



Universidad Internacional de La Rioja
Facultad de Educación

La programación como vehículo para desarrollar las competencias clave en Educación Primaria

Trabajo fin de grado presentado por:	Marcos Darío Verdasco Cebrián
Titulación:	Grado en Maestro en Educación Primaria
Línea de investigación	Proyecto Educativo
Director:	Dr. Eduardo García Pardo

Valencia 25 de junio de 2015

A handwritten signature in blue ink, appearing to read "Marcos Darío Verdasco Cebrián", is placed over a light blue rectangular background.

Firmado por Marcos Darío Verdasco Cebrián

CATEGORÍA TESAURO: 1.1.1 Medios Audiovisuales y nuevas tecnologías aplicadas a la educación.

ÍNDICE

ÍNDICE DE FIGURAS.....	III
ÍNDICE DE TABLAS.....	IV
LISTADO DE ACRÓNIMOS.....	V
RESUMEN.....	VI
PALABRAS CLAVE.....	VI
AGRADECIMIENTOS.....	VII
CAPÍTULO 1: INTRODUCCIÓN.....	1
1.1. MOTIVACIÓN.....	2
1.2. DEFINICIONES PREVIAS.....	3
1.3. MARCO LEGISLATIVO.....	5
1.4. METODOLOGÍA.....	6
1.5. ESTRUCTURA DE LA MEMORIA.....	7
CAPÍTULO 2: OBJETIVOS.....	9
2.1. OBJETIVO GENERAL.....	9
2.2. OBJETIVOS ESPECÍFICOS.....	9
CAPÍTULO 3: MARCO TEÓRICO.....	11
3.1. HISTORIA DE LA PROGRAMACIÓN.....	11
3.2. HISTORIA DE LOS LENGUAJES DE PROGRAMACIÓN PARA EDUCACIÓN.....	12
3.3. EXPERIENCIAS PREVIAS SOBRE INCLUSIÓN DE LA PROGRAMACIÓN Y MICROELECTRÓNICA EN ENTORNOS EDUCATIVOS.....	13
3.4. PROYECTOS GUBERNAMENTALES.....	16
3.5. INICIATIVAS PRIVADAS PARA PROMOVER LA PROGRAMACIÓN EN LA EDUCACIÓN.....	17
CAPÍTULO 4: PROYECTO EDUCATIVO.....	19
4.1. INTRODUCCIÓN.....	19
4.2. OBJETIVOS EDUCATIVOS.....	19
4.3. ACTIVIDADES PROPUESTAS.....	20
4.3.1 Metodología.....	21
4.3.2 Actividades formativas.....	22
4.4. TEMPORALIZACIÓN.....	34
4.5. EVALUACIÓN DEL ALUMNADO.....	35
4.6. EVALUACIÓN DEL PROYECTO EDUCATIVO.....	36
CAPÍTULO 5: LIMITACIONES, CONCLUSIONES Y PROSPECTIVAS.....	39
5.1 LIMITACIONES.....	39

5.2. CONCLUSIONES.....	40
5.2.1. Conclusiones del Trabajo Fin de Grado.....	40
5.2.2. Conclusiones del Proyecto Educativo.....	41
5.3. PROSPECTIVA.....	41
REFERENCIAS BIBLIOGRÁFICAS.....	43
BIBLIOGRAFÍA.....	46

ÍNDICE DE FIGURAS

Figura 3.1. Modelo de aprendizaje según Resnick.....	14
Figura 3.2 : Sistema CHERP de programación mediante bloques inteligentes.....	15
Figura 4.1. Programación necesaria para hacer aparecer cada elemento de la presentación.....	24
Figura 4.2. Programa que ejecuta el cambio de disfraz y reproduce la explicación.....	25
Figura 4.3. Programa que permite reconocer si un objeto ha sido pulsado por el ratón.....	26
Figura 4.4. Programación necesaria para crear un juego de preguntas y respuestas.....	27
Figura 4.5. Programa necesario para que al pulsar un botón se muestre una imagen y reproduzca una explicación.....	28
Figura 4.6. Ejemplo de la programación a desarrollar para 3 preguntas.....	29
Figura 4.7. Esquema de conexión básico en una placa protoboard.....	30
Figura 4.8. Placa Arduino Mega conectada a motor y LED.....	31
Figura 4.9. Programa que permite controlar el ascensor con el teclado.....	31
Figura 4.10. Programa que permite obtener datos y operar con ellos.....	32
Figura 4.11. Programa necesario para juego de preguntas y respuestas con temporizador.....	33

ÍNDICE DE TABLAS

Tabla 1.1. Cronograma metodológico.....	6
Tabla 4.1. Ejes de coordenadas cartesianas y desplazamiento en Scratch.....	24
Tabla 4.2. Eventos «On clic» en Scratch.....	25
Tabla 4.3. Introducción a las condiciones y bucles en Scratch.....	26
Tabla 4.4. Introducción a las variables en Scratch (Variables I).....	27
Tabla 4.6. Contenedores y variables (II) en Scratch.....	29
Tabla 4.7. Circuitos eléctricos con protoboards.....	30
Tabla 4.8. Microelectrónica con Bitbloq.....	31
Tabla 4.9. Operaciones aritméticas con Scratch.....	32
Tabla 4.10. Preguntas, respuestas, contadores, LED y botón.....	33
Tabla 4.11. Cronograma para las actividades formativas programadas en 5º curso de EP.....	34
Tabla 4.12. Cronograma para las actividades formativas programadas en 6º curso de EP.....	34
Tabla 4.13. Rúbrica de evaluación del Proyecto Educativo.....	36

LISTADO DE ACRÓNIMOS

- **ALGOL:** del inglés *Algorithmic Language* (Lenguaje algorítmico).
- **BASIC:** del inglés *Beginners All-purpose Symbolic Instruction Code* (Código de instrucción simbólico «multipropósito» para principiantes).
- **CAA:** Competencia en Aprender a Aprender.
- **CCEC:** Competencia en Conciencia y Expresiones Culturales.
- **CD:** Competencia Digital.
- **CHERP:** del inglés *Creative Hybrid Environment for Robotic Programming* (Entorno híbrido creativo para la programación robótica).
- **CCL:** Competencia en Comunicación Lingüística.
- **CMCT:** Competencia Matemática y competencias básicas en Ciencia y Tecnología.
- **COBOL:** del inglés *Common Business Oriented Language* (Lenguaje orientado a negocios corrientes).
- **CSC:** Competencias Sociales y Cívicas.
- **EP:** Educación Primaria.
- **FORTTRAN:** del inglés *Formula Translation* (traducción de fórmula).
- **LLL:** del inglés *Long Life Learning* (aprendizaje a lo largo de toda la vida).
- **LOE:** Ley Orgánica de la Enseñanza de 2006.
- **LOMCE:** Ley Orgánica 8/2013, de 9 de diciembre, para la Mejora de la Calidad Educativa
- **MIT:** del inglés *Massachusetts Institute of Technology* (Instituto Tecnológico de Massachusetts).
- **PE:** Proyecto Educativo.
- **S4A:** del inglés, *Scratch for Arduino* (Scratch para Arduino).
- **SIEE:** Sentido de Iniciativa y Espíritu Emprendedor.
- **STEM:** del inglés, *Science, Technology, Engineering and Mathematics* (Ciencias, Tecnología, Ingeniería y Matemáticas).
- **TFG:** Trabajo Fin de Grado.

RESUMEN

A nadie le pasa por alto que la tecnología impregna la cotidianidad de la sociedad, estando cada vez más presente en todos los aspectos de la vida. Los alumnos aprenden a usar aplicaciones web, paquetes de ofimática, crean blogs e incluso participan en proyectos colaborativos tipo wiki. Se trata, sin duda alguna, de nativos digitales. Ser nativo digital debe implicar, necesariamente, estar capacitado para producir contenido digital y, por ello, se ha de promover el cambio de rol, pasando de meros consumidores a productores. Sin embargo, el sistema educativo únicamente forma usuarios. Por este motivo, el presente trabajo plantea un proyecto educativo transversal que favorece la incorporación de la programación y la microelectrónica en el sistema educativo, en los dos últimos años de Educación Primaria.

PALABRAS CLAVE: Educación Primaria, programación, Scratch, Arduino, educación transversal.

AGRADECIMIENTOS

A Iván, Rubén y Eric por entender que su padre pase tanto tiempo delante del ordenador. A mi Alma, que siempre me ha empujado cuando lo necesitaba. A mis padres, por estar siempre que se les necesita. Gracias a todos por esperar estos años a que complete un nuevo proceso formativo.

CAPÍTULO 1: INTRODUCCIÓN

La comunicación es una capacidad que transforma al animal en persona. Esta capacidad convierte al animal en ser social y, por ende, le permite vivir en sociedad. A lo largo de la historia de la Humanidad han existido diversas culturas y civilizaciones caracterizadas por su lengua. Lenguas que han evolucionado o que simplemente han desaparecido. Lenguas que, en última instancia, han permitido la comunicación entre las personas. Sin embargo, desde la aparición de las máquinas programables, más conocidas como computadoras, «microcontroladores» o simplemente ordenadores, ha surgido una nueva necesidad, la necesidad de comunicarse con las máquinas para programar su funcionamiento. Así es cómo nacen los lenguajes de programación, como respuesta a la necesidad de programar el funcionamiento de máquinas que dan respuestas previamente establecidas ante los «estímulos» adecuados.

A esta necesidad de comunicación con las máquinas, se une el hecho de que el alumnado de Educación Primaria (EP) es un alumnado nacido en un mundo gobernado por la tecnología. Como acuñó Marc Prensky (2001), son «nativos digitales». En este sentido, los «nativos digitales» aprenden el manejo de la tecnología casi de forma autónoma. No obstante, el contexto actual les conduce a ser meros consumidores, hecho que no otorga una ventaja, sino que genera una dependencia hacia lo que otros producen. Ser consumidores significa pasividad, significa aceptar las soluciones externas sin buscar las propias. Por tanto, la profesión docente no puede permitir que el alumnado abandone el sistema educativo como consumidores. Es necesario realizar un esfuerzo para que se produzca el cambio en el paradigma de consumidor pasivo a productor activo. Por ambos motivos, es necesario que los denominados «nativos digitales» comprendan los mecanismos que permiten el funcionamiento de las tecnologías y, para ello, el conocimiento de la lógica de la programación es un gran aliado.

Como consecuencia de esta necesidad, este Trabajo Fin de Grado (TFG) plantea un Proyecto Educativo (PE) transversal en el que la enseñanza de la programación sirve como vehículo para alcanzar los objetivos propios de Ciencias Naturales y Ciencias Sociales en 5º y 6º de EP. Así, de forma transversal, una enseñanza que no aparece reflejada de forma explícita en el currículo español, puede integrarse en la programación de aula mediante la elaboración de proyectos puntuales. Se persigue por tanto, aglutinar los objetivos propios de las materias antes mencionadas, con los conocimientos que permitan la introducción de la programación como una

herramienta y, por tanto, como un medio y no como un fin, para el desarrollo de las competencias clave.

Tras esta breve introducción, el trabajo presenta las razones que motivan la elaboración de este PE. Después, se detallan, a modo de resumen, las definiciones previas que permiten al lector desambiguar el significado de términos usados en el texto. Finalmente, se hace mención al marco legislativo que da soporte a esta propuesta en el Estado Español.

1.1. MOTIVACIÓN

Tal y como se ha esbozado anteriormente, dos de los elementos esenciales que constituyen la motivación de este TFG son: la necesidad de comunicación entre el ser humano y las máquinas; y convertir a los denominados «nativos digitales» en productores y no en meros consumidores. No obstante, existen otras motivaciones adicionales para el desarrollo de este TFG.

Desde una perspectiva histórica, en 1936, Alan M. Turing publica un trabajo teórico donde postula las especificaciones que debería cumplir un ordenador capaz de realizar cálculos de cualquier tipo. Así, su diseño teórico conocido como *Máquina Universal de Turing*, sentó las bases para el posterior desarrollo de los lenguajes de programación. Dos años después, Bernard Zuse fabrica la primera computadora digital programable. A raíz de sus trabajos, el mundo cambia de forma radical. La tecnología está presente en todos los aspectos de la vida y ofrece posibilidades antes inimaginables. Dispositivos electrónicos como «smartphones» y «tablets» inundan las casas de nuestro país. De hecho, según indica el Instituto Nacional de Estadística (2014) el 74% de los hogares españoles disponen de acceso a Internet. Pero dicho informe va más allá, ya que muestra que el 90,7% de los niños y niñas españolas de 10 años utilizan el ordenador y el 89% utilizan Internet. Visto así, no cabe sino plantearse la necesidad de formar a estos “*nativos digitales*” (Prensky, M., 2001) en la comprensión de las tecnologías que utilizan cada día.

Conocer las tecnologías permite al usuario sacar partido de ellas, sin embargo se debe ir más allá, es necesario enseñar al alumnado a programar. Comprender conceptos básicos y, después aprender a programar, permite la comunicación con la tecnología y decidir su funcionamiento. Sin embargo, el objetivo a tener en cuenta no puede ser el mero aprovechamiento de la tecnología. El objetivo debe ser el desarrollo de las capacidades de los niños, capacidades expresadas en forma de competencias clave que tal y como indica marco legislativo vigente, deben ser trabajadas de forma transversal en todas las materias o áreas durante la EP.

Así pues, el presente trabajo plantea una programación de aula que pretende el desarrollo del alumnado en aspectos tales como la capacidad de atención, autonomía y creatividad (Douglas H., 1986), capacidad para la resolución de problemas, desarrollo de la imaginación y la originalidad (Pardamean y Bens, 2011) y, finalmente, favorecer la sociabilización y el trabajo en grupo (Dasgupta y Sayamindu, 2014).

1.2. DEFINICIONES PREVIAS

En este apartado se presentan definiciones que facilitan la comprensión de este trabajo. Inicialmente, se presentan definiciones relacionadas con la tecnología y posteriormente, se procede a describir un segundo grupo de términos relativos a educación.

- **Arduino:** plataforma física de computación de código abierto basada en una sencilla tarjeta «microcontroladora» que permite al usuario experimentar con el mundo físico mediante sensores y «accionadores». Existen diversas versiones de Arduino, comenzando por Arduino Uno, el primer modelo en ver la luz o Arduino Mega, desarrollada para ofrecer un mayor número de conexiones.
- **Lenguaje de programación:** lengua diseñada para el desarrollo de programas informáticos que son ejecutados en máquinas programables.
- **Nativo digital:** persona nacida y desarrollada rodeada por la tecnología y que se desenvuelve con soltura en esta nueva sociedad regida por la misma.
- **Programación:** término que hace referencia a programación informática. La programación se define como los procesos de diseño, codificación, depuración y mantenimiento de un código fuente o texto escrito en un lenguaje de programación que tras su compilación dan lugar a lo que se conoce como *software*.
- **Protoboard:** también conocidas como placas de prototipado. Se trata de placas para la conexión rápida de elementos electrónicos. Así pueden realizarse prototipos electrónicos sin necesidad de realizar soldaduras. Son frecuentemente utilizados como complemento de Arduino y otros proyectos para el aprendizaje de microelectrónica y robótica.
- **S4A:** *software* libre basado en Scratch (ver definición más abajo) diseñado para programar e interaccionar con Arduino.

- **Scratch:** *software* libre desarrollado por el MIT (*Massachusetts Institute of Technology*) que da acceso a un entorno de programación gráfico pensado para la enseñanza de la lógica de la programación.
- **Script:** también conocido como “Archivo de procesamiento por lotes” es un archivo informático que almacena órdenes que determinan la ejecución secuencial de programas almacenados en el sistema operativo.

A continuación se presentan las definiciones de objetivos, competencias, contenidos y criterios de evaluación extraídos literalmente del Real Decreto 126/2014, de 28 de febrero, por el que se establece el currículo básico de la EP.

- **Objetivos:** logros que el alumno debe alcanzar al finalizar el proceso educativo, como resultado de las experiencias de enseñanza-aprendizaje intencionalmente planificadas.
- **Competencias:** capacidades para aplicar de forma integrada los contenidos propios de cada enseñanza y etapa educativa, con el fin de lograr la realización adecuada de actividades y la resolución eficaz de problemas complejos. Las competencias establecidas en el currículo de EP son: Competencia en Comunicación Lingüística (CCL), Competencia Matemática y competencias básicas en Ciencia y Tecnología (CMCT), Competencia Digital (CD), Competencia en Aprender a Aprender (CAA), Competencias Sociales y Cívicas (CSV), Competencia en Sentido de Iniciativa y Espíritu Emprendedor (CSIEE) y, finalmente, Competencia en Conciencia y Expresiones Culturales (CCEC).
- **Contenidos:** conjunto de conocimientos, habilidades, destrezas y actitudes que contribuyen al logro de los objetivos de cada enseñanza y etapa educativa y a la adquisición de competencias. Los contenidos se ordenan en asignaturas, que se clasifican en materias, ámbitos, áreas y módulos en función de las enseñanzas, las etapas educativas o los programas en que participe el alumnado.
- **Criterios de evaluación:** son el referente específico para evaluar el aprendizaje del alumnado. Describen aquello que se quiere valorar y que el alumnado debe lograr, tanto en conocimientos como en competencias; responden a lo que se pretende conseguir en cada asignatura.

- **Metodología didáctica:** conjunto de estrategias, procedimientos y acciones organizadas y planificadas por el profesorado, de manera consciente y reflexiva, con la finalidad de posibilitar el aprendizaje del alumnado y el logro de los objetivos planteados.

1.3. MARCO LEGISLATIVO

El presente trabajo se desarrolla dentro del marco legislativo propugnado por la Ley Orgánica 8/2013, de 9 de diciembre, para la mejora de la calidad educativa (LOMCE) que modifica el anterior marco legislativo definido en la Ley Orgánica de la Enseñanza de 2006 (LOE). Además, el currículo de EP es desarrollado por el Real Decreto 126/2014, de 28 de febrero, por el que se establece el currículo básico de la Educación Primaria. Finalmente, el marco legislativo queda definido por la Orden ECD/65/2015, de 21 de enero donde se definen las relaciones entre las competencias y los contenidos a desarrollar en EP.

Es el marco legislativo el que da soporte al desarrollo de este proyecto. Así, este nuevo contexto educativo, se caracteriza entre otros aspectos, por dotar al profesorado de mayor autonomía para lograr una mayor personalización de la educación. El R.D. 126/2014 especifica que sea el docente el que diseñe actividades enfocadas a desarrollar las competencias clave. Esto, sin duda, da pie a planificar actividades con el propósito de alcanzar los objetivos establecidos en el currículo. Se trata de una apuesta por la personalización de la enseñanza que permite plantear, sin lugar a dudas, este proyecto educativo en el que la programación informática es motor y guía del desarrollo de las competencias. También se encuentran referencias claras a la incorporación de las TIC en la enseñanza en el preámbulo de la LOMCE donde se expresa la necesidad de desarrollar en el alumnado la creatividad, el pensamiento crítico y la autoconfianza, aspectos que como se demuestra en este trabajo, se asocian al aprendizaje de algún lenguaje de programación.

Finalmente, el artículo 8 de la LOMCE, estipula la necesidad de incorporar las TIC en las aulas como herramienta transversal de desarrollo personal. Por tanto, se ha de concluir que no solo es lícito sino necesario plantear un proyecto en el que la programación informática sea el eje vertebrador para el desarrollo de las competencias clave.

1.4. METODOLOGÍA

La realización del presente TFG se lleva a cabo mediante un proceso escalonado de trabajo. Primero, se plantean los objetivos del TFG. A continuación, se realiza una revisión bibliográfica con el objetivo de contrastar y ampliar las ideas que, a priori, se tenían sobre el estado de la cuestión. Para realizar esta búsqueda, se opta por utilizar Internet como fuente de información. Sin embargo, los recursos accesibles en Internet no siempre están debidamente referenciados. Por este motivo, en el proceso de revisión bibliográfico, se opta por seleccionar artículos de revistas de investigación educativa así como otras fuentes debidamente referenciadas como conferencias TED, páginas web del Ministerio de Educación, Cultura y Deporte, libros, etcétera. Con el objetivo de garantizar la objetividad y veracidad de la información analizada, se elige Google Académico¹ como sistema primordial de búsqueda de información. Una vez seleccionados los artículos y tras una primera lectura rápida, se procede a su descarte o inclusión en un gestor de referencias bibliográficas (*Refworks*²). Posteriormente, se realiza una lectura en profundidad de los artículos seleccionados y, tras la adquisición de los conocimientos teóricos necesarios, se redacta el marco teórico. Una vez desarrollado el marco teórico y, teniendo en mente tanto las experiencias analizadas en la bibliografía consultada como la legislación vigente, se procede al desarrollo del PE. Dicho PE se desarrolla tomando como referentes los objetivos, contenidos y competencias establecidos en la LOMCE para después programar las actividades. Finalmente, se acomete el desarrollo de los mecanismos de evaluación que guían la evaluación del proceso de enseñanza aprendizaje. A continuación y, antes de analizar la estructura de la memoria en la Sección 1.4, en la Tabla 1.1 se presenta un sencillo cronograma en el se reflejan las diferentes etapas metodológicas seguidas durante la elaboración de este Proyecto.

Temporalización	Etapas
Febrero y marzo de 2015	Reflexión sobre ideas previas.
Abril de 2015	Desarrollo de objetivos iniciales.
Abril de 2015	Recopilación y selección de información.
Principios de mayo de 2015	Lectura en profundidad y descarte de material.
Mediados de mayo de 2015	Desarrollo del Marco Teórico.
Principios de junio de 2015	Desarrollo de PE.
Mediados de junio de 2015	Revisión y reformulación del PE.
Mediados de junio de 2015	Desarrollo del sistema de evaluación del PE.
Finales de junio de 2015	Revisión del PE en conjunto.

Tabla 1.1. Cronograma metodológico.

1 Puede acceder a Google Académico en <https://scholar.google.es/>

2 Puede acceder a Refworks en <https://www.refworks.com/>

1.5. ESTRUCTURA DE LA MEMORIA

En esta sección, se presenta al lector de forma clara y concisa, la estructura general de este TFG. Dicha estructura, se desarrolla a lo largo de cinco capítulos que a continuación se desglosan.

A lo largo del Capítulo 1 se han presentado, a modo de introducción, los aspectos que han motivado el desarrollo de este TFG, así como aquellas definiciones previas necesarias para desambiguar algunos conceptos presentados en este trabajo. Además, en este capítulo, se presentan el contexto educativo, la metodología seguida y la estructura de la memoria. Ya en el Capítulo 2, se define el objetivo general así como los objetivos específicos que guían el desarrollo de este TFG. Por su parte, el Capítulo 3 presenta el marco teórico, en el que se esboza a grandes rasgos, la historia de la programación y las experiencias y proyectos en los que se promueve la incorporación de la programación en la escuela. Una vez analizado el Marco Teórico, en el Capítulo 4, se expone el PE, en el que se presentan los objetivos educativos, la secuencia de actividades, su temporalización y su evaluación. Finalmente, en el Capítulo 5, se exponen las conclusiones y perspectivas derivadas de la realización de este TFG.

CAPÍTULO 2: OBJETIVOS

El presente capítulo se dedica a presentar los objetivos de este TFG. A continuación, se procede a presentar el objetivo general (ver Sección 2.1) para después, en la Sección 2.2 desglosar este en sus objetivos específicos.

2.1. OBJETIVO GENERAL

El objetivo general de este TFG podría ser enunciado como:

«Crear un proyecto educativo basado en la enseñanza de la programación, que permita el desarrollo de las competencias clave establecidas en el currículo oficial del Estado Español».

La idea que subyace a este objetivo es el desarrollo de una guía que pueda ser utilizada por profesores del sistema educativo español para introducir la programación y la «microelectrónica» en las aulas de EP. Esta guía pretende facilitar un camino hacia la inclusión de la programación como herramienta pedagógica. Entendiendo la programación no como un fin, sino como un medio, para el desarrollo de los contenidos y competencias fijadas por el Gobierno en el currículo oficial.

2.2. OBJETIVOS ESPECÍFICOS

Para la consecución del objetivo general, se plantean los siguientes objetivos específicos:

1. Estudiar la historia de los lenguajes de programación.
2. Presentar una reseña histórica centrada en los lenguajes de programación orientados a la educación.
3. Desarrollar un marco teórico que de soporte teórico a este proyecto educativo.
4. Precisar los objetivos de enseñanza de este proyecto educativo.
5. Definir los contenidos del currículo oficial implicados en este proyecto educativo.
6. Programar una temporalización adecuada a los contenidos a planteados.
7. Preparar las acciones educativas conducentes a la consecución de los objetivos didácticos desarrollados en este proyecto.

8. Desarrollar los criterios de evaluación.
9. Implementar las herramientas necesarias para evaluar el proceso de enseñanza y aprendizaje.
10. Obtener las conclusiones y plantear una perspectiva para la implementación del proyecto educativo.

Una vez establecido el objetivo general y los objetivos específicos de este PE, es necesario realizar una revisión de los conocimientos teóricos existentes en el panorama científico actual. Por ello, en el Capítulo 3 se presenta el marco teórico que dota de apoyo científico a este PE.

CAPÍTULO 3: MARCO TEÓRICO

En este capítulo se presenta al lector una visión general de aquellos aspectos relevantes que dan soporte teórico al desarrollo de este PE. Así, primero (ver Sección 3.1) se describe brevemente la historia de la programación y aquellos lenguajes utilizados en la educación (ver Sección 3.2). A continuación, en la Sección 3.3, se presentan diversos estudios científicos que relacionan la programación con el desarrollo de competencias clave. Finalmente, en las Secciones 3.4 y 3.5 se da lectura a proyectos gubernamentales e iniciativas privadas, desarrolladas con el objetivo de incorporar la programación en la educación reglada.

3.1. HISTORIA DE LA PROGRAMACIÓN

El desarrollo de las primeras computadoras programables lleva acompasado el necesario desarrollo de un lenguaje específico para poder comunicarse con cada uno de sus componentes. A estos lenguajes se les conoce como lenguajes máquina. Con la aparición de los circuitos integrados, los lenguajes de programación evolucionan hacia el desarrollo de lenguajes simbólicos. Lenguajes que permiten programar con independencia de la configuración física de la máquina. Así, tras unos primeros intentos poco afortunados de desarrollar un lenguaje de programación de propósito general, nace, en 1957, el lenguaje FORTRAN (*Formula Translation*). Dos años después, ve la luz COBOL (*Common Business Oriented Language*) y casi de forma simultánea, en 1960, son presentados los *Report Program Generator* y ALGOL (*Algorithmic Language*).

Posteriormente, se inicia una nueva etapa caracterizada por el desarrollo de lenguajes de programación universales. Estos desarrollos, que pretenden conglomerar lo mejor de los lenguajes FORTRAN y ALGOL, terminan por plasmarse en el primer lenguaje de programación diseñado para ser usado por cualquier tipo de usuario, independientemente de su formación académica. Nace así, en 1964, BASIC (*Beginners All-purpose Symbolic Instruction Code*). Más adelante, en 1970, es presentado el lenguaje de programación C, diseñado para ser utilizado bajo sistemas operativos UNIX. Este lenguaje y sus posteriores revisiones suponen a día de hoy uno de los lenguajes de programación más extendidos en el mundo de la programación.

Finalmente, la década de los 90 y principios del siglo XXI se caracteriza por la aparición de nuevos lenguajes. Lenguajes como Java o *Visual Basic .NET* desarrollados para campos específicos y que

han obtenido gran aceptación en todo el mundo, así como otros lenguajes ampliamente extendidos (muchos de ellos orientados a la programación web) como PHP, Javascript o Python.

3.2. HISTORIA DE LOS LENGUAJES DE PROGRAMACIÓN PARA EDUCACIÓN

BASIC podría ser considerado el primer lenguaje de programación pensado y desarrollado para ser usado por principiantes. Sin embargo, la aparición de BASIC viene seguido de nuevos lenguajes diseñados para la enseñanza. En 1967, de la mano del MIT, nace el lenguaje de programación Logo, basado en la lógica constructivista de Piaget y con un objetivo claro, acercar la programación a los más jóvenes. Más adelante, en 1970 el profesor suizo Niklaus Wirth desarrolla con fines pedagógicos el lenguaje Pascal. Este nuevo desarrollo presenta un lenguaje estructurado, que permite definir variables y funciones de forma previa a su utilización dentro del propio programa, lo que sin duda, favorece la comprensión del código desarrollado. Por su parte, el MIT, en colaboración con la compañía de juguetes LEGO, desarrolla sendas versiones de Logo en 1986 y posteriormente en 1993. Estas nuevas versiones se caracterizan por presentar un entorno de programación gráfico basado en bloques que pueden ser arrastrados y apilados para desarrollar el programa. De esta forma, el MIT implementa un sistema de programación donde no existen problemas de sintaxis puesto que no hay que escribir código. Este nuevo desarrollo, utilizado para programar la primera generación de robots LEGO, se le conoce como *Logo Bricks*.

En 2003, tras varios años de desarrollo, el MIT presenta la primera versión de escritorio de *Scratch*, un lenguaje de programación gráfico basado en Squeak (considerado la evolución de *Smalltalk* el cual es el primer lenguaje de programación orientado a objetos) y *Logo Bricks* que permite desarrollar programas sin escribir código. La potencia de *Scratch* reside en su versatilidad y la eliminación de la barrera sintáctica que supone el aprendizaje de un lenguaje de programación estándar (Idlbi, 2009). Tras la irrupción de *Scratch* en el panorama mundial, se desarrollan otros lenguajes de programación basados en bloques. Como por ejemplo, Kodu, desarrollado por Microsoft en 2009. *Kodu* es una iniciativa gratuita que pretende fomentar el desarrollo de videojuegos en plataformas *Windows*. Gracias a estas herramientas y, en concreto a *Scratch*, puede introducirse la programación a partir de los 9 años sin que se produzca el rechazo que lleva acompasado el aprendizaje de lenguajes de programación de bajo nivel (Idlbi, 2009).

Finalmente, cabe destacar las iniciativas *Scratch for Arduino* (S4A) y AppInventor. Ambas aplicaciones derivadas de Scratch y que posibilitan programar placas «microcontroladoras» Arduino y dispositivos Android respectivamente.

3.3. EXPERIENCIAS PREVIAS SOBRE LA INCLUSIÓN DE PROGRAMACIÓN Y MICROELECTRÓNICA EN ENTORNOS EDUCATIVOS

Antes de presentar los estudios más representativos que tratan sobre la inclusión de la programación y la microelectrónica en entornos educativos, es necesario presentar las evidencias que apoyan el juego como vía de aprendizaje. De hecho, existen evidencias desde los años 90 que demuestran que cuando los niños juegan, crean conocimiento, es decir aprenden (Dau, 1999 y Levin, 1996 citados por Pramling y Asplund, 2008). Pramling y Asplund, citando a Glover (1999) presentan evidencias que demuestran que jugar con compañeros promueve el desarrollo del autocontrol, la cooperación y la socialización. Todas ellas, actitudes que deben ser desarrolladas en nuestros alumnos de acuerdo con la LOMCE. Además, al ser preguntados sobre lo que más les gusta hacer, la respuesta de los niños es unánime: “Jugar” (Pramling y Asplund, 2008).

Por su parte, Mitchel Resnick del MIT defiende como modelo de aprendizaje para toda la vida aquel empleado en las guarderías, es decir el aprendizaje mediado por el juego, en el que el hecho de jugar es tan sólo una de las etapas en el proceso de aprendizaje (Resnick, 2007). Este proceso de aprendizaje, ilustrado en la Figura 3.1, muestra como aprender es tan solo la consecuencia de un proceso cíclico en el que se comienza por *imaginar* para después *crear, jugar, compartir y reflexionar*. Se trata pues, de un modelo en espiral en el que la reflexión da pie a un nuevo ciclo de aprendizaje al ser invocado el proceso imaginativo.

Tras presentar este modelo de aprendizaje basado en el juego, es necesario analizar las experiencias previas que han demostrado la idoneidad del aprendizaje de la programación como medio para el desarrollo de las competencias clave. Desde la aparición en el mundo educativo de Logo, se suceden múltiples proyectos de investigación. El objetivo principal de estos proyectos, es el de comprobar los beneficios de su incorporación al sistema educativo. Así, se demuestra que el aprendizaje de lenguajes de programación favorece el desarrollo de habilidades y actitudes así como la adquisición de conceptos. A continuación, se presentan en orden cronológico, los proyectos de investigación más destacables a este respecto.

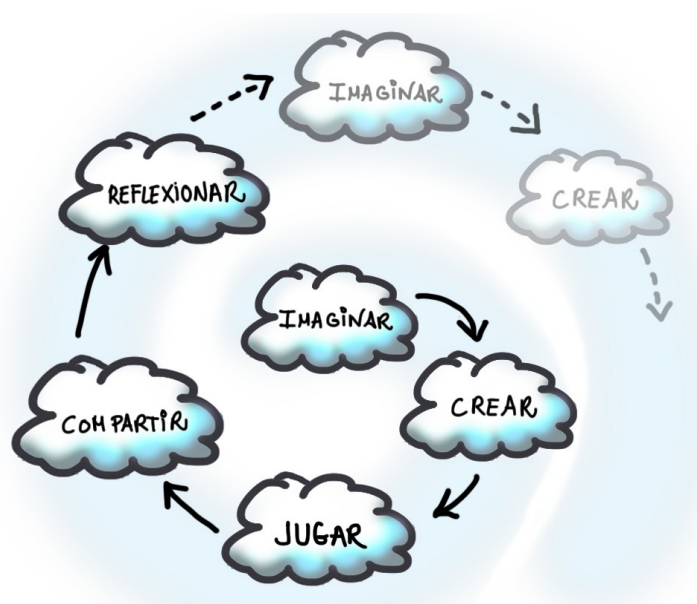


Figura 3.1. Modelo de aprendizaje según Resnick. Modificado de Resnick (2007).

Una de las primeras experiencias documentadas sobre este tema es la de Clements en 1986. Clements demuestra que alumnos instruidos en el uso del lenguaje Logo desarrollan una mayor capacidad de atención, autonomía y creatividad comparados con aquellos que no aprenden un lenguaje de programación. Pero además, el estudio de Clements afirma que el mero uso de las tecnologías de la información en el flujo de clase no representa una ventaja para la adquisición de mayor autonomía, creatividad o capacidad de atención. Es por tanto, el aprendizaje de un lenguaje de programación el desencadenante de esta mejora. Un aprendizaje que tiene lugar basándose en 25 sesiones de 45 minutos repartidas a razón de una sesión semanal a lo largo del curso lectivo (Clements, 1991). Por su parte, los estudios concretos que citan Scratch como medio para el desarrollo cognitivo han obtenido resultados similares a los observados en el aprendizaje de Logo. Se ha comprobado que el aprendizaje de Scratch desarrolla el pensamiento creativo, la habilidad para la resolución de problemas, así como una mejora en la capacidad de aprendizaje (Resnick, 2007). Sin embargo, son muchos los autores que destacan el rechazo existente al aprendizaje de la programación debido a la dificultad que supone el aprendizaje de la sintaxis de los lenguajes (Idlbi (2009), Bers (2010) y Smith (2010)). Idlbi asocia la falta de motivación por aprender a programar a lo poco llamativa de esta tarea si la comparamos con la construcción de robots. En su artículo Idlbi (2009) propone como solución a esta baja motivación, el uso de Scratch en niños de entre 7 y 15 años. De esta forma, propone el autor, es factible un acercamiento progresivo a lenguajes de programación tipo C++ sin que exista un rechazo frontal motivado por la dificultad que representa la sintaxis.

De forma paralela al desarrollo de Scratch, el grupo de investigación *TangibleK* de la Universidad de Tufts en Massachussets, lidera, junto al MIT, el desarrollo de un sistema de programación mixto gráfico y tangible denominado CHERP (*Creative Hybrid Environment for Robotic Programming*). Dicho proyecto hace uso de cubos de madera marcados con un código de barras que pueden ser unidos a modo de puzzle por los niños. El montaje resultante, ilustrado en la Figura 3.2, es fotografiado, después registrado por CHERP y finalmente convertido en un programa que es enviado a robots de tipo Lego WeDo. En definitiva, el sistema CHERP permite programar de forma manipulativa robots (Bers, 2010). De forma paralela, Smith (2010) presenta un proyecto que combina Scratch con Arduino para fomentar el desarrollo de la programación de forma tangible. Basándose en estas y otras experiencias, Dasgupta y Sayamindu (2014) conciben el futuro de Scratch asociado a «scripts» desarrollados por terceros que conecten Scratch con el mundo exterior mediante sensores y módulos *hardware* diseñados o derivados de Arduino, Raspberry Pi, Picoboard o Lego MindStorm entre otros. De esta forma, se lograría conectar el mundo físico con el mundo abstracto que representa la programación. Por otra parte, Dasgupta y Sayamindu plantean la necesidad de trabajar con gran cantidad de datos que deberán ser almacenados en variables y que permiten al alumno acercarse a la realidad de la programación, donde la recogida de datos y su posterior análisis es el verdadero foco de atención.



Figura 3.2 : Sistema CHERP de programación mediante bloques inteligentes.

Imagen obtenida de <http://ase.tufts.edu>

Haciendo hincapié sobre el número de sesiones necesarias para el desarrollo de las capacidades antes mencionadas, Pardamean y Bens (2011) demuestran que alumnos de quinto curso de EP, tras un breve curso de Logo de tan sólo 15 sesiones de 40 minutos, desarrollan de forma significativa

sus capacidades para la resolución de problemas, desarrollo de la imaginación y la originalidad. También se ha demostrado que proyectos de duración superior al semestre se asocian con resultados por encima de lo esperado. Esta mejora, se produce como consecuencia del uso del *software* fuera del aula (Chang, 2014). Por último, Kalelioglu y Gülbahar (2014) comprueban como en tan sólo diez sesiones no es posible desarrollar las capacidades antes citadas, aunque sí en la autoconfianza del alumnado acerca su capacidad en la resolución de problemas.

Antes de terminar esta sección, parece necesario citar una frase de Michel Resnick, máximo responsable del proyecto Scratch en el MIT: *“Mientras se aprende a programar, se aprenden muchas otras cosas. No se está aprendiendo únicamente a programar, se está programando para aprender”*. Esta aseveración, hace referencia a todo lo presentado en esta sección, en la que se deja claro que aprender a programar favorece el desarrollo de competencias necesarias durante toda la vida y que, a priori, poco tienen que ver con aprender a programar como, por ejemplo, la sociabilización y el trabajo en grupo (Dasgupta y Resnick, 2014) .

3.4. PROYECTOS GUBERNAMENTALES

A nivel internacional existen múltiples iniciativas que pretenden acercar la programación a las aulas. Entre ellas cabe destacar la de países como Israel³, Nueva Zelanda, Japón, Finlandia y Reino Unido⁴ (Díaz et al., 2014 y Artopoulos, 2015). Se trata de países que cuentan con proyectos curriculares donde se incorporan las TIC como materia a impartir pero que, además, incorporan la programación como parte de esta. Junto con de la incorporación de la programación al currículo oficial, existen otras propuestas que pretenden el impulso de la programación como, por ejemplo, el proyecto Norteamericano de capacitar a 10.000 profesores para que puedan enseñar

3 El sistema educativo israelí incorpora desde el curso 2010-2011 una formación en nuevas tecnologías en EP haciendo hincapié en el desarrollo de la capacidad para la resolución de problemas. Puede analizarse el currículo en:

http://cms.education.gov.il/EducationCMS/Units/MadaTech/englishsifria/AdaptingtheEducationSystemtothe21Century/mismach_htocnit_english.htm

4 El currículo inglés de EP plantea para el curso 2015-2016 una nueva materia denominada «*Computing*» que sustituye a la antigua «*ICT*» ampliando contenidos relativos a programación, haciendo mención expresa a la simulación de sistemas físicos y a la inclusión de diversas formas de entrada y salida de información. Puede analizarse el currículo en la siguiente dirección:

https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/239033/PRIMARY_national_curriculum_-_Computing.pdf

programación en las escuelas de Secundaria, el proyecto Program.AR en Argentina, las Olimpiadas Sirias en Informática o el proyecto *ProgeTiiger*⁵ del gobierno estonio (Díaz et al., 2014). Este último, caracterizado por promover el aprendizaje de programación mediante Scratch y la robótica mediante *Lego MindStorm*.

Tras este breve repaso a las iniciativas gubernamentales es necesaria una breve reflexión. Los videojuegos y, por ende, la programación, son uno de los sectores productivos con mayor auge en la actualidad, de hecho son los empleados novatos más demandados y mejor pagados (Ochsner, 2014). Sin embargo, resulta evidente que son pocos los países que han incluido la programación como un elemento más de currículo de EP. Quizá este hecho es uno de los motivos que ha llevado al sector privado a impulsar iniciativas que promuevan el aprendizaje de la programación en la escuela.

3.5. INICIATIVAS PRIVADAS PARA PROMOVER LA PROGRAMACIÓN EN LA EDUCACIÓN

Las iniciativas privadas son numerosas pero quizá puede considerarse a Marc Prensky como un pionero en la promoción de las TIC en la enseñanza. Así, en 2001 escribe “*La fluidez digital requiere también de la habilidad para diseñar, crear e inventar con los nuevos medios.*” Esta aseveración, recuperada de su artículo “*Digital natives, Digital immigrants*” deja ya asentada la necesidad de ir más allá en la enseñanza de las nuevas tecnologías. No basta enseñar a usar, es necesario enseñar a programar para dar el salto de consumidor a productor.

Independientemente del peso específico de Mark Prensky, hay que destacar el proyecto público sin ánimo de lucro conocido como CODE.org⁶. Esta organización busca llamar la atención de los gobiernos para que estos se hagan eco de la necesidad de incorporar la programación dentro de currículo oficial de EP y Educación Secundaria. Para promover esta incorporación, CODE.org desarrolla «*tutoriales*» que comparte en su web de forma totalmente gratuita y fomenta lo que han

5 El proyecto del gobierno estonio, aun siendo voluntario y enfocado principalmente a Educación Secundaria, es relevante por lo completo de su programa, el cual incluye des lenguajes de programación gráfico y programación de robots Lego Mindstorm, hasta competiciones de robótica y otros lenguajes de programación como Python. Puede consultarse el proyecto en: <http://www.progetiiger.ee/>

6 Se trata de un proyecto que recibe el apoyo de entre otros Bill Gates, Microsoft, Apple y Mark Zuckerberg.

dado en llamar «*Code Hour*», eventos de promoción que pueden ser desarrollados en cualquier punto del planeta y donde la programación es la protagonista.

Otro proyecto que merece mención por su implantación a nivel de EP es el desarrollado por Microsoft para plataformas Windows y Xbox. Se trata de una herramienta de programación de videojuegos denominado *Kodu Game Lab* (Ochsner y Anton, 2014). Esta herramienta destaca por presentar al estudiante un módulo de diseño de mundos tridimensionales donde pueden incorporarse personajes que deben ser programados para interactuar con el entorno creado. El modo de programación aún siendo diferente al de Scratch, presenta similitudes, ya que se trata de programación gráfica en la que se deben añadir diversos módulos (condiciones, variables, sensores, etcétera) para generar el código de cada uno de los personajes.

Una iniciativa española que debe ser citada en este trabajo es la que desarrollan los docentes españoles asociados a la plataforma Programamos.com, quienes de forma altruista promocionan la enseñanza de programación en nuestro país, ofreciendo asesoría, experiencia y actividades ya puestas en marcha en las aulas españolas desde EP hasta Secundaria y Ciclos Formativos.

Finalmente, antes de abordar el PE, es necesario recordar la iniciativa Arduino, una plataforma abierta que ha creado una placa «*microcontroladora*» multiplataforma, de precio reducido y fácil programación que permite la interacción entre *software* y el mundo físico mediante el uso de actuadores y sensores.

Así pues, una vez revisado el marco teórico, en el siguiente capítulo se presenta el PE, en el cual se desarrolla un programa de actividades formativas orientadas al desarrollo de competencias clave.

CAPÍTULO 4: PROYECTO EDUCATIVO

A lo largo de este capítulo se presenta un PE orientado al desarrollo de las competencias clave en EP. La primera Sección se dedica a presentar de forma somera los pormenores de este proyecto. A continuación, en la Sección 4.2, se describen los objetivos educativos que se persiguen alcanzar con este PE. Tras presentar los objetivos, en la Sección 4.3 se exponen las actividades a desarrollar. Posteriormente, se desarrolla la temporalización en la Sección 4.4. Finalmente, en la Sección 4.5 se comenta el mecanismo de evaluación que debe ser puesto en práctica durante el proyecto.

4.1. INTRODUCCIÓN

Como ya se ha adelantado en los capítulos anteriores, la idea que subyace en esta PE es el uso de la programación y la microelectrónica como motor para el desarrollo de las competencias clave en 5º y 6º de EP. No obstante, y debido a los límites en extensión fijados para este TFG, se ha optado por concretar las acciones en dos materias del currículo oficial, Ciencias Naturales y Ciencias Sociales. Sin embargo, esta concreción no debe entenderse como un escollo hacia su aplicación en otras áreas, ya que el PE desarrollado a continuación, es extensible a cualquier área o materia del currículo oficial. Así pues, el PE que se presenta en este capítulo se estructura en torno a los contenidos establecidos en la LOMCE para estas materias. A la par, se han incluido los contenidos referentes a programación, los cuales han sido secuenciados de forma que en cada actividad se sienten las bases de la siguiente, favoreciendo un aprendizaje progresivo y apoyado en los conocimientos previos persiguiendo, por tanto, un aprendizaje significativo.

4.2. OBJETIVOS EDUCATIVOS

Los objetivos de este PE se plantean basándose en un objetivo general que es guía y andamiaje sobre el que apoyan los objetivos específicos. Sin embargo, son los objetivos específicos los que permiten establecer metas discretas gracias a las cuales se puede alcanzar el objetivo general. Así pues, con la intención de precisar el sentido de este proyecto, se define el objetivo general de este PE como:

«Desarrollar las competencias clave mediante la integración de la programación y la microelectrónica en el flujo de trabajo del aula de quinto y sexto curso de Educación Primaria en Ciencias Naturales y Ciencias Sociales».

Definido el objetivo general, es necesario plantear unos objetivos específicos, los cuales se alcanzan de forma paulatina a lo largo esta programación de aula. Dichos objetivos se explicitan en cada una de las actividades programadas en la Sección 4.4. No obstante, además de los objetivos específicos plasmados en cada ficha, hay objetivos que son trabajados de forma transversal en todas ellas. Estos objetivos son los siguientes:

1. Desarrollar la competencia matemática.
2. Capacitar al alumnado en la resolución de problemas.
3. Comprender la sociabilización como una ventaja para el trabajo en equipo.
4. Comprender la importancia del cumplimiento de las normas de convivencia de cara al desarrollo de un trabajo en equipo fructífero.
5. Desarrollar la iniciativa personal, la curiosidad, el interés, la creatividad así como el espíritu emprendedor necesario que permita al alumnado crear aplicaciones que resuelvan cuestiones planteadas en el aula.
6. Crear diferentes representaciones y expresiones artísticas e iniciarse en la construcción de propuestas visuales y audiovisuales.

4.3. ACTIVIDADES PROPUESTAS

En esta sección se desarrollan las actividades planteadas para alcanzar el objetivo general explicitado en la anterior. Para favorecer su puesta en práctica, las actividades se desarrollan como fichas en las que se indica: materia y curso, temporalización, objetivos, contenidos, competencias, recursos materiales, agrupamiento, criterios de evaluación y una descripción paso a paso de la actividad. Además de los elementos mencionados, para la puesta en marcha de este PE, es necesario definir una metodología docente. Así pues, a continuación, y antes de presentar las actividades propuestas, en la Sección 4.3.1 se presenta la metodología docente que debe guiar la puesta en marcha de las actividades.

4.3.1 Metodología

Las actividades que se presentan a continuación, han sido diseñadas basándose en el modelo de aprendizaje cíclico propuesto por Resnick (2007) en el que se indica, tal y como se explica en la Sección 3.3 (Figura 3.1) la secuencia de acciones a seguir para que el aprendizaje sea constructivo

(imaginar, crear, jugar, compartir y reflexionar). En este proceso cíclico, el primer paso es plantear una idea primigenia que debe ser desarrollada por el alumnado mediante una actividad grupal de tormenta de ideas. De esta forma, son los alumnos, guiados y orientados por el docente, quienes imaginan una solución. Sin embargo, las soluciones planteadas por el alumnado pueden no ser conducentes a resolver el problema o simplemente no ser viables por motivos técnicos. Por este motivo, se hace necesario que el docente conduzca este proceso. Así, el objetivo del docente, será encauzar las ideas hacia una solución similar a la planificada en la actividad, aunque siempre manteniendo una visión abierta a nuevas opciones, permitiendo por tanto el pensamiento divergente.

Una vez desarrollada esta tormenta de ideas, y siempre durante la sesión inicial, comienza la explicación sobre como crear el trabajo. Tras realizar las explicaciones oportunas, comienza el trabajo, el cual se distribuye a lo largo de una, dos o tres sesiones en función de la complejidad de la actividad. Durante las sesiones de trabajo se debe permitir a cada grupo resolver la actividad de forma independiente, con posibles variaciones sobre la idea original, por ejemplo en el código, siempre y cuando conduzcan a una solución funcional. No obstante, para impedir situaciones de ansiedad o frustración, el docente debe estar atento y ayudar en las ocasiones que así lo requieran. No se trata, por tanto, de evaluar constantemente la capacidad resolutoria del alumnado sino de conseguir un aprendizaje mediado por el esfuerzo.

Una vez completado el trabajo, se da inicio a la tercera etapa, aquella que implica jugar, compartir y reflexionar, tareas que tienen lugar durante la última sesión de cada actividad. Así, al inicio de esta sesión, cada grupo juega con su propio trabajo. Después, los alumnos van cambiando de lugar de trabajo y acceden a los “juegos” realizados por sus compañeros. De esta forma, todos los alumnos comparten sus creaciones y juegan con ellas. Finalmente, se hace una puesta en común o reflexión general en la que se expone aquello que ha supuesto un hito en su proceso de aprendizaje.

4.3.2 Actividades formativas

Una vez analizada la metodología, y antes de presentar las actividades formativas, se procede a comentar cada una de ellas. En esta breve presentación se describe la actividad y se exponen los conceptos trabajados en torno a la programación.

- **Actividad formativa nº1 “Nutrición y reproducción”:** se trata de una actividad de iniciación a Scratch donde en 3 sesiones se crea una animación. Los aspectos relativos a programación que se trabajan son: ejes de coordenadas cartesianas, movimiento de objetos así como mostrar y ocultar objetos.
- **Actividad formativa nº2 “Energías renovables, consumo y desarrollo sostenible”:** en esta actividad, realizada con Scratch a lo largo de 4 sesiones, se desarrolla un juego interactivo en el que se crean botones no visibles o áreas activas que reaccionan cuando son pulsados por el ratón. Los aspectos relativos a programación que se trabajan son: cambios de disfraz, eventos «*On clic*» así como grabación y reproducción de audio en Scratch.
- **Actividad formativa nº3 “El Sistema Solar”:** esta actividad ha sido diseñada para descubrir a lo largo de 4 sesiones, los diferentes mecanismos existentes en programación para crear una misma función. Por ese motivo se genera un juego similar en funcionamiento a la actividad formativa 2, pero en el que se trabajan aspectos relativos a programación diferentes: bucles y condiciones.
- **Actividad formativa nº4 “Relieve en la península Ibérica”:** en esta actividad, en la que se trabaja durante 5 sesiones con Scratch, se crea un juego de preguntas y respuestas con marcador de aciertos. Los aspectos nuevos relativos a programación que se trabajan son: variables.
- **Actividad formativa nº5 “Las huellas del tiempo”:** esta actividad permite al alumno aplicar los conocimientos adquiridos sobre programación para generar, durante un proyecto de 5 sesiones, una presentación interactiva en la que se realiza una línea de tiempo. En esta línea de tiempo, quedan representados y explicados elementos definitorios de la Edad Moderna en la península Ibérica.
- **Actividad formativa nº6 “Fisiología y anatomía del cuerpo humano”:** esta actividad, de una duración de 5 sesiones, persigue que los alumnos creen un juego de preguntas y respuestas en Scratch. A diferencia del caso anterior, en este caso, la actividad persigue profundizar en el uso de contenedores o listas de datos que almacenan las respuestas correctas.
- **Actividad formativa nº7 “La célula”:** se trata de una actividad en la que el alumnado, durante 4 sesiones, descubre el funcionamiento de los circuitos eléctricos mediante el trabajo con placas para crear prototipos («*protoboards*»).

- **Actividad formativa nº8 “Tecnología, objetos y máquinas”**: en esta actividad, de duración 6 sesiones, el alumnado descubre la conexión entre la programación y el control de elementos tangibles mediante Scratch for Arduino (S4A) y Arduino.
- **Actividad formativa nº9 “Características demográficas”**: esta actividad, con duración de tan sólo 3 sesiones, aprovecha la capacidad de Scratch para almacenar datos en variables y su posterior tratamiento matemático. Así, los alumnos deben desarrollar un programa que almacene datos y después calcule un valor relativo a demografía.
- **Actividad formativa nº10 “De la revolución liberal a la democracia”**: actividad de 6 sesiones de duración que integra S4A con botones y LED para generar un juego de preguntas y respuestas sobre la historia contemporánea española. El juego, el cual se inicia tras pulsar un botón, presenta un indicador LED que emite un destello cada segundo determinando así el tiempo restante.

Tras describir de forma escueta las actividades formativas que componen este PE, a continuación, se presentan organizados en tablas. Las primeras actividades, pertenecientes a 5º curso de EP aparecen identificados por el color azul del encabezado, mientras que las restantes, correspondientes a 6º curso de EP, se identifican por el color gris oscuro. En cada tabla se ha incluido el título, el curso y la materia en la que se imparte, la duración de la actividad formativa expresada en número de sesiones, los objetivos y los contenidos didácticos, los recursos materiales necesarios para la puesta en marcha, las competencias clave que se pretenden desarrollar, el agrupamiento del alumnado durante la actividad formativa, la descripción sesión a sesión de la actividad y por último, los criterios de evaluación.

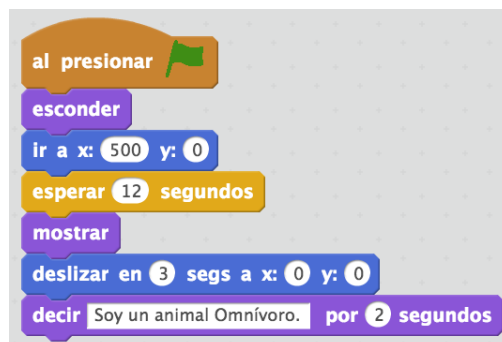
Actividad formativa nº1: Nutrición y reproducción en el Reino Animal y Vegetal	
Materia: Ciencias Naturales (5º EP).	Temporalización: 3 sesiones.
Objetivos	• Describir los sistemas de reproducción y nutrición más relevantes del Reino Animal y Reino Vegetal. • Crear presentaciones y animaciones con Scratch.
Contenidos	• Nutrición y reproducción en el Reino Animal. • Nutrición y reproducción en el Reino Vegetal. • Mostrar y esconder objetos en Scratch. • Ejes de coordenadas, movimiento y cambios de apariencia en Scratch.
Recursos materiales	• Aula de informática con acceso a Internet. • Scratch. • Esquemas de nutrición y reproducción.
Competencias	CMCT, CSC, CD y CAA.
Agrupamiento	Actividad desarrollada individualmente.
Actividad	<i>Sesión 1: Imaginar</i> 1) Idea primigenia: presentar los principales mecanismos de nutrición y reproducción existentes en animales y vegetales mediante una presentación con movimiento. 2) Tormenta de ideas: se persigue que los alumnos lleguen a plantear el desarrollo de una animación en la que se presentan los diversos mecanismos de nutrición y reproducción estudiados. 3) Programar todos los elementos para que vayan apareciendo uno tras otro al tiempo que informan de su nombre y características.  <i>Sesión 2: Crear</i> 1) Buscar imágenes en Google. 2) Importar las imágenes en Scratch o crear los elementos necesarios con la herramienta de dibujo de Scratch. <i>Sesión 3: Jugar, Compartir y Reflexionar</i>
Criterios de evaluación	• Identifica los diferentes las etapas y diferentes mecanismos de nutrición y reproducción del Reino Animal y Vegetal. • Reconoce los ejes de coordenadas cartesianas en Scratch. • Esconde, muestra y mueve objetos en Scratch.

Tabla 4.1. Ejes de coordenadas cartesianas y desplazamiento en Scratch.

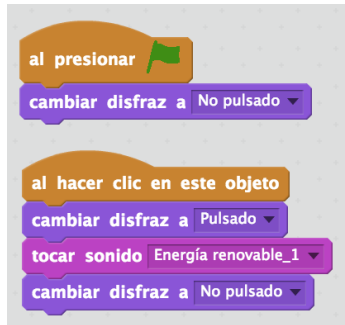
Actividad formativa nº2: Energías renovables, consumo y desarrollo sostenible		
Materia: Ciencias Naturales (5º EP).		Temporalización: 4 sesiones.
Objetivos	- Diferenciar y valorar las fuentes de energía renovables y no renovables. - Desarrollar hábitos de consumo responsable y de desarrollo sostenible. - Crear presentaciones interactivas con audio en Scratch.	
Contenidos	- Fuentes de energía renovables y no renovables. - Desarrollo sostenible y consumo energético responsable. - Cambios de disfraz y eventos «On clic» en Scratch. - Grabación y reproducción de audio en Scratch.	
Recursos materiales	- Aula de informática con acceso a Internet. - Imágenes que muestren actitudes y usos de la energía responsables y no responsables.	- Scratch. - Gimp⁷. - Micrófonos.
Competencias	CMCT, CSC, CD y CAA.	
Agrupamiento	Actividad desarrollada en pareja.	
Actividad	<i>Sesión 1: Imaginar</i> 1) Idea primigenia: generar una aplicación que explique las diferencias entre energías renovables y no renovables así como conductas y usos responsables de la energía. 2) Tormenta de ideas: se persigue que los alumnos lleguen a plantear una actividad interactiva en la que tras pulsar sobre cada elemento representativo de la imagen se escuche una explicación sobre la conducta o elemento seleccionado. <i>Sesiones 2 y 3: Crear</i> 1) Buscar imágenes en Google. 2) Realizar las grabaciones explicativas. 3) Utilizar Gimp para recortar los elementos del ecosistema y exportarlos en formato PNG. Importarlos en Scratch. 4) Programar los elementos (objetos) para que al ser pulsados cambien de disfraz y ejecuten la grabación.  <i>Sesión 4: Jugar, Compartir y Reflexionar</i>	
Criterios de evaluación	- Identifica los diferentes tipos, usos y actitudes respecto de las energías. - Utiliza eventos «On clic» y los asocia a grabaciones realizadas en Scratch.	

Tabla 4.2. Eventos «On clic» en Scratch.

7 En este «videotutorial» se explica como recortar imágenes en Gimp: <https://youtu.be/8CIkeyR3VYs>

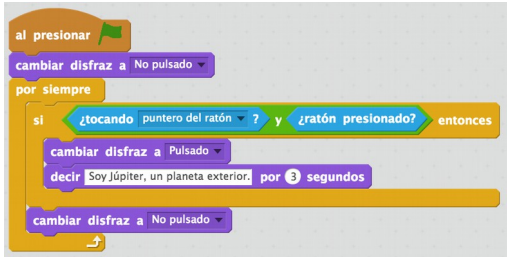
Actividad formativa nº3: El Sistema Solar	
Materia: Ciencias Sociales (5º EP).	Temporalización: 4 sesiones.
Objetivos	• Describir el Sistema Solar y sus componentes. • Descubrir las posibilidades de bucles, condiciones y sensores en Scratch.
Contenidos	• El Universo y el Sistema Solar. Cuerpos celestes. • Bucles, condiciones y sensores en programación.
Recursos materiales	• Aula de informática con acceso a Internet. • Scratch. • Imágenes del Sistema Solar y la Vía Láctea. • Gimp.
Competencias	CMCT, CSC, CD y CAA.
Agrupamiento	Actividad desarrollada en pareja.
Actividad	<i>Sesión 1: Imaginar</i> 1) Idea primigenia: presentar la Vía Láctea, el Sistema Solar y los cuerpos celestes existentes en él. 2) Tormenta de ideas: se persigue que los alumnos planteen el desarrollo de un programa interactivo similar al desarrollado en la Ficha 2 (Tabla 4.2). 4) Programar cada elemento para que reaccione cuando sea pulsado con el ratón. Así, cuando el elemento sea pulsado, se debe informar al jugador sobre su naturaleza.  <i>Sesiones 2 y 3: Crear</i> 1) Buscar imágenes en Google. 2) Utilizar Gimp para recortar los elementos del Sistema Solar y exportarlos en formato PNG. 3) Importar los elementos a Scratch. <i>Sesión 4: Jugar, Compartir y Reflexionar</i>
Criterios de evaluación	• Identifica los planetas y los cuerpos celestes del Sistema Solar. • Modifica el escenario en Scratch e incluye elementos nuevos. • Utiliza bucles y condiciones simples en Scratch.

Tabla 4.3. Introducción a las condiciones y bucles en Scratch.

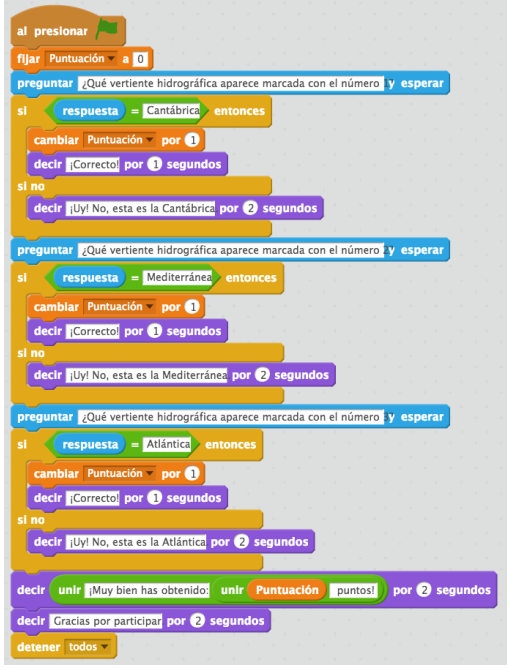
Actividad formativa nº 4: Relieve en la Península Ibérica	
Materia: Ciencias Sociales (5º EP).	Temporalización: 5 sesiones.
Objetivos	- Nombrar y situar en un mapa las principales unidades de relieve de la Península Ibérica y relacionarlo con los factores climáticos. - Descubrir y utilizar variables en Scratch.
Contenidos	- Principales paisajes: mediterráneo, oceánico, de montaña y subtropical. - Vertientes hidrográficas, relieve y factores climáticos en la península Ibérica. - Variables como almacenes de información en Scratch.
Recursos materiales	- Aula de informática con acceso a Internet. - Gimp. - Mapas geográficos de la Península ibérica. - Scratch.
Competencias	CMCT, CSC, CD y CAA.
Agrupamiento	Actividad desarrollada en pareja. Cada pareja diseña y crea un juego con contenidos diferentes (relieve, ríos, factores climáticos, paisajes,...).
Actividad	<i>Sesión 1: Imaginar</i> 1) Explicar el problema: reconocer el relieve de la Península Ibérica, paisajes, vertientes hidrográficas y factores climáticos. 2) Tormenta de ideas: se persigue que los alumnos inventen juegos interactivos con preguntas, respuestas y un marcador con la puntuación obtenida. <i>Sesiones 2 y 3: Crear</i> 1) Importar los mapas geográficos como fondos en Scratch. 2) Programar las preguntas y la variable “Puntuación” para que esta aumente en una unidad con cada respuesta correcta. <i>Sesiones 4 y 5: Jugar, Compartir y Reflexionar</i> 
Criterios de evaluación	- Relaciona factores climáticos con los principales paisajes de la Península. - Utiliza variables en Scratch.

Tabla 4.4. Introducción a las variables en Scratch (Variables I).

Actividad formativa nº5: Las huellas del tiempo	
Materia: Ciencias Sociales (5º EP).	Temporalización: 5 sesiones.
Objetivos	- Representar mediante líneas de tiempo hechos históricos relativos a la Edad Moderna en la Península Ibérica. - Utilizar la programación como medio para crear una línea de tiempo.
Contenidos	- La creación de Al Ándalus. - La Monarquía Hispánica y su expansión en América y Europa. - Ejes de coordenadas, grabaciones de audio, eventos “On clic” y cambio de disfraz.
Recursos materiales	- Aula de informática con acceso a Internet. - Scratch. - Micrófono.
Competencias	CMCT, CSC, CD y CAA.
Agrupamiento	Actividad desarrollada en pareja. Cada pareja crea una línea de tiempo explicando los hechos más relevantes de la Edad Moderna en la Península Ibérica.
Actividad	<i>Sesión 1: Imaginar</i> 1) Explicar el problema: cómo memorizar eventos en orden cronológico. 2) Tormenta de ideas: se persigue que los alumnos desarrollen una línea de tiempo interactiva en la que expliquen con sus palabras los hechos históricos. <i>Sesiones 2, 3 y 4: Crear</i> 1) Buscar información e imágenes en Internet. 2) Realizar las grabaciones, explicando los hechos históricos. 3) Programar la interactividad.  <i>Sesión 5: Jugar, Compartir y Reflexionar</i>
Criterios de evaluación	- Relaciona factores climáticos con los principales paisajes de la Península. - Utiliza variables en Scratch.

Tabla 4.5. Proyecto de aplicación: Líneas de tiempo.

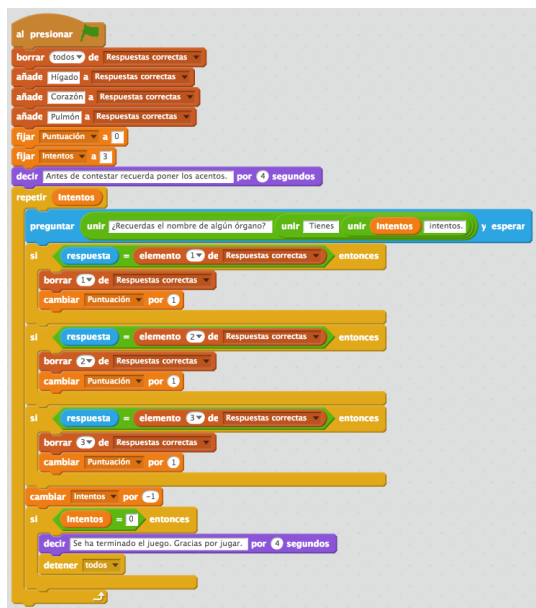
Actividad formativa nº6: Fisiología y anatomía del cuerpo humano	
Materia: Ciencias Naturales (6º EP).	Temporalización: 5 sesiones.
Objetivos	• Describir los principales sistemas y aparatos del cuerpo humano. • Función de nutrición y reproducción en el ser humano. • Utilizar contenedores como almacén de datos en Scratch.
Contenidos	• Niveles de organización en los seres vivos. • Listas o Contenedores en Scratch.
Recursos materiales	• Aula de informática con acceso a Internet. • Scratch. • Imágenes de tejidos, órganos, sistemas y aparatos.
Competencias	CMCT, CSC, CD y CAA.
Agrupamiento	Actividad desarrollada en pareja. Cada pareja diseña y crea un juego con contenidos diferentes (tejidos, órganos, sistemas, aparatos, reproducción,...).
Actividad	<i>Sesión 1: Imaginar</i> 1) Explicar el problema: identificar y recordar los niveles de organización en los seres vivos (desde tejidos hasta aparatos). 2) Tormenta de ideas: se persigue que los alumnos se planteen crear un juego dónde un personaje pregunte sobre los diferentes niveles de organización y si la respuesta es correcta aumente la puntuación. <i>Sesiones 2, 3 y 4: Crear</i> 1) Insertar imágenes de fondo que sirvan para ilustrar las preguntas. 2) Incorporar un personaje sobre el que recaerá la programación. 3) Crear un contenedor o lista con las respuestas correctas. 4) Programar las preguntas. 5) Incorporar dos variables (contador de aciertos y contador de intentos).  <i>Sesión 5: Jugar, Compartir y Reflexionar</i>
Criterios de evaluación	• Recuerda y cataloga nombres de tejidos, órganos, sistemas y aparatos. • Utiliza bucles y condiciones en Scratch. • Comprende el uso de variables y contenedores.

Tabla 4.6. Contenedores y variables (II) en Scratch.

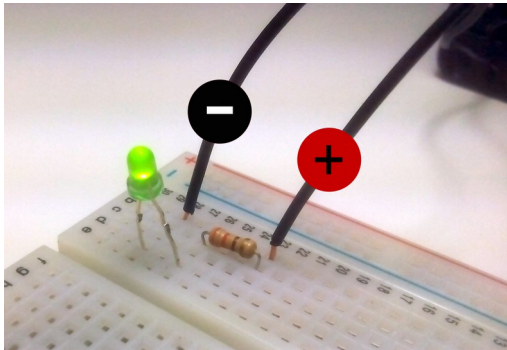
Actividad formativa nº7: La célula	
Materia: Ciencias Naturales (6º EP).	Temporalización: 4 sesiones.
Objetivos	• Describir los diferentes tipos de células y sus orgánulos más importantes. • Crear un circuito eléctrico.
Contenidos	• Células Eucariotas animales y vegetales. Células Procariotas. • La electricidad. El circuito eléctrico. • Conductores y aislantes.
Recursos materiales	• Aula laboratorio. • Kit protoboard (LED, cables, placa de conexión, resistencias y pilas). • Tablero de madera, broca manual y taladro para madera.
Competencias	CMCT, CSC, CD y CAA.
Agrupamiento	Actividad desarrollada en grupos de tres. Cada grupo diseña y crea un juego con contenidos diferentes (tipos celulares, la célula animal, orgánulos,...).
Actividad	<i>Sesión 1: Imaginar</i> 1) Idea primigenia: diseñar un juego de mesa con el que se pueda aprender los tipos celulares y sus orgánulos. 2) Tormenta de ideas: se persigue que los alumnos lleguen a plantear el desarrollo de un juego de mesa de tipo <i>Lectron</i>⁸. <i>Sesión 2: Crear</i> 1) Buscar imágenes en Google. 2) Dibujar en tamaño A4 las células elegidas. 3) Pegar el dibujo al tablero de madera y realizar las perforaciones en los lugares adecuados. 4) Realizar el montaje eléctrico por la parte trasera, dejando libres los extremos que permitan cerrar los circuitos.  <i>Figura 4.7. Esquema de conexión básico en una placa protoboard.</i> <i>Sesión 4: Jugar, Compartir y Reflexionar</i>
Criterios de evaluación	• Reconoce las diferencias esenciales entre los diferentes tipos de células. • Identifica los elementos básicos de un circuitos eléctrico.

Tabla 4.7. Circuitos eléctricos con protoboards.

8 Juegos de mesa de la marca Diset que permiten jugar a descubrir la respuesta correcta mediante un circuito que al cerrarse enciende una bombilla.

Actividad formativa nº8: Tecnología, objetos y máquinas		
Materia: Ciencias Naturales (6º EP).		Temporalización: 6 sesiones.
Objetivos	• Montar juguetes electrónicos que contengan poleas. • Utilizar bucles, condiciones y eventos en Scratch for Arduino (S4A). • Descubrir la programación como elemento de control de actuadores y sensores en el mundo físico.	
Contenidos	• Poleas y máquinas con operadores combinados. • Condiciones, bucles y eventos en S4A.	
Recursos materiales	• Aula de informática con acceso a Internet. • 9 Kits Arduino Mega con «shield» de conexión⁹ Grove. • Madera de marquetería y herramientas para madera.	
Competencias	CMCT, CSC, CD y CAA.	
Agrupamiento	Actividad desarrollada en grupos de 4 alumnos.	
Actividad	<i>Sesión 1: Imaginar</i> 1) Idea primigenia: ¿Cómo funciona un ascensor?. 2) Tormenta de ideas: se persigue que el alumnado diseñe un ascensor controlado desde el ordenador. <i>Sesión 2, 3, 4 y 5: Crear</i> Figura 4.8. Placa Arduino Mega conectada a motor y LED. 1) Conectar LED y motor en la placa Arduino Mega. 2) Programar funcionamiento de los botones, LED y motor. 3) Diseño y construcción del soporte y del ascensor. 4) Construcción y montaje del sistema. Figura 4.9. Programa que permite controlar el ascensor con el teclado. <i>Sesión 6: Jugar, Compartir y Reflexionar</i>	
Criterios de evaluación	• Diseña, programa y monta un ascensor. • Reconoce el funcionamiento de los mecanismos con poleas.	

Tabla 4.8. Microelectrónica con Bitbloq.

⁹ Entre el material que contiene el kit, encontramos placas Arduino Mega con «shield» de conexión rápida Grove, motores de movimiento continuo y LED montados en bloques de conexión rápida.

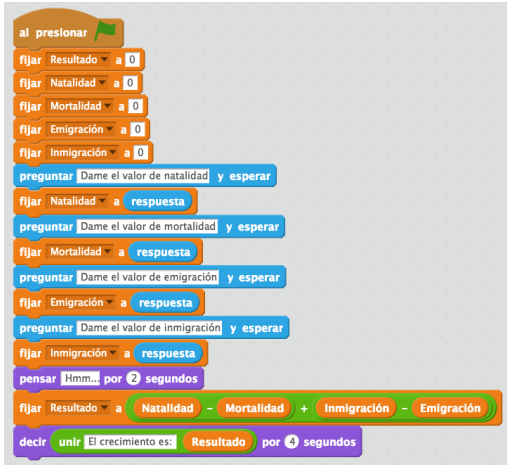
Actividad formativa nº9: Características demográficas	
Materia: Ciencias Sociales (6º EP).	Temporalización: 3 sesiones.
Objetivos	• Comparar características demográficas de diferentes países europeos. • Usar variables para almacenar datos y operar matemáticamente con ellos.
Contenidos	• Natalidad, mortalidad, inmigración y emigración. • Variables y operaciones aritméticas.
Recursos materiales	• Aula de informática con acceso a Internet. • Valores de natalidad, mortalidad, inmigración y emigración de países de la Unión Europea. • Scratch.
Competencias	CMCT, CSC, CD y CAA.
Agrupamiento	Actividad desarrollada en parejas.
Actividad	<i>Sesión 1: Imaginar</i> 1) Idea primigenia: calcular de forma rápida el crecimiento de la población. 2) Tormenta de ideas: se persigue que el alumnado diseñe un programa que almacene datos en variables y después opere con ellos para dar un valor numérico.  <i>Sesión 2: Crear</i> 1) Programar preguntas y variables que se igualan a la respuesta para después poder crear una variable resultado que se imprime en pantalla. <i>Sesión 3: Jugar, Compartir y Reflexionar</i>
Criterios de evaluación	• Reconoce la importancia de las características demográficas para explicar la evolución de los países. • Valora la potencia de las variables. • Identifica el momento en el que almacenar las respuestas en variables.

Tabla 4.9. Operaciones aritméticas con Scratch.

Actividad formativa nº10: De la revolución liberal a la democracia	
Materia: Ciencias Sociales (6º EP).	Temporalización: 6 sesiones.
Objetivos	- Reconocer los cambios más importantes en la España contemporánea. - Reconocer y utilizar sensores y actuadores desde S4A.
Contenidos	- La España contemporánea. - Botones y LED.
Recursos materiales	- Aula de informática con acceso a Internet y S4A instalado. 9 Kits Arduino Mega con «shield» de conexión Grove.
Competencias	CMCT, CSC, CD y CAA.
Agrupamiento	Actividad desarrollada en parejas.
Actividad	<i>Sesión 1: Imaginar</i> 1) Idea primigenia: ¿Cómo aprender los eventos más importantes de la España contemporánea? 2) Tormenta de ideas: se persigue que el alumnado diseñe un juego de preguntas y respuestas. Este juego empieza cuando el jugador pulsa un botón y tiene 10 segundos para contestar cada pregunta. El tiempo lo marca un LED parpadeando. <i>Sesión 2, 3 y 4: Crear</i> 1) Conectar el botón y el LED a la placa Arduino Mega y comprobar su funcionamiento. 2) Realizar la programación del juego, incluyendo una variable como marcador y otra como cronómetro. <i>Sesión 5 y 6: Jugar, Compartir y Reflexionar</i>
Criterios de evaluación	- Sitúa en una línea de tiempo los eventos más importantes. - Organiza correctamente las órdenes necesarias que funcionen los contadores.

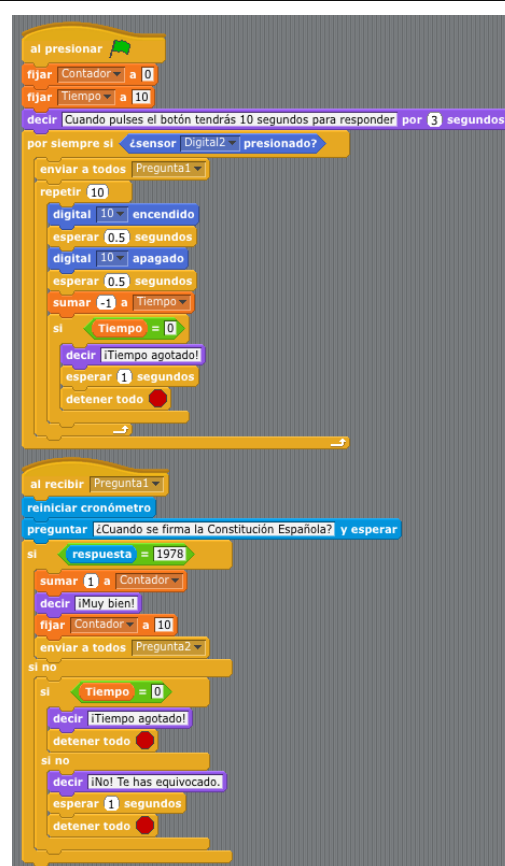


Figura 4.11. Programa necesario para juego de preguntas y respuestas con temporizador.

Tabla 4.10. Preguntas, respuestas, contadores, LED y botón.

4.4. TEMPORALIZACIÓN

Teniendo en cuenta los resultados presentados en la Sección 3.3 de este trabajo, queda clara la necesidad de establecer una duración del proyecto superior o igual a quince sesiones (Clements, 1991 y Pardamean y Bens, 2011). Además, se ha de tener en cuenta a Chang (2014) quien demuestra que un proyecto de duración superior al semestre favorece el aprendizaje al conseguir que el alumnado practique desde casa. Así pues, teniendo en cuenta el número de sesiones disponibles en Ciencias Naturales y Ciencias Sociales en 5º y 6º de EP y los aspectos antes mencionados, se ha diseñado una temporalización que permite el aprendizaje progresivo de programación a lo largo de dos cursos lectivos a razón de unas 20 sesiones por curso. A continuación, en las Tablas 4.11 y 4.12 se presenta el cronograma para la puesta en marcha del PE expuesto este capítulo. En estas tablas, la duración de las actividades formativas ha sido indicado con cuadros de color.

	5º Curso de EP									
Mes	Sep.	Oct.	Nov.	Dic.	Ene.	Feb.	Mar.	Abr.	May.	Jun.
Actividad formativa 1										
Actividad formativa 2										
Actividad formativa 3										
Actividad formativa 4										
Actividad formativa 5										

Tabla 4.11. Cronograma para las actividades formativas programadas en 5º curso de EP.

	6º Curso de EP									
Mes	Sep.	Oct.	Nov.	Dic.	Ene.	Feb.	Mar.	Abr.	May.	Jun.
Actividad formativa 6										
Actividad formativa 7										
Actividad formativa 8										
Actividad formativa 9										
Actividad formativa 10										

Tabla 4.12. Cronograma para las actividades formativas programadas en 6º curso de EP.

Una vez presentada la temporalización del PE, queda por presentar uno de los aspectos más importantes de todo proceso formativo, la evaluación. A continuación, en las Secciones 4.5 y 4.6 se presentan respectivamente la evaluación del alumnado y del PE.

4.5. EVALUACIÓN DEL ALUMNADO

Todo proceso de enseñanza y aprendizaje requiere de un proceso de reflexión que determine el nivel de consecución de los objetivos fijados de forma previa a su puesta en marcha, a este proceso se le denomina evaluación. La evaluación, no puede ser entendida como una acción aislada que se desarrolla al finalizar el proceso formativo, sino como un proceso que debe comenzar incluso antes que la propia formación. Así, se establecen los siguientes mecanismos de evaluación para este PE:

- **Evaluación diagnóstica:** se debe realizar en 4º de EP, durante las últimas semanas del curso y siempre una vez terminados los controles de evaluación final. En este periodo, se realiza una actividad sencilla con Scratch, permitiendo al alumnado explorar las posibilidades de la herramienta de dibujo. De esta forma, se puede obtener información sobre el alumnado y su destreza en el manejo del ordenador, pudiendo definir con mayor precisión el ritmo de las actividades a realizar durante el siguiente curso.
- **Evaluación formativa:** se lleva a cabo a lo largo de las sesiones programadas y se compone de:
 - Heteroevaluación: proceso de observación cotidiano y sistemático que quede reflejado en un diario de incidencias.
 - Coevaluación: periodo de reflexión realizado al finalizar cada actividad en el que, es el alumnado el que hace una puesta en común dónde se refleja el aprendizaje y los obstáculos que han ido saliendo durante las sesiones.
 - Autoevaluación: al finalizar cada trimestre, es muy importante dedicar unos minutos para que cada alumno, de forma individual reflexione sobre su aprendizaje.
- **Evaluación sumativa:** desarrollada al finalizar cada uno de los bloques de contenidos definidos de las materias Ciencias Naturales y Ciencias Sociales. No obstante, de cara a evaluar el aprendizaje respecto a programación, es necesario plantear dos actividades que se llevarán a término de forma individual y durante el último mes del curso.
 - Cuestionario que incluye capturas de pantalla de programas generados en Scratch por el propio alumnado. Para cada captura de pantalla, se requiere al alumno explicar que ocurrirá paso a paso una vez se ejecute el programa.
 - Plantear una situación problema y solicitar que el alumno diseñe, conceptualmente, un programa que resuelva dicha situación.

4.6. EVALUACIÓN DEL PROYECTO EDUCATIVO

Respecto a la evaluación del PE, se hace necesario establecer un sistema de control que favorezca la mejora del proyecto tras su puesta en práctica. Para ello se diseña un sistema que determine las debilidades y amenazas que puede sufrir el PE y por otro lado las fortalezas y oportunidades que brinda el PE. Con este propósito se desarrolla una rúbrica de evaluación (Tabla 4.9) que permita al docente determinar el nivel de consecución de los objetivos.

Materia:		Curso:		
Actividad evaluada:		Trimestre:		
	Nivel de consecución			
	Bajo (0)	Medio (1)	Alto (2)	Nota
Temporalización	Insuficiente.	Aceptable.	Adecuada.	
Objetivos	No se alcanzan	Se alcanzan algunos.	Se alcanzan todos.	
Contenidos	No se desarrollan en su totalidad.	Se desarrollan en su totalidad.	Se amplían contenidos.	
Recursos materiales	Inadecuados.	Insuficientes aunque adecuados.	Suficientes y adecuados.	
Agrupamientos	Derivan en situaciones antipedagógicas.	Adecuados.	El trabajo se realiza con total cercanía y espontaneidad.	
Desarrollo de la actividad	Inadecuado. No se desarrollan las 5 etapas de Resnick.	Aceptable. Se desarrollan la mayor parte de las etapas.	Adecuado. Se desarrollan todas las etapas planificadas.	
Evaluación	No se alcanzan los criterios de evaluación.	Se alcanzan gran parte de los criterios de evaluación.	Se cumplen todos los criterios de evaluación.	

Tabla 4.13. Rúbrica de evaluación del Proyecto Educativo.

Además de poner en práctica esta rúbrica de evaluación, es necesario llevar a cabo una recogida de información diaria, mediante la observación minuciosa y rigurosa y la posterior anotación de aquellos eventos que puedan suponer un cambio sobre la rutina de trabajo diaria. Durante esta recogida de información, realizada en un diario de incidencias, no hay que olvidar prestar atención a los periodos de reflexión del alumnado al terminar cada actividad, aprovechando así la puesta en común para recabar información sobre los problemas, incidencias e inquietudes del alumnado en referencia a la estructura de la clase, la metodología y otros aspectos que se consideren

importantes. En cuanto a las debilidades, amenazas, fortalezas y oportunidades detectadas, a priori, hay que destacar las siguientes:

- **Debilidades:** la falta de experiencia previa en el centro.
- **Amenazas:** la falta de formación del profesorado.
- **Fortalezas:** el hecho de que sea un programa de complejidad creciente que se desarrolla con el tiempo necesario para evitar su fracaso.
- **Oportunidades:** el hecho de comenzar la formación en EP permitirá el desarrollo de proyectos más complejos en Educación Secundaria.

CAPÍTULO 5: LIMITACIONES, CONCLUSIONES Y PROSPECTIVAS

Para finalizar este TFG es necesario acometer un proceso reflexivo encaminado a la mejora del proyecto. Con este objetivo se desarrolla este capítulo, en el que se presentan, por este orden, las limitaciones del proyecto, las conclusiones referentes al TFG y al PE y aquellos aspectos que podrían ser implementados en un futuro.

5.1 LIMITACIONES

Para el desarrollo de este TFG ha habido que superar algunos problemas que han dificultado su realización. A continuación, se exponen las limitaciones más patentes que han determinado, en alguna medida, la realización de este TFG.

- Desconocimiento de aplicaciones diferentes a Scratch utilizadas en la enseñanza de programación. Este hecho, fue resuelto dedicando unos días de trabajo a la evaluación de cada una de las herramientas descubiertas durante la realización del marco teórico (Alice, Kodu, S4A, entre otras) concluyendo finalmente que Scratch y Bitbloq son las herramientas con más ventajas de cara al aprendizaje del alumnado.
- Dificultad para desarrollar una metodología práctica que favorezca el aprendizaje no solo de procedimientos sino también de contenidos y actitudes. Este punto se ha resuelto gracias al estudio del marco teórico. Durante su estudio, se localiza el método de trabajo cíclico descrito por Resnick (2007) en el que se aúna el aprendizaje de estos tres aspectos, dotando de sentido y lógica al modelo de enseñanza presentado en este PE.
- Falta de tiempo para desarrollar un proyecto de mayor envergadura en el que no solo se plasme un proyecto para la inclusión de la programación en Ciencias Sociales y Ciencias Naturales sino también en otras materias del currículo oficial. Este problema no ha podido ser superado debido a la limitación temporal que impone el curso lectivo.

Tras analizar las limitaciones encontradas en este TFG, en la siguiente sección se especifican las conclusiones obtenidas en este trabajo.

5.2. CONCLUSIONES

En la presente sección se presentan las conclusiones de este trabajo, conclusiones que toman como referente los objetivos planteados en el TFG. En primer lugar, en la Sección 5.2.1 se presentan las conclusiones relativas a los objetivos establecidos en el Capítulo 2 y, a continuación, en la Sección 5.2.2 las conclusiones derivadas del PE.

5.2.1. Conclusiones del Trabajo Fin de Grado

Para demostrar la consecución de los objetivos, en esta sección se establece una correlación entre los objetivos establecidos en el Capítulo 2 y las acciones puestas en práctica para su consecución. Así, el objetivo general de este TFG ha sido definido como *«Crear un proyecto educativo basado en la enseñanza de la programación que permita el desarrollo de las competencias clave establecidas en el currículo oficial del Estado Español»*. Este objetivo ha sido cumplido mediante el diseño de un PE en el que se han establecido unas acciones formativas. Estas acciones formativas, con una duración de entre 3 y 6 sesiones, han sido diseñadas para favorecer la enseñanza de la programación. Pero, además, el objetivo de las acciones formativas ha perseguido trabajar los contenidos establecidos en el currículo oficial, desarrollando actitudes y procedimientos que favorecen la adquisición de las competencias clave.

Por su parte, los objetivos específicos se han ido cumpliendo de forma gradual mediante el desarrollo de un marco teórico en el Capítulo 3. Capítulo que, a su vez, se articula en diversas secciones: en la Sección 3.1 se han cumplido los objetivos referentes al estudio de la historia de los lenguajes de programación y, en la Sección 3.2 el estudio de los lenguajes de programación orientados a la educación. Más adelante, en el Capítulo 4, se ha expuesto el PE, donde inicialmente se presentan los objetivos de enseñanza que guían el desarrollo del PE. Una vez, explicitados los objetivos, a lo largo de la Sección 4.3 han sido planteados los contenidos referentes al currículo de Ciencias Naturales y Ciencias Sociales así como las acciones educativas conducentes a la consecución de los objetivos didácticos. Tras los contenidos, en la Sección 4.4 se ha recogido la temporalización, en la que se esquematiza el proceso de enseñanza a lo largo de los cursos implicados en este PE. Posteriormente, en la Sección 4.5 se han expuesto los criterios de evaluación así como los mecanismos de evaluación necesarios para constatar hecho educativo. Finalmente, en este capítulo se han presentado las conclusiones y se han planteado las posibles implementaciones a poner en práctica en un futuro próximo. En resumen, se podría decir que se han cumplido todos los objetivos marcados en este TFG en el Capítulo 2.

5.2.2. Conclusiones del Proyecto Educativo

Los objetivos establecidos para el PE (Sección 4.1) hablan sobre el desarrollo de las competencias clave mediante la incorporación de la programación y la microelectrónica en el flujo de trabajo de quinto y sexto curso de EP en Ciencias Naturales y Ciencias Sociales. Para alcanzar este objetivo general así como los específicos, el PE que se ha desarrollado, dota al profesorado de una herramienta práctica que permite la inclusión de la programación en el aula. Este PE permite al docente acometer de una forma pautada y progresiva una labor que, a priori, resulta compleja. Gracias a esta secuencia de complejidad creciente y, mediante la implementación de actividades lúdicas que favorecen el aprendizaje de la programación, el estudiante se ve abocado a desarrollar, de forma inconsciente, las competencias establecidas en el currículo. Al implicar al alumnado en la resolución de problemas mediante el uso de la lógica de la programación, no solo se desarrolla la competencia matemática, sino también la de resolución de problemas al tiempo que se favorece el trabajo en equipo y por tanto se consigue socializar al alumnado. Además, el hecho de imaginar, proponer, planificar, desarrollar, resolver y reflexionar sobre los problemas planteados permite al estudiante desarrollar su iniciativa, su curiosidad y por ende su creatividad y capacidad artística al tiempo que se fomenta su espíritu emprendedor. Se trata por tanto, de un PE que cumple los objetivos didácticos planteados previamente a su desarrollo.

5.3. PROSPECTIVA

Para terminar este capítulo, en esta sección, se plantean aquellos elementos que en un futuro permitirían la mejora del PE presentado en este TFG. Entre los elementos que deben ser citados, destaca la futura puesta en marcha de este proyecto durante el curso 2015-2016 en un centro privado. Dicha puesta en marcha, ya avalada por la Dirección del mismo, probará la calidad del mismo, certificando si la programación, actividades y la metodología elegidas son tan efectivas como, a priori, parecen.

Por otra parte, antes de la puesta en marcha de un PE como el presentado en este TFG, quedan por resolver dos posibles problemas que dificultarán su desarrollo. A saber, la accesibilidad a la tecnología en el domicilio del alumnado y la formación del profesorado (Dwyer, 2007). Respecto al primero de los inconvenientes, la accesibilidad a la tecnología fuera del centro, las soluciones podrían pasar por favorecer el acceso a Internet desde casa dando charlas a los familiares y responsables del alumnado y asignando tareas al alumnado para realizar desde casa. Respecto del segundo problema, la formación del profesorado, que siempre resulta imprescindible, en este caso

lo es más si cabe. Abordar un PE como el presentado en este TFG requiere de la formación del profesorado implicado, favoreciendo así no sólo la puesta en marcha del proyecto sino también el desarrollo personal y profesional del docente.

Por último, cabe destacar la importancia que la programación supondrá para el futuro de nuestra sociedad. Una sociedad controlada por los que hoy son alumnos en aulas de EP. Alumnos que, sin duda, son enviados a las escuelas con el firme propósito de que desarrollen todo su potencial cognitivo. Como consecuencia de la confianza depositada en la comunidad educativa, los docentes, cara palpable del sistema educativo, no puede dejar pasar la oportunidad de enseñar programación en las aulas. Oportunidad que propiciará la formación de «nativos digitales» plenamente competentes.

REFERENCIAS BIBLIOGRÁFICAS

- Bers, M. (2010). El programa de robótica TangibleK. pensamiento computacional aplicado para niños pequeños. *Investigación y Práctica De La Niñez Temprana*, 12(2)
- Chang, C. (2014). Effects of using alicia and scratch in an introductory programming course for corrective instruction. *Journal Educational Computing Research*, 51(2), 185-204.
- Clements, D. (1986). Effects of logo and CAI environments on cognition and creativity. *Journal of Educational Psychology*, 78(4), 309-318.
- Clements, D. (1991). Enhancement of creativity in computer environments. *American Educational Research Journal*, 28(1), 173-187.
- Dasgupta, S., & Resnick, M. (2014). Engaging novices in programming, experimenting, and learning with data. *ACM Inroads*, 5(4), 72-75.
- Dwyer, J. (2007). Computer-based learning in a primary school: Differences between the early and later years of primary schooling. *Asia-Pacific Journal of Teacher Education*, 35(1), 89-103.
- Habibah, L., & Vasugiammai, M. (2011). Technology integrated teaching in malaysian schools: GIS, a SWOT analysis. *World Journal on Educational Technology*, 3, 64.
- Idlbi, A. (2009). Taking kids into programming (contests) with scratch. *Olympiads in Informatics*, 3, 17-25.
- INE, Instituto Nacional de Estadística. (2014). Encuesta sobre equipamiento y uso de tecnologías de información y comunicación en los hogares. Recuperado el 20 de abril de 2015 de <http://www.ine.es/prensa/np864.pdf>

- Kalelioglu, F., & Gülbahar, Y. (2014). The effects of teaching programming via scratch on problem solving skills: A discussion from learners' perspective. *Informatics in Education*, 13(1), 33-50.
- Ley Orgánica 2/2006, de 3 de mayo, de Educación*. Boletín Oficial del Estado, 106, de 4 de mayo de 2006.
- Ley Orgánica 8/2013, de 9 de diciembre, para la Mejora de la Calidad Educativa*. Boletín Oficial del Estado, 295, de 10 de diciembre de 2013.
- Lifelong Kindergarten Group at the MIT Media Lab. (2014). Hardware that can connect to scratch. Recuperado el 2 de mayo de 2015 de http://wiki.scratch.mit.edu/wiki/Hardware_That_Can_Connect_to_Scratch#PicoBoard
- Marshall, S., & Taylor, W. (2014). ICT adoption and use in training, learning and counselling. *International Journal of Education and Development using Information and Communication Technology*, 10(3), 2.
- Ochsner, A., & Anton, G. (2014). *Talking with kids on game design, computer programming, and taking over the world with dragons*. Unpublished manuscript.
- Orden ECD/65/2015, de 21 de enero donde se definen las relaciones entre las competencias y los contenidos a desarrollar en Enseñanza Primaria*. Boletín Oficial del Estado, 25, de 29 de enero de 2015.
- Pardamean, B., Evelin, & Honni. (2011). The effect of logo programming language for creativity and problem solving. *Proceedings of the 10th WSEAS International Conference on E-Activities*, Jakarta, Indonesia. 151-156.
- Pramling, S., & Asplund, M. (Diciembre 2008). The playing learning child: Towards a pedagogy of early childhood. *Scandinavian Journal of Educational Research*, 52(6), 623.
- Prensky, M. (2001). Digital natives, digital immigrants. *MCB University Press*, 9(5)

Prensky, M. (2014). *No me molestes mamá, estoy aprendiendo*. España: SM Editores.

Real Decreto 126/2014, de 28 de febrero, por el que se establece el currículo básico de la Educación Primaria. Boletín Oficial de Estado, 22, de 1 de marzo de 2014.

Resnick, M. (2007). All I really need to know (about creative thinking) I learned (by studying how children learn) in kindergarten. Paper presented at the Washington DC: Association for Computing Machinery (ACM). 1.

Smith, A. (2010). Dialando: Tangible programming for the novice with scratch, processing and arduino. *6th International Workshop on Technology for Innovation and Education in Developing Countries*, Maputo, Mozambique. 1-4.

BIBLIOGRAFÍA

- Alba, R. (2008). Aprender a programar... ¿desde pequeños? Recuperado el 20 de abril de 2015 de <http://recursostic.educacion.es/observatorio/web/ca/software/programacion/593-rafael-alba>
- Artopoulos, A. (2015). De sadosky a nasdaq: 30 años de instituciones de las tecnologías de la información y la comunicación en argentina. In CICOMRA, Cámara de Informática y Comunicaciones de la República Argentina (Ed.), *El impacto de las TIC en la economía y la sociedad* (pp. 253) Editorial Autores de Argentina.
- Díaz, J., Banchoff, C., Queiruga, C., & Martín, E. (2014). Experiencias de la facultad de informática en la enseñanza de programación en escuelas con software libre. *Congreso Iberoamericano De Ciencia, Tecnología, Innovación y Educación*, Buenos Aires, Argentina.
- Marshall, S., & Taylor, W. (2014). ICT adoption and use in training, learning and counselling. *International Journal of Education and Development using Information and Communication Technology*, 10(3), 2.
- Martínez, R., & García-Beltrán, A. (2000). *Breve historia de la informática*. Unpublished manuscript. Recuperado el 2 de mayo de 2015 de <http://ocw.upm.es/ciencia-de-la-computacion-e-inteligencia-artificial/fundamentos-programacion/otrosrecursos/brevehistoriainformatica.pdf>
- Morrison, N. (2015). Pascal programming for schools. Recuperado el 28 de abril de 2015 de <http://pp4s.co.uk/>
- Ochsner, A., & Anton, G. (2014). *Talking with kids on game design, computer programming, and taking over the world with dragons*. Unpublished manuscript.

- Programamos. (2013). Evidencias científicas de los beneficios de aprender a programar desde infantil. Recuperado el 25 de abril de 2015 de <http://programamos.es/evidencias-cientificas-de-los-beneficios-de-aprender-a-programar-desde-infantil/>
- Programmable brick from LEGO. (1998). Recuperado el 5 de mayo de 2015 de <http://el.media.mit.edu/logo-foundation/pubs/logoupdate/v7n1/v7n1-pbrick.html>
- Randstad. (2015). El sector servicios, turismo e industria impulsarán la creación de nuevos puestos de trabajo durante 2015. Recuperado el 20 de abril de 2015 de <http://www.randstad.es/tendencias360/profesionales-mas-demandados-para-2015>
- Resnick, M. (2012). TED talking: Let's teach kids to code. Recuperado el 8 de mayo de 2015 de https://www.ted.com/talks/lang/es/mitch_resnick_let_s_teach_kids_to_code
- Resnick, M. (2013). Learn to code, code to learn. Recuperado el 2 de mayo de 2015 de <https://www.edsurge.com/n/2013-05-08-learn-to-code-code-to-learn>
- Smith, A. (2010). Dialando: Tangible programming for the novice with scratch, processing and arduino. *6th International Workshop on Technology for Innovation and Education in Developing Countries*, Maputo, Mozambique. 1-4.
- What is logo? (2011). Recuperado el 8 de mayo de 2015 de <http://el.media.mit.edu/logo-foundation/logo/index.html>