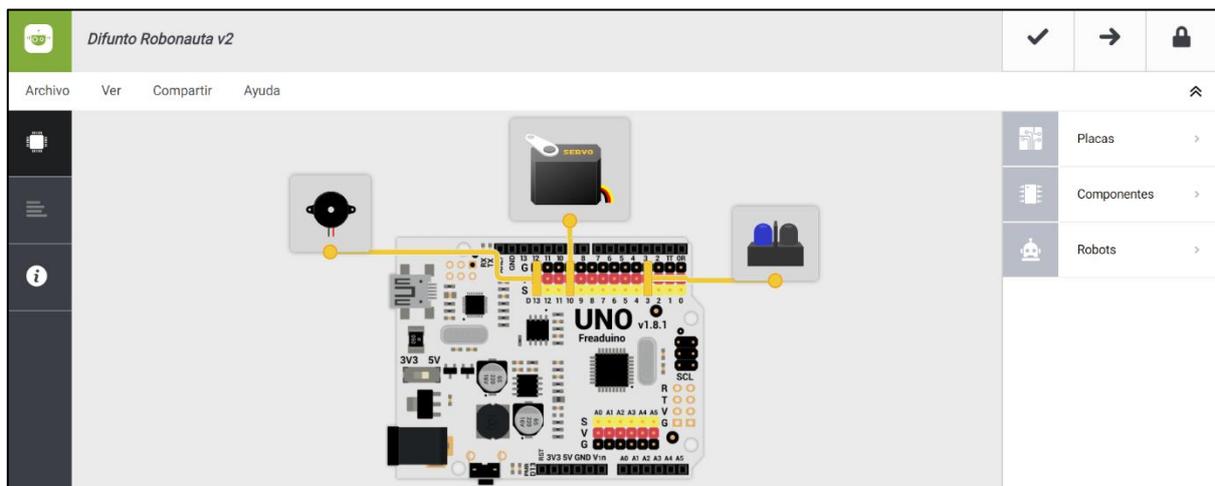


Ilustración 20 - Bitbloq - Montaje HW

Difunto Robonauta v2 ✓ → 🔒

Archivo Editar Ver Compartir Ayuda

Bloques Código

+ Variables globales, funciones y clases

- Instrucciones iniciales (Setup)

Mover servo0pin10 a 90 grados

- Bucle principal (Loop)

Si Leer sensorIR0Pin3 = Verdadero ejecutar:

Mover servo0pin10 a 0 grados

Repetir 2 veces

| | | | | | | | |
|-------------------|------------------------|----------------|-------------|-----|---------|-----|----|
| Sonar el zumbador | Variable (componentes) | zumbador0Pin13 | con la nota | 100 | durante | 25 | ms |
| Sonar el zumbador | Variable (componentes) | zumbador0Pin13 | con la nota | 180 | durante | 50 | ms |
| Sonar el zumbador | Variable (componentes) | zumbador0Pin13 | con la nota | 200 | durante | 100 | ms |

de lo contrario, ejecutar:

Mover servo0pin10 a 90 grados

Esperar 200 ms

Componentes >
 Fun Funciones >
 Var Variables >
 Cód Código >
 Mat Matemáticas >
 Tex Texto >
 Con Control >
 Lóg Lógica >
 Cla Clases >

Ilustración 21 - Bitbloq - Código problema final Difunto Robonauta

7. Evaluación Clases Tipo

En este capítulo queda resumida, y evaluada, la información obtenida de la ejecución de las actividades. Primero se detalla qué métodos de obtención de información se han seguido y el diseño de los mismos. A continuación, y para cada una de las cuatro actividades realizadas, se dispone de una sección que cuenta con: (i) las notas tomadas por el profesor, (ii) una tabla con los resultados resumidos de las entrevistas individuales a los alumnos y (iii) unas conclusiones acerca de estas dos fuentes de información. Finalmente, se exponen unas conclusiones sobre todos los resultados obtenidos.

7.1 Consideraciones sobre el sistema de evaluación

Tal y como se indica en la metodología del capítulo 4, para evaluar el desarrollo de la actividad, los progresos, y las opiniones de los alumnos, se hace uso de dos fuentes principales de información:

1. Las **notas del profesor**, en las que este anota información que considera interesante a lo largo del desarrollo de la actividad, como situaciones excepcionales o posibles incidencias, apuntes sobre cómo se ha desarrollado la sesión, completitud de las actividades por parte de los alumnos.
2. Unas breves **entrevistas a los alumnos**, que se han realizado de forma individual y una vez finalizada la actividad, a cada uno de los que ha asistido a la misma.

Se decide realizar entrevistas, frente a otras técnicas de recogida de información, como los cuestionarios, por las siguientes razones:

- Se dispone de suficiente tiempo, al finalizar las sesiones, para entrevistar a cada alumno por separado, ya que las actividades están diseñadas para menos de 10 alumnos por actividad.
- Es un método más flexible para obtener información, que permite asegurarse que el alumno comprenda las preguntas que se le están haciendo, incluso permite recordarle cuales son las opciones de las que dispone a la hora de responder (por ejemplo, ante la pregunta de “*¿cuál ha sido la actividad que menos le ha gustado?*”, enumerar todas las existentes, etc.).
- Es un tipo de encuesta más personal que el cuestionario, permitiendo recabar información extra en caso de existir, y asegurarse, en la medida de lo posible, de la veracidad de las respuestas.

Estas entrevistas están orientadas a crear un perfil de los alumnos que han asistido a la clase y evaluar su experiencia personal a lo largo de la misma. Cuentan con la siguiente forma y contenidos:

1. **El perfil del alumno:** se desean conocer las siguientes características del alumno; (i) su edad, (ii) su sexo, (iii) quien ha tomado la iniciativa a la hora de asistir a la actividad (si ha sido idea suya o de sus padres), (iv) si dispone de un ordenador en casa, smartphone o Tablet y, en caso de disponer de alguno, (v) si lo utiliza o no.
2. **La experiencia previa del alumno:** conocer si el alumno (i) había asistido a alguna actividad similar previamente, y, de haberlo hecho, (ii) si se trataba de actividades extra escolares, actividades de un día, campamentos, etc. También se quiere averiguar si el alumno (iii) conocía, o había usado, la herramienta con anterioridad.
3. **La valoración de la clase por el alumno:** finalmente se pregunta al alumno acerca de su opinión sobre la Clase Tipo en la que acaba de participar. Por un lado, se quiere conocer: (i) si le ha gustado o no la actividad, (ii) cuál es la parte de la actividad que más sencilla le ha resultado, y (iii) cuál la más difícil, además de conocer (iv) qué concepto le ha resultado más difícil. Por el otro, se desea saber: (v) si le gustaría repetir y asistir a otra clase similar y, de querer repetir, (vi) si le gustaría probar otra herramienta o trabajar con la misma que ha estado utilizando.

7.2 Resultados evaluaciones Clases Tipo

En esta sección se recogen los resultados obtenidos del proceso de ejecución de las distintas Clases Tipo impartidas.

En todas las sucesivas secciones se sigue el siguiente orden: Primero se dispone del resumen de las notas tomadas por el profesor, a continuación, se exponen en forma de tabla y de una manera resumida, los resultados de las entrevistas individuales a los alumnos realizadas al final de la actividad, y, finalmente, se exponen unas breves conclusiones acerca de estos resultados.

7.2.1 Evaluación Code, Hora del Código – Programa con Ana y Elsa

I - Notas del profesor.

Asistieron los seis alumnos que estaban apuntados a la actividad, además esta se desarrolló con normalidad, pudiendo seguir el programa establecido y realizando al completo la actividad de “*Hora del Código – Programa con Ana y Elsa*”.

Finalmente, indicar que todos los alumnos completaron todos los problemas planteados (los 20 niveles de la actividad) en el tiempo propuesto, no habiendo grandes diferencias entre los tiempos de finalización de los alumnos.

II - Resultados de las entrevistas a los alumnos.

Tabla 12 - Entrevistas Alumnos - Clase Code, Hora del Código – Programa con Ana y Elsa

| Alumnos - Code - Hora del Código - Programa con Ana y Elsa | | | | | | |
|--|--------------------------------|----------------------------------|--------------------------------|----------------------------------|----------------------------------|----------------------------------|
| | Alumno 1 | Alumno 2 | Alumno 3 | Alumno 4 | Alumno 5 | Alumno 6 |
| Perfil Alumno | | | | | | |
| Edad | 7 | 6 | 8 | 8 | 8 | 7 |
| Sexo (Hombre/Mujer) | Hombre. | Hombre. | Mujer. | Mujer. | Hombre. | Hombre. |
| ¿Te has apuntado por iniciativa propia o te han apuntado tus Padres? | Iniciativa propia. | Iniciativa de los padres. | Iniciativa de los padres. | Iniciativa propia. | Iniciativa de los padres. | Iniciativa de los padres. |
| ¿Tienes ordenador en casa? | Si. | Si. | Si. | Si. | No. | Si |
| ¿Utilizas algún Ordenador, Tablet o Smartphone? | Si | Si | Si | Si | Si (Smartphone) | Si |
| Experiencia Previa | | | | | | |
| ¿Habiais estado en alguna clase de este tipo alguna vez? | Si . | No. | No. | Si. | No. | Si. |
| ¿En cuáles? | Actividad extra-escolar. | | | Actividad extra-escolar. | | Actividad extra-escolar. |
| ¿Conocías esta herramienta? | No. | | | No. | | Si. |
| Valoración de la Clase | | | | | | |
| ¿Qué te ha parecido? | Me ha gustado. | Me ha gustado. | Me ha gustado mucho. | Me ha gustado. | Me ha gustado mucho. | Me ha gustado. |
| ¿Qué actividad/juego es la que más te ha gustado? | Dibujo Libre (lv120). | Dibujar Copo de Nieve (lv119). | Dibujo Libre (lv120). | Dibujar Aspas (lv19). | Dibujar Aspas (lv19). | Dibujo Libre (lv120). |
| ¿Qué actividad/juego es la que menos te ha gustado? | Dibujar Paralelogramo (lv111). | Dibujar muchos Círculos (lv115). | Dibujar Paralelogramo (lv111). | Dibujar muchos Círculos (lv115). | Dibujar muchos Círculos (lv115). | Dibujar muchos Círculos (lv115). |
| ¿Qué es lo que te ha parecido más difícil? | Los grados que hay que girar. | Los grados que hay que girar. | Los grados que hay que girar. | Los grados que hay que girar. | Los grados que hay que girar. | Los grados que hay que girar. |
| ¿Te gustaría repetir una clase de este tipo? | Si. | Si. | Si. | Si. | Si. | Si. |
| ¿Y te gustaría repetir usando esta herramienta o probar otra? | Otra. | Esta, pero con Minecraft. | Otra. | Esta, pero con Angry Birds. | Esta, pero con Minecraft. | Esta, pero con Minecraft. |

III – Conclusiones.

Tal y como se recoge en la Tabla 12, se trata de un grupo de niños entre 6 y 8 años, con mayoría de niños frente a niñas, en el que prácticamente todos ellos disponían de ordenador en casa y que estaban familiarizados con la tecnología.

La mayoría asistieron a la actividad por iniciativa de sus padres, y solo la mitad de ellos habían participado en una actividad de este tipo previamente.

Se observa que la sesión “gustó” o “gustó mucho” a los alumnos, y que a todos ellos les gustaría repetir, aunque la mayoría preferirían hacer una nueva actividad con otros “protagonistas”, lo que es debido a los gustos personales de cada uno de los niños.

Además, resulta remarcable que, de toda la actividad, que incluía enseñar el concepto de un bucle que itera varias veces y el aprender a manejar una interfaz de programación por bloques (en muchos casos por primera vez), lo que más complicado les resultó a todos los alumnos, fue el trabajar con grados y orientar correctamente la figura de “Ana” o “Elsa”. Situación que se encuentra prácticamente desde el inicio de la actividad (nivel 2) hasta su final (nivel 19), y que es el problema principal a resolver en esta actividad de la hora del código.

Finalmente, el hecho de que todos los alumnos terminasen todos los problemas, y todos ellos en tiempos similares, indica que la actividad de “*Hora del Código – Programa con Ana y Elsa*” se ajusta perfectamente a los tiempos estimados.

7.2.2 Evaluación Code, Hora del Código – Escribe tu primer programa

I - Notas del profesor.

A esta sesión asistieron cinco de los seis alumnos que se habían apuntado. Este hecho en la práctica no supuso ningún problema para el desarrollo normal de la clase, ya que los alumnos trabajan de forma individual, cada uno con su ordenador, y no había programada ninguna actividad de grupo en la que se necesitase la asistencia de todo el grupo.

Se realizó todo el programa propuesto, completando la actividad de “*Hora del Código - Escribe tu primer programa*”, dentro de los límites establecidos de tiempo, y de igual forma que en la actividad anterior, todos los alumnos terminaron la actividad más o menos a la par.

Cabe indicar como pequeña incidencia que al principio de la clase se experimentaron problemas de conexión a internet en algunos equipos, tardando mucho en cargar la herramienta. Este problema se corrigió rápidamente, cambiando la red Wifi de los ordenadores que no cargaban satisfactoriamente la actividad, a otra red disponible y con menor tráfico.

Finalmente, y como ocurrió con la otra actividad de Code, destaca el hecho de que todos los alumnos terminasen todos los problemas dentro de los límites establecidos y, además, en tiempos similares, lo que refuerza la idea de que las actividades de “Hora del Código” propuestas por Code cumplen con lo especificado en sus programas educativos y están correctamente adaptadas al trabajo con niños de esas edades.

7.2.3 Evaluación LearnToMod, Introduction To Loops

I - Notas del profesor.

Se realizó todo el programa establecido, completando los 3 pasos, con sus correspondientes 4 problemas, de la actividad

En este caso sí que se detectó disparidades en el progreso de los alumnos, observando dos casos en los que se apreciaba claramente como tenían mayores dificultades que el resto a la hora de manejarse con la herramienta, según quedan reflejados en la tabla 14, se trataban de los alumnos 5 y 6. Por lo demás, la clase transcurrió con normalidad dentro de los tiempos y progresos esperados.

II - Resultados de las entrevistas a los alumnos.

Tabla 14 - Entrevistas Alumnos - Clase LearnToMod -Introduccion To Loops

| Alumnos - LearnToMod - Introduction To Loops | | | | | | | |
|--|---------------------------------|---------------------------------|----------------------|-------------------------|---------------------------------|---------------------------------|------------------------|
| | Alumno 1 | Alumno 2 | Alumno 3 | Alumno 4 | Alumno 5 | Alumno 6 | Alumno 7 |
| Perfil Alumno | | | | | | | |
| Edad | 7 | 8 | 10 | 9 | 8 | 8 | 7 |
| Sexo (Hombre/Mujer) | Hombre | Mujer | Mujer | Hombre | Hombre | Hombre | Hombre |
| ¿Te has apuntado por iniciativa propia o te han apuntado tus Padres? | Padres | Iniciativa propia. | Iniciativa propia. | Padres | Padres | Padres | Padres |
| ¿Tienes ordenador en casa? | Si. | No. | No. | Si. | Si. | Si. | Si. |
| ¿Utilizas algún Ordenador, Tablet o Smartphone? | Si. | No. | No. | Si (smartphone). | No. | Si. | Si. |
| Experiencia Previa | | | | | | | |
| ¿Habíais estado en alguna clase de este tipo alguna vez? | No. | No. | Si | Si | No. | No. | Si. |
| ¿En cuáles? | | | Academia. | Actividad extra-escolar | | | Academia. |
| ¿Conocías esta herramienta? | No. | No. | Si | No. | No. | No. | Si. |
| Valoración de la Clase | | | | | | | |
| ¿Qué te ha parecido? | Me ha gustado. | Me ha gustado poco. | Me ha gustado mucho. | Me ha gustado. | Me ha gustado poco. | No me ha gustado. | Me ha gustado mucho. |
| ¿Qué actividad/juego es la que más te ha gustado? | El simulador. | Probarlo en Minecraft. | El simulador. | Probarlo en Minecraft. | Probarlo en Minecraft. | El simulador. | Probarlo en Minecraft. |
| ¿Qué actividad/juego es la que menos te ha gustado? | Las funciones. | Las funciones. | Las funciones. | Las funciones. | Las funciones. | Las funciones. | Las funciones. |
| ¿Qué es lo que te ha parecido más difícil? | Las funciones. | Las funciones. | Las funciones. | Las funciones. | Las funciones. | Las funciones. | Las funciones. |
| ¿Te gustaría repetir una clase de este tipo? | Si | No. | Si | Si | No | No. | Si |
| ¿Y te gustaría repetir usando esta herramienta o probar otra? | Otra | | Esta. | Otra | | | Esta. |

III - Conclusiones.

Tal y como se recoge en la Tabla 14, en esta actividad se trabajó con un grupo mayor que en las dos actividades anteriores, teniendo un total de 7 niños con edades comprendidas entre los 7 y los 10 años.

La gran mayoría de los alumnos asistieron por iniciativa de los padres, y la mayoría de todos ellos no habían asistido previamente a ninguna actividad de este tipo. Destacan los casos de los alumnos 5 y 6 que no se encontraban nada motivados. Este hecho repercutió negativamente en sus avances y, en menor medida, en los de sus compañeros, ya que el profesor tenía que asistirlos cada poco tiempo para evitar que se descolgaran completamente de la actividad.

Esta actividad ha sido la que menos ha gustado, teniendo dos alumnos que dijeron que les había “gustado poco” e incluso un alumno que afirmó que “no le había gustado”. Además, estos tres alumnos indicaron que no les gustaría repetir una actividad de este tipo.

La parte de la actividad que fue considerada más complicada, y que además menos gusto, fue la fase 3, en la que se describirán los bloques y conceptos relacionados con las funciones. Por otro lado, lo que mayor éxito tuvo entre los alumnos fue probar el programa final dentro del videojuego de Minecraft.

Finalmente, y como ocurrió en las dos actividades anteriores de Code, todos los alumnos terminaron todos los ejercicios y fases propuestos por la actividad, aunque esto solo se consiguió porque el profesor estuvo ayudando de forma individual a los dos alumnos, 5 y 6, más rezagados.

7.2.4 Evaluación Bitbloq, Difunto Robonauta

I - Notas del profesor.

En esta clase fallaron dos de los alumnos que estaba apuntados previamente, este hecho no tuvo mayor repercusión que el haber preparado, y creado, más material del necesario para la Clase Tipo (kits de robótica y equipos sin usar, y fotocopias sobrantes del recortable del “*Difunto Robonauta*”).

Se completo la actividad en su totalidad y en los tiempos previstos, y se pudo observar como a lo largo de toda la sesión los alumnos fueron progresando a la par, sin que quedase ninguno rezagado o muy por delante de los demás.

En esta ocasión destaca un especial interés por parte de los alumnos debido al poder interactuar, aparte de con el ordenador, con elementos del mundo real que controlaban desde el ordenador, es decir, las piezas del robot.

Además, el realizar una pequeña pausa cuando se llevaban transcurridos unos 45 minutos de la clase, para hacer el montaje del “*Robonauta*”, sirvió para que los niños descansasen y estuviesen más atentos en la sexta y última fase de esta actividad, que es a su vez la más complicada.

II - Resultados de las entrevistas a los alumnos.

Tabla 15 - Entrevistas a los Alumnos – Clase Bitbloq Difunto Robonauta

| Alumnos - Bitbloq - Difunto Robonauta | | | | | | |
|--|---------------------------|------------------------------|------------------------------|---------------------------------|---------------------------|---------------------------------|
| | Alumno 1 | Alumno 2 | Alumno 3 | Alumno 4 | Alumno 5 | Alumno 6 |
| Perfil Alumno | | | | | | |
| Edad | 9 | 9 | 10 | 7 | 10 | 10 |
| Sexo (Hombre/Mujer) | Mujer. | Mujer. | Hombre. | Hombre. | Hombre. | Hombre. |
| ¿Te has apuntado por iniciativa propia o te han apuntado tus Padres? | Iniciativa propia. | Iniciativa propia. | Iniciativa propia. | Padres | Iniciativa propia. | Padres |
| ¿Tienes ordenador en casa? | Si. | Si. | Si. | No. | Si. | Si. |
| ¿Utilizas algún Ordenador, Tablet o Smartphone? | Si (Tablet). | No. | Si. | No. | Si. | Si. |
| Experiencia Previa | | | | | | |
| ¿Habiais estado en alguna clase de este tipo alguna vez? | Si | Si | Si | No. | Si | No. |
| ¿En cuáles? | Academia. | Academia. | Academia. | | Academia. | |
| ¿Conocías esta herramienta? | Si | Si | Si | No. | Si | No. |
| Valoración de la Clase | | | | | | |
| ¿Qué te ha parecido? | Me ha gustado. | Me ha gustado mucho. | Me ha gustado mucho. | Me ha gustado. | Me ha gustado. | Me ha gustado. |
| ¿Qué actividad/juego es la que más te ha gustado? | Conectar los componentes. | Programarlo para que muerda. | Programarlo para que muerda. | Programarlo para que muerda. | Programarlo normal. | Conectar los componentes. |
| ¿Qué actividad/juego es la que menos te ha gustado? | Programarlo normal. | Conectar los componentes. | Conectar los componentes. | Programarlo normal. | Conectar los componentes. | Programarlo normal. |
| ¿Qué es lo que te ha parecido más difícil? | Cargar el programa. | Conectar los componentes. | Conectar los componentes. | Cargar el programa. | Cargar el programa. | Cargar el programa. |
| ¿Te gustaría repetir una clase de este tipo? | Si | Si. | Si | Si | Si. | No. |
| ¿Y te gustaría repetir usando esta herramienta o probar otra? | Esta. | Esta. | Esta. | Otra | Otra | Otra |

III - Conclusiones.

Tal y como se observa en la Tabla 15, en esta actividad se contó con un grupo de 6 alumnos con edades comprendidas entre los 7 y 10 años.

No se apreciaron grandes diferencias en el ritmo de trabajo, de hecho, todos ellos terminaron todas las actividades propuestas y dentro de los límites de tiempo establecidos.

En este caso la mayoría asistió a la actividad por iniciativa propia, y se trataba de alumnos que ya habían trabajado con Bitbloq u otras herramientas similares previamente. Esto se notó en el desarrollo de la clase, ya que se avanzó a un ritmo rápido, y se pudo observar como mostraron un gran interés a lo largo de toda la actividad.

Esta es la actividad mejor valorada, teniendo dos alumnos a los que les había “gustado mucho” y al resto que afirmó que les había “gustado”, y queriendo todos ellos repetir y participar en otra actividad similar, la mitad de ellos usando esta misma herramienta y la otra mitad cambiando.

7.3 Conclusiones

Reuniendo todas las evaluaciones de cada una de las actividades expuestas en este capítulo, se llegan a las siguientes conclusiones finales:

Sobre el tipo de docente que debe impartir las clases, se concluye que resultaría ideal que contase con algún tipo de formación relacionada con la informática, no necesariamente a nivel universitario. Sí que es necesario que tenga formación como profesor y sepa trabajar con niños.

Las actividades orientas a enseñar informática a niños son sencillas, de modo que cualquier adulto del ámbito de la enseñanza, disponga de formación previa o no, debería ser capaz de entenderlas y explicarlas a niños pequeños, siempre y cuando hiciese el esfuerzo de realizar y comprender las actividades previamente.

Sobre el alumnado a los que deben de ir dirigidas estas actividades, se concluye que:

- Deben de estar motivados, se puede enseñar informática a niños siempre y cuando estos tengan intención de aprender y quieran participar en la actividad.
- El que tengan acceso a un ordenador o no en su casa no es un factor determinante para asistir a una actividad de este tipo. Sí que ayuda que los alumnos tengan cierto manejo y soltura en el uso básico de un ordenador, pero no es un requisito indispensable.

Sobre el contexto y desarrollo de las actividades y como deben diseñarse, se concluye que:

- Deben estar orientadas para trabajar con niños de 6 años en adelante. Aunque existen herramientas adaptadas para edades muy tempranas, se considera ideal empezar con 6 años, ya que para entonces los niños tienen suficientes capacidades psicomotrices como para manejar correctamente el ratón y teclado, comprender los conceptos básicos de programación, y, además, en muchos casos, ya han interactuado con ordenadores en alguna ocasión.
- Deben tener una duración aproximada de una hora o hora y media. Con sesiones más cortas no da tiempo a que los alumnos se familiaricen con la herramienta que están usando y a la vez aprendan nuevos conceptos de programación o informática.

Además, con una sesión por semana debería de ser suficiente, ya que estas actividades están pensadas para enseñar informática de una forma gradual, y al tratarse de enseñar conceptos de cierta dificultad, es necesario repasarlos y asentarlos antes de pasar al siguiente.

- Se debe trabajar con grupos reducidos de alumnos, lo que ayuda enormemente al desarrollo y progreso adecuado de cada uno de los participantes.

Si el profesor puede atender de forma individual a aquellos niños que tengan más problemas realizando la actividad, se consigue que la clase avance a un mismo ritmo, no dejando atrás a los alumnos que necesiten de una mayor atención, ya sea porque no comprendan los conceptos que se están enseñando o por su idiosincrasia.

Se ha podido constatar en la evaluación de las clases que aquellos alumnos que más les cuesta realizar la actividad son los menos motivados, y viceversa. El poder resolver las dudas y ayudar a los niños que tengan mayores dificultades resulta clave para que estos se interesen por la actividad y consigan ponerse al nivel de sus compañeros.

- Tener un enfoque tanto lúdico como educacional. Si uno se centra solo en uno de estos dos enfoques, se puede llegar a dos escenarios desfavorables distintos:
 - Si uno se centra solo en el aspecto lúdico, la actividad se acaba convirtiendo en un tiempo de juego, quedando la función educativa completamente relegada.
 - Por el contrario, de centrarse únicamente en un enfoque educacional, se corre el riesgo que la actividad se convierta en “una clase más” y el alumno pierda motivación. No hay que olvidar en ningún momento que se trata de niños, y hay que plantearles las actividades para que les resulten atractivas y estimulantes.
- Incluir cambios de ritmo y breves descansos cuando sean necesarios. Resulta recomendable dividir la clase en dos o más partes distintas, de esta forma se mantiene la atención del alumno.

Por ejemplo, si se trabaja con herramientas relacionadas con la robótica, una parte de la clase se debería dedicar al montaje y explicación del hardware, mientras que la otra se debería dedicar a la programación del software.

8. Conclusiones

Este capítulo final del presente documento contiene: conclusiones finales acerca de las actividades destinadas a enseñar informática a niños pequeños, una valoración personal del trabajo realizado, y una última sección en la que se plantean posibles ideas y alternativas de trabajo futuro que se pueden realizar sobre este trabajo de fin de grado.

8.1 Conclusiones

A lo largo de este trabajo se ha tenido ocasión de conocer, analizar, y trabajar, con multitud de herramientas educativas distintas, cada una de ellas con características diferentes, y sus propios puntos fuertes y débiles. De este conocimiento y análisis se ha elaborado la siguiente **lista de características que se consideran ideales para una herramienta destinada a enseñar conceptos de informática a niños**. Por tanto, se consideraría ideal aquella herramienta que:

- Tuviera tanto una versión web como de escritorio. En ocasiones no se dispone de una buena conexión a Internet, y, si no se carga la herramienta, es imposible llevar a cabo la actividad. Además, puede darse la situación de que no sea posible instalar nuevo software en los ordenadores que se van a utilizar, por lo que contar con una versión web de la herramienta resultaría también necesario.
- Contase con distintos grados de dificultad, y distribuidos progresivamente de más fácil a más difícil, introduciendo nuevas instrucciones o conceptos de programación o informática poco a poco.

Aunque el poder trabajar desde el principio con todas las posibilidades que ofrece la herramienta (todos los bloques/instrucciones), resulta adecuado para personas que conozcan la herramienta o que estén familiarizados con la informática, para niños pequeños que nunca hayan trabajado con ellas, puede resultar confuso. Es mejor que se introduzcan y habiliten poco a poco.

- Trabajara con elementos de software y hardware. En el desarrollo y evaluación de las clases se observó que aquellas actividades que trabajaban tanto con software como con hardware resultaban más entretenidas para los alumnos. El observar los progresos del trabajo, no solo en la pantalla del ordenador, sino también en “la realidad”, motiva especialmente a los niños.
- Utilizase elementos de programación visual o por bloques enlazándolos con algún juego o actividad que resultase llamativo para un niño. Por ejemplo, Code trabaja con personajes de distintos mundos (“Frozen”, “Ice Age”, “Flappy Bird”, etc.), y LearnToMod trabaja con el videojuego Minecraft.

Adicionalmente, se ha tenido la oportunidad de realizar varias actividades diseñadas por distintas empresas, incluso se han diseñado actividades desde cero empleando un determinado subconjunto de herramientas. De la experiencia del trabajo realizado con ellas se ha elaborado la siguiente **lista de características que se consideran ideales para una actividad destinada a enseñar conceptos de informática a niños**. Por tanto, se consideraría ideal aquella actividad que:

- La impartiese al menos un profesor que supiese trabajar y enseñar a niños pequeños, y tuviese conocimientos de informática.
- Tuviera un programa educativo adaptado a la edad de los alumnos que van a participar en la actividad, teniendo estos como mínimo 6 años.
- Tuviese una duración estimada entre hora y hora y media, y teniendo una sesión por semana.
- Estableciese un número límite de alumnos por clase en función del número de profesores disponibles, 8 alumnos por cada profesor.
- Mantuviese siempre un enfoque tanto lúdico como educativo, teniendo en cuenta tanto las capacidades de atención de los niños participantes como la necesidad de avanzar y asentar los conceptos que se desean enseñar.
- Incluir varios cambios de ritmo y trabajo a lo largo de la clase, diversificando las actividades y problemas que han de resolver los alumnos con ánimo de maximizar su atención y entretenimiento.
- Describiese claramente que conceptos de programación, informática, o nuevas tecnologías, se deben de enseñar en cada sesión, y adaptándose a las capacidades de los alumnos.
- Incluyese problemas pertenecientes a otras disciplinas, como las matemáticas y las ciencias, que ya conociese el alumno, provocando que en esta ocasión tuviera que solucionarlos utilizando la programación.
- Que enseñe metodologías de trabajo y resolución de problemas como el pensamiento computacional, que ayuda a estructurar el pensamiento y procesos lógicos de los niños para que les resulte más sencillo resolver problemas complejos de naturaleza científica.

8.2 Valoración personal

Siempre me ha llamado la atención la docencia y los niños, teniendo desde hace dos años, la oportunidad de trabajar como profesor impartiendo cursos de robótica y programación de videojuegos a niños desde los 7 hasta los 12 años.

Desde el mismo momento que empecé, me quedé fascinado por la ingente cantidad de herramientas y actividades existentes, y de todas las posibilidades que esto tenía. No paraba de pensar que, de ser yo un niño ahora, me encantaría a asistir a clases de Robótica o de programación de videojuegos, no porque crea que me habría ayudado cuando entre en la universidad, sino porque realmente me parecen actividades que son realmente divertidas a la par que educativas.

Sin embargo, esta gran cantidad de herramientas y posibilidades también puede ser un problema, me di cuenta de que, a la hora de preparar nuevas clases, talleres, y cursos, existían demasiadas opciones, y no había un canon, unas reglas establecidas a la hora de escoger qué tipo de herramienta o actividad.

Es por este motivo que me decidí realizar este trabajo, con ánimo de encontrar aquellas características que ha de tener una buena herramienta y actividad.

Si bien se podrían haber incluido los diseños y los resultados de más clases que he realizado y evaluado, me di cuenta de que no iba a variar el resultado obtenido y mostrado en las conclusiones del capítulo 7 y a lo largo de la sección anterior (8.1). De modo que estoy satisfecho con mi trabajo.

8.3 Trabajo futuro

Una vez finalizado el trabajo de fin de grado, se presentan algunas posibles mejoras y desarrollos como trabajo futuro, en concreto se propone afrontar alguna de las tres cuestiones siguientes:

1. Ampliar el rango de estudio de este TFG, diseñando, realizando, y evaluando, más actividades y más herramientas distintas con ánimo de descubrir nuevas características que se deberían tener una actividad y herramienta ideales.
2. Desarrollar una herramienta que tuviera las características nombradas en la sección 8.1 de este capítulo, que son las que se consideran ideales.
3. Elaborar un plan de estudios que abarcara un curso entero de actividades para niños de primaria ateniéndose a las pautas establecidas en la sección 8.1 sobre que características tienen una actividad ideal.

Bibliografía:

- [1] Mark Sanders, "STEM, STEM Education, STEMmania", <https://vtechworks.lib.vt.edu/bitstream/handle/10919/51616/STEMmania.pdf?sequence=1> , última visita: 12-06-17.
- [2] David A. Koonce, Ohio University, "WHAT IS STEM?", <http://www.asee.org/public/conferences/1/papers/289/download> , última visita: 12-06-17.
- [3] Richard M. Felder y Rebecca Brent, "¿CÓMO SE PRESENTARÁ LA EDUCACIÓN EN CIENCIA, TECNOLOGÍA, INGENIERÍA Y MATEMÁTICAS (STEM) EN CINCO AÑOS O ANTES?", http://cdn02.pucp.education/academico/2016/05/02132518/conferencia_aprendizaje_eval_stem020516.pdf , última visita: 12-06-17.
- [4] Seymour Papert, "Teaching children to be mathematicians vs teaching about mathematics", <ftp://publications.ai.mit.edu/ai-publications/pdf/AIM-249.pdf> , última visita: 12-06-17.
- [5] Origen del pensamiento computacional, origen: revista "Bits de Ciencia, primer trimestre 2015.
- [6] Jeannette M. Wing, "Computational Thinking", <http://www.cs.cmu.edu/afs/cs/usr/wing/www/publications/Wing06.pdf> , última visita: 12-06-17.
- [7] "Introduction to computational thinking", <http://www.bbc.co.uk/education/guides/zp92mp3/revision/1> , última visita: 12-06-17.
- [8] Jesús Moreno León, "¿Qué es el pensamiento computacional?", <http://programamos.es/que-es-el-pensamiento-computacional/> , última visita: 12-06-17.
- [9] Jesús Valverde Berrocoso, María Rosa Fernández Sánchez y María del Carmen Garrido Arroyo, "El pensamiento computacional y las nuevas ecologías del aprendizaje", http://www.um.es/ead/red/46/valverde_et_al.pdf , última visita: 12-06-17.
- [10] "Computing in the national curriculum A guide for secondary teachers", <http://community.computingatschool.org.uk/files/3383/original.pdf> , última visita: 12-06-17.
- [11] Donna M. Ogle, "K-W-L: A Teaching Model That Develops Active Reading of Expository Text", http://www.jstor.org/stable/20199156?seq=1#page_scan_tab_contents , última visita: 12-06-17.
- [12] "List of educational programming languages", https://en.wikipedia.org/wiki/List_of_educational_programming_languages , última visita: 12-06-17.
- [13] Jenn Choi, "The complete guide to buying a robot for your kids", <https://qz.com/309365/the-complete-guide-to-buying-a-robot-for-your-kids/> , última visita: 12-06-17.
- [14] Science Buddies, "Kid-Friendly Programming Languages", http://www.sciencebuddies.org/science-fair-projects/project_ideas/CompSci_Kid_Programming.shtml , última visita: 18-06-17.
- [15] Science Buddies, "Resources for STEM Education Through Video Game and Animation Creation", [http://www.sciencebuddies.org/science-fair-](http://www.sciencebuddies.org/science-fair-projects/project_ideas/CompSci_Kid_Programming.shtml)

- [projects/project_ideas/CompSci_Teacher_VideoGame_Resources.shtml](#) , última visita: 18-06-17.
- [16] Beate Jost, Markus Ketterl, Reinhard Budde, "Graphical Programming Environments for Educational Robots: Open Roberta - Yet Another One?", <http://ieeexplore.ieee.org/document/7033055/> , última visita: 12-06-17.
- [17] Rémi Dehouck , "The maturity of visual programming", <http://www.craft.ai/blog/the-maturity-of-visual-programming/> , última visita: 12-06-17.
- [18] Goran Zaharija, Sasa Mladeovic, Ivica Boljat, "Introducing basic programming concepts to elementary school children", <https://bib.irb.hr/datoteka/663249.1-s2.0-S1877042813048015-main.pdf> , última visita: 18-06-17.
- [19] Richard Holowczak, "PROGRAMMING CONCEPTS: A BRIEF TUTORIAL FOR NEW PROGRAMMERS", <http://holowczak.com/programming-concepts-tutorial-programmers/> , última visita: 18-06-17.
- [20] BRIGHAM YOUNG UNIVERSITY, "DigHT 210 Basic Programming Concepts", <http://revolution.byu.edu/programmingconcepts/controlstruct.php> , última visita: 18-06-17.
- [21] BBC Bitesize, "Boolean logic", <http://www.bbc.co.uk/education/guides/zqp9kqt/revision>
- [22] Google for Education Blockly, <https://developers.google.com/blockly/> , última visita: 12-06-17.
- [23] Cade Metz, Wired, "GOOGLE BLOCKLY LETS YOU HACK WITH NO KEYBOARD", <https://www.wired.com/2012/06/google-blockly/> , última visita: 12-06-17.
- [24] Barbara E. Hernandez, NBC, "Google's Blockly Teaches You to Create App", <http://www.nbcbayarea.com/blogs/press-here/Googles-Blockly-Teaches-You-to-Create-Apps-158935465.html> , última visita: 18-06-17.
- [25] Code, <https://code.org/> , última visita: 12-06-17.
- [26] Code, "Courses from Code Studio", <https://studio.code.org/courses> , última visita: 18-06-17.
- [27] Code, "CS Fundamentals Unplugged", <https://code.org/curriculum/unplugged> , última visita: 18-06-17.
- [28] Code, "Descargas de una Hora de Código - Hora de código de Minecraft", <https://studio.code.org/download/mc> , última visita: 18-06-17.
- [29] Code, "Make a Flappy Game: An Hour of Code Tutorial", <https://code.org/hourofcode/flappy> , última visita: 12-06-17.
- [30] Code, "Make a Flappy Game: An Hour of Code Tutorial", nivel 9/10, <https://studio.code.org/flappy/9> , última visita: 12-06-17.
- [31] LearnToMod, <https://www.learntomod.com/> , última visita: 12-06-17.
- [32] LearnTomod, "What is Modding? What is LearnToMod?", <https://www.learntomod.com/minecraft-for-parents> , última visita: 12-06-17.
- [33] Scratch, <https://scratch.mit.edu/> , última visita: 12-06-17.
- [34] Scratch, "Scratch 2 Offline Editor Download", <https://scratch.mit.edu/scratch2download/> , última visita: 12-06-17.

- [35] MIT, Lifelong Kidergarten group, <https://ilk.media.mit.edu/> , última visita: 12-06-17.
- [36] Communications of the ACM, Noviembre 2009,"Scratch: Programming for All", <http://web.media.mit.edu/~mres/papers/Scratch-CACM-final.pdf> , última visita: 12-06-17.
- [37] Scratch, "Scratch Blocks", <https://scratch.mit.edu/developers> , última visita: 12-06-17.
- [38] Scratch, "Acerca de Scratch", <https://scratch.mit.edu/about/> , última visita: 12-06-17.
- [39] Minecraft Education Edition, <https://education.minecraft.net/> , última visita: 12-06-17.
- [40] Minecraft Education Edition, "Download", <https://education.minecraft.net/get-started/download/> , última visita: 12-06-17.
- [41] Marsh Davies, "NEW INTELLIGENCE: EDUCATION EDITION ARRIVES NOV 1!", <http://mojang.com/2016/09/new-intelligence-education-edition-arrives-november-1/> , última visita: 12-06-17.
- [42] Minecraft Education Edition, "RESOURCES TO HELP YOU GET STARTED", <https://education.minecraft.net/class-resources/starter-kit/> , última visita: 12-06-17.
- [43] Emma Boyle, "A guide to Minecraft: Education Edition", <http://www.techradar.com/news/a-guide-to-minecraft-education-edition> , última visita: 12-06-17.
- [44] Nick Morphus, "How You Can Use Minecraft Education Edition in Your School", <http://blog.capterra.com/how-you-can-use-minecraft-education-edition-in-your-school/> , última visita: 12-06-17.
- [45] Minecraft Education Edition, "LESSON PLANS", <https://education.minecraft.net/class-resources/lessons/> , última visita: 12-06-17.
- [46] Minecraft Education Edition, "LEARN TO PLAY MINECRAFT | STAGE ONE", <https://education.minecraft.net/trainings/learn-to-play-minecraft-stage-one/> , última visita: 12-06-17.
- [47] Minecraft Education Edition, "Code Builder for Minecraft: Education Edition Now Available", <https://education.minecraft.net/code-builder-for-minecraft-education-edition-now-available/> , última visita: 12-06-17.
- [48] Minecraft Education Edition, "CODE BUILDER FOR MINECRAFT: EDUCATION EDITION", <https://education.minecraft.net/trainings/code-builder-for-minecraft-education-edition/> , última visita: 12-06-17.
- [49] LearnToMod, Pricing, <https://www.learntomod.com/minecraft-modding-features/#pricing> , última visita: 12-06-17.
- [50] Minecraft Education Edition, "How much does it cost?", <https://education.minecraft.net/faq/how-much-does-it-cost/> , última visita: 12-06-17.
- [51] BitBloq, <http://bitbloq.bq.com/> , última visita: 12-06-17.
- [52] BitBloq, "About us", <http://bitbloq.bq.com/#/aboutus> , última visita: 12-06-17.
- [53] BQ, "Robot Zowi", <https://www.bq.com/es/zowi> , última visita: 12-06-17.
- [54] Comunicae, "Ya se pueden programar los kits de Makeblock con Bitbloq", <https://www.comunicae.es/nota/ya-se-pueden-programar-los-kits-de-makeblock-1183980/> , última visita: 12-06-17.

- [55] Bitbloq, "Explore", <http://bitbloq.bq.com/#/explore> , última visita: 12-06-17.
- [56] BitBloq, "Learn", <http://bitbloq.bq.com/#/learn> , última visita: 12-06-17.
- [57] LEGO MindStorms, <https://www.lego.com/es-es/mindstorms> , última visita: 12-06-17.
- [58] Lifelong Kindergarten group, "Programmable Bricks", <https://web.archive.org/web/20131021090205/http://ilk.media.mit.edu/projects.php?id=135> , última visita: 12-06-17.
- [59] MIT, "The MIT Programmable Brick", <https://web.archive.org/web/20051025132815/http://lcs.www.media.mit.edu/groups/el/projects/programmable-brick/> , última visita: 12-06-17.
- [60] LEGO MindStorms, "31313 MINDSTORMS EV3", <https://www.lego.com/es-es/mindstorms/products/mindstorms-ev3-31313> , última visita: 12-06-17.
- [61] LEGO MindStorms, "Descarga SOFTWARE DE PROGRAMACIÓN EV3 (PC/MAC)", <https://www.lego.com/es-es/mindstorms/downloads> , última visita: 12-06-17.
- [62] LEGO, "Welcome to the MINDSTORMS Community!", <https://wwwsecure.us.lego.com/en-us/mindstorms/community/browse?query=&order=date&category=> , última visita: 12-06-17.
- [63] LEGO Shop, "Edades" <https://shop.lego.com/es-ES/Poredad> , última visita: 12-06-17.
- [64] Tracy Gardner, "LEGO Mindstorms EV3 Age Recommendations - Consider Starting Under 10 for Techie Kids", <http://www.techagekids.com/2015/09/lego-mindstorms-ev3-age-recommendations.html> , última visita: 12-06-17.
- [65] LEGO MindStorms, "Construye el robot EV3RSTORM", <https://www.lego.com/es-es/mindstorms/build-a-robot/ev3rstorm> , última visita: 18-06-17.
- [66] PComponentes, "BQ Mi primer Kit de Robótica", <https://www.pcomponentes.com/bq-mi-primer-kit-de-robotica> , última visita: 18-06-17.
- [67] LEGO Shop, "LEGO® MINDSTORMS® EV3", <https://shop.lego.com/es-ES/LEGO-MINDSTORMS-EV3-31313> , última visita: 18-06-17.
- [68] Robot Turtles, <http://www.robotturtles.com/> , última visita: 12-06-17.
- [69] KickStarter, "Robot Turtles: The Board Game for Little Programmers", <https://www.kickstarter.com/projects/danshapiro/robot-turtles-the-board-game-for-little-programmer> , última visita: 12-06-17.
- [70] Robot Turtles, "Instructions", <http://www.robotturtles.com/instructions/> , última visita: 12-06-17.
- [71] Code Monkey Island, <http://codemonkeyplanet.com/> , última visita: 12-06-17.
- [72] KickStarter, "Code Monkey Island - making programming child's play", <https://www.kickstarter.com/projects/rajsidhu/code-monkey-island-making-programming-childs-play> , última visita: 12-06-17.
- [73] Code Monkey Island, "Explorer's guide", <http://codemonkeyplanet.com/explorers-guide.pdf> , última visita: 12-06-17.
- [74] Youtube, recurso en vídeo, "EdTechVideos - Code Monkey Island", https://www.youtube.com/watch?v=jtwfOw_8jsI , última visita: 12-06-17.

- [75] Mamie Chen, "Code Monkey Island – For Future Coders and Fun Seekers Alike", <http://jrmagellan.com/code-monkey-island-for-future-coders-and-fun-seekers-alike/> , última visita: 12-06-17.
- [76] Amazon, "Robot Turtle Game", https://www.amazon.com/gp/product/B00HN2BXUY/ref=as_li_tl?ie=UTF8&camp=1789&creative=9325&creativeASIN=B00HN2BXUY&linkCode=as2&tag=gtst-20 , última visita: 12-06-17.
- [77] Amazon, "Code Monkey Island", <https://www.amazon.com/gp/product/B00LTH8Y5G> , última visita: 12-06-17.
- [78] Code, "Hour fo Code", <https://hourofcode.com/es> , última visita: 12-06-17.
- [79] Code, Programa educativo “Hora del Código - Programa con Ana y Elsa”: <https://code.org/files/hoc-frozen-lp.pdf> , última visita: 12-06-17.
- [80] Code, Programa educativo “Hora del Código - Escribe tu primer programa”, origen: <https://code.org/files/hoc-maze-lp.pdf> , última visita: 12-06-17.
- [81] Code, Hoja de respuestas “Hora del Código - Programa con Ana y Elsa”, origen: <https://docs.google.com/presentation/d/1McUzaAQyGLfgJKR2Mbre-L-bL9xNAW3KdPQmH4nGR78/edit#slide=id.p3> , última visita: 12-06-17.
- [82] LearnToMod, “Ejercicio 1 – Actividad Introduction To Loops”, http://mod.learntomod.com/program_profiles/2052603 , última visita: 12-06-17.
- [83] LearnToMod, “Ejercicio 2 – Actividad Introduction To Loops”, http://mod.learntomod.com/program_profiles/2052702 , última visita: 12-06-17.
- [84] LearnToMod, “Ejercicio 3 – Actividad Introduction To Loops”, http://mod.learntomod.com/program_profiles/2052708 , última visita: 12-06-17.
- [85] LearnToMod, “Ejercicio 4 – Actividad Introduction To Loops”, http://mod.learntomod.com/program_profiles/2052754 , última visita: 12-06-17.
- [86] BQ DIWO, Bitbloq, "Actividad el difunto Robonauta", <http://diwo.bq.com/bq-invento-especial-halloween/> , última visita: 12-06-17.
- [87] Bitbloq, "Ejemplo 1 funcionamiento Mini servo y Zumbador", <http://bitbloq.bq.com/#/bloqsproject/594bf2c549a65600171143bc> , última visita: 12-06-17.
- [88] Bitbloq, "Ejemplo 2 funcionamiento sensor infrarrojo y Zumbador", <http://bitbloq.bq.com/#/bloqsproject/594bf7d949a65600171143be> , última visita: 12-06-17.
- [89] BQ DIWO, "Plantilla recortable Difunto Robonauta", http://diwo.bq.com/wp-content/uploads/2014/11/bq_Halloween_DifuntoRobonauta.pdf , última visita: 12-06-17.
- [90] Bitbloq, "Difunto Robonauta v2", <http://bitbloq.bq.com/#/bloqsproject/594b971e49a6560017114396> , última visita: 12-06-17.

Anexo A – Transcripciones de las entrevistas

A continuación, se encuentran las transcripciones de todas las entrevistas realizadas para el capítulo tres de este trabajo. Se encuentran en el mismo orden que aparecen en las tablas que resumen sus respuestas.

Índice - Anexo A - Transcripciones de las entrevistas

| | |
|---|-----|
| 1º Entrevista - Adrián Blasco (Enseñalia) | 69 |
| 2º Entrevista - Jesús Mínguez (Marianistas) | 76 |
| 3º Entrevista - David Bordonada (Libelium)..... | 81 |
| 4º Entrevista - Daniel Espinosa y Raquel García (Crom Developer)..... | 86 |
| 5º Entrevista - Pablo Aliaga (Autónomo DLABS) | 97 |
| 6º Entrevista - Beatriz García (Niubit) | 105 |
| 7º Entrevista - Andreu Camps (Avanza Tiempo Libre) | 115 |

1º Entrevista - Adrián Blasco (Enseñalia)

1 - Datos Personales

1.1 Nombre:

Adrián Blasco.

1.2 Centro/Profesión:

Profesor en Franquicias Enseñalia (<http://www2.ensenalia.com/>).

1.3 Titulación:

Ingeniería Industrial.

1.4 ¿Cuánto tiempo llevas impartiendo formación de este tipo?

Desde 2013-14, aproximadamente unos tres años.

1.5 ¿Cómo/Porqué empezaste a dar formación de este tipo?

Buscaba trabajo y encontré que en Enseñalia buscaban profesores que diesen formación de Robótica y nuevas tecnologías para niños. Nunca antes habían impartido formación de este tipo, así que al final el temario de los cursos lo propuse yo.

1.6 ¿Has dado formación de este tipo en algún otro centro? ¿Cuáles?

Si, en la Asociación HackerSpace Dlabs (<https://www.dlabs.co>) de la que formo parte.

2 - Opinión sobre la formación de STEM

2.1 ¿Crees que es importante impartir este tipo de formación? ¿Por qué?

Si, ya que hay muchos conceptos que más adelante se ven en la Universidad pero que ya se podrían haber introducido previamente en edades más tempranas. Cosas como el programar una placa, o hacer uso de un bucle o instrucciones condicionales, conceptos que nos pueden parecer muy complejos pero que realmente se pueden y que se deberían enseñar mucho antes, aunque sea solo una base y no con un planteamiento técnico, sino haciendo juegos y actividades entretenidas para los niños.

Hay conceptos, como la lógica booleana, algoritmia básica o como programar una placa que actualmente se enseñan en Bachiller o en la carrera universitaria, que se les podría enseñar a un niño pequeño si este está motivado y le gusta el tema. De hecho, estos conceptos se enseñan en la mayoría de cursos a niños que imparto.

Las nuevas generaciones tienen un contacto directo y constante con la tecnología prácticamente desde su nacimiento, de modo que, el saber programar, como funciona un programa o saber cómo manejar un ordenador, son asuntos fundamentales en la formación de un niño, aunque luego no vaya a dedicarse a trabajar en STEM.

2.2 ¿A partir de que edades? ¿Por qué?

Hay un error al aplicarlo por edades, depende de la capacidad, motivación o desarrollo personal de cada niño. Niños de mismas edades pueden diferir enormemente en su desarrollo, capacidades o motivación y por tanto no se les debería de tratar de enseñar igual, no es lo mismo tratar con niños motivados y con ganas que con niños que están en la actividad solo porque le han apuntado sus padres.

Si es cierto que, a la hora de organizar un curso, es necesario establecer unos límites de edad, ya que resultaría muy complicado ir evaluando previa e individualmente a cada niño para asignarle un curso más o menos avanzado, pero en condiciones ideales se debería de evaluar a cada niño para intentar asignarle a un grupo u otro. Por norma general, empezar en primero de primaria ya a la mayoría de los niños se les puede enseñar.

3 - Cursos/Clases

3.0 ¿Cuántos cursos o clases distintas impartes?

Ahora mismo doy cursos de: Robótica, Animatrónica con Impresión 3d, Programación de Videojuegos, Arduino Avanzado y Snap for Arduino. Además, también doy talleres (que son más cortos, de unas 20 horas aproximadamente) de: Chibitronics electrónica con stickers, Code y diseño e impresión 3D

Es una oferta bastante amplia de cursos, pero habría que aclarar comparten entre sí bastantes elementos, como que en casi todos en los que hay programación, es programación por bloques. Sí que se cambia el enfoque, dificultad y las herramientas y kits de Robótica con las que se trabajan.

3.1 ¿Cuántos alumnos tienes por clase? ¿Consideras que es un número de alumnos adecuado o debería ser mayor/menor?

El límite es de ocho alumnos máximo. Con una mayor cantidad resulta imposible atenderlos a todos, ya que en la mayoría de las aplicaciones aún hay muchas cosas por adaptar y hay que corregir fallos producidos por la plataforma/herramienta que estés usando, aparte de corregir los fallos que esté realizando el propio alumno.

Como límite mínimo serían cuatro alumnos, por rentabilidad de la clase para la Academia. Además, preparar clases para menos es distinto, son como clases particulares que se han de enfocar de forma distinta.

Estos límites máximos y mínimos son los que considero ideales.

3.2 ¿Qué perfil de alumno tienes? (edad, motivación, conocimientos previos, etc.)

Normalmente son niños de entre siete y trece años, que les interesa mucho todo lo relacionado con la informática y nuevas tecnologías, videojuegos, robots, etc. suelen ser ellos quienes deciden apuntarse.

Respecto a formación previa hay de todo, aunque los que suelen entrar nuevos a algún curso más básico como el de Robótica, no han hecho nunca nada, y suelen repetir en los años posteriores haciendo los más avanzados de Programación de Videojuegos o Arduino Avanzado. En las clases suele haber más niños que niñas, en grupos de 6 normalmente se tiene solo 2 niñas, aunque depende de la actividad y el curso.

3.3 ¿Cuánto duran las clases? ¿Consideras que es un tiempo adecuado o debería ser mayor/menor?

Depende del curso y la edad de los niños, niños pequeños (ocho a diez once años) una hora en programación sería lo ideal, cuando se enseña diseño 3D o se usan herramientas más “amenas” o “entretenidas” se podría estar más tiempo, ya que no es lo mismo enseñar una hora con una herramienta de programación por bloques, que una hora enseñando circuitos básicos con “RedStone” en Minecraft. Con alumnos más mayores (doce o trece años en adelante) sería un máximo de hora y media.

Además, es importante ir alternando de herramienta o actividad a lo largo del curso o incluso de la propia clase, ya que estar todo el tiempo con la misma herramienta acaba cansando tanto al alumno como al propio profesor.

Dejando a un lado lo que considero que es más correcto, en Enseñalia las clases tanto para mayores como para pequeños duran hora y media, y los talleres son de cuatro horas con un descanso de diez a veinte minutos en medio.

3.4 ¿Cuántas clases por semana? ¿Consideras que es un número de días adecuado o debería ser mayor/menor?

Una clase por semana, porque al final se trata de una actividad extraescolar de la que no hay un temario o una guía docente exacto definido, como si se tratara de una asignatura del colegio.

Se preparan las clases para que los alumnos aprendan, pero también se entretengan, ya que vienen porque ellos quieren aprender de estos temas en vez de dedicar la tarde a otras actividades. De hecho, se prioriza que el alumno se divierta aprendiendo antes que aprendan sí o sí.

Así que con una sesión por semana creo que es suficiente.

3.5 ¿Los alumnos disponen de material propio para las clases (por ejemplo, se traen sus propios portátiles) o los proporciona el centro?

El material por normal general lo proporciona el centro (portátiles y algunos componentes y kits).

En determinados cursos, como el de Robótica, los alumnos aparte de pagar el curso en sí, pagan un extra para los kits de robótica que van a usar. Dichos kits permanecen en el centro mientras dura el curso, pero una vez finalizado se les entregan a sus dueños.

Además, en el curso de Robótica, y algún otro, se diseñan pequeñas piezas o robots que se acaban imprimiendo con dos impresoras 3D que tenemos en el centro, para que los alumnos se las lleven a casa junto con su kit y así tengan un proyecto de un robot completo.

3.6 Dinámica de clase

3.6.1 ¿Cómo son las clases, explicas todo con ejemplos y luego das tiempo para que se trabajen en ejercicios, se van haciendo todos a la vez, etc.?

Depende del curso, por ejemplo, motivaciones para niños pequeños sería contar historias, es decir, desarrollo una historia que van a comprender y van a saber relacionar y a partir de esa historia propongo una actividad, se realiza y finalmente hago una conclusión.

La dinámica para chicos más mayores sería proponerles algo más práctico. Realizar una actividad en la cual primero enseñe cual sería el resultado y después ir explicando paso a paso como sería su desarrollo hasta llegar a ese resultado final.

Además, siempre intento relacionar con temas o asignaturas que puedan estar haciendo en el colegio, como la teoría del color, matemáticas, etc. Ya que es otra forma de mantenerlos motivados y que a la vez relacionen conceptos de distintas áreas.

3.6.2 ¿Siempre se trabaja con ordenadores o también realizas actividades sin ellos?

Sí, muy pocas actividades no requieren del portátil, porque se pueden sacar muchos recursos de un ordenador, al final hay que enseñarles a trabajar y manejar un ordenador, Internet, etc.

Aunque sí que en alguna ocasión se realizan actividades con recortables o se realizan pequeños cuestionarios en papel, pero siempre es además de trabajar con el ordenador, como por ejemplo una actividad que hicimos el año pasado por Halloween, que consistía en una calavera hecha con recreable en la que podías guardar unas monedas y con un sensor de infrarrojos y un servo programabas para que se cerrase si alguien metía la mano para robarte las monedas.

Y siempre cada alumno con su ordenador y kit de robótica propio, salvo que durante la clase se realice alguna actividad especial que necesite formar grupos.

3.6.3 ¿Mandas deberes/trabajo para casa?, en caso afirmativo, ¿con que frecuencia?

No, no mando deberes, aunque si aviso de que existe la posibilidad de hacer ejercicios o actividades desde casa, por si algún alumno se ha quedado con ganas de más.

3.6.4 ¿Haces exámenes pruebas de evaluación?, en caso afirmativo, ¿con que frecuencia?

No, sí que hago pequeñas actividades con papel y bolígrafo en las que planteo preguntas, o ejercicios, o planteo un problema para todos, pero no los evalúo o puntúo, son más para romper la rutina si llevamos muchas clases haciendo lo mismo con la misma herramienta que con ánimo de poner una nota.

4 - Conceptos que enseñas y herramientas que usas

4.1 ¿Qué conceptos enseñas? (P. Ej., condicionales, bucles, funciones, etc.)

Dependen del Curso, en los básicos, condicionales, bucles, funciones, variables, llamadas a funciones, un poco de E/S analógicas y digitales y un poco de comunicaciones Bluetooth.

También algunos conceptos de matemáticas, teoría del color... y en general cualquier concepto de ciencias que pueda encontrar una relación o aplicación con el curso o la clase que estoy dando. Se intenta siempre relacionar conceptos de programación con otras ciencias con las que quizás sí que estén familiarizados, tanto para motivarles como para que les resulte más sencillo entender los conceptos de programación y lo útiles que puede resultar por ejemplo a la hora de resolver problemas de matemáticas, por ejemplo.

En cursos más avanzados, se profundiza más en lo nombrado previamente y también en programación orientada a objetos y programación escrita (es decir, sin usar herramientas de programación por bloques).

4.1.1 ¿Por qué?

Porque son los básicos necesarios para ir aprendiendo programación y son los que la mayoría de herramientas y plataformas promulgan o facilitan.

4.1.2 ¿Hay algún otro concepto que te gustaría enseñar?

Más contenidos prácticos o aplicados, por ejemplo, clases, objetos, herencia. Que son partes que cuestan más y que en los básicos no llegan y en los avanzados se quedan con lo justo.

También más telecomunicaciones, ya que se da muy poco de comunicación Bluetooth y dar otras como Wifi o Xmin.

Y también programación Web (JavaScript), ya que hay tecnologías muy interesantes que muchos de los alumnos acabarán trabajando o usando.

4.2 ¿Qué herramientas usas? (P. Ej. Bitbloq, Code.org, Scratch, etc.)

Siempre intento utilizar herramientas abiertas o que ellos luego en casa las puedan seguir utilizando. Por norma general aplicaciones web, porque son multiplataforma y creo que todos nuestros alumnos tienen conexión a Internet en casa y pueden trabajar por su cuenta.

En los distintos cursos utilizo Bitbloq, TinkerCAD, LearnToMod, Snap for Arduino, Scratch y Code.org.

4.2.1 ¿Por qué?

Porque son las que considero mejores en sus “categorías”, además como te decía los alumnos puedan hacer uso de ellas de casa y cuentan con suficientes funcionalidades y tutoriales.

4.2.2 ¿Hay alguna otra herramienta que te gustaría usar?

Suponiendo que no hubiese un límite en el presupuesto para materiales, como comentaba antes me gustaría disponer de kits de robótica o componentes especializados en comunicaciones para poder extender esa parte.

5 - Opinión sobre Herramientas

5.1 Bitbloq

5.1.1 ¿Conoces esta herramienta? ¿Qué te parece?

Si, la conozco y la uso, y me parece una herramienta muy completa, cuenta con muchos ejemplos y en general está bien.

5.1.2 ¿Qué hechas en falta/Qué mejorarías?

Soporte por parte de BQ, ya que, aunque es una herramienta relativamente nueva y muy buena, cuando encuentras algún fallo, tardan bastante en solucionar los fallos que vas encontrando, y si la plataforma no funciona y has preparado una clase en la que haces uso de la misma... pues estás vendido.

Se echa de menos que dispongan de más componentes, ya que se han quedado con lo básico: potenciómetro, motor, etc. En comunicaciones se quedan cortos, incluso con los kits que incluyen Bluetooth con el que no se puede interactuar todo lo que me gustaría.

5.2 Code.org

5.2.1 ¿Conoces esta herramienta? ¿Qué te parece?

Si, la conozco y la uso, y me parece una herramienta muy completa, cuentan con un montón de ejemplos y tutoriales.

Además, dispone de unas guías docentes por edades o temas muy buenas, que enseguida te permiten montar un curso o una clase. Las actividades tienen un montón de gráficos distintos para hacerlas más amena según el niño prefiera una u otra.

Se nota que es una herramienta muy trabajada en la que se ha invertido mucho dinero y cuenta con el respaldo y patrocinio de personas famosas del sector de la informática, como es Zuckerberg.

5.2.2 ¿Qué hechas en falta/Qué mejorarías?

Nada destacable, es una herramienta muy completa tanto para los alumnos como para los profesores y además tiene un buen soporte y pocos o ningún fallo en ella.

5.3 Scratch

5.3.1 ¿Conoces esta herramienta? ¿Qué te parece?

Sí, me parece bien. Como herramienta para programar por bloques es bastante completa tiene una serie de manuales y tutoriales de apoyo y dispone de distintas “ramas” que se enfocan cada una en el temario que quieras dar (rama de matemáticas para hacer ejercicios enfocados a matemáticas, etc.).

5.3.2 ¿Qué hechas en falta/Qué mejorarías?

Las posibilidades resultan infinitas (número de cursos y actividades disponibles que puedes localizar ya dentro de la propia aplicación de Scratch como de gente que las haya propuesto por Internet) y es labor del profesor establecer el límite de que cosas quiere dar. No hay unas guías docentes como puede haber en Code.

6 Recomendaciones, otros

6.1 Si tuvieses que recomendar una herramienta, ¿cuál sería? ¿Por qué?

Yo me quedaría con todas las nombradas, todas tienen su uso y mala no hay ninguna; y las que son malas no las uso.

Además, para recomendar alguna habría que tener en cuenta varios factores, como que temario se desea dar, la edad/capacidad del alumno y también que es lo que desea aprender, si quiere aprender java, habría que buscar cursos online o herramientas que faciliten el aprendizaje de java y así...

6.2 ¿Alguna otra valoración sobre este tipo de formación?

Otra consideración, resulta muy importante el lenguaje o el cómo nos comunicamos, no hay que olvidar que tratamos con niños que vienen de forma voluntaria a que les demos clase, Vienen a aprender, pero divirtiéndose mientras lo hacen.

Nunca utilizar palabras como “examen” o “ejercicios”, en su lugar, “Vamos a hacer una prueba” o una “actividad” o “juego”, a fin de cuentas, son niños que vienen a aprender, pero también a divertirse.

Se echa de menos un canon, no está establecido el temario que debe de darse para todas estas nuevas tecnologías, de modo que te encuentras con muchísimos ejemplos y temarios distintos y distintas formas de darlo y es labor del profesor el quedarse de todas estas opciones con las que más le interesan

6.3 ¿Sabes de algún otro profesor/centro/academia que imparta formación de este tipo?

Crom Developer, Etopia Kids

2º Entrevista - Jesús Mínguez (Marianistas)

1 - Datos Personales

1.1 Nombre:

Jesús Mínguez.

1.2 Centro/Profesión:

Profesor y Administrador de Sistemas, Colegio Santa María del Pilar Marianistas (<http://www.marianistas.net/>).

1.3 Titulación:

Ingeniería Electrónica Industrial.

1.4 ¿Cuánto tiempo llevas impartiendo formación de este tipo?

Quince años dando formación en Secundaria y Bachiller y en los dos últimos años a primaria (Cuarto, Quinto y Sexto de primaria, es decir, de nueve a doce años).

1.5 ¿Cómo/Porqué empezaste a dar formación de este tipo?

Se requería por parte del colegio dar formaciones de este tipo.

1.6 ¿Has dado formación de este tipo en algún otro centro? ¿Cuáles?

No, solo en Marianistas.

2 - Opinión sobre el tema

2.1 ¿Crees que es importante impartir este tipo de formación? ¿Por qué?

Sí, creo que hay que empezar a fomentar el pensamiento computacional lo antes posible, porque fomenta un modo de resolución de problemas y una estructuración mental muy correcta y adecuada a la hora de enfrentarse a problemas de la vida real.

2.2 ¿A partir de que edades? ¿Por qué?

Cuanto antes mejor, estoy viendo y trabajando con cosas preparadas para niños de dos y tres años en adelante. Si la herramienta está adaptada, se puede empezar perfectamente a esas edades a enseñar pensamiento computacional.

3 - Cursos/Clases

3.0 ¿Cuántos cursos o clases distintas impartes?

Actualmente hacemos proyectos anuales a alumnos de Quinto y Sexto de Primaria (diez a doce años), aunque nuestra intención es ir extendiendo a cursos con alumnos más pequeños hasta Segundo o Tercero (es decir, de siete a nueve años).

Aunque la dificultad y temario de los cursos difiere en función de la edad de los niños, siempre es temario relacionado con la programación, son cursos introductorios evidentemente, intentando siempre relacionarlo todo con otras disciplinas como el Ajedrez, Matemáticas o Ciencias. Además, damos clases extraescolares de Robótica a cursos de Secundaria.

3.1 ¿Cuántos alumnos tienes por clase? ¿Consideras que es un número de alumnos adecuado o debería ser mayor/menor?

Normalmente, unos veinticinco alumnos, y es un número de alumnos mayor del que nos gustaría tener.

Con grupos más reducidos y mejores instalaciones se trabajaría mejor, ya que dispones de más tiempo para dedicarle a las dudas que les puedan surgir a cada uno de ellos, y con mejor material muchas cosas irían más rápidas, no habría tantos cuelgues, etc.

3.2 ¿Qué perfil de alumno tienes? (edad, motivación, conocimientos previos, etc.)

Alumnos de entre diez y doce años con pocos o ningún conocimiento previo y distintas motivaciones, ya que el curso forma parte de la guía docente, y es, por tanto, de obligada asistencia.

3.3 ¿Cuánto duran las clases? ¿Consideras que es un tiempo adecuado o debería ser mayor/menor?

Duran cincuenta minutos y lo considero un tiempo adecuado.

En ese tiempo se puede dar una primera introducción a lo que se va a trabajar a lo largo de la clase, trabajar en ello y mantener al alumno motivado todo el rato y que no se canse antes de tiempo de la actividad.

3.4 ¿Cuántas clases por semana? ¿Consideras que es un número de días adecuado o debería ser mayor/menor?

Una hora semanal dentro del área de Matemáticas (es decir, de las horas semanales que están asignadas a clase de Matemáticas, una de ellas se dedica enteramente a este tipo de formación).

3.5 ¿Los alumnos disponen de material propio para las clases (por ejemplo, se traen sus propios portátiles) o los proporciona el centro?

Cada alumno dispone de un portátil para cada uno que el colegio proporciona mientras dura la clase.

3.6 Dinámica de clase

3.6.1 ¿Cómo son las clases, explicas todo con ejemplos y luego das tiempo para que se trabajen en ejercicios, se van haciendo todos a la vez, etc.?

Al principio se propone un ejercicio y se van dando distintos pasos para llegar a solucionarlo. Se da un tiempo entre cada paso para que, a partir de las explicaciones iniciales y los pasos previos, el alumno sepa solucionar el problema, o al menos intuir por donde hay que ir.

Se intenta incentivar distintas formas de alcanzar la solución, de modo que al final de la actividad, cuando se ha alcanzado ese resultado final, el alumno debe darse cuenta de la gran variedad de soluciones existentes a parte de la que ha usado él.

No hay un único resultado posible, hay cientos, de modo que se le dan las herramientas o medios para llegar a alguno de ellos. Lo más importante es que se den cuenta de que su opción no es la única, ni necesariamente la mejor, que sean conscientes de la cantidad de herramientas que tienen a sus disposiciones para poder resolver el problema.

3.6.2 ¿Siempre se trabaja con ordenadores o también realizas actividades sin ellos?

Sí, siempre trabajamos con el ordenador haciendo uso de una herramienta u otra.

3.6.3 ¿Mandas deberes/trabajo para casa?, en caso afirmativo, ¿con que frecuencia?

No, solo se trabaja en el aula, aunque cada alumno es libre de trabajar en casa si así lo desea.

3.6.4 ¿Haces exámenes pruebas de evaluación?, en caso afirmativo, ¿con que frecuencia?

No, no es necesario.

4 - Conceptos que enseñas y herramientas que usas

4.1 ¿Qué conceptos enseñas? (P. Ej., condicionales, bucles, funciones, etc.)

No me centro en temas tan técnicos como puede ser enseñar un bucle o una variable, intento enseñar a los chavales a como pensar que funcionan las cosas, lo que se llama pensamiento comunicacional, como programar, como darse cuenta de que las cosas no ocurren solas, sino que hay una consecución de pasos, un programa detrás, que es el que hace que funcionen las cosas correctamente.

Enseñar más bien la lógica de funcionamiento, no es hablar de bucles o condicionales o este tipo de cosas, porque quizás muchos de ellos no saben de lo que estamos hablando ni siquiera en la realidad como para luego aplicarlo a un mundo matemático.

Lo que sí tienen que aprender es cierta lógica de trabajo y de funcionamiento y con herramientas de desarrollo muy sencillas ellos mismos van descubriendo que existen condicionales, bucles, etc.

4.1.1 ¿Por qué?

Creo que es más relevante enseñar pensamiento computacional que conceptos tan técnicos como bucles, variables, etc.

4.1.2 ¿Hay algún otro (concepto/herramienta) que te gustaría usar?

No, con más edad sí que nos meteríamos en cuestiones técnicas, pero para las edades que barajamos, está bien así.

4.2 ¿Qué herramientas usas? (P. Ej. Bitbloq, Code.org, Scratch, etc.)

Trabajo principalmente con el IDE de Arduino, Bitbloq y Code.org, aparte de herramientas de diseño gráfico 3D como OpenScap, tinkercAD. Todas ellas en el contexto de la asignatura extraescolar de robótica, no en la de introducción a la programación.

Para introducción a la programación, la clase para niños de Primaria, solo trabajamos con Scratch.

4.2.1 ¿Por qué?

Las de robótica porque son bastante completas y permiten hacer una programación y diseño básico de robots. Y luego Scratch es tremendamente sencillo y completo y cubre todas las necesidades para los cursos de introducción a la programación en Primaria.

4.2.2 ¿Hay algún otro (concepto/herramienta) que te gustaría usar?

No hay nada que me llame especialmente la atención, sí que hay varias soluciones que son muy buenas como por ejemplo los kits de robótica de Lego, pero es un plug and play y no representa los problemas de verdad de electrónica que te vas a encontrar en el futuro, que es algo que a mí sí que me gusta enseñar, aunque si aprendes a programar y ves los resultados de lo que estas programando.

5 - Opinión sobre Herramientas que analizo

5.1 Bitbloq

5.1.1 ¿Conoces esta herramienta? ¿Qué te parece?

Si y me gusta, aunque de vez en cuando tiene algunos problemas de conectividad y tal, pero en general funciona bien.

5.1.2 ¿Qué hechas en falta/Qué mejorarías?

Aunque Bitbloq me gusta, prefiero trabajar directamente con el IDE de Arduino porque no falla y conecta bien y tiene miles de posibilidades, la contra de este es que ya no es programación por bloques sino programación escrita tradicional.

Además, trabajar con Bitbloq implica trabajar con una plataforma Web a la que no siempre se puede acceder porque falla la red o esta caída o la están actualizando. Una posible mejora sería tenerla como si se tratase de una aplicación de escritorio, como Scratch.

5.2 Code.org

5.2.1 ¿Conoces esta herramienta? ¿Qué te parece?

Sí que la conozco, pero he trabajado poco con él.

5.2.2 ¿Qué hechas en falta/Qué mejorarías?

No he trabajado lo suficiente como para conocerla bien y proponer alguna mejora.

5.3 Scratch

5.3.1 ¿Conoces esta herramienta? ¿Qué te parece?

Si y me encanta, para los chavales es espectacular como aprenden con esta herramienta y ya cuenta de por sí con una gran cantidad de ejemplos y ejercicios ya actividades propuestas.

Además, en Internet, aparte de lo que ya tienes en la herramienta de por sí, hay muchísimas actividades, tutoriales y problemas planteados por otros usuarios y profesores, algunos de ellos muy chulos y completos.

5.3.2 ¿Qué hechas en falta/Qué mejorarías?

Poco o nada.

6 - Recomendaciones, otros

6.1 Si tuvieses que recomendar una herramienta, ¿cuál sería? ¿Porqué?

Para introducción a la programación y que los chavales aprendan a programar me quedo con Scratch y para Robótica con el IDE de Arduino.

6.2 ¿Alguna otra valoración sobre este tipo de formación?

Por ejemplo, algo con lo que no he trabajado aún y me gustaría es con Minecraft que creo que se pueden generar cosas muy majas.

Además, en casa con mi hija de siete años, a nivel particular, estoy usando unos Robots llamados Dash and Dot, de aspecto parecido al robot nuevo de la Guerra de las Galaxias y en el que se puede programar rutas aparte de controlarlo como si fuesen un coche tele-dirigido.

6.3 ¿Sabes de algún otro profesor/centro/Academia que imparta formación de este tipo?

No caigo en nadie ahora mismo.

3º Entrevista - David Bordonada (Libelium)

1 - Datos Personales

1.1 Nombre:

David Bordonada.

1.2 Centro/Profesión:

Comercial de Libelium y Cooking Hacks (<http://www.libelium.com/>) y Profesor WaspMote.

1.3 Titulación:

Ciencias Empresariales

1.4 ¿Cuánto tiempo llevas impartiendo formación de este tipo?

Estuve 3 años dando formación y apoyo a profesores y centros que usan nuestros kits, formaba parte de nuestro trabajo aparte de hacer contacto con los distintos Centros y Colegios, el dar asesoramiento para las clases.

1.5 ¿Cómo/Porqué empezaste a dar formación de este tipo?

Cuando empecé a trabajar con Libelium tuve mi primer contacto con Arduino y me tocó vender nuestros productos y dar formaciones de este tipo a profesores y de apoyo también para los alumnos que estos iban a tener.

1.6 ¿Has dado formación de este tipo en algún otro centro? ¿Cuáles?

No, solo con Libelium y su filial para comercio al por menor Cooking Hacks.

2 - Opinión sobre el tema

2.1 ¿Crees que es importante impartir este tipo de formación? ¿Por qué?

Sí, creo que está bien ir formando en nuevas tecnologías a los niños desde que son pequeños.

2.2 ¿A partir de que edades? ¿Por qué?

Desde primero de primaria a casi Bachillerato, ya que nosotros dábamos una extra escolar de Robótica en familia en la que aparte del propio niño estaba el padre o la madre como apoyo para la actividad.

3 - Cursos/Clases

3.0 ¿Cuántos cursos o clases distintas impartes?

Eran sobre todo clases extra escolares de Robótica, Robótica en familia, talleres dedicados a cada uno de los kits que vendemos como el de WaspMote, etc.

3.1 ¿Cuántos alumnos tienes por clase? ¿Consideras que es un número de alumnos adecuado o debería ser mayor/menor?

El primer año teníamos entre veintisiete a treinta alumnos en total, que hacía que resultase complicado dar las clases porque eran demasiados, y eso que estábamos varios profesores.

A partir del segundo año ya decidimos dividirnos en dos grupos más reducidos de diez o doce personas divididas por niveles, grupos más avanzados para los que ya tenían cierta experiencia con los kits o para los alumnos más mayores y básicos para los niños pequeños o que fuese el primer año que entraban a la actividad.

3.2 ¿Qué perfil de alumno tienes? (edad, motivación, conocimientos previos, etc.)

De edades muy variadas, como te decía teníamos chicos desde primero de primaria hasta casi bachiller y con interés por el tema.

Respecto a si tenían conocimientos previos o no, había de todo, tanto niños que ya sabían algo de programación por bloques como niños que empezaban.

3.3 ¿Cuánto duran las clases? ¿Consideras que es un tiempo adecuado o debería ser mayor/menor?

Eran dos horas los viernes por la tarde.

El hecho de que fuese los viernes por las tardes provocaba que en algunas ocasiones faltase gente porque se preparaban puentes o salían el fin de semana. Con lo que había ocasiones que al faltar alumnos una semana, a la semana siguiente era necesario ponerles al día y así, por lo que las clases se ralentizaban.

Y el tiempo de dos horas me parece adecuado, eran actividades junto con sus padres muy entretenidas que no se les hacían largas, ya que también las dinámicas de las clases incluían un tiempo de “juego” haciendo uso de los conceptos que se les habían enseñado en esa sesión.

3.4 ¿Cuántas clases por semana? ¿Consideras que es un número de días adecuado o debería ser mayor/menor?

Se daba clase un viernes cada dos semanas.

3.5 ¿Los alumnos disponen de material propio para las clases (por ejemplo, se traen sus propios portátiles) o los proporciona el centro?

Los proporciona el centro, por un lado, los ordenadores necesarios para programar la placa y luego los kits y material que nos habían comprado previamente a nosotros.

3.6 Dinámica de clase

3.6.1 ¿Cómo son las clases, explicas todo con ejemplos y luego das tiempo para que se trabajen en ejercicios, se van haciendo todos a la vez, etc.?

Son clases en las que los niños están con sus padres, que sirven de apoyo para los conceptos que vamos explicando.

Al principio de la misma se explicaba lo que se iba a dar y luego se desarrollaba poco a poco, dejando al final tiempo para que los niños y las familias jueguen con los kits y los conceptos aprendidos.

3.6.2 ¿Siempre se trabaja con ordenadores o también realizas actividades sin ellos?

Siempre con los ordenadores y los kits.

3.6.3 ¿Mandas deberes/trabajo para casa?, en caso afirmativo, ¿con que frecuencia?

No, era un extra-escolar.

3.6.4 ¿Haces exámenes pruebas de evaluación?, en caso afirmativo, ¿con que frecuencia?

No.

4 - Conceptos que enseñas y herramientas que usas

4.1 ¿Qué conceptos enseñas? (P. Ej., condicionales, bucles, funciones, etc.)

Conceptos de electrónica y programación básica.

4.1.1 ¿Por qué?

Porque son los que se pretende enseñar con nuestros kits.

4.1.2 ¿Hay algún otro (concepto/herramienta) que te gustaría usar?

No.

4.2 ¿Qué herramientas usas? (P. Ej. Bitbloq, Code.org, Scratch, etc.)

Scratch 2 y el adaptador para enlazar Scratch y Arduino, S4A.

4.2.1 ¿Por qué?

Se adapta perfectamente a nuestras necesidades y nuestros kits.

4.2.2 ¿Hay algún otro (concepto/herramienta) que te gustaría usar?

No.

5 - Opinión sobre Herramientas que analizo

5.1 Bitbloq

5.1.1 ¿Conoces esta herramienta? ¿Qué te parece?

Sí que he oído hablar de ella y conozco a gente que está trabajando para BQ y con la rama de Robótica y sus kits, pero no la he usado

5.1.2 ¿Qué hechas en falta/Qué mejorarías?

-No da lugar-

5.2 Code.org

5.2.1 ¿Conoces esta herramienta? ¿Qué te parece?

No, no la conozco.

5.2.2 ¿Qué hechas en falta/Qué mejorarías?

-No da lugar-

5.3 Scratch

5.3.1 ¿Conoces esta herramienta? ¿Qué te parece?

Si, y nosotros usamos el fork S4A que creo que lo hizo una universidad de Cataluña, que básicamente permite montar aplicaciones de Arduino usando Scratch.

5.3.2 ¿Qué hechas en falta/Qué mejorarías?

Lo veo bien tal y como está.

6 - Recomendaciones, otros

6.1 Si tuvieses que recomendar una herramienta, ¿cuál sería? ¿Porqué?

Recomendaría Scratch y S4A, que son las que conozco y usamos y que creo que están muy bien tal y como están.

6.2 ¿Alguna otra valoración sobre este tipo de formación?

Antes el perfil de cliente era personas más técnicas, con formación en el tema de electrónica y demás, pero cada vez se pone en contacto con nosotros gente de Centros Educativos o Colegios que quieren aprender desde cero y no cuenta con tantos conocimientos previos de electrónica y programación y tenemos más consultas o que quieren solucionar dudas que les han ido surgiendo con los kits. Para esto último contamos con el foro público de Libelium donde se van respondiendo todas las dudas y consultas que nos van llegando.

Nosotros no somos un centro de formación, así que no nos encargamos de dar las clases tal cual, sí que hacemos talleres de formación a los profesores que van a usar nuestros kits y asistimos a las primeras sesiones, pero la idea es que el Centro y sus

profesores aprendan a usar nuestros kits y sean ellos los que decidan como dar las clases o que enseñar.

Además, resolvemos las dudas que se les puedan plantear a los profesores y facilitamos muchos ejemplos y ejercicios o actividades que se pueden hacer con los kits y como te decía disponemos de un foro donde pueden poner cualquier duda o consulta y que tiene muchísima información.

6.3 ¿Sabes de algún otro profesor/centro/Academia que imparta formación de este tipo?

No que ahora caiga.

4º Entrevista - Daniel Espinosa y Raquel García (Crom Developer)

1 - Datos Personales

1.1 Nombre:

Daniel Espinosa y Raquel García.

1.2 Centro/Profesión:

Autónomos y fundadores de la empresa de Crom Developer (<http://cromdeveloper.com>), empresa de desarrollo de Software y aplicaciones a medida y formación de nuevas tecnologías.

1.3 Titulación:

Grado Superior de Desarrollo de Aplicaciones Informáticas y Formación Profesional de Desarrollo de Aplicaciones Multiplataforma.

1.4 ¿Cuánto tiempo llevas impartiendo formación de este tipo?

Desde el inicio de la empresa, hace ya cinco años. Aunque el planteamiento inicial era dar formaciones a Profesores y Centros de enseñanza, no directamente a niños. Los primeros cursos a Profesores eran sobre uso de Redes Sociales, nuevas tecnologías e Internet.

1.5 ¿Cómo/Porqué empezaste a dar formación de este tipo?

A raíz de un profesor que nos pidió ayuda para un Scratch Day, luego salieron las primeras colonias de Etopia Kids y entramos ya a dar formaciones y participar en ellas.

Luego ya cursos y actividades extra escolares en colegios porque eran muy demandados por los padres. Finalmente empezamos a dar “Creatividad Tecnológica” que es el curso que damos principalmente en los colegios, que en realidad más que un curso es una actividad extra-escolar.

1.6 ¿Has dado formación de este tipo en algún otro centro? ¿Cuáles?

No, siempre ha sido con nuestra empresa Crom Developer, aunque sí que hemos estado subcontratados por otras organizaciones o empresas, como por ejemplo para Etopia Kids en las colonias que hacen en verano.

2 - Opinión sobre el tema

2.1 ¿Crees que es importante impartir este tipo de formación? ¿Por qué?

Creemos que es importante dar formación de este tipo, pero también creemos que se está pasando la gente apuntando a niños muy jóvenes que no son capaces ni de interaccionar correctamente con un ordenador, manejar el ratón, etc.

2.2 ¿A partir de que edades? ¿Por qué?

Nuestra edad límite serían los ocho años en adelante, pero nos encontramos a muchos padres de niños de seis años que quieren que sus hijos hagan actividades tecnológicas. Y, aunque sí que se les puede preparar formación muy específica y adaptada, hay que tener en cuenta que a esas edades la mayoría de los niños solo han interactuado o se saben desenvolver con móviles y tablets, no con un ratón y un ordenador. No tienen suficiente psicomotricidad en las manos como para manejar un ratón con soltura y ni mucho menos escribir en un teclado.

Además, también es necesario un nivel de comprensión lectora que no todos los niños tienen a edades tan tempranas, si aún no son capaces de leer y comprender con fluidez lo que están leyendo es imposible proponer una actividad de programación por bloques, porque ya van a tardar demasiado solo en leer lo que pone en la etiqueta de cada bloque.

Aunque sean nativos digitales hay muchos conocimientos o conceptos que no tienen interiorizados, como por ejemplo apagar un ordenador, muchos niños no saben cómo se hace, y piensan que con bajar la tapa del portátil basta ya es suficiente o es que directamente no saben dónde hay que ir para hacerlo.

También nos encontramos con el problema de que, en el Colegio, donde se les debe enseñar muchas cosas básicas, hay muchos profesores que no están preparados o no son capaces de enseñarles lo necesario. Son profesores que son igual de desconocidos de la tecnología que los propios alumnos, puede que más incluso.

3 - Cursos/Clases

3.0 ¿Cuántos cursos o clases distintas impartes?

El curso que damos nosotros se llama “Creatividad Tecnológica”, y está planteado como una actividad extra-escolar para Centros Educativos y Colegios que se desarrolla a lo largo del todo el año (de octubre a mayo). Evidentemente es un curso genérico cuya dificultad adaptamos en función de las edades los niños a los que se lo vamos a impartir.

3.1 ¿Cuántos alumnos tienes por clase? ¿Consideras que es un número de alumnos adecuado o debería ser mayor/menor?

Solemos trabajar con aproximadamente diez más menos dos alumnos, porque grupos más grandes o más pequeños no son viables. Partiendo de la base de que nos gusta nuestro trabajo, pero también tenemos que poder vivir de él, ya que es una de las actividades principales de nuestra empresa.

Respecto a si resulta un número adecuado o no de alumnos, nos gustaría poder trabajar con grupos más reducidos (grupos de cuatro sería lo ideal), pero no nos resulta rentable, solo podríamos plantearnos grupos tan reducidos si a raíz de realizar unas pocas sesiones con cuatro o cinco alumnos en un colegio, luego nos salieran veinticinco más de ese mismo centro interesados en participar.

Si el único enfoque que hubiera que tener en cuenta fuese el educativo, por norma general, cuantos menos alumnos mejor. Aunque también teniendo un límite a la baja, todo dependiendo del tipo de alumno que te toca en el grupo. Porque, aunque en condiciones normales cuatro alumnos es un buen número, a veces todos los alumnos del grupo son más bien parados o poco participativos, y en ese caso tener uno o dos alumnos más mejora la dinámica de la clase.

Hay que tener un equilibrio, ya que, aunque el número de alumnos es importante, también lo es que clase de grupo te toca, si hay alumnos más o menos movidos, si están interesados en el tema, si se conocen de antes o son amigos entre ellos, etc.

Además, a menor número de alumnos, normalmente, más avanza a la hora de desarrollar las actividades. En grupos reducidos es más fácil ir atendiendo las dudas de cada uno individualmente y por tanto resulta más sencillo que toda la clase vaya a un mismo ritmo, con lo que se avanza más que en grupos grandes, donde vas a tener muchos alumnos que se te van a adelantar porque les resulta más fácil y alumnos que van a ir más despacio porque se han descolgado en algún momento, etc. También con el matiz de que, dada una clase con seis o siete alumnos, poca diferencia va a haber con una de diez.

Vamos, que depende principalmente de cómo son los niños y de sus edades. Un grupo de edades heterogéneas acarrea más problemas porque el desarrollo personal de cada niño es distinto, y ya tienes que plantear la clase teniendo en cuenta que el niño de diez años va a terminar el problema que propongas antes que los que tengan seis o siete años. Así que hay que plantear actividades extra o retos adaptados a cada uno para que no haya ninguno descolgado o que se aburra en clase, ya sea porque avanzan mucho o porque son actividades muy exigentes para su edad. Grupos reducidos de edades parecidas es lo ideal.

3.2 ¿Qué perfil de alumno tienes? (edad, motivación, conocimientos previos, etc.)

Depende de la edad, pero en los colegios, aunque son mayoría los que se apuntan porque ellos quieren, la última palabra la tienen siempre sus padres que son los que les pueden inscribir o no en las actividades extraescolares. De hecho, en muchas ocasiones la iniciativa de apuntarles al curso es de los padres antes que, del niño, aunque luego al niño le encante e insista en repetir en años posteriores.

Normalmente, conforme se empieza a subir edad, los niños van ganando independencia y si se acaban apuntando al curso es por su propia iniciativa.

También se suele dar el caso de que hay una mayor continuidad entre los niños pequeños que van repitiendo robótica cada año. Luego, los mayores, conforme empiezan la adolescencia, varían más en gustos y un año hacen robótica, al siguiente se apuntan cocina, luego a otra cosa, luego vuelven, etc.

Además, en nuestra actividad solemos tener muchos más niños que niñas, dato que se acentúa mucho más conforme van creciendo y van entrando en la adolescencia.

3.3 ¿Cuánto duran las clases? ¿Consideras que es un tiempo adecuado o debería ser mayor/menor?

Una hora para las clases orientadas a niños de “Creatividad Tecnológica” y hora y media para alumnos de instituto.

Sobre lo que debería durar, menos de cuarenta y cinco minutos sería muy poco tiempo, porque no daría tiempo a plantear una actividad y desarrollarla y una hora y media sería demasiado, son niños pequeños que necesitan levantarse moverse, y tener estímulos constantes y sería difícil mantener su atención constante durante tanto tiempo.

En las ocasiones que hemos tenido que estar cuatro horas seguidas, como en las colonias de Etopia Kids y se hace muy duro, tanto para los niños como para el propio profesor.

Igualmente habría que tener en cuenta qué tipo de actividad se desea desarrollar: no es lo mismo estar trabajando con TinkerCAD o una aplicación de diseño en 3D que estar con programación por bloques o enseñando algo más teórico, programando un juego o haciendo alguna actividad manual. Cuanto más relajada sea la actividad y menor concentración necesiten los niños para llevar a cabo la actividad más tiempo podrán estar trabajando en ella sin tener que parar para descansar.

3.4 ¿Cuántas clases por semana? ¿Consideras que es un número de días adecuado o debería ser mayor/menor?

Una vez por semana durante todo el curso.

Creemos que sería preferible hacer sesiones de una hora dos veces por semana, aunque, de momento, los cuatro años que llevamos con la actividad, nos está funcionando muy bien este horario.

3.5 ¿Los alumnos disponen de material propio para las clases (por ejemplo, se traen sus propios portátiles) o los proporciona el centro?

Normalmente son los propios Centros educativos y Colegios los que proporcionan el material, aunque en otras ocasiones nos toca ponerlo a nosotros, disponemos para estos casos de un aula portátil con seis equipos y si es necesario la podemos llevar al centro donde se tiene que impartir la clase.

Normalmente tenemos dos alumnos por cada ordenador de forma intencionada, porque preferimos que los alumnos trabajen por parejas ya que se estimula el que compartan el material, se ayuden los unos a los otros y trabajen en grupo. Además, es mejor que trabajen por parejas que en grupos más grandes. En un grupo de tres niños siempre va a haber uno que se descuelgue de la actividad porque no quiere participar porque se aburre o porque los otros no le dejan.

Los alumnos más mayores en ocasiones sí que vienen con sus propios portátiles, pero porque ellos así lo prefieren, ya que así tienen todo el material que hemos hecho en clase y luego ellos en casa pueden seguir por su cuenta con todas las aplicaciones y herramientas que utilizamos en clase ya instaladas y configuradas.

3.6 Dinámica de clase

3.6.1 ¿Cómo son las clases, explicas todo con ejemplos y luego das tiempo para que se trabajen en ejercicios, se van haciendo todos a la vez, etc.?

Normalmente es como una clase guiada, vamos paso a paso explicando cada una de las cosas que hay que hacer y con cada actividad que hacemos vamos pasando de un bloque de programación a otro (Ahora en esta actividad proponemos un problema que hay que usar el bloque de mover y los de condicional, ahora en el siguiente problema además de eso vamos a utilizar sonidos, etc.).

Seguimos esa dinámica de trabajo, salvo que se esté plateando una actividad súper creativa, como por ejemplo esta semana, que en un centro estamos trabajando y desarrollando actividades con píxeles y se les propuso diseñar naves espaciales. En ese tipo de actividades sí que se deja más sueltos a los alumnos y cada uno va poco a poco haciendo la actividad por su cuenta, estando nosotros atentos por si les surgen dudas para poder ayudarles.

Sí que se podría dar más libertad y proponer un problema para que resuelvan en aquellos casos en los que los niños ya sepan cómo se programa y se usa la herramienta o ya lleven varios años con nosotros.

Por ejemplo, cuando tenemos clases con alumnos de distintas edades, no podemos plantear las mismas actividades para niños de ocho que para otros de diez, normalmente cuanto más pequeños más necesario es el hacer actividades guiadas, con alumnos más mayores sí se puede dar más libertad.

No se persigue tanto el que sepan programar como el que entiendan lo que hay detrás de un programa o un juego, o cómo funciona un ordenador en general y que sean conscientes del esfuerzo que conlleva el hacer un programa y sepan apreciarlo. El año pasado realizamos una actividad en la que programábamos un videojuego y realizábamos los gráficos y todo desde cero, y los chicos al final de la misma comentaban que no se iban a volver a quejar jamás de los gráficos de un videojuego después de comprobar el trabajo que lleva hacer unos sencillos gráficos en 2D.

3.6.2 ¿Siempre se trabaja con ordenadores o también realizas actividades sin ellos?

Sí, siempre intentamos utilizar tecnología en las actividades que realizamos, a fin de cuentas, es lo que tratamos enseñar, es nuestra rama.

Aunque en alguna actividad aislada sí que hemos hecho cosas sin ordenador, como cuando enseñábamos código binario con bolitas y papeles, o estos últimos Pilares, que en unos talleres de Etopia Kids hicimos un “traga-chicos” con recortables de papel y unos servos.

3.6.3 ¿Mandas deberes/trabajo para casa? En caso afirmativo, ¿con que frecuencia?

No, no mandamos deberes, pero si algunas actividades de las creativas se alargan mucho, se les puede decir que las terminen en casa, como con la clase de píxeles que decíamos antes, que tenían que diseñar unas naves y les dijimos que las terminaran en casa o sino usaríamos algunas que ya estuvieran hechas.

Por norma general si alguno de nuestros alumnos hace algo en casa es por propia iniciativa y nos lo enseña en la siguiente sesión para que comprobemos si lo ha hecho bien o mal y corregirle si es necesario.

3.6.4 ¿Haces exámenes pruebas de evaluación? En caso afirmativo, ¿con que frecuencia?

No, no hacemos exámenes y además no nos parecen bien. Se trata de una actividad extraescolar en la que prima más que los niños se lo pasen bien y aprendan mientras se divierten antes que aprendan muchos conceptos distintos.

En algunos centros sí que se nos pide que evaluemos a los alumnos, pero no solemos dar notas numéricas, sino que indicamos que por ejemplo el alumno se ha portado bien o si ha comprendido lo que se le ha intentado enseñar.

Tampoco es necesario el hacer una prueba escrita o un examen para evaluar a alguien. Un examen sirve principalmente para justificar una nota, pero si un alumno te viene normalmente a clase se observa perfectamente el desarrollo que va teniendo y si se entera o no de lo que se está enseñando, no necesitas hacerle un examen para comprobar cómo va porque ves su evolución día a día.

4 - Conceptos que enseñas y herramientas que usas

4.1 ¿Qué conceptos enseñas? (P. Ej., condicionales, bucles, funciones, etc.),

4.1.1 ¿Por qué?

Les enseñamos que pasos han de seguir para que un programa haga lo que se supone que tiene que hacer. El primer día seguramente ya te toca explicar lo que es una condicional, pero el objetivo no es tanto que aprenda lo que es una condicional en si como a que sepan usarla para desarrollar el programa que estamos haciendo y para futuros problemas.

En cursos más avanzados que están más centrados en la programación, con alumnos más mayores (de once o doce años en adelante) sí que se plantean las clases alrededor de cuestiones más técnicas, pero en clases para niños no.

Nos interesa que descubran lo que hay debajo de un programa, que puedan crear, pero no dando un contenido típico de qué es una condicional o qué es un bucle. Las clases no están centradas en la parte teórica, sino en adquirir un conocimiento de otro tipo de forma más lúdica. El fin no es aprender la tecnología sino ver que hay.

Ya en el primer día que trabajas con Scratch y planteas hacer un juego ya estas poniendo una condición, del estilo “si el muñeco se choca contra la pared que pierda

una vida o se dé la vuelta”. Es un aprendizaje súper fluido en el que no te estás dando cuenta de muchas de las cosas que estás aprendiendo, no estas siendo consciente de la cantidad de conceptos que estas usando y puedes usar y aplicar por tu cuenta.

Es establecer una base, para que cuando sean más mayores y les expliquen de forma más técnica lo que es un bucle o una condicional, se den cuenta de que es lo que ya usaban cuando programaban de pequeños robots o videojuegos.

Todas estas herramientas que están orientadas en principio para niños son también súper útiles para dar un primer contacto con la programación, se tenga la edad que se tenga. Por ejemplo, nuestro profesor de programación de la formación profesional, Luismi, no en nuestro curso, pero en el siguiente, los primeros diez o quince días de programación los impartió usando Scratch. Con ello consigues enseñar algoritmia, las herramientas y estructuras básicas de programación, estructura general de un programa, etc., de una forma muy sencilla.

Para gente que no ha tenido jamás contacto con la informática o la programación, el utilizar al principio alguna de estas herramientas puede resultar tremendamente útil, porque es todo más sencillo, más gráfico, más fácil de comprender para alguien que no está acostumbrado a ver líneas de código o no sabe directamente lo que es un “for” o un “if”.

4.1.2 ¿Hay algún otro (concepto/herramienta) que te gustaría usar?

Creo que está bien como está. Hablando sobre conceptos que les cuestan más asimilar, funciones es difícil que aprendan, pero por ejemplo las variables no les cuestan mucho.

Si le planteas una variable como todo aquello que cambia a lo largo del juego, como las vidas, los puntos, las estrellas que consigues, ya está, están familiarizados con esas ideas y lo usan y ya está. Aunque no entren a profundizar o entender realmente lo que es una variable, las pueden usar correctamente para programar un juego.

Hay que adaptarse a los niños porque ellos no tienen ni idea de lo que les estas contando, te tienes que rebajar a su edad, a su nivel y lenguaje, para que ellos entiendan lo que les dices. Por eso también lo de trabajar en parejas es muy bueno, porque puede que lo que estás explicando solo lo entienda uno de los niños, que se lo explica al otro usando su mismo lenguaje.

4.2 ¿Qué herramientas usas? (P. Ej. Bitbloq, Code.org, Scratch, etc.) 4.2.1 ¿Por qué?

Usamos Scratch y derivaciones del mismo para poder conectarlo con Arduino como S4A y Mblock por norma general.

Intentamos usar software libre que sea multiplataforma y que no dependa de una conexión a Internet. E intentamos que sea así porque en cada Centro Educativo o Colegio donde damos el curso nos podemos encontrar ordenadores con distintas distribuciones de Linux o Windows.

Además, puede ser que no haya conexión a Internet, o de haberla, puede que ésta sea lenta o haya micro cortes o se caiga directamente, y esto te parte

completamente una clase. No puedes arriesgarte a llegar un día al centro y que no haya Internet y por tanto no puedas dar la clase que ya tenías preparada de antemano.

Incluso cuando tenemos que usar herramientas como SculptGL (que es una herramienta de escultura digital abierta y disponible en GitHub, <https://github.com/stephomi/sculptgl>) para actividades de diseño y modelado, lo hacemos descargándola a los ordenadores.

También usamos herramienta de modelado porque generalmente al final del curso cada alumno tiene su propio proyecto y su propio robot o videojuego que puede necesitar de piezas diseñadas en 3D para funcionar/tener sentido, en cuyo caso las acabamos imprimiendo.

4.2.2 ¿Hay algún otro (concepto/herramienta) que te gustaría usar?

Como comentábamos antes, hay herramientas muy potentes pero que requieren o de una conexión constante a Internet o que solo funcionan para Windows o determinadas distribuciones de Linux y por tanto quedan descartadas. Con las que usamos actualmente estamos sobrados para las necesidades que tenemos.

5 - Opinión sobre Herramientas que analizo

5.1 Bitbloq

5.1.1 ¿Conoces esta herramienta? ¿Qué te parece?

Sí que nos suena, pero no hemos trabajado en nuestros cursos con ella. Hemos estado en algunos cursos de formación que daban los de BQ colaborando, y sí que creemos que es una buena herramienta y tiene cosas interesantes como los robots ya preparados del Sigue-líneas o el Zowi, pero tienes que morir al palo de comprar sus productos y kits para poder funcionar.

De hecho, nosotros tenemos la versión anterior de software libre, Bob, del robot Zowi que han sacado recientemente, en la que los de BQ basaron el diseño de Zowi.

Hay que reconocer que son kits bastante completos, con muchos componentes y sensores distintos, y los robots modelo son muy chulos, pero el precio de los mismos nos echa para atrás. Y también necesitas o una impresora 3D o directamente comprarle las piezas de los robots que tienen.

5.1.2 ¿Qué hechas en falta/Qué mejorarías?

Que se tratase de una herramienta libre que se pudiese usar sin tener que comprar sus productos.

5.2 Code.org

5.2.1 ¿Conoces esta herramienta? ¿Qué te parece?

Sí que la conocemos, pero no la usamos por lo que hemos hablado anteriormente de que intentamos evitar siempre que podemos herramientas que dependan de una conexión Internet para poder funcionar.

5.2.2 ¿Qué hechas en falta/Qué mejorarías?

Que se pudiese descargar y tener una aplicación de escritorio multiplataforma. De ser así le daríamos una oportunidad.

5.3 Scratch

5.3.1 ¿Conoces esta herramienta? ¿Qué te parece?

Como te hemos estado contando, es la que usamos a diario en nuestras clases de “Creatividad Tecnológica” y estamos muy contentos con ella.

5.3.2 ¿Qué hechas en falta/Qué mejorarías?

A nosotros nos gusta tal y como está, para trabajar con los chavales no hace falta más, pero sí que, comentándolo con otros profesores, y a nivel de programación, se les echa en falta el poder disponer y programar herencias.

6 - Recomendaciones, otros

6.1 Si tuvieses que recomendar una herramienta, ¿cuál sería? ¿Porqué?

La que usamos actualmente de Scratch 2 y los pequeños forks/adaptadores de S4A y Mblock para poder enlazarlo con Arduino y poder programar placas servos, etc.

6.2 ¿Alguna otra valoración sobre este tipo de formación?

Sobre la proporción de niños niñas en los cursos relacionados con la informática.

Conforme va aumentando la edad, las niñas se van decantando por otras actividades de perfil no tecnológico, ya sea por grupo de amistades o lo que sea, y el grupo hace fuerza hacia otro tipo de actividades y vas viendo una evolución, de modo que conforme va aumentando la edad va disminuyendo las chicas que tienes en las clases.

Creemos que debe ser cultural, porque al principio las que se apuntan les gusta un montón, son súper activas y dispuestas a hacer estas cosas.

(Raquel) El año pasado estuve dando charlas del Instituto de la Mujer y contaba mi caso de ser mujer y profesora de nuevas tecnologías y al final de las charlas todas las niñas asistentes solían decir que de mayor querían trabajar como ingenieras o científicas. Pero como que, a los doce años, parece que hay un chip que les hace cambiar de idea y pierden interés por las carreras relacionadas con las tecnologías.

Creo que es un problema que tenemos en nuestra sociedad y en nuestra cultura: que tiene asociada la idea que los trabajos relacionados con las STEM son más para chicos que para chicas. Me ha llegado a pasar que, al empezar a dar una clase a niños pequeños, me pregunten sorprendidos si yo, por ser mujer, voy a dar la clase de robótica. Son estereotipos que un niño pequeño no debería tener.

También me ha pasado que, en ocasiones, me han propuesto dar actividades o clases de robóticas orientadas a chicas, del tipo con temas o estereotipos que son más de chicas, cosas de ropas o de ese estilo. Y es algo que me niego a dar, no

tiene por qué haber una diferenciación de este tipo. Imaginar el caso opuesto, clases orientadas solo para niños, tampoco sería correcto porque no hay algo orientado a chicas o a chicos, son tecnologías, robótica y programación.

A la hora de decidir que kit de robótica comprar, hay que tener en cuenta, entre otros factores el tema de la electrónica.

Se podrían agrupar los kits como 3 grandes grupos en función de cómo plantean la electrónica de los componentes que contienen. Kits que vienen directamente con los leds tradicionales de dos patas, resistencias, servos y un protoboard donde realizar las conexiones (como por ejemplo el kit de Cooking hacks), kits como el de robótica de BQ, que te viene el cableado simplificado con tres pines ya unidos para solo conectar para la alimentación y la toma de tierra, o finalmente kits como los de Lego que son de tipo plug and play.

Todo depende de si se quiere eliminar esa parte o no, hay mucha gente que prefiere eliminarla para centrarse solo en la programación del robot, o en la programación del robot y su diseño.

Sobre STEM y educación en nuevas tecnologías

Dar clases de nuevas tecnologías, informática o robótica es algo que se puede decir que está en auge, está de moda, como lo de implantar de formas sistemática el bilingüismo en los colegios. Y creemos que con ambas áreas hay un problema, y es que depende del profesor el llevar a cabo estos cursos bilingües o enseñar nuevas tecnologías, y no todo el profesorado actual está preparado para ello.

Si el profesor no sabe nada sobre robótica o incluso le cuesta manejarse con un ordenador, salvo que se le dé una formación previa interna, va a ser imposible que imparta bien clases de este tipo, con los problemas que esto acarrea. Un caso opuesto pero que también se puede dar son profesores que sí tengan conocimientos como para dar clases de informática y nuevas tecnologías, pero el centro donde las imparte no les da libertad o equipamiento adecuado para llevarlas a cabo, enseñar informática sin ordenadores resulta complicado.

Cualquier técnica de enseñanza puede funcionar si la puede llevar a cabo el profesor, como con la disyuntiva de trabajar por proyectos o no. Si con trabajar por proyectos consigues alcanzar todos los objetivos, es positivo, porque los alumnos están aprendiendo a fin de cuentas y no te estás restringiendo a un temario existente y el modelo típico de contar teoría sobre algo para luego hacer problemas, pero si no se ajusta a tus alumnos o no logras que aprendan, mejor no utilizarlo.

Por ejemplo, tenemos una profesora que nos solicita ayuda de vez en cuando y plantea las clases de una forma muy distinta a la tradicional. Es profesora de plástica y hace unos días fuimos a ayudarla a montar un robot para que le ayudara en las clases, lo tenía pensado para que, con ese robot, los alumnos visitasen museos y explicase los cuadros que hay en ellos. Y quiere montar uno cada trimestre para cada sección de su asignatura, porque ve que es una herramienta que llama la atención y les gusta a los

alumnos y se aprovecha de ese interés para enseñarles otras cosas que sí que están más relacionados con el temario de su asignatura.

La forma de que funcione la educación es que los chavales a los que vas a educar se interesen por lo que les tienes que enseñar, la clave es encontrar una herramienta o una metodología o lo que sea para que se interesen por lo que les tienes que contar, puede ser tecnología puede ser lo que quieras, pero que les guste y lo sepa hacer.

Una herramienta, por sí sola, no mejora la clase. Por tener pizarras digitales o un robot en clase los profesores no van a dar mejor clase si el profesor no las sabe usar y no las sabe incorporar correctamente. El que marca la diferencia sigue siendo el profesor, que evidentemente, de cuantos más y mejores recursos disponga, mejor, pero si no sabe hacer un uso correcto de los mismos o simplemente que hacer con ellos, evidentemente, no sirven de nada.

En conclusión, sí, la educación STEM está muy bien siempre y cuando sepa llevarla a cabo correctamente.

6.3 ¿Sabes de algún otro profesor/centro/Academia que imparta formación de este tipo?

Avanza tiempo Libre, NiuBit, ConMasFuturo, Océano Atlántico.

5º Entrevista - Pablo Aliaga (Autónomo DLABS)

1 - Datos Personales

1.1 Nombre:

Pablo Aliaga Cremades.

1.2 Centro/Profesión:

Profesor Autónomo, he trabajado para Crom Developer, Etopia Kids, con la asociación Dlabs, de la que formo que soy ahora mismo el presidente, y con la Universidad entre otros.

1.3 Titulación:

Ingeniería Informática.

1.4 ¿Cuánto tiempo llevas impartiendo formación de este tipo?

No mucho tiempo, como un año aproximadamente, me he dedicado sobre todo a la formación para adultos, con niños, como te comento, llevo relativamente poco.

1.5 ¿Cómo/Porqué empezaste a dar formación de este tipo?

Porque ahora mismo sale bastante trabajo de esto y es bastante sencillo de llevar si tienes un mínimo de preparación y organización.

A mí me gusta la formación, y he estado relacionado con la formación en muchos aspectos, sobre todo en la universidad, he sido profesor de Universidad y todo eso, y tanto la formación como los niños se me dan bien.

1.6 ¿Has dado formación de este tipo en algún otro centro? ¿Cuáles?

En varios centros como te decía antes, Crom Developer, Etopia Kids... ya que soy profesor autónomo.

2 - Opinión sobre el tema

2.1 ¿Crees que es importante impartir este tipo de formación? ¿Por qué?

Creo que está bien, más que nada porque vayan aprendiendo un poco los intereses que pueden tener más que nada para que puedan decidir ellos por sí mismos. Es decir, enseñarles que la informática y las nuevas tecnologías son algo en lo que pueden trabajar en el futuro.

También por enseñar algo más que la informática tradicional o el estar usando el telefonéo móvil y no saber lo que hay detrás. Que enseñándoles un poco de robótica o informática se den cuenta todo lo que hay detrás de ello, que no solo se queden con “qué majo el juguete que se mueve hacia un lado u otro”, sino que sepan porqué lo hace, o al menos sean conscientes que hay una programación, una tecnología detrás.

Hay niños que se les ve enseguida que no tienen ningún interés, “oh, es un juguete” y ya está, pero hay otros niños que tienen curiosidad de saber cómo está funcionando, por qué se mueve ahora y no en otro momento, etc.

2.2 ¿A partir de que edades? ¿Por qué?

Esto para los niños puede ser un juego, cuando doy formación a niños lo intento enfocar como si fuera un juego, tiene que ser un juego para ellos y aprender jugando y si tiene que ser un juego, tiene que ser divertido, y por lo tanto puede enseñarse a cualquier edad siempre que no se pierda ese enfoque.

También hay que tener en cuenta las capacidades de un niño, por ejemplo, cuatro, cinco años los veo muy jóvenes, creo que deberían de estar con sus cosas, pelotas y juegos más de psicomotricidad y ya.

Como es normal un niño tan pequeño, de cuatro años o así, no te va a saber mover un ratón ni mucho menos escribir en un teclado, y va a estar más interesado en una pelota que en lo que se ve en la pantalla del ordenador.

Con seis años empezar a introducir algún pequeño juego, algún concepto básico como una condicional, etc. lo veo mejor, así que diría de empezar alrededor de esa edad.

3 - Cursos/Clases

3.0 ¿Cuántos cursos o clases distintas impartes?

Normalmente son actividades extra escolares o pequeños talleres, cursos completos daría a gente más mayor, centrados en conceptos más avanzados de robótica e informática, que es lo que más me gusta a mí. Aunque también es cierto que los niños son súper agradecidos cuando les enseñas algo.

En verano lo que he hecho son las colonias de Etopia, y eso sí que era más formación de robótica para niños todas las semanas unas horas. En total, cursos distintos, más de 3 o 4 cursos, son talleres que preparo cada vez que se me pide.

3.1 ¿Cuántos alumnos tienes por clase? ¿Consideras que es un número de alumnos adecuado o debería ser mayor/menor?

En las colonias teníamos unos veinticuatro alumnos por clase y estábamos tres monitores para hacer las cosas.

Desde luego es recomendable tener menos niños, cuando tienes veinticuatro alumnos, dos o más te van a salir revoltosos y se te van a llevar al resto de la clase, uno llama uno, este llama a otro y así, es muy complicado llevar un ritmo de clase normal porque no das a basto.

Con doce, uno o dos pueden ser revoltosos, pero ya es más fácil controlarlo. Creo que con doce alumnos un profesor sería mejor, ya puedes dar un trato más personal y atender mejor las dudas que les van surgiendo a cada uno de ellos. Sobre

todo, en cursos de robótica para niños en los que tienes actividades del tipo, “enchufar el cable rojo aquí, ahora el azul en este otro sitio” y así, por mucho que les expliques que hace cada cable y como se conecta, siempre va a haber más de uno que no le funciona o no lo ha entendido y vas a tener que solucionárselo uno por uno poniéndoles los cables correctamente.

3.2 ¿Qué perfil de alumno tienes? (edad, motivación, conocimientos previos, etc.)

Me vienen niños de todo tipo, aunque normalmente sí que el niño sabe algo o ya ha hecho alguna actividad taller o curso similar en el colegio o en algún lado.

En otras ocasiones son los padres que no saben dónde meterlos y te los traen a clase, “Yo quiero que mi niño aprenda esto”, “Yo quiero que mi niño sea ingeniero”, igual que otros que apuntan a sus padres a fútbol porque quieren que sus hijos sean futbolistas.

En cuanto a las edades, en las colonias teníamos a partir de seis años hasta catorce.

3.3 ¿Cuánto duran las clases? ¿Consideras que es un tiempo adecuado o debería ser mayor/menor?

Las Colonias duraban todo el mañana, de 9:00 a 14:00 aproximadamente, haciendo un descanso a mitad, es decir, unas cinco horas.

Respecto a si era un tiempo adecuado, todas las actividades estaban planteadas como juegos, entonces no tenía sentido plantear unos tiempos concretos. Por ejemplo, vamos a estar jugando a hacer unas piezas y el tiempo que se estaba con ello era hasta que se acababan de hacer las piezas.

El tiempo que se estaba con una actividad u otra venía definido por la propia actividad en sí, no por unos horarios de “vamos a estar cincuenta minutos, paramos, venga otra vez cincuenta minutos, etc.”.

No era como el modelo tradicional de clase: Teníamos a los niños sentados, pero no solos, sino que en grupo; no hay una pizarra, hay una pared de cristal donde pintas lo que se te va ocurriendo para enseñarles; no es el espacio tradicional de una clase, hay mesas y sillas, en eso es en lo único que se perecen.

3.4 ¿Cuántas clases por semana? ¿Consideras que es un número de días adecuado o debería ser mayor/menor?

Todos los días mientras duraban las colonias. Y en total se estaban cinco semanas, del veintisiete de junio al veintinueve de Julio.

3.5 ¿Los alumnos disponen de material propio para las clases (por ejemplo, se traen sus propios portátiles) o los proporciona el centro?

Lo proporcionábamos nosotros, los ordenadores los ponía Etopia, con Linux todos ellos, y el material extra necesario, que eran unos pequeños brazos robóticos de plástico, también los habían comprado en Etopía.

Los ordenadores eran todos sobremesas con Linux, y en cada uno trabajaban dos alumnos, que también compartían un único brazo robótico, aunque en algunas actividades los agrupábamos en grupos más grandes, de cuatro o seis niños.

Esta organización era fruto del material disponible, no los poníamos a trabajar en parejas de forma intencionada. Aunque creo que grupos de dos sí que se pueden tener dando las clases, grupos de tres no, porque siempre va haber uno que se escaquea o no le dejan hacer tanto y acaba acarreando problemas porque al final de la actividad es eso, o no ha hecho anda porque no ha querido o porque no le han dejado, y en cualquier caso va a quejarse, como es normal.

3.6 Dinámica de clase

3.6.1 ¿Cómo son las clases, explicas todo con ejemplos y luego das tiempo para que se trabajen en ejercicios, se van haciendo todos a la vez, etc.?

Primero les contábamos un poco en lo que consistía el juego, lo que iban a aprender, por ejemplo: con el brazo acrobático lo primero que hacíamos era enseñarles el brazo robótico en movimiento.

Lo primero es enseñarles el objetivo final, lo que van a tener ellos cuando acaben la actividad, y luego desarrollarla poco a poco. Una vez que ya tienes el objetivo ya sabes cómo llegar a ello.

Con el planteamiento opuesto, el de dar solo instrucciones, no motiva tanto a los alumnos, “Venga ahora vamos a montar esta pinza, ahora esta pieza”, no resulta muy motivador no saber el que estás haciendo y porque lo estás haciendo (qué además son los conceptos que precisamente queremos enseñar, los porqués), te estas limitando a dar unas órdenes para que ellos la sigan como robots. Si ya saben lo que están haciendo, lo enfocan más motivados porque han visto ese resultado final y quieren alcanzarlo.

3.6.2 ¿Siempre se trabaja con ordenadores o también realizas actividades sin ellos?

No siempre trabajábamos con ordenadores, también hacíamos actividades sin ellos de papel y lápiz. Por ejemplo, había que hacer una secuencia de movimientos con el brazo robótico, se les pedía que primero lo hiciesen a papel y luego lo pasaran al ordenador.

Con esto conseguir que hicieran una especie de diagrama de flujo de movimientos del robot antes que lo programasen, que hicieran un poco de algoritmia en papel y luego aplicarla en el ordenador.

3.6.3 ¿Mandas deberes/trabajo para casa?, en caso afirmativo, ¿con que frecuencia?

No, no mandábamos nada para casa.

3.6.4 ¿Haces exámenes pruebas de evaluación?, en caso afirmativo, ¿con que frecuencia?

El objetivo es que todos hayan aprendido, siempre ves quien aprende y quien no aprende, pero vamos, el objetivo último es que aprendan divirtiéndose, son actividades lúdico-prácticas. Que aprendan no por la parte de me siento y escucho, sino la de me divierto y aprendo.

Por eso no hacíamos exámenes ni pruebas de evaluación.

4 - Conceptos que enseñas y herramientas que usas

4.1 ¿Qué conceptos enseñas? (P. Ej., condicionales, bucles, funciones, etc.),

4.1.1 ¿Por qué?

No les pueden plantear de primeras una condición. Para enseñar lo que es una condicional lo tienes que hacer a base de juegos y actividades que proponemos en clase, no como algo estrictamente teórico.

También puedes usar videojuegos normales, por ejemplo, muchos niños juegan o han jugado al Buscaminas, que es un juego estrictamente condicional, pero claro, muchos de ellos no son conscientes de esto, o al menos, aunque sepan jugar, no enlazan con que son condicionales.

Alguna vez, para descansar de lo que estuviéramos haciendo o si se ponían especialmente revoltosos, dábamos un poco de suelta del tipo: “venga niños, 3 minutos de Buscaminas”. Es un juego que se puede resolver en segundos si sabes las reglas, que son a fin de cuentas condiciones, y está muy bien para así ponerles ejemplos prácticos de condicionales a los niños y que vean que no hay minas porque sí, sino que los números que aparecen son los que indican cuantas minas y donde están, etc.

No teníamos una guía docente estricta, aunque teníamos unos objetivos generales y específicos marcados, lo que constituye un proyecto educativo.

Objetivo general, ¿Qué es lo que queremos hacer/aprender? Pues robótica, los movimientos de un robot. De ese objetivo general se derivan ya objetivos específicos: “como se mueve un servo”, “como se programa un servo”, “como es por dentro un servo”, “como se construye un servo”, etc.,

Luego dividíamos estos objetivos específicos por áreas de conocimiento, conocimientos de robótica, conocimientos de informática, conocimientos de construcción, etc. De ahí ya sacar talleres o actividades para enseñar esos objetivos específicos es casi directo.

Por ejemplo, si queremos que aprendan servos, se diseña una actividad de cómo es un servo por dentro y de para qué sirve o como se programaría. Nos centramos sobre todo en que aprendan el porqué de las cosas.

No hay que olvidar que son niños, no les puedes enseñar un bucle o una condición como se lo plantearíamos a un adulto, siempre hay que usar ejemplos con los que estén familiarizados y un lenguaje que entiendan, ponerles ejemplos del tipo “Esto de las condiciones es como cuando la señorita en clase os dice, si no te callas copiaras 100 veces algo”.

4.1.2 ¿Hay algún otro (concepto/herramienta) que te gustaría usar?

Ninguno que se me ocurra ahora mismo.

4.2 ¿Qué herramientas usas? (P. Ej. Bitbloq, Code.org, Scratch, etc.) 4.2.1 ¿Por qué?

Trabajábamos sobre todo con Snap for Arduino, S4A (<http://snap4arduino.org/>), una derivación/adaptación de Scratch para poder trabajar también con Arduino y con todos sus componentes, ya que Scratch solo te deja mover 3 servos como mucho y el S4A te da una compatibilidad total con placas de Arduino y sus componentes.

Aparte de eso, algún recurso de video para que viesen alguna cosa o algún ejemplo de robots o cosas que se pueden llegar a hacer con Arduino.

También en alguna ocasión hemos usado otras herramientas de programación por bloques como Bitbloq o Code.org, pero muy poco.

4.2.2 ¿Hay algún otro (concepto/herramienta) que te gustaría usar?

Estoy contento con S4A, me permite enseñar todo lo que necesito.

5 - Opinión sobre Herramientas que analizo

5.1 Bitbloq

5.1.1 ¿Conoces esta herramienta? ¿Qué te parece?

Sí que la conozco, pero la he usado poco.

5.1.2 ¿Qué hechas en falta/Qué mejorarías?

Si la he usado poco es precisamente porque no me termina de convencer el depender de una plataforma online para hacer una clase. Estas dependiendo siempre de un entorno remoto al que no tienes que porque poder acceder siempre, si tienes una clase programada con ese recurso y por lo que sea no está disponible, te quedas sin clase.

5.2 Code.org

5.2.1 ¿Conoces esta herramienta? ¿Qué te parece?

Sí, pero no la he usado mucho.

5.2.2 ¿Qué hechas en falta/Qué mejorarías?

Le encuentro los mismos problemas que a Bitbloq, que sea una plataforma web.

5.3 Scratch

5.3.1 ¿Conoces esta herramienta? ¿Qué te parece?

Sí que lo conozco, y me parece bien, pero prefiero usar S4A.

5.3.2 ¿Qué hechas en falta/Qué mejorarías?

Que funcionara como funciona S4A.

6 - Recomendaciones, otros

6.1 Si tuvieses que recomendar una herramienta, ¿cuál sería? ¿Porqué?

Obviamente recomendaría S4A, que aporta todas las posibilidades que tienes de base con Scratch y que además te permite comunicarte perfectamente con Arduino y sus componentes, tener librerías propias, trabajar mejor en general, acceder a todos los pines de una manera más fácil.

Algo que le echaría de menos al S4A es que no te permite cargar programas en la placa de Arduino, de modo que el robot tiene que ser conectado por USB todo el rato al ordenador para que haga lo que le has programado hacer.

6.2 ¿Alguna otra valoración sobre este tipo de formación?

Sobre el preparar las clases y el lenguaje y la interacción con los niños

Hay que tener las clases preparadas de ante-mano, al principio tuvimos problemas con las Colonias porque no se hizo correctamente. No puedes llegar, por mucho que sepas, sin nada preparado: son niños y la habilidad de un adulto y lo mucho que sepas de programación no te va a servir para una respuesta a un niño o para mantenerlos atentos e interesados un buen rato.

La clase tiene que estar muy preparada y saber lo que tienes que hacer en cada momento, cómo son las actividades o juegos que vas a realizar, saber el guion de la misma, etc., no porque necesariamente luego lo vayas a seguir a pies juntillas, pero si tener algo preparado previamente y saber por dónde salir e ir tirando.

Y, sobre todo, saber que son niños y que no tiene por qué acabar siendo “cerebritos” todos, y también tener en cuenta el lenguaje que usas con ellos, en ningún momento son deberes o son problemas: “esto es divertido y son juegos o actividades”. Si no te va a gustar no lo hagas, porque si lo haces, pero no te gusta te vas a enfurruñar y vas a distorsionar la clase y al profesor. Hay que saber cómo planteárselo todo para que les resulte atractivo y divertido.

No hay que olvidarse que casi siempre estamos hablando de actividades extra escolares, o talleres, o colonias, o actividades a las que se apuntan por iniciativa

suya (o en ocasiones, de sus padres). Normalmente estas clases no forman parte de la guía docente o los planes de estudios de sus colegios y por lo tanto no son de obligada asistencia, así que han de ser atractivas siempre.

Sobre el pensamiento Computacional

Lo básico es resolver un problema, un problema normalmente cuando nos lo enseñan en matemáticas son del tipo, este es el problema, esto son los pasos para resolverlo y ya está; pero no te dan ninguno problema semi-complejo que lo puedas dividir tú, no te animan a aplicar el pensamiento computacional y dividir un problema en partes más pequeñas, etc., etc.

Debería animarse más a tener un pensamiento crítico y a que se haga un análisis previo de los problemas que se plantean, más que este es el tipo de problema que te tengo que enseñar y se soluciona aplicando estas fórmulas o estos métodos en este orden.

6.3 ¿Sabes de algún otro profesor/centro/Academia que imparta formación de este tipo?

Avanza tiempo Libre, NiuBit, ConMasFuturo, Océano Atlántico.

6º Entrevista - Beatriz García (Niubit)

1 - Datos Personales

1.1 Nombre:

Beatriz García.

1.2 Centro/Profesión:

Profesora y co-fundadora de la empresa Niubit (<http://www.niubit.es/>), empresa de formación en nuevas tecnologías, cursos de Robótica, Programación e Impresión 3D.

1.3 Titulación:

Ingeniería Informática de Sistemas.

1.4 ¿Cuánto tiempo llevas impartiendo formación de este tipo?

Fundamos Niubit hace ya cuatro años, sobre finales de 2013 es cuando arrancamos a hacer talleres de programación para niños con fines lúdicos ya que creemos mucho en la gamificación y en hacer atractivas las actividades para que los niños aprendan.

Así que dando clases a niños llevaremos unos 4 años.

1.5 ¿Cómo/Porqué empezaste a dar formación de este tipo?

Ninguno de los dos que arrancamos la empresa habíamos trabajado en el sector de la enseñanza, siempre en la informática, lo más parecido que habíamos hecho era formación a usuarios dentro del ámbito de la empresa cuando habíamos desarrollado alguna aplicación. Vimos que había mercado y decidimos arrancar la academia.

Nosotros empezamos un poco en este mundo con los kits de Lego WeDO, que son los que llevan los motores, sensores y demás para el tema de programación. Son unos kits que están muy chulos porque ya aparte de proporcionarte unos modelos, luego también puedes montar tus propios robots y tus propios diseños. Además, puedes usar las herramientas de Lego para la programación o utilizar Scratch, que nos parece un poco más interesante porque la de lego nos parece un poco más “baby”.

Estos kits de Lego llevan un Hub que lo pinchas en el ordenador y es donde vuelcas la programación que han hecho previamente, además de varios sensores (como sensores de inclinación y de distancia) y motores que te permiten elaborar pequeños videojuegos, como un de un barquito que suba y baja, etc. Es una manera muy lúdica de iniciarles en el mundo de la robótica y vayan descubriendo poco a poco las partes de un robot, su programación y así.

1.6 ¿Has dado formación de este tipo en algún otro centro? ¿Cuáles?

Siempre ha sido con nuestra empresa NiuBit, aunque si hemos trabajado en colegios y para otros centros, peros siempre sub contratados por estos colegios o centros haciendo alguna actividad extra escolar de informática o robótica.

2 - Opinión sobre el tema

2.1 ¿Crees que es importante impartir este tipo de formación? ¿Por qué?

Cuando nosotros empezamos con este tipo de formaciones aún no se habían puesto del todo de moda, aquí arrancamos con los talleres los fines de semana, y a mi compañero en los colegios le demandan que hiciera una extra escolar de informática, se negó a hacer una extraescolar al uso de informática y planteo cambiarla por una de Robótica, al colegio le pareció bien y arrancamos ya con eso.

Realmente en estos dos últimos años se ha puesto de moda, Robótica es la palabra de moda, pero es que creo que es un tipo de formación muy importante. A lo mejor no el concepto de Robótica que a todo el mundo se le ha metido en la cabeza, que es como la extra escolar de moda, sino como a nivel de programación.

Yo creo que realmente la robótica y la programación deberían de entrar en plan de estudios y ser una asignatura obligatoria más, y creo que acabará llegando. Creo que es algo muy importante y que ayuda mucho a los niños a estructurarse, organizarse mejor.

Realmente en el día de mañana es muy probable que vayan a necesitar programar porque prácticamente cualquier elemento de tu vida cotidiana dentro de diez años, ósea, cuando los niños que nosotros tenemos aquí sean adultos, va a necesitar de esos conceptos de programación. Ya solo con el internet de las cosas, te vas a comprar un frigorífico y vas a tener que configurarlo, y aunque se hacen esfuerzos para que todo sea lo más sencillo e intuitivo posible, el saber cómo es un programa o el saber programar te va ayudar enormemente.

Aunque todos los nuevos sistemas operativos y herramientas se piensen y se hagan más fáciles, hay ciertas cosas que te están obligando a aprender para poder usarlos correctamente. Por ejemplo, hace poco estuvimos dando formación a adultos y había algunos que comentaban que necesitan el ordenador, aunque no quisieran trabajar con él, si ya es necesario ahora, cuando los niños de hoy sean adultos, lo será aún más.

El mundo tiende a que cada vez sea más necesario entender y utilizar tecnologías. Para estas generaciones que vienen que están ahora en primaria secundaria e infantil, es necesario que aprendan a manejarse con ordenadores, porque en su futuro lo van a necesitar.

2.2 ¿A partir de que edades? ¿Por qué?

Se debería empezar en primero de primaria (seis años), que, aunque son todavía pequeños, son muy receptivos.

En edades más tempranas, como en infantil, a lo mejor sí que hay algunos conceptos de robótica que podrían ir introduciéndose y podrían llegar a entender, pero tiene otra complejidad añadida porque los niños aún no saben leer ni escribir. Sería otro tipo de formación, de juego, el que tendrías que plantearles para introducirles ese tipo de conceptos y desde luego sin usar el ordenador. Todo se puede hacer realmente, en algún centro con niños pequeños hemos hecho talleres y actividades de programación

desenchufada en que los niños tienen que trabajar dibujando, haciendo actividades similares a resolver problemas con algoritmia.

Ya están empezando a sacar muchos materiales y kits para infantil, como esos pequeños juegos que consisten en ir conectando fichas de madera o pequeños robots que son más intuitivos y que sí que los pueden manejar. Pero vamos, yo arrancarí en primaria, creo que la inversión en tiempo y material que se tendría que hacer para poder llevarlo a infantil no compensa y en primera están en una buena edad para empezar.

3 - Cursos/Clases

3.0 ¿Cuántos cursos o clases distintas impartes?

Nosotros hacemos una única actividad que es un combinado de muchas, programación robótica, diseño 3D, etc.

Lo que vamos haciendo es centralizar todas esas actividades distintas en un único curso que variarla de contenidos y dificultad en función a quien vaya enfocado, o si es para talleres de finde semana.

Es una única actividad genérica de robótica en la que tendríamos los kits de lego, con la programación nos ayudamos de las propias aplicaciones de programación con Lego, como Lego MindStorms, o también con Scratch. Para el diseño 3D utilizamos Minecraft, que nos sirve perfectamente para enseñar posicionamiento de un robot en las tres dimensiones, hacer pruebas de cómo se va a mover el robot más adelante, etc., ya que son pequeños como para enseñárseles conceptos como eje de corneadas y poder centrar las 3 dimensiones.

Sobre los cursos, tendríamos cuatro cursos que inicialmente están divididos por edades. El primero correspondiendo a alumnos de primaria de primero a tercero, el segundo para alumnos de primaria de cuarto a sexto, el tercero para alumnos que ya hayan participado en los dos anteriores para aún no hayan llegado a secundaria (es decir, alumnos que ya tienen conocimientos de programación y robótica y que llevan uno o más años trabajando con nosotros) y un cuarto y último para todos los alumnos de secundaria o incluso de bachiller.

Pero como te decía, la dinámica de trabajo y herramientas son las mismas para los cuatro cursos, solo varía la intensidad y dificultad de las actividades que se les plantean.

En algunas ocasiones para algunos talleres o actividades extra escolares, tenemos otras dinámicas, como lo que te comentaba antes de la programación desenchufada o por ejemplo una actividad que realizamos hace poco, que consistía llevar una impresora 3D a un colegio y enseñarles su funcionamiento y que los niños con papel fueran programando los movimientos que creía que tenía que hacer la impresora para imprimir una determinada figura. Queremos un imprimir este dibujo, el primer paso es este, como se va a tener que mover la impresora en el eje de coordenadas para acabar imprimiendo correctamente lo planteado, etc.

3.1 ¿Cuántos alumnos tienes por clase? ¿Consideras que es un número de alumnos adecuado o debería ser mayor/menor?

Como máximo tenemos catorce niños por clase, aunque en los cursos para los más pequeños (el curso uno para alumnos de primero a tercero de primaria) intentamos tener aún menos alumnos por cada grupo, nueve o diez como máximo, ya que requieren de mayor atención.

Este límite es por el poder atender a todos ellos y el manejo de los trabajos en grupo o en equipos, si creces mucho más en el número de niños llega un momento que no das a basto.

Sobre límites mínimos, sí que nos gusta contar con al menos cuatro o seis alumnos por clase, ya que hay muchas actividades que se realizan por parejas o por tríos. Aunque en alguna ocasión hemos trabajado con grupos más pequeños de dos o tres alumnos, simplemente porque no se han apuntado más niños a ese curso o actividad en concreto.

3.2 ¿Qué perfil de alumno tienes? (edad, motivación, conocimientos previos, etc.)

Edades, como te comentaba, a partir de seis años (primero de primaria) hasta bachiller. Trabajamos con todo el espectro entre esos cursos.

Respecto a motivaciones, la verdad es que depende, tenemos tanto casos como el del niño es el que dice a sus padres que le gustan estas cosas y que quiere apuntarse a actividades de este tipo, como casos que son los padres los que directamente nos traen al chico porque quiere que su hijo aprende informática y nuevas tecnologías. En cualquier de los dos casos, normalmente, les suele gustar y suelen repetir ya porque el niño quiere repetir. También está el caso de niños que quieren venir a hacer robótica pero no saben exactamente lo que es o qué van a hacer, solo que les gusta el nombre o han odio hablar de robótica y quieren probar. O niños porque viene su amigo y quieren probar, etc.

Pero en general los alumnos nuevos son porque los apuntan los padres, si eso los niños prueban en algún taller o actividad de verano, o que han estado en alguna actividad de robótica en el colegio o en Etopia Kids y viene porque quieren hacerlo como actividad extra escolar aquí, que han oído campanas y quieren probar.

3.3 ¿Cuánto duran las clases? ¿Consideras que es un tiempo adecuado o debería ser mayor/menor?

Depende tenemos clases organizadas de una hora o hora y media, las que realizamos en el centro son todas de hora y media. Las de una hora son tan cortas normalmente porque es la disponibilidad de tiempo que te ofrece el colegio, al ser una actividad extra escolar.

A nosotros nos gusta más el formato de hora y media porque te permite llegar a todo, plantear la clase, realizar la actividad que planetas y llegar a una conclusión. Además, esto nos permite hacer un cambio de actividad en mitad de la clase, que es algo que nos gusta mucho. Básicamente se trata que no estén toda la sesión trabajando en lo

mismo o haciendo las mismas cosas, de esta manera romper la rutina y en ningún momento llegan a hartarse de estar trabajando con lo mismo. Es una manera de mantenerlos atentos y motivados tanto a los alumnos como al propio profesor.

En las clases, normalmente, solemos tener una primera parte de montaje del robot y una segunda de programación, se hace en ese orden porque los niños no pueden esperar a ver el resultado de lo que has programado, no pueden probarlo sino tiene montado ya el robot previamente.

3.4 ¿Cuántas clases por semana? ¿Consideras que es un número de días adecuado o debería ser mayor/menor?

Normalmente una sesión por semana, aunque en algún grupo se llegan a tener dos.

Por eso también lo de que preferimos son clases de hora media antes que solo una hora, porque se aprovecha mucho más el tiempo en una única sesión de una hora y media, que, en dos sesiones de una hora, aunque sea más tiempo. Se pierde mucho tiempo cada vez que tienes que arrancar la actividad y volver a poner a los alumnos en contexto.

Aunque todas estas cosas en las que se pierde el tiempo cuando se arranca o se finaliza la clase, como encender y Apgar correctamente los ordenadores, que se metan en la plataforma que vamos a usar, que se aprendan las contraseñas y sean responsables de su trabajo, va dentro de la formación.

De hecho, muchos padres cuando llegan con sus hijos nos preguntan “¿Pero van a aprender informática? ¿Van a aprender a manejar ordenadores?” Hay que contestarles que no, que no enseñamos informática como pueden entender ellos, de utilizar el Excel, etc., pero sí conceptos de programación, robótica y en general a saber manejarse con un ordenador, saber defenderse con la tecnología.

Además, que los niños son nativos y realmente les cuesta muy poco aprender estas cosas, el otro día me venía una niña con su madre y la madre nos pidió que le enseñáramos a manejar las Libre Office, que se las acababan de instalar en casa y tenían problemas. Nos pusimos con la niña solo y, paso a paso, en diez minutos, le habíamos aclarado ya todas las dudas que tenía, y no porque se lo explicáramos, “mira, aquí se guarda, aquí se hace tal” sino que le planteábamos, “como crees que se abre una nueva hoja de Excel o cómo crees que se guarda” y ella sola lo sacaba, solo había que darle pequeñas pistas, como “acuérdate que con el botón derecho salen muchas más opciones y así”. Los niños están ya acostumbrados a manejarse con las herramientas que usamos y con internet y todas las aplicaciones que hay ahora son muy intuitivas y están muy estandarizadas, el símbolo de guardar siempre va a ser el mismo, etc.

3.5 ¿Los alumnos disponen de material propio para las clases (por ejemplo, se traen sus propios portátiles) o los proporciona el centro?

El material lo proporcionamos siempre nosotros, todo lo que se refiere a los kits de robótica.

En lo que se refiere a ordenadores, aunque intentamos usar de los que disponen en el colegio donde vamos, en muchas ocasiones nos ha tocado llevar nuestros propios equipos porque los que tiene el colegio no dan, eran equipos muy antiguos o desactualizados.

3.6 Dinámica de clase

3.6.1 ¿Cómo son las clases, explicas todo con ejemplos y luego das tiempo para que se trabajen en ejercicios, se van haciendo todos a la vez, etc.?

La gamificación, el plantear la enseñanza como un juego, a los niños les encanta y están mucho más dispuestos a hacer cualquier actividad que les plantees. Aparte tenemos hijos que están en el rango de edades que trabajamos, así que en casa vamos probando distintas herramientas y métodos primero con ellos y lo que nos convence lo traemos al aula.

Cada niño tiene un nivel de aprendizaje y unas capacidades, pero mientras este receptivo y motivado ya está bien, porque se le va a poder acabar enseñando lo que quieres, y la dinámica es esa, hacer las cosas divertidas para tenerlos motivados. Lo importante es que el niño cuando salga de clase les diga a sus padres que se lo ha pasado bien, o que sean los propios padres que te lo comenten cuando te los encuentres por la calle, porque es garantía de que el niño está motivado, y por tanto receptivo. Los conceptos se van a entrar sin que se enteren porque se lo están pasando bien y quieren más, se lo toman como un juego y que están aprendiendo las normas de un juego, no necesariamente que están aprendiendo bucles o condicionales técnicamente hablando.

En general, se les plantea el montar un modelo que luego tienen que programar. Por ejemplo, el otro día estuvieron montando bicicletas y moto sierras para ver el tema de engranajes y demás.

Les planteas una serie de modelos, les das la explicación y luego les vas sacando preguntas, como “¿Cómo tenemos que montarlo para que vaya más rápido?” “¿Cómo tenemos que montarlo para que haga esto otro?”. El tema de preguntas se va haciendo a lo largo de toda la clase para desafiarles y comprobar si efectivamente están entendiendo los conceptos que se les están enseñando.

También se va premiando conforme van respondiendo correctamente o si ayudan a sus compañeros. Para el este tema de premiar los buenos comportamientos, colaboran en clase, alcanzan ciertos objetivos o ayudan a sus compañeros, etc. utilizamos varias herramientas y sistemas.

Con los pequeños utilizamos un sistema de puntuación con “Biticoins”, que son como nuestra moneda, ellos van acumulando monedas que al final de las clases pueden intercambiar por premios. Estos premios suelen ser figuras o pequeños juegos o piezas que tengamos fabricados con la impresora 3D, por ejemplo, ahora estamos con Pokemon y se llevan Pokemon que hayamos impreso. Así les enseñamos también como se imprime una pieza, como se diseña y luego al final se la acaban llevando a casa. Con esto entran en la dinámica de lo quiero hacer todo muy bien porque me quiero llevar esta pieza o este Pokemon que se está imprimiendo.

Con los mayores utilizamos una herramienta web que se llama ClassCraft, que es como si fuera un juego de rol, hay equipos, hay distintas clases, guerreros curanderos, etc. Van subiendo de nivel conforme logran objetivos, conforme suben van mejorando su personaje, comprando mascotas, equipo, se les puede dar monedas para premiarles por las actividades que han hecho a lo largo de la clase, etc. Además, los padres que quieran darse de alta también pueden interactuar y premiar a sus hijos cuando se han portado bien o si hacen alguna actividad, que también es una manera de que los padres estén al tanto de los progresos de sus hijos en las clases y que además participen y se involucren en las mismas.

Esta herramienta, realmente, se utiliza solo cinco minutos en la clase, pero es algo que les motiva, porque es entrar al ClassCraft y vean el evento del día, que lo puede haber preparado el profesor de antemano para que tenga que ver con la actividad que van a hacer, “pues ha habido una catástrofe y perdemos todos 5000 puntos y para recuperarlos hay que llegar a ciertos objetivos”, además, con actividades por grupos, unos se pueden ayudar a otros, y conseguir que se premien entre ellos. Se favorece que trabajen en equipo y se ayuden unos a otros haciendo uso de la app.

Como resumen, tenemos la experiencia de que cuando trabajamos con sistemas de puntos, no se genera competencia, con los “Biticoins”, por ejemplo, cuando vas a darlos por haber completado una actividad en clase, das a todos los alumnos un punto y quizás a algún alumno que haya sobresalido en la misma le das uno más, pero todos tienen puntos y todos van progresando juntos.

3.6.2 ¿Siempre se trabaja con ordenadores o también realizas actividades sin ellos?

Se varían entre actividades de manejo del modelo con lego, del robot y la programación del mismo.

El montaje se hace por parejas mientras que la programación se hace de forma individual, cada alumno con su ordenador.

También, en otras actividades, como cuando trabajamos con Minecraft, podemos a llegar hacer equipos de tres o cuatro alumnos.

Los equipos no son fijos, van variando a lo largo de las clases y actividades, se intenta rotar a los alumnos para que no siempre haya los mismos grupos de amigos que se ponen juntos.

3.6.3 ¿Mandas deberes/trabajo para casa?, en caso afirmativo, ¿con que frecuencia?

No, no mandamos nada para casa. Pero hay una plataforma online donde se cuelgan sus fichas y los trabajos que se van realizando, y donde también disponen de varios juegos y más actividades que el alumno puede completar si así lo desea desde casa. Ya que cada alumno tiene un usuario y contraseña para acceder a dicha plataforma

3.6.4 ¿Haces exámenes pruebas de evaluación?, en caso afirmativo, ¿con que frecuencia?

No, no hacemos exámenes.

4 - Conceptos que enseñas y herramientas que usas

4.1 ¿Qué conceptos enseñas? (P. Ej., condicionales, bucles, funciones, etc.),

4.1.1 ¿Por qué?

Depende de con que grupo estés trabajando, si son más pequeños o más mayores y el desarrollo que va teniendo mismo del grupo. Hay grupos que les cuesta avanzar más que a otros.

Se les va enseñando todos los conceptos de programación, bucles, variables, etc. Se va avanzando al nivel de los niños y se les intentan enseñar todos los conceptos de programación.

Trabajamos con los más pequeños con Scratch, mientras que con los más mayores se programa con Arduino usando el IDE de Arduino (y los mayores programan por tanto a mano escribiendo las instrucciones en vez de programación por bloques), también dentro del Minecraft trabajamos con LUA para elementos constructivos.

Planificamos por meses en los que decidimos que conceptos vamos intentar enseñar a cada grupo, y hacemos una programación de las sesiones para todo ese mes. Un solo concepto de promoción no lo damos en una sola sesión, ya que con los niños siempre hay que repetir para que entiendan bien el concepto.

Hay que remarcar que nuestros cursos, aunque sí que inicialmente están divididos por en que curso escolar se encuentra el niño, también influye el tiempo que llevamos trabajando con él. Niños más pequeños que llevan desde el principio con nosotros tienen más conocimientos que alumnos que vengan nuevos más mayores, y por tanto estarán en un grupo donde se le enseñen conceptos más complicados o se avance a un ritmo más rápido. Hemos llegado a juntar clases de niños de ocho años con uno de doce en momentos puntuales, tenemos clases muy heterogéneas en los que a edades se refiere. Nuestra experiencia con esta mezclad de edades además ha sido buena, porque a los pequeños le gusta eso de estar con el niño mayor y a los mayores les entra el modo hermano mayor y ayudan a los pequeños con lo que pueden.

4.1.2 ¿Hay algún otro (concepto/herramienta) que te gustaría usar?

No.

4.2 ¿Qué herramientas usas? (P. Ej. Bitbloq, Code.org, Scratch, etc.)

4.2.1 ¿Por qué?

Las herramientas que usamos, Scratch, IDE de Arduino, kits de robótica de Lego, que algunos son programables y otros no, y Minecraft para diseño 3D y también para la programación.

Con estos kits hacemos especial hincapié en la transmisión de movimiento y montaje de modelos paso a paso, nos interesa que los alumnos por sí mismo sepan montar y engranar algo el robot por su cuenta y luego puedan programarlo. Es decir, la construcción del robot, la programación del robot y luego ya los más mayores el diseño en 3D del robot.

Optamos por kits de Lego porque llevan muchísimos años trabajando con el tema de la robótica orientada a niños, ya no tanto por el tema lúdico de que los niños se lo pasen bien y sea un juguete, sino porque son kits que se notan que han sido pensados y a los que se les ha dado muchas vueltas para que todo encaje y sea fácil enseñar con ellos.

4.2.2 ¿Hay algún otro (concepto/herramienta) que te gustaría usar?

Funcionamos bien con los Kits de Lego.

5 - Opinión sobre Herramientas que analizo

5.1 Bitbloq

5.1.1 ¿Conoces esta herramienta? ¿Qué te parece?

La conocemos, pero no la usamos.

5.1.2 ¿Qué hechas en falta/Qué mejorarías?

-No da lugar-

5.2 Code.org

5.2.1 ¿Conoces esta herramienta? ¿Qué te parece?

Sí, pero no la usamos de forma habitual en nuestras clases. Aunque es una herramienta muy chula y con posibilidades.

5.2.2 ¿Qué hechas en falta/Qué mejorarías?

No caigo ahora en nada que mejoraría.

5.3 Scratch

5.3.1 ¿Conoces esta herramienta? ¿Qué te parece?

Sí que lo conozco, y trabajamos normalmente con ella

5.3.2 ¿Qué hechas en falta/Qué mejorarías?

No le echo de menos nada, trabajamos con la versión 1.4 y estamos satisfechos tal y como funciona.

6 - Recomendaciones, otros

6.1 Si tuvieses que recomendar una herramienta, ¿cuál sería? ¿Porqué?

El kit de Lego WeDO que usamos aquí porque es con el kit con el que llevamos trabajando desde el principio y que nos ha funcionado siempre bien.

6.2 ¿Alguna otra valoración sobre este tipo de formación?

Sobre la relación Niñas/Niños:

Tenemos menos niñas que niños, pero normalmente tanto para niños como niñas quien viene y se engancha se engancha hasta el final, no conforme se van haciendo más mayores se van desapuntando más niñas que niños. Además, el tener grupo mixto ayuda bastante al desarrollo de la clase, a la dinámica de la misma.

A lo mejor en los colegios sí que hay mayor número de niñas y por tanto una proporción más equitativa, es más fácil que se apunten a una actividad extra escolar que ofrece el colegio en su oferta de actividades extra escolares que vengan de propio a nuestra academia a apuntarse a actividades relacionadas con la informática.

Normalmente en los colegios los padres tienen que apuntar a los niños a actividades para cubrir horarios, por ello los apuntan un poco a lo que les parece más interesante, por eso creo que hay más proporción de niñas en los grupos que llevamos en los colegios.

Sobre la educación STEM

Conozco el concepto y me parece interesante, aunque lo estoy viendo también por ser madre, aparte del enfoque de profesor de la academia. Cuantas más cosas aportes a los niños de una forma divertida mejor. Cuanto más separes la educación del aprender de memoria conceptos y la educación tradicional, mejor.

Un problema de los que se te plantea en el día a día requiere de un montón de materias y disciplinas para solucionarlo, por eso tiene sentido el enseñar la relación que hay entre las disciplinas y aún más cuando se hace el enfoque que utiliza la educación STEM.

Sobre el pensamiento computacional

Creo que el pensamiento computacional y la informática deberían de incluirse como asignatura dentro del programa escolar normal. Creo que el pensamiento computacional sirve para muchas cosas aparte de desarrollar una aplicación, es una herramienta más que puedes usar a la hora de enfrentarte a los problemas que te vayan surgiendo. Creo que ya debería de estar incluido en los programas educativos.

6.3 ¿Sabes de algún otro profesor/centro/Academia que imparta formación de este tipo?

Creo que ya has hablado con todos los centros que conozco.

7º Entrevista - Andreu Camps (Avanza Tiempo Libre)

1 - Datos Personales

1.1 Nombre:

Andreu Camps

1.2 Centro/Profesión:

Avanza Tiempo Libre (<http://www.avanzatiempolibre.com/>)

1.3 Titulación:

Bachillerato y varios cursos (monitor de tiempo libre, etc.) no relacionados con la informática.

1.4 ¿Cuánto tiempo llevas impartiendo formación de este tipo?

Empecé en la Asociación de Altas Capacidades de Aragón (<http://sinlimites.altacapacidad.net/>) hace 6, 7 años, aunque con avanza tiempo libre llevamos unos 4 o 5 años.

1.5 ¿Cómo/Porqué empezaste a dar formación de este tipo?

Empezamos experimentando poco a poco enseñando robótica a los niños. Vimos que la robótica era algo con lo que se podía enseñar conceptos de programación e informática de manera sencilla y decidimos desarrollarlo.

Además, cada vez es más importante saber relacionarse con la informática y las nuevas tecnologías, cada vez más niños tienen como un juguete el móvil o la Tablet de sus padres y cada vez hay más tecnología a nuestro alrededor.

Lo que tratamos de enseñárselos es qué es la tecnología, utilizarla, mejorarla y sobre todo que sean ellos lo que controlen la tecnología. Y esto se hace sobre todo enseñándoles desde pequeños que es la robótica y la programación.

Nuestra organización se dedica principalmente a impartir extra escolares en otros centros educativos, que es donde tenemos la mayoría de grupos y cursos, aunque también tenemos unos pocos grupos con los que trabajamos en el centro. Normalmente contratan nuestros servicios a través de otras academias u organizaciones que quieren ofertar cursos de informática o robótica, y solo algunas ocasiones es el colegio el que contacta directamente con nosotros.

1.6 ¿Has dado formación de este tipo en algún otro centro? ¿Cuáles?

Sí, en la Asociación de Altas Capacidades de Aragón

2 - Opinión sobre el tema

2.1 ¿Crees que es importante impartir este tipo de formación? ¿Por qué?

Si sin duda. Y además tratamos hacer una robótica de campaña, para niños, para niñas, para el que le guste y para el que no.

Sé que la robótica puede sonar a una actividad extra escolar más “frikí”, se suelen apuntar más los niños más aficionados a los videojuegos, etc. Nosotros intentamos que la actividad les guste a todos los niños y que todos sean capaces de entender cómo va la programación, intentar quitarle la etiqueta de que son cosas difíciles y frikis. Nuestro principal interés es que se lo pasen bien y que aprendan.

Intentamos que aprendan de una manera visual, ya que van a tener en las manos los materiales, los kits de robótica y otros recursos con los que trabajamos, y que aprendan qué es la tecnología, para qué sirve y qué podemos hacer con ella. Que en el futuro si se quieren dedicar a estudiar una a carrera como una ingeniería informática estupendo, pero que, si optan por otra como periodismo, cuenten con recursos tecnológicos con los que ya están familiarizados para hacer su trabajo de una forma más eficiente.

2.2 ¿A partir de que edades? ¿Por qué?

Nosotros empezamos con primero de primaria, lo que son 5 o 6 años. Más pequeños, hemos hecho pruebas con infantil, pero es inviable, son demasiado pequeños para que puedan hacer las actividades.

Hay una capacidad muy importante y necesaria a la hora de hacer robótica y es la capacidad espacial. A la hora de hacer rutas de programación, hay que entender la robótica como un árbol, según qué circunstancias detecte, el robot actuará de una manera u otra, esta parte imaginativa de que podemos hacer con el robot, como niños tan pequeños no se puede. También la psicomotricidad necesaria para montar un robot no la tienen, por lo tanto, trabajar con niños más pequeños de primaria resulta complicado y por tanto recomendamos empezar a partir de primero de primaria

3 - Cursos/Clases

3.0 ¿Cuántos cursos o clases distintas impartes?

Ahora mismo estamos centrados en el tema de la robótica y cursos relacionados con ella. Tenemos dos niveles, uno para primero y segundo de primaria que utilizan unos kits más adaptados a esas edades. Y otro para alumnos a partir de tercero de primaria.

Tenemos una evolución escalonada, tanto dentro del curso como a años que repitan. Esto significa que, aunque le niño repita la actividad varios años, no va a repetir las mismas actividades y diseños que hizo en años pasados, sino que hay una evaluación a años. Creo que hay unos 4 o 5 años que se podría apuntar un alumno de forma continua sin que llegase a repetir actividades en ninguno de esos cursos consecutivos, aunque fueran del mismo nivel.

3.1 ¿Cuántos alumnos tienes por clase? ¿Consideras que es un número de alumnos adecuado o debería ser mayor/menor?

Una media de entre 12 y 15 niños por clase. Nosotros tenemos los cursos limitados a 15 niños, a partir de esa cantidad se abre un nuevo grupo.

Además, consideramos que el número ideal de niños para una clase son 12 alumnos.

3.2 ¿Qué perfil de alumno tienes? (edad, motivación, conocimientos previos, etc.)

Niños de primaria, que se apuntan a la actividad extra escolar que oferta su colegio y que nosotros realizamos. Los suelen apuntar los padres, aunque algunos de ellos acuden por iniciativa propia, porque les llama la atención las tecnologías.

Sobre conocimientos previos, hay de todo, aunque normalmente son niños que no han trabajado con robótica y se apuntan por primera vez y que en años posteriores repiten si les ha gustado.

3.3 ¿Cuánto duran las clases?, ¿Consideras que es un tiempo adecuado o debería ser mayor/menor?

Una hora dentro del horario extraescolar que disponga el colegio que ha contratado nuestros servicios.

Consideramos que una hora es el tiempo adecuado para una clase, ya que los niños a fin de cuentas son niños, y se cansan si les planteas clases de mayor duración.

3.4 ¿Cuántas clases por semana?, ¿Consideras que es un número de días adecuado o debería ser mayor/menor?

Una vez por semana, y me parece una frecuencia adecuada.

3.5 ¿Los alumnos disponen de material propio para las clases (por ejemplo, se traen sus propios portátiles) o los proporciona el centro?

Utilizamos las aulas de informática de los centros donde impartimos las extras escolares. Siempre intentamos que sea con un nivel invasivo mínimo para el centro, es decir, no guardamos archivos que se quedan luego en los ordenadores, no editamos configuraciones o añadimos programas a los equipos, etc. Al final del curso, o incluso de cada clase, intentamos dejar el aula de informática que nos prestan tal y como nos la hemos encontrado.

Respecto a los kits de robótica u otros recursos necesarios para las actividades, siempre los aportamos nosotros y llevamos su gestión y mantenimiento, si algún componente se rompe nos encargamos de remplazarlo o arreglarlo si es posible.

3.6 Dinámica de clase

3.6.1 ¿Cómo son las clases, explicas todo con ejemplos y luego das tiempo para que se trabajen en ejercicios, se van haciendo todos a la vez, etc.?

Los alumnos siempre trabajan por grupos, en el caso de los pequeños 2-3 alumnos por robot y para los mayores 3-4 alumnos por robot.

La metodología de trabajo es más o menos la misma para todos los cursos, primero llega un monitor, organiza por grupos a los alumnos, reparte el material y los modelos que se van a montar y programar en esa clase y se ponen con la actividad.

Los programas de los modelos están muy guionizados y siempre se les proporcionan a los alumnos, de modo que ellos solo tienen que seguir paso a paso el modelo que tienen delante tanto para construir como para programar el robot. Las clases siempre tienen dos partes, una primera parte de construcción y otra de programación, normalmente les gusta más la parte de construcción, por eso siempre tenemos de las dos partes, para tenerlo equilibrado y que no se aburran haciendo un programa muy largo durante varias clases.

El monitor está de asesor por si se pierden o necesitan ayuda o si necesitan cualquier cosa.

Los cursos están divididos en 3 trimestres. El primero consiste en construir unos modelos básicos con una programación básica para ir adaptando a los niños e ir cogiendo ritmo de construcción, y programación.

El segundo trimestre está orientado a hacer distintos modelos de robot que vienen también guionizados como en el primer trimestre, pero al final de cada modelo el monitor les plantea retos sobre cómo funciona ese robot que acaban de construir, tanto para comprobar que los alumnos han aprendido como para asegurarse que lo han construido y programado correctamente.

En el tercer trimestre el monitor plantea un problema que los alumnos han de resolver ayudándose de todo lo aprendido hasta ahora, utilizando todo el material que tienen a su disposición construyen un robot desde cero que resuelva el problema planteado.

3.6.2 ¿Siempre se trabaja con ordenadores o también realizas actividades sin ellos?

No están con el ordenador constantemente ya que en todas las clases siempre hay un apartado de construcción y montaje del robot para el cual el ordenador no es necesario, pero después de eso han de utilizar el ordenador para programarlo.

De modo que no se está trabajando toda la clase con el ordenador, pero sí que un rato en todas las clases se está.

3.6.3 ¿Mandas deberes/trabajo para casa?, en caso afirmativo, ¿con que frecuencia?

Nunca mandamos deberes para casa.

3.6.4 ¿Haces exámenes pruebas de evaluación?, en caso afirmativo, ¿con que frecuencia?

No, nunca.

4 - Conceptos que enseñas y herramientas que usas

4.1 ¿Qué conceptos enseñas? (P. Ej., condicionales, bucles, funciones, etc.)

4.1.1 ¿Por qué?

Bucles, condicionales, variables, enseñamos todos estos conceptos, pero siempre en relación con un robot. En el caso de los motores, por ejemplo, enseñamos que se puede programar un motor para que actúe por rotaciones, para situarse a ciertos grados, o que funcionen durante un determinado número de segundos. En el caso de los sensores, enseñamos qué detecta cada uno y para qué sirven, y como el disponer de sensores nos invita a utilizar condicionales para que el robot se comporte de una forma u otra, que disponga de un árbol de decisiones.

Hay que decir que es programación por bloques muy visual, en la que queda muy diferenciada que hace cada bloque y para qué sirve.

Lo más difícil que pueden llegar a aprender o por lo menos lo que más le suele costar, aunque parezca curioso, son los bucles y cuando utilizarlos. Porque las condicionales pueden llegar a entenderlo antes, al tener la analogía de que el robot puede seguir dos caminos si se encuentra con un obstáculo delante, o que un interruptor tiene dos posibilidades, estar encendido o apagado. Sin embargo, con los bucles les cuesta saber cuándo usarlos e incluirlos en el programa. Sobre todo, el concepto de un bucle infinito que determine el comportamiento general de un robot, que esté todo el rato comprobando a través de sensores si está siguiendo el camino correcto.

4.1.2 ¿Hay algún otro (concepto/herramienta) que te gustaría usar?

Con chavales de primaria, ya tiene bastante con lo que le enseñamos, con secundaria sí que intentamos en ocasiones meternos en variables y toma de decisiones complejas en función de los datos obtenidos por los sensores y parámetros que ya les hayamos definido. Realmente es trabajar con los conceptos básicos que hemos enseñado y que conocen, pero dándoles varias vueltas más.

Aunque normalmente no llegamos a niveles tan complejos, son pocos alumnos los que se interesan tanto por las actividades de programación como para tener que llegar a ese nivel de complejidad.

4.2 ¿Qué herramientas usas? (P. Ej. Bitbloq, Code.org, Scratch, etc.)

4.2.1 ¿Por qué?

Hasta ahora lo que más nos gusta son las versiones educativas de lego. Modificadas un poco a nuestra manera, adaptados a nuestra forma de trabajo y currículo. Lego Mindstorms v3 es la versión que usamos. Además, nos gusta que sea una aplicación de escritor en lugar de una aplicación web, ya que nos quitamos el problema de desconexiones o de que vayan lentas porque la red no da para más.

Hemos probado todo, pero de cara los chavales, las herramientas de Lego son las que más nos gustan.

No obstante, si en el futuro saliese al mercado un kit de robótica que consideramos mejor y más interesante que los que utilizamos actualmente, lo cambiaríamos. No estamos “casados” con Lego en ningún caso, y siempre buscamos utilizar las herramientas y kits que entendemos que son mejores para trabajar y enseñar a los niños.

4.2.2 ¿Hay algún otro (concepto/herramienta) que te gustaría usar?

Con Lego estamos muy contentos con este último modelo, tanto con los kits de robótica como con la programación. Así que no, no hay otras herramientas que nos gustaría usar.

5 - Opinión sobre Herramientas que analizo

5.1 Bitbloq

5.1.1 ¿Conoces esta herramienta? ¿Qué te parece?

Sí que la conozco.

5.1.2 ¿Qué hechas en falta/Qué mejorarías?

No sabría decirte, ya que no he trabajado con ella en profundidad. Que fuese una aplicación de escritorio en lugar de una aplicación web.

5.2 Code.org

5.2.1 ¿Conoces esta herramienta? ¿Qué te parece?

Sí que la conozco.

5.2.2 ¿Qué hechas en falta/Qué mejorarías?

Me paso como con la anterior, la conozco poco, pero sé que es una aplicación web y preferiría que fuese de escritorio

5.3 Scratch

5.3.1 ¿Conoces esta herramienta? ¿Qué te parece?

Sí que lo conozco y sé que lleva tiempo, pero a nosotros no nos gusta, creemos que está anticuado.

5.3.2 ¿Qué hechas en falta/Qué mejorarías?

Considero que esta anticuado por la construcción en columna de los bloques, a nivel visual, porque a poco que tengas un programa largo ya es difícil trabajar. Además, que solo es programación pura y dura, sin robot, que creo que les acaba cansando rápidamente que si no tienes un parte de construir y diseñar un robot.

6 - Recomendaciones, otros

6.1 Si tuvieses que recomendar una herramienta, ¿cuál sería? ¿Porqué?

Hoy por hoy, si tengo que recomendar una herramienta sería Lego y sus kits. A nivel educativo creo que es lo más potente, aunque también es la más cara.

6.2 ¿Alguna otra valoración sobre este tipo de formación?

Los niños lo que hacen en nuestras clases es jugar, el objetivo es que se lo pasen bien, y creo que debería ser la norma para todos estos cursos. Es difícil tratar de enseñar conceptos tan complejos sino planteas actividades lúdicas para que aprendan jugando.

Sobre la proporción chicos chicas

La robótica está asociada a una actividad “friki” de chicos, pero cada vez se quita más esa etiqueta y cada vez tenemos más chicas en los cursos. De hecho, en varios centros que impartimos la actividad extraescolar, ya tenemos grupos homogéneos de chicos y chicas en los que hay tantos chicos como chicas.

Hay que intentar plantear la actividad de manera que resulte atractiva a todo el mundo, chicos y chicas, niños que les guste más la tecnología y niños a los que menos.

6.3 ¿Sabes de algún otro profesor/centro/Academia que imparta formación de este tipo?

No conozco a nadie que no hayas entrevistado ya.