



**Universitat de les
Illes Balears**

Facultad de Educación Guillem Cifre

Memoria del Trabajo Final de Grado

*La enseñanza de la programación informática en
Educación Primaria: situación actual, análisis y
presentación de experiencias de aula en España.*

Héctor Arranz de la Fuente

Grado de Educación Primaria

Año académico 2014-15

DNI del alumno: 41535637Y

Trabajo tutelado por: Adolfin Pérez Garcias

Departamento de Sociedad de la información y el conocimiento

El autor autoriza el acceso público a este Trabajo Final de Grado.

Palabras clave:

Programación informática, pensamiento computacional, experiencias, Educación Primaria, TIC.

LA ENSEÑANZA DE LA PROGRAMACIÓN INFORMÁTICA EN EDUCACIÓN PRIMARIA: SITUACIÓN ACTUAL, ANÁLISIS Y PRESENTACIÓN DE EXPERIENCIAS DE AULA EN ESPAÑA.

*‘Teaching computer programming in elementary school: the current state of affairs, analysis and classrooms’
experiences in Spain.’*

*Héctor Arranz de la Fuente
Universitat de les Illes Balears*

RESUMEN

Existe un movimiento cada vez más numeroso que apuesta por la enseñanza de la programación informática en la escuela primaria como medio de desarrollo de capacidades transversales como el pensamiento analítico, la resolución de problemas, el trabajo en equipo o la creatividad. Prueba de ello son los países que ya la han incorporada a sus currículums de enseñanza como Estonia, Francia o Reino Unido, o la proliferación de asociaciones de promoción de la enseñanza de la programación informática desde las edades más tempranas.

Este trabajo tiene como objetivo mostrar diferentes experiencias reales de enseñanza de la programación informática con alumnos de Educación Primaria. Para ello, habiendo enmarcado teóricamente y contextualmente el contenido, se lleva a cabo un trabajo de investigación de metodología mixta. Por un lado, una encuesta realizada a todos los centros de educación primaria públicos y concertados de Mallorca, y por otro la búsqueda sistemática de experiencias en bases de datos, revistas educativas y portales web especializados. Como resultado se presentarán las experiencias españolas más ilustrativas de cómo enseñar programación informática en diferentes contextos de aula.

PALABRAS CLAVE: Programación informática, pensamiento computacional, experiencias, Educación Primaria, TIC.

ABSTRACT

There is a growing movement in elementary school that focuses on teaching computer programming as a development of transversal skills such as analytical thinking, problem solving, teamwork and creativity. There is some countries, who have already incorporated computer programming into the course itineraries as Estonia, U.K. There is also a lot of organizations who promulgate the teaching of computer programming in the elementary school.

This work aims to provide a ballpark figure of the different computer programming experiences in the case of primary education' students. Accordingly, we have reviewed a vast body of research and, hence, offered a strong theoretical framework. Regarding to methodological issues, we did decide to adopt a mixed methodology. First, a survey was conducted in all public and state-subsidized primary education centres for Mallorca. Last but not least, we did perform a systematic data search of a more qualitative nature in well-known databases, educational magazines, and websites. As a result, we present the most illustrative Spanish experiences of how to teach computer programming in different contexts.

KEY WORDS: Computer programming, computational thinking, experiences, primary education, ICT.

1. INTRODUCCIÓN

La escuela, aunque se trate de una institución históricamente recelosa a los grandes cambios, en los últimos años ha incorporado masivamente las TIC a las aulas (Fraga, Gewerc, 2006), ya sea en forma de pizarras digitales, ordenadores o tabletas. Este fenómeno debería, en buena parte, cambiar la dinámica de trabajo en el aula, pero, salvando experiencias aisladas, y con aisladas nos referimos a que no aparecen de manera coordinada ni provenientes del currículum (Moreno J., 2014) este cambio no se da de manera general. No se hace un uso revolucionario, no se utiliza el potencial que nos brindan estas herramientas. La escuela 'se

apropia de aquello que responde de manera general a los mitos y creencias compartidas” (Fraga, Gewerc, 2006).

A pesar de ello, existe un movimiento, cada vez más numeroso, que apuesta por la enseñanza de la programación en la escuela con el objetivo de cambiar estas metodologías para favorecer el desarrollo cognitivo del niño y fomentar una nueva forma de aprender. Se trata de una “nueva” manera de entender la escuela como tal (Papert, 1995). Y decimos “nueva”, por qué las ideas en las cuales se sustenta provienen de la mitad del siglo XX.

Este trabajo, enmarcado en el la elaboración del Trabajo Final de Grado de Educación Primaria, tiene como objetivo conocer y mostrar qué prácticas se están llevando a cabo en las aulas de Educación Primaria de España en relación a la programación informática. Para ello, enmarcaremos teóricamente la enseñanza de la programación informática, además de mostrar qué países europeos la han incorporado a sus currículos de enseñanza, presentar las asociaciones de promoción de la programación informática en las aulas y explicar cuáles son sus funciones.

1.1 Marco teórico de la enseñanza de la programación informática

1.1.1 Seymour Papert y Logo

Para hablar sobre la enseñanza en de la programación, hay que empezar por Seymour Papert¹, creador del primer software de programación educativo para niños LOGO en 1968, junto a Cynthia Solomon, Daniel Bobrow y Wally Feurzeig. Para Papert, el ordenador cambiaba las características del ambiente de aprendizaje del alumnado, proporcionando nuevas formas de aprender más acordes a la naturaleza del niño, y la programación era una de esas nuevas maneras. LOGO tiene un lenguaje de programación fácil, de instrucciones sencillas, que básicamente presenta retos intelectuales que los alumnos han de resolver escribiendo códigos, creando programas. Los niños deben programar la computadora, no la computadora

¹ En el apartado de conclusiones se hará especial mención a Seymour Papert y al Construccinismo, por su gran influencia en el campo

a ellos (Paulo Blinkstein, 2013). Influído por Piaget, y a partir del aprendizaje por descubrimiento, propuso una nueva visión del aprendizaje, enfocándola como si de un videojuego se tratara, buscando los intereses naturales de los alumnos, los propios de su niñez. “La esencia de Piaget era que el aprendizaje ocurre sin ser planeado u organizado por los profesores o escuelas. ¡Su idea era que los niños se desarrollen intelectualmente sin ser enseñados!” (Paulo Blinkstein citando a Papert, 2013). Con Logo en particular, los niños podían pensar sobre sus procesos cognitivos, reflexionar sobre sus errores y utilizarlos para crear nuevos programas, favoreciendo así sus actividades metacognitivas (Martí, 2002).

Aunque entre sus ventajas estaba el desarrollo de la capacidad metacognitiva, la mejora en resolución de problemas y en la capacidad de orientar-se en el espacio, especialmente en la concepción de la geometría (Delval, 1986; citando a Clements & Nastasi, 1999; Clements & Sarama, 2002), en las primeras décadas de vida de LOGO, aparecieron investigaciones sobre su influencia en el desarrollo cognitivo del alumno que obtuvieron resultados desfavorables y ambiguos. Delval (1986), siguiendo a Johnson (1986), apuntaba a investigaciones sin cambios significativos y resultados poco alentadores. Un ejemplo de ello son: Ramírez (s.n.) explica que Howe, O’Shea y Weir (1980) llevaron a cabo un estudio sobre el efecto del uso de LOGO en el desarrollo de la geometría con alumnos de 12 años, divididos en dos grupos. Uno de ellos llevaba dos años de experiencia con el lenguaje que utiliza LOGO; el otro no. En un primer test realizado, los alumnos con experiencia sacaron mejores resultados que el otro grupo. En un segundo test, en cambio, no se observaron diferencias, incluso fue el grupo sin experiencia quien obtuvo mejores resultados. Pea y Kurland (1984) realizan un estudio trabajando con dos grupos, uno de ellos trabaja con LOGO, y el otro sin la ayuda del software. El estudio rebeló que los dos grupos obtuvieron los mismos resultados. Cabe destacar que estos dos últimos ejemplos fueron investigaciones de una duración de entre 50 y 60 horas, tiempo insuficiente para originar cambios significativos. Adelson (1981) cree que la dificultad se encuentra en el dominio del lenguaje, que se necesita tener un alto dominio para conseguir cambios en los esquemas de aprendizaje.

Ramírez (s.n.) afirma que autores como Arithman-Smith, Haywood y Bransford (1984), Delclos y Haywood (1984) o Martí (1992), propusieron superar las deficiencias de la metodología de Papert, utilizando una metodología más instructiva. Destaca el trabajo de

Martí, que hace una propuesta que se sustenta en dos tipos de metodología: por un lado, utilizar la mediación del aprendizaje. Por el otro, la aplicación en situaciones específicas instructivas del constructivismo, todo ello a través del medio informático y humano. Martí piensa que con el aprendizaje por descubrimiento se pueden adquirir determinados esquemas generales, pero no obtener aprendizajes específicos. Siguiendo este enfoque encontramos la investigación de Delclos y Littlefield (1984), que trabajaron con LOGO con dos grupos utilizando metodologías diferentes. El primero a partir del aprendizaje por descubrimiento, y el segundo con la ayuda de un tutorial estructurado. Los resultados determinaron que el grupo que había trabajado con el tutorial tenía más facilidad a la hora de crear programas.

Por otro lado, sí que encontramos investigaciones prometedoras y que demuestran efectos positivos en el desarrollo cognitivo a través de la enseñanza de la programación. En el año 1986, el artículo *Effects of Logo and CAI environments on cognition and creativity* (Douglas H. Clements, 1986), que realizaba un estudio sobre 80 alumnos de entre 6 y 8 años, expuso que se encontraron resultados positivos en competencia operativa y creatividad, además de mayor capacidad de atención, más autonomía y más motivación por adquirir nuevos aprendizajes. Otro estudio, *Logo and Geometry* (Douglas H. Clements, 2001), realizado a alumnos de 12 años, muestra en sus conclusiones que los niños obtuvieron mejores resultados en matemáticas, razonamiento y resolución de problemas, tal y como apuntaban Papert, Disesse y Weirn (1979). Douglas también observó efectos significativos en el área de matemáticas cuando acababa de empezar el proyecto *LOGO Geometry*. Otras investigaciones más recientes, como *Technology and school change: New lamps for old?* (Douglas H. Clements & Sudha Swaminathan, 2012), demuestran que con la enseñanza de la programación se consiguen mejores resultados en el desarrollo de las habilidades cognitivas, socio-emocionales y en creatividad.

Al programar estamos dando instrucciones al ordenador para solucionar un determinado problema. Este proceso requiere de una planificación, de precisión en el uso del lenguaje de programación, crear y comprobar hipótesis, identificar secuencias para conseguir objetivos y depurar errores. Es decir, se hace explícito el proceso que hay que seguir para llevar a cabo una tarea determinada (Delval, 1985). Delval siguiendo a Feurzig (1981), apostaba que al programar y al hacer manifiesto el proceso que acabamos de describir, se originarían cambios

en el pensamiento que favorecerían la realización de otras tareas que conllevan procesos análogos, tales como la resolución de problemas.

1.1.2 Pensamiento Computacional

Neelie Kroes, vicepresidenta de la Comisión Europea y comisaria de Agenda Digital y Androulla Vassiliou, comisaria de Educación, Cultura, Multilingüismo y Juventud, en una carta enviada a los diferentes ministros europeos, instan a los gobiernos de la UE a incluir la programación en sus currículums escolares por, entre otros motivos, sus beneficios en el desarrollo de capacidades transversales:

“La programación no sólo puede apoyar los estudios en matemáticas, ciencia, tecnología e ingeniería. También ayudará directamente a los estudiantes a desarrollar habilidades transversales, como el pensamiento analítico, la resolución de problemas, el trabajo en equipo y la creatividad” (Kroes N. & Vassillou, A. 2014).

Siguiendo con este enfoque, en 2006 surge el movimiento que más fuerza está cobrando en los últimos años en el campo de la enseñanza de la programación. Se trata del Pensamiento Computacional, propuesto por Jeannette Wing, el cual “implica resolver problemas, diseñar sistemas y comprender el comportamiento humano, haciendo uso de los conceptos fundamentales de la informática” (Wing, 2006). Alfredo V. Aho en su artículo *Computation and Computational Thinking* para *The Computer Journal* (2012) lo define como el proceso que permite formular problemas de forma que sus soluciones puedan ser representadas como secuencias de instrucciones y algoritmos. Otra definición interesante, es la que hizo Steve Furber, en el documento *Shut down or restart? The way forward for computing in UK schools* para *The Royal Society*:

“El pensamiento computacional es el proceso de reconocimiento de aspectos de la informática en el mundo que nos rodea, y aplicar herramientas y técnicas de la informática para comprender y razonar sobre los sistemas y procesos tanto naturales como artificiales” (Furber S., 2012).

En 2011, La Sociedad Internacional de la Tecnología en la Educación (ISTE) y la Asociación de Profesores de Informática (CSTA), colaboraron con líderes mundiales de la investigación, la industria y la educación superior, secundaria y primaria, para establecer una definición operativa del pensamiento computacional, establecer sus características principales y su vocabulario esencial. El objetivo era crear un marco teórico sobre el cual los profesionales

podieran trabajar. Según esta definición, el pensamiento computacional es un proceso de resolución de problemas (aunque no se limita sólo a esta capacidad) que incluye las siguientes características, entre otras: formular problemas de forma que se permita el uso de un ordenador y otras herramientas para ayudar a resolverlos; organizar y analizar lógicamente la información; e identificar, analizar e implementar posibles soluciones con el objetivo de lograr la combinación más efectiva y eficiente de pasos y recursos.

Mitchel Resnick, en su artículo *Learn to code, code to learn* (MIT, 2013) equipara la programación a una extensión de la escritura. Explica que gracias a la escritura nos podemos comunicar, pero que además nos ayuda a desarrollar muchas otras capacidades: aprendemos a organizar, refinar y reflejar nuestras ideas, por lo cual existen muchas razones para que todo el mundo deba aprender a escribir. La programación te permite “escribir” nuevas cosas: historias interactivas, juegos, animaciones o simulaciones. Y del mismo modo que pasa con la escritura, con la programación se desarrollan otras capacidades y aprendizajes, por lo que también existen muchas razones para que todo el mundo aprenda a programar: desarrollar y comunicar ideas (creatividad), estrategias de resolución de problemas o aprender conceptos matemáticos e informáticos. La programación se trata de una nueva forma de expresión y un nuevo contexto para el aprendizaje.

Para finalizar con este apartado, queremos destacar que existen estudios que demuestran que al aprender a programar a edades tempranas los alumnos tienen menos estereotipos de género en relación a las Carreras STEM – Ciencias, tecnologías, ingenierías y matemáticas (Burke Ronald J. & Mattis Mary C., 2007 y Intef², 2014).

1.2 La programación informática en los currículums de educación en Europa

Como se ha comentado anteriormente, Neelie Kroes, la vicepresidenta de la **Comisión Europea** y comisaria de Agenda Digital, y Androulla Vassiliou, la comisaria de Educación, Cultura, Multilingüismo y Juventud (2014), enviaron una carta conjunta a los diferentes

² Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado.

ministros europeos instándoles a desarrollar el programa ‘‘Opening Up Education³’’ en sus países, y aumentar el nivel de entrenamiento digital de los docentes, alumnos y familias facilitando, entre otras medidas, oportunidades a éstos para que puedan aprender a programar en las escuelas.

Kroes y Vassillou (2014), creen en el aprendizaje de la programación desde la escuela se encuentra la solución al desempleo juvenil europeo en el futuro, ya que calculan que la industria de la informática y las comunicaciones (TIC) demandará unos 900.000 empleos en el año 2020, mientras que el número de graduados en informática en Europa se está estancando. Además- y esto es lo que más nos interesa- apuestan por la programación para que los alumnos desarrollen capacidades transversales de aprendizaje, como ‘‘el pensamiento analítico, la resolución de problemas, el trabajo en equipo y la creatividad’’ (Kroes N. & Vassillou A. 2014). La comisión Europea también organiza eventos como la Code Week (<http://codeweek.eu/>), en la cual se llevan a cabo cientos de talleres y eventos de programación para niños a lo largo de toda Europa.

El país pionero en Europa en introducir la enseñanza de la programación en su currículum es Estonia (Mateos Manuel, 2014), que en 2012 introdujo el proyecto ProgeTiiger⁴, impulsado por la Tiger Leap Foundation, de forma experimental con una serie de centros, y que a partir del 2013 ya se implantó en todas las escuelas de forma progresiva. El objetivo principal del proyecto es aumentar la alfabetización tecnológica tanto de alumnos como de maestros. Para ello, en un primer momento se dio la formación necesaria a los maestros, enseñándoles no sólo cómo programar, sino que también a desarrollar metodologías y actividades atractivas y qué conocimientos se pueden transmitir en ellas.

Con la enseñanza de la programación, pretenden elevar el interés de los alumnos en las ciencias de la ingeniería, además de desarrollar su pensamiento algorítmico y la habilidad en la resolución de problemas. Los niños aprenden a programar desde los siete años.

³ En el programa ‘‘Opening Up Education’’, se expone una agenda europea para fomentar una docencia y un aprendizaje innovadores y de alta calidad a través de nuevas tecnologías y contenidos digitales.

⁴ Portal del Proyecto Progetiiger: <http://progetiiger.ee/>

En el Reino Unido, a partir de este curso 2014/2015, se ha implementado un nuevo currículum para la enseñanza de la informática, ya que, según su Secretario de Estado de Educación Michael Gove, el existente hasta el momento era “aburrido que sólo enseñaba a sus alumnos a manejar Word y Excel de la mano de profesores aburridos”. El nuevo currículum tiene como objetivo principal “ofrecer una formación informática de calidad que prepare a los estudiantes para que puedan usar el pensamiento computacional y la creatividad para entender y cambiar el mundo que les rodea” (Gove M. 2014)

Los objetivos que pretenden conseguir con este nuevo currículum son (Computing programmes of study: key stages 1 and 2. National curriculum in England, 2014):

- Entender y aplicar los principios y conceptos fundamentales de la informática, incluyendo la abstracción, la lógica, los algoritmos y la representación de la información.
- Analizar problemas en términos computacionales, y tener experiencia práctica escribiendo programas informáticos para solucionarlos.
- Evaluar y aplicar tecnologías informáticas de manera analítica para resolver problemas, incluyendo tecnologías novedosas.

Así, los contenidos que plantean para conseguir estos objetivos son:

Stage 1, equivalente al 1º y 2º curso de primaria:

- Comprender qué son los algoritmos y cómo funcionan en programas en dispositivos digitales; y que los programas se ejecutan siguiendo una serie de instrucciones precisas.
- Crear y depurar programas simples.
- Usar el razonamiento lógico para predecir el comportamiento de programas simples.

Stage 2, equivalente al 3º, 4º, 5º y 6º curso de primaria:

- Diseñar, escribir y depurar programas que consiguieran objetivos específicos, incluyendo el control o la simulación de sistemas físicos; resolver problemas descomponiéndolos en partes más pequeñas.
- Usar estructuras de secuencia, selección y repetición en los programas; trabajar con variables y distintas formas de entrada y salida de datos.
- Utilizar el razonamiento lógico para explicar la forma en que trabajan algoritmos sencillos y para detectar y corregir errores en algoritmos y programas.
- Seleccionar, usar y combinar distintos tipos de software (incluyendo servicios de internet) en diferentes dispositivos digitales para diseñar y crear programas, sistemas y contenidos que consigan objetivos determinados, incluyendo la recolección, análisis, evaluación y presentación de datos e información.

Para ello, el gobierno está dando gran apoyo y poniendo a la disposición de los maestros materiales y cursos de formación en programación informática.

Otro país que ha incorporado la enseñanza de la programación a sus planes de estudio desde este mismo curso es Francia, tanto en primaria como en secundaria, en forma de asignaturas optativas o extraescolares. El Ministro de Educación Nacional, Benoît Hamon, explica en una entrevista a *Le Journal Diamanche*, que los objetivos básicos de esta nueva implantación son que todos los alumnos han de ser capaces de entender el código fuente, conocer los pilares fundamentales de la programación y que, al final del proceso, sean capaces de desarrollar sus propias aplicaciones con algoritmos simples. Hay que destacar, que la enseñanza no se centra en que todos los alumnos sean programadores, sino en detectar nuevos talentos en un sector tan importante y estratégico como el de la informática (Hamon, 2014). Además, Hamon explica que gracias a la programación se consigue una mayor participación e interacción en clase. También elimina el miedo a equivocarse: permite a los escolares a superar

las dificultades y a desinhibir-se. Aboga por que sean los maestros de matemáticas y tecnología quienes sean los que primero enseñen la programación en la escuela. Sin embargo, todas las escuelas (ESPE) se inscribieron en el plan de formación inicial.

En el otoño de 2016, Finlandia introducirá un nuevo plan de estudios en el cual se incluye la programación. En la página web koodi2016.com, Leo Pahkin, Consejero de Educación de Finlandia, presenta una guía en la cual se explica cuáles son las razones, los objetivos de cada etapa y resuelve todas las dudas que los docentes puedan tener sobre la enseñanza de la programación. Además, también hace un repaso de lo que se hace en cada país, de sus referentes. Los objetivos para cada etapa educativa se dividen de esta manera:

- **1º y 2º grado, equivalente a 2º y 3º curso:** los alumnos deben aprender a dar órdenes precisas e inequívocas de una persona a otra, al igual que se hace al programar un ordenador. “Por ejemplo “dar tres pasos hacia adelante” y no “dar tres pasos”, que pueden ser hacia los lados también”, explica Pahkin.
- **Grados 3-6, equivalente a 4º, 5º, 6º y 1º de la ESO:** saber utilizar un lenguaje de programación visual. La herramienta no es el código en sí, sino que se utilizaran lenguajes de programación visual como, por ejemplo, el software Scratch.
- **Grados 7-9, equivalente a 2º, 3º y 4º de la ESO:** los alumnos comienzan con la familiarización con uno de los lenguajes de programación. El objetivo no es entender lo que el código es en sí, sino que el objetivo es la comprensión de los conceptos básicos del lenguaje y saber interpretar el código del programa.

En esta guía, también se ofrece un amplio manual de cómo aprender y enseñar programación para los docentes, basado en tutoriales que podemos encontrar en la web de code.org o scratch.mit.edu. También muestran formas de aprender programación sin la necesidad de tener un ordenador.

En Alemania, existe un gran debate sobre la necesidad de enseñar, o no, la

programación en las escuelas. Como cada estado federal del país establece sus propias normas, sólo tres de los 16 estados federales del país imparten programación en sus aulas: Saxonia, Mecklenburg-Western Pomerania y Bavaria (Schmundt H. 2013). El resto de estados realiza cursos introductorios en el procesamiento de textos y usos de internet, y creen que con el uso de ordenadores portátiles o de las PDI en el aula, ya se solventan las necesidades en el desarrollo de la competencia digital de los alumnos, dice Steffen Friederich, profesor de Didáctica de las Ciencias de la Computación en la Universidad Técnica de Dresde.

El debate gira entorno a la necesidad de saber utilizar la tecnología o de saber cómo funciona (Schmundt H., 2013). En un lado, encontramos a Josef Kraus, el presidente de la Federación de Maestros Alemanes, el cual tiene un gran peso en la toma de decisiones en cuanto a la educación en el país germano, que se opone totalmente a la introducción de la programación al plan de estudios. “Los alumnos no tienen que saber cómo funciona una máquina de escribir, lo más importante es que pueden utilizar una”, dice Kraus. Por otra parte, encontramos a profesores como Steffen Friederich o Norbert Breier, profesor de Educación de la Universidad de Hamburgo, los cuales piensan que no se está llevando a cabo la reforma del sistema de educación que su país necesita e, incluso, se están dando pasos atrás, eliminando programas de enseñanza de la programación que se estaban llevando a cabo en Hamburgo.

1.3 Plataformas europeas e intercontinentales de promoción de la programación informática

En la carta enviada por parte de la Comisión Europea a los gobiernos europeos para promulgar la enseñanza de la programación (Kroess N. & Vassilliou A., 2014), se destaca la labor que hacen múltiples iniciativas voluntarias que están naciendo por toda Europa, como CoderDojo, las cuales proponen actividades de aprendizaje atractivas para niños, o no tan niños, ya sea en forma de actividades extraescolares o en formato on-line.

La fundación Coder Dojo⁵ fue creado en 2011 en Dublín por el emprendedor y hacker, James Whellton, y que en ese mismo año recibió el apoyo del empresario y filántropo, Bill Liao, que estaba interesado en hacer crecer el proyecto. Se trata de un movimiento global de

⁵ Información relativa a Coder Dojo disponible en: www.coderdojo.com

clubes de programación dirigidos por voluntarios que quieren enseñar programación a los más jóvenes de forma gratuita.

“En un Dojo (club), los jóvenes, entre 7 y 17 años, aprenden cómo programar, desarrollar sitios web, aplicaciones, programas, juegos y explorar la tecnología en un ambiente informal y creativo. Además de aprender, permite a los asistentes conocer personas afines y a estar expuestos a las posibilidades de la tecnología”, explican en su portal web coderdojo.com.

En estos clubes se promueve la comunidad, el aprendizaje entre iguales, la tutoría entre jóvenes y el autoaprendizaje, además de poner énfasis en ‘‘cómo la programación es la fuerza para cambiar el mundo’’ (Coder Dojo, 2015). Los temas que se tratan en cada Dojo varían según el voluntario que lo dirige, pero comúnmente se hace una introducción a Scratch, software de programación para niños; desarrollos de sitios web utilizando HTML, CSS y PHP; también trabajan con lenguajes de programación como JavaScript, Python o Ruby para desarrollar videojuegos; o experimentan con hardwares de robótica como Raspberry Pis o placas de Arduino.

De esta manera, el primer Dojo fue organizado en 2011 en el Centro Nacional de Software de Cork (Irlanda), y el movimiento se ha ido incrementando hasta tener más de 675 clubes de programación en 54 países del mundo. Así, cualquier voluntario puede crear un Dojo, siempre que sea capaz de encargarse de la creación, el funcionamiento y mantenimiento del club, siempre de forma gratuita, siguiendo y manteniendo el espíritu que fomenta Coder Dojo.

Desde Italia nos llegan dos iniciativas promovidas por la Universidad de Urbino y la asociación NeuNet: Code’s Cool y CodeWeek.it⁶. Code’s Cool es una comunidad de programación abierta e informal, que se basa en la cooperación entre iguales (no hay distinción entre maestros y alumnos) para un mejor aprendizaje. Esta comunidad tiene como objetivo involucrar a los niños, jóvenes, familias, escuelas, estudiantes y académicos en la necesidad de aprender programación, dado que ésta es ‘‘el lenguaje de las cosas’’, ya que ‘‘muchos de los objetos que nos rodean están programados para poder funcionar’’. Además, apuesta por la

⁶ La información relativa a Code’s Cool y codeweek.it disponible en: <http://informática.uniurb.it/codescool> y codeweek.it/mooc

programación para el estímulo del pensamiento creativo, fundamental en el desarrollo de los jóvenes. De esta manera, facilita a todos los usuarios amplios canales de comunicación en los cuales poner en común experiencias, proyectos, dudas o cualquier cosa relacionada con la causa. CodeWeek.it, por su parte, fue creado para que los niños, familias, escuelas o cualquier colectivo italiano interesado, pueda participar en la Code Week que organiza la Comisión Europea cada año.

Estas dos organizaciones, colaboran para la creación de un MOOC (Masive Open Course Online), basado en toda una serie de video tutoriales, los cuales están diseñados para llevar a cabo en aulas, pero que cualquiera puede utilizarlos independientemente. Proponen 13 tutoriales, los cuales van acompañados de sugerencias para que su desarrollo en las escuelas sea más fructífero y atractivo.

Para aprender programación no hace falta tener un ordenador con el cual interactuar. Desde la Universidad de Canterbury (Nueva Zelanda), el Grupo de Investigación en Educación CS, comparte gratuitamente el proyecto CS Unplugged, que se trata de un conjunto de actividades de aprendizaje de ciencias de computación que introducen a los usuarios en el pensamiento computacional, a través de conceptos como los números binarios, los algoritmos o la comprensión de datos. El libro *Cs Unplugged*⁷. *Un programa de extensión para niños de escuela primaria*, el cual comparten bajo una licencia creative commons, contiene actividades de rompecabezas y juegos de acoplamiento que utilizan tarjetas, cuerdas, lápices de colores y muchos otros materiales, las cuales tienen como objetivo que alumnos de primaria aprendan los elementos fundamentales de la programación. Destacan que para ser utilizado por parte de un maestro no hace falta tener conocimientos de programación, ya que son actividades fáciles de aprender y llevar a cabo y que, además, van acompañadas de instrucciones para que su uso en las aulas sea más atractivo.

La organización que más influencia ha tenido en todo el mundo en la promoción de la programación en las escuelas es Code.org y su proyecto Hour of Code. Code.org es una

⁷ Compartido con licencia creative commons. Libro disponible en: <http://csunplugged.org/wp-content/uploads/2014/12/unpluggedTeachersDec2008-Spanish-master-ar-12182008.pdf>

organización pública sin ánimo de lucro, creada en el 2013 por los hermanos Hadi y Ali Partovi, que tiene como objetivo principal aumentar el interés y la participación en las Ciencias Computacionales, poniéndose a disposición de las escuelas y de todo tipo de alumnos. Su meta es que la programación sea una parte fija del currículum escolar. ‘‘El niño debe programar, y no ser programado’’, promulgan desde la organización⁸ (code.org, 2012).

Desde su creación, code.org se ha desarrollado gracias al apoyo de multitud de personalidades y empresas, hasta llegar a ser una organización de apoyo al movimiento mundial de aprendizaje de la programación. Entre los apoyos más destacados destacan los del presidente de los Estados Unidos de América, Barack Obama; los del creador de Microsoft, Bill Gates; los del creador de Facebook, Mark Zuckerberg; los de la estrella de la NBA, Chris Boss; o los de la estrella del pop-rock, Will.i.am. También son socios y colaboradores empresas como Apple, Amazon o Electronic Arts, entre muchas otras.

Hour of Code son actividades con tutoriales de una hora de introducción a las ciencias computacionales, diseñadas para desmitificar la programación y enseñar que cualquiera pueda aprender los fundamentos básicos, a través de más de 25 tutoriales. Tienen opciones para cada edad y nivel de experiencia, empezando por las edades más tempranas (4 años). También añaden una guía para maestros de cómo llevar a cabo una hora de programación.

De esta manera, cualquiera puede acceder a aprender programación desde cualquier parte del mundo. Cuentan con más de 15 millones de usuarios y se han organizado más de 77 mil eventos en todo el mundo.

Cabe destacar, que no solamente crean sus propios cursos, sino que también se asocian con otros proyectos, capacitan a los maestros, asesoran al gobierno de EE.UU. a diseñar nuevas políticas educativas, expanden el movimiento internacionalmente colaborando con diferentes asociaciones y lanzan campañas de marketing para romper estereotipos en favor del aprendizaje de la programación por parte de las mujeres y de los colectivos más desfavorecidos.

⁸ Toda la información sobre code.org y hour of code disponible en: <http://code.org/>

Como hemos comentado, grandes empresas de informática, como Apple o Microsoft, se han unido al movimiento Code.org. Las dos empresas organizan en todo el mundo eventos de la hora de la programación. Pero no son las únicas empresas que apuestan por la enseñanza de la programación para el desarrollo de los alumnos. Google, en colaboración con la administración de los Estados Unidos, ha desarrollado la iniciativa CS First, la cual tiene objetivos similares a code.org. Microsoft, Google y Intellect, colaboraron junto al CAS (Computing at School), un grupo de trabajo que pertenece a la British Computer Society, en el desarrollo de un currículum de informática que sirviese como marco de referencia para escuelas y maestros.

Otras iniciativas para la promoción de la programación para niños y jóvenes son: Codecademy, Code School, Scratch, Kids Ruby, Rails Girls...

Como podemos comprobar, son múltiples las iniciativas que surgen desde de todas partes del mundo demandando la enseñanza de la programación en las escuelas, basándose tanto en sus beneficios en el desarrollo del alumno, como la importancia que va a tener este sector en la economía futura mundial.

1.4 La programación informática en el currículum de Educación Primaria en España

La primera Comunidad Autónoma de España que ha incorporada la enseñanza de la programación informática a su currículum de Educación Primaria ha sido Navarra (Moreno Jesús, 2014). En 2014 publicaron un nuevo currículum de enseñanzas en la Educación Primaria de la **Comunidad Foral de Navarra** (BON número 174, de 5 de septiembre de 2014) en el cual se determinan los contenidos que deben llegar a conseguirse en el área de programación informática, pero dejan en manos de los centros el tipo de proyecto que se llevará a cabo para alcanzar los objetivos. Sin embargo, sí que recomiendan trabajar con el software Scratch. Cabe destacar que la programación informática se incluye en el currículum de matemáticas para 4º y 5º de curso.

Estos son los contenidos, criterios de evaluación y recomendaciones metodológicas que aparecen en el currículum para 4º de Primaria (Currículo de las enseñanzas Primaria de la

Comunidad Foral de Navarra – Área de Matemáticas, 2014):

BLOQUE 1. PROCESOS, MÉTODOS Y ACTITUDES EN MATEMÁTICAS

- Integración de las tecnologías de la información y la comunicación, así como de los lenguajes y herramientas de programación en el proceso de aprendizaje.
- Utilizar herramientas y lenguajes de programación para modelizar y resolver problemas.
- De manera guiada, realizar un proyecto de programación donde hay que describir el algoritmo, descomponer el problema en partes más pequeñas y codificarlo con un lenguaje de programación visual formal (como Scratch).

Estos son los contenidos, criterios de evaluación y recomendaciones metodológicas que aparecen en el currículo para 5º de Primaria (Currículo de las enseñanzas Primaria de la Comunidad Foral de Navarra – Área de Matemáticas, 2014):

BLOQUE 1. PROCESOS, MÉTODOS Y ACTITUDES EN MATEMÁTICAS

- Integración de las tecnologías de la información y la comunicación, así como de los lenguajes y herramientas de programación en el proceso de aprendizaje.
- Utilizar herramientas y lenguajes de programación para modelizar y resolver problemas.
- Diseñar y realizar proyectos de programación donde se utilizan secuencias de comandos, bucles, condicionales, variables, así como distintas formas de entrada y salida de datos (interacción con el ordenador).

Este currículo se implantó de manera gradual, el primer año para los alumnos de 5º de Primaria, para el posterior curso extenderlo a los alumnos de 4º. El objetivo era dar tiempo a los docentes y centros para formarse en los conocimientos necesarios para su aplicación

(Gobierno de Navarra, 2014). De la misma manera, el departamento de Educación de Navarra ofreció un portal web (<http://codigo21.educacion.navarra.es/>) con materiales y ofreció formación al profesorado, a través de cursos como ‘‘Aprender a programar software en Primaria para mejorar la resolución de problemas. Programa Scratch’’ (Gobierno de Navarra, 2014).

En Cataluña también se ha introducido la enseñanza de la programación en el currículum, pero no en primaria, sino que en secundaria. La Generalitat, con la colaboración de la Fundación Mobile World Capital Barcelona, desarrolla el programa ‘‘mSchools’’, el cual pone a disposición de los alumnos de 4º de ESO una asignatura donde aprender a desarrollar aplicaciones para móviles (Departament d’Ensenyament de la Generalitat de Catalunya, 2014).

1.5 Plataformas españolas de promoción de la programación informática

Entre las asociaciones españolas de promoción de la enseñanza de la programación informática desde las edades más tempranas destaca Programamos, la cual se define como una ‘‘asociación sin ánimo de lucro cuyo objetivo fundamental es promover el desarrollo del pensamiento computacional a través de la programación de videojuegos y aplicaciones para móviles en todas las etapas escolares, desde educación infantil hasta formación profesional’’ (Programamos, 2013).

Jesús Moreno, uno de los cuatro fundadores del proyecto, en una entrevista para eldiario.es, explica por qué apuestan por la enseñanza de la programación:

‘‘En Programamos estamos más interesados en los beneficios educativos asociados a este aprendizaje. Al aprender a programar y al desarrollar el pensamiento computacional se fomenta la creatividad, el emprendimiento y la cultura libre, aumenta la motivación, mejora la autonomía, se trabajan estrategias de resolución de problemas, se conocen diferentes formas de comunicación de ideas... Todas estas destrezas son realmente útiles para cualquier estudiante sin importar la disciplina de su futura actividad profesional, y consideramos que es la preparación ideal para el mundo de hoy, que cambia a velocidad de vértigo y en el que la gente debe inventar soluciones innovadoras constantemente para enfrentarse a nuevas situaciones inesperadas’’ (Moreno J., 2014).

En su portal, programamos.es, ofrecen materiales ya creados y experiencias para que los interesados tomen ejemplos o puedan utilizar estos recursos en sus clases; formación a medida para docentes, como aprender a programar o cómo utilizar la programación para

trabajar transversalmente; organizan y participan en eventos, ponencias o talleres relacionados con la programación informática y desarrollan investigaciones en el campo. Además, incorporan la Comunidad Programamos, una red social de programación en la cual poder compartir experiencias, materiales, aprender, interaccionar y resolver dudas en comunidad.

Otras asociaciones españolas son: Club de jóvenes programadores en Valladolid, Cutilab en Barcelona o Complubot en Madrid.

2. METODOLOGÍA

El objetivo principal del trabajo es mostrar buenas prácticas sobre la utilización de la programación informática en las aulas españolas, con el objetivo de animar y demostrar a otros docentes la fácil aplicación que tiene en el aula.

Cómo objetivos específicos, por un lado queremos conocer la proporción de centros que enseña programación informática en Mallorca. Por otro, hacer un análisis general de experiencias de aula en las cuales se enseñe programación informática o se utilice como herramienta de enseñanza- aprendizaje en España.

El trabajo realizado se ha basado en una metodología mixta, definida por Hernández, Sampieri y Mendoza (2010, p.546) como la combinación en un mismo estudio o proyecto de investigación de componentes cuantitativos y cualitativos. Por este motivo, la investigación se ha dividido en dos fases:

- Identificar y analizar experiencias desarrolladas en Mallorca mediante una encuesta realizada a todos los centros públicos y privados (datos cuantitativos). Además, posterior entrevista con los centros que sí enseñan programación informática (datos cualitativos).
- Análisis de experiencias desarrolladas en España mediante análisis documental, a

partir de la búsqueda sistemática de experiencias en bases de datos, revistas educativas y páginas especializadas (datos cualitativos).

2.1 Fase I: Encuesta.

La primera fase del trabajo consiste en la recogida de datos cuantitativos a partir de una encuesta construida con escala Likert dirigida a 291 centros públicos y concertados de Mallorca. La encuesta, realizada junto a compañeros de la Universidad de las Islas Baleares, tiene como objetivo investigar sobre diferentes aspectos referentes al uso de las TIC en las escuelas públicas y concertadas de Mallorca, entre ellos la enseñanza de la programación informática. Concretamente se realizaban 4 preguntas sobre este apartado:

- ¿Se enseña programación informática en la escuela?
- ¿En qué curso?
- ¿En qué área?
- ¿Qué software o lenguaje de programación se utiliza?

El objetivo de la encuesta es conocer la proporción de uso que hacen de la programación informática en las escuelas, en la realidad. Utilizamos un cuestionario en formato electrónico, el cual hacíamos llegar a los centros a partir de su e-mail oficial, elaborado con la herramienta Formularios de Google, con el objetivo de que sea más ágil y fácil su respuesta y también la recopilación de los datos. Las ventajas de este tipo de cuestionario son (Casas, Repullo y Donado, 2003), entre otras:

- Se evita tener que realizar la entrevista cara a cara.
- Permite el acceso a cualquier persona, sin tener en cuenta la distancia.
- Facilita al entrevistado la realización de esta, ya que puede elegir el momento.

Además, “el cuestionario permite la recolección de datos provenientes de fuentes primarias, es decir, de personas que poseen la información que resulta de interés” (García, 2004).

Después de la encuesta, se procederá a realizar una entrevista a aquellos centros que sí utilicen la programación informática para conocer más en profundidad los proyectos que llevan a cabo.

2.2 Fase II: Búsqueda sistemática.

La última fase se ha basado en la búsqueda y análisis de experiencias desarrolladas en España mediante el análisis documental de artículos existentes en las principales bases de datos educativas, revistas educativas y portales web especializados en la enseñanza de la programación informática en escuelas; concretamente se han rastreado las siguientes:

- ISOC: <http://bddoc.csic.es:8080/isoc.do>
- REDINED: <http://redined.meecd.gob.es/xmlui/>
- Dialnet: <http://dialnet.unirioja.es/>
- DocE: http://www.eurosur.org/DOCE/form_prensa.html
- Educ@ment. Universitat Rovira i Virgili: <http://teledoc.urv.es/educ@ment/>
- Cuadernos de Pedagogía: <http://www.cuadernosdepedagogia.com/content/Inicio.aspx>
- Eduteka: <http://www.eduteka.org/>
- Educared: http://www.fundaciontelefonica.com/educacion_innovacion/experiencias-educativas/
- INTEF: <http://educalab.es/intef>
- Redie: <http://redie.uabc.mx/index.php/redie>
- Revista RED: <http://www.um.es/ead/red/red.html>
- Editorial Graó: <http://www.grao.com/>
- @TIC: <http://www.educacontic.es/>
- Programamos: www.programamos.es
- Educación 3.0: www.educacionrespuntocero.com

La estrategia de búsqueda se ha desarrollado a partir de los descriptores que aparecen en la tabla I. Cabe destacar que, cuando ha sido posible, se han usado diferentes combinaciones terminológicas con la ayuda de booleanos:

Tabla I: Descriptores de búsqueda utilizados

Castellano	Catalán
Programación	Programació
Programación + informática	Programació + informàtica
Programación + informática + educación	Programació + informàtica + educació
Programación + informática + actividades escolares	Programació + informàtica + activitats escolars
Programación + informática + actividades extraescolares	Programació + informàtica + activitats extraescolars
Programación + informática + primaria	Programació + informàtica + primària
Programación + informática + experiencias	Programació + informàtica + experiències
Programación + informática + España	Programació + informàtica + Espanya

Sólo se han utilizado descriptores de búsqueda en castellano y catalán porque el objetivo es encontrar experiencias llevadas a cabo en España. Una vez localizados los documentos, se pasó a la selección de cada una de ellas a partir de las dimensiones que se especifican en la tabla II. Las experiencias debían describir los elementos que se mencionan en vista de qué pudieran servir de ejemplo para otros docentes:

Tabla II: Dimensiones de análisis.

Dimensión / Variable	Descripción
¿Nivel educativo?	Curso de Educación Primaria.
Metodología utilizada	Definir qué estilo de enseñanza se especifica.
Agrupamiento de los alumnos	Definir qué tipo de agrupamientos se utilizan.
Áreas en que se lleva a cabo	¿En qué áreas utilizan la programación informática?
Software utilizado	¿Qué software de programación utilizan?

Muestra: 17 documentos.

Trabajo de campo: Marzo, abril y mayo de 2015.

Presentación de los datos: En el apartado de resultados, además de enseñar los resultados del

análisis, se presentaran tres experiencias tipo, las cuales nos sirven como prototipo de sesión de enseñanza de la programación informática en diferentes cursos de Educación Primaria en diferentes contextos basándonos en las experiencias documentadas. Hemos elegido aquellas que pensamos que son más ilustrativas y que pueden animar a los docentes a utilizar la programación informática en sus clases.

3. RESULTADOS

3.1 Resultados fase I: Encuesta.

La encuesta, realizada a un total de 291 de la isla de Mallorca, recibió la respuesta de 36 centros, 29 (80,55) de ellos públicos y 7 (19,45) concertados.

De estos centros, solamente uno de ellos imparte clases de programación informática, concretamente el Centro concertado Nuestra Señora de la Consolación del Alcudia, que enseñan programación informática en el tercer ciclo de primaria en el área de informática utilizando el software de programación Scratch.

Los resultados de la encuesta nos muestran datos bastante contundentes y reveladores. Que sólo un centro de los encuestados imparta programación informática nos hace ver que se trata de un contenido que muy pocas escuelas apuestan por impartir-lo en este momento.

El siguiente paso fue entrevistar al docente Jordi Canals, maestro el cual se encarga de esta área, quien accedió a describirnos el proyecto en más profundidad.

Decidieron empezar a trabajar la programación informática el tercer trimestre de este curso de manera casi experimental principalmente en vista de que este movimiento está cogiendo mucha fuerza en el ámbito educativo.

En 4º, 5º y 6º curso de primaria han trabajado sobre qué es la programación con la ayuda de apps como Run Marco, Daysi the Dinosaur, Scratch Jr, ya que trabajan con i-pad. Además,

también desarrollan la programación informática con alumnos de último curso de Educación Infantil, utilizando los tutoriales más fácil que propone code.org.

En el tercer ciclo el proyecto consistía en que los alumnos tenían que crear videojuegos individualmente con la app Scratch Jr. Se organizó un concurso al mejor videojuego entre todos los que se programaron.

Cabe destacar que también utilizan la programación como herramienta de aprendizaje. Es decir, no tienen una asignatura específica donde se desarrolle el tema, sino que lo incorporan en otras áreas. Como consecuencia, se realiza programación informática durante una o dos horas semanales, dependiendo del curso. Su objetivo es, que aparte de que los alumnos aprendan a pensar, se trabajan otras materias como pueden ser inglés o matemáticas. También destacan que es una materia que despierta gran motivación e entusiasmo entre los niños/as.

Los agrupamientos de trabajo que se utilizan con los alumnos siempre es individual o en parejas, buscando la máxima interacción de los alumnos.

3.2 Resultados fase II: Búsqueda sistemática.

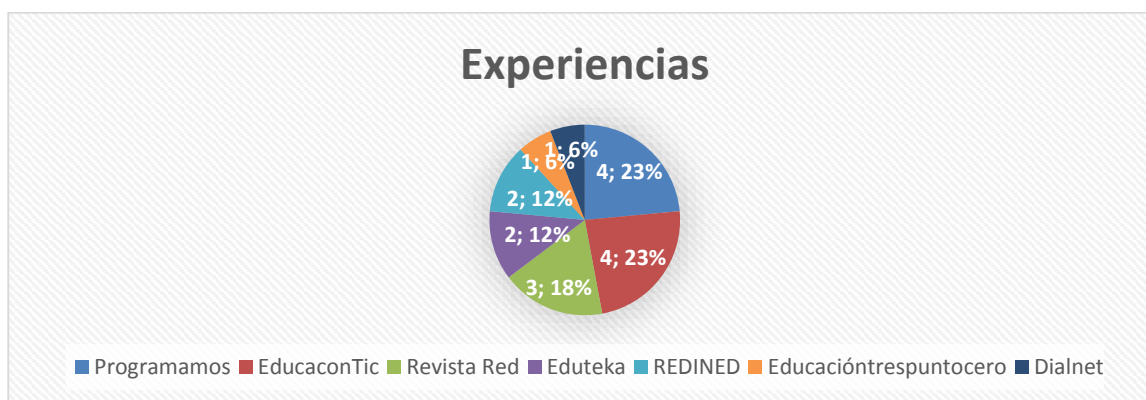
En este apartado se presentaran los resultados que hemos obtenido a partir de la búsqueda sistemática en las bases de datos, revistas y portales web especializados que antes hemos mencionado.

En primer lugar se mostraran los datos extraídos del análisis general de los documentos según las dimensiones que hemos descrito. Seguidamente se presentaran tres experiencias modelo de cómo enseñar y motivar a los alumnos a aprender programación informática de manera fácil y sin la necesidad de tener grandes conocimientos de programación. El objetivo es, a través de tres buenas prácticas, mostrar cómo de manera fácil se puede introducir la enseñanza de la programación en las aulas tanto como contenido, como herramienta de enseñanza.

3.2.1 Análisis general

En primer lugar, destacar que nuestra búsqueda se centraba en encontrar experiencias de aula documentadas, que en España no abundan. Sí que se han encontrado otras de otros países, pero nos queríamos ajustar sólo a España. Por eso la muestra de experiencias analizadas no es muy grande (17). También cabe destacar que se han utilizado aquellas que permitían un acceso libre al documento. De las bases de datos, revistas y portales web utilizados, Programamos y Educacontic son las que más experiencias nos han aportado, un 23% cada una:

GRÁFICO 0: EXPERIENCIAS ANALIZADAS.



Fuente: elaboración propia a partir de la búsqueda sistemática

A continuación se presentan los datos extraídos del análisis general que se han hecho de las 15 experiencias documentadas a partir de las dimensiones que hemos especificado antes:

3.2.1.1 Nivel educativo

En el gráfico II se observa la proporción de experiencias según el nivel educativo. Podemos observar que la mayoría se realizan en tercer ciclo de primaria, un total de 8. Sin embargo, también encontramos varias tanto en primer ciclo (3) como en segundo ciclo (4). En el apartado de experiencias encontraremos la descripción de una sesión llevada a cabo en primer curso de primaria, para ver que también se puede enseñar con los alumnos más pequeños (7-8 años).

GRÁFICO II: N° DE EXPERIENCIAS SEGÚN NIVEL EDUCATIVO

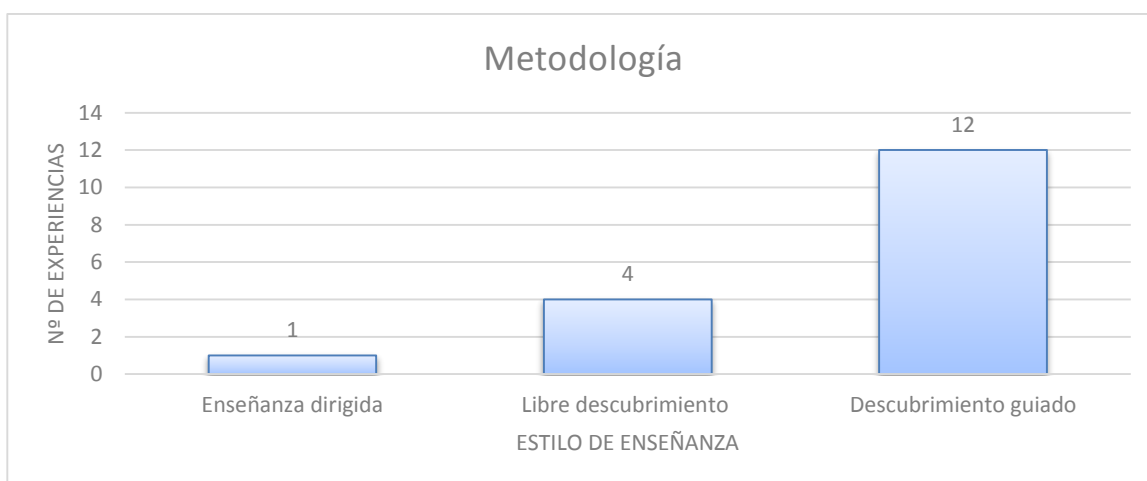


Fuente: elaboración propia a partir de la búsqueda sistemática.

3.2.1.2 Metodología

En el gráfico III se demuestra que en la mayoría de las experiencias el tutor se decanta por utilizar una metodología de descubrimiento guiado. Cabe destacar que en los primeros cursos de primaria se tiende a utilizar una metodología más dirigida. A medida que aumentamos de curso, la tendencia cambia a utilizar metodologías por descubrimiento guiado o por descubrimiento libre. También depende del tipo de actividad que se lleve a cabo.

GRÁFICO III: N° DE EXPERIENCIAS SEGÚN LA METODOLOGÍA

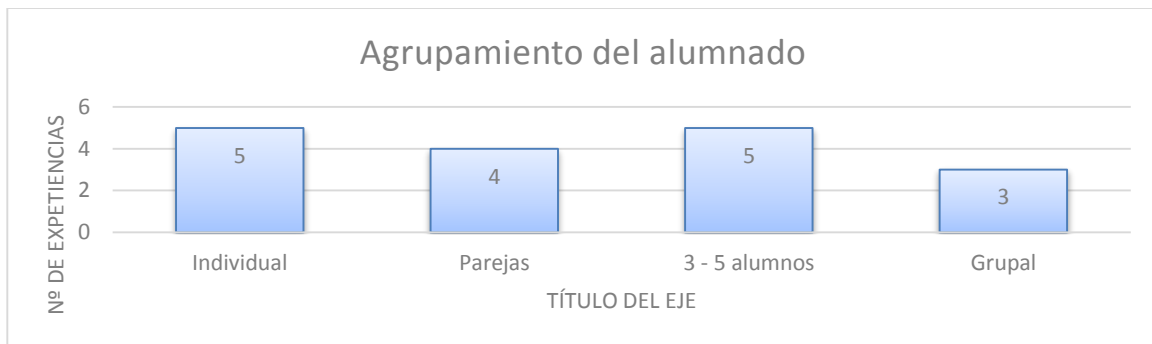


Fuente: elaboración propia a partir de la búsqueda sistemática.

3.2.1.3 Agrupamiento de los alumnos.

Los emparejamientos suelen variar según el nombre de dispositivos de los cuales disponemos o el tipo de actividad que se quiera llevar a cabo. Por este motivo, se observan resultados muy parejos entre todos los tipos en el gráfico IV.

GRÁFICO IV: N° DE EXPERIENCIAS SEGÚN AGRUPAMIENTO DEL ALUMNADO

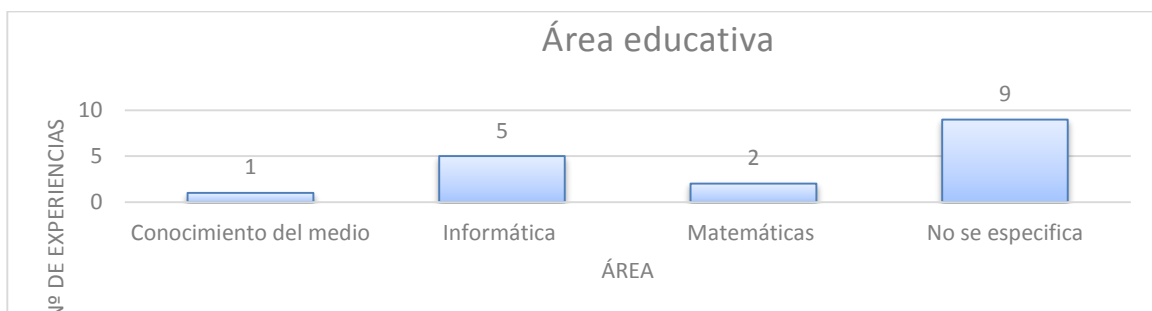


Fuente: elaboración propia a partir de la búsqueda sistemática.

3.2.1.4 Área en la que se lleva a cabo.

De esta dimensión de análisis hay que decir que no se pueden extraer ni datos ni conclusiones claras porque en la gran mayoría no se especifica el área en la que se lleva cabo. Pero sí que podemos concluir que la programación informática puede servir para enseñar cualquier contenido, ya que nos sirve como herramienta de enseñanza.

GRÁFICO V: N° EXPERIENCIAS SEGÚN EN ÁREA



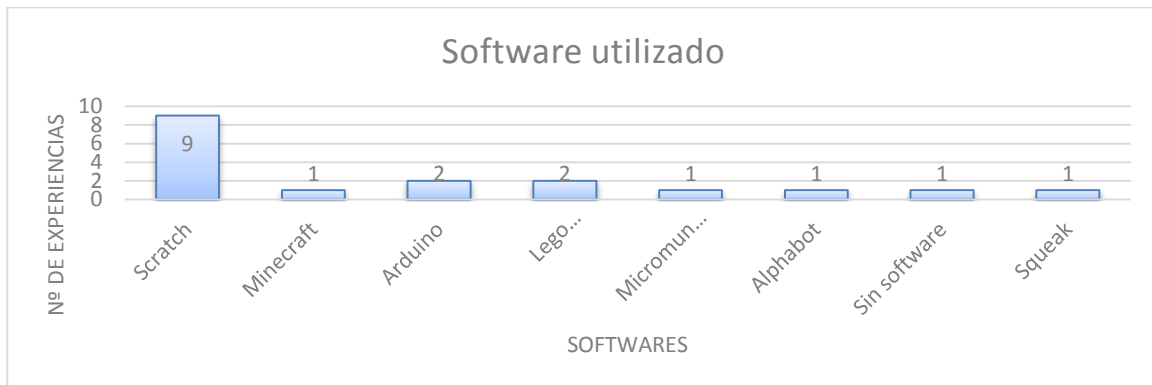
Fuente: elaboración propia a partir de la búsqueda sistemática.

3.2.1.5 Software utilizado.

En el gráfico VI podemos observar cómo un total de 13 experiencias han utilizado el software Scratch, ya sea de manera única o combinado con otros softwares. Su interfaz atractiva y su código visual y basado en bloques le hacen un software del cual es muy fácil aprender. Cabe destacar que no es imprescindible tener utilizar un soporte informático para poder aprender a

programar. Por ese motivo, también aparecen experiencias en las cuales no se utiliza un software.

GRÁFICO VI: SOFTWARE DE PROGRAMACIÓN UTILIZADO EN LAS EXPERIENCIAS



Fuente: elaboración propia a partir de la búsqueda sistemática.

3.2.2 Descripción de experiencias

Como hemos comentado anteriormente, además del análisis de los datos de la búsqueda sistemática, se van a presentar tres experiencias tipo o modelo, las cuales nos sirven como prototipo de sesión de enseñanza de la programación informática en diferentes cursos de Educación Primaria en diferentes contextos. Se han elegido estas tres buenas prácticas por su transferibilidad y su facilidad de aplicación en cualquier contexto. El objetivo es animar a los docentes y mostrar que la programación informática no es un contenido difícil de utilizar en práctica en nuestras clases ya sea como contenido a enseñar o como herramienta de enseñanza.

3.2.2.1 Experiencia en 3^r ciclo de primaria utilizando el software Minecraft (Romero, L. 2012).

Esta experiencia nos llega desde el centro Alameda de Osuna de Madrid, cuya maestra de Competencias Digitales y formadora TIC Lara Romero, lleva cabo un proyecto con los alumnos de 6^o de primaria usando Minecraft desde el curso 2012/13.

“Este proyecto tiene como objetivo desarrollar el trabajo digital en equipo, la organización espacio digital y la coordinación y velocidad óculo-manual junto con la creatividad en las construcciones que permite Minecraft (Romero L., 2013) ”. Además, también les sirve para desarrollar la competencia lingüística de los alumnos, ya que tienen que ir explicando a sus compañeros el desarrollo de su proyecto tanto de forma oral como de manera escrita.

Consiste en que los alumnos de sexto de primaria (cuatro líneas), agrupados en parejas, deberán construir una casa con el software en un trimestre. Cada uno de los alumnos tiene un rol: el que tiene más experiencia en la utilización de Minecraft desarrollará el rol de maestro o guía de otro que no conoce el juego. En primer lugar, los alumnos deben planificar el proceso: decidir qué van a construir, poner en común sus ideas y escoger la que más le guste atendiendo que al tiempo que van a tener para construirla (45 minutos semanales). El segundo paso es la propia construcción de la casa entre los dos compañeros. En tercer y último lugar, los alumnos deben grabar la presentación de su producto y exponerlo a los compañeros. En esta presentación, se debe explicar todo el proceso que se ha seguido. Durante el proyecto se usó la red social Edmodo para poner en común las experiencias y poder resolver dudas.

El proyecto se evalúa tanto desde la perspectiva del maestro como la del alumno. Por parte del maestro, la evaluación fue continua, es decir, durante todas las sesiones: el proceso de planificación, la creación, la presentación, el trabajo en equipo, la organización entre los alumnos,... Además también se evaluó el resultado final. Por parte del alumno, se valoran individualmente y en grupo en cada fase del proyecto utilizando un cuestionario.

3.2.2.2 Experiencia en 1º ciclo de primaria utilizando el software Scratch (León, J.M. 2013)

La segunda experiencia que vamos a describir nos llega de la mano de Jesús Moreno León, cofundador de la asociación Programamos, de la cual hemos hablado anteriormente en el artículo, y Alberto Molina Coballes, profesor de Ciclos Formativos de Informática en Sevilla, los cuales llevaron a cabo un taller de enseñanza del software Scratch a 80 alumnos de 2º de primaria del CEIP Ortiz de Zúñiga de Sevilla.

Su objetivo principal fue hacer ver que los ordenadores no son sólo máquinas que vienen ya hechas, sino que se pueden modificar sus esquemas de funcionamiento. En definitiva, pasar de ser consumidores a creadores.

En primer lugar, como trabajo previo al taller, pidieron a los alumnos que se inventaran un videojuego y prepararan una redacción donde se describiese su funcionamiento e inventaran los personajes, dibujos incluidos, para poder partir de sus intereses. Escogieron varias de estas ideas para preparar la base de varios videojuegos para después poder terminarlos en aula junto a los alumnos.

Una vez ya en aula, una vez se habían presentado y explicado el porqué de la sesión, escogieron un juego conocido por todos (angry birds para Scratch) para llamar su atención. Después de jugar una partida rápida, les enseñaron el código del juego, y les mostraron como si se modificaba ese código también se modificaba el comportamiento de los objetos en el juego. Los alumnos fueron proponiendo cosas a modificar, como la velocidad, el sonido, provocar explosiones,... Seguidamente, crearon tres videojuegos a partir de las ideas que ellos mismos habían ideado, mostrando a los alumnos el procedimiento. Los alumnos iban proponiendo cosas y los profesores las llevaban a cabo en la PDI para que todos lo pudieran ver. La última parte de la sesión la dirigieron a resolver las dudas que los alumnos pudieran tener.

Como actividades posteriores a la sesión, por un lado facilitaron una circular informativa a los alumnos para que le hiciesen llegar a sus padres. En esa carta se explicaba el objetivo que había tenido la sesión y se les animaba a seguir investigando con sus hijos en casa.

Por otro lado, las maestras del centro siguieron realizando tareas relacionadas con la sesión que habían llevado a cabo, como por ejemplo enviar una carta animada a un amigo y otra carta a los organizadores del taller.

En sus conclusiones nos hablan de cómo habían disfrutado en el proceso tanto los alumnos como los maestros, y de qué originales ideas aparecían entre los alumnos.

3.2.2.3 Experiencia con alumnos con NEE (discapacidad intelectual) (López, C.).

Esta tercera experiencia nos llega de la mano de Carmen López Escribano, de la Universidad Complutense de Madrid, y de Rafael Sánchez Montoya, de la Universidad de Cádiz, y nos relatan su experiencia de trabajo de la programación informática con alumnos con discapacidad intelectual en la “Feria de Realidad Virtual, Ocio Electrónico y Discapacidad” celebrada en Salamanca en el año 2007. Aunque ésta no tuvo lugar en un aula de primaria, el objetivo de los autores era animar a los docentes a ponerlo en práctica en sus clases.

Utilizaron el software de programación Scratch. En primer lugar presentaron la pantalla inicial del software a los alumnos, y les hicieron una breve descripción del mismo. Seguidamente, enseñaron a utilizar los bloques de movimiento (color azul), concretamente el bloque de “mover X pasos”. Introduciendo un número en la variable, conseguían que el objeto, en este caso un gato, se moviese tantos pasos como se quisiera. Y una vez acabada la explicación de estas instrucciones, fueron los alumnos quienes empezaron a utilizar el software.

Como nos explican los autores, no fue difícil para los alumnos llegar a la conclusión de que cuanto más grande fuera el número que introducían, el gato se desplazaba más lejos. Mientras experimentaban, a uno de los alumnos le surgió una pregunta: “¿y qué pasaría si el gato llega al borde, desaparecería o se chocaría?”. Sin buscarlo, uno de sus alumnos había construido una hipótesis de trabajo, y por tanto decidieron probarla. El gato desapareció cuando llegó al borde, ante el asombro de los niños/as. Los instructores les preguntaron en este momento: “Y ahora ¿Cómo hacemos que vuelva?”. Y a partir de aquí, comenzaron a trabajar la resolución de problemas. Pero no fueron las únicas preguntas y actividades que surgieron durante la hora

que duró la actividad. El grupo de alumnos estaba entusiasmado ya que podían controlar lo que pasaba en sus pantallas, nos cuentan los organizadores del evento.

“Lo que queremos mostrar con esta experiencia es como Scratch plantea un modo diferente de enseñanza/aprendizaje. Con Scratch el alumno es protagonista, necesita pensar, puede plantear preguntas y soluciones a esas preguntas. El aprendizaje es activo y constructivo” (López C. y Sánchez F., 2007).

4. CONCLUSIONES

A tenor de los resultados podemos deducir, por un lado, que la programación informática no se enseña en las escuelas de manera sistemática, tal y como comentábamos al principio del presente trabajo (Moreno J., 2014). La encuesta nos revela unos datos que, aunque la muestra total es de 36 centros, nos hacen tener una visión sobre la proporción de centros que sí apuestan por insertar la programación informática en sus programas de estudio.

Por otro lado, si nos fijamos y analizamos las experiencias que hemos descrito, podemos ver que no es un contenido que sea difícil de introducir en las aulas, ya que nos posibilita llevar a cabo multitud de actividades diferentes, ya sea una sesión dirigida, o un proyecto en el cual los alumnos tengan que elaborar un trabajo libremente. La programación informática se puede adaptar al tipo de actividad que queramos llevar a cabo. Incluso hemos visto que es un contenido que se puede trabajar con alumnos con necesidades educativas especiales. No sólo podemos adaptar la programación al tipo de actividad, sino que, y lo que es más importante, a las características de cada uno de los alumnos. Si nos fijamos en los datos del análisis general de las experiencias documentadas, vemos que se pueden trabajar con diferentes tipos de metodología, agrupamiento de los alumnos, en cualquier área y con diferentes tipos de software de programación.

Aunque en un primer momento pueda dar la sensación de que la programación informática es un área que utiliza complejos códigos y estructuras abstractas, los softwares de los que hemos hablado son de muy fácil aprendizaje y uso gracias ya que utilizan lenguajes

muy visuales con interfaces muy atractivas. Además, es un contenido que entusiasma a los alumnos y que se puede enseñar incluso sin la necesidad de disponer de un ordenador, tal y cómo se describe en el marco teórico.

Estonia, Finlandia, Francia o Reino Unido, de la mano de sus presidentes y ministros de educación ya se han posicionado a favor de la programación informática y ya la han introducido en programas de estudio nacionales. Incluso la Comisión Europea ya se ha posicionado al respecto. En España, aunque desde la administración se ofertan cursos de formación en la red INTEF y que en la LOMCE se promulga la importancia de la incorporación de las TIC en el currículum, la enseñanza de la tecnología y de la programación no es una prioridad (Moreno J., 2014). Prueba de ello es la denuncia que hace la Plataforma Estatal de Asociaciones del Profesorado de Tecnología en su artículo ‘*Destrucción de la Tecnología en la LOMCE (2012)*’.

Hemos podido observar, pero, que los primeros pasos desde las administraciones de cara a la incorporación de la programación al currículum de Educación Primaria ya se están dando. Pero no son suficientes. El profesorado en general empezará a introducir la programación de manera coordinada si ésta se añade al currículum de enseñanza general. Nosotros pensamos que se debe establecer un contenido sobre programación informática en el currículum cuantos antes posible, y que se implante de manera gradual, para facilitar así la formación del profesorado. Además, se debe seguir con la oferta de cursos de formación y enseñar este contenido en las Universidades.

La manera en cómo se debe enseñar queda en manos de cada uno de los docentes. A lo largo de todo el trabajo hemos descrito diversas opciones de desarrollo de la programación: desde la utilización de videojuegos como Minecraft; la elaboración de los propios videojuegos con Scratch; video tutoriales de Code.org o Code’s Cool; e incluso desarrollarla sin utilizar ordenador. Se trata de un contenido muy amplio que se puede adaptar a todas las características del contexto educativo.

Otro aspecto que queremos reseñar es la labor que hacen todas las asociaciones sin

ánimo de lucro de promoción de enseñanza de la programación informática como Code.org, Coder Dojo o Programamos. Sirven de fuente de información, de plataforma de divulgación científica, como centro de formación y de comunidad de interacción para docentes, sino que alumnos y familias. Desde aquí queremos agradecer la colaboración de Programamos, que durante todo el proceso de elaboración del artículo han estado a nuestra disposición para informarnos y formarnos en esta manera de enseñar y aprender a través del juego.

Muchos son los motivos que dan tanto la Comisión Europea, los países y las asociaciones para integrar este contenido a las aulas de Educación Primaria. Por un lado encontramos los motivos pedagógicos, entre los que destacan el desarrollo del pensamiento analítico, la resolución de problemas, el trabajo en equipo o la creatividad. Por otro lado, encontramos los motivos políticos o económicos, que ven en el campo de la programación informática y de las carreras asociadas el futuro de la industria mundial.

Nosotros creemos que los motores que nos deben conducir a la utilización de ésta como herramienta de enseñanza-aprendizaje deben ser los pedagógicos, y que nos ayuden a resolver las deficiencias del sistema educativo español de hoy en día. Creemos en una nueva forma de aprendizaje basado en el Construccinismo de Seymour Papert (1995), primer impulsor de las ideas que han generado este movimiento global, y del Pensamiento Computacional de Jeannette Wing (2006). Todos los agentes implicados apuestan por la programación informática ya que supone una nueva forma de pensar y de aprender: ‘‘ Learn to code, code to learn’’ (code.org, 2013).

5. Referencias

- Aho, A. V. (2013). Computation and Computational Thinking. (ACM, Ed.) *The computer Journal*, 42(1), pp. 38-43. Retrieved Marzo 16, 2015, from The Computer Journal: <http://comjnl.oxfordjournals.org/content/55/7/832.abstract>
- Apple. (2015). *Aprende a programar. Cambia el mundo*. Retrieved from <https://www.apple.com/es/retail/code/>
- BCS, C. a. (2012). *Computer Science: a curriculum for schools*. United Kingdom: Creative Commons Licensed. Retrieved Abril 24, 2015, from <http://www.computingschool.org.uk/data/uploads/ComputingCurric.pdf>
- Beatriz Cía López, J. L. (2014). EXPERIENCIAS DE ROBÓTICA Y PROGRAMACIÓN EN EDUCACIÓN INFANTIL. Recuperado el 29 de Abril de 2015, de [educa2.madrid.org](http://www.educa2.madrid.org): <http://www.educa2.madrid.org/documents/397121/7216583/EXPERIENCIAS+DE+PROGRAMACI%C3%93N+Y+ROB%C3%93TICA+EN+EDUCACI%C3%93N+INFANTIL/411d445a-133c-46c5-b26a-c81a9b119399>
- Blinkstein, P. (2013). *Seymour Papert's Legacy: Thinking about learning and learning about thinking*. Retrieved Febrero 28, 2015, from Transformative Learning Technologies Lab: <https://tltl.stanford.edu/content/seymour-papert-s-legacy-thinking-about-learning-and-learning-about-thinking>
- Bogliolo, A. (2014). *Il sito italiano del CodeWeek*. Retrieved from <http://codeweek.it/>
- López Escribano, C. R. S. (n.d.). Scratch y Necesidades Educativas Especiales: Programación para todos. *Revista de Educación a Distancia. RED*. (34). Retrieved MARZO 29, 2015, from <http://www.um.es/ead/red/34/scratch.pdf>
- Cellan-Jones, R. (2014, Enero 23). Are teachers ready for the coding revolution? *BBC*. Retrieved Abril 9, 2015, from <http://www.bbc.com/news/technology-25857276>
- Clements, D. H. (1986). Effects of Logo and CAI environments on cognition and Creativity. *Journal of Educational Psychology*, 78(4), 309-318. Retrieved Marzo 12, 2015, from <http://dx.doi.org/10.1037/0022-0663.78.4.309>
- Coder Dojo Foundation. (n.d.). *Coder Dojo movement*. Retrieved from [Coderdojo.com](https://coderdojo.com/about/): <https://coderdojo.com/about/>
- Comisión Europea. (2013). Apertura de la educación: Docencia y aprendizaje para todos a través de nuevas tecnologías y recursos educativos abiertos. *COMUNICACIÓN DE LA COMISIÓN AL PARLAMENTO EUROPEO, AL CONSEJO, AL COMITÉ ECONÓMICO Y SOCIAL EUROPEO Y AL COMITÉ DE LAS REGIONES*. Bruselas. Retrieved Marzo 20, 2015, from <http://eur-lex.europa.eu/legal-content/ES/TXT/PDF/?uri=CELEX:52013DC0654&from=EN>
- Coronel, M. M. (9 de Octubre de 2013). *Una experiencia con Scratch en 1º de primaria*. Recuperado el 23 de Abril de 2015, de programamos.es: <http://programamos.es/una-clase-de-programacion-con-scratch-para-ninos-de-1o-de-primaria/>
- Delval, J. (1985). *Memoria de progreso de Proyecto de Investigación : Un estudio en profundidad sobre los efectos de la utilización de los ordenadores en el desarrollo cognitivo del niño*. Centro de Investigación y Documentación Educativa. Retrieved MARZO 30, 2015, from <file:///C:/Users/Usuario/Downloads/008199200047.pdf>
- Departament d'Ensenyament Generalitat de Catalunya. (2014). *Programa mSchools*. Retrieved Marzo 15, 2015, from [xtec.cat](http://www.xtec.cat): <http://www.xtec.cat/web/projectes/mschools>

- Department of Education UK. (2013). *Computing programmes of study: key stages 1 and 2*. United Kingdom. Retrieved Abril 6, 2015, from http://www.computingschool.org.uk/data/uploads/primary_national_curriculum_-_computing.pdf
- Departamento de Educación. Gobierno de Navarra. (2015). *Código21. Tecnologías creativas*. Retrieved from <http://codigo21.educacion.navarra.es/>
- Douglas H. Clements y Sudha Swaminathana. (1985). Technology and School Change New Lamps for Old? In *Childhood Education* (Vol. 71, pp. 275-281). New York. Retrieved from <http://www.tandfonline.com/doi/abs/10.1080/00094056.1995.10522619?journalCode=uiced20>
- Douglas H. Clements y Michael T. Battista Y Julie Srama. (2001). *Journal for Reserch in Mathematics Education: Logo and Geometry*. Reston, Virginia. Retrieved Mayo 11, 2015, from <http://www.jstor.org/stable/749924> [Consultado el: 11/04/2015]
- Educación 3.0. (21 de Mayo de 2015). *Como introducir la robótica educativa en Primaria*. Recuperado el 2 de Junio de 2015, de [educaciontrespuntocero.com](http://www.educaciontrespuntocero.com/): <http://www.educaciontrespuntocero.com/experiencias/aprender-y-divertirse-con-la-robotica-educativa/26363.html>
- Fernández, J. I. (27 de Noviembre de 2013). *Programamos es una idea al sur*. Recuperado el 27 de Mayo de 2015, de programamos.es: <http://programamos.es/programamos-es-una-idea-al-sur/>
- Furber, S. (2012). 3.2.2 The educational case for Computer Science. In P. Nurse, *Shut down or restart? The way forward for computing in Uk schools* (p. 29). London, United Kingdom: Creative Commons Licensed. Retrieved Abril 18, 2015, from <https://royalsociety.org/~media/education/computing-in-schools/2012-01-12-computing-in-schools.pdf>
- García Córdoba, F. (2004). El cuestionario: Recomendaciones metodológicas para el diseño de un cuestionario. México, D.F.: Editorial Limusa, S.A. Recuperado de https://books.google.es/books?id=JPW5SSWuWOUC&pg=PA7&dq=el+cuestionario+un+instrumento+para+la+investigaci%C3%B3n+en+las+ciencias+sociales&hl=ca&source=gbs_toc_r&cad=4#v=onepage&q&f=false
- Gewerc, F. F. (2006). Profesorado y Squeak ¿Una oportunidad para romper los mitos de la tecnología en la escuela? *Revista latinoamericana de tecnología educativa*, 5(2), 465-482. Retrieved Marzo 17, 2015, from <file:///C:/Users/Usuario/Downloads/DialnetProfesoradoYSqueakUnaOportunidadParaRomperLosMitos-2229242.pdf>
- Gillespie, C. W. (2004). *La visión de Seymour Papert para la educación de la niñez: Estudio descriptivo de estudiantes de Head Start y Kindergarten en aulas basadas en el descubrimiento y enriquecidas con el lenguaje de programación Logo*. Retrieved MARzo 30, 2015, from Investigación y Práctica de la Niñez Temprana: <http://ecrp.uiuc.edu/v6n1/gillespie-sp.html>
- Gobierno de Estonia. (2013, Enero 21). *ProgeTiiger*. Retrieved Abril 13, 2015, from [hitsa.ee](http://www.hitsa.ee/ikt-haridus/progetiiger): <http://www.hitsa.ee/ikt-haridus/progetiiger>
- Gobierno de Finlandia. (2014). *Miten OPS muuttuu - ja miksi?* Retrieved Abril 18, 2015, from [Koodi2016.com](http://koodi2016.com/): <http://koodi2016.fi/ops.html>
- Gobierno de Navarra. (5 de Septiembre de 2014). *Currículo de las enseñanzas de la Educación Primaria en la Comunidad Foral de Navarra*. Recuperado el 15 de Marzo de 2015, de [navarra.es](http://www.navarra.es/): http://www.navarra.es/NR/rdonlyres/B62A9CFB-C17B-461E-BD7D-BBEE005C2096/0/F1410295_EducacionPrimaria.pdf
- Gobierno de Navarra. (2014, Abril 25). *Los centros escolares navarros enseñarán a programar software en 4o y 5o de Primaria*. Retrieved Marzo 15, 2015, from [nota de prensa](http://www.navarra.es/): <https://www.navarra.es/NR/rdonlyres/F1451720-4AE5-4782-A1EC-B7F4C523DD8/281708/centrosescolaresnavarraensenaranprogramarsoftwaree.pdf>

- Gutiérrez, I. (11 de Octubre de 2014). *Mis amigos Robot*. Recuperado el 17 de Mayo de 2015, de programamos.es: <http://programamos.es/web/wp-content/uploads/2014/10/MisAmigosRobot.pdf?6ed978>
- Hamon, B. (2014, Julio 13). Hamon : "Le code informatique à l'école dès septembre". *Le Journal du Diamanche*. (A. F. Neveux, Interviewer) Francia . Retrieved Abril 22, 2015, from <http://www.lejdd.fr/Societe/Hamon-Le-code-informatiqu-a-l-ecole-des-septembre-675912>
- Hernández Sampieri, R., Fernández Collado, C., & Pilar Baptista, L. (2010). *Metodología de la investigación*. México: Mc Graw Hill. Recuperado de <http://es.slideshare.net/Igneigna/metodologia-de-la-investigacion-5ta-edicion-de-herndez-sampieri>
- INTEF. (2014). *La programación informática como herramienta didáctica*. Recuperado el 11 de Mayo de 2015, de Instituto Nacional de Tecnología Educativa y Formación del Profesorado: <http://blog.educalab.es/intef/2014/02/26/la-programacion-informatica-como-herramienta-didactica/>
- International Society for Technology in Education (ISTE) y Computer Teachers Associations. (2011). *Operational Definition of Computational Thinking for K-12 Education*. Retrieved Marzo 3, 2015, from <http://csta.acm.org/Curriculum/sub/CurrFiles/CompThinkingFlyer.pdf>
- J. Casas Anguita, J.R. Repullo Labrador y J. Donado Campos. (2003). La encuesta como técnica de investigación. Elaboración de cuestionarios y tratamiento estadístico de los datos (II). *Atención Primaria*, 592-600. Retrieved Mayo 3, 2015, from <http://www.sciencedirect.com/science/article/pii/S0212656703792221>
- Karen Brenna, C. B. (2014). *Guia Creative Computing*. Recuperado el 23 de Mayo de 2015, de programamos.es: <file:///C:/Users/Usuario/Downloads/CreativeComputing20141015-spanish.pdf>
- Kurland, Roy D. Pea y D. Midian. (1984). On the cognitive effects of learning computer programmins. In *New ideas Psychol* (Vol. 2, pp. 137-168). New York: 1984 Pergamon Press, ltd. Retrieved Marzo 14, 2015, from http://www.tcnj.edu/~ijjims/previous/Readings/Week1/Cog_Effects_Prog.pdf
- León, J. M. (2013, Marzo 19). *Preparar un taller de programación para niños de 8 años*. Retrieved Mayo 12, 2015, from programamos.com: <http://programamos.es/preparar-un-taller-de-scratch-para-80-ninos-de-8-anos/>
- León, J. M. (2014). *¿Qué es el pensamiento computacional?* Retrieved Marzo 12, 2015, from programamos.com: <http://programamos.es/que-es-el-pensamiento-computacional/>>
- León, J. M. (18 de Julio de 2014). *Países que han introducido la programación en la escuela*. Recuperado el 26 de Marzo de 2015, de programamos.com: <http://programamos.es/paises-que-han-introducido-la-programacion-en-la-escuela/>
- León, J. M. (2014, Agosto 3). *Bruselas insta a los gobiernos europeos a promover la enseñanza de la programación en las escuelas*. Retrieved Marzo 12, 2015, from programamos.es: <http://programamos.es/bruselas-insta-a-los-gobiernos-europeos-a-promover-la-ensenanza-de-la-programacion-en-la-escuela/>
- León, J. M. (2014, Febrero 10). *Sobre el nuevo curriculum de informática en primaria en Reino Unido*. Retrieved Abril 8, 2015, from programamos.es: <http://programamos.es/sobre-el-nuevo-curriculum-de-informatica-en-primaria-en-reino-unido/>
- Madrid, I. G. (26 de Junio de 2014). *Programación y robótica educativa en primaria (I)*. Recuperado el 18 de Mayo de 2015, de profundiza.org: <http://profundiza.org/programacion-y-robotica-educativa-en-primaria-i/>
- Mateos, M. (1 de Setiembre de 2014). *Programación y educación: qué países la tienen en su plan de estudios*. Recuperado el 16 de Abril de 2015, de Genbeta.com: <http://www.genbeta.com/a-fondo/programacion-y-educacion-que-paises-la-tienen-en-su-plan-de-estudios>

- Mercedes Jimeno Badiola, A. E. (1 de Agosto de 2014). *Red experimentar y compartir*. Recuperado el 28 de Mayo de 2015, de educacontic.com: <http://www.educacontic.es/blog/red-experimentar-y-compartir>
- Nellie Krooes, A. V. (25 de Julio de 2015). *Carta conjunta a los ministros europeos*. Comisión Europea, Bruselas. Recuperado el 11 de Abril de 2015, de <http://programamos.es/bruselas-insta-a-los-gobiernos-europeos-a-promover-la-ensenanza-de-la-programacion-en-la-escuela/>
- Papert, S. (1995). Capítulo 1: Anhelantes e Instructores. In S. Papert, *La máquina de los niños. Replantearse la educación en la era de los ordenadores*. Barcelona: Paidós. Retrieved Marzo 26, 2015, from <https://es.scribd.com/doc/26568218/Papert-La-maquina-de-los-ninos#scribd>
- Ramírez, S. U. (n.d.). *INFORMÁTICA Y TEORÍAS DEL APRENDIZAJE*. Retrieved Marzo 7, 2015, from tecnologiaeducativa.com: http://tecnologiaedu.us.es/cuestionario/bibliovir/gte41.pdf
- Resnick, M. (2013). *Learn to code, code to learn*. Retrieved MARzo 23, 2015, from Edsurge: <https://www.edsurge.com/n/2013-05-08-learn-to-code-code-to-learn>
- Romero, L. (2012). *Minecraft en el Aula*. Retrieved Marzo 22, 2015, from [educacontic.com: http://www.educacontic.es/blog/minecraft-en-el-aula](http://www.educacontic.es/blog/minecraft-en-el-aula)
- Ronal J. Burke, M. C. (2007). *Women and Minorities in Science, Technology, Engineering and Mathematics*. Massachusetts, USA: Edward Elgar Publishing Limited. Retrieved Abril 13, 20015, from <http://www.elgaronline.com/abstract/9781845428884.xml>
- Schmundt, H. (2013, Mayo 16). Reading, Writing, Algorithms: Should IT Classes Be Required? *Der Spiegel*. Retrieved Abril 26, 2015, from <http://www.spiegel.de/international/germany/experts-in-germany-divided-on-computer-science-in-school-curriculum-a-899979.html>
- Tim Bell, I. H. (2008). *Computes Science Unnplugged. Un programa de extensión para niños de educación primaria*. Licencia libre. Retrieved Abril 23, 2015, from <http://csunplugged.org/wp-content/uploads/2014/12/unpluggedTeachersDec2008-Spanish-master-ar-12182008.pdf>
- Universidad Di Urbino. Departamento de Ciencias y Tecnologías de la información. (2015). *Code's Cool – Programmare è forte!* Retrieved from <http://informatica.uniurb.it/codescool/>
- Valeria de Elía, P. d. (2014). Niños creadores de tecnología. *Congreso Iberoamericano de Ciencia, Tecnología, Innovación y Educación*, (págs. 8-14). Buenos Aires. Recuperado el 19 de Mayo de 2015, de <http://www.oei.es/congreso2014/memoriactei/469.pdf>
- Vasagar, J. (2012, Enero 11). Michael Gove to scrap 'boring' IT lessons. *The guardian*. Retrieved Abril 7, 2015, from <http://www.theguardian.com/politics/2012/jan/11/michael-gove-boring-it-lessons>
- Velasco, J. J. (2014, Agosto 19). Niños programadores: para qué sirve la enseñanza de programación en las escuelas. *El diario*. Retrieved Marzo 15, 2015, from http://www.eldiario.es/turing/Ninos-programadores-ensenanza-programacion-escuelas_0_293970921.html
- Wing, J. (2006, March). Computational Thinkig. *COMMUNICATIONS OF THE ACM*, 49(3). Retrieved Marzo 29, 2015, from <https://www.cs.cmu.edu/~CompThink/papers/Wing06.pdf>